
SCRJ Lakers Fan-Engagement-App

Dokumentation

BACHELORARBEIT

Studiengang Informatik
OST - Otschweizer Fachhochschule
Campus Rapperswil-Jona

Frühjahrssemesters 2022

Autoren:	Simon Hager Philipp Emmenegger Joel Schaltegger
Version:	1.0
Datum:	17. Juni 2022
Betreuer:	Prof. Dr. Markus Stolze
Experte:	Markus Flückiger
Gegenleser	Prof. Dr. Daniel Politze
Projektpartner:	Lakers Sport AG

Abstract

Die Lakers Sport AG besitzt bereits eine eigene Webseite und eine mobile App mit Informationen über den Verein. Damit sich Fans intensiver mit dem Verein auseinandersetzen, soll innerhalb des vorliegenden Projekts eine entsprechende Plattform entwickelt werden. Für die Implementierung der Plattform gibt es keine Vorgaben oder Einschränkungen des Vereins und des Betreuers. Das Projektteam evaluiert selbstständig geeignete Lösungen und beschliesst in enger Zusammenarbeit mit dem Kunden das weitere Vorgehen.

In der Konzeptionsphase wurde zusammen mit der Lakers Sport AG die Idee eines interaktiven Tippspiels ausgearbeitet. Darin sollen Fans Voraussagen über zukünftige Spielereignisse machen können. Richtige Antworten werden mit Punkten und der Teilnahme an einem Gewinnspiel belohnt. Um die User Experience auf die Bedürfnisse der Fans abzustimmen, wurde ein *User-centered-Design*-Ansatz angewendet. Die daraus entstandenen Anforderungen dienten als Ausgangslage für den Softwarearchitektur-Entwurf. In einem iterativen Prozess wurde anschliessend anhand des evaluierten Prototypens das Endprodukt entwickelt.

Das Backend des Tippspiels besteht aus verschiedenen *Amazon Web Services (AWS)*-Services. Durch die Verwendung einer cloudbasierten Serverless-Architektur lässt sich die Applikation skalieren, sobald sich die Last verändert. Die Infrastruktur erfährt an Spieltagen kurzfristig eine grosse Belastung. Dank dem Einsatz von FaaS-Komponenten muss nur die Leistung bezahlt werden, die auch effektiv benötigt wird. Um die Bandbreite an AWS-Services besser verwalten zu können, wurde das *AWS Amplify Command Line Interface (CLI)* eingesetzt. Damit wurde unter anderem die API mit *GraphQL* aufgesetzt. Mit diesem Ansatz sind Realtime-Updates möglich, welche zuvor als Anforderung festgelegt wurden.

Das mit React implementierte Frontend wird als *Progressive Web Application (PWA)* ausgeliefert. Der integrierte *Service Worker* gewährleistet die Installierbarkeit und Push-Benachrichtigungen. Die Fans können die Applikation entweder anonym nutzen oder sich ohne Passwort mit ihrer E-Mail sowie über externe Login-Provider authentifizieren. Anonym abgegebene Tipps werden bei einer späteren Registrierung in das persönliche Profil übernommen. Das Tippspiel wird durch einen Regiebereich von einem Administrator gesteuert. Die Bedienung ist in einer Benutzeranleitung beschrieben.

Die Leistungsfähigkeit der Fan-Engagement-App bei grösserer Last wird mit API- und End-to-end Tests überprüft. In einem Crowd Test wird die App von vielen Personen gleichzeitig zu einem realitätsnahen Szenario benutzt, um den zukünftigen produktiven Einsatz zu simulieren.

Danksagung

Markus Stolze danken wir für die Betreuung, die guten Ratschläge und wertvolle Unterstützung.

Mirko Stocker danken wir für den fachlichen Erfahrungsaustausch bezüglich Cloud Computing und AWS Amplify.

Michael Gfeller danken wir für die ausführlichen und konstruktiven Code Reviews.

Frieder Loch danken wir für grosszügige Bereitstellung seiner Vorlesung zugunsten unseres Crowd Tests.

Michael Bart, stellvertretend für die Lakers Sport AG, danken wir für die gute Zusammenarbeit und Unterstützung.

Marius Zindel danken wir für die Latex Dokumentenvorlage.

Disclaimer

Gender

In der vorliegenden Arbeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Es werden das generische Maskulinum verwendet, wobei beide Geschlechter gleichermaßen gemeint sind.

Hervorhebungen

Zur besseren Lesbarkeit werden gewisse Begriffe in den Fliesstexten dieser Bachelorarbeit hervorgehoben. Dabei wird zwischen zwei Formatierungen unterschieden.

Kursive Begriffe

Nicht allgemein bekannte Begriffe und Abkürzungen werden im Text *kursiv* dargestellt und in einem Glossar erklärt.

Beispiel: Ein *User-centered-Design*-Ansatz wird verfolgt.

Einfache Guillements

Dateinamen, Funktionsnamen, E-Mail-Adressen, URLs und selbst deklarierte Namen werden mit <einfachen Guillements> gekennzeichnet.

Beispiel: In der Datei <disclaimer.tex> wird die Semantik der Dokumentation beschrieben.

Inhaltsverzeichnis

1. Management Summary	7
1.1. Ausgangslage	7
1.2. Vorgehen	7
1.3. Ergebnisse	7
1.4. Ausblick	8
2. Einführung	9
2.1. Ausgangssituation	9
2.2. Thematik	9
2.3. Vision	11
3. Anforderungsspezifikation	12
3.1. Aktoren	12
3.2. User Journey	12
3.3. Storyboard	13
3.4. Use Cases	14
3.5. Nichtfunktionale Anforderungen	27
3.6. Nicht-Anforderungen	31
3.7. Domain Model	32
4. Design	34
4.1. Systemarchitektur	34
4.2. Konzepte	35
4.3. User Interfaces	46
4.4. Externe Schnittstellen	49
5. Implementierung	51
5.1. Frontend	52
5.2. Backend	55
5.3. Deployment	71
5.4. Einschränkungen	74
5.5. Qualitätssicherung	77
6. Evaluierung	79
6.1. Pilot Test	79
6.2. Crowd Test	79
6.3. Usability Test	80
6.4. Last- und Performancetests	91
6.5. Engagement Test	100
6.6. Systemtest	104
7. Weiterentwicklung	105
7.1. Challenge-Typen	105
7.2. Automatisierung	105
7.3. Ausbau der Push-Benachrichtigungen	106
7.4. Newsletter-Registrierung	106
7.5. Laufzeit-Optimierung Lambdas	106
7.6. Passwordless-SMS-Login	107
7.7. Ausbau der Gamification	107
7.8. Aufwandschätzung	108
8. Schlussbericht	109
A. Testprotokolle	110
A.1. Usability Tests	110

A.2. Crowd Test	122
A.3. Engagement Test	127
A.4. API Tests	128
A.5. Kostenanalyse	130
A.6. Code reviews	131
A.7. System Test 15.06.2022	136
B. Projektplan	141
B.1. Einführung	141
B.2. Projektorganisation	141
B.3. Zeitmanagement	143
B.4. Meilensteine	144
B.5. Iterationen (Sprints)	146
B.6. Besprechungen	146
B.7. Risikomanagement	147
B.8. Arbeitspakete	152
B.9. Qualitätsmassnahmen	152
C. Projektmonitoring	155
C.1. Projektstruktur	155
C.2. Zeitauswertung	156
D. Verzeichnisse	159
E. Weiterführende Dokumente	168
E.1. Admin Panel Anleitung	168
E.2. Storyboard Visualisierung	184
E.3. Figma Prototyp Entwurf	216
E.4. Aufgabenstellung	219
E.5. Besprechungsbeschlüsse	221
E.6. GitLab Issues	223
F. Präsentationen	229
F.1. Crowd Test	230
F.2. Zwischenpräsentation	242
F.3. SCRJ meets OST	253

1. Management Summary

1.1. Ausgangslage

Die Lakers Sport AG ist ein professioneller Eishockeyverein mit einer Mannschaft in der höchsten Liga der Schweiz. Ihr Stadion ist nur wenige Fussminuten vom Standort Rapperswil der Ostschweizer Fachhochschule (OST) entfernt. Daher war es naheliegend, eine Zusammenarbeit (‹SCRJ meets OST›) zu starten, von der beide Parteien profitieren sollten. In diesem Rahmen wurde die Idee einer Fan-Engagement-App geboren, welche in dieser Bachelorarbeit von Informatikstudierenden entwickelt werden soll.

Fan-Engagement ist ein Prozess, der das Ziel verfolgt, die Interaktion zwischen den Fans und dem Verein zu steigern und die Bindung der Fans an die Sportorganisation zu stärken. Damit ist es ein treibender Faktor für ein umsatzstarkes Sportunternehmen.

1.2. Vorgehen

Im Zuge der App-Entwicklung wurde darauf geachtet, die Anforderungen mit einem maximalen Mehrwert für den Kunden zu erfüllen. Das Konzept der App sollte abwechslungsreich sein. Darüber hinaus sollte das Produkt einfach wartbar, gut erweiterbar sein sowie einsatzbereit übergeben werden. Zudem sollte die Infrastruktur kostengünstig und zweckerfüllend gewählt werden.

Um den sehr umfassenden Begriff Fan-Engagement für dieses Projekt einzugrenzen, trafen sich die Studierenden zunächst mit dem Marketingleiter der Lakers Sport AG. In diesem Zusammenhang entstand das Konzept eines Tippspiels, welches die Zuschauer und Sponsoren für den Verein begeistern soll. Anschliessend wurde eine detaillierte Version der Idee visualisiert und von Kundenseite bestätigt. Darin waren bereits erste Skizzen von Bildschirmen enthalten. Die Skizzen wurden anschliessend als Inspiration verwendet, um einen interaktiven Prototypen der Fan-Engagement-App zu entwickeln. Mit dem Prototypen konnte bereits ein erstes Feedback von potenziellen Nutzern eingeholt werden. Die daraus gesammelten Erkenntnisse wurden benutzt, um die Bedienung zu vereinfachen und somit die Benutzerfreundlichkeit zu verbessern.

Aus dem resultierten Design der Benutzeroberflächen wurde ein Produkt entwickelt, welches die Minimalanforderungen erfüllte. Anschliessend startete ein iterativer Prozess, bei dem regelmässig neue Features implementiert wurden, um zusätzliche Anforderungen abzudecken. Der Fortschritt wurde stetig getestet, optimiert und mit dem Kunden validiert.

1.3. Ergebnisse

Die Fan-Engagement-App ist der offiziellen Lakers-Domain angegliedert. Die installierbare Webseite ist für mobile Geräte ausgelegt, kann aber auch über Desktop-Computer genutzt werden. Ein Administrator kann Fragen mit einem Spieler aus dem Kader als Antwortmöglichkeit veröffentlichen und die Fans können ihren Tipp dazu abgeben. Für richtige Antworten erhalten die Fans Punkte. Fans können sich mit einer persönlichen E-Mail-Adresse ohne Passwort oder mit einem bestehenden Social-Media-Account im Tippspiel anmelden. Falls sie zum Spielende ein Benutzerkonto angelegt haben, nehmen sie automatisch an der Verlosung von Preisen teil. Bei der ersten Anmeldung kann der Fan selbst entscheiden, ob sein Benutzername in der öffentlichen Rangliste dargestellt wird, oder nicht. Zudem kann er in seinem Benutzerprofil die Statistiken zu eigenen Tippspiel-Aktivitäten einsehen.

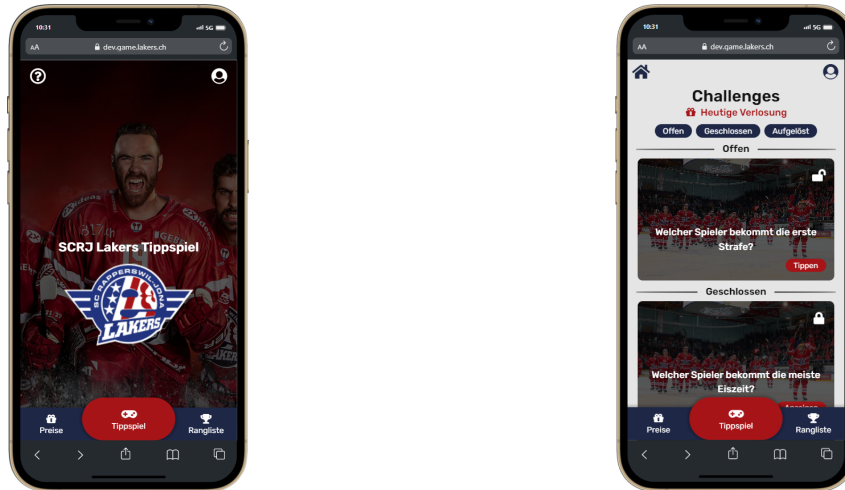


Abbildung 1: Screenshots Fan-Engagement-App

In einem separaten Bereich kann die Fan-Engagement-App von der Regie mit einem Administrator konfiguriert werden. Konkret kann dieser die Fragen, Sponsoren, Preise und das Kader erfassen sowie Tippspiele vorbereiten und verwalten. Aktionen, die er in diesem Bereich tätigt, werden auf den Bildschirmen der Nutzer in Echtzeit angezeigt. Für diese Tätigkeiten erhält die Lakers Sport AG eine Anleitung, die den gesamten Funktionsumfang und die Prozesse beschreibt.

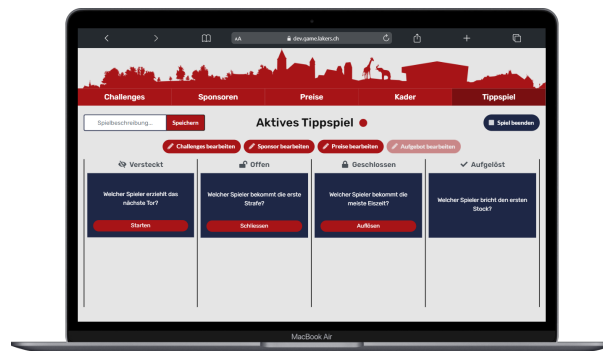


Abbildung 2: Screenshot Admin-Bereich

1.4. Ausblick

Nach Übergabe übernimmt das *Institut für Software (IFS)* der OST Rapperswil die Wartung und Weiterentwicklung der App. Die Eishockey-Saison 2021/2022 der National League war etwas zu kurz, um die App während eines Eishockeyspiels im Stadion der SCRJ Lakers testen zu können. Ausserdem entstanden im Verlauf der Arbeit viele Ideen und Visionen, welche aufgrund des begrenzten Rahmens der Bachelorarbeit nicht weiterverfolgt werden konnten. Das mittelfristige Ziel besteht darin, die Fan-Engagement-App zum Saisonstart im September 2022 zu lancieren. Bis dahin verbleiben noch einige Monate, in denen die Vision des Projektteams weiterentwickelt werden kann.

2. Einführung

Die Lakers Sport AG ist der Projektpartner und somit der Kunde des Fan-Engagement-Apps. Im Verlauf der Arbeit ist ebenfalls von den SCRJ Lakers die Rede, womit der Sportverein gemeint ist.

2.1. Ausgangssituation

Die SCRJ Lakers besitzen bereits eine eigene Webseite und eine Mobile App mit Informationen über den Verein, die Teams und dessen Spiele. Diese beiden digitalen Plattformen sind die Anlaufstellen für einen Spielbesuch im SCRJ Stadion. Beispielsweise lassen sich darüber Tickets kaufen, kann sich in die aktuelle sportliche Situation einlesen und sich mit neuen Fan-Artikeln eindecken. Die App hat aktuell mehr als 1000 Downloads im Google Play Store. [1]

2.2. Thematik

Auch der Profisport hat die Vorteile der Digitalisierung erkannt und versucht diese für sich zu nutzen. Obschon die Präsenz im Internet für den Sport immer bedeutsamer werden, dürfen die physischen Erlebnisse und echten Emotionen nicht verloren gehen. Diese Werte sollen auch in dieser App bewahrt werden und so einen Beitrag zu höherem Fan-Engagement leisten.

2.2.1. Fan-Engagement

«Fan engagement is a specific form of customer engagement in the sport context. In marketing, researchers define customer engagement as a consumer's spontaneous, interactive, and cocreative behaviors primarily in nontransactional consumer-company exchanges to achieve his or her individual and social purposes»[2]

Kurzgesagt umfasst Fan-Engagement jegliche emotionale, kognitive und verhaltensbezogene Aspekte, die in der Interaktion zwischen einem Kunden, in diesem Kontext dem Fan, und einer Organisation im Sportumfeld auftreten.

Ziel

Das generelle Ziel von Fan-Engagement ist eine starke nachhaltige Fan-Bindung an einen Verein oder an ein Team. Je stärker sich ein Fan an den Verein gebunden fühlt beziehungsweise je zahlreicher seine Interaktion mit dem Verein ist, desto tiefer verankert sich die Bindung mit dem Verein und er wird vermehrt Spiele besuchen, andere dazu einladen und positiv darüber berichten. Diese Fans sind konsumfreudiger und sorgen mit Mund-zu-Mund-Propaganda für kostenlose und vertraute Werbung.[3]

Für die SCRJ Lakers ist Fan-Engagement beziehungsweise dieses Projekt eine von mehreren Massnahmen, um den Zuschauerzulauf und Konsum an Heimspielen sowie den Ertrag aus eigenen Produkten zu steigern. Zusätzlich möchte man mehr Informationen über die Fans gewinnen und attraktiver für Sponsoren werden. Neue Konzepte und Anreize sind notwendig. Denn die Zuschauerzahlen stagnieren beziehungsweise sind sogar leicht rückläufig. So entspricht der Schnitt der abgelaufenen Saison in etwa jenem der Saison 2019/20 (4218), in der ebenfalls mehrheitlich ohne Einschränkungen gespielt werden konnte.[4]

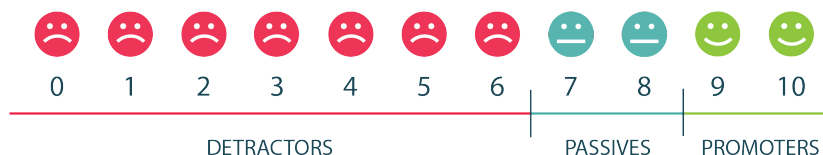
Phase	Heimspiele	Minuskulisse	Rekordkulisse	Zuschauerschnitt
Qualifikation	26	2647	5235	3927
Playoff	4	6100	6100	6100

Tabelle 1: Zuschaueraufkommen an Heimspielen der Saison 2021/22[5]

Messbarkeit

Das Marketing der SCRJ Lakers hat selber keine *Spezifische, Messbare, Attraktive, Realistische und Terminierte (SMART)*-Ziele, welche den Erfolg der Fan-Engagement-Projekte beurteilen. Deshalb wird der aus dem Marketing bekannten *Net Promoter Score* zur Validierung der Resultate vorgeschlagen. Damit lässt sich der Erfolg von Fan-Engagement ansatzweise quantitativ messen. Dabei wird dem Kunden eine einfache Frage gestellt. Einige Beispiel sind[6]

- Wie wahrscheinlich ist es, dass du eine Kollegin/ein Kollege zu einem Lakers Heimspiel einlädst?
- Wie wahrscheinlich ist es, dass du mit einer Kollegin/einem Kollegen über die Lakers/diesen Spielbesuch der Lakers sprichst?



$$\text{😊 \%} - \text{😞 \%} = \text{NET PROMOTER SCORE}$$

Abbildung 3: Net Promoter Score[7]

Die Befragten beantworten die Frage auf einer Bewertungsskala von 0 (komplett unwahrscheinlich) bis 10 (sehr wahrscheinlich). Anschliessend können diese in drei Gruppen eingeteilt werden. Eine Bewertung von 9 oder 10 geben die Promoter ab. Diese werden höchst wahrscheinlich ungezwungen Werbung betreiben und somit das Produkt anderen weiterempfehlen. Ihre Bindung zum Verein ist genügend stark, sodass sie auch schwierige Zeiten übersteht. Die passiven Fans sind zwar relativ zufrieden, tragen diese Zufriedenheit jedoch nicht in die Welt hinaus. Ihre Anteilnahme kann durch den sportlichen (Miss-)erfolg stark beeinflusst werden und sie sind bereit, die Zugehörigkeit unter Umständen zu wechseln. Den Abschluss machen die Bewerter von 0 bis 6. Sie sind unzufrieden mit dem Produkt und sind eine Gefahr für den Verein, da sie möglicherweise ihre negativen Erfahrungen mit anderen teilen. Aus den prozentualen Anteilen lässt sich letztendlich der *Net Promoter Score* ableiten. Diese Messmethode hat auch ihre Schwäche. So ist dessen Aussagekraft stark beschränkt. Es fehlt die Begründung der Bewertung und ist stark von der subjektiven Anwendung der Skala abhängig. Deshalb sollte man sich nicht alleine auf diese Messung verlassen.[8]

2.2.2. Gamification

Unter dem Begriff Gamification versteht man spieltypische Elemente in einer spielfremden Applikation. Dieses Konzept verbessert das Nutzungserlebnis einer App mit den gleichen motivierenden Techniken, die Menschen dazu bringen, Spiele zu lieben. Da das Eishockeyspiel ein zentraler Bestandteil der Fan-Engagement-App ist, soll auch die App den Charakter eines Spiels bekommen und die gleichen Emotionen hervorrufen können.

Für die Ausarbeitung der geeigneten Massnahmen wird ein Paper zum Thema Gamification in Sport-Apps herangezogen. Darin heisst es

«Specifically, based on the empirical results obtained, it can be concluded that interacting with achievement-related game elements (e.g. badges, results, progress bars, rankings and difficulty levels) while using gamified sport apps facilitates the satisfaction of the three psychological needs of competence, autonomy and relatedness.»[9]

und später konkret auf die Verbundenheit bezogen

«Finally, elements such as rankings evoke a greater perception of relatedness with others, because players thereby can compare their performance with that of other users.»[9]

Es gibt also unterschiedliche Reize, die man mit bestimmten Elementen erzeugen kann. In diesem Paper werden auch noch andere Möglichkeiten genannt, aber für den Rahmen dieses Projektes lassen sich aus den beiden Zitaten bereits genügend Funktionalitäten ableiten.

- Badges für Meilensteine und spezielle Ereignisse
- Resultate beziehungsweise Punkte für richtig gelöste Aufgaben
- Rangliste mit allen Teilnehmern und ihren Punkten
- Belohnungen für die besten Teilnehmer

Die gewonnenen Erkenntnisse werden in den Use Cases, Anforderungen und damit im Produkt angewendet.

2.3. Vision

Die SCRJ Lakers Fan-Engagement-App ist ein grosser Mehrwert für die Besucher eines Heimspiels der SCRJ Lakers. Sie verstärkt das Erlebnis an der Sportveranstaltung und schafft dadurch stärkere Anreize für häufigere Spielbesuche.

Die Spielbesucher des SCRJ Lakers können über eine App an Challenges während den Heimspielen des SCRJ teilnehmen. Challenges können einfache Fragen zum Spielgeschehen oder zu Statistiken sein. Gewinner dieser Challenges nehmen an der Verlosung von Preisen teil, welche gegen physische Gutscheine eingelöst werden können. Resultatvergleiche mit Kollegen und Bekannten animieren zum fortlaufenden gegenseitigen übertreffen bisheriger Höchstwerte.

Die starke Nutzung der neugeschaffenen Fan-Engagement-App und die verstärkte Interaktion mit den Besuchern schafft neue spannende Werbemöglichkeiten für Partner und Sponsoren der SCRJ Lakers. Partner und Sponsoren präsentieren sich visuell und sind sogar wesentlicher Bestandteil des Spiels. Diese App generiert dem Verein höhere Einnahmen und trägt zu einer nachhaltigen Umsatzsteigerung bei.

3. Anforderungsspezifikation

3.1. Aktoren

Aktor	Beschreibung
Besucher	Zu den Besuchern zählen Fans der SCRJ Lakers und andere Zuschauer in der St.Galler Kantonalbank Arena. Die primäre Zielgruppe der App sind die Fans der SCRJ Lakers. Die Besucher sind die mehrheitlichen Nutzer der App und sollen einen Mehrwert zur eigentlichen Sportveranstaltung, dem Heimspiel der SCRJ Lakers, erhalten. Besucher beantworten die veröffentlichten Challenges und erhalten dafür eine Belohnung in Form von Punkten, Auszeichnungen und Preisen
Administrator	Der Administrator verwaltet die Fragestellungen, definiert die Belohnungen und schaltet diese zu gegebenem Zeitpunkt frei. Ein freiwilliger Helfer wird diese Rolle einnehmen.
Mitarbeiter Kasse	Ein Mitarbeiter an der Kasse wandelt die erworbenen Preise in gültige physische Gutscheine um.

Tabelle 2: Aktoren

3.2. User Journey

Eine User Journey unterstützt das Verständnis für die Zielgruppe des Produktes. Es macht die Kommunikation effizienter und minimiert Streuverluste. Für die SCRJ Fan-Engagement-App wurde eine User Journey mit Schwerpunkt Gaming herangezogen.

3.2.1. Besucher

Schritt	Touchpoints	Erklärung
1 Awareness	Social Media Kollegen Werbung vor Stadion Webseite Flyer	Die Existenz der Fan-Engagement-App wird auf verschiedenen Portalen der SCRJ Lakers beworben. Einzelne Plakate und Flyer vor dem Stadion machen darauf aufmerksam.
2 Consideration	Videowürfel Stadiondurchsage Benachrichtigung	Der Videowürfel im Stadion fordert die Besucher auf, eine vorbereitete Challenge in der App zu lösen. Der Speaker untermalt die Visualisierung auf dem Würfel mit einer entsprechenden Durchsage. Besucher im Stadion, welche die App bereits nutzen, erhalten zusätzlich eine Benachrichtigung auf ihrem Smartphone.
3 Play	App	Der Besucher öffnet die App und navigiert zur Challenge-Übersicht. Challenges können konkrete Fragen zu einer nächsten Spielsituation, statistische Werte und Resultate sein. Der Besucher diskutiert in der Gruppe über die richtige Lösung und gibt letztendlich seine Antwort ein. Nach Abschluss der Challenge wird die Lösung präsentiert und die gewonnenen Punkte gutgeschrieben. Ist das Spiel zu Ende wird dem Benutzer möglicherweise eine Belohnung übertragen.

4 Share	Social Media	Der Besucher teilt seine Resultate mit seinen Bekannten und Kollegen.
---------	--------------	---

Tabelle 3: User Journey eines Besuchers

3.3. Storyboard

Die folgenden Storyboards beschreiben einen Arbeitstag der Fan-Engagement-App. Sie basieren auf einer visualisierten Version in einem *Miro*-Board, welches genutzt wird um das Konzept und die funktionalen Anforderungen mit dem Kunden erarbeiten zu können. Darin abgebildet sind grobe Abläufe und Wireframes, auf welchen der Prototyp aufgebaut wird. Das visualisierte Storyboard befindet sich im Anhang (Kapitel E.2).

3.3.1. Game erstellen - optional

Administrator

Es sind zwei Stunden vor Spielbeginn. Mike führt heute Regie und bereitet sich auf das Spiel vor. Er ist auch verantwortlich, das digitale Spiel für die Besucher in der Fan-Engagement-App zu betreuen. Dafür muss er zuerst ein neues Game konfigurieren und dieses mit Challenges und Belohnungen füllen.

Schritt	Erklärung
1 App öffnen	Mike öffnet die App im Browser.
2 Login	Mike authentifiziert und autorisiert sich als Administrator.
3 Game erstellen	Mike erstellt ein neues Game und ergänzt die fehlenden Informationen.
4 Neue Challenge einrichten	Mike richtet eine neue, noch fehlende Challenge ein. Dafür navigiert er zur Challenge-Übersicht. Er schreibt die Fragenstellung und die zu gewinnenden Punkte hin. Die Challenge wird gespeichert.
5 Challenges zuweisen	Mike fügt dem Game die heute zu spielenden Challenges hinzu.
6 Belohnungen zuweisen	Mike fügt dem Game die heutigen bereits existierenden Belohnungen hinzu.

Tabelle 4: Schritte des Administrators zur Erstellung eines Spiels

Das Game ist nun für den Einsatz vorbereitet und das Heimspiel kann beginnen.

3.3.2. Challenge (vorbereiten und) durchführen

Administrator

Während dem Spiel werden die vorbereiteten Challenges von Mike, dem heutigen App-Verantwortlichen, gestartet und später wieder abgeschlossen. Für eine hohe Nutzfrequenz werden die Besucher über den Videowürfel visuell und per Mikrofon akustisch zum Lösen der Challenge aufgefordert. Der Spielverlauf passt zu einer noch nicht veröffentlichten Challenge. Mike fügt dem Game weitere Challenges spontan hinzu und lässt diese von den Besuchern spielen.

Schritt	Erklärung
1 App öffnen	Mike öffnet die App im Browser.
2 Login	Mike authentifiziert und autorisiert sich als Administrator.

3 Challenge aktivieren	Mike schaltet die Challenge frei.
4 Challenge schließen	Mike beendet die Challenge. Die Lösung wird aufgeschaltet und Punkte verteilt.

Tabelle 5: Schritte des Administrators zur Durchführung einer Challenge

Besucher

Alexander besucht an diesem Abend mit seinem Studienkollege das Heimspiel der SCRJ Lakers. Im ersten Spielabschnitt wird er auf dem Videowürfel zur Teilnahme an einem digitalen Spiel aufgefordert, worin es etwas zu gewinnen gibt. Er fühlt sich davon angesprochen und fordert sein Glück heraus.

Schritt	Erklärung
1 App öffnen	Alexander öffnet die App auf dem Smartphone.
2 Challenge lesen	Alexander kann die Fragestellung der Challenge lesen.
3 Challenge lösen	Dem Besucher werden Antwortmöglichkeiten beziehungsweise ein Antwortfeld angezeigt, wo er seine Lösung eingeben und bestätigen kann.
4 Challenge auflösen	Nach Ablauf der Frist wird die Challenge aufgelöst und die Lösung dargestellt.
5 Identifizierung	Alexander wird zur Anmeldung aufgefordert, um seinen Spielstand zu speichern und am Tippspiel teilzunehmen.

Tabelle 6: Schritte des Besuchers zur Durchführung einer Challenge

3.3.3. Tippspiel abschliessen - Gutschein erhalten**Besucher**

Je nach Ausgang des Tippspiels erhält der Besucher eine Belohnung. Alexander hat richtig getippt und ist zufälligerweise der glückliche Gewinner eines Tagespreises. In der App wird das weitere Vorgehen im Gewinnfall beschrieben. Alexander macht sich auf zur Abendkasse, um die digitale Belohnung in einen echten Gegenwert einzulösen.

Schritt	Erklärung
1 App öffnen	Alexander öffnet die App auf dem Smartphone.
2 Belohnung zeigen	Alexander zeigt die Belohnung in der App dem Mitarbeiter an der Kasse.
3 Gutschein erhalten	Alexander erhält einen Gegenwert zur Belohnung in der App.
4 Belohnung einlösen	Alexander markiert die Belohnung als eingelöst, womit weitere Ansprüche verfallen. Dieser Status wird dem Mitarbeiter erneut gezeigt.
5 Belohnung abholen	Alexander erfreut sich am Geschenk

Tabelle 7: Schritte des Besuchers zur Einlösung einer gewonnenen Belohnung

Ein rundum gelungener Abend für Alexander, obschon die SCRJ Lakers unglücklich verloren haben. Er hatte trotzdem seinen Spass und darf sich nun an seinem Ersatzgewinn erfreuen.

3.4. Use Cases

Für eine erste Übersicht der funktionalen Anforderungen werden Uses Cases erarbeitet. Diese sollen Akteuren und deren Rolle in den Prozessen der Fan-Engagement-App erläutern.

3.4.1. Use Case Diagramm

In folgendem Use Case Diagram werden alle beschriebenen Use Cases und deren Beziehung mit den jeweiligen Akteuren visualisiert. Zusätzlich werden Abhängigkeiten zwischen den Use Cases dargestellt. Die Unterscheidung von *Minimum Viable Product (MVP)* spezifischen Use Cases und solchen, die sich auf optionale Funktionalitäten beziehen, ist farblich ersichtlich.

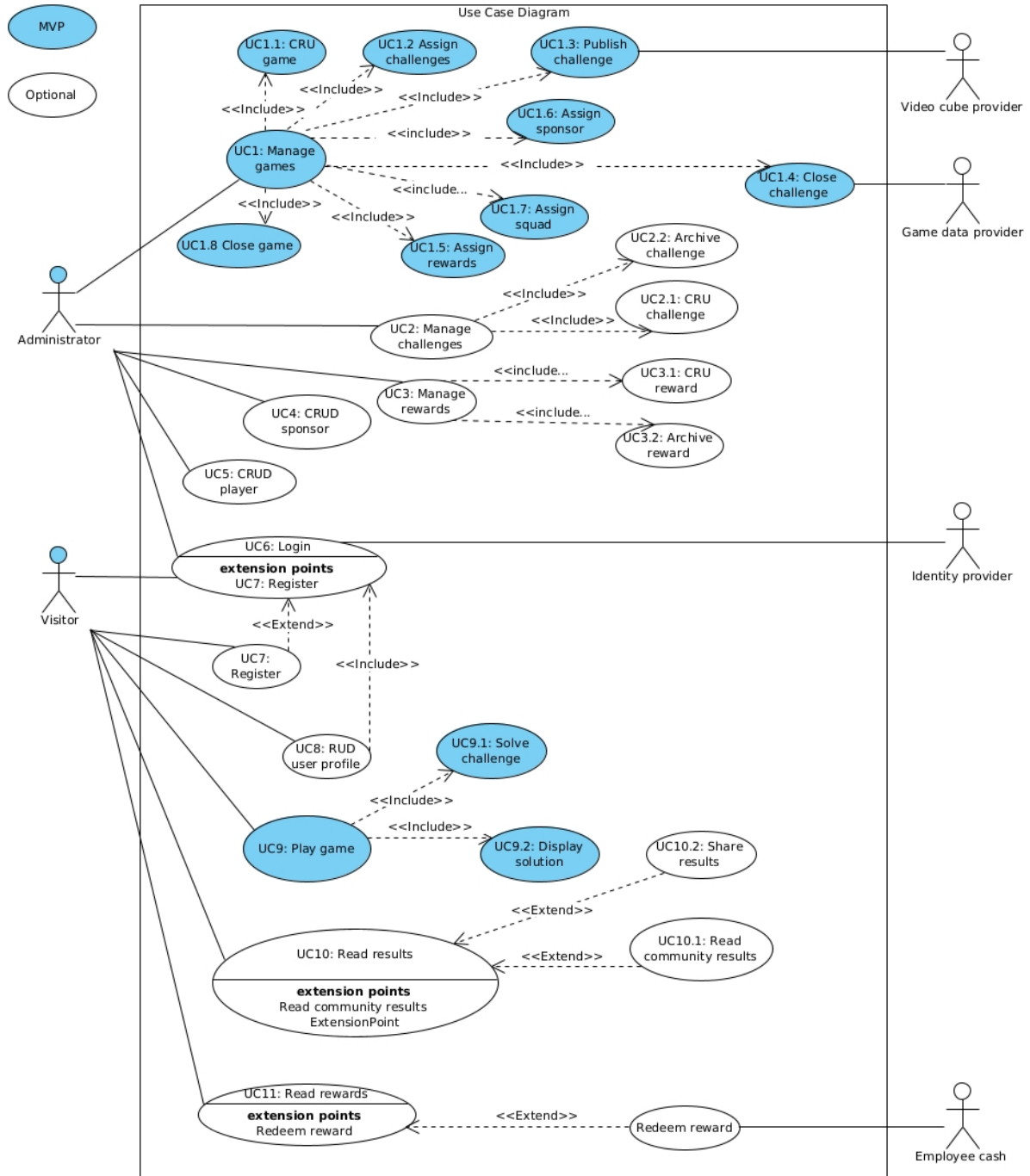


Abbildung 4: Use Case Diagramm

3.4.2. Use Cases

Use Case 1: Manage game

Beinhaltet sämtliche Aktivitäten im Lebenszyklus eines Games.

Abschnitt	Kommentar
Name	UC1.1: CRU game
Beschreibung	Ein eingeloggter Benutzer kann auf der Game-Übersicht neue Games erstellen sowie bestehende Games anzeigen und bearbeiten. Ein Game beinhaltet verschiedene Challenges, Rewards, Sponsoren und das Aufgebot.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Games verwalten
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Game-Daten sind sowohl in der Endbenutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ul style="list-style-type: none"> • C: Benutzer navigiert auf Game-Übersicht und erstellt ein neues Game. • R: Benutzer navigiert auf Game-Übersicht und sieht dessen Details. • U: Benutzer navigiert auf Game-Übersicht. Der Benutzer bearbeitet die Daten und speichert seine Änderungen.
Alternative Flüsse	-

Tabelle 8: Use Case 1.1 CRU Game

Abschnitt	Kommentar
Name	UC1.2: Assign challenges
Beschreibung	Für ein Game können verschiedene Challenges ausgewählt werden.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Challenge zuweisen
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Game-Challenge-Zuweisungen sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer selektiert eine Challenge 3. Benutzer speichert seine Änderungen und weist die Challenge so einem Game zu

Alternative Flüsse	-
--------------------	---

Tabelle 9: Use Case 1.2 Assign challenges

Abschnitt	Kommentar
Name	UC1.3: Publish challenge
Beschreibung	Ein eingeloggter Benutzer kann in der Game-Detailansicht eine ausgewählte Challenge starten.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Challenge freischalten
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Besucher kann freigeschaltete Challenges einsehen und spielen.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer startet eine Challenge
Alternative Flüsse	-

Tabelle 10: Use Case 1.3 Publish challenge

Abschnitt	Kommentar
Name	UC1.4: Close challenge
Beschreibung	Ein eingeloggter Benutzer kann in der Game-Detailansicht eine freigeschaltete Challenge beenden. Dafür muss die Lösung ergänzt beziehungsweise bestätigt werden. Anschliessend wird die Lösung publiziert.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Challenge beenden
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Besucher kann freigeschaltete Challenge nicht mehr spielen.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer beendet eine aktuell laufende Challenge
Alternative Flüsse	-

Tabelle 11: Use Case 1.4 Close challenge

Abschnitt	Kommentar
Name	UC1.5: Assign rewards
Beschreibung	Für ein Game können eine oder mehrere Rewards ausgewählt werden. Diese werden nach Abschluss eines Games unter den besten Besuchern oder nach Zufallsprinzip verteilt.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Rewards zuweisen
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Game-Rewards-Zuweisungen sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer selektiert einen Reward 3. Benutzer speichert seine Auswahl
Alternative Flüsse	-

Tabelle 12: Use Case 1.5 Assign rewards

Abschnitt	Kommentar
Name	UC1.6: Assign sponsor
Beschreibung	Für ein Game kann ein Sponsor ausgewählt werden. Dieses wird beim aktiven Game den Besuchern dargestellt.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Sponsor zuweisen
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Game-Sponsor-Zuweisungen sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer selektiert einen Sponsor 3. Benutzer speichert seine Auswahl
Alternative Flüsse	-

Tabelle 13: Use Case 1.6 Assign sponsor

Abschnitt	Kommentar
Name	UC1.7: Assign squad
Beschreibung	Für ein Game kann das Kader vom Spieltag ausgewählt werden. Die ausgewählten Spieler dienen bei den Challenges als Antwortmöglichkeit.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Kader zuweisen
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Game-Kader-Zuweisungen sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer selektiert die Spieler 3. Benutzer speichert seine Auswahl
Alternative Flüsse	-

Tabelle 14: Use Case 1.7 Assign squad

Abschnitt	Kommentar
Name	UC1.8: Close game
Beschreibung	Nachdem alle Challenges eines Games gespielt wurden, sollte das Game beendet werden. Damit werden die vorgesehenen Rewards verteilt. Dies geschieht basierend auf einer Zeitangabe oder manuell durch den Administrator.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Game beenden
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Game ist sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich. Die vorgesehenen Rewards werden an die aktiven, erfolgreichen und registrierten Besucher verteilt.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die Game-Detailansicht 2. Benutzer beendet das Game
Alternative Flüsse	<p>Automatisches beenden</p> <p>Der Administrator hat den Zeitpunkt für das Game-Ende bereits definiert. Basierend auf dieser Angabe wird das Game automatisch beendet.</p>

Tabelle 15: Use Case 1.8 Close game

Use Case 2: Manage challenges

Beinhaltet sämtliche Aktivitäten im Lebenszyklus einer Challenge.

Abschnitt	Kommentar
Name	UC2.1: CRU challenge
Beschreibung	Ein eingeloggter Benutzer kann auf der Challenge-Übersicht neue Challenges erstellen sowie bestehende Challenges anzeigen und bearbeiten.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Challenges verwalten
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Challenge-Daten sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ul style="list-style-type: none"> • C: Benutzer navigiert auf Challenge-Übersicht. Benutzer erstellt eine neue Challenge mit dem passendem Challenge- und schreibt die Aufgabenstellung hin. • R: Benutzer navigiert auf Challenge-Übersicht. Benutzer navigiert zu einer bestimmten Challenge und sieht dessen Details. • U: Benutzer navigiert auf Challenge-Übersicht. Benutzer navigiert zu einer bestimmten Challenge und wechselt in den Bearbeitungsmodus. Der Benutzer bearbeitet die Daten und speichert seine Änderungen.
Alternative Flüsse	-

Tabelle 16: Use Case 2.1 CRU challenge

Abschnitt	Kommentar
Name	UC2.2: Archive challenge
Beschreibung	Ein eingeloggter Benutzer kann eine bestehende Challenge archivieren, damit vergangene Spiele mit dieser Challenge nicht ihre Wertigkeit verlieren. Archivierte Challenges können nicht mehr gespielt werden.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Challenges archivieren Besucher: Will seinen Fortschritt nicht verlieren
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Challenge-Daten sind sowohl in der Endbenutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.

Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf Challenge-Übersicht. 2. Benutzer navigiert zu einer bestimmten Challenge und archiviert diese.
Alternative Flüsse	-

Tabelle 17: Use Case 2.2 Archive challenge

Use Case 3: Manage rewards

Abschnitt	Kommentar
Name	UC3.1: CRU reward
Beschreibung	Ein eingeloggtter Benutzer kann auf der Rewards-Übersicht neue Rewards erstellen sowie bestehende Rewards anzeigen und bearbeiten.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Rewards verwalten
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Reward-Daten sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ul style="list-style-type: none"> • C: Benutzer navigiert auf Reward-Übersicht und erstellt einen neuen Reward. • R: Benutzer navigiert auf Reward-Übersicht. Benutzer navigiert zu einem bestimmten Reward und sieht dessen Details. • U: Benutzer navigiert auf Reward-Übersicht. Benutzer navigiert zu einem bestimmten Reward und wechselt in den Bearbeitungsmodus. Der Benutzer bearbeitet die Daten und speichert seine Änderungen.
Alternative Flüsse	-

Tabelle 18: Use Case 3.1 CRU reward

Abschnitt	Kommentar
Name	UC3.2: Archive reward
Beschreibung	Ein eingeloggtter Benutzer kann eine bestehende Reward archivieren, damit vergangene Spiele mit dieser Reward nicht ihre Wertigkeit verlieren. Archivierte Rewards können nicht mehr vergeben werden.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Reward archivieren Besucher: Will seinen Preis nicht verlieren

Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Reward-Daten sind sowohl in der Endbenutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf Reward-Übersicht. 2. Benutzer navigiert zu einer bestimmten Reward oder selektiert eine bestimmte Reward und archiviert diesen.
Alternative Flüsse	-

Tabelle 19: Use Case 3.2 Archive reward

Use Case 4: CRUD sponsor

Abschnitt	Kommentar
Name	UC4: CRUD sponsor
Beschreibung	Ein eingeloggter Benutzer kann Platzierungen von Sponsoren innerhalb der App anpassen.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Sponsoren erfassen Sponsor: Möchte gute, auffallende Platzierung für seine Werbung gegenüber potenziellen Kunden.
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Sponsoren sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. C: Benutzer navigiert auf Sponsoren-Übersicht und erstellt einen neuen Sponsor. 2. R: Benutzer navigiert auf Sponsor-Übersicht. Benutzer navigiert zu einem bestimmten Sponsor und sieht dessen Details. 3. U: Benutzer navigiert auf Sponsor-Übersicht. Benutzer navigiert zu einem bestimmten Sponsor und wechselt in den Bearbeitungsmodus. Der Benutzer bearbeitet die Daten und speichert seine Änderungen. 4. D: Benutzer navigiert auf Sponsor-Übersicht. Benutzer navigiert zu einem bestimmten Sponsor und wechselt in den Bearbeitungsmodus. Der Benutzer löscht den Sponsor.
Alternative Flüsse	-

Tabelle 20: Use Case 4 CRUD sponsor

Use Case 5: CRUD player

Abschnitt	Kommentar
Name	UC5: CRUD player
Beschreibung	Ein eingeloggtter Benutzer kann die Spieler des aktuellen Teams erfassen.
Primärer Akteur	Benutzer in der Rolle Administrator
Stakeholder und Interessensgruppen	Administrator: Will Spieler erfassen
Vorbedingungen	Benutzer muss authentifiziert und als Administrator autorisiert sein.
Erfolgsgarantie	Spieler sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. C: Benutzer navigiert auf Kader-Übersicht und erstellt einen neuen Spieler. 2. R: Benutzer navigiert auf Kader-Übersicht. Benutzer navigiert zu einem bestimmten Spieler und sieht dessen Details. 3. U: Benutzer navigiert auf Kader-Übersicht. Benutzer navigiert zu einem bestimmten Spieler und wechselt in den Bearbeitungsmodus. Der Benutzer bearbeitet die Daten und speichert seine Änderungen. 4. D: Benutzer navigiert auf Kader-Übersicht. Benutzer navigiert zu einem bestimmten Spieler und wechselt in den Bearbeitungsmodus. Der Benutzer löscht den Spieler.
Alternative Flüsse	-

Tabelle 21: Use Case 5 CRUD player

Use Case 6: Login

Abschnitt	Kommentar
Name	UC6: Login
Beschreibung	Benutzer kann sich möglichst einfach einloggen.
Primärer Akteur	Besucher an einem Eishockeyspiel.
Stakeholder und Interessensgruppen	Besucher: Will sich einloggen.
Vorbedingungen	Benutzer ist registriert
Erfolgsgarantie	Benutzer hat sich authentifiziert.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die User-Seite. 2. Benutzer lässt sich einen Link für die Anmeldung zusenden. 3. Benutzer klickt auf den Link und ist eingeloggt.

Alternative Flüsse	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die User-Seite. 2. Benutzer authentifiziert sich über Google oder Facebook.
--------------------	--

Tabelle 22: Use Case 6 Login

Use Case 7: Register

Abschnitt	Kommentar
Name	UC7: Register
Beschreibung	Benutzer kann sich mit einem gewählten Benutzernamen registrieren.
Primärer Akteur	Besucher an einem Eishockeyspiel.
Stakeholder und Interessensgruppen	Benutzer: Will sich registrieren.
Vorbedingungen	Benutzer ist noch nicht registriert.
Erfolgsgarantie	Benutzer ist mit Referenz und Benutzernamen in der Datenbank abgespeichert.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert auf die User-Seite. 2. Benutzer gibt einen Benutzernamen für das Profil an. 3. Benutzer registriert sich.
Alternative Flüsse	-

Tabelle 23: Use Case 7 Register

Use Case 8: RUD user profile

Abschnitt	Kommentar
Name	UC8 RUD user profile
Beschreibung	Ein eingeloggtter Benutzer kann auf der Profil-Übersicht seinen Rang einsehen, seinen Benutzernamen ändern und festlegen, ob er in der öffentlichen Rangliste dargestellt wird.
Primärer Akteur	Besucher an einem Eishockeyspiel.
Stakeholder und Interessensgruppen	Benutzer will sein Profil verwalten.
Vorbedingungen	Benutzer ist eingeloggt
Erfolgsgarantie	Profil-Daten sind sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.

Haupterfolgsszenario	<ol style="list-style-type: none"> 1. R: Benutzer navigiert auf Profil-Übersicht. Benutzer sieht seine Profildaten. 2. U: Benutzer navigiert auf Profil-Übersicht. Benutzer bearbeitet seinen Benutzernamen und speichert seine Änderungen. 3. D: Benutzer navigiert auf Profil-Übersicht. Benutzer löscht seinen Account und damit seine Daten.
Alternative Flüsse	-

Tabelle 24: Use Case 8 RUD User profile

Use Case 9: Play game

Ein Benutzer kann eine freigeschaltene Challenge spielen.

Abschnitt	Kommentar
Name	UC9.1: Solve challenge
Beschreibung	Ein Benutzer kann die freigeschaltene Challenge lösen.
Primärer Akteur	Benutzer in der Rolle Besucher
Stakeholder und Interessensgruppen	Besucher: Will Challenge richtig lösen
Vorbedingungen	Challenge wurde vom Administrator freigeschalten.
Erfolgsgarantie	Challenge-Details werden in der Benutzeransicht angezeigt. Der abgegebene Tipp ist sowohl in der Benutzeransicht, als auch in der Datenbank aktualisiert und einheitlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer scannt QR Code oder navigiert selbstständig zur Startseite 2. Benutzer öffnet Challenge. 3. Benutzer löst Challenge. 4. Benutzer sendet Auswahl ab. 5. Benutzer erhält visuelle Bestätigung der Auswahl. 6. Benutzer wartet auf Abschluss der Challenge.

Alternative Flüsse	Challenge wurde noch nicht gestartet → Dem Benutzer wird eine Meldung dargestellt, dass aktuell keine Challenge offen ist. Sobald die Challenge startet, wechselt die Ansicht.
	Spieler vergisst seine selektierte Auswahl abzuschliessen → Die letzte Auswahl (sofern vorhanden) wird automatisch übermittelt, sodass auch Eingaben ohne Bestätigung in den Challenge-Abschluss einfließen.
	Challenge wurde bereits beendet → Dem Benutzer wird eine Meldung eingeblendet, dass er zur Teilnahme an dieser Challenge leider zu spät war. Optional wird zusätzlich die Lösung dieser Challenge eingeblendet.
	Game wurde bereits beendet Dem Benutzer wird eine Meldung eingeblendet, dass aktuell kein Tippspiel aktiv ist.

Tabelle 25: Use Case 9.1 Solve challenge

Abschnitt	Kommentar
Name	UC9.2: Display solution
Beschreibung	Ein Benutzer kann die Lösung einer Challenge nach dessen Abschluss einsehen.
Primärer Akteur	Benutzer in der Rolle Besucher
Stakeholder und Interessensgruppen	Besucher: Will Lösung sehen
Vorbedingungen	Challenge wurde vom Administrator abgeschlossen.
Erfolgsgarantie	Lösung ist in der Benutzeransicht ersichtlich.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer befindet sich noch in der laufenden Challenge. 2. Benutzer sieht die Beendigung der Challenge. 3. Benutzer sieht die richtige Lösung der Challenge.
Alternative Flüsse	

Tabelle 26: Use Case 9.2 Display solution

Use Case 10: Read results

Der Besucher kann Resultate aus vergangenen Challenges einsehen. Zentral soll ein Score sein, welches das Engagement in der App und Erfolge aus den Challenges widerspiegeln soll.

Abschnitt	Kommentar
Name	UC10.1 Read community results
Beschreibung	Ein Benutzer kann die Resultate anderer Benutzer sehen. Es gibt Ranglisten.

Tabelle 27: Use Case 10.1 Read community results

Abschnitt	Kommentar
Name	UC10.2 Share results
Beschreibung	Ein Benutzer kann seine Resultate mit vorbereiteten Grafiken in sozialen Medien teilen.

Tabelle 28: Use Case 10.2 Share results

Use Case 11: Read rewards

Abschnitt	Kommentar
Name	UC11 Read rewards
Beschreibung	Ein Benutzer kann seine Rewards einsehen. Rewards sind physische Gewinne aus den Tippspielen. Diese können eingelöst werden.
Primärer Akteur	Benutzer in der Rolle Besucher
Stakeholder und Interessensgruppen	Besucher: Will Rewards einlösen können.
Vorbedingungen	Benutzer muss authentifiziert und als Besucher autorisiert sein.
Erfolgsgarantie	Rewards sind in der Benutzeransicht ersichtlich und entsprechen den Daten in der Datenbank. Rewards können als gültig und eingelöst klassifiziert werden.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Benutzer navigiert zur Reward-Übersicht 2. Benutzer navigiert zu einem bestimmten Reward 3. Benutzer sieht die Bedeutung eines Rewards
Alternative Flüsse	-

Tabelle 29: Use Case 11 Read rewards

3.4.3. Priorisierung der optionalen Use Cases

Auch die optionalen Use Cases sind ein wichtiger Bestandteil einer benutzerfreundlichen und interessanten Fan-Engagement-App. Daher werden im Rahmen der Bachelorarbeit möglichst viele davon implementiert. Für die Evaluierung der Umsetzungsreihenfolge werden sie anhand ihres Mehrwertes priorisiert.

1. UC11 Read rewards
2. UC7: Register
3. UC6: Login
4. UC2: Manage challenges
5. UC3: Manage rewards
6. UC10.1 Read community results
7. UC4: CRUD sponsors
8. UC5: CRUD players
9. UC8: RUD user profile
10. UC10.2 Share results

3.5. Nichtfunktionale Anforderungen

Die *Nichtfunktionale Anforderungen (NFA)* der Fan-Engagement-App werden anhand des ISO-25010 Standards für Softwarequalität aufgeführt.[10]

3.5.1. Functional Suitability

Kriterium	Beschreibung
Functional Completeness	Jeder Benutzer kann, die in diesem Dokument beschriebenen und schlussendlich implementierten, Use Cases vollständig ausführen.
Functional Correctness	Alle Benutzer haben Zugriff auf aktive Challenges eines Tippspiels.
	Ein Benutzer kann nur eine Antwort per Challenge abgeben.
Functional Appropriateness	Sämtliche Abhängigkeiten und Bedingungen zwischen einzelnen Use Cases sind aufgelöst und eingehalten.
	Bei der Verwendung der App wird der Benutzer auf die Sponsoren aufmerksam gemacht.

Tabelle 30: NFA Suitability

3.5.2. Performance efficiency

Kriterium	Beschreibung
Time Behaviour	Eine erfasste Antwort wird innerhalb von 5 Sekunden verarbeitet und gespeichert.
	Eine neue Challenge wird innerhalb von 5 Sekunden in der App dargestellt
	Ein Benutzer wird innerhalb von 5 Sekunden eingeloggt.
Capacity	Die Applikation kann mit bis zu 2000 simultan aktiven Benutzern umgehen, ohne die Zeitanforderungen zu überschreiten.

Tabelle 31: NFA Performance efficiency

3.5.3. Compatibility

Kriterium	Beschreibung
Co-existence	Die App ist über die offizielle lakers.ch Domain zugänglich.
Interoperability	Für die Kommunikation zwischen Softwarekomponenten werden standardisierte Schnittstellen verwendet.

Tabelle 32: NFA Compatibility

3.5.4. Usability

Kriterium	Beschreibung
Appropriateness recognizability	Auf der Startseite bekommt der Benutzer einen Überblick über den Funktionsumfang der App.
Learnability	Es wird eine Anleitung für die Bedienung des Regiebereiches als Begleitung des Produktes entwickelt.
Operability	Der gesamte Funktionsumfang kann mittels Touchscreen genutzt werden. Dafür sind Interaktionselemente (Bsp. Button) mindestens 32 Pixel breit.

	<p>Die Web-Applikation wird für gängige Smartphones optimiert und für folgende Bildschirmgrößen übersichtlich dargestellt:</p> <ul style="list-style-type: none"> • Smartphones: 390x844 Pixel (iPhone 12), 428x926 (iPhone 12 Pro Max), 393x851 (Google Pixel 5), 360x800 (Samsung Galaxy S21 Ultra) • Tablet: 1194x834 Pixel (iPad 11) • Desktop: 1920x1080 Pixel (Meistverwendete Monitore) <p>Die Tests werden mit diesen Werten durchgeführt. Die App soll aber auch bei Abweichungen übersichtlich sein.</p>
User error protection	<p>Ausschlaggebende Aktionen muss der Nutzer bestätigen</p> <p>Der Nutzer wird bei ausserordentlichem Verhalten der App (Bsp. Ausfall von Umsystemen) durch eine passende Fehlermeldung informiert.</p>
Accessibility	<p>Statusinformationen werden nicht nur mit Farbe dargestellt, sondern auch mit Symbolen/Text.</p> <p>Ein Farbkontrast von mindestens 5.7:1 (Level AA) wird auf dem gesamten UI gewährleistet.</p> <p>Das UI erfüllt den Google Chrome Accessibility Audit[11].</p> <p>Die Webseite ist zoombar.</p>

Tabelle 33: NFA Usability

3.5.5. Reliability

Kriterium	Beschreibung
Availability	Das Tippspiel bleibt während den Heimspielen erreichbar.
Fault tolerance	Die Benutzereingaben werden bereits im Frontend validiert.
	Bei ungültiger Benutzereingabe wird dem Nutzer eine Fehlermeldung angezeigt.
	Falsche Eingaben vom Benutzer auf der Webseite haben keine Auswirkungen auf die Verfügbarkeit der Dienstleistung.
	Fallen externe Schnittstellen aus, sind betroffene Features deaktiviert und der Nutzer wird darüber informiert.

Tabelle 34: NFA Reliability

3.5.6. Security

Kriterium	Beschreibung
Confidentiality	Passwörter werden falls nötig als Hash-Code gespeichert.
	Logs enthalten keine vertraulichen Informationen (Bsp. Benutzerdaten).
Integrity	Die Kommunikation zwischen einzelnen Softwarekomponenten und den externen Schnittstellen ist mit <i>Transport Layer Security (TLS)</i> verschlüsselt.
	Antworten können ab dem Ende der Challenge nicht geändert werden.

Accountability	Jede Antwort auf eine Challenge ist einem Nutzer zuweisbar.
Authenticity	Für das Abholen von Preisen wird dem Nutzer eine personalisierte und eindeutige Bestätigung erstellt.

Tabelle 35: NFA Security

3.5.7. Maintainability

Kriterium	Beschreibung
Analysability	Der Programmcode ist verständlich oder dokumentiert.
	Komplexe Design-Entscheidungen werden dokumentiert.
	Definierte Style-Guidelines werden eingehalten.
Testability	Das Software Design ermöglicht ein einfaches Mocking von Bestandteilen.
	Unit Tests werden mit der <i>Continuous Integration/Continuous Delivery (CI/CD)</i> Pipeline von <i>GitLab</i> automatisiert und bei jeder Änderung im Respository ausgeführt.

Tabelle 36: NFA Maintainability

3.5.8. Portability

Kriterium	Beschreibung
Adaptability	Die Web-Applikation wird auf den gängigsten Browsern korrekt dargestellt.
	Der Funktionsumfang ist ab den aufgeführten Browser-Versionen uneingeschränkt verfügbar: <ul style="list-style-type: none"> • Google Chrome: Version 58+ • Firefox: Version 54+ • Safari: Version 10.1+

Tabelle 37: NFA Portability

3.5.9. Fan-Engagement

Kriterium	Beschreibung
Attraktivität	Die zu entwickelnde Plattform ist für den Fan motivierend und macht den Verein attraktiver.
Identifikation	Einem Nutzer ermöglicht die App, sich intensiver mit dem Verein auseinanderzusetzen und die Beziehung zu stärken.
Emotionen	Die Fan-Engagement-App wird durch mindestens eine Methode auf ihren emotionalen Effekt überprüft.

Tabelle 38: NFA Fan-Engagement

3.5.10. Kosten

Kriterium	Beschreibung
Spiel-Kosten	Bei einer aktiven Teilnahme von 2000 Nutzern steigen die Kosten nicht über 50 CHF pro Eishockeyspiel.

Tabelle 39: NFA Kosten

3.5.11. Styleguide

Kriterium	Beschreibung
Farben	Das Farbschema der offiziellen SCRJ Lakers Webseite wird übernommen.
Typographie	Die Schriftart «Rubik» wird in der gesamten App verwendet.

Tabelle 40: NFA Styleguide

3.6. Nicht-Anforderungen

Nicht-Anforderungen sind Anforderungen, welche im Verlaufe des Projekts bewusst entfernt werden. Dies kann unterschiedliche Gründe haben. Zum Beispiel bieten sie keinen Mehrwert oder schränken das Fan-Engagement-App in den falschen Aspekten ein.

Stadium-Only

Im Verlaufe des Projekts zeigte sich, dass der Verein mit dem Tippspiel auch besser Menschen erreichen kann, die gerade nicht sehr intensiv den Verein unterstützen. Es kann ein erster Anreiz sein, wieder vermehrt den Weg ins Stadion zu suchen, zwangsläufig um eine gewonnene Belohnung abzuholen. Ohne die örtliche Einschränkung können auch Challenges schon vor dem Spiel oder in der Saisonvorbereitung gestellt werden. Beispielsweise Fragen zu Leistungen einzelner Spieler im Fitnessraum.

Video-Würfel

Der Video Würfel wird generell manuell über vorprogrammierte Programme gesteuert. Eine Anbindung in die Fan-Engagement-App, um automatisch neu veröffentlichte Challenges darzustellen, würde die Würfel-Bedienung beeinträchtigen. Bei Bedarf können die Challenges auch manuell auf dem Video-Würfel präsentiert werden. Deshalb braucht es die Anbindung an den Video-Würfel gar nicht. Der Mehrwert ist minimal.

Preise pro Challenge

Preise werden pro Tippspiel und nicht pro Challenge vergeben. Grund dafür ist, dass es eine faire Gewinnverteilung verhindert. Jemand könnte bei jeder Challenge richtig tippen aber nie einen Preis gewinnen, weil durch Zufall immer einen der anderen richtigen Tipps berücksichtigt wird. Werden alle Tipps über den gesamten Eishockey-Abend zusammengenommen, sind die Punkte breiter gestreut und die Preise werden eher an die besten Tippspieler verteilt. Ausserdem ist ein stärkerer Anreiz gegeben, an sämtlichen Challenges eines Tippspiels statt nur an wenigen ausgewählten Challenges teilzunehmen. Die Zuschauer verbringen mehr Zeit in der App und das macht es wieder spannender für die Sponsoren.

Reward-Einlösung

Die Rewards können aus einem Benutzerprofil nicht mehr gelöscht werden. Das Risiko wäre zu gross, dass ein Benutzer den Slider zum Einlösen ausprobiert und dabei den Preis löscht, ohne dass er abgeholt werden konnte. Um trotzdem eine Übersicht zu bieten sind die Preise nach eingelöst, respektiv nicht eingelöst kategorisiert. Die bereits abgeholt Preise sind zusätzlich mit Datum und Uhrzeit versehen. Zudem wird ein Zuschauer nicht allzu viele Preise gewinnen und es stört kaum jemanden, wenn zusätzlich noch alte Preise dargestellt werden.

Parallel aktive und vergangene Tippspiele

Es wird immer nur ein Tippspiel gleichzeitig aktiv sein. Das ohnehin schon eher komplexe Admin Panel würde mit mehreren aktiven Tippspielen zudem noch viel komplizierter werden. Es wäre fehleranfälliger und würde eine klare Bedienung behindern. Auch die Darstellung für den Besucher wäre um einiges komplizierter und würde zu mehr Klicks für ein Tippspieltipp führen. Das ist definitiv nicht die Absicht. Auch auf die Darstellung vergangener Tippspiele wird im Admin-Bereich bewusst verzichtet, weil es keinen wirklichen Mehrwert bringt.

3.7. Domain Model

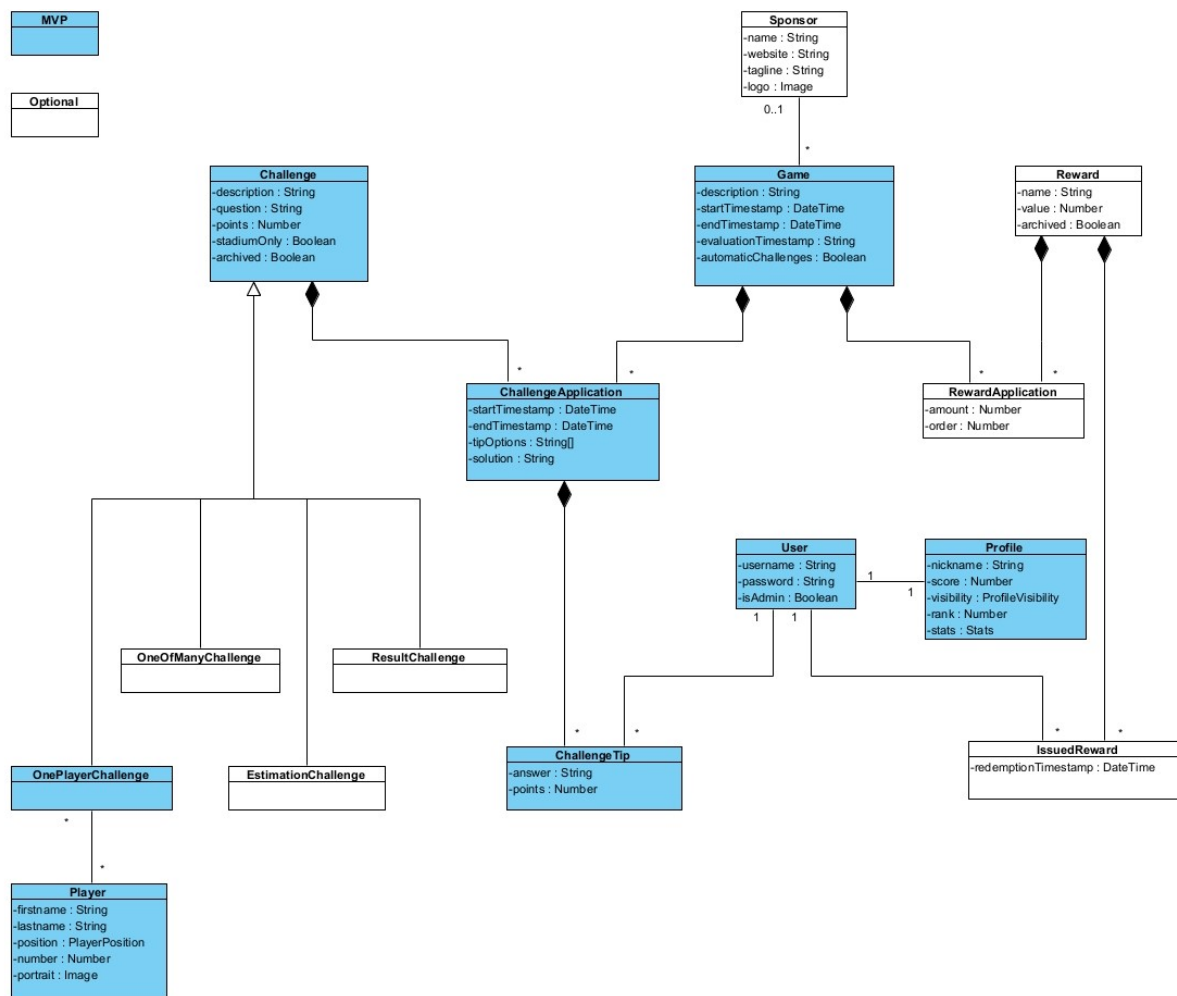


Abbildung 5: Domain Model

Objekt	Beschreibung
Game	Ein <Game> entspricht einem Tippspiel und beinhaltet die damit assoziierten weiteren Objekte.
Challenge	Eine <Challenge> ist eine beliebige Aufgabe, beispielsweise nach dem Frage-Antwort-Schema.
ChallengeApplication	<ChallengeApplication> beinhaltet sämtliche Informationen zu einer einzelnen Durchführung einer Challenge.

ChallengeTip	Jeder Teilnehmer hat sein eigenes Resultat zu jeder $\langle \text{ChallengeApplication} \rangle$.
User	Ein $\langle \text{User} \rangle$ beinhaltet sämtliche Informationen, welche für ein erfolgreiches Login beziehungsweise für eine erfolgreiche Authentifizierung benötigt werden. Zusätzlich kann ein Administratoraccount als solcher erkannt werden.
Profile	Jeder registrierte $\langle \text{User} \rangle$ hat ein Profil. Darin sind persönliche teils öffentliche Daten im Zusammenhang mit der Applikation abgelegt.
Sponsor	Alle Informationen rund um einen Sponsor, der in der App präsentiert werden soll, sind im Objekt $\langle \text{Sponsor} \rangle$ abgelegt.
Player	Alle Informationen rund um einen Spieler, die für die Selektierung benötigt werden.
Reward	Alle Arten von Preisen, wie beispielsweise Sachwerte oder Gutscheine.
RewardApplication	$\langle \text{RewardApplications} \rangle$ repräsentieren Preise, die es in einem einzelnen $\langle \text{Game} \rangle$ zu gewinnen gibt.
IssuedReward	Erworbene $\langle \text{Rewards} \rangle$ sind immer einem $\langle \text{User} \rangle$ zugewiesen. Grundsätzlich sind $\langle \text{IssuedRewards} \rangle$ valide bis diese in physische Gutscheine oder Warengut umgetauscht wurden.

Tabelle 41: Objekte im Domain Model

4. Design

4.1. Systemarchitektur

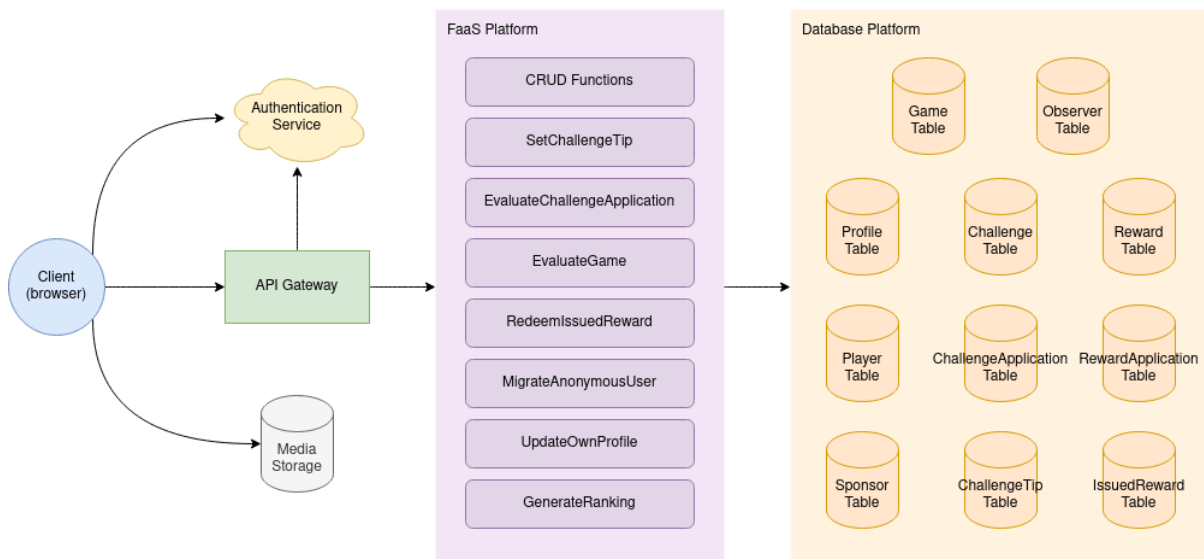


Abbildung 6: Makroarchitektur

4.1.1. Serverless Computing

Für die Bereitstellung sämtlicher Backend-Dienste wird auf eine *Serverless*-Architektur gesetzt. Die *Serverless*-Dienste bieten eine sehr hohe Flexibilität in der Skalierung und Leistungsfähigkeit. Je nach Bedarf steht der Applikation dynamisch mehr oder weniger Leistung zur Verfügung. Für den Benutzer bleibt die Benutzung somit stets stabil, unabhängig der sonstigen Auslastung. Es gibt auch finanzielle und organisatorische Anreize auf eine *Serverless*-Infrastruktur zu setzen. Bezahlt wird nämlich nur, was auch gebraucht wird. In der Zeit, in der keine Last anfällt, entstehen auch keine Kosten. Denn *Serverless*-Dienste werden immer nur durch bestimmte Ereignisse ausgelöst, womit die typische Standby-Zeit und unverbrauchte Rechenleistung von Systemen in der Verechnung wegfällt. Code wird für ereignisspezifische Aufgaben in *Serverless* Functions (Lambdas) geschrieben und somit stark voneinander separiert. Das macht die Applikation und dessen Bestandteile besser wart- und austauschbar. Lambda Functions beinhalten zustandslosen Code, der erst zum Zeitpunkt eines Ereignisses geladen und ausgeführt wird. Da *Serverless* Computing für jeden Ereignisaufwurf verschiedene Instanzen verwendet, werden Fehler sehr isoliert behandelt und betreffen nicht das ganze System. Die Anforderungen für diese Applikation sind ideal für den Einsatz einer *Serverless*-Infrastruktur.

4.1.2. Frontend

Sowohl eine Nativ-, als auch eine Web-Applikation kann die funktionalen Anforderungen an das Produkt erfüllen. Der Zugang für die Fans soll angenehm gestaltet und somit die Hemmschwelle für die Nutzung minimiert werden. Deshalb wird in den *Nichtfunktionale Anforderungen (NFA)* beschrieben, dass die App der bestehenden SCRJ Lakers Webseite angegliedert werden soll. Somit ist eine Native-App ausgeschlossen. Um eine Installation trotzdem anbieten zu können, wird die App als *PWA* entwickelt.

4.1.3. Deployment

Produkt

Das Produkt wird in einer Public Cloud gehostet und betrieben.

Der Kunde möchte eine möglichst einfach zu betreibende, langlebige Applikation haben, die minimal auf technische Expertise und Betreuung angewiesen ist. Das hängt auch mit dem veranschlagten Kosten

zusammen. Diese sollen möglichst gering bleiben und sich hauptsächlich auf die effektive Applikationsnutzung beschränken. Eine Public Cloud kann diese Anforderungen am besten erfüllen. Je grösser die Server-Infrastruktur ist, desto kostengünstiger ist dessen Betrieb. Deshalb ist eine eigene Serverinfrastruktur einiges teurer als der Einkauf in eine bestehende Public Cloud. Zusätzlich überzeugt die Public Cloud mit ihrer Zuverlässigkeit und Stabilität. Solche Anbieter haben sich auf den Betrieb von fremden Applikationen spezialisiert, kümmern sich um sehr viele verschiedene Applikationen gleichzeitig und können dadurch auf einen beachtlichen Erfahrungsschatz zurückgreifen. *On-Premise*-Lösungen wie beispielsweise die Serverinfrastruktur der OST hat da das Nachsehen. Zudem ist die effektive Auslastung dieser Applikation schwierig abzuschätzen. Public Cloud Anbieter erlauben eine sehr flexible Skalierung der Ressourcen. Das hält den Wartungsaufwand über längere Zeit insgesamt sehr gering.

Eine Public Cloud birgt aber auch ein paar Risiken. Die direkte Kontrolle über Datenspeicherung, Software-Updates und Sicherheitsmassnahmen wird abgegeben. Es braucht Vertrauen in die Arbeit des Drittanbieters, dass dieser seiner Verantwortung nachkommt. Die Daten sollen beispielsweise zur Gewährleistung des Datenschutzes in Europa oder besser in der Schweiz gelagert werden. Eine eigene Kontrolle über dessen Einhaltung kann aber nicht durchgeführt werden.

Entwickler- und Projekttools

Die Anforderungen an den Betrieb solcher Tools ist sehr viel geringer. So müssen diese nicht permanent laufen, sondern dürfen auch mal ausfallen oder heruntergefahren werden. Sicherheitsaspekte wie die Aktualität der Software sind nur von Bedeutung, wenn diese öffentlich im Internet hängen. Dafür soll die Systemumgebung möglichst pragmatisch und individuell konfigurierbar sein. Das kann eine *On-Premise*-Infrastruktur am besten bieten. So laufen alle Entwickler- und Projekttools auf der Infrastruktur der OST.

4.2. Konzepte

4.2.1. Spielablauf

Der Verlauf in einem Spiel ist mehrheitlich sequenziell und beinhaltet drei verschiedene Ebenen. Diese sind durch drei verschachtelte Zustandsdiagramme abgebildet. Die `<Challenge>` in der State Machine bezieht sich auf eine `<ChallengeApplication>` im Domain Modell.

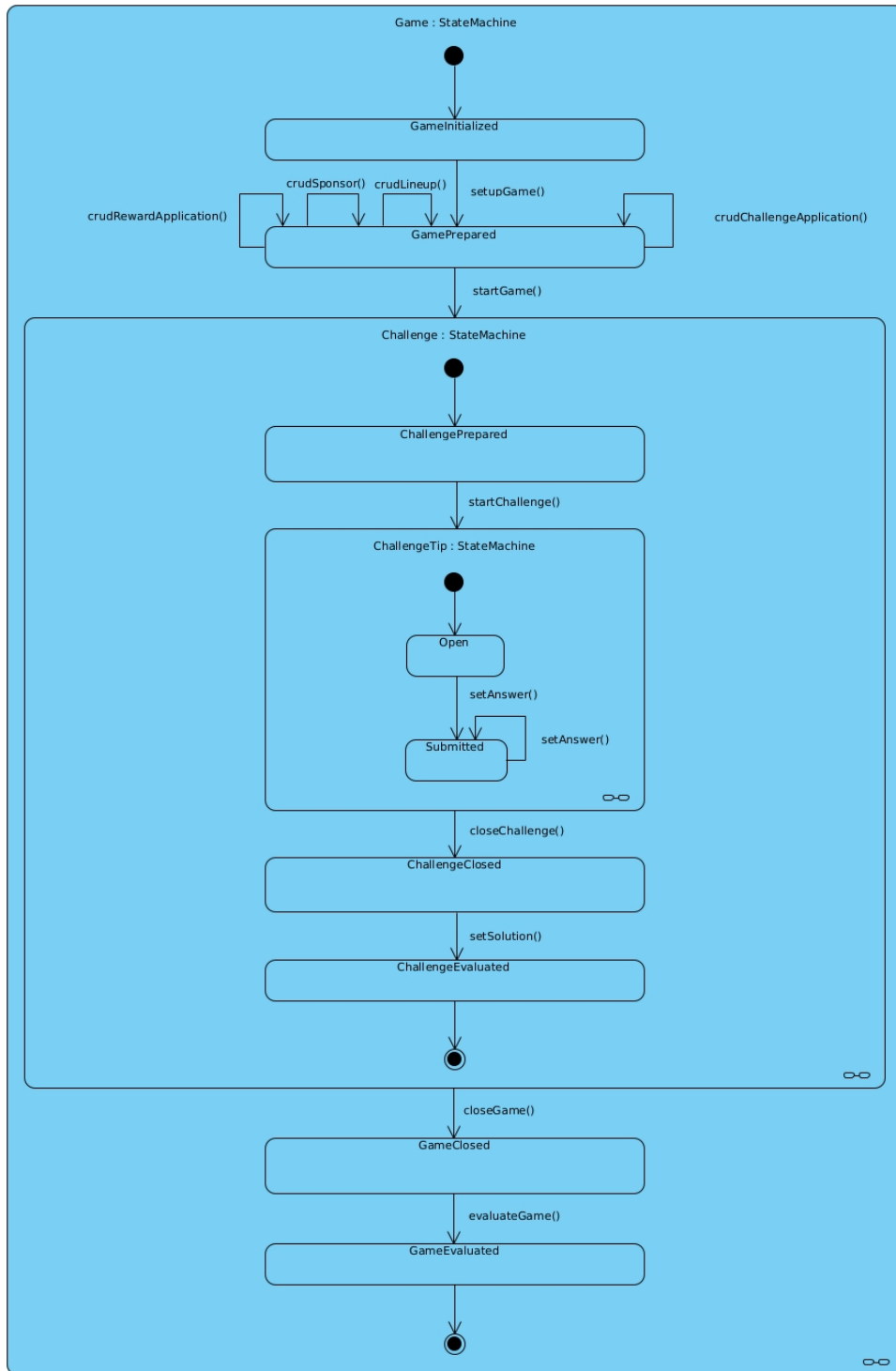


Abbildung 7: State Diagram eines Games

State Machine Game

Beschreibt den Lebenszyklus eines $\langle \text{Game} \rangle$. Dieses wird gesteuert durch den Administrator.

Zustand	Beschreibung
GameInitialized	Das $\langle \text{Game} \rangle$ wird aufgesetzt und mit den nötigen Informationen angereichert.
GamePrepared	Das $\langle \text{Game} \rangle$ wurde erstellt. $\langle \text{Challenges} \rangle$, $\langle \text{Rewards} \rangle$ und der $\langle \text{Sponsor} \rangle$ können beliebig und jederzeit hinzugefügt und wieder entfernt werden.
GameOpen	Ein $\langle \text{Game} \rangle$ kann gestartet werden, womit weitere Funktionen im Zusammenhang mit $\langle \text{Challenges} \rangle$ freigeschalten werden. Diese sind im State Diagram Challenge ersichtlich.
GameClosed	Ein gestartetes $\langle \text{Game} \rangle$ kann beendet werden. Damit werden auch alle $\langle \text{Challenges} \rangle$ unspielbar.
GameEvaluated	Ein beendetes $\langle \text{Game} \rangle$ kann ausgewertet werden. Damit werden die $\langle \text{Challenge} \rangle$ -Resultate zusammengetragen und die vorab definierten $\langle \text{Rewards} \rangle$ verteilt.

Tabelle 42: Zustände eines Games

State Machine Challenge

Diese läuft für jede ausgewählte $\langle \text{Challenge} \rangle$ unabhängig. Auch diese ist in der Kontrolle des Administrators.

Zustand	Beschreibung
ChallengePrepared	Ausgewählte $\langle \text{Challenges} \rangle$ können gestartet werden.
ChallengeOpen	Während die $\langle \text{Challenge} \rangle$ aktiv ist, können Besucher diese lösen. Die detaillierten Möglichkeiten sind im State Diagram $\langle \text{ChallengeTip} \rangle$ abgebildet.
ChallengeClosed	Ist eine $\langle \text{Challenge} \rangle$ beendet, können keine weiteren $\langle \text{ChallengeTips} \rangle$ abgeben oder bestehende Antworten geändert werden.
ChallengeEvaluated	Die Antworten wurden mit der Lösung verglichen und entsprechende Punkte verteilt.

Tabelle 43: Zustände einer Challenge

State Machine ChallengeTip

Beinhaltet die Status einer laufenden $\langle \text{Challenge} \rangle$. Dabei sind die Besucher beziehungsweise $\langle \text{Challenge} \rangle$ -Spieler in der Verantwortung.

Zustand	Beschreibung
Open	Eine $\langle \text{Challenge} \rangle$, die von einem Besucher noch nicht gelöst wurde.
Submitted	Eine $\langle \text{Challenge} \rangle$, die von einem Besucher gelöst und eingereicht wurde. Der Besucher kann den Lösungsvorschlag beliebig oft anpassen.

Tabelle 44: Zustände eines ChallengeTips

4.2.2. Anonyme Tipps

Ein Benutzer soll möglichst viele Inhalte sehen und auch Tipps abgeben können, ohne dass er sich zuerst noch für die Applikation registrieren und anmelden muss. Dafür müssen anonyme Tipps abgegeben werden können, die dennoch später einem Benutzer eindeutig zugewiesen und an einen registrierten Benutzer übertragen werden können.

Zu diesem Zweck wird in der App für den Benutzer bei fehlender Authentifizierung im Hintergrund ein anonymes Benutzerkonto angelegt, welches sich in der Gestalt kaum von einem registrierten Benutzer unterscheidet. Es wird gemäss eines registrierten Benutzers für Anfragen und die persistente Speicherung von Eingaben verwendet. Der offensichtlichste Unterschied besteht in der Sichtbarkeit beziehungsweise Unsichtbarkeit im Frontend je nach Benutzertyp. Im Backend wird die Unterscheidung zwischen anonymen und registrierten Benutzern durch verschiedene Benutzergruppen gewährleistet.

4.2.3. Benutzermigration

Registriert sich oder meldet sich ein bereits aktiver Benutzer in der Applikation an, so werden die gespeicherten `<ChallengeTips>` und `<IssuedReward>` vom anonymen Benutzer auf den registrierten Benutzer übertragen. Gibt es von beiden Benutzern `<ChallengeApplication>`, wird der neuere übernommen und der andere entfernt. Der anonyme Benutzer wird danach ebenfalls nicht mehr benötigt und daher gelöscht.

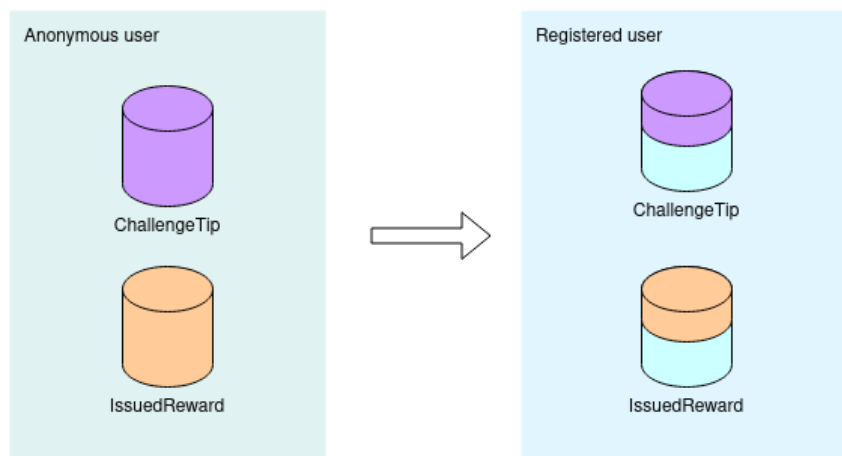


Abbildung 8: Benutzermigration

4.2.4. Passwordless-Login

In der App können sich Benutzer ohne Passwort authentifizieren. Das verkürzt den Verifizierungs- und vereinfacht den Anmeldeprozess. Der Benutzer muss nicht mehr die E-Mail-Adresse bestätigen und sich ein Passwort merken, geschweige denn eingeben. Dessen Nutzen werden stattdessen mit einem sogenannten Magic Link erreicht, welcher dem Benutzer über einen externen Kommunikationskanal, beispielsweise E-Mail, mitgeteilt wird und diesen für eine sehr beschränkte Zeit authentifiziert.

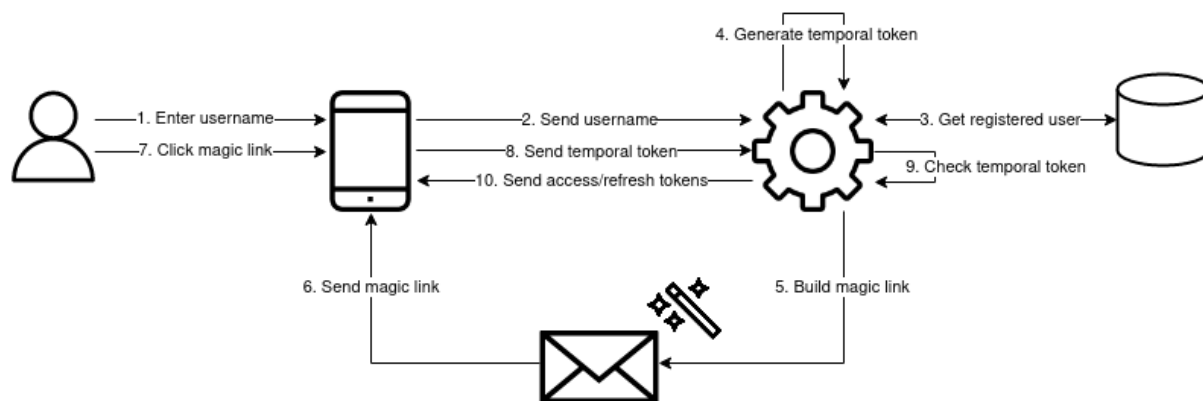


Abbildung 9: Passwordless-Login-Prozess

Um davon Gebrauch zu machen, gibt der Benutzer seinen Benutzernamen in der App ein und schickt diesen ab. Das Backend sucht einen passenden registrierten Benutzer und generiert einen temporären Token zur dessen Authentifizierung. Dieser Token wird in eine bestimmte *Uniform Resource Locator (URL)* eingearbeitet und wird somit zum Magic Link. Dieser Magic Link wird auf einem alternativem Weg an den Benutzer retourniert. Mit einem Klick auf diesen Link kann sich der Benutzer als solcher gegenüber der App identifizieren und bestätigen, dass er im Besitz der E-Mail-Adresse ist.

4.2.5. Datenbank

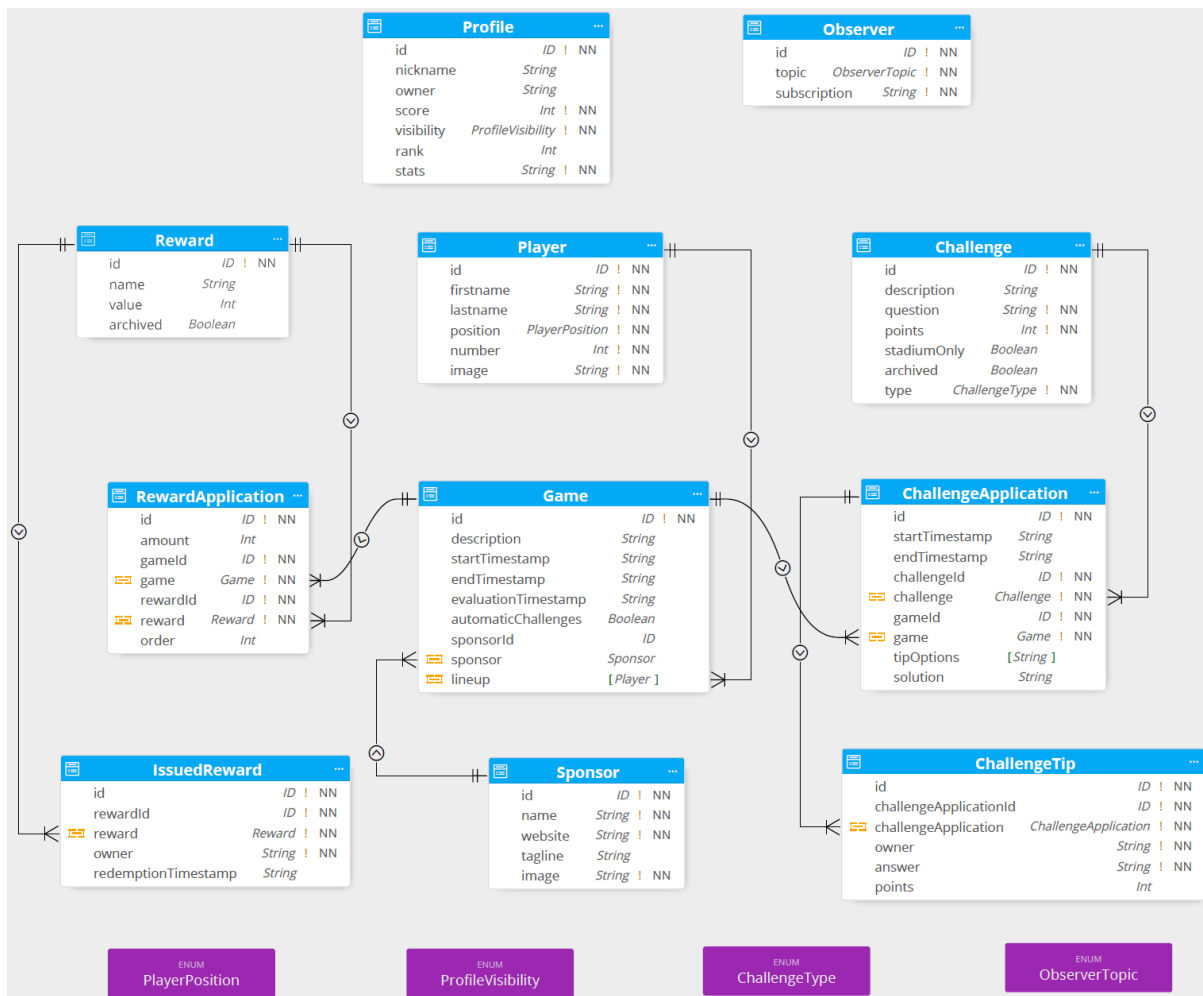


Abbildung 10: Schema der Datenbank

Das Datenbankschema entspricht weitgehend dem Domain Modell. Relationen zwischen den Tabellen sind visuell durch zwei Felder repräsentiert. In der Datenbank wird jeweils nur der Fremdschlüssel beziehungsweise die jeweilige id abgespeichert. Jeder Tabelleneintrag hat zusätzlich einen Erstellungs- und Aktualisierungszeitpunkt, damit Mutationen zeitlich besser nachvollzogen werden können. Folgende Datenbankeinträge sind benutzerspezifisch.

- Profile (öffentlich)
- ChallengeTip (privat)
- IssuedReward (privat)

Diese sind durch das Feld <owner> mit seinem Besitzer markiert. Die Datenbankzugriffe auf private Objekte werden dahingehend eingeschränkt.

Modifikationen von anderen Objekten erfordern die Administrator-Rolle.

- Game
- Challenge
- ChallengeApplication
- Reward

- RewardApplication
- Sponsor

Diese Objekte sind aber ausnahmslos öffentlich lesbar. Die meisten Objekte im Schema werden zusätzlich durch das System, konkret durch entwickelte Funktionen, bearbeitet.

Die <Observer>-Tabelle wird für Push-Benachrichtigungen benötigt. Diese speichert Kontaktinformationen derjenigen Browser, die Benachrichtigungen bei bestimmten Ereignissen wünschen.

Datenbankmodell

Was spricht für den Einsatz einer SQL-Datenbank?

- Schema hat sehr viele Relationen
- Abfragen benötigen häufig mehrere verknüpfte Modelle
- Erfahrung der Entwickler
- *Atomicity-Consistency-Isolation-Durability (ACID)*-Unterstützung über mehrere Tabellen

Was spricht für den Einsatz einer NoSQL-Datenbank?

- Lässt sich sehr einfach deployen
- Bietet oft bessere Integration für Echtzeit-Streaming
- Schnellere Abfragen (von Dokumenten)
- Unkomplizierte Migration (Umgang mit Schema-Modifikationen)
- Sehr flexibel in Skalierung (horizontal)
- Einfachere Integration in *GraphQL*
- Dokumente sind in entwicklerfreundlichem Format
- Günstiger im Betrieb
- Einfacher in Wartung

Betrachtet man nur das Datenbankschema und die Abfragen, ist eine SQL-Datenbank die geeignetste Wahl. Die vielen Relationen und der Fakt, dass alle Informationen normalisiert sind, sprechen klar für den Einsatz einer SQL-Datenbank. Eine NoSQL-Datenbank bietet aber in diesem Projekt wesentlich mehr Vorteile. Besonders ins Gewicht fallen dabei der deutlich tiefere Entwicklungs- und Verwaltungsaufwand. Die sehr einfache Handhabung im Zusammenhang mit *GraphQL* und tieferen Kosten im Betrieb sind ebenfalls starke Argumente, auf NoSQL zu setzen. Die Entwickler möchten zudem mehr Erfahrungen mit NoSQL-Datenbanken sammeln. Letzendlich basiert die Entscheidung zugunsten einer NoSQL-Datenbank also auf mehreren Aspekten.

4.2.6. Datenfluss

Frontend-Backend

Aktuell gibt es zwei moderne weitverbreitete Konzepte - *Representational State Transfer (REST)* und *GraphQL*. Die Entscheidung ist zugunsten von *GraphQL* ausgefallen. Denn *GraphQL* hat gegenüber *REST*-Schnittstellen einige schwerwiegende Vorteile. Im Zusammenhang mit komplexen Abfragen, welche in dieser Applikation häufig notwendig sind, kann *GraphQL* wesentlich mehr bieten als *REST*. Denn *GraphQL* erlaubt sehr flexible Abfragen auf mehrere verschiedene Ressourcen gleichzeitig. Das verringert die Anzahl Anfragen (Round Trips) auf das Backend und schlussfolgernd die Antwortzeit im Frontend. *GraphQL* schafft die Flexibilität mit einem eigenen Typsystem. Die API ist strikt durch Datentypen beschrieben und lässt sich somit beliebig anordnen und verschachteln. So entstehen automatisch die Beziehungen zwischen einzelnen Objekten. Durch die starke Typisierung ist auch immer klar, was von der Gegenseite gefordert beziehungsweise geliefert wird. Zusammengefasst ist *GraphQL* wesentlich präziser und effizienter als *REST*.^[12]

Backend-Backend

Grosser Bestandteil der backend-internen Kommunikation sind die Datenbankabfragen. Ansonsten sind es vereinzelt Abfragen auf Dienste wie E-Mail oder Authentifizierung. Im Backend ist Einfachheit und Wartungsfreundlichkeit das oberste Gebot. Deshalb sollen möglichst bestehende vorgefertigte Schnittstellen und *Software Development Kit (SDK)*s verwendet werden.

4.2.7. Challenge-Typen

Es gibt verschiedene Challenge-Typen. Diese haben entweder eine eigene visuelle Darstellung im Frontend oder werden im Backend unterschiedlich ausgewertet. Diese eigene Charakteristik bedarf einer Unterscheidung.

onePlayerChallenge

Die `<onePlayerChallenge>` hat einen bestimmten Spieler als Lösung. Dem Benutzer steht eine Liste mit Spielern zur Verfügung aus denen er seine Antwort aussuchen kann. Zusätzlich kann auch `<Niemand>` ausgewählt werden, wenn die Fragestellung auf keine Person zutreffen wird. Mögliche Fragestellungen sind

- Welcher Spieler erzielt das erste/nächste Tor?
- Welcher Spieler bekommt die erste Strafe?
- Welcher Spieler bricht den ersten Stock?
- Welcher Spieler macht heute den schnellsten Sprint?
- Welcher Feldspieler hat heute die meiste Eiszeit?

oneOfManyChallenge

Mit der `<oneOfManyChallenge>` werden dem Benutzer zwei bis vier Antwortmöglichkeiten angezeigt. Die Antwortmöglichkeiten werden im Attribut `<tipOptions>` als String-Array abgelegt.

- Spieler mit welcher Position schießt das nächste Tor? Verteidiger, Center, Angreifer oder Niemand
- Was passiert zuerst? Strafe, Tor, Offside oder Icing

estimationChallenge

Bei der `<estimationChallenge>` wird dem Benutzer ein Feld zur Eingabe einer beliebigen Zahl angezeigt. Der Benutzer kann damit eine Schätzfrage beantworten. Beispiele hierfür sind

- Wie viele Zuschauer sind heute im Stadion?
- Wie viele Meter hat Spieler x heute zurückgelegt?
- Wie viele Schüsse hat das Heimteam heute abgegeben?
- Wie viele Minuten ist heute der aktivste Feldspieler auf dem Eis?

resultChallenge

Die `<resultChallenge>` erwartet als Antwort einen String bestehend aus zwei Zahlenwerten, getrennt durch einen passenden Teiler (beispielsweise 3:5 oder 1-2). Damit können Drittel- oder Spielresultate erraten werden.

- Wie steht es nach dem ersten Drittel?
- Mit welchem Resultat gehts heute in die nächste Pause?
- Was ist das Torverhältnis nach dem nächsten Drittel?
- Mit welchem Resultat wird das Spiel heute enden?

4.2.8. Challenge-Auswertung

Für die Auswertung einer gespielten Challenge sind mindestens vier, selten sogar fünf Informationen von Bedeutung.

- Challenge-Typ
- Punktzahl

- Lösung
- Antworten aller Tippabgaben (teilweise)
- Einzelne Antwort

Je nach Challenge Typ wird eine Challenge anders ausgewertet.

Gleichheit

Damit werden sowohl `<onePlayerChallenges>` als auch `<oneOfManyChallenges>` ausgewertet. Die Auswertung basiert einzig und allein auf der exakten Gleichheit von Antwort- und Lösungsfeld. Die Semantik und Syntax des Antwort-Strings spielt dabei keine Rolle. Es zählt einzig und allein die Gleichheit der beiden Strings. Ist diese gleich, erhält der Tipp die volle Punktzahl, ansonsten geht der Tipp mit null Punkten leer aus.

Ein kleines Beispiel bei 100 möglichen Punkten.

Antwort	Lösung	Punkte
2c8b02e9-48f1-409c-9d4f-cc178667e426	2c8b02e9-48f1-409c-9d4f-cc178667e426	100
d0d8ae1a-2bbc-4595-8325-eae47fa89cc7	2c8b02e9-48f1-409c-9d4f-cc178667e426	0

Tabelle 45: Beispiel zur Auswertung auf Gleichheit

Resultat

Die `<resultChallenge>` wird gesondert ausgewertet. Denn wer das Resultat exakt tippt, erhält natürlich das Punktemaximum. Für ansatzweise richtige Resultate gibt es aber auch Teilpunkte. Die einzelnen Positionen werden für die persönliche Punktzahl miteinander addiert.

- 50% bei korrektem Sieger
- 20% bei korrekter Anzahl Heimtore
- 20% bei korrekter Anzahl Gasttore
- 10% bei korrekter Tordifferenz (Sieger muss richtig sein)

Geht man beispielsweise von 5:3 als Lösung und 100 zu holenden Punkten aus, ergibt das folgende möglichen Szenarien

Antwort	Lösung	Sieger	Heimtore	Gasttore	Tordifferenz	Punkte
0:2	5:3					0
1:0	5:3	X				50
3:1	5:3	X			X	60
5:2	5:3	X	X			70
5:3	5:3	X	X	X	X	100
4:3	5:3	X		X		70
0:3	5:3			X		20

Tabelle 46: Beispiel zur Auswertung auf Resultat

Schätzung

Bei `<estimationChallenges>` erhalten diejenigen mit der niedrigsten Differenz zur Lösung die maximale Punktzahl. Desweiteren bekommen die maximal 20% besten Schätzer ebenfalls noch Teilpunkte. Die Punkte werden ähnlich einer gescherten Gausschen Funktion verteilt.

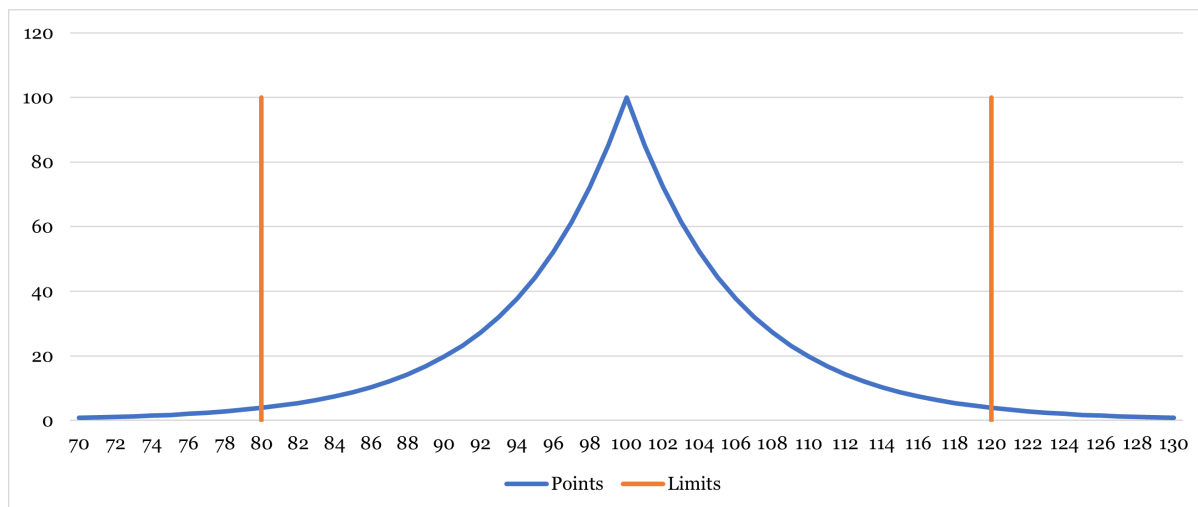


Abbildung 11: State Diagram eines Games

In diesem Beispiel ist 100 die maximale mögliche Punktzahl und es sind insgesamt 200 Antworten eingegangen. Dadurch können maximal 40 Teilnehmer Punkte erhalten. Jede Antwort bildet eine andere Zahl zwischen 1 und 200 ab. Die Lösung ist die Zahl 100. Damit hat derjenige Tipper mit ebenfalls 100 als Antwort gewonnen.

Antwort	Lösung	Punkte
79	100	0
80	100	3
81	100	4
...
99	100	85
100	100	100
101	100	85
...
119	100	4
120	100	3
121	100	0

Tabelle 47: Beispiel zur Auswertung auf Schätzung

4.2.9. Spielauswertung

Bei der Spielauswertung geht es um die Vergabe der Preise. Jeder Preis hat eine Priorität. Jeder Spieler hat eine Gesamtpunktzahl vom Spiel. Die Preise werden absteigend gemäss dieser Wertung an registrierte Spieler verteilt. Anonyme Spieler sind vom Gewinnspiel ausgeschlossen. Der beste Spieler erhält den Preis mit der höchsten Priorität. Der zweitbeste Spieler den Preis mit zweithöchster Priorität. Haben zwei oder mehrere Spieler dieselbe Punktzahl, so entscheidet das Zufallsprinzip. Sind die Preise vergeben, gehen die verbliebenen Spieler leer aus. Jeder Spieler kann damit maximal einen Preis pro Spiel gewinnen.

Zur besseren Veranschaulichung nachfolgend ein Beispiel. In diesem Spiel können folgende Preise gewonnen werden.

Name	Priorität	Menge
Topsorer-Shirt	1	1
Pommes Frites	2	3
Bier	3	1

Tabelle 48: Mögliche Gewinne in einem Beispiel-Spiel

Geht man von mindestens sechs Spielern aus, die Punkte im Spiel gesammelt haben, so werden die Preise folgendermassen vergeben.

Nickname	Punkte	Preis
Dävodet	400	Topsorer-Shirt
Isoldemadusche	250	Pommes Frites
FanHoch2	175	Pommes Frites
WerwillmeinePommes	175	Pommes Frites
Guenther	150	Bier ¹
Chef	150	keinen ¹

¹ Diese Preise sind zufällig zugeordnet

Tabelle 49: Gewinnverteilung in einem Beispiel-Spiel

4.3. User Interfaces

4.3.1. Style Guide

Das Design im Frontend wird von Grund auf entworfen. Die Anforderungen des Kunden bestehen lediglich daraus, den Styleguide der offiziellen SCRJ Lakers Webseite anzuwenden. Daher sind das Farbschema, die Typographie und Bilder gegeben.

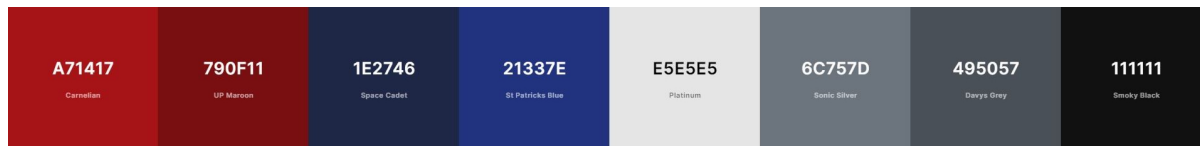


Abbildung 12: Farbschema der SCRJ Lakers

Als Schriftart wird, wie auf der bestehenden Webseite der SCRJ Lakers, Rubik verwendet. Sie besitzt eine Open Font Lizenz [13] und kann daher kommerziell verwendet werden. In allen Schriftzügen und Texten der App wird das Standard-Letterspacing der Schrift genutzt.

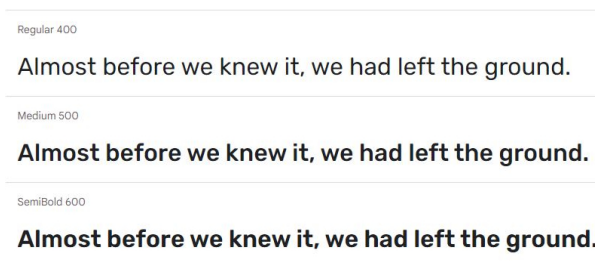


Abbildung 13: Schriftart Rubik

Die Orientierung und Kommunikation in der Fan-Engagement-App werden mit dem Einsatz von Piktogrammen verbessert. Es werden *Font Awesome Icons* verwendet, welche in der *SemanticUI*-Bibliothek eingebettet sind. Um Verwechslungen zu vermeiden, werden Symbole mit Beschreibungen ergänzt.



Abbildung 14: Font Awesome Piktogramme[14]

4.3.2. Design Prozess

Die gesamte User Experience und das spezifische Design werden selbstständig entworfen und optimiert. Für diesen Prozess wird ein *User-centered-Design*-Ansatz verfolgt. Dies ist ein iteratives Vorgehen, bei dem die Anforderungen des Nutzers im Mittelpunkt steht. Es werden Prototypen entworfen, welche beispielsweise mit Usability Tests evaluiert und anhand der gesammelten Erkenntnissen weiterentwickelt werden.

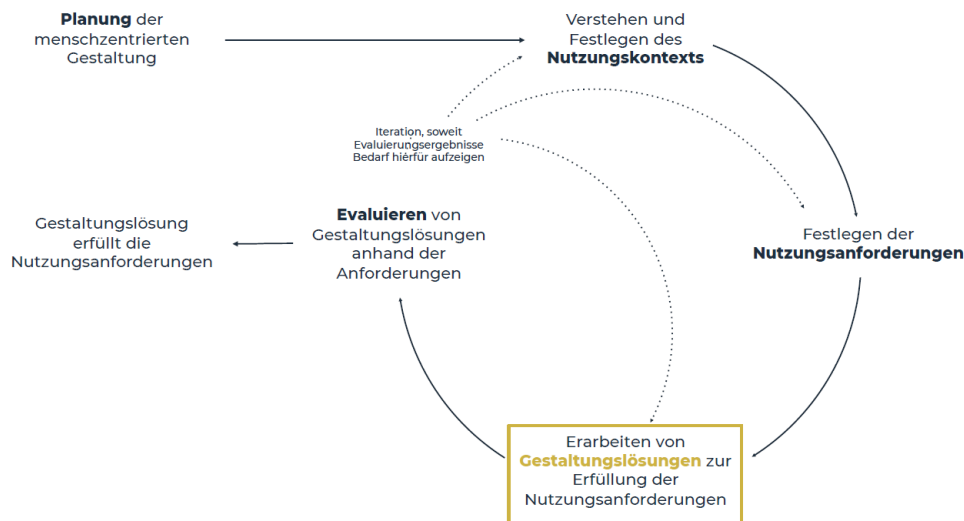


Abbildung 15: User Centered Design Prozess [15]

4.3.3. Prototyping

Anhand der Wireframes, welche im Storyboard skizziert sind, wird ein erster Entwurf der Fan-Engagement-App erstellt. Dafür wird das Tool *Figma* verwendet. Die Entwicklung in einem *High-Fidelity-Prototyping-Tool* hat den Vorteil, mit relativ wenig Aufwand zu einem täuschend echten Entwurf zu gelangen. Dadurch können verschiedene Ideen und Lösungsvorschläge entwickelt und verglichen werden, ohne eine Zeile Code zu schreiben. Im Prototyp wird neben der geplanten Funktionalität auch das visuelle Design abgebildet. *Figma* ermöglicht ausserdem, den Designentwurf interaktiv zu gestalten. Die Absicht ist eine Mischform zwischen einem flachen und doch tiefen Prototypen zu entwickeln. Er soll einen Gesamteindruck der App vermitteln, aber gleichzeitig für Usability-Tests verschiedene Szenarien mit ausreichend detaillierten Screens abbilden. Es folgen nun einige Eindrücke des ersten Entwurfs. Die restlichen Screens befinden sich im Anhang (Kapitel E.3).



Abbildung 16: Startbildschirm im Prototyp

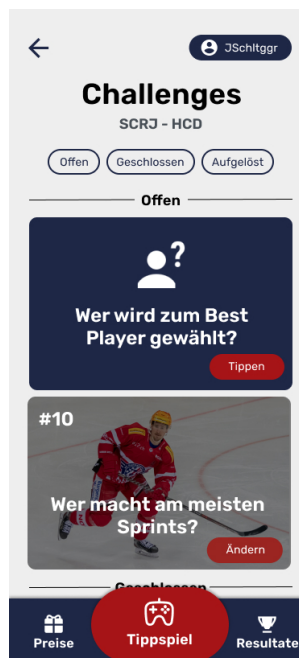


Abbildung 17: Tippspiel-Ansicht im Prototyp

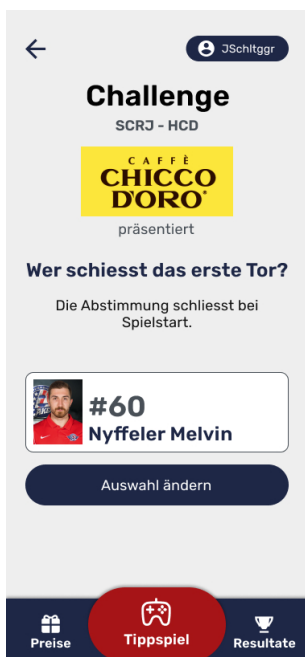


Abbildung 18: Challenge-Ansicht im Prototyp

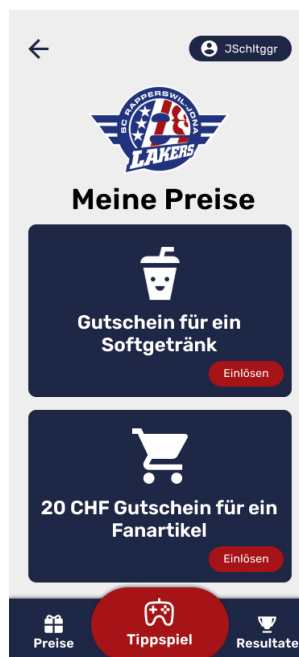


Abbildung 19: Preis-Ansicht im Prototyp

4.3.4. Validierung

Mit dem Entwurf des Prototyps werden Usability Tests durchgeführt, um die Gebrauchstauglichkeit zu validieren und offene Fragen zu klären. Die daraus resultierenden Erkenntnisse können für die Verbesserung der User Experience genutzt werden. Die unbearbeiteten Daten wie Testprotokolle sind im Anhang (Kapitel A.1), Testkonzepte und Analyse in der Evaluierung (Kapitel 6) zu finden.

4.4. Externe Schnittstellen

Die notwendigen Informationen für den Einsatz bestimmter externer Schnittstellen sind vorhanden, noch ist aber keine Schnittstelle implementiert. Weiterführende Dokumente mit den genauen Schnittstellenspezifikationen befinden sich im `<Functions>`-Repository, an der geplanten Integrationsstelle. Sie werden im *README* verlinkt, damit sie bei einer Weiterentwicklung des Backends genutzt werden können.

4.4.1. Spieldaten (National League)

Von der National League, der höchsten Schweizer Eishockeyliga, können Spielinformationen und vereinzelte Spielstatistiken abgefragt werden.

Push-Benachrichtigungen sind die primär zu nutzende Quelle für Informationen. Das Live-System versendet die neuen Informationen stets an interessierte Endpunkte. Dadurch wird die nötige Kontinuität ermöglicht. Es gibt kein Testsystem, weshalb nur während der aktiven Spielzeit getestet werden kann.

Pull-Benachrichtigungen sind auch möglich, sofern die IP-Adresse des aufrufenden Servers im Verbandsystem hinterlegt ist. Diese sind nur als Ergänzung zu den Pushmeldungen gedacht. Eine derartige Abfrage ist dafür jederzeit machbar.

Für diese Applikation sind folgende Informationen von Interesse.

Information	XML Datenpaket (ContentType)
Spielkader	ActualGames-XML
Spielerstatistiken	GameDetail-XML
	Topscorer-XML
	Topgoalkeeper-XML
Teamstatistiken	GameDetail-XML

Tabelle 50: Inhalte der National League Schnittstelle

Für weitere Informationen und Übermittlung der erforderlichen IP-Adressen oder Endpunkte ist Werner Schuller (`<werner.schuller@sihf.ch>`), IT-Verantwortlicher der National League, zu kontaktieren.

4.4.2. Newsletter (Automation Monkey)

Die SCRJ Lakers nutzen bereits die Newsletterverwaltung Automation Monkey vom Dienstleister *Marketingmonkeys*. Was fehlt, ist die Ansteuerung von dieser Applikation und Übermittlung newsletterwilligen Benutzer an diese *Marketingmonkeys*-Datenbank. *Marketingmonkeys* bieten eine *REST*-Schnittstelle zur Aktualisierung der Kontaktliste an. Die hinzugefügten E-mail-Adressen müssen aber zwingend vorher von dieser Applikation verifiziert werden, erst dann dürfen neue E-Mail-Adressen hinzugefügt werden. Aktuell werden neben der E-Mail-Adresse auch noch andere Werte erwartet.

- Anrede
- Vorname
- Nachname

Möglicherweise dürfen stattdessen generische Werte wie «Hallo Tippspiel Nutzer» eingesetzt werden. Das muss aber zwingend mit dem Kunden abgesprochen werden. Es ist auch bereits ein Testsegment eingerichtet, um die Prozesse testen zu können. Der Kontakt bei *Marketingmonkeys* ist Alain Habegger (`<alain@marketingmonkeys.ch>`).

4.4.3. Spielertracking (Kinexon)

Die Spieler führen während den Spielen einen Tracker mit sich, der die persönlichen Leistungen aufzeichnet. Kinexon bietet eine *REST* API, um die ausgewerteten Informationen abzugreifen. René Schmid (⟨rene.schmid@lakers.ch⟩), der IT-Verantwortliche der SCRJ Lakers, steht bei Fragen zur Seite.

5. Implementierung

In diesem Kapitel werden die Ergebnisse im Front- und Backend aufgezeigt. Die App ist mittels drei Repositories bestmöglich in ihre logischen Bestandteile unterteilt.

- Frontend
- Functions
- Deployment

Informationen, welche für die Wartung und Weiterentwicklung des Produktes nötig sind, werden in jedem Repository in Form eines *READMEs* beschrieben. Darin enthalten sind Anleitungen zur erstmaligen Systemeinrichtung und Weiterentwicklung sowie Erläuterungen zur Struktur und zum Inhalt. Ausserdem werden die verwendeten Bibliotheken und Frameworks aufgelistet und deren Lizenz erwähnt. Diese Inhalte können sehr kurzlebig sein und eignen sich dadurch nicht für diese Dokumentation.

5.0.1. Sprachen

Die primär verwendete Programmiersprache in diesem Produkt ist Typescript. Typescript liefert eine starke Typisierung. Dies unterstützt die Entwicklung, indem bereits zur Übersetzungszeit bestimmte Fehler aufgedeckt werden können. Es verringert die Fehleranfälligkeit und erhöht die Programstabilität. Für den produktiven Einsatz im Web beziehungsweise im Browser des Benutzers und für die Ausführung von Programmcode in Backend-Funktionen wird schlussendlich vorkompiliertes Typescript beziehungsweise Javascript verwendet. Beim Einsatz von Javascript für *Serverless* Functions kann eine merklich bessere Performance gegenüber Kompilersprachen wie C# und Java erzielt werden. Die Latenz von Funktionsstart bis -ausführung ist deutlich geringer, denn Kompilersprachen benötigen für die erfolgreiche Ausführung extra eine virtuelle Maschine und zusätzliche Code-Ressourcen während Javascript nur eine leichtgewichtige Laufzeitumgebung braucht.

Im Frontend wird zusätzlich *Syntactically Awesome Style Sheets (SASS)* als Stylesheet-Sprache eingesetzt. *SASS* gilt als CSS-Präprozessor und erweitert CSS um weitere nützliche Funktionen. Für den produktiven Einsatz wird *SASS* in CSS kompiliert.

5.0.2. Libraries und Frameworks

Verwendete Libraries und Frameworks können sich sehr rasch wieder ändern, deshalb sind diese detailliert in den jeweiligen *READMEs* aufgelistet. An dieser Stelle werden die persistenten und tief verankerten Pakete erläutert.

React

Gemäss der erarbeiteten Systemarchitektur ist das Frontend mit React implementiert und erfüllt die Anforderungen einer *Progressive Web Application (PWA)*. Für die Initialisierung der zurzeit aktuellsten Version 17.0.2 wurde Create-React-App[16] eingesetzt. Dieses Tool ermöglicht es, das React Template mit TypeScript und den benötigten *Service Worker* Konfiguration mittels einem Kommandozeilenbefehl zu generieren.

Semantic UI React

Klassische UI Komponenten wie Buttons, Icons, Modals, Ladeanimationen, Formulare und Popups werden nicht selber implementiert. Für eine effiziente Entwicklung des Frontends werden sie von der Semantic UI React Bibliothek eingebunden und gemäss den eigenen Anforderungen und Styleguides angepasst. Es wird die aktuell neuste Distribution mit der Version 2.1.3 verwendet. Dafür sind die benötigten Dateien (JavaScript sowie Stylesheet) im `<index.html>` eingebunden.

Jest

Jest ist das eingesetzte Test-Framework für Front- und Backend. Im Frontend sind die Vorteile nicht von der Hand zu weisen. Die starke Unterstützung von Jest für React und das Gross an bereits eingebauten Test-Funktionalitäten reduzieren die Aufwände zur Testeinrichtung enorm. Da im Backend ebenfalls mit

TypeScript programmiert wird, kann auch für das Backend Jest verwendet werden. Jest kann Tests mit einem Befehl auf verschiedenen TypeScript-Projekten gleichzeitig durchführen. Das vereinfacht das Testing mit den verschiedenen *Serverless Functions*.

AWS SDK

In ziemlich jeder Funktion wird ein weiterer AWS Service beansprucht. Mit der *AWS SDK* können solche Abfragen realisiert werden. Mittlerweile gibt es zwei aktiv gewartete Versionen - die Version 2 und 3. Diese unterscheiden sich in der Architektur und Syntax. Die Version 2 hat eine Monolith-Architektur, während die Version 3 modular aufgebaut ist. AWS Amplify setzt nach wie vor auf die Dienste der Version 2 und auch die Laufzeitumgebung der *AWS Lambdas* bringt standardmässig die *AWS SDK* mit Version 2 mit. Die Version 3 ist noch im aktiven Aufbau und muss analog einem normalen Package in den entwickelten Code integriert werden. Es gibt trotz dieser Umstände Bestrebungen auf Version 3 zu updaten, da die neue Syntax eleganter ist. Insbesondere im Umgang mit asynchronen Aufrufen, und die Package-Grösse durch die Modularität sehr viel kleiner ist. Erste Tests mit der neuen Version sind aber fehlgeschlagen. Deshalb wird dieses Vorhaben zurückgestuft. Die aktuelle Implementierung erfüllt seinen Zweck und kann auf sehr viel Erfahrung und Hilfestellungen zählen, was bei der neuen Version noch nicht vorhanden ist.[17]

5.1. Frontend

Die im *User-centered-Design*-Prozess entworfenen und validierten Designs werden in React als Funktions-Komponenten entwickelt. Für Lifecycle-Methoden und Zustände werden daher *Hooks* genutzt. Die Komponenten sind aufgeteilt in `<pages>` und `<components>`. Zu `<pages>` gehören jene, die im React-Router als Route genutzt werden. Die restlichen Komponenten von ihnen getrennt, da sie wiederverwendbar implementiert sind und auf verschiedenen Seiten dargestellt werden können.

5.1.1. Progressive Web Application

Damit eine React Applikation als *PWA* ausgeliefert werden kann, müssen einige Voraussetzungen erfüllt sein. Die wichtigste Komponente macht der *Service Worker* aus, der für die Installierbarkeit, Offline-Funktionalität und für weitere Tätigkeiten im Hintergrund benötigt wird. Dessen Implementierung besteht aus zwei Dateien. Die *Service Worker* Instanz wird in `<service-worker.ts>` festgelegt. Die Handhabung der Registrierung und Abmeldung wird in `<serviceWorkerRegistration.ts>` geregelt. Im letzteren wird ausserdem der *PushService* aufgesetzt, der benötigt wird um Push-Benachrichtigungen an das Gerät eines Nutzers senden zu können. Dafür müssen aber zuerst die Berechtigungen vom Benutzer eingeholt werden, was in diesem Fall beim Besuch der Tippspiel-Seite passiert.

Ein weiterer Bestandteil einer *PWA* ist das Manifest. Dieses beschreibt die wichtigsten Eigenschaften der App, wie den Namen, Farben, und Icons. Im Manifest muss mindestens ein Icon als `<maskable>` deklariert werden. Dieses benötigt einen Hintergrund und der Inhalt sollte so platziert sein, dass er bei allen möglichen Grössen und Formen von Verknüpfungen sichtbar ist.

5.1.2. State-Management

Für das Verwalten von Statusinformationen im Frontend werden verschiedene Möglichkeiten in Betracht gezogen. Die trivialste Lösung wäre die ausschliessliche Verwendung von lokalen Statusinformationen in den Komponenten mittels `<useState>`-Hooks, welche bei Bedarf den Sub-Komponenten übergeben werden können. Um ein sogenanntes *Property Drilling* zu vermeiden, wurde dieser Ansatz rasch verworfen. Folglich werden zwei andere Möglichkeiten evaluiert, welche beide Vor- und Nachteile aufweisen.

Redux

Eine bereits vertraute Variante ist der Einsatz von *Redux*, einem Status-Container für JavaScript Applikationen. Mit dessen Einsatz können React-Komponenten auf Statusinformationen des Containers und Dispatcher-Funktionen zugreifen, mit welchen der Status verändert werden kann. Dieser Ansatz ermöglicht eine übersichtliche Handhabung des State-Managements dank dem Einsatz von *Actions* und einem modularen *Reducer*. Dafür wird aber viel *Boilerplate*-Code benötigt. Dieser befindet sich sowohl bei der

Konfiguration als auch in Komponenten, welche auf die Statusinformationen zugreifen.[18]

Kontext

Eine ähnliche Lösung wie mit *Redux* kann auch durch den Einsatz der von React angebotenen Hooks `<useContext>` und `<useReducer>` erreicht werden. Auch hier wird viel Code für die Konfiguration benötigt, vor allem bei einer starken Typisierung mit TypeScript. Gleich wie bei *Redux* müssen `<Actions>` und `<Reducer>` implementiert werden, um die Änderung des States zu ermöglichen. Die Verwendung der Statusinformationen in Komponenten kann bereits mit einer Zeile Code sichergestellt werden, was den Einsatz sehr komfortabel macht.

Entscheidung

Beide Ansätze bieten eine brauchbare Lösung. Für das State-Management in diesem Produkt werden aber Kontext- und Reducer-Hooks eingesetzt, die von React zur Verfügung gestellt werden. Der bequeme Einsatz in den Komponenten hat überzeugt, da dieser Aspekt bei *Redux* etwas beschwerlich ist. Diese Auswahl des State-Managements hat sich im Verlauf des Projekts bewährt. Lediglich der grosse `<Reducer>` ist auf den ersten Blick etwas unübersichtlich.

Implementierung

Der Kontext ist auf Applikationsebene (`<App.tsx>`) gemäss dem deklarierten Schema (`<context.ts>`) initialisiert. Damit alle Komponenten auf diesen zugreifen können, befindet sich um alle Routen der Applikation ein Context-Provider. Diesem werden der initiale Zustand und die Dispatcher-Funktionen übergeben, welche mit dem `<useReducer>`-Hook generiert werden. Dadurch sind alle Voraussetzungen geschaffen, damit Komponenten mit dem `<useContext>`-Hook auf die Statusinformationen zugreifen können. Für Änderungen im Kontext sind insgesamt 43 ActionTypen implementiert, welche vom Dispatcher aufgerufen werden können.

AppState
challenges: Challenge[]
newGame: Game
activeGame: Game
players: IPlayer[]
sponsors: ISponsor[]
rewards: Reward[]
issuedRewards: IssuedRewards[]
user: IUser
ranking: Profile[]
imageSources: Object[]
error: Error

Abbildung 20: Typ des Zustands im Frontend

5.1.3. Authentisierung

Die Benutzerverwaltung im Frontend basiert auf dem User-Management-Prozess aus den Anforderungen. Aufgrund der spezifischen Eigenschaften, wie anonymen Benutzern und der Passwordless-Login-Option, sind die von AWS Amplify angebotenen UI-Komponenten nicht ausreichend. Die eigene `<Authenticator.tsx>`-Komponente und deren Bestandteile ermöglichen diese Features. Die Kommunikation mit AWS Cognito wird mit der `<Auth>`-Klasse von AWS Amplify vollzogen. Diese bietet alle nötigen Funktionen für das Benutzermanagement.

5.1.4. Styling

Als Designsprache für das Frontend wird *SASS* eingesetzt. Dabei handelt es sich um eine Erweiterung von CSS, dem Standard beim Styling im Web. *SASS* bietet einige Features, welche die Handhabung

der Styles vereinfachen. Beispielsweise können bei der Kompilierung mehrere Dateien zu einer einzigen Datei zusammengetragen werden. Dadurch wird eine übersichtliche Entwicklung ermöglicht. Trotzdem wird für die Auslieferung der Styles nur ein einziger HTTP-Request benötigt. Dank diesem Feature kann eine logische Trennung der globalen Styles gemäss dem *SASS* 7-1 Design-Pattern gemacht werden. Für die Darstellung von einzelnen React-Komponenten, hat jede ihre eigene *SASS* Datei.[19]

5.1.5. Custom Hooks

React Hooks erlauben einen Zustand zu halten, ohne Klassen schreiben zu müssen. Hooks sind die Funktionen, welche von Funktionskomponenten auf den React-Zustand und Lifecycle-Features zugreifen. React bietet bereits eine grosse Bandbreite an Hooks an, welche verschiedene Zwecke erfüllen. Für die Verwaltung von Ladeanimationen und Errors bei API-Aufrufen werden für dieses Projekt eigene Hooks entwickelt, welche die Wiederholung von Code vermeiden.

Der `<useAction>`-Hook lädt bei der Erfüllung einer Bedingung eine Ressource vom Backend. Der Rückgabewert des Hooks besteht aus einem Boolean `<loading>`, welcher den Ladezustand einer Ressource anzeigt, und einer `<error>`-Variable für Fehlermeldungen.

Für asynchrone Aktionen, welche eine Ladeanimationen auslösen sollen und ein Error-Handling benötigen, wird der `<useAsync>`-Hook eingesetzt. Dieser muss einmal pro Komponente initialisiert werden. Die mitgelieferte Callback-Funktion kann die asynchronen Aufrufe auslösen und die ebenfalls mitgelieferten Statusinformationen verändern.

5.1.6. Realtime Updates

Der eingesetzte API-Service AWS Appsync basiert auf *GraphQL* und bietet Realtime Updates. Diese Realtime Updates werden genutzt, um dem Benutzer ein dynamischeres Nutzungserlebnis zu bieten und ihn damit stärker an die App zu binden. In der `<Game.tsx>`-Komponente werden fünf Subscriptions verwendet. Diese führen beim Öffnen des Tippspiels zur Erstellung von Websockets. Darüber werden den Clients Ereignisse im Backend mitgeteilt.

onCreateChallengeApplication

Dem Tippspiel wird eine `<ChallengeApplication>` hinzugefügt. In diesem Fall wird die erstellte Challenge im State des aktiven Tippspiels gespeichert.

onDeleteChallengeApplication

Eine `<ChallengeApplication>` wird aus dem Tippspiel entfernt. Die gelöschte Challenge wird aus dem State des aktiven Tippspiels entfernt.

onUpdateChallengeApplication

Der Status einer Challenge hat sich geändert. Die Challenge wird im State des aktuellen Tippspiels aktualisiert. Dies könnte beispielsweise durch die Schliessung der Abstimmung passieren.

onEvaluateChallengeApplication

Das Resultat wurde bei einer Challenge eingetragen. Beim Auflösen einer Challenge werden im Backend die Tipps ausgewertet und die gewonnenen Punkte dem entsprechendem Profil zugeteilt. Im Frontend werden aus diesem Grund der abgegebene Tipp, das eigene Profil und die öffentliche Rangliste neu geladen und im State aktualisiert.

onUpdateGame

Ein neues Tippspiel wurde gestartet oder ein bestehendes wurde beendet. Beim Starten eines Tippspiels wird dieses im State gespeichert. Beim Beenden könnte ein Benutzer Preise gewonnen haben, weshalb die für ihn ausgestellten Preise geladen und entsprechend im State aktualisiert werden.

5.1.7. Error Management

Ein wichtiger Bestandteil des Error Managements sind die Custom Hooks, welche für API-Calls verwendet werden. Diese fangen potentielle Fehler ab und geben sie über die `<error>`-Variable zurück. Die Komponente, welche den Aufruf getätigt hat, ist anschliessend für die Verwaltung des Fehlers zuständig. Falls der Benutzer über den Fehler informiert werden soll, kann sie den Fehler in den Context der Applikation übergeben. Auf der Theme-Ebene werden solche Fehler in Form von einem Popup an den Benutzer übergeben. Das Popup wird mit der Error-Message für zehn Sekunden eingeblendet.




Abbildung 21: Error Popup

5.1.8. Caching

Das Frontend ist mit einem *Service Worker* ausgestattet, welcher statische Dateien im Cache ablegt. Solche statischen Inhalte sind zum Beispiel das Hintergrundbild auf der Startseite sowie HTML- und CSS-Code. Andere Bilder wie Sponsoren und Spielerporträts sind dynamisch und werden nicht im Cache gespeichert. Sie sind im AWS S3 Bucket (Kapitel 5.2.1) abgelegt und können mit einer `<Storage.get()>` Abfrage geladen werden. Bei diesem Aufruf wird eine 24h gültige URL generiert. Hinter dieser URL befindet sich das entsprechende Bild.

Um die Storage-Abfragen auf den S3 Bucket zu reduzieren, werden die URLs der Bilder im Context gespeichert. Solange vom Benutzer kein Page Reload gemacht wird (löscht den aktuellen Context), muss also auch kein S3 Request mehr getätigt werden.

5.1.9. Testing

Die Logik des Frontends wird in Form von automatisierten Unit Tests validiert. Um das Testing zu vereinfachen, wird bei der Entwicklung die Logik von Komponenten bestmöglich in Helper-Funktionen und Actions ausgelagert. Die Stufen von Komponenten- und Integrationstests werden im Frontend ausgeklammert. Diese bedeuten einen enormen Aufwand, bringen dem Produkt aber keinen ausreichenden Mehrwert. Durch den Einsatz von Cypress[20] für automatisierte End-to-end Tests werden die Komponenten und deren Kommunikation mit dem Backend bereits zu einem befriedigenden Mass getestet.

5.1.10. Accessibility

Eine zentrale Anforderung an das Produkt ist eine gute Accessibility. Der nötige Kontrast bei der Darstellung von Textinhalten wird mit einem fixen Farbschema eingehalten. Dieses basiert auf den bereits bestehenden Farben der offiziellen SCRJ Lakers Webseite. Im gesamten Produkt werden ausschliesslich jene Farben verwendet. Schwarze Schrift wird nur auf weissem oder hellgrauem Hintergrund dargestellt. Für blaue, rote, grüne oder dunkle Flächen wird eine weisse Schrift eingesetzt. Ausserdem wird gemäss den Nichtfunktionalen-Anforderungen ein Google Lighthouse Audit genutzt, um restliche Accessibility-Anforderungen zu testen.

5.2. Backend

5.2.1. AWS Cloud

Das Entwicklerteam verfügt über keine Erfahrung hinsichtlich Cloud Computing. Die Anbieter-Evaluierung stützt sich entsprechend stark auf Drittmeinungen und -einschätzungen. Sie wurde zusätzlich auf die beiden populären Anbieter Google Cloud und AWS beschränkt.

AWS ist der Platzhirsch in Cloud Computing. Die Menge an Diensten ist dort entsprechend am grössten. Die meistgenutzten und innerhalb dieses Projekts relevanten Funktionen bietet aber auch Google Firebase an.

Es werden die beiden populären Anbieter Google Cloud und AWS miteinander verglichen. AWS ist der Platzhirsch in Cloud Computing. Die Fülle an Diensten ist dort entsprechend am grössten. Die meistgenutzten und innerhalb dieses Projekts relevanten Funktionen bietet aber auch Google an. Beide Anbieter bieten für ein vereinfachtes Deployment von Webanwendungen eigene Entwicklungsplattformen an. Google nennt dieses Google Firebase und AWS hat AWS Amplify. In der Evaluierung werden diese Plattformen mit einbezogen und erhalten entsprechend grosses Gewicht.

Faktor	AWS Amplify	Google Firebase
Plattform	Open-Source Framework	Proprietäre Technologie
Preis	günstiger	teurer
Preismodelle	Free Tier Pay as you go	Free Tier Pay as you go
Cloud	AWS	Google Cloud Platform (GCP)
API	REST, GraphQL	REST
Datenbank	Relationale und nicht-relationale Datenbanken	Google Firestore (NoSQL-Datenbank)
Konfiguration	Mehr Vorarbeit erforderlich	Weniger aufwendig
Continuous Deployment	GitHub, GitLab, Bitbucket	Cloud Build
Testing	Build-in functions, Admin UI	Test Lab, A/B Testing

Tabelle 51: Vergleich von AWS und Google Cloud [21, 22]

Google Firebase überzeugt mit seiner Einfachheit. Die Lernkurve ist niedrig und die ersten Resultate kommen schnell. Die Lernkurve von AWS Amplify ist höher, diejenige von AWS ohne AWS Amplify nochmals viel höher. Die Preise sind dagegen sehr ähnlich, aber über alle benötigten Produkte gesehen doch noch etwas günstiger bei AWS. AWS Amplify weis auch mit den Deployment-Möglichkeiten zu überzeugen. AWS lässt verschiedenste DevOps-Plattformen zu, während für Google Firebase die Einrichtung von Cloud Build notwendig ist. Ein sehr grosser Pluspunkt ist die sehr einfache Integration von *GraphQL* durch AWS Amplify. Google bietet nur *REST* an. Generell sind viele Dienste auf den beiden Plattformen sehr ähnlich und heissen nur unterschiedlich. Mit einigen Abstrichen liesse sich die Applikation aufgrund dieser Datenlage mit beiden Plattformen umsetzen. Diese Auswertung stützt sich auf Informationen der jeweiligen Produktseiten beider Anbieter.

Da das Entwicklerteam über keine Erfahrung hinsichtlich Cloud Computing verfügt, zählt man auch auf die Meinungen und Einschätzungen eines Experten. Für die abschliessende Validierung der richtigen Lösung wird die Meinung von Mirko Stocker eingeholt. Aus den Anforderungen an die Infrastruktur, den Ergebnissen aus den Recherchen und der abschliessenden Meinung des Experten ist AWS beziehungsweise AWS Amplify als die beste Lösung für diese Applikation hervorgegangen.

Mit dieser Erkenntnis werden für das Tippspiel-Backend von AWS folgende Dienste benötigt. Wenn möglich, werden die Angebote vom Standort eu-central-1 (Frankfurt) genutzt, damit bestimmt die *Datenschutz-Grundverordnung der Europäischen Union (DSGVO)* eingehalten wird.

AWS Amplify

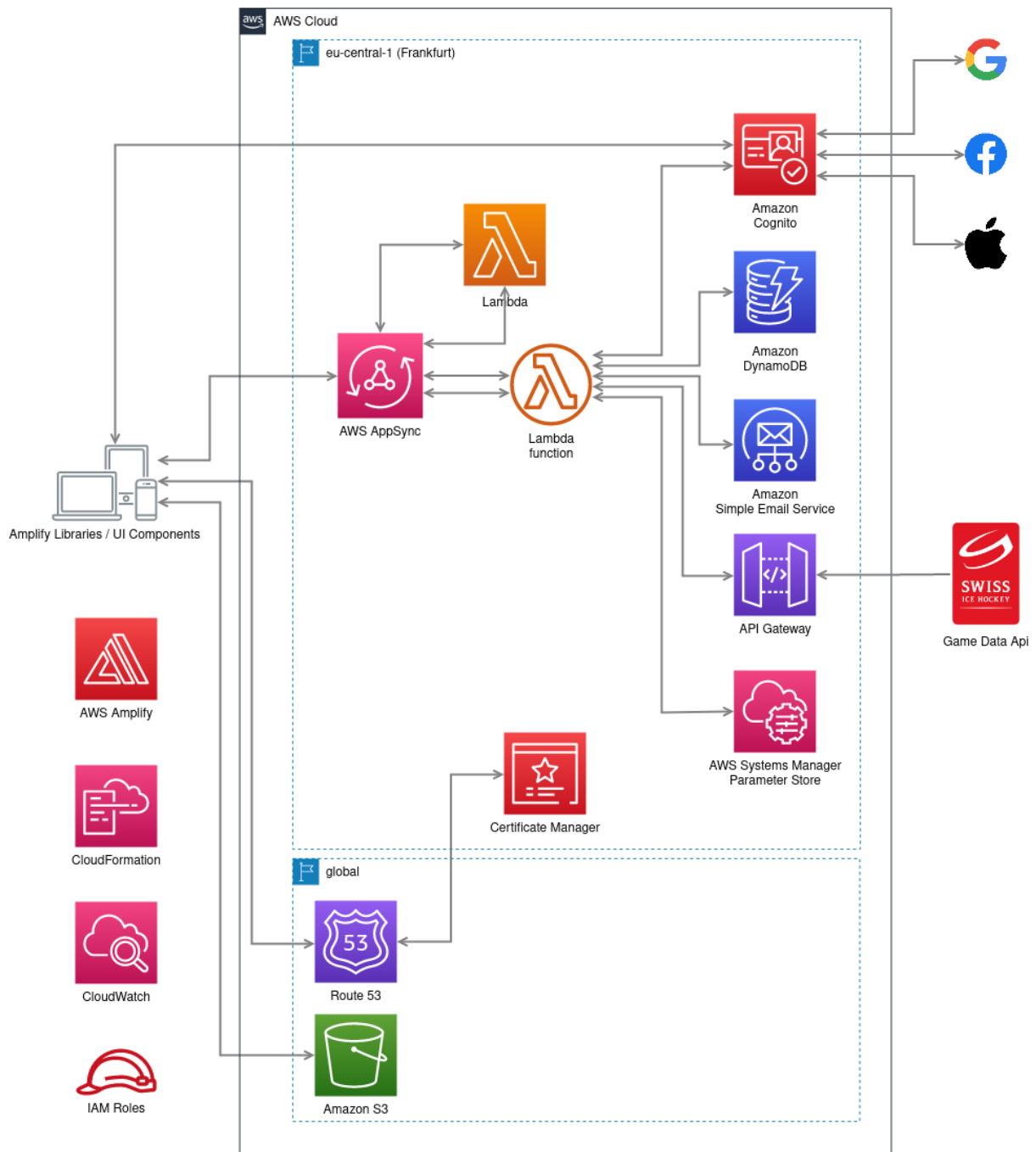


Abbildung 22: AWS Cloud Systemübersicht

AWS Amplify

AWS Amplify ist ein Backend-as-a-Service (BaaS). Damit können AWS Dienste in den Bereichen

- Authentication
- Storage
- *GraphQL* API
- DataStore
- Functions
- Hosting
- Push Notifications

schnell und einfach bereitgestellt werden. Es ist aber kein geschlossenes Ökosystem, sondern erlaubt auch die manuelle Integration beliebiger AWS Dienste ausserhalb von AWS Amplify. Es bietet neben einer *CLI* auch exklusive Libraries für eine vereinfachte Integration ins Frontend. Das Deployment von verschiedenen Environments und das Hosting im Web wird auch unterstützt.

AWS Amplify arbeitet deklarativ. Es erstellt aufgrund der *CLI*-Eingaben einfache Konfigurationsdateien und generiert daraus AWS-lesbare Formate wie Cloud Formation Templates und *GraphQL* Schemas. Insbesondere die Policies auf bestimmte AWS Ressourcen werden innerhalb dieses Prozesses in die Templates eingearbeitet. Aus dem einfachen *GraphQL* Schema lässt AWS Amplify Tabellen und Schnittstellen inklusive grundlegender Validierung und Autorisierer entstehen. Verschiedene AWS Dienste werden dadurch automatisch miteinander verbunden.

Das Backend des Tippspiels ist vorwiegend mithilfe von AWS Amplify aufgebaut und wird auch aktiv damit betrieben. So wird das abschliessende Deployment von Änderungen sowohl in Front- als auch im Backend über AWS Amplify gestartet und durchgeführt.

AWS Cloud Formation

Cloud Formation ist der Inbegriff für *Infrastructure as code* bei AWS. Das gesamte Deployment von AWS Diensten inklusive deren Konfiguration wird in sogenannten Cloud Formation Templates deklarativ abgebildet. Diese Templates werden von AWS Amplify automatisch generiert und anschliessend in einem Amazon S3 Bucket abgelegt und in die AWS Cloud übernommen. Dadurch hat dieser Service eine sehr hohe Bedeutung obwohl keine direkte Interaktion damit notwendig ist.

AWS CloudWatch

Mit CloudWatch können AWS Dienste überwacht werden. In diesem Projekt werden vor allem die Logs der Lambda Funktionen darin begutachtet.

AWS IAM Roles

Alles in AWS braucht seine Berechtigung. Dafür gibt es verschiedene vorbereitete Rollen und Benutzer. Eigene können ergänzt werden oder werden durch andere AWS Dienste, wie AWS Amplify, systematisch hinzugefügt. Auch jeder Entwickler benötigt Rechte, damit dieser die AWS Console und speziell AWS Amplify vollständig nutzen kann.

AWS Route 53

Das ist der DNS-Dienst von AWS. Für diese App ist eine Hosted Zone namens *game.lakers.ch* erstellt worden, was App-Zugriffe und valide Zertifikate von einer spezifischen Domain ermöglicht. Die von AWS Diensten benötigten Einträge werden meist durch diese automatisch erstellt. Zwingend darauf angewiesen ist das AWS Amplify Hosting und der Certificate Manager beziehungsweise indirekt der AWS SES E-Mail-Service.

AWS Certificate Manager

Zertifikate erhöhen das Vertrauen in die Dienstleistung und sind für bestimmte Zwecke zwingend. Deshalb gibt es ein Wildcard-Zertifikat für die Domain *game.lakers.ch*, dass die Integrität gewährleistet. So können beispielsweise vertrauenswürdige E-Mails von dieser Domain versendet werden.

AWS Simple Storage Service (S3)

S3 erlaubt die Speicherung von Objekten. Aktuell werden zweierlei Dinge persistent gespeichert.

Bucket	Beschreibung
amplify-scrjgame-\${env}*-*-deployment	CloudFormation Templates und kompilierter Code zuhanden Deployment
scrjgame1945mediafiles*-\${env}	Dynamische Medien aller Art für das Frontend. Momentan sind das Bilder von Spielern und Sponsoren.

Tabelle 52: Buckets

AWS AppSync

Bei diesem Service handelt es sich um das Eintrittstor zum Backend. Alle Anfragen vom Frontend gehen über diese Schnittstelle, ausgenommen sind Authentifizierungsdienste. Er erlaubt die Nutzung von *GraphQL*, einer modernen und effizienten Datenabfrage- und Manipulationssprache, und garantiert eine sichere Verbindung zu Datenquellen wie AWS DynamoDB und Lambdas. Beispielsweise prüft AppSync die Autorisierung bei geschützten Eintrittspunkten. Zudem erlauben sogenannte Subscriptions die Datenaktualisierung in Echtzeit mittels Websockets.

AWS DynamoDB

DynamoDB ist eine AWS eigene *Serverless*-NoSQL-Key-Value-Datenbank. Darin werden sämtliche dynamischen, textbasierten Informationen der Applikation abgelegt. Aktuell braucht die App zwölf Tabellen

- Challenge
- ChallengeApplication
- ChallengeTip
- Game
- GamePlayer
- IssuedReward
- Observer
- Player
- Profile
- Reward
- RewardApplication
- Sponsor

AWS Lambda

Nicht alle Funktionen der App können mit einfachen CRUD-Anfragen an die Datenbank abgedeckt werden. AWS Lambda Funktionen werden entweder über bestimmte AppSync-Aufrufe oder über backen-dinterne Ereignisse ausgelöst. Dabei decken die Lambdas unterschiedliche Funktionsbereiche ab. Einige Funktionen erweitern die CRUD-Funktionen, indem sie mehr Mutationen auf der Datenbank gleichzeitig vornehmen und damit die Anfrage fürs Frontend stark vereinfachen. Einige Funktionen schränken dagegen die Mutationen ein, weil bestimmte Bedingungen im Vorfeld geprüft werden müssen.

- evaluateChallengeApplication
- evaluateGame
- generateRanking
- redeemIssuedReward
- setChallengeTip
- updateOwnProfile

Ein grosser Anteil der entwickelten Funktionen sind für die Authentifizierungsprozesse im Zusammenhang mit dem AWS Cognito erforderlich. Diese passen die Standardprozesse von AWS Cognito auf die appspezifischen Bedürfnisse an. Die Anmeldung ohne Passwort und anonyme Benutzer als erweitertes Authentifizierungskonzept sind nicht im Standardpaket enthalten und müssen selber komplett implementiert werden. Der Lebenszyklus des mit dem Account verknüpften Datenbankeintrags mit den persönlichen App-Errungenschaften muss ebenfalls eigenständig verwaltet werden.

- deleteAnonymousUser
- deleteRegisteredUser
- migrateAnonymousUser
- userpoolanonymousCleanup
- userpoolanonymousPreSignup
- userpoolregisteredCreateAuthChallenge
- userpoolregisteredDefineAuthChallenge
- userpoolregisteredPostAuthentication
- userpoolregisteredPreAuthentication
- userpoolregisteredPreSignup
- userpoolregisteredVerifyAuthChallengeResponse

Damit bei bestimmten Ereignissen der Benutzer mit Push-Nachrichten versorgt werden kann, muss einerseits das Ereignis festgestellt und andererseits eine Nachricht an die Bedürftigen weitergeleitet werden. Beispielsweise sollen Benutzer über einen Spielstart rechtzeitig benachrichtigt werden, ohne dass diese selber aktiv werden müssen.

- onGameMutation
- sendPushNotifications

Im Ansatz wurden einige externen Dienste eingebunden beziehungsweise hat man diese in Betracht gezogen. Einzelne Lambda Funktionen dienen deshalb als Schnittstelle für solche Drittanbieterdienste.

- onNationalLeaguePush
- subscribeToNewsletter

Es gibt noch einige automatisch generierte Lambdas von Amplify, diese sind mit dem Prefix `amplify-scrjgame- $\{env\}$` gekennzeichnet.

- amplify-scrjgame- $\{env\}$ -*-HostedUIProvidersCustomR-*
- amplify-scrjgame- $\{env\}$ -*-UpdateRolesWithIDPFunci-*
- amplify-scrjgame- $\{env\}$ -*-authTriggerFn7FCFA449-*
- amplify-scrjgame- $\{env\}$ -*-UserPoolClientLambda-*
- amplify-scrjgame- $\{env\}$ -*-OAuthCustomResource-*
- amplify-scrjgame- $\{env\}$ -*-authus-RoleMapFunction-*
- amplify-scrjgame- $\{env\}$ -*-HostedUICustomResource-*

AWS Cognito

Die Registrierung und Anmeldung von Benutzern sowie die Zugriffskontrolle auf die Dienste des Backends werden mit AWS Cognito abgewickelt. Dieser Dienst unterstützt auch die Anmeldung mit sozialen Identitätsanbieter wie Apple, Facebook und Google. Apple-ID als weitere Anmeldeoption wurde im aktuellen Release weggelassen, da dies mit zusätzlichen jährlichen Kosten (100 CHF) gegenüber Apple verbunden wäre und dies fürs erste nicht zwingend ist.

Es werden zwei Userpools verwendet, um unterschiedliche Einstellungen setzen und Anforderungen an einen Benutzer stellen zu können. Benutzer in beiden Pools können die gleichen AWS Dienste nutzen. Jedoch werden diese im Hintergrund teils unterschiedlich berücksichtigt und dann andere Mutationen auf der Datenbank vorgenommen.

Der anonyme Userpool `<userpoolanonymous>` dient der anonymen Identifikation. Dieser hält alle anonymen Benutzer und wickelt dessen Prozesse ab. Trotzdem muss ein solcher Benutzer registriert werden. Dafür wird lediglich ein Benutzername und Passwort benötigt. Diese Angaben werden im Hintergrund vom User-Management im Frontend generiert. Die Lebensdauer eines solchen Benutzers ist beschränkt. Nach Ablauf einer Frist wird der Account und die damit verbundenen Benutzerdaten von einer Lambda-Funktion unwiderruflich vollständig gelöscht.

Davon befreit sind Benutzer im registrierten Userpool `<userpoolregistered>`. Diese bleiben auf ewig bestehen, ausser der Benutzer möchte von sich aus den Account löschen. Bei der Registrierung wird die E-Mail und ein Nickname verlangt. Diese Informationen werden für die spätere Anmeldung und erweiterten Features benötigt. Alternativ kann das Benutzerprofil mit einem Google oder Facebook Account verknüpft werden. Dann ist keine weitere Registrierung mit E-Mail mehr erforderlich. Zugriff auf privilegierte Funktionen erhalten Benutzer, die der Benutzergruppe Admin angehören.

AWS Simple E-Mail Service (SES)

Der AWS SES wird verwendet, um dem Benutzer eines registrierten Accounts den Magic Link für die Anmeldung zukommen zu lassen. Dafür wird eine Identität namens `<game.lakers.ch>` angelegt, welche das Versenden von der E-Mail `<signin@game.lakers.ch>` erlaubt.

AWS Systems Manager Parameter Store

Amplify erlaubt die gesonderte Speicherung für von AWS Lambdas verwendete Secrets. Diese werden im Parameter Store vom AWS Systems Manager gespeichert. Es gibt aktuell zwei eingesetzte Secrets.

Name	Funktion	Zweck
Token Secret	<code>...registeredPreAuthentication</code>	Signierung von JWT-Tokens
Token Secret	<code>...VerifyAuthChallengeResponse</code>	Verifizierung von JWT-Tokens
Vapid Private Key	<code>sendPushNotifications</code>	Versand von Push-Benachrichtigungen

Tabelle 53: Secrets

AWS API Gateway

Nicht immer verwenden externe Dienstleister bereits *GraphQL*. Häufiger trifft man *REST* oder XML an. Die National League nutzt letzteres für ihre Push-Nachrichten. Der API Gateway von AWS verarbeitet solche Anfragen und kann diese analog AppSync an Lambdas weiterleiten oder anders verbleiben.

5.2.2. Benutzerverwaltung

Im Zentrum der Benutzerverwaltung steht AWS Cognito. Mit diesem Dienst werden verschiedene Lambda Funktionen verknüpft, welche auf bestimmte Authentifizierungsereignisse reagieren. Der Userpool `<userpoolregistered>` ist direkt im Amplify konfiguriert. Da Amplify nur einen User pool vorsieht, ist der zweite Userpool `<userpoolanonymous>` als Custom Ressource in AWS Amplify eingebunden.

Authentifizierungsstufen

- Unauthenticated
- Authenticated

Zugriffe auf die App sind sowohl mit als auch ohne Authentifizierung möglich. Eine durch Amplify generierte Rolle im AWS IAM erlaubt AppSync-Anfragen von nicht authentifizierten Benutzern. Derartige Benutzer können aber keine Modifikationen in der App vornehmen. Solche Anfragen sind authentifizierten Benutzern vorenthalten. Die spezifischen Berechtigungen sind im *GraphQL*-Schema gesetzt.

Benutzertypen (authentifiziert)

- Anonymous
- Registered

Der Benutzer kann die Applikation mit zwei unterschiedlichen Benutzertypen nutzen. Das Mehr an Funktionen steht beiden Typen zur Verfügung. Der Übergang zwischen den beiden Typen ist fließend und für den Benutzer praktisch unsichtbar.

Benutzerrollen

- User
- Admin

In dieser App können registrierte Benutzer zwei verschiedene Benutzerrollen einnehmen. Diese sind durch die Userpool Gruppe <Admin> voneinander unterscheidbar. Benutzer in dieser Gruppe können zusätzlich zur User-Rolle auch die Rolle <Admin> einnehmen.

E-Mail-Registrierung

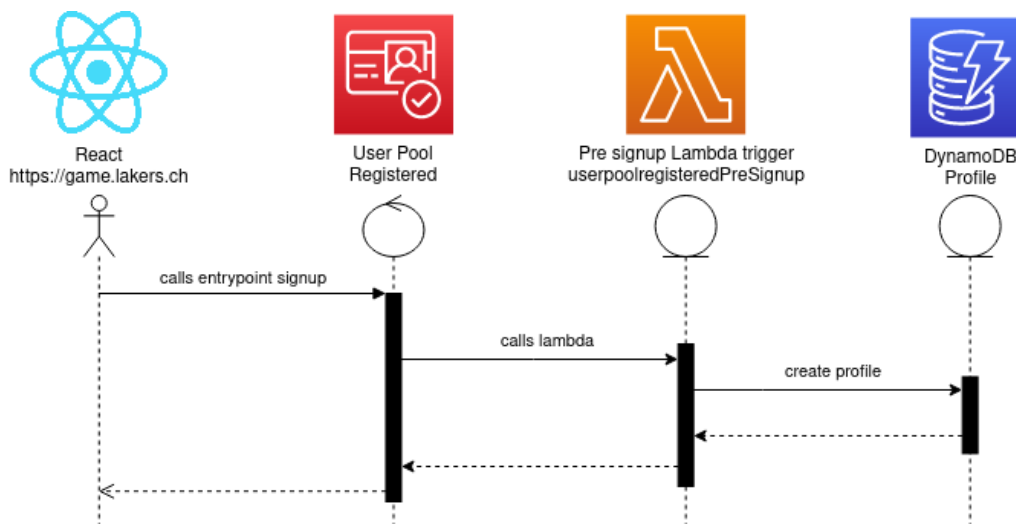


Abbildung 23: E-Mail-Registrierungsprozess

Der Benutzer schickt eine Registrierungsanfrage mit E-Mail, Nickname und Profil-Sichtbarkeit an AWS Cognito. Dieses Ereignis löst den Lambda Trigger <userpoolregisteredPreSignup> aus. Dieses Lambda erstellt sodann ein neues Profil.

```

1  {
2    Password: "GEBD/_w'Tq:r6c$'?^fAv%\\+",
3    UserAttributes: [
4      0: {
5        Name: "nickname",
6        Value: "Isoldemadusche"
7      }
8    ],
9    Username: "user@example.com",
10   ValidationData: [
11     0: {
12       Name: "visibility",
13       Value: "private|public"
14     }
15   ]
16 }
  
```

Listing 1: Auszug der Post-Anfrage SignUp

Registrierung mit Login Provider

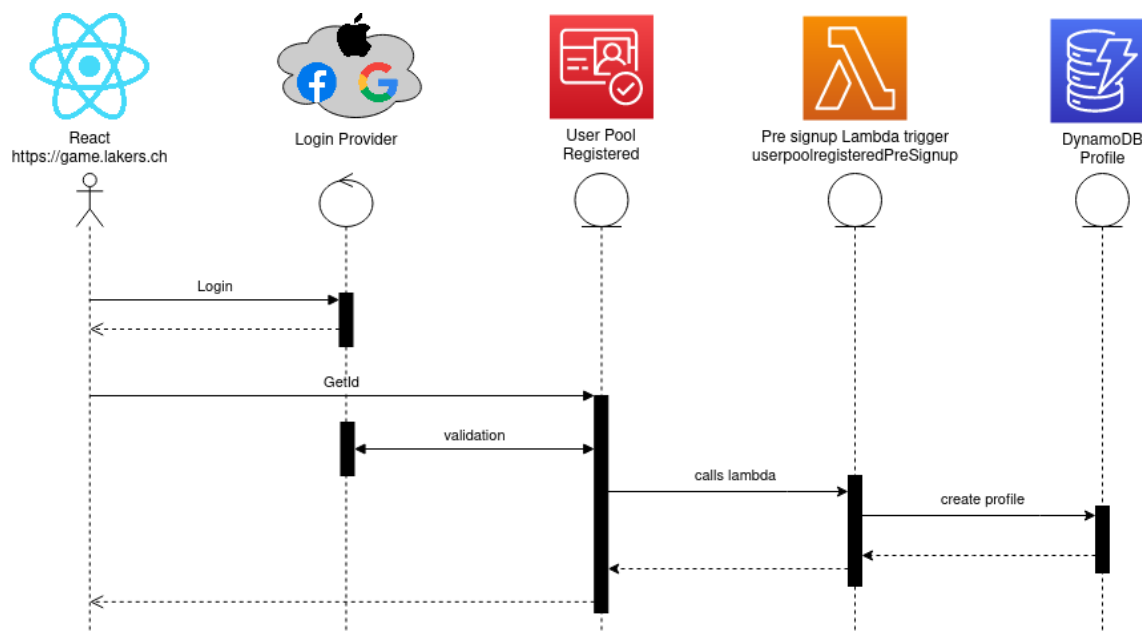


Abbildung 24: Login Provider Registrierungsprozess

Der Benutzer meldet sich zuerst bei dem gewählten Login Provider an und erlaubt dessen Nutzen für diese Applikation. Der authentifizierte Account wird an AWS Cognito geschickt und dessen Integrität gegenüber dem Login Provider verifiziert. Nach erfolgreicher Verifizierung wird der Signup Lambda Trigger `<userpoolregisteredPreSignup>` aufgerufen, worin ein persönliches Profil erstellt wird. Ausserdem wird der Benutzer verifiziert, damit sich dieser auch anmelden kann. Abschliessend findet zu diesem Zeitpunkt die erste Authentisierung statt. Dem Profil fehlt noch der Nickname und dessen Sichtbarkeit. Diese Informationen werden nach dem automatischen Login eingefordert und durch eine gewöhnliche Profilaktualisierung persistiert.

Anonyme Anmeldung

Damit ein anonymer Benutzer verwendet werden kann, muss zuerst ein solcher im System registriert werden. Dafür wird für die spätere Anmeldung ein sicheres Passwort generiert und ein Benutzername im Format eines Universally Unique Identifier (UUID) festgelegt. Mit dem von AWS Cognito ausgelösten Lambda `<userpoolanonymousPreSignup>` wird die Anmeldung freigeschalten.

```

1  {
2    Password: "T{TntX0/z6usi3vpMGh6;wcX",
3    UserAttributes: [],
4    Username: "7561163d-753b-4ec5-be86-1bbc0d2f049e",
5    ValidationData: null
6  }
  
```

Listing 2: Auszug der Post-Anfrage SignUp

Im Anschluss an die Registrierung wird der anonyme Benutzer einmalig angemeldet. Danach übernimmt der Refresh Token die fortlaufende Authentifizierung bis zum Lebensende des Benutzers.

```

1  {
2    AuthFlow: "USER_PASSWORD_AUTH",
3    AuthParameters: {
4      USERNAME: "7561163d-753b-4ec5-be86-1bbc0d2f049e",
5      PASSWORD: "T{TntX0/z6usi3vpMGh6;wcX",
6    }
7    ClientMetadata: {}
8  }
  
```

Listing 3: Auszug der Post-Anfrage InitiateAuth

E-Mail Anmeldung

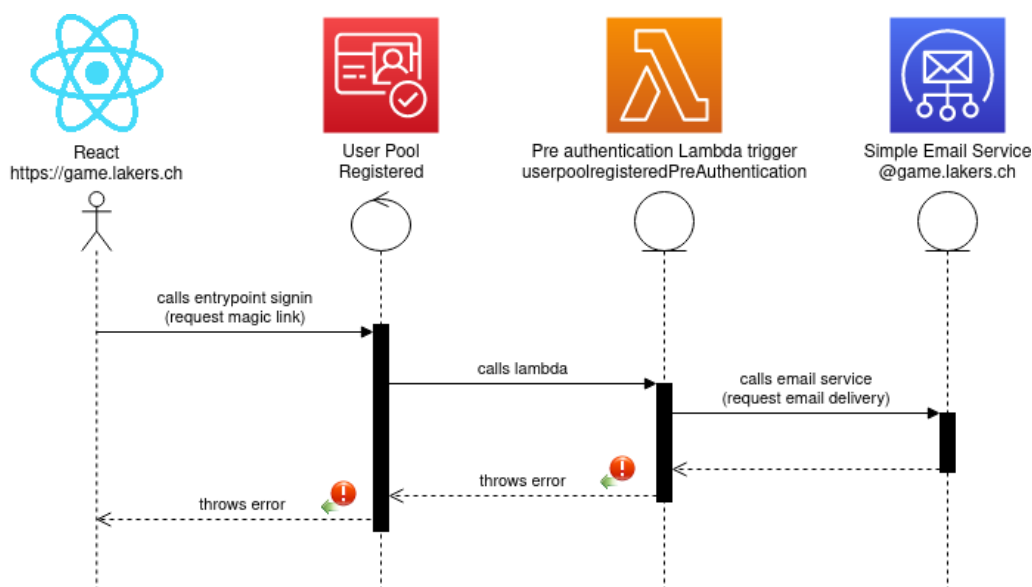


Abbildung 25: E-Mail-Anmeldeprozess

Die Anmeldung mit E-Mail ist zweiphasig. Das Frontend schickt den Benutzernamen zusammen mit einem Vermerk, dass es sich um eine Anmeldung ohne Passwort handelt, innerhalb einer Signin-Anfrage an AWS Cognito.

```

1  {
2    AuthFlow: "USER_SRP_AUTH",
3    AuthParameters: {
4      USERNAME: "user@example.com",
5    }
6    ClientMetadata: {
7      passwordless: "true",
8      migrationToken: "ey..."
9    }
10 }

```

Listing 4: Auszug der Post-Anfrage InitiateAuth

Folglich wird das Lambda `userpoolregisteredPreAuthentication` ausgeführt. Darin wird ein temporärer JWT-Zugangstoken in einen Magic Link verpackt und mit dem AWS SES per E-Mail an den Benutzer retourniert.

```

1  {
2    "email": "user@example.com",
3    "username": "aedc3b08-6300-48a0-b045-20ce3cddc496",
4    "migrationToken": "ey..."
5  }

```

Listing 5: Payload des temporären Tokens

Da AWS die Authentifizierung an dieser Stelle fortsetzen würde, wird bewusst ein Fehler 400 zur sofortigen Beendigung der Anmeldeanfrage geworfen.

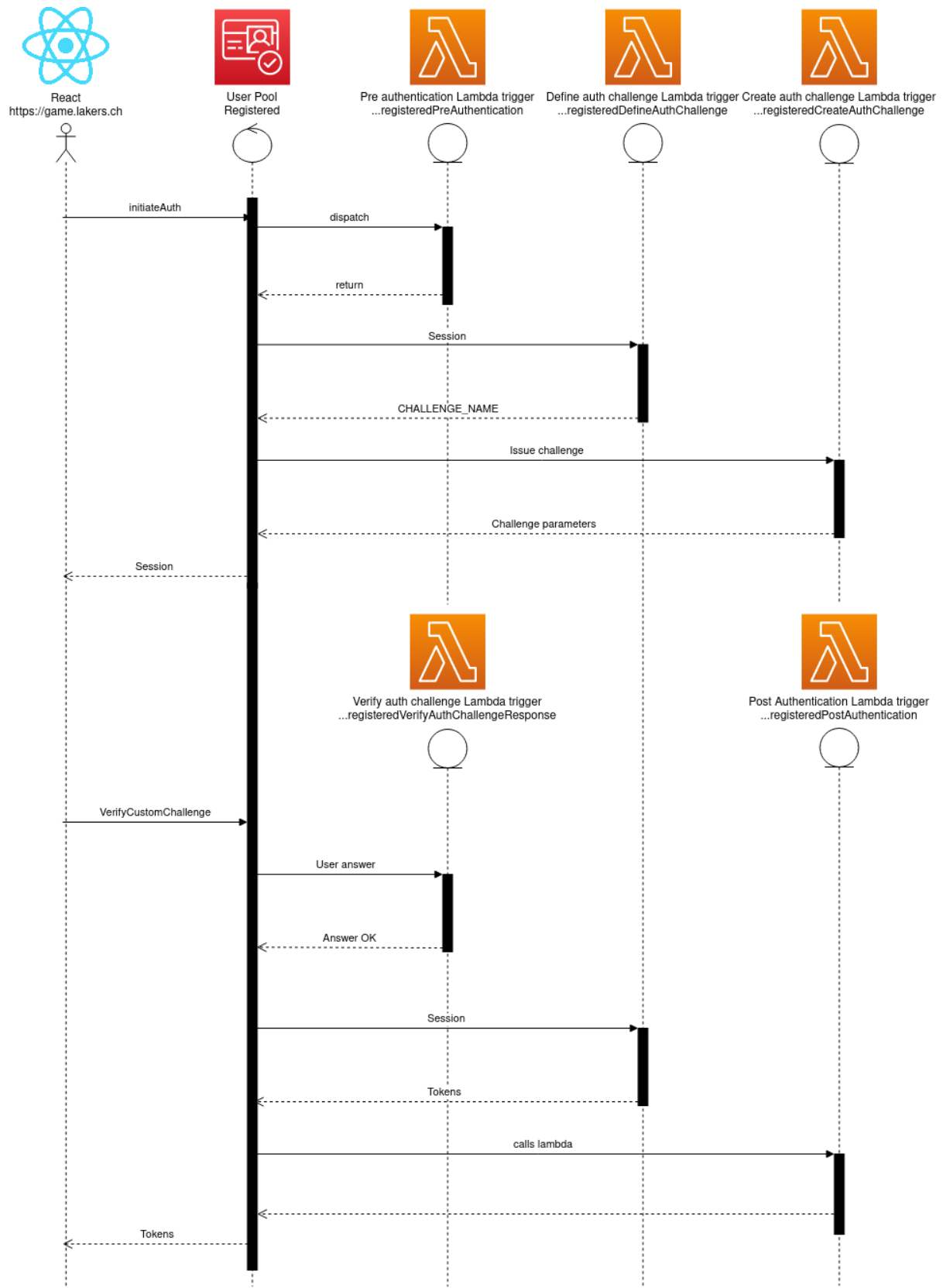


Abbildung 26: E-Mail-Anmeldeprozess

In der zweiten Phase wird zuerst nochmals eine Anmeldeanfrage gesendet, um ein Benutzerobjekt mit einer aktiven Session zu erhalten und die Challenge zu initialisieren. Da bei dieser Anfrage das klassische Passwort fehlt, wird die Anfrage als `<CUSTOM_AUTH>` interpretiert.

```
1  {
2    AuthFlow: "CUSTOM_AUTH",
3    AuthParameters: {
4      USERNAME: "user@example.com",
5    }
6  }
```

Listing 6: Auszug der Post-Anfrage `InitiateAuth`

Damit werden weitere Cognito-Ereignisse ausgelöst. Der Challenge-Prozess beginnt mit dem Lambda `<userpoolregisteredDefineAuthChallenge>`. Dieses Lambda wird im ganzen Prozess insgesamt zweimal aufgerufen. Einmal, um die Custom Challenge zu initialisieren und einmal um die Custom Challenge am Ende abzuschliessen. Das Lambda prüft, ob die Bedingungen für eine erfolgreiche oder abweisende Verifikation erfüllt sind. Ist die Challenge-Antwort nicht zufriedenstellend, wird eine neue Challenge erstellt und dafür das Lambda `<userpoolregisteredCreateAuthChallenge>` aktiviert. Dieses erzeugt die zu lösende Challenge. Daraufhin kann das Frontend eine `<sendCustomChallengeAnswer>` mit dem Token aus dem E-Mail an AWS Cognito senden. Damit lässt sich die `<CUSTOM_CHALLENGE>` verifizieren.

```
1  {
2    ChallengeName: "CUSTOM_CHALLENGE",
3    ChallengeResponses: {
4      ANSWER: "ey...",
5      USERNAME: "user@example.com"
6    }
7  }
```

Listing 7: Auszug der Post-Anfrage `RespondToAuthChallenge`

Mit aktiver Benutzer-Session und der obigen Anfrage wird das Lambda `<userpoolregisteredVerifyAuthChallengeResponse>` ausgelöst. Darin wird der JWT-Token auf seine Gültigkeit im Zusammenhang mit dem angegebenen Benutzer geprüft. Stimmen die Angaben, ist die Authentifizierung erfolgreich und das Lambda `<userpoolregisteredDefineAuthChallenge>` kann in seinem zweiten Aufruf die abschliessende Token-Erstellung an den Benutzer veranlassen. Ist dieser Anmeldeversuch der erste, wird die E-Mail im Lambda `<userpoolregisteredPostAuthentication>` ebenfalls als verifiziert markiert.

Login Provider Anmeldung

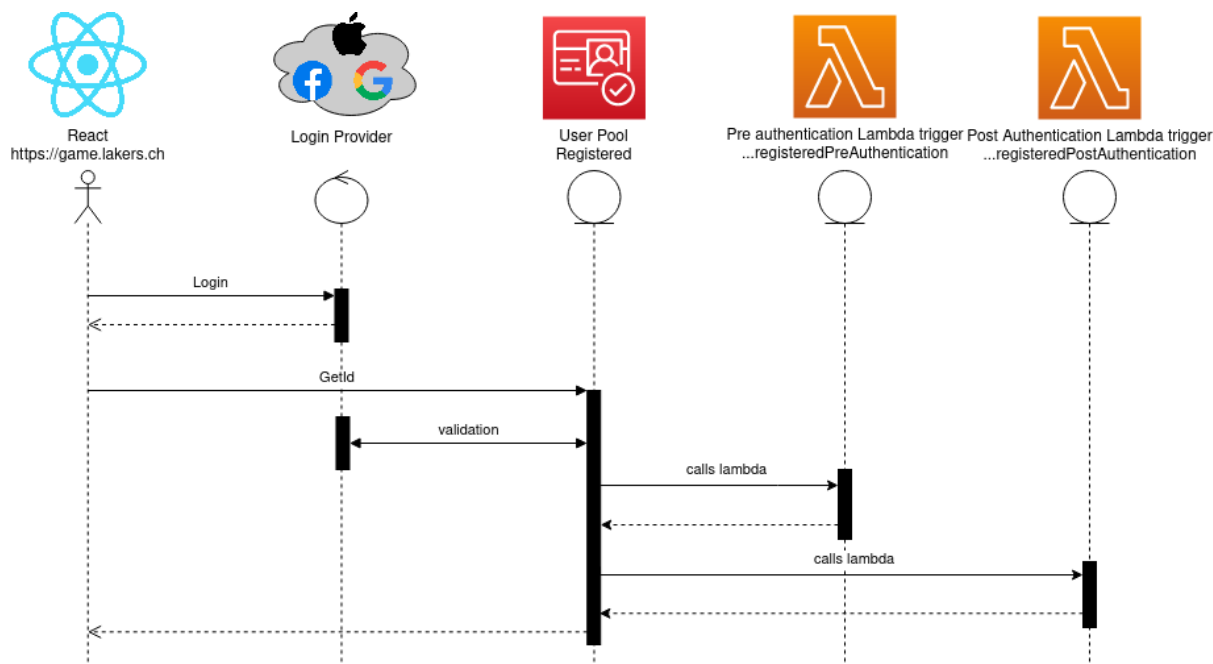


Abbildung 27: Login Provider Anmeldeprozess

Die Anmeldung mit einem Login Provider unterscheidet sich nur minimal zum Registrierungsprozess. Der erste Teil ist derselbe, jedoch werden abschliessend andere Ereignisse und entsprechende Lambda Trigger ausgelöst. In diesen Funktionen werden für diese Loginart keine weiteren Aktionen vorgenommen.

Zugangsdauer

Der Refresh Token eines anonymen Benutzers ist 30 Tage, jener eines registrierten Benutzers ist 180 Tage lang gültig. Da einem anonymen Benutzer die Zugangsdaten fehlen, gehen sämtliche Daten nach Ablauf dieser Frist verloren.

Benutzerlöschung

Die Löschung eines Benutzers verlangt auch die Löschung seiner persönlichen Inhalte. Diese unterscheiden sich je nach Benutzertyp. Es gibt für beide Benutzertypen eine eigene Funktion, die solche Löschungen vornimmt.

	Anonymer Benutzer	Registrierter Benutzer
Funktion	deleteAnonymousUser	deleteRegisteredUser
Zu löschende Inhalte	<ul style="list-style-type: none"> • Userpool User • ChallengeTips • IssuedRewards 	<ul style="list-style-type: none"> • Userpool User • Profile • ChallengeTips • IssuedRewards

Tabelle 54: Benutzerlöschung

Benutzerzombies

Das Lambda `<userpoolanonymousCleanup>` veranlasst die Löschung jener anonymen Benutzer, welche die maximale Nutzungsdauer von 30 Tagen überschritten haben. Das hält die Daten im Userpool `<user-`

poolanonymous» konsistent und wartbar. Die Säuberung wird täglich um 2 Uhr morgens ausgeführt.

Benutzermigration

Ist man im Besitz eines gültigen Access Tokens von einem anonymen Benutzer kann man dessen persönliche Inhalte mit der Lambda Funktion <migrateAnonymousUser» auf einen registrierten Benutzer übertragen.

5.2.3. Profil

Ein Profil speichert die persönlichen Informationen eines registrierten Benutzers, welche über ein einzelnes Spiel hinaus konsistent bleiben.

Funktion	nickname	visibility	score
evaluateChallengeApplication			+points
userpoolregisteredPreSignup	X	X	
migrateAnonymousUser			=score
updateOwnProfile	X	X	

Tabelle 55: Profil-Mutationen

Ranking

Der <rank» jedes Profiles muss immer aktualisiert werden, wenn sich die Sichtbarkeit eines Profiles oder dessen Punktestand ändert. Dafür wird das Lambda <generateRanking» ausgelöst.

Nickname

Der Nickname ist der sichtbare Benutzername in der App. Es ist die einzige von Benutzern anpassbare Information, die unter Umständen auch anderen Benutzern angezeigt wird. Gleichzeitig ist es auch die einzige Information, welche redundant im AWS Cognito für den Benutzer und auch im persönlichen Profil gespeichert wird. Deshalb erfordert der Nickname eine besondere Aufmerksamkeit bei dessen Mutationsprozess. Ein Validierer soll dieses Datenfeld rudimentär vor Missbrauch schützen. Der Validierer missbilligt folgende Muster

- URLs
- E-Mail
- HTML
- Wörter mit weniger als 3 Zeichen
- Wörter mit mehr als 20 Zeichen
- Obszöne Wörter

Die Liste an obszönen Wörtern ist das Produkt mehrerer einfacher Blacklist-Textdateien. Diese Listen können beliebig mit weiteren Wörtern ergänzt werden. Enthält ein Nickname eines der oben genannten Muster, wird dieser Änderungsantrag abgelehnt.

Benutzereingaben

Das Lambda <updateOwnProfile» validiert die vom Benutzer beantragte Profiländerungen. Die möglichen Änderungen beschränken sich auf die Informationen Sichtbarkeit und Nickname. Ein geänderter Nickname muss konsistent im AWS Cognito und Profil aktualisiert werden. Die Sichtbarkeit wird nur mit einem vorhandenen Nicknamen auf <public» gestellt.

Stats

Die Statistiken sind Bestandteil des Profils. Insgesamt sind drei Funktionen für Änderungen an den Statistikwerten verantwortlich.

Funktion	participatedGames	participatedTips	successfulTips
evaluateChallengeApplication		+1	(+1)
evaluateGame	+1		
migrateAnonymousUser	=games	=tips	=successfulTips

Tabelle 56: ProfilStats-Mutationen

5.2.4. Challengeverwaltung

Alle Challenges unabhängig ihres Types werden im gleichen Datenmodell abgespeichert. Die unterschiedlich interpretierbaren Attribute werden als String beziehungsweise als String-Liste gespeichert und basierend auf dem ChallengeTyp in das erforderliche Format (zurück) konvertiert. Dies erlaubt sehr einheitliche Prozesse für verschiedene ChallengeTypen, bedingt dafür ein gemeinsames Verständnis vom Frontend und Backend für die Attribute.

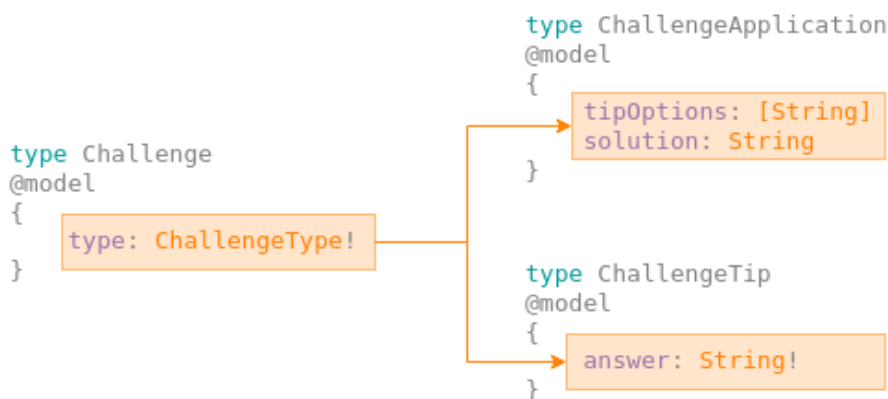


Abbildung 28: Gleiche Attribute für alle ChallengeTypen

Die folgende Tabelle zeigt die definierten Datenformate in den dynamischen Attributen für jeden ChallengeTyp.

ChallengeTyp	tipOptions	solution	answer
onePlayerChallenge	[Player.ID]	Player.ID	Player.ID
oneOfManyChallenge ¹	[String]	String	String
estimationChallenge ¹	-	Number	Number
resultChallenge ¹	-	/\d*\D*\d*/g ²	/\d*\D*\d*/g ²

¹ Noch nicht im Frontend implementiert ² Regex für Muster: Zahl-Nichtzahl-Zahl, bspw. 3:5 oder 7-2

Tabelle 57: ChallengeTypen und ihre Datenformate in den dynamischen Attributen

Tippen

Mit dem Lambda <setChallengeTip> können Tipps abgegeben werden. Da jeder Benutzer nur einen Tipp pro <ChallengeApplication> abgeben darf, wird ein bestehender Tipp überschrieben oder ein neuer Tipp erstellt. Tipps dürfen nur im Status <Open> abgegeben werden. Der Status wird mittels drei Attributen von der entsprechenden <ChallengeApplication> ermittelt.

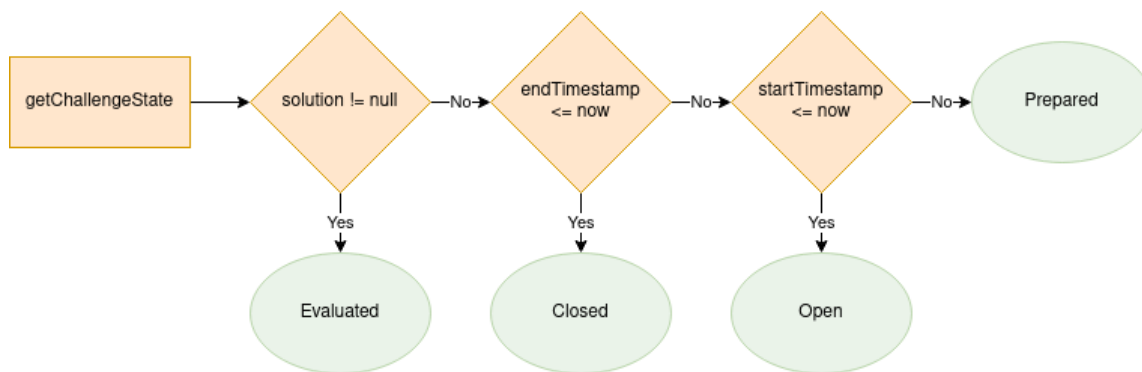


Abbildung 29: Challenge States

Evaluierung

Eine Challenge kann durch den Aufruf der Lambda <evaluateChallengeApplication> evaluiert werden. Die darin enthaltene <EvaluatorFactory> wählt basierend auf dem ChallengeTyp den passenden Evaluator für die dazugehörigen Challenge Tipps aus. Die eigentliche Auswertung der Tipps ist unabhängig ihres Challenge Typs immer dieselbe.

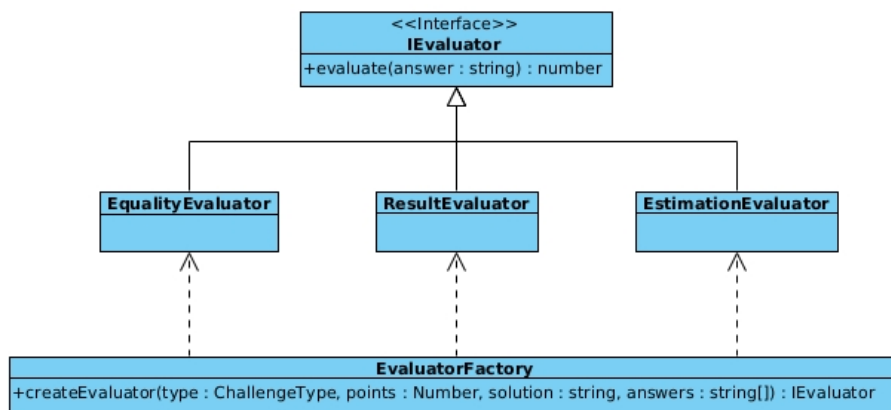


Abbildung 30: Simple Factory Evaluator

Challengetyp	Evaluator
onePlayerChallenge	EqualityEvaluator
oneOfManyChallenge ¹	EqualityEvaluator
estimationChallenge ¹	EstimationEvaluator
resultChallenge ¹	ResultEvaluator

¹ Noch nicht im Frontend implementiert

Tabelle 58: Challengetypen und ihre Auswertungsstrategie

5.2.5. Spielverwaltung

Evaluierung

Das Lambda `<evaluateGame>` wertet ein Spiel aus. Es enthält die gesamte Logik zur Preisverteilung.

5.2.6. Preisverwaltung

Einlösung

Es muss verhindert werden, dass Preise missbräuchlich mehrfach eingelöst werden können. Deshalb kann die Einlösung nur durch das Lambda `<redeemIssuedReward>` vollzogen werden.

5.2.7. Testing

Die Logik der Lambda Functions ist mit Unit Tests getestet. In Fokus steht das gesamte Spielkonzept. Automatische Integrationstests mit der DynamoDB sind keine implementiert, da in allen evaluierten Methoden jede Schema-Änderung im Datenbank-Mock manuell übernommen werden muss. Dieser Wartungsaufwand steht nicht im Verhältnis zum Ertrag.

5.3. Deployment

Alle Anleitungen zum Deployment von Entwicklung- und Produktivumgebungen sind in entsprechenden Repository *READMEs* beschrieben.

5.3.1. Infrastruktur

Damit die Applikation funktioniert, benötigt es eine auf die Applikation zugeschnittene Umgebung. Das Tippspiel wird in folgenden Systemen betrieben, entwickelt und getestet.

System	Bezug/Zweck
localhost	Die Applikation wird lokal weiterentwickelt und getestet. AWS Amplify liefert eine Testumgebung, damit eine Vielzahl der Backend-Dienste lokal simuliert werden können.
gitlab.ost.ch	Eine <i>GitLab</i> -Instanz der OST, worin der gesamte Prozess Softwareentwicklung vom Commit-Push bis zum Amplify-Deployment abgewickelt wird. Zusätzlich wird darin das Projektmanagement betrieben und der Product Backlog geführt.
AWS Cloud	Verschiedene Dienste, die miteinander verbunden, das Produkt in den Environments <code><Dev></code> und <code><Prod></code> anbieten.
sinv-56032.rj.ost.ch	Eine virtuelle Maschine, worauf alle eigenen Tools zuhanden Projektmanagement und Entwicklung laufen.
sinv-56033.rj.ost.ch	Eine virtuelle Maschine, worauf Tests durchgeführt werden.

Tabelle 59: Infrastruktur

5.3.2. Continuous Deployment (CD)

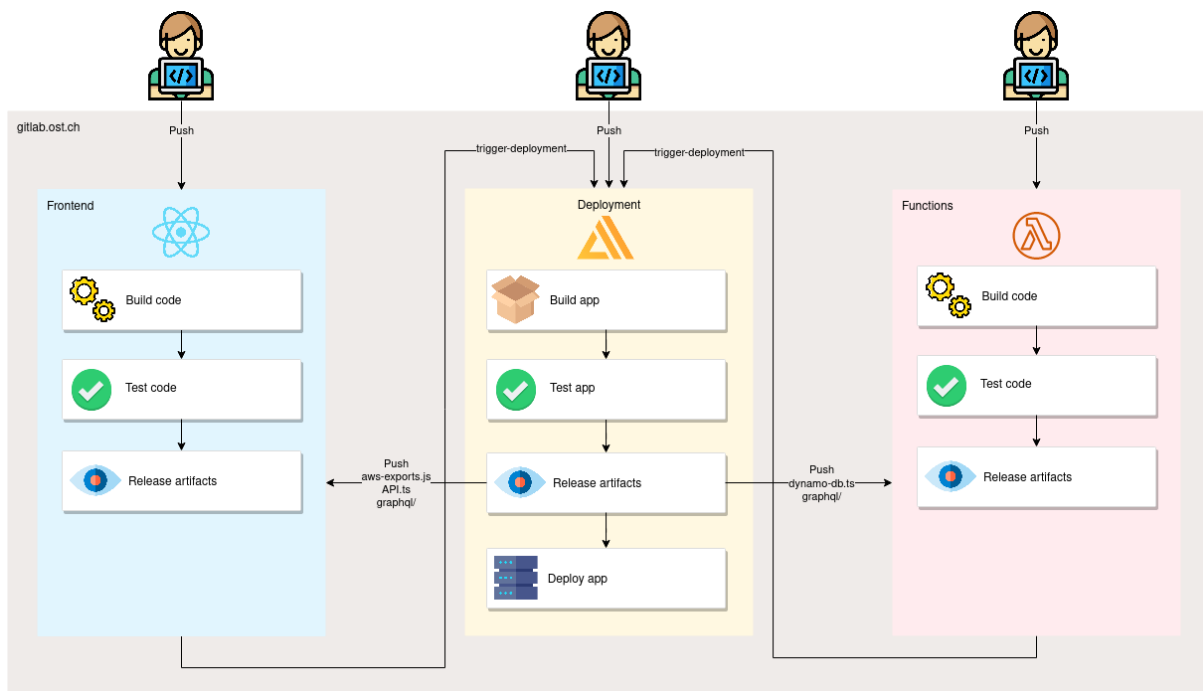


Abbildung 31: GitLab Pipelines

Mithilfe der *GitLab* Pipeline werden Pushes von Commits in den `<Dev>`- oder `<Main>`-Branch der Repositories `<Frontend>`, `<Functions>` und `<Deployment>` automatisch in der AWS Cloud veröffentlicht. Hauptverantwortlich für diesen Prozess ist das Repository `<Deployment>`. Dieses enthält sämtliche für Amplify erforderliche Ordner und Dateien mit Konfigurationen. Ergänzt wird diese Struktur mit den Inhalten aus den Repositories `<Frontend>` und `<Functions>`. Somit liegt eine vollwertig konfigurierte Amplify Applikation vor. Das definitive Deployment wird in der Stage `<Deploy>` ausgeführt.

Das Deployment in die AWS Cloud kann aber auch manuell lokal von einer Entwicklungsmaschine gemacht werden.

5.3.3. Client Code Generation

AWS Amplify generiert auf Nachfrage den Schnittstellencode zum definierten *GraphQL*-Schema. Dieser ist sowohl im Frontend als auch in den Funktionen eingebunden. Neben der manuellen Aktualisierung werden Änderungen im *GraphQL*-Schema auch durch die *GitLab* Pipeline des `<Deployment>`-Repository im Stage `<Release>` in den darauf angewiesenen Repositories `<Frontend>` und `<Functions>` aktualisiert.

5.3.4. Frontend

In der Amplify Projekt-Konfiguration ist der Pfad zum Frontend-Projekt, dessen Typ und der passende Build- sowie Startbefehl definiert. Damit kann Amplify im Deploymentprozess das Frontend selbständig kompilieren und später starten.

5.3.5. Functions

	Entwicklung	Betrieb
Sprache	TypeScript	JavaScript (CommonJS)
Inhalte	<pre>common/**/* <function-name>/src-ts/**/* node_modules/**/*</pre>	<function-name>/src/index.js

Tabelle 60: Entwicklungs- und Betriebsumgebung

Jede Funktion hat seine eigene Laufzeitumgebung und Eintrittspunkt. In AWS Amplify gibt es deshalb für die Konfiguration und Logik jeder Funktion einen eigenen Ordner. In Unterordner `<src>` liegt das vom Lambda auszuführende Javascript-Projekt mit einer `<index.js>` Datei und allenfalls nötigen `<package.json>` Datei. Diese enthält den Eintrittspunkt. Amplify kann nur diese vordefinierte, unveränderliche Dateienstruktur richtig interpretieren und verarbeiten. Geschriebener Code muss daher exakt in dieses Zielformat gebracht werden. AWS Amplify und AWS Lambda unterstützen grundsätzlich auch kein TypeScript. Die Kompilierung in akzeptiertes CommonJS muss vorgängig veranlasst werden.

Trotzdem soll über den ganzen Code hinweg das Prinzip *Don't repeat yourself (DRY)* eingehalten werden und zweckgleicher Code wie beispielsweise unterschiedliche Datenbankzugriffe auch im Code zusammenliegen.

Geteilter Code

Dafür existiert neben dem funktionsspezifischen Code ein für alle Funktionen verwendbarer Ordner namens `<common>` mit mehrfach genutzten Codefragmenten. Dieser Ordner liegt zwangsläufig ausserhalb der Code-Projekte einer Funktion und bildet ein eigenes TypeScript-Projekt ab. Damit Code des `<common>`-Projekts in Funktionscode wiederverwendet werden kann, muss `<common>` durch TypeScript Project References[23] und `<paths>`-Links im `<tsconfig.json>` eines Funktion-Projektes referenziert werden. Das vereinfacht die Kompilierung über Projektgrenzen hinweg und macht die Nutzung von projekt-fremdem TypeScript-Code erst möglich. Damit kann ein sehr gut wartbarer Code garantiert werden.

Gebündelter Code

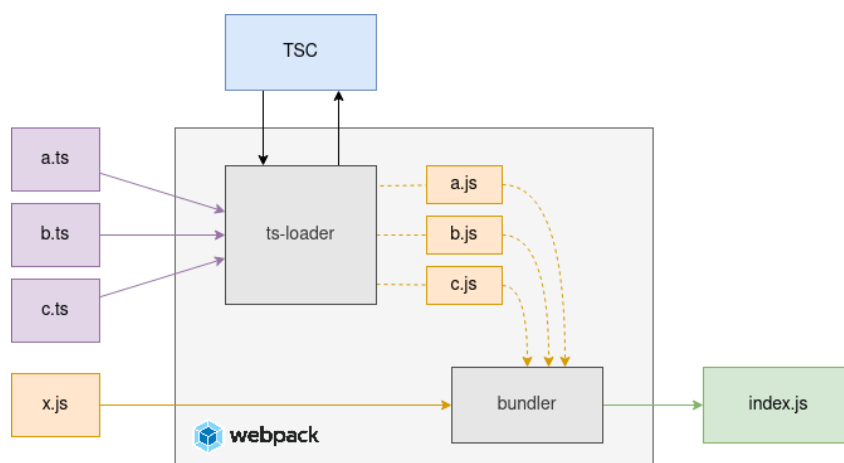


Abbildung 32: Aufbau Webpack

Bundling, konkret mit dem Bundler webpack[24], bietet einige Vorteile. So vereinfacht es den Build und Transfer von ausführbarem Code in die Zielumgebung einer Funktion. Mithilfe von webpack lässt sich im Vergleich zum rudimentären TypeScript-Building sehr viel zuverlässiger der genaue Speichername und -ort festlegen. Da die Abhängigkeiten zu Packages durch die direkte Code-Integration wegfallen, müssen

auch keine Packages mehr in der Zielumgebung installiert werden. Die Kopie von `<package.json>` in den Zielordner wird dadurch obsolet. Es verbleibt eine einzige Datei namens `<index.js>`, welche sämtliche Anforderungen an eine Funktion erfüllt und die gesamte Funktionslogik beinhaltet.

Der Wegfall der Packages und nicht referenzierter Codeabschnitte macht den statischen Output sehr viel kleiner als die vielen losen Dateien mit Abhängigkeiten zu ganzen Packages.

Multi-Building

Eine letzte Herausforderung ist die möglichst allgemeingültige Anstossung der Build-Prozesse jeder Funktion. Ziel sind von AWS Amplify veröffentlichbare und von AWS Lambda ausführbare Funktionen. Man muss bedenken, dass vereinzelt Funktionen nur in lesbarem JavaScript geschrieben sind, weil deren Inhalt zu geringfügig für eine Typisierung mit TypeScript ist. Andere Funktionen sind auf Dateien angewiesen, die im Bundling nicht berücksichtigt werden und extra kopiert werden müssen.

Mit `npm workspaces`[25] können nicht nur die spezifisch oder mehrfach benötigten Packages einzelner Funktionen gesamtheitlich verwaltet werden, was die Package-Installation um ein Vielfaches beschleunigt, sondern auch gebündelt mehrere Skripte mit dem gleichen Zweck in spezifizierten Workspaces gestartet werden.

5.4. Einschränkungen

Einige der eingesetzten Technologien und Produkte zeigen gewisse Schwächen. Die wichtigsten Erfahrungen und Erkenntnisse werden aufgelistet.

5.4.1. AWS DynamoDB

Als Datenbank wurde die Dynamo DB von AWS gewählt. Im Verlauf des Projekts zeigte sich, dass die dokumentbasierte Datenbank Einschränkungen bei einer grossen Datenmenge mit sich bringt. Das Problem betrifft List-Queries auf Tabellen mit vielen Einträgen. Die Datenbank liefert die Daten mit einer Art Pagination aus. Falls die Datenmenge 1 MB oder 100 Einträge übersteigt, beinhaltet das Resultat neben den Daten einen Next-Token, mit welchem man die nächsten Elemente abfragen könnte.[26]

Eine Lösung zu dieser Einschränkung ist, dass man neben der ID einen Secondary-Index definiert.[27] Die Datenbank erstellt dabei eine Zwischentabelle mit einer alternativen Key Struktur. Bei der Implementation der Tippspiel Abfrage kann aber nicht auf einen Secondary Index zurückgegriffen werden. Grund dafür ist, dass nach dem Tippspiel gefiltert wird, welches noch keinen `<StartTimestamp>`, respektiv `<EndTimestamp>` hat. Somit existiert zu diesem Zeitpunkt kein Anhaltspunkt für einen Key.

Um die beschriebene Einschränkung zu umgehen, wurde bei der List Query das Item Limit manuell erhöht. Die Filterung wird nach dem Lesen der Items angewendet und das Resultat wird anschliessend an den Client geliefert.[28]. Die betroffene Query liefert immer nur maximal ein Element, sodass bei der Query nie die 1MB Daten (führt zu Pagination) übertroffen wird.

5.4.2. AWS Cognito

Parameter- Weitergabe an Trigger

AWS Cognito stellt Trigger für verschiedene Authentifizierungsereignisse bereit. Aber nur an drei Trigger kann das Frontend beliebige `<ValidationData>` oder `<ClientMetaData>` zustellen.

- Pre signup
- Pre authentication
- User migration

Das erschwert die Speicherung von Benutzereingaben, die nicht für AWS Cognito gedacht sind und daher dort nicht erwartet werden. Beispielsweise hat die Newsletteranmeldung und die Sichtbarkeit des Profils nichts mit der Authentifizierung zu tun, sondern sind applikationsspezifische Informationen. Dennoch

werden sie im Registrierungsprozess abgefragt. Dessen Datenspeicherung und die darauf aufbauenden Prozesse müssen durch die Limitierung zwingend im `<Pre authentication>` Trigger angestossen werden.

Bei der Registrierung mit Login Provider kann ebenfalls nichts mitgegeben werden. Applikationsspezifische Informationen müssen nachträglich eingefordert werden.

Passwordless-Unterstützung

Weder AWS Amplify noch AWS Cognito bieten eine eingebaute Passwordless-Lösung an. Aber es gibt einen Blog, der die Integration von Passwordless E-Mail Authentifizierung in AWS Cognito beschreibt. [29] Dieses Konzept unterscheidet sich aber in verschiedenen Punkten von der in dieser Applikation eingesetzten Lösung. So muss der Benutzer einen Code aus der E-Mail in der geöffneten Session zur Verifizierung eingeben. Möchte man einen klickbaren Link statt einzugebenden Code für die Verifikation anbieten, verändert sich die Nutzung der einzelnen Trigger signifikant.

Der Magic Link öffnet naturgemäss eine neue Session in der Applikation. Damit kann kein Rückschluss auf bisherige Authentifizierungsschritte der alten Session wie die bereits erfolgte Anforderung eines Magic Links gezogen werden. Das Frontend muss selbstständig unterscheiden und AWS Cognito mitteilen, ob ein neuer Magic Link beantragt oder der erhaltene Token eingelöst werden möchte. Somit trifft auch den passwordless Loginprozess die Limitation mit der Parameter-Weitergabe. Der Magic Link muss dadurch zwingend im `<PreAuthentication>` Trigger statt erst innerhalb des `<CreateAuthChallenge>` Triggers ausgelöst werden. Da Custom Auth Flows den `<PreAuthentication>` Trigger nichtmal aufrufen, muss diese Anfrage als normale Authentifizierung getarnt werden. Nach der Versendung des Magic Link ist die Authentifizierung noch unvollständig aber keineswegs fehlerhaft. AWS Cognito sieht im `<PreAuthentication>` Trigger keine abwartende Haltung vor, diese ist anderen Triggern vorenthalten. Deshalb muss ein geplanter Fehler geworfen werden, obschon alles korrekt ist. [30]

User pools API-Anfragen

AWS Cognito API-Anfragen sind unterschiedlichen Limitierungen ausgesetzt. Von Bedeutung für diese Applikation sind nur die Anfrageratenquoten. Diese werden sekundlich zurückgesetzt. Nichtsdestotrotz müssen die Anfragen an diesen Dienst möglichst tief gehalten werden. Einige empfohlenen Massnahmen werden bereits angewendet. Im Backend stossen noch vier Funktionen eine API-Anfrage an.

- Verifizierung der E-Mail beim ersten Login
- Nickname-Mutationen
- Benutzerlöschung
- Userpool Aufräumung

Die meisten Quoten können alternativ durch einen formellen Antrag angehoben werden.[31]

5.4.3. AWS AppSync

Fehlerbehandlung

AWS AppSync erlaubt keine formatierte Fehler-Rückgabe, die von einer Funktion ausgeht. Stattdessen muss ein Fehler geworfen, dieser zusätzlich in Response-Resolvem von AppSync aufgefangen und in das akzeptierte Zielformat gebracht werden. Momentan wird darauf verzichtet. Die Response von AppSync bildet den Fehler deshalb nur durch den HTTP-Code, den Errorotypen `<Lambda:Unhandled>` und die mitgegebene Fehlermessage ab.[32]

Beschränkte dynamische Autorisierung

Das Modell `<Profile>` braucht eine dynamische tag-basierte Autorisierung. Denn in der Applikation, im Frontend, sind zwar nur die `<public>` Profile sichtbar, im Hintergrund können aber auch die `<private>` Profile abgefragt werden. Ein Filter-Argument in der Abfrage sorgt schlussendlich für den Unterschied.

Aktuell kann der Zugriff auf private Profile nur mit etwas Bastelei gänzlich verhindert werden. Es gibt die Dynamic group authorization, womit AppSync basierend auf Userpool Gruppen autorisiert, die namensgleich als Wert in einem Attribut des Datenobjektes vorkommen. Der zugreifende Benutzer muss

dieser Gruppe dann angehören, um entsprechende Rechte auf die Ressource zu haben.[33]

Im Workaround müssen sämtliche Benutzer der Gruppe `<public>` angehören. Das Attribut `<visibility>` des Modells `<Profile>` wird als Gruppen-Information haltendes Feld markiert und die Leserechte auf diese dynamische Gruppenautorisierung beschränkt. Steht der Wert `<private>` im Feld, könnten nur noch Benutzer der nicht existenten Gruppe `private` diese Profile lesen. Der Zugriff auf öffentliche Profile für nicht authentifizierte Benutzer über IAM kann als Konsequenz nicht mehr erlaubt werden.

Alternativ müssten sämtliche Profilabfragen über einen benutzerdefinierten Resolver beziehungsweise ein benutzerdefiniertes Lambda geschleust werden.

5.4.4. AWS Amplify

Self-hosted GitLab Instanz

In AWS Amplify können sehr einfach verschiedene DevOps-Plattformen wie *GitLab* angebunden werden. AWS Amplify bemüht sich hierbei selbstständig um die neusten Versionen der Applikation. Der stetige Versionsabgleich ist gewährleistet.

Diese Unterstützung beschränkt sich leider auf die Online-Version von *GitLab*. On-Premise betriebe GitLab Instanzen können nur mit dessen Administratorrechten und grösserer Anstrengung gleichwertig in AWS Amplify integriert werden.[34]

Diese Applikation wird mit einem selbstgebauten Pipeline-Prozess manuell in die AWS Cloud transferiert.

Team workflow

AWS Amplify teilt einige Konzepte von Git. Es gibt Environments, welche bei Git den Branches entsprechen. Mit der Amplify *CLI* kann man Änderungen pushen oder pullen. Bei einem Pull muss man sich aber entscheiden, ob die lokalen Änderungen beibehalten oder der produktive Stand von der Cloud lokal übernommen werden soll. Eine Mischform gibt es nicht. Die Merge-Mechanismen sind noch zu wenig ausgereift, dass sowohl lokale als auch in die Cloud gepushte Aktualisierungen in eine lokale Konfiguration übernommen werden können. Dieser Umstand sorgt regelmässig für viele Überschreibungen der Konfiguration anderer. Solange auf verschiedenen Environments gearbeitet wird, gibt es keine direkten Berührungspunkte und damit auch keine Probleme. Arbeiten mehrere Entwickler aktiv im gleichen Environment gibt es dagegen regelmässig Konflikte. Dass es umständlich und frustrierend sein kann als Team mit Amplify zu arbeiten war schon nach der Amplify Research bekannt. Zum Beispiel wird es in einem YouTube Video[35] bei den Negativpunkten zu Amplify angesprochen.

AWS beschreibt auf einigen Seiten, wie Teams mit Amplify zu arbeiten haben.[36] Für diese Applikation übernimmt GitLab das Deployment von AWS Amplify. Somit gilt als persistent, was in den Git-Repositories vom GitLab-Projekt liegt.

5.4.5. AWS Billing

Fehlende Echtzeitdaten

Die angegebenen Kosten in AWS Billing sind jeweils ungefähr 12 Stunden alt. Generiert man einen `<AWS Usage Report>`, beinhalten diese immerhin schon die Werte von vor 4 Stunden. Aufgaben mit Bezug zu den Finanzen, wie das Kalkulieren der effektiven Kosten, erstrecken sich dadurch über einen längeren Zeitraum.

Fehlender Kill-Switch

In der Entwicklungs- und Testphase möchte man das Risiko einer Kostenexplosion aufgrund Programmier- oder Konfigurationsfehlern möglichst gering halten. Auch die Gefahr eines ungeplant oder geplanten DDoS-Angriffs ist gegeben.

Im AWS Billing können Budgets erstellt werden. Übersteigen die Kosten einen vordefinierten Schwellenwert, können automatisch Benachrichtigungen (Alarme) oder Aktionen ausgelöst werden. Die möglichen

Aktionen beschränken sich auf das Anpassen von IAM Policies und Stoppen von EC2-Instanzen. Beachtet man die maximal mögliche Datenaktualität verzögert sich dieser Effekt um mehrere Stunden. In dieser Zeit können ein abnormales Verhalten weitere Schäden verursachen.

Für eine Cloud Computing Umgebung gibt es seitens AWS somit keine integrierte Methode, um die gesamte Applikation phasenweise zu stoppen oder anzuhalten. Auch AWS Amplify bietet nichts dergleichen. Im Internet gibt es für den eigenen Bedarf entwickelte Beispiele eines Killswitches. Für den Einsatz in einem produktiven Umfeld wird aber dringend davon abgeraten. Zur Bekämpfung von DDoS-Angriffen empfiehlt sich stattdessen präventiv die AWS WAF zu aktivieren. Dieser Dienst kostet zusätzlich, kann aber günstiger sein als ein böswilliger Angriff auf die Infrastruktur.[37] Eine kostenlose alternative ist die manuelle Abschaltung von im öffentlichen Netzwerk verfügbaren Diensten.

AWS Dienst	Massnahmen
AppSync	⟨Default authorization mode⟩ auf ⟨API key⟩ setzen, die anderen Provider entfernen
Cognito	App clients löschen
S3	Frontend: In AWS Amplify Zugriffskontrolle mit Passwort aktivieren Buckets: ⟨Block all public access⟩ auf ⟨On⟩ setzen
API Gateway	Throttling bei allen Methoden auf 0 setzen

Tabelle 61: AWS Dienste deaktivieren

Damit sollte der Zugang von ausserhalb unterbunden werden. Für eine Rückkehr in den Normalbetrieb müssen die Massnahmen wieder rückgängig gemacht werden.

5.5. Qualitätssicherung

5.5.1. SonarQube

Im Deploymentprozess der *GitLab* Repositories ⟨Functions⟩ und ⟨Frontend⟩ wird in der ⟨Release⟩ Stage die Code-Qualität geprüft. Dies ist auf die Pipelines von Entwicklungs- und Release-Branches beschränkt. SonarQube prüft den Code auf Bugs, Vulnerabilities, Security Hotspots, Code Smells, Testabdeckung und Code Duplikation. Generierte Dateien und Test-Code sind davon ausgeschlossen.

Für beide Repositories gibt es *Quality Gates*, welche bei jeder Pipeline-Ausführung erfüllt werden müssen. Unschönheiten wie Code Smells sind davon nicht betroffen, werden aber stets mit Refactorings behoben. Dazu gehört auch die Komplexität aller Funktionen welche von SonarQube gemessen werden. Wird der akzeptierte Schwellenwert überschritten, wird sie reduziert oder ausdrücklich Erlaubt (z.B. bei dem *Reducer* im Frontend).

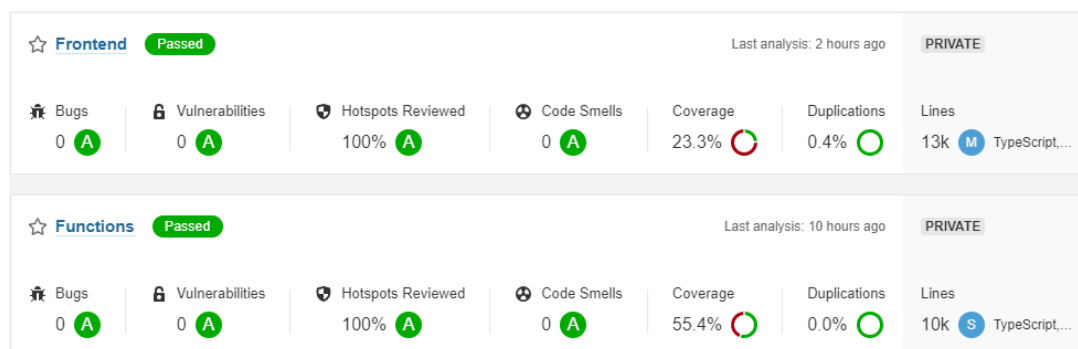


Abbildung 33: SonarQube Dashboard

5.5.2. Code Style Guide

Damit der Code einheitlich formatiert, übersichtlich und sauber ist, wird ein Code Style Guide angewendet. Um den manuellen Aufwand minimieren zu können, werden in allen Entwicklungsumgebungen zwei Erweiterungen verwendet. Die Regeln werden in den Config-Dateien `<.eslintrc.json>` und `<.prettierrc.json>` festgelegt.

Prettier

Mit dieser Erweiterung wird die einheitliche Formatierung garantiert. Die definierten Regeln werden bei jeder Speicherung einer Datei automatisch umgesetzt.

ESLint

Dank der statischen Quellcode Analyse von ESLint werden problematische Stellen und Muster im TypeScript Code identifiziert.

5.5.3. Frontend

Für das Frontend wird neben SonarQube eine weitere Analyse genutzt, um die Qualitätsanforderungen zu garantieren. Mit dem Google Lighthouse Audit werden Aspekte wie Performance, Accessibility, Best Practices, *Search Engine Optimization (SEO)* sowie Anforderungen an eine *PWA* ausgewertet. Aus der Summe der Resultate in jedem Bereich wird ein Score berechnet.

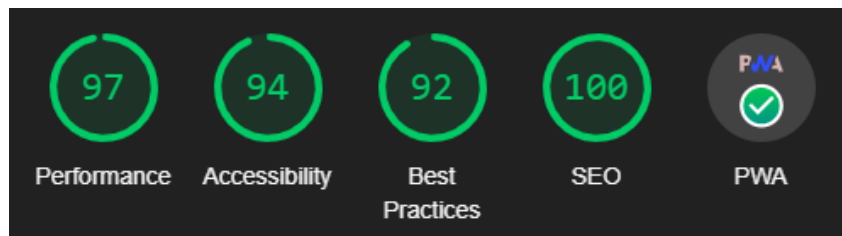


Abbildung 34: Google Lighthouse Audit Resultat

6. Evaluierung

6.1. Pilot Test

6.1.1. Konzept

Der Pilot Test soll noch in der laufenden Saison durchgeführt werden und mit 50-100 Personen getestet werden. Ob der Test effektiv durchgeführt wird und dieser auch den Anforderungen der Lakers entspricht wird vorgängig mit Michael Bart besprochen.

Absicht

Ziel des Pilot Tests ist die Prüfung der Prozesse zwischen Regie und Besucher. Der Administator soll Fragestellungen in der App aufschalten und der Besucher Tipps dazu abgeben. Die Tipps werden nach dem jeweiligem Ereignis ausgewertet und dem Benutzer in der App dargestellt. Ein essenzieller Punkt dieses Tests ist auch das Erlebnis des Besuchers. Hierfür werden die Teilnehmer nach dem Spiel aufgefordert, ein Feedback zu geben.

6.1.2. Durchführung

Der Pilot Test wurde abgesagt. Dafür gibt es mehrere Gründe. Einerseits sind die SCRJ Lakers ausgeschieden, noch bevor die Applikation eine Pilotreife erlangt hat. Andererseits haben im Vorfeld mehrere Diskussionen einen tiefen Aufwand/Nutzen-Faktor ergeben. Der Aufwand für einen Pilot Test ist immens. Die ganze Vor- und Nachbereitung, sei es technisch oder organisatorisch, um ein Pilot Test live im Stadion durchzuführen, beansprucht sehr viel Zeit. Zudem ist der Ausgang des Tests sehr ungewiss und es ist fraglich, ob er die gewünschten Ergebnisse liefern kann. Ein weiterer Absagegrund ist das ausgeprägte Risiko eines Imageverlustes oder tiefer Retention rate infolge eines Misserfolgs beziehungsweise Enttäuschung der App im Test.

Es wurden aber andere Testformen gefunden, womit eine höhere Ergebnisdichte zu niedrigeren Konditionen erreicht werden kann.

6.2. Crowd Test

Der Crowd Test verfolgt mehrere Ziele. Es werden Ergebnisse für

- Usability
- Fan Engagement
- Last- und Performance

eingeholt. Diese Aspekte unterscheiden sich stark im Konzept und der Auswertung. Deshalb sind diese Bereiche in den jeweiligen Testformen eingegliedert.

6.2.1. Durchführung

Der Crowd Test wird in einer Vorlesung mit Studierenden durchgeführt. Es stehen maximal 30 Minuten zur Verfügung. Durch den Test und insbesondere das Tippspiel wird mithilfe einer vorbereiteten Präsentation geführt. Diese befindet sich im Anhang (Kapitel ??). Es werden nur Informationen zum richtigen Verständnis der Präsentationsinhalte abgegeben. Zur Nutzung der App wird keine Auskunft gegeben.

Parameter	Information
Zeitpunkt	Donnerstag, 12.05.2022 1020-1050
Testpublikum	Vorlesung Objektorientierte Programmierung 2 (OO2)
Zielumgebung/-system (Systemstand)	AWS Cloud Prod Environment

Tabelle 62: Konfiguration Crowd Test

6.3. Usability Test

Mit diesen Usability Tests soll die Gebrauchstauglichkeit des Produktes überprüft und damit nachhaltig die User Experience und Zufriedenheit gesteigert werden. Potentielle Designproblematiken sollen frühzeitig erkannt und behoben werden.

Mit dem ersten Usability Test in diesem Projekt werden drei Ziele verfolgt.

Fokus	Erklärung
Navigation	Zukünftige User sollen ihre Ziele möglichst effizient und zufriedenstellend erreichen.
Präsentation	Das Design soll der Zielgruppe vertraut sein und dessen Interesse/Neugierde wecken.
Bedienung	Die Interaktion soll möglichst einfach und intuitiv sein.

Tabelle 63: Usability Testziele

6.3.1. Konzept

Mit dem erarbeiteten Prototyp sollen folgende Hypothesen und Fragestellungen überprüft werden.

Sind die einzelnen Funktionen gut auffindbar?

- Wissen die Testpersonen, wie sie auf den Startbildschirm zurückgelangen?
- Wissen die Testpersonen, wie sie zu ihrem Profil gelangen?
- Ist die Hauptfunktionalität Tippen genügend prominent platziert?
- Wie lange braucht eine Testperson (Anzahl Klicks, Zeit), um einen Tipp abzugeben?

Wissen die Testpersonen immer, wo sie sich befinden?

Sind Prozesse verständlich und funktionieren diese?

- Können die Testpersonen eine Belohnung/einen Gutschein einlösen?
- Können sich die Testpersonen einloggen?
- Wissen die Testpersonen, ob sie eingeloggt sind?
- Wissen die Testpersonen, ob sie gewonnen haben?

Sind die Texte und Begriffe verständlich?

- Können die Testpersonen offene von geschlossenen und ausgewerteten Challenges unterscheiden?
- Können die Testpersonen getippte von ungetippten Challenges unterscheiden? Werden die visuellen Elemente richtig interpretiert?

Testszenarios

Szenario	Vorgeschichte
Neuen Account erstellen	Du hast die App bereits ausprobiert und möchtest nun in den Genuss aller Funktionalitäten kommen. Dafür ist ein Account erforderlich. Diesen möchtest du nun erstellen. Du bevorzugst die Verknüpfung mit deinem Google Account, damit du nicht zusätzliche Anmeldeinformationen merken musst.
Eigene Platzierung in Rangliste finden	Du hast bereits einige Challenges in der App gespielt und Punkte gesammelt. Nun möchtest du deinen Fortschritt in der öffentlichen Rangliste einsehen. Welche Platzierung hast du?

Einen neuen Tipp abgeben	Es wurde eine neue Challenge veröffentlicht. Du möchtest diese neue Challenge lösen, indem du in der Auswahl Melvin Nyffeler als Lösung auswählst.
Einen bestehenden Tipp anpassen	Vor ein paar Minuten hast du die Challenge ‹Wer macht am meisten Sprints?› beantwortet. Nach einer längeren Diskussion mit einem Kollegen hast du deine Antwort überdacht und möchtest nun auf Roman Cervenka als Antwort setzen. Dafür passt du die Auswahl in der App entsprechend an.
Ausgewerteter Tipp anschauen	Eine Challenge ist bereits abgelaufen und auch bereits bewertet. Natürlich möchtest du deine Performance anschauen. Wie richtig hast du getippt?
Gewonnenen Gutschein abholen	Ju di hui! Du hast gewonnen! Aber wie geht es jetzt weiter? Wie und wo kommst du zu deinem Preis?
Fremde Platzierung nachschauen	Das Spiel ist vorbei. Nun möchtest du deine Platzierung mit jener eines Kollegen vergleichen. Sein Benutzername ist ‹zindable›. Findest du ihn in der öffentlichen Rangliste?

Tabelle 64: Usability Testszenarios

Vorinterview

Mit dem Vorinterview soll sich die Testperson in der Testumgebung anklimateisieren können und folgende Informationen preisgeben. Einfachheitshalber sind alle Beispielfragen in der du-Form geschrieben.

Information	Beispielfragen
Technisches Vorwissen	Wie viele Stunden pro Woche nutzt du das Smartphone? Würdest du dich als Technical-Native bezeichnen?
Demografische Daten	Wie alt bist du? Was bist du von Beruf?
Bezug zum Testobjekt	Wie regelmässig und auf welchen Plattformen verfolgst du das Schweizer Eishockey? Wie häufig besuchst du ein Heimspiel der SCRJ Lakers in einer Saison? In welcher Konstellation besuchst du ein Spiel? Nutzt du die SCRJ Lakers App?

Tabelle 65: Usability Test Vorinterview

Nachinterview

Mit dem Nachinterview sollen die Erfahrungen während der Tests reflektiert werden. Einfachheitshalber sind alle Beispielfragen in der du-Form geschrieben.

Information	Beispielfragen
Verständnis	Sind während den Tests Fragen oder Unklarheiten entstanden?
Probleme	Bist du irgendwo angestanden und wusstest nicht mehr weiter?
Besondere Aspekte	Was hat dir gut gefallen? Musstest du bei einer Aufgabe besonders lange überlegen? In welchem Moment warst du besonders unsicher? Was war nicht so gut?

Tabelle 66: Organisationsstruktur

Testform und -inhalt

Für die Usability-Tests wird ein UX-Prototyp in *Figma* erstellt, womit die Testszenarios interaktiv durchgespielt und protokolliert werden können. Die Teilnehmer absolvierten den Usability Test an einem Computer mit dem geöffneten Prototyp auf der Startseite der Anwendung.

Testpersonen

Für die Tests werden insgesamt vier bis acht Personen der primären Zielgruppen ausgewählt, damit die Vielfalt der Nutzer möglichst repräsentativ abgedeckt wird. Aus diesem Grund wurden Personen mit unterschiedlichem Alter und technischen Kenntnissen ausgewählt, die eine grundlegende aber unterschiedliche Kenntnis von Eishockey haben. Jede Testperson bekommt die gleichen Testszenarios vorgesetzt.

Zielgruppe	Anforderungen
Kinder/Student	Technisch affin, neugierig, jung
Mutter/Vater	Technisch vertraut, Routinier
Senior	Technisch inkohärent, alt

Tabelle 67: Usability Testpersonen

6.3.2. Durchführung

Testperson

Vor dem Start werden die Testpersonen über die Durchführung orientiert und zum lauten Denken ermuntert. Ihnen soll bewusst sein, dass sie nichts Falsches machen oder sagen können - ihre Meinung zählt. Anschliessend wird ihnen eine Liste von Aufgaben ausgehändigt, die sie selbständig und ohne zusätzliche Instruktionen erfüllen sollen. Ihr Verhalten mit der Anwendung wird von einem Beobachter verfolgt und die daraus entstandenen Erkenntnisse notiert. Letztendlich wird das abschliessende Feedback der Teilnehmer bezüglich ihrer gemachten Erfahrungen mit der Anwendung in einem Nachinterview eingeholt.

Diese Tests werden immer mit neuen, unvoreingenommenen Probanden durchgeführt, damit sie sich mit vorherigen Durchführungen besser vergleichen lassen.

Testleiter

Für den Testleiter gibt es einige Verhaltensregeln zu beachten

- Nichts sagen/beantworten, nötigenfalls ausweichend zurückfragen mit W-Fragen
- Testleiter hilft der Testperson bei der Bewältigung der Aufgaben in keinsten Weise
- Keine Wörter der präsentierten Navigation nennen
- Ausgiebig beobachten

6.3.3. Auswertung Test 1

Der erste Usability Test hat einige Schwachstellen im visuellen Konzept offenbart.

Label Resultate

Die Beschriftung «Resultate» ist mehrdeutig und widersprüchlich. So können unter diesem Ausdruck auch sportliche Resultate der SCRJ Lakers erwartet werden oder Statistiken zu den eigenen Tippspielteilnahmen. In Wahrheit findet man nur eine Rangliste vom Tippspiel. Daher ist eine Beschriftung «Rangliste» aussagekräftiger.



Abbildung 35: Navigationsleiste vorher



Abbildung 36: Navigationsleiste nachher

Preisübersicht

Im Bereich Preise gibt es eine Übersicht der eigenen Preise. Aber auch diese Seite kann unterschiedlich interpretiert werden. Aus einem anderen Blickwinkel zeigt diese Seite nämlich die gewinnbaren Preise an - nicht die persönlichen. Wo wir auch bereits bei einer weiteren Erkenntnis sind. Dem Benutzer werden aktuell nirgends die möglichen Belohnungen angezeigt. Er macht somit bei einem Gewinnspiel mit ohne die Preise zu kennen.

Die bestehende Seite für die persönlich gewonnenen Preise wird verdeutlicht, indem von ‹Meine Preise› gesprochen wird. Für alle gewinnbaren Preise eines Tippspiels wird eine zusätzliche Seite geschaffen. Auf diese wird vom Menüpunkt ‹Preise› und ‹Tippspiel› verwiesen.

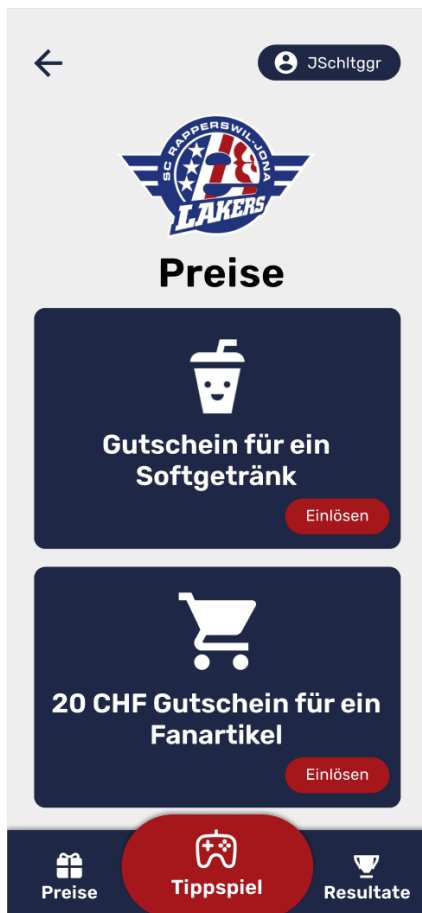


Abbildung 37: Preisübersicht vorher

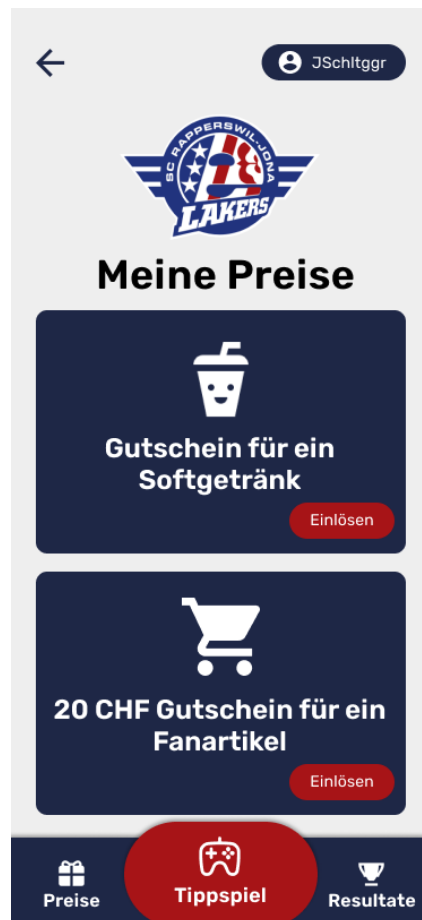


Abbildung 38: Preisübersicht nachher

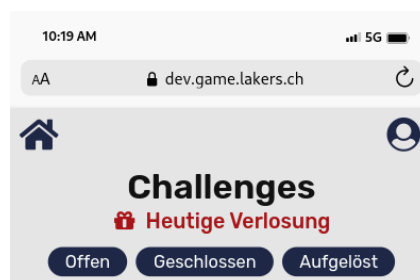


Abbildung 39: Ergänzende Verknüpfung für neue Preis-Seite

Preis einlösen

Für die Probanden war es unklar, was das Wischen genau für Konsequenzen nach sich zieht. Nicht selten wurde daher einfach gewischt ohne dessen Bedeutung zu hinterfragen. An dieser Stelle muss der Ablauf und insbesondere die Wischgeste genauer erklärt werden. Es muss visuell verhindert werden, dass fälschlicherweise zu früh, noch vor der eigentlichen Gutscheinübergabe, gewischt wird.

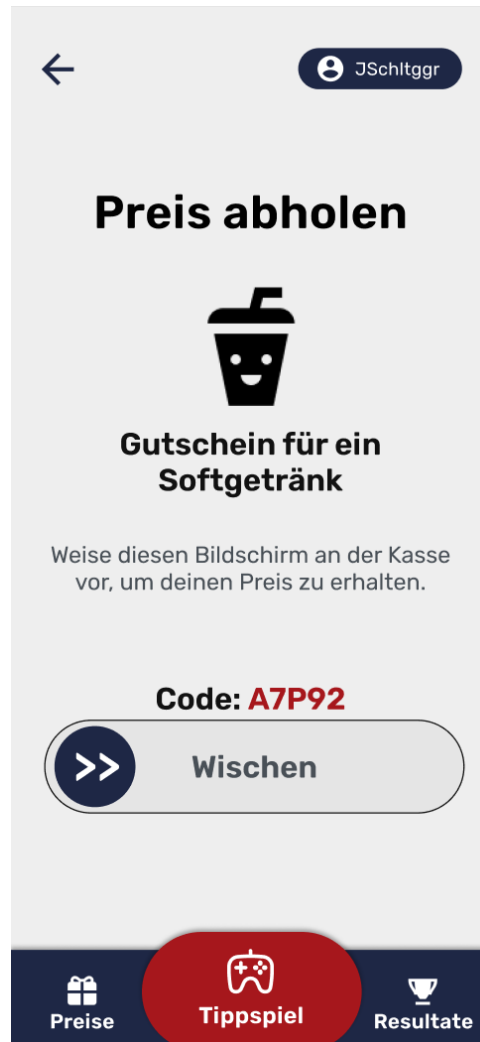


Abbildung 40: Preis einlösen

Bestätigung der Tippauswahl

Auf der Antwortauswahl-Seite konnte man auch unterschiedliches Verhalten beobachten. Einige haben instinktiv nach Auswahl eines Spielers auf die Schaltfläche Auswahl ändern geklickt, in der Annahme damit die Auswahl definitiv zu speichern. Andere wussten schlicht nicht, ob die Eingabe nun gespeichert wurde. Weshalb der Pfeil zurück nur unter grosser Zurückhaltung betätigt wurde. Diese Seite hat für grösste Verunsicherung gesorgt. Dem Benutzer war unklar, was bereits passiert ist und was er als nächstes tun soll. Für den nächsten Test sollen verschiedene alternative Varianten erarbeitet und verglichen werden. Beispielsweise soll der Button neu die Auswahl speichern und folglich direkt auf die Challengeübersicht zurückkehren. Idealerweise liesse sich diese Varianten mit A-/B-Tests validieren. Aufgrund der vorhandenen Kapazitäten werden die Tests aber in einer vereinfachten Form durchgeführt.

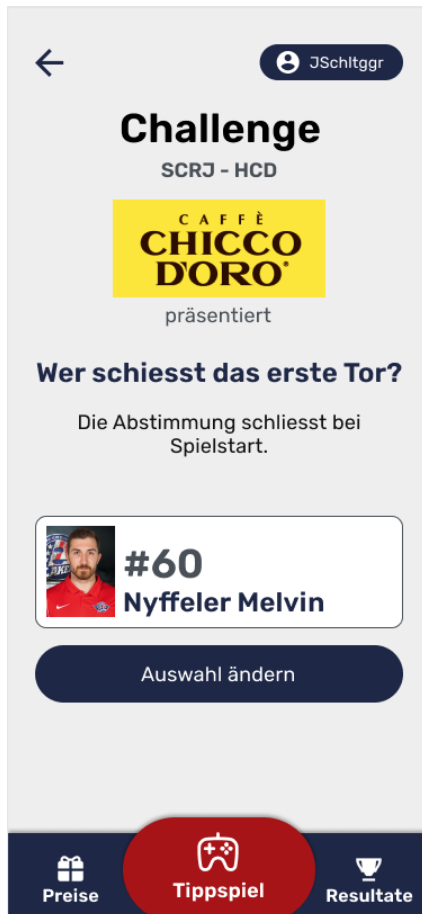


Abbildung 41: Antwortauswahl vorher



Abbildung 42: Antwortauswahl angepasste Variante

Challenge-Status

Für sehr viel Diskussion sorgte auch die Challenge-Übersicht. Einerseits gaben die Beschriftungen der einzelnen Status zu reden. So entsprachen Offen, Geschlossen und Aufgelöst nicht immer der gleichen Vorstellung. Für die einen bedeutete Geschlossen, dass der Tipp bereits abgegeben wurde. Auf der anderen Seite war es manchmal schwierig, bestehende Tipps von neuen, ungetippten Challenges zu unterscheiden. Dies ist möglicherweise auch darauf zurückzuführen, dass der Prototyp einen vorgeschrittenen Spielstand zeigt und nicht bei Null beginnt. An dieser Stelle werden nun ebenfalls verschiedene Varianten erarbeitet und in einem weiteren Test validiert.

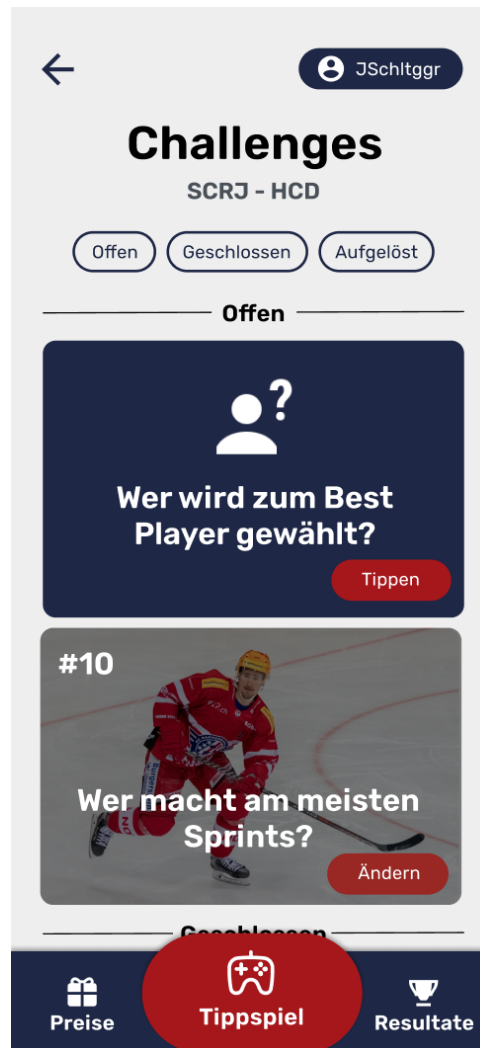


Abbildung 43: Challengeübersicht vorher

Auf Grundlage des ersten Usability Tests wurden erkannte Schwachstellen mit verschiedenen Lösungsansätzen reduziert und in einem weiteren Usability Test validiert. Dieser Test bedient sich der Inhalte aus dem ersten Test und ergänzt diese mit den neun Erkenntnissen aus dem ersten Test.

6.3.4. Auswertung MVP-Test mit Michael Bart

Im Vergleich zur ersten Durchführung wird als Testtool nicht mehr *Figma*, sondern die lauffähige Applikation verwendet.

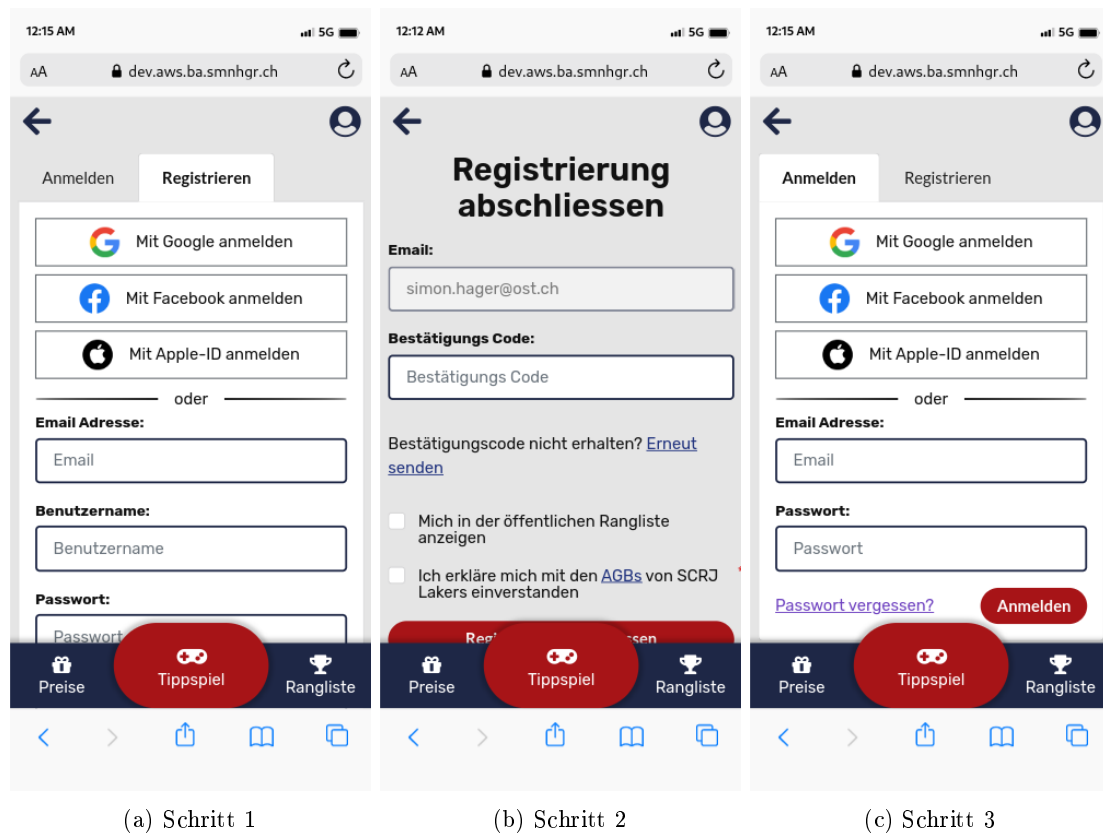


Abbildung 44: Registrierung vorher

Durch die Beobachtung der App-Nutzung von Michael Bart ist der gesamte Authentifizierungsprozess aus Benutzersicht, insbesondere die Anmeldung mit einer E-Mail, sehr negativ aufgefallen. Sie benötigt sehr viele Klicks und ist wenig intuitiv. Alles beginnt mit der richtigen Auswahl des Tabs. Anmelden oder Registrieren. Sie passen generell nicht in das restliche Bedienkonzept und wurden auch in diesem Praxistest eher schlecht als recht ausgewählt. Zudem muss auf kleineren Bildschirmgrößen oft gescrollt werden, um sämtliche Felder auszufüllen oder einen Button zu erreichen. Ist das Registrierungsformular versendet, muss der Code im Verifikations-E-Mail gemerkt oder kopiert und wieder in der App eingegeben werden. Auch auf dieser Seite muss teilweise gescrollt und zusätzliche Angaben gemacht werden. Dies ist ebenfalls nicht ideal. Eine weitere Analyse hat gezeigt, dass sich die Registrierung von der Anmeldung nur bei der Methode mit E-Mail unterscheidet. Bei der Registrierung und Login mit Google, Facebook oder Apple-ID wird derselbe Prozess angestossen und bedarf deshalb keiner Unterscheidung.

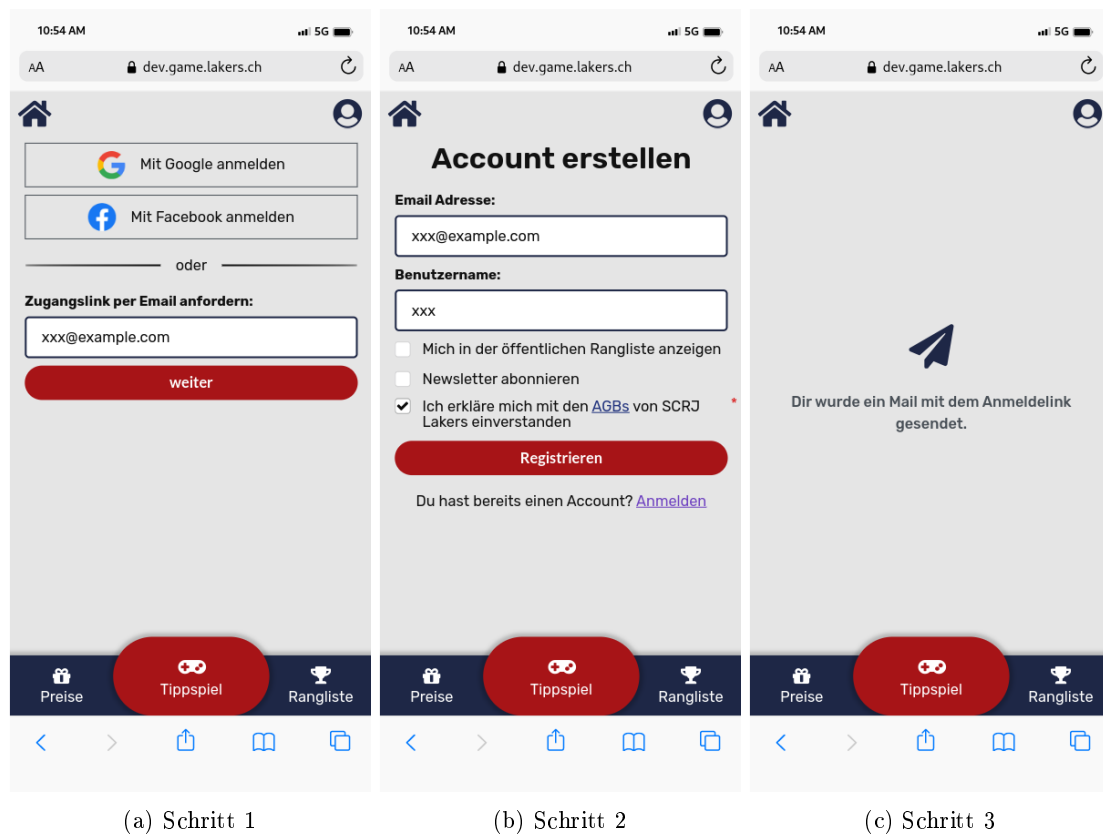


Abbildung 45: Registrierung nachher

Abhilfe schafft das passwordless Login. Dieses verzichtet auf die Angabe eines Passworts und da eine E-Mail für jedes Login notwendig ist, braucht es keine separate Verifikation mehr. Die beiden Tabs werden durch eine kombinierte Authentifizierungsmaske ersetzt. Aufgrund der Benutzereingaben wird das eine Formular für den Prozess Login oder Registrierung dynamisch angepasst.

6.3.5. Auswertung Crowd Test

Dieser Test wird mit mehreren Probanden gleichzeitig durchgeführt, um weitere Testresultate zu Fan Engagement und Performance zu erhalten. Entsprechend ist der Testaufbau leicht angepasst. Im Vergleich zum ersten Usability Test haben gerade einmal einer von 33 Testpersonen einen Bezug zum Eishockey. Getestet wird in verschiedenen Smartphone-Browsern und Gerätegrößen. Die gemachten Feststellungen sind aber allgemein und beziehen sich nicht auf bestimmte Browser und Bildschirmgrößen.

Browser

- Brave (3)
- Mozilla Firefox (4)
- Google Chrome (15)
- Safari (9)
- Samsung Browser (1)
- GNU IceCat (1)

Bildschirmgrößen

- 6.8 Zoll (1)
- 6.6 Zoll (1)
- 6.5 Zoll (4)
- 6.4 Zoll (7)
- 6.3 Zoll (1)
- 6.1 Zoll (10)
- 6 Zoll (1)
- 5.8 Zoll (4)
- 5.4 Zoll (1)
- 5.2 Zoll (1)
- unbekannt (2)

Sortierung

Kritisiert werden überraschend ausgeprägt die Sortierung verschiedener Auflistungen. Der Fehler in der Ranglistensortierung ist technischer Natur und bereits bereinigt. Die Sortierung in der Challenge-Übersicht innerhalb eines Status wird den Anregungen entsprechend angepasst. Die neuesten Challenges erscheinen zuoberst, ältere weiter unten.

Über die Sortierung der Spielerauswahl in der Tippabgabe wird gestritten. Soll sie konsequent nach Trikotnummer oder nach Position und anschliessend nach Nummer oder Name sortiert werden? Fortan werden die Spieler erst nach Position, dann nach Nummer sortiert. Manche Probanden fordern eine Suchfunktion für die Spielerauswahl, was definitiv umgesetzt werden soll. Diese Funktionalität wird aber nicht mehr Bestandteil dieses Projektes sein.

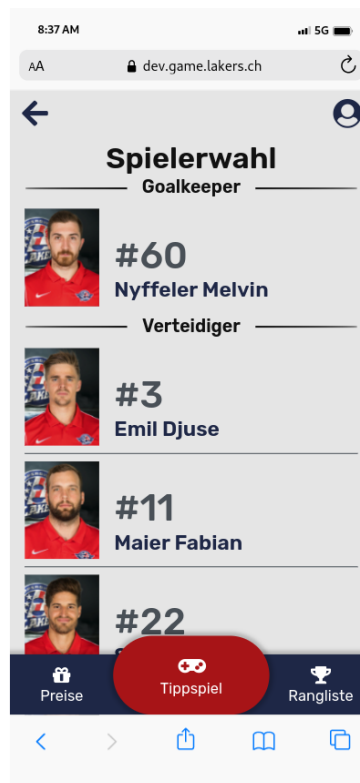


Abbildung 46: Sortierung Spielerauswahl

Informationen

Für einige sind die Informationen zum Tippspiel noch zu spärlich. Ohne Einführung eines Instructors ist der Sinn und Zweck dieser App nicht gegeben. Dass die App installierbar ist, erfahren die Benutzer auch nur über die einmalige Aufforderung des Browsers. Gewisse anleitenden Hinweise fehlen dem Benutzer noch, da nicht alle zweckdienlichen Pop-Ups und Push-Benachrichtigungen umgesetzt werden können. Beispielsweise, dass eine Registrierung oder Anmeldung für die Teilnahme an der Verlosung erforderlich ist, oder ein anonymer Benutzer seine Tipps nach 30 Tagen verliert. Solange diese nicht als interaktive Meldungen erscheinen, sollen die benötigten Informationen zumindest in Fliesstext vorliegen. Dafür wird auf der Startseite oben links ein Icon platziert, worüber auf ein FAQ navigiert werden kann. Auf dieser Seite können auch rechtliche und organisatorische Informationen diskret verlinkt werden.



Abbildung 47: Menüpunkt für Informationen

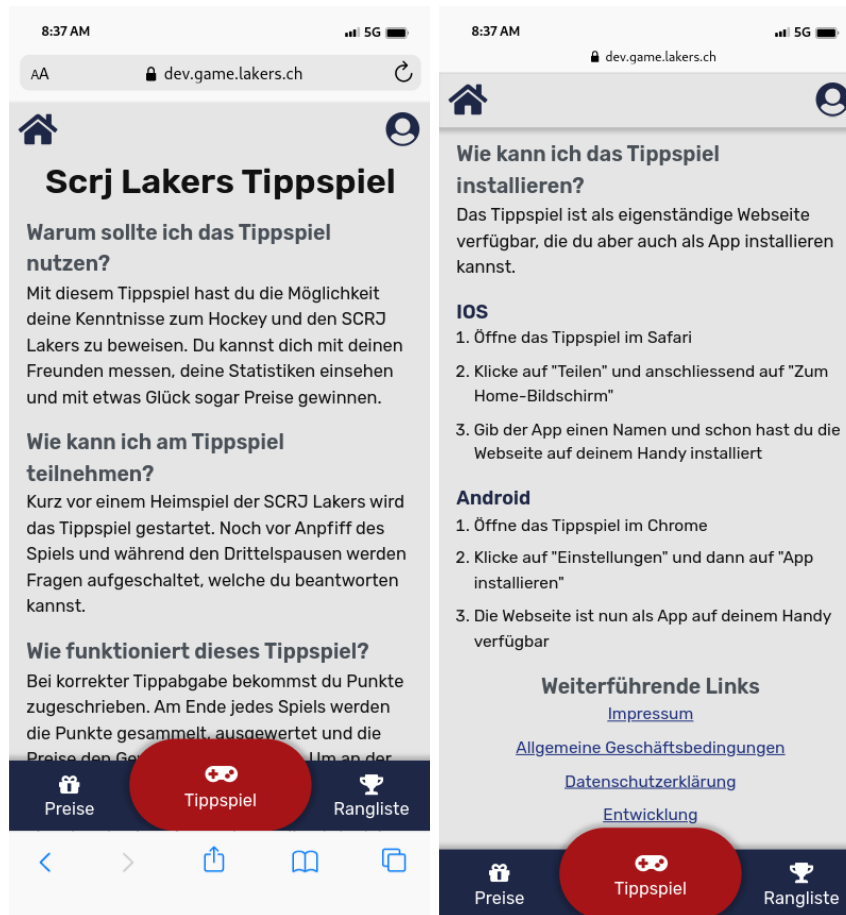


Abbildung 48: Informationsbereich

Bestätigung der Tippauswahl

Im letzten Test fehlte den Probanden noch die Bestätigung für ihre Tippabgabe. Deshalb wurde auf der Tippauswahl-Seite ein Button <Auswahl speichern> ergänzt. Mit dem Wissen, dass auch diese Lösung mit dem aktionslosen Button nicht perfekt ist und weiterer Optimierung bedarf. Nun beanstanden viele Probanden, dass die Tippabgabe zu viele Klicks benötigt, der Button <Auswahl speichern> doch unnötig sei und sich deshalb unerwartet verhält.

Der Speicherprozess wird nun im Loader, welcher bei der Auswahl eines Spielers zur Tipp-Übersicht zurückführt, mit einem entsprechenden Text verdeutlicht. Damit soll ein besseres Verständnis für die automatische Speicherung geschaffen werden, damit es den <Auswahl speichern> Button für niemanden mehr braucht.

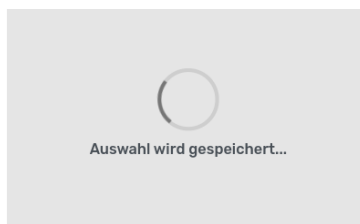


Abbildung 49: Loader nach der Auswahl eines Spielers

Diese Aspekte sollen in zukünftigen Tests besonders beachtet beziehungsweise wieder aufgegriffen werden.

- Sortierung Spielerauswahl
- Auswahlbestätigung
- Auffinden von Informationen über die App
- Installation der PWA

6.3.6. Auswertung SCRJ meets OST Besprechung

Die 7 Probanden haben einen kleinen Auszug der Challenges im Crowd Test vorgesetzt bekommen. Sie fanden sich gut zurecht, konnten die Aufgaben mit Bravour erledigen und das abschliessende Feedback war ebenfalls sehr positiv. Aus diesem Test resultierten nicht sehr viele neue Erkenntnisse. Lediglich ein eher seltener Bug konnte erkannt werden. Die Sprache der App war auf Englisch eingestellt. Mit eingestelltem Browser-Übersetzer werden dadurch einzelne Spielernamen zu neuen Wörtern formiert. Die Sprache ist nun auf <de-CH> eingestellt. Die Übersetzer sollten jetzt nicht mehr eingreifen.

6.4. Last- und Performanctests

Die Applikation wird über eine beschränkte Zeit besonders stark beansprucht werden, während es den Grossteil der Laufzeit auf Standby läuft. Schätzungsweise nehmen bis zu 2000 Personen an den Challenges teil. Deshalb muss mit vielen gleichzeitigen Anfragen gerechnet werden. Dieses Szenario wurde mit verschiedenen Ansätzen systematisch getestet.

6.4.1. Api Tests

Mit dieser Testsuite lassen sich sehr effizient und ressourcenschonend verschiedenste Spielszenarien auf dem Backend simulieren.

Konzept

Dafür wird ein typischer Spielablauf mit den grundlegenden Anfragen nachgebaut. Der Administrator richtet ein Spiel ein. Die Tipper geben ihre Tipps für offene Challenges ab und am Ende beendet der Administrator das Spiel. Der Ablauf im Detail:

1. Administrator meldet sich an
2. Administrator erstellt das Game und startet es
3. Administrator erstellt die Challenges und startet diese
4. Administrator erstellt die Rewards
5. Alle Teilnehmer erstellen anonymen Benutzer
6. Alle Teilnehmer melden sich an
7. Alle Teilnehmer setzen für sämtliche Challenges eine Antwort
8. Administrator beendet und wertet sämtliche Challenges aus
9. Administrator beendet das Game
10. Administrator wertet das Game aus

Das Setup erlaubt die beliebige Definierung der Anzahl Tipper sowie eine einfache Challenge- und Rewardkonfiguration für ein einzelnes Spiel. In einem Test wird jeweils eines der folgenden Szenarien konfiguriert.

Test	Challenges	Rewards	Teilnehmer	ChallengeTips
1	1	1	125	125
2	1	1	500	500
3	1	1	2000	2000
4	5	5 ¹	125	125
5	5	5 ¹	500	500
6	5	5 ¹	2000	2000

¹ Die Anzahl Rewards wird auf drei Rewards mit der Menge 1, 3, 1 aufgeteilt

Tabelle 68: Testszzenarien der Api Tests

Durchführung

Die Testumgebung soll immer möglichst gleich sein, um eine gute Vergleichbarkeit der Resultate zu gewährleisten. Dafür werden alle Tests vom gleichen Server gestartet.

Parameter	Information
Zeitpunkt	Montag, 03.05.2022 2300-2400
Testtool	k6
Testclient	Server sinv-56033.rj.ost.ch (Ubuntu)
Zielumgebung/-system (Systemstand)	AWS Cloud Dev Environment

Tabelle 69: Konfiguration der Api Tests

Wird ein Test gestartet, werden gezielte Anfragen an das Backend (AWS AppSync und AWS Cognito) gesendet. Anschliessend werden die Testresultate, insbesondere die Laufzeiten und Fehler, an eine Datenbank übermittelt.

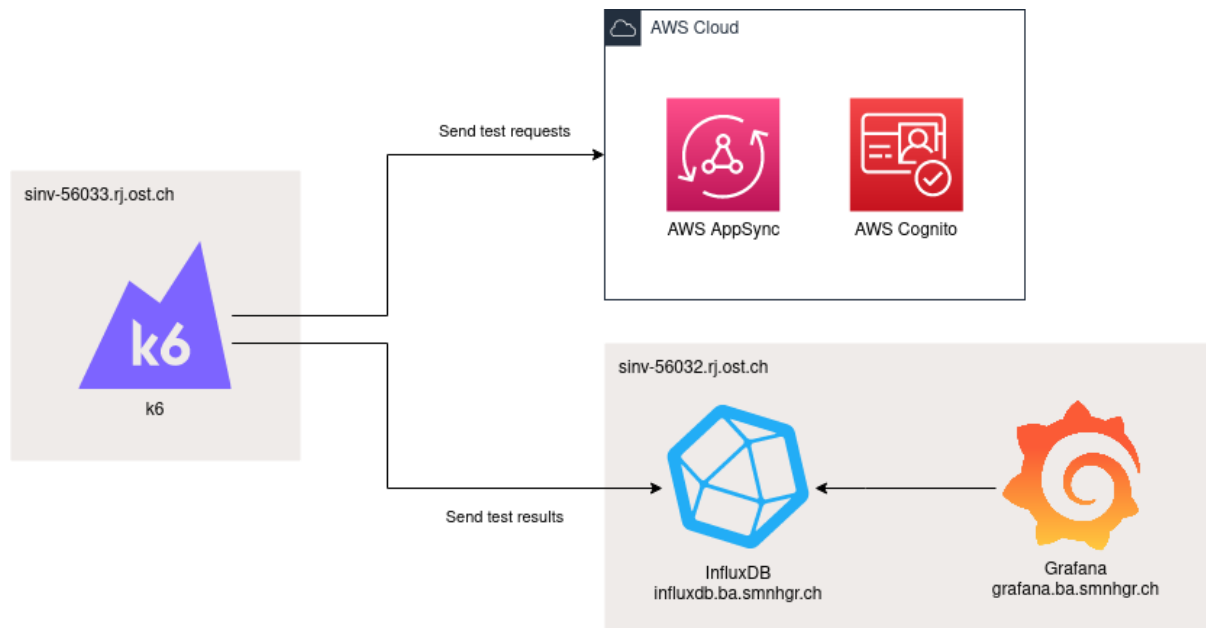


Abbildung 50: Architektur der Api Tests

Auswertung

Für die Auswertung des Tests wird vorwiegend Grafana, eine bekannte Open Source Analyse- und Überwachungslösung, eingesetzt. Darin werden die Ergebnisse in einem eingerichteten Dashboard visualisiert. Dafür nutzt sie die Testresultate aus der Datenbank.[38]

URLs and latencies (By Name)						
URL	count	min	median	max	mean	P95
createAnonymousUser	1970			410 ms		
createChallengeApplication	3		201 ms	206 ms	201 ms	206 ms
createGame	1	327 ms	327 ms	327 ms	327 ms	327 ms
createRewardApplication	3		201 ms	206 ms		206 ms
evaluateChallengeApplication	3	4 s	5 s	7 s	6 s	7 s
evaluateGame	1	7 s	7 s	7 s	7 s	7 s
loginUser	1971		241 ms	2 s	243 ms	279 ms
setChallengeTip	5910		204 ms	2 s	235 ms	352 ms

Abbildung 51: Grafana Dashboard von TestszENARIO 6

Untersucht werden die Latenzen unterschiedlicher Anfragen in Bezug auf ihr TestszENARIO.

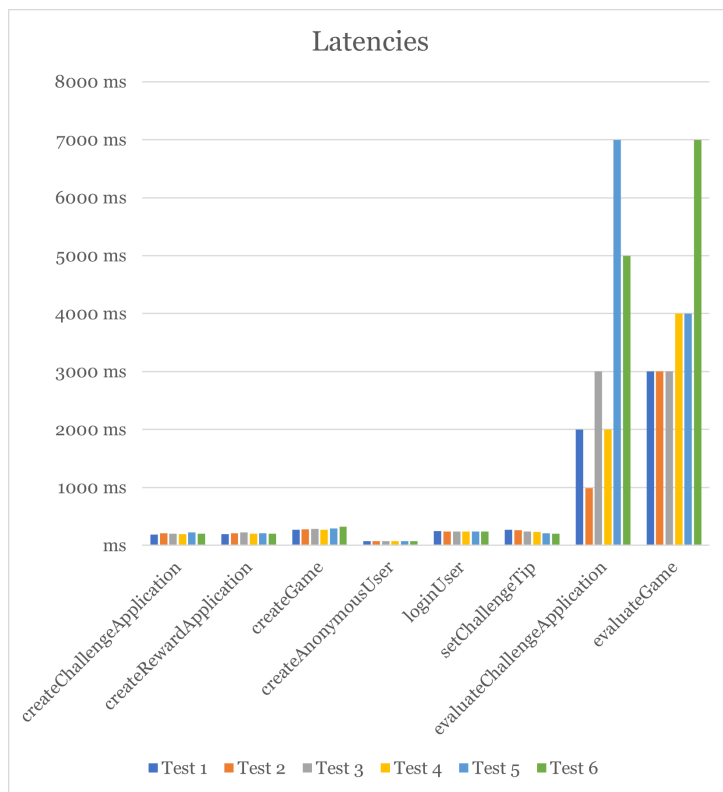


Abbildung 52: Latenzen von Testszenario 6

Während die Zugriffszeiten bei den meisten Anfragen relativ stabil tief bleiben, schwingen die Evaluierungsprozesse `<evaluateChallengeApplication>` und `<evaluateGame>` oben aus. Mit bis zu 7 Sekunden Laufzeit bei 2000 Challenge Tipps in der Challenge-Auswertung beziehungsweise 6000 Challenge Tipps bei der Spiel-Auswertung fallen diese beiden Anfragen ziemlich negativ ins Gewicht. Grund dafür sind die vielen parallelen und teils seriellen Datenbank-Anfragen je Challenge Tipp. Das führt unweigerlich zu einer linear steigenden Auslastung bei grösserem Tipp-Vorkommen. Als Massnahme wird jenen Funktionen mit hohem parallelem Anfragebedarf zusätzlicher RAM zugewiesen. Der Effekt lässt sich sehen. Die Latenzen sinken dadurch um ungefähr 50%.

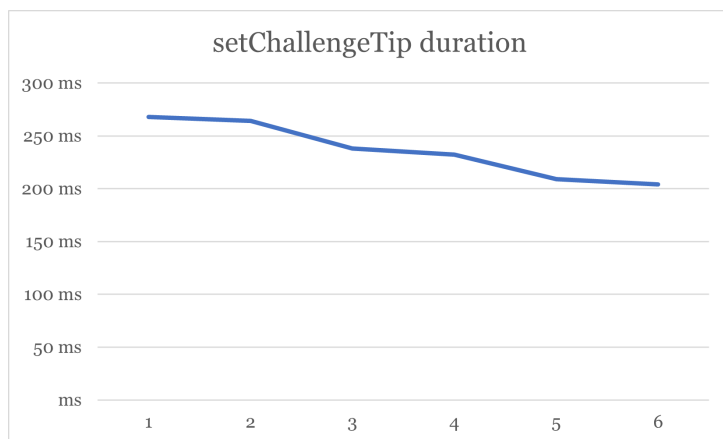


Abbildung 53: Zeiten von setChallengeTip

Auffallend ist die kontinuierlich sinkende Dauer für den Aufruf `<setChallengeTip>`. Anfänglich sind sämtliche Anfragen sogenannte Cold starts. Nach den ersten Tippabgaben kann die aufgebaute Infrastruktur für weitere Tipps wiederverwendet werden. Damit fällt ein merklicher Anteil der Latenz weg. Dieses

Verhalten kann man sehr gut mit dem zweiten Bild mit den spitzen Picks am Anfang und dem sehr konstanten späteren Verlauf nachvollziehen.[39]

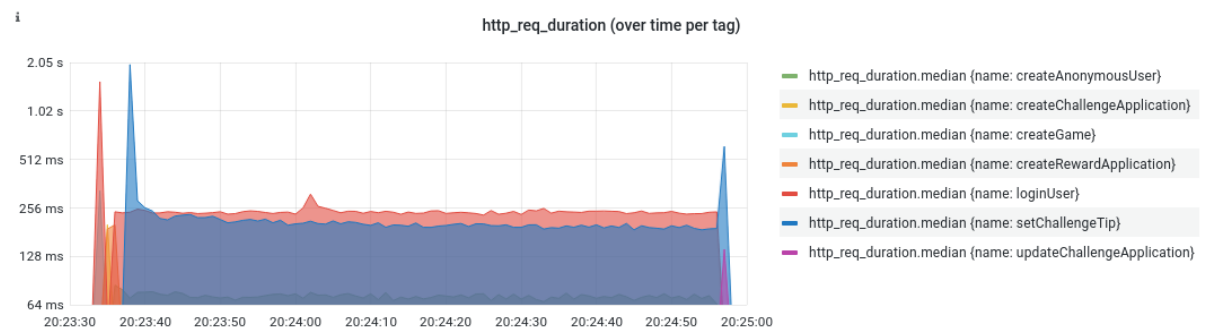


Abbildung 54: Grafana Dashboard von TestszENARIO 6 mit sichtbaren Cold-Starts

Cold starts können mit Provisioned Concurrency reduziert werden. Konkret wird die Funktion damit initialisiert und warm gehalten. Sodass diese direkt auf Anfragen reagieren kann.

6.4.2. End-to-end Tests

Es liegt in der Natur der Sache, dass sich das Frontend merklich anders verhält und das Backend auch anders nutzt als die künstlichen Api Tests. Es fehlen sekundäre Abfragen an das Backend, die für einzelne Ansichten zwar notwendig, für den funktionalen Erfolg jedoch nicht zwingend sind. Desweiteren werden weitere Ressourcennutzungen, wie der Download von Bildern in den bisherigen Tests nicht berücksichtigt. Deshalb wurde ein realitätsnaher End-to-end Test aufgebaut, womit Benutzereingaben im Browser systematisch nachgestellt werden können.

Konzept

Für diese Testart wurde ein Benutzerverhalten für ein Game nachgebaut.

Phase 1

1. Teilnehmer öffnet die Startseite
2. Teilnehmer klickt im Menü auf Tippspiel
3. Teilnehmer klickt auf erste offene Challenge
4. Teilnehmer wählt ersten Spieler aus
5. Teilnehmer kehrt auf Tippspielübersicht zurück
6. Teilnehmer klickt erneut auf erste offene Challenge

Phase 2

1. Teilnehmer öffnet die Startseite
2. Teilnehmer klickt im Menü auf Tippspiel
3. Teilnehmer klickt auf erste geschlossene Challenge

Phase 3

1. Teilnehmer öffnet die Startseite
2. Teilnehmer klickt im Menü auf Tippspiel
3. Teilnehmer klickt auf erste ausgewertete Challenge

Phase 4

1. Teilnehmer öffnet die Startseite
2. Teilnehmer klickt im Menü auf Profil
3. Teilnehmer füllt das Formular mit einer registrierten E-Mail-Adresse aus

Phase 5

1. Teilnehmer öffnet die E-Mail
2. Teilnehmer klickt auf den Link in der E-Mail.
3. Teilnehmer öffnet das Profil

Phase 6

1. Teilnehmer öffnet die Startseite
2. Teilnehmer klickt im Menü auf Tippspiel
3. Teilnehmer klickt auf erste ausgewertete Challenge

Phase 7

1. Teilnehmer öffnet die Startseite
2. Teilnehmer klickt im Menü auf Rangliste

Durchführung

Bevor ein End-to-end Test durchgeführt werden kann, muss ein Game vorbereitet und gestartet werden. Zur Vorbereitung muss ein Sponsor bestimmt und mindestens 20 Spieler zum Aufgebot im Admin-Panel hinzugefügt werden. Während eines Tests müssen für bestimmte Phasen gewisse Vorbedingungen erfüllt werden. Diese entsprechen im produktivem Umfeld teilweise den Eingaben eines Administrators während einem Spiel. Diese können manuell in der Administratoransicht der Applikation vorgenommen werden.

Phase	Vorbedingungen
Phase 0	Ein Game erstellt und gestartet
Phase 1	Eine Challenge geöffnet
Phase 2	Eine Challenge abgeschlossen
Phase 3	Eine Challenge ausgewertet
Phase 4	Keine
Phase 5	Ein E-Mail mit einem Magic Link erhalten (automatisch)
Phase 6	Ein Teilnehmer als registrierter Benutzer eingeloggt (automatisch)
Phase 7	Ein Teilnehmer als registrierter Benutzer eingeloggt (automatisch)

Tabelle 70: Vorbedingungen einzelner Testphasen

Die Testumgebung soll immer möglichst gleich sein, um eine gute Vergleichbarkeit der Resultate zu gewährleisten. Dafür werden alle Tests vom gleichen Server gestartet.

Parameter	Information
Zeitpunkt	Montag, 09.05.2022 2300-2400
Testtool	cypress
Testclient	Server sinv-56033.rj.ost.ch (Ubuntu)
Zielumgebung/-system (Systemstand)	AWS Cloud Dev Environment
Anzahl Benutzer	50 (10 synchron)

Tabelle 71: Konfiguration End-to-end Tests

Ein erfolgreicher Test sieht folgendermassen aus.

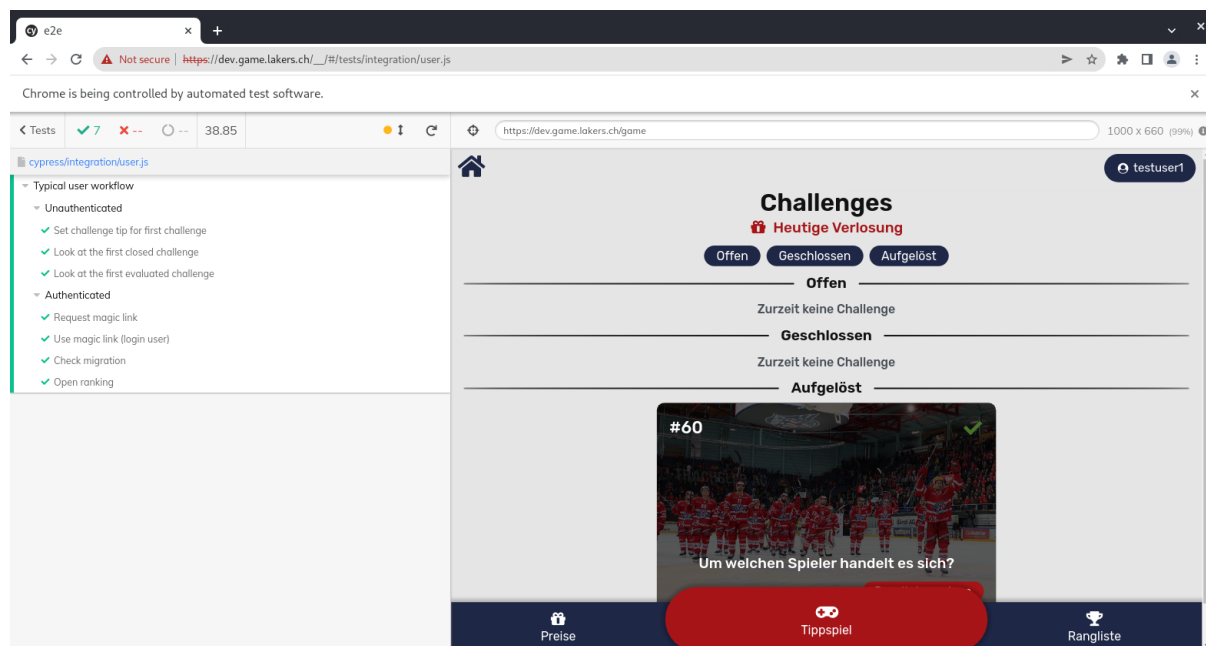


Abbildung 55: Erfolgreiche E2E Testdurchführung mit *Cypress*

Auswertung

Für konkrete Aussagen zu den effektiven und prognostizierten Kosten werden die Zahlen im AWS Billing herangezogen. Das AWS Billing bietet hierfür verschiedene Reports und Tools, womit der Service-Gebrauch und die resultierenden Kosten analysiert werden können. Mit diesen Zahlen wird eine genauere Hypothese aufgestellt, wie viel das System im Betrieb beanspruchen wird. Die Zahlen der 50 aktiven Benutzer aus dem Test werden auf 2000 aktive Nutzer hinaufgerechnet. Sämtliche verwendeten Dienste bieten ein Free Tier an. Bleibt die Nutzung innerhalb dieses Kontingents, entstehen keine Kosten für ein Spiel. Nachfolgend wurden die Free Tiers nicht berücksichtigt, um die maximal möglichen Kosten für ein einzelnes Spiel und die wiederkehrenden Kosten (alle exklusiv MwSt.) möglichst transparent und nachvollziehbar zu halten. Ein Spiel kostet gemäss der Analyse USD 2.25.

AWS Service	Kosten je Spiel	Kosten pro Monat
AWS Amplify	USD 0.92	USD 0.00
AWS AppSync	USD 0.51	USD 0.00
AWS CloudWatch	USD 0.00	USD 0.00
AWS Cognito	USD 0.00	USD 0.00
AWS DynamoDB	USD 0.29	USD 0.00
AWS Key Management Service (KMS)	USD 0.01	USD 0.00
AWS Lambda	USD 0.02	USD 0.00
AWS Route 53	USD 0.00	USD 0.50
Simple E-Mail Service (SES)	USD 0.14	USD 0.00
AWS Simple Storage Service (S3)	USD 0.36	USD 0.00
Total	USD 2.25	USD 0.50

Tabelle 72: Kosten gemäss End-to-end Tests

Stellt man die einzelnen Dienste in Relation zueinander und filtert die unbedeutenden Positionen heraus, ergibt das folgende Preiszusammensetzung für ein Spiel.

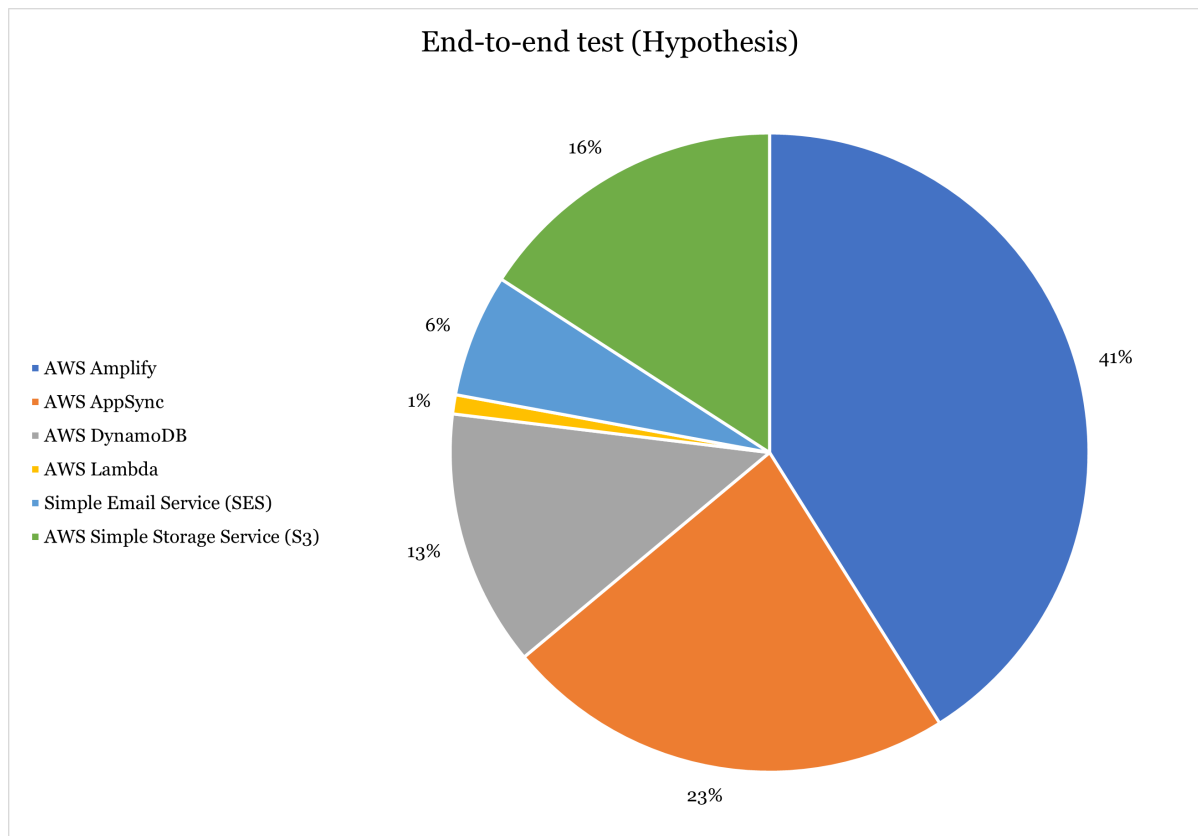


Abbildung 56: Preiszusammensetzung eines Spiels gemäss End-to-end tests

Der Free Tier von AWS Amplify läuft nach 12 Monaten ab und wird spätestens dann anteilmässig die meisten Kosten verursachen. Erwartbar sind die Kosten von AppSync, DynamoDB und S3. Sie bieten das Rückgrat für die meisten Funktionen der App und werden entsprechend auch gebraucht sowie abgerechnet. Die Lambdas sind dagegen erstaunlich günstig, machen diese doch nur 1% der Gesamtkosten aus. Zu berücksichtigen ist hierbei, dass noch kein Caching und Optimierung der Startzeit eines Lambdas vorgenommen wurde. Bekanntlich wirken sich solche Einstellungen signifikant auf diese Kostenposition aus. Bemerkenswert sind die Kosten für den Simple E-Mail Service. Mit gut einem E-Mail pro Benutzer bei der Anmeldung ist dieser Dienst bereits für 6% der Kosten verantwortlich.

6.4.3. Crowd Test

Der Crowd Test wird einmalig in einer Vorlesung mit 30+ Studierenden durchgeführt. Es ist ein abschliessender Test, welcher die natürliche Nutzung der Applikation in einem vorgegebenem Rahmen abbilden soll. Dieser Test dient auch der Validierung, wie aussagekräftig die anderen Tests sind. Dieser Test liefert Ergebnisse zur natürlichen App-Bedienung mehrerer gleichzeitig aktiver Nutzer. Dafür wird auch in diesem Test ein Spiel durchgespielt. Das Spielszenario soll in der Durchführung möglichst realitätsnah sein, um valide Daten zur Nutzung und Kosten zu erhalten.

Konzept

Alle erfolgreich implementierten Funktionen werden für das Tippenspiel des Crowd Tests aktiviert beziehungsweise vorbereitet.

- 1 Sponsor ist ausgewählt
- 20 Spieler im Aufgebot
- 5 verschiedene Spieler-Challenges
- 3 verschiedene Preise

Die Tester werden über den grundlegenden Sinn und Zweck der App informiert, sie sollen aber ziemlich frei in der App navigieren und ihrer Intuition folgen. Der Test wird in die typischen Phasen eines Eishockeyspiels gegliedert. Die Teilnehmer erhalten jeweils 20-30 Sekunden Zeit, ihre Antwort einzugeben.

Phase	Aufgabe
Auf Weg ins Stadion	Challenge ‹Wer ist Liga-Topscorer der 2021/22?› durchführen
Vor Anpfiff	Challenge ‹Wer ist es?› durchführen
Im ersten Drittel	Challenge ‹Wer schießt das erste Tor?› durchführen
Drittelpause	Zur Registrierung animieren
Im zweiten Drittel	Challenge ‹Wer schießt das nächste Tor?› durchführen
Drittelpause	Zur Registrierung animieren
Im dritten Drittel	Challenge ‹Wer bekommt die erste Strafe?› durchführen
Schlusspfiff	Spiel auswerten

Tabelle 73: Kosten gemäss End-to-end Tests

Auswertung

Insgesamt haben 33 Personen am Crowd Test teilgenommen, davon haben sich 14 Personen in der App registriert. Die Testgrösse wurde für die Kosteneinschätzung auf 2000 aktive Benutzer hochskaliert. Ein Spiel kostet gemäss dieser Analyse maximal USD 7.71.

AWS Service	Kosten je Spiel	Kosten pro Monat
AWS Amplify	USD 3.79	USD 0.00
AWS AppSync	USD 1.66	USD 0.00
AWS CloudWatch	USD 0.00	USD 0.00
AWS Cognito	USD 0.00	USD 0.00
AWS DynamoDB	USD 1.29	USD 0.00
AWS Key Management Service (KMS)	USD 0.01	USD 0.00
AWS Lambda	USD 0.04	USD 0.00
AWS Route 53	USD 0.00	USD 0.50
Simple E-Mail Service (SES)	USD 0.23	USD 0.00
AWS Simple Storage Service (S3)	USD 0.69	USD 0.00
Total	USD 7.71	USD 0.50

Tabelle 74: Kosten gemäss Crowd Test

Stellt man die einzelnen Dienste in Relation zueinander und filtert die unbedeutenden Positionen heraus, ergibt das folgende Preiszusammensetzung für ein Spiel.

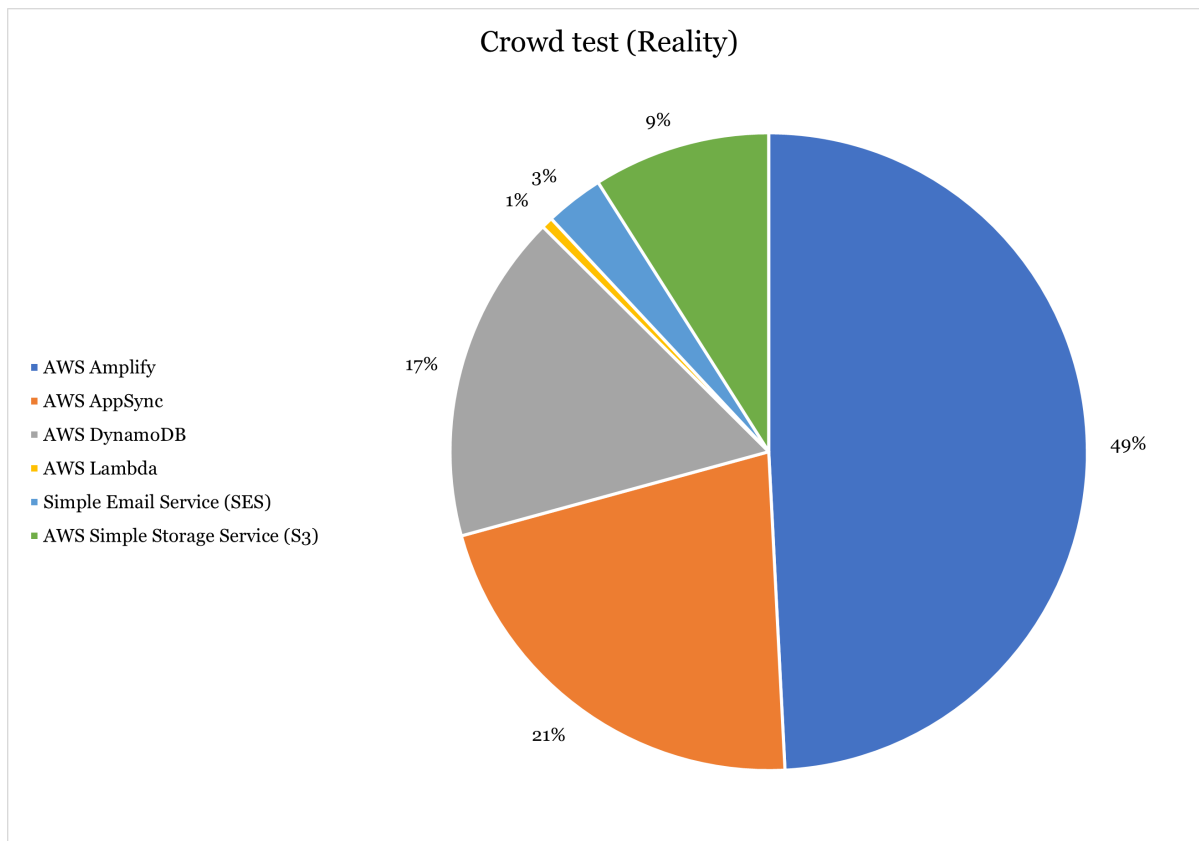


Abbildung 57: Preiszusammensetzung eines Spiels gemäss Crowd Test

Vergleicht man diesen Chart mit jenem aus den End-to-end Tests ist auch hier der Kostenanteil von AWS Amplify sehr gross. Sehr erstaunlich ist die starke Reduktion der S3-Kosten im Kontrast zu den E2E-Tests. Dies lässt sich durch das bessere Caching des Benutzer-Browsers gegenüber dem Test-Browser rechtfertigen. Da nur rund 50% der Benutzer eine Registrierung vorgenommen haben, ist auch die Halbierung der SES-Aufwände nachvollziehbar.

6.5. Engagement Test

Diese App soll das Fan Engagement unterstützen. Entsprechend gibt es auch eine Anforderung dafür zu erfüllen. Mit diesem Test sollen die beim Benutzer ausgelösten Emotionen gemessen und eingeordnet werden können.

6.5.1. Konzept

Die Emotionen werden mit zwei verschiedenen Konzepten getestet.

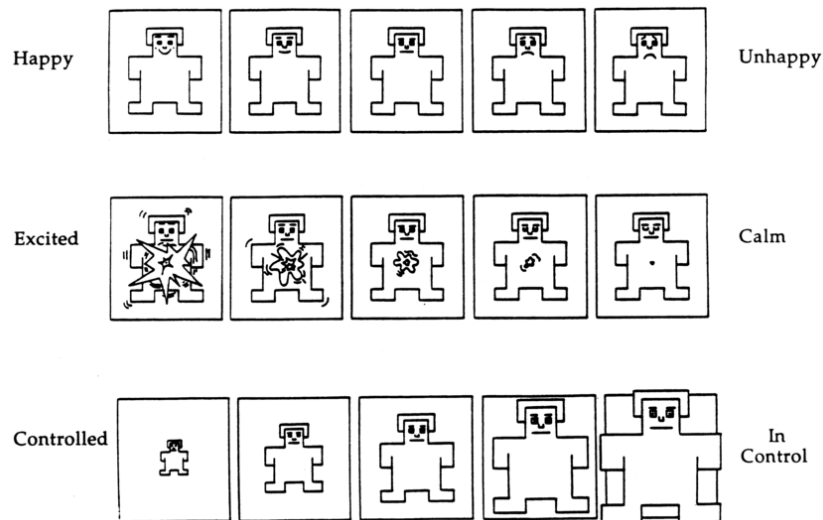


Abbildung 58: SAM Skala [40]

Das *Self-Assessment Manikin (SAM)* ist eine sehr kurze und gut verständliche Methode, um Emotionen zu messen. Sie basiert auf der Drei-Faktoren-Theorie der Emotionen (Russell & Mehrabian, 1977), welche jede emotionale Empfindung in die drei Komponenten Valenz, Erregung und Dominanz einteilt. In diesem Usability Test wird aber auf die Dominanz bewusst verzichtet. Die verwendete Skala geht von 1 bis 5 und wird mit passenden Piktogrammen visuell unterstützt. Die Probanden füllen den Fragebogen einmal vor und einmal nach der Interaktion mit der Applikation aus. Damit kann festgestellt werden, wie sich die Nutzung dieser Applikation auf das emotionale Empfinden des Probanden auswirkt. [41, 42]

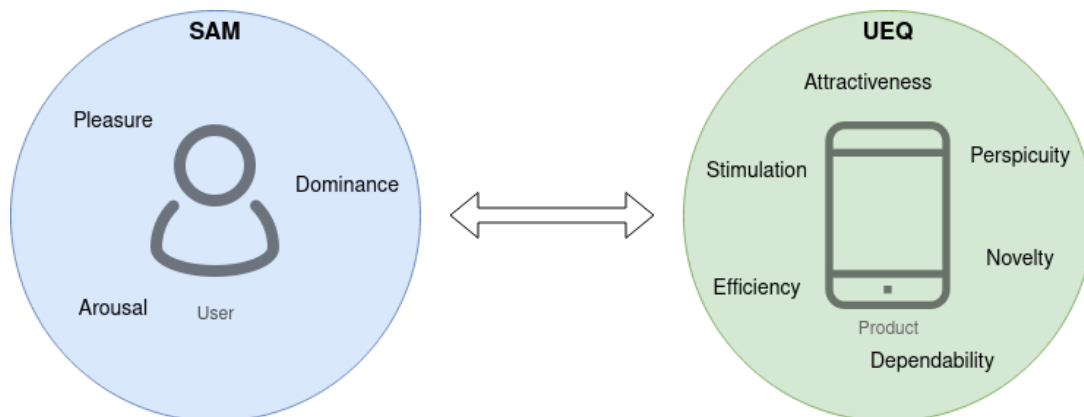


Abbildung 59: SAM und UEQ im Vergleich

Im Vergleich zu *SAM* liefert *User Experience Questionnaire (UEQ)* eine Möglichkeit die Eindrücke gegenüber einer Applikation, um damit dessen Qualität zu messen. Jede Frage im Formular hat 7 Auswahlmöglichkeiten. Die insgesamt 26 Skalen zu verschiedensten Gefühlen erlauben eine qualitative Beurteilung des UX hinsichtlich Attraktivität, Durchschaubarkeit, Effizienz, Vorhersagbarkeit, Stimulation und Originalität.

Skala	Fragestellung
Attraktivität	Gesamteindruck der App. Mögen die Benutzer die App oder nicht?
Durchschaubarkeit	Wie einfach kann man sich mit der App vertraut machen und dessen Benutzung lernen?
Effizienz	Können die Benutzer ihre Aufgabe ohne unnötigen Aufwand lösen und wie schnell reagiert die App?
Vorhersagbarkeit	Hat der Nutzer das Gefühl, die Interaktion zu kontrollieren? Ist sie zuverlässig und vorhersehbar?
Stimulation	Ist die App-Benutzung spannend und motivierend? Macht sie Spass?
Originalität	Ist das Design der App kreativ? Weckt sie das Interesse der Benutzer?

Tabelle 75: UEQ Skalen

Dieser Fragebogen wird einmalig im Nachinterview ausgefüllt. Die genaue Systematik der Methode kann im Handout auf der offiziellen *UEQ* Webseite nachgelesen werden.[43]

Vorinterview

Im Vorinterview werden einige demografische Fragen gestellt, um den Proband besser einzuordnen. Für *SAM* wird ein erster Gefühlsstand abgeholt.

Information	Beispielfragen
Bezug zum Thema	Wie häufig bist du in einem Hockeystadion? Verfolgt du Icehockey?
Gefühlszustand	Gebe bitte im Formular an, wie du dich jetzt im Moment fühlst.

Tabelle 76: Engagement Test Vorinterview

Nachinterview

Information	Beispielfragen
Gefühlszustand	Gebe bitte im Formular an, wie du dich jetzt im Moment fühlst.
Charakteristik	Wie würdest du die App in seiner Coolness bewerten?

Tabelle 77: Engagement Test Nachinterview

6.5.2. Durchführung

Das Engagement wird im Rahmen des Crowd Tests gemessen. Die benötigten Informationen werden mit einer Onlineumfrage abgefragt. Somit lassen sich gehäufte Resultate sehr schnell und fehlerfrei analysieren.

6.5.3. Auswertung

Zur Auswertung der Emotionen nach *SAM* werden die Mittelwerte für die zwei Dimensionen einzeln berechnet. Die beiden Ergebnisse aus dem Vor- und Nachinterview werden für jede Dimension separat verglichen.

Dimension	Vorinterview	Nachinterview
-----------	--------------	---------------

Valenz	2.39	2.27
Erregung	3.81	3.48

Tabelle 78: Engagement SAM Resultate

Die Mittelwerte haben sich in beiden Dimensionen verschlechtert. Auf den ersten Blick suggerieren diese Werte einen negativen emotionalen Effekt der App auf die Benutzer. Es gibt aber durchaus noch andere Erklärungen für diese überraschende Wertveränderung. Laut Vorinterview ist gerade einmal eine Person Fan von Eishockey und somit in der Zielgruppe unserer App. Die restlichen 32 Testteilnehmer haben entweder nie oder selten einen Kontakt zum Eishockey. Bei diesen Personen positive Emotionen mit einem ihnen weder vertrauten noch spannenden Kontext zu wecken, grenzt an Fiktion. Ein weiterer Grund für das unerfreuliche Ergebnis könnte die Umfrage selber liefern. So ist es möglicherweise nicht gelungen, mit den Fragestellungen und Karrikaturen genügend Klarheit für die Fragen zu schaffen. Das führt zu Unmut und Lustlosigkeit und schliesslich zu beliebigen Eingaben.

Für die Analyse der *UEQ*-Umfrage wird auf ein bereitgestelltes Excelsheet zurückgegriffen. In diesem Dokument können die Antwortblätter eingelesen und für die Analyse aufbereitet werden. Die wissenschaftliche Grundlage für diesen Test liefert ebenfalls das *UEQ* Handout. Werte zwischen -0.8 und 0.8 entsprechen einer neutralen, Werte darüber einer positiven und Werte darunter einer negativen Bewertung.[43]

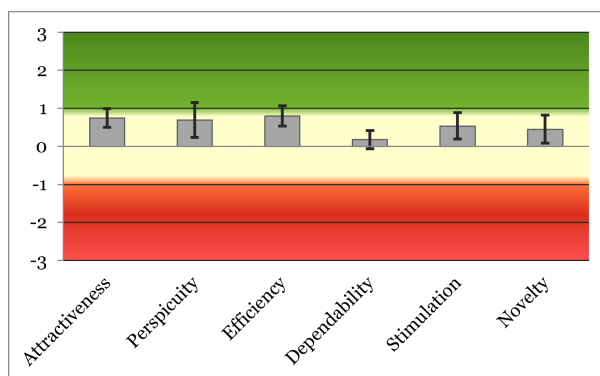


Abbildung 60: UEQ Resultate

In diesem Sinne wird die Applikation in allen Skalen als neutral bewertet. Die höchsten Werte werden in der Effizienz erzielt. Die niedrigsten Werte erreicht die Vorhersagbarkeit. Um die Wirkung der App auf die Benutzer abschliessend richtig einzordnen, sollen Resultate von Konkurrenzprodukten oder frühere App-Versionen für einen Vergleich herangezogen werden. Diese Daten sind im gegebenen Kontext jedoch nicht vorhanden. Deshalb wird der Vergleich mit einem allgemeinen öffentlichen Datenset gemacht.

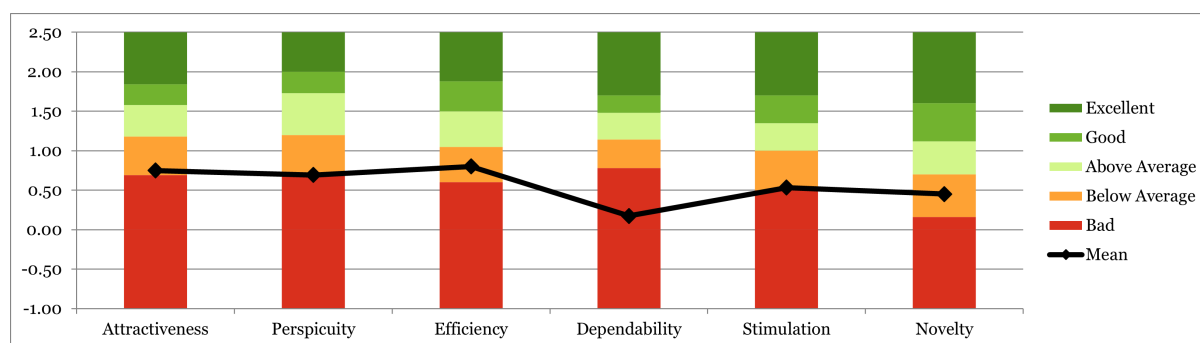


Abbildung 61: UEQ Vergleich mit Benchmark

Anhand der Resultate aus 468 anderen Studien wird jede Skala in fünf prozentual unterschiedliche Bereiche eingestuft. Die App muss sich darin unterdurchschnittlich einordnen. Besonders in der Vorhersagbarkeit fällt sie massiv ab. Das mag manche schockieren, aber solche Auswertungen sind immer mit Vorsicht zu geniessen. Beachtet man vermindert die Einwände aus der *SAM*-Analyse und die Tatsache, dass für diese Analyse mindestens 50 Teilnehmer empfohlen werden, bietet leider auch diese Studie zu viel Varianz und Interpretationsspielraum in seiner Aussage. Nichtsdestotrotz ist diese Messung ein erster Eindruck und kann als Basis für zukünftige verwendet werden. Schaut man sich nämlich die Antworten im Detail an, fällt die besonders negative Bewertung in der einzelnen Fragestellung zur Vorhersehbarkeit auf. Korreliert man diese Information mit den Ergebnissen aus dem Usability Test kann man dies möglicherweise auf die fehlende Berechenbarkeit in der Tippabgabe zurückführen. Wo der genaue Speicherzeitpunkt und die Benutzerführung noch unklar sei.

6.6. Systemtest

Der Systemtest prüft den gesamten Funktionsumfang des Tippspiels. Es gibt eine Checkliste, die alle Prozesse abbildet. Der Fokus ist das Zusammenspiel der verschiedenen Systemkomponenten, insbesondere zwischen Front- und Backend. Der Erfolg wird bei jeder Durchführung festgehalten. Die Checklisten von vollendeten Prüfungen sind im Anhang (Kapitel A.7) unter den Testprotokollen ersichtlich.

7. Weiterentwicklung

Die Fan-Engagement-App hat einen Zustand erreicht, in welchem sie eingesetzt werden kann. Der Funktionsumfang deckt den gesamten Ablauf eines Tippspiels ab. Im Verlauf der Bachelorarbeit sind noch weitere Ideen entstanden, welche aufgrund des beschränkten Umfangs der Arbeit nicht behandelt wurden. Sie zeigen das bestehende Potenzial, das Projekt weiterzuentwickeln, um die User Experience abwechslungsreicher zu gestalten. Einige Vorbereitungen dafür sind bereits getroffen worden.

7.1. Challenge-Typen

Im aktuellen Zustand können nur Challenges erstellt und verwendet werden, welche Spieler als Antwortmöglichkeiten besitzen. Dies beschränkt die Summe der möglichen Challenges. Das Tippspiel soll abwechslungsreich bleiben, um das Fan-Engagement zu verbessern. Dies ist auch die Hauptaufgabe der App. Aus diesem Grund wurden bereits in der Planungsphase Vorbereitungen getroffen, um neben der `<onePlayerChallenge>` weitere Challenge-Typen anbieten zu können. Die Logik im Backend für `<oneOfManyChallenge>`, `<estimationChallenge>` und `<resultChallenge>` ist bereits implementiert und einsatzbereit. Lediglich das Frontend muss ausgebaut werden, um diese Funktionalität anbieten zu können.

Die `<ChallengeDetail.tsx>`-Komponente ist zurzeit für die Darstellung der Challenge verantwortlich. Sie beinhaltet auch die Komponente für das Erstellen und Bearbeiten eines Tipps. Damit neue Challenge-Typen angeboten werden können, müssen neue React-Komponenten für die Abgabe der Antworten entwickelt werden. Diese können dann in typenspezifische `<ChallengeDetail>`-Komponenten eingebunden werden, die es ebenfalls zu implementieren gilt.

7.2. Automatisierung

Für den Betrieb eines Tippspiels müssen derzeit alle Daten im Admin-Bereich manuell konfiguriert werden. Um den Aufwand für die Durchführung eines Spieles zu verringern, sollen Daten von verschiedenen APIs dazu genutzt werden, Tippspiel-Prozesse zu automatisieren.

Verbands-API

Diese API stellt Daten zu allen durchgeführten Spielen bereit und bietet sogar einen Push-Service an, durch den bestimmte Informationen ohne Nachfrage in Echtzeit gesendet werden.

Kinexon-API

Mit den Tracking-Daten von Kinexon könnten ebenfalls automatisch Challenges gespielt werden. Jeder Spieler trägt einen Tracker, womit die Bewegungen und Aktionen auf dem Feld erfasst werden.

7.2.1. Kader

In der Spielvorbereitung ist besonders die Erfassung des Aufgebots mühsam und fehleranfällig. Mit einer entsprechenden Integration der Verbands-API könnte das Aufgebot für ein Spiel automatisch importiert werden, die manuelle Auswahl wäre dadurch nicht mehr nötig.

Dafür muss ein Tippspiel mit einem Spiel und die Spieler in der App mit einem Spielerprofil des Verbandes assoziiert werden können. Erst danach ist eine Importierung vom Kader zu einem bestimmten Spiel technisch möglich.

7.2.2. Challenge-Durchführung

Die Verbands- und Kinexon-API kann ein breites Repertoire an Spieldaten liefern. Diese Fülle an Informationen sollen von der Applikation erfasst und zur autonomen Durchführung von Challenges verwendet werden. Solche Challenges würden selbstständig starten, stoppen und/oder sich auswerten. Die Applikation wechselt den Status solcher Challenges je nach Spielverlauf und -ereignis und der daraus resultierenden Informationen.

Beispielsweise könnte man dieses Konzept mit vordefinierten, nicht archivierbaren und veränderlichen Challenges realisieren, die dann auf eine spezifische API-Abfrage gemappt werden.

7.3. Ausbau der Push-Benachrichtigungen

Öffnet ein Nutzer das Tippspiel zum ersten Mal in seinem Browser, wird er nach der Erlaubnis für Push-Benachrichtigungen gefragt. Diese Auswahl kann er in seinen Einstellungen verändern. Geräte, auf welchen die erlaubnis vorhanden ist, erhalten beim Start eines neuen Tippspieles eine Benachrichtigung. Bisher ist das der einzige Zeitpunkt, bei dem dieser Notification-Service genutzt wird. Eine weitere, sinnvolle Einsatzmöglichkeit ist die Verlosung der Preise. Damit Benutzer einen gewonnenen Preis sehen können, müssen sie aktuell die App öffnen. Um zu verhindern, dass Benutzer dieses Ereignis verpassen, könnte auch den Gewinnern von Verlosungen eine Push-Benachrichtigung gesendet werden.

Um selektive Benutzer zu benachrichtigen, muss der Datenbankeintrag der Push-Service-Subscriptions ausgebaut werden. Bisher werden einfachheitshalber keine Benutzerinformationen zu den Push-Service-Registrierungen gespeichert. Somit existiert keine Verbindung zwischen Benutzern und Subscriptions.

7.4. Newsletter-Registrierung

Die SCRJ Lakers versenden regelmässig einen Newsletter mit Informationen zu den Heimspielen und weiteren Aktivitäten. Auf ihrer Webseite haben sie ein Formular, über das sich Besucher für diesen Newsletter anmelden können.

Um weitere Empfänger für den Newsletter zu akquirieren, entstand die Idee, dass Benutzern der Fan-Engagement-App auch eine Möglichkeit der Anmeldung geboten wird. Diese soll in Form einer Checkbox in den Registrierungsprozess eingebettet werden. Im Frontend der Applikation ist dies eine kleine Anpassung. Die Herausforderung besteht im nachfolgenden Prozess. Im Registrierungsprozess eines Benutzers ist die E-Mail-Adresse noch nicht verifiziert und diese Verifikation ist eine Voraussetzung für die Newsletter-Registrierung. Erschwerend kommt hinzu, dass bestehende Newsletter-Registrierungen samt Anrede, Vorname und Nachname sind. Diese Informationen fehlen bisher in der App und müssten allenfalls ergänzt werden.

7.5. Laufzeit-Optimierung Lambdas

Einige Prozesse, wie die Benutzermigration, sind generell noch eher träge. Andere Abfragen zeigen bei grösserer Nutzlast Schwächen. Einige Engpässe konnten bereits durch Parallelität im Code und die Anhebung des verfügbaren Arbeitsspeichers beseitigt werden. In der AWS Cloud gibt es aber noch weitere Möglichkeiten, die Latenz zu verringern. Das Potential ist noch nicht ausgeschöpft und die Reaktionszeit für den Benutzer noch nicht immer zufriedenstellend. Hiermit soll bewusst Zeit für die Reduktion der Laufzeiten von Lambdas genommen werden.

Provisioned Concurrency

Kaltstart sind für einen beachtlichen Anteil der Latenz verantwortlich. Provisioned Concurrency ist eine Methode, um Kaltstarts zu verringern. Dabei werden präventiv eine geplante Anzahl Lambdas hochgefahren, noch bevor diese aufgerufen werden. Dadurch müssen diese Funktionen nicht mehr initialisieren und können direkt mit der Bearbeitung einer Anfrage loslegen.[39]

BatchWrite

Momentan werden sämtliche Schreibbefehle an die Datenbank für jeden Datensatz einzeln gestellt. Müssen viele Datenbankeinträge gleichzeitig aktualisiert werden, führt dies zu sehr vielen Verbindungen zwischen Lambda und Datenbank. Ist das Leistungs- oder Verbindungskontingent aufgebraucht, müssen restliche Operationen zuwarten bis Kapazitäten wieder frei werden. Der Verbindungsauf- bis abbau braucht viel Zeit, die nicht zu unterschätzen ist. AWS hat Möglichkeiten geschaffen, mehrere Aktionen mit einer einzigen Anfrage und Verbindung auszuführen.[44]

Trigger

Einige Datenbankeinträge müssen aufgrund anderer Datenbankänderungen durchgeführt werden. Diese verhältnismässig zeitintensiven Prozesse sind Bestandteil von Benutzeraktionen. Darunter leidet die Antwortzeit, da die vollständige Durchführung abgewartet werden muss. Die Idee ist, dass diese Prozesse aufgrund von Datenbank-Mutationen statt Benutzeraktionen gestartet werden und somit unabhängig laufen. Diese Konzepte sollen testweise angewendet und mit dem bisherigen Konzept praktisch verglichen werden.

7.6. Passwordless-SMS-Login

Das Tippspielgerät ist das Smartphone. Jeder hat dieses im Stadion dabei. Aber nicht jeder kann seine E-Mails darauf abrufen. Deshalb entstand die Idee, das Passwordless-Login auch mit einer SMS zu ermöglichen oder gar zu forcieren. Der Loginprozess bliebe derselbe. Es ändert sich nur der Kommunikationskanal für den Magic Link von der E-Mail zum SMS. Die Registrierung mit einer Telefonnummer ist im Hintergrund bereits möglich. Was fehlt sind angepasste Formulare und der SMS-Versand.

7.7. Ausbau der Gamification

Das Spielkonzept wurde auf die Anforderungen zugeschnitten und in einem iterativen Prozess weiterentwickelt. Auch in diesem Bereich entstanden Ideen, die nicht umgesetzt werden konnten. Diese wurden in ein Backlog aufgenommen und eignen sich für die Weiterentwicklung der Fan-Engagement-App.

7.7.1. Badges

Bisher werden richtige Tipps mit Punkten und der Teilnahme am Gewinnspiel belohnt. Eine weitere Belohnung könnte in der Ausstellung von sogenannten Badges bestehen. Dieser Mechanismus wird bereits von vielen Spielen genutzt, um Anwendern einen weiteren Anreiz für die aktive Nutzung zu geben. Das Sammeln von Badges befriedigt bei vielen Menschen das Bedürfnis nach Anerkennung von anderen Spielern.

7.7.2. Loyalitätsbonus

Die stetige Teilnahme an Tippspielen könnte mit der Ausstellung eines Loyalitätsbonus belohnt werden. Für jedes aufeinanderfolgende Tippspiel könnten dem Spieler direkt Punkte gutgeschrieben werden. Eine weitere Variante ist es, einen Faktor für erzielte Punkte zu implementieren, welcher sich durch die regelmässige Tippenspiel-Aktivität erhöht. Die je aktuelle Anzahl aufeinanderfolgender Teilnahmen könnten zur Motivation im Profil dargestellt werden.

7.7.3. Streaks

Sind mehrere konsekutive Antworten eines Spielers richtig, bilden diese eine <Streak>. Diese kann dem Benutzer dargestellt werden und zusätzliche Punkte für folgende, korrekte Tipps generieren. Die aktuelle und die längste <Streak> könnten in den Profilstatistiken aufgeführt werden.

7.8. Aufwandschätzung

Die prognostizierten Aufwände entsprechen einem Entwickler, der sich bereits in das Produkt eingearbeitet hat und die Technologien kennt. Es handelt sich effektiv nur um Schätzungen, die als Anhaltspunkte dienen. Diese hängen stark davon ab, was genau erreicht werden soll. Einige Ideen sind sehr ausgefallen und noch konzeptlos. Auf eine Schätzung dieser Features wird deshalb verzichtet.

Priorität	Feature	Aufwand
1	Newsletter-Registrierung ¹	48 h
2	Push-Benachrichtigung für Gewinne ¹	24 h
3	Laufzeit-Optimierung Lambdas ¹	48 h
4	Challengetyp <estimationChallenge>	24 h
5	Challengetyp <resultChallenge>	24 h
6	Passwordless SMS-Login	40 h
7	Kader-Import von Verbands-API	56 h
8	Automatische Challengeauswertung	120 h
9	Challengetyp <oneOfManyChallenge>	40 h

¹ Entwickler empfehlen Umsetzung noch vor erstem Release

Tabelle 79: Aufwandschätzung

8. Schlussbericht

Die im Vorfeld der Arbeit definierten Ziele sind erreicht worden. Die Anforderungen werden durch die Fan-Engagement-App vollständig erfüllt. Ein Grossteil der ursprünglichen Use Cases sind realisiert und die restlichen sauber dokumentiert. Der Kunde ist mit dem Ergebnis restlos zufrieden und eine Weiterentwicklung wird angestrebt.

Auch in technischer und methodischer Hinsicht können durch das Projekt neue Erkenntnisse gewonnen und Schlussfolgerungen gezogen werden.

Mit User-Centered-Design sind die UX früher und ressourcenschonender entwickelt. Denn durch die Feedbacks aus den Usability Tests und das effiziente Prototyping ist das UX bereits auf ein sehr gutes Niveau optimiert und die Anforderungen angepasst bevor detaillierte Pläne umgesetzt werden. Auch die fortlaufenden Usability Tests mit dem Produkt leisten einen überraschend grossen Beitrag zur finalen App. Das ist die erste von vielen Erkenntnissen aus diesem Projekt.

In der Architektur auf Serverless Computing zu setzen und das Deployment in einer Public Cloud zu machen ist immer noch der absolut richtige Ansatz für diese Art von Applikation. Die vorgelegten Argumente zugunsten dieses Konzeptes sind nach wie vor erfüllt. Die Kosten für ein sehr gut ausgelastetes Tippspiel mit 2000 Nutzern sind maximal in einem einstelligen Bereich und die dynamische Skalierung gewährleistet jederzeit die vollständige Funktionalität. Auch die Zuverlässigkeit und Verfügbarkeit leidet nicht darunter.

AWS Amplify ist nach wie vor eine sehr mächtige Lösung. Trotz mehrerer Environments bleibt für den Entwickler die Verwaltung der Konfiguration sehr angenehm. Es setzt auch alle Berechtigungen für die eingesetzten Dienste in der AWS Cloud. Das sind nicht zu unterschätzende Mehrwerte. Die *CLI* ist aber nicht 100% fehlerfrei und führt teilweise zu unerwarteten Fehlern. Im Einsatz für diese Applikation ist AWS Amplify auch an einige Grenzen gestossen. Aus den gemachten Erfahrungen und mit den sich veränderten Anforderungen gibt es daher erhebliche Zweifel, ob diese Entwicklungsplattform in einer erneuten Evaluierung noch die Beste wäre. Beispielsweise ist auch ein grosser Aufwand für die Etablierung der neu entstandenen Anforderung «Passwordless-Login» betrieben worden. Google Firebase beschreibt in seiner Dokumentation eine adäquate vollständig integrierte Lösung. Es wäre auch interessant, mit welchem Aufwand die verwendeten Dienste direkt für die AWS Cloud konfiguriert werden könnten. Nichtsdestotrotz vereinfacht AWS Amplify den Einstieg in die AWS Cloud massiv.

In der Fan-Engagement App haben erste Gamification-Elemente Einzug gehalten. Mit den Tippspielpreisen, der Rangliste und den rudimentären Statistiken sind drei Anreize geschaffen, die Fans zum regelmässigen Tippen im Stadion zu animieren.

Das mit diesem Projekt entstandene Konzept zur Messung des Fan Engagements hat im Crowd Test nicht die erwünschten Ergebnisse erzielt. Studenten ohne Eishockeybezug und die Atmosphäre Vorlesungssaal sind nicht geeignet für eine wertige Messung der Emotionen. Eine Validierung mit Fans im Stadion kann im Projektzeitraum nicht durchgeführt werden, da die Saison zu kurz war.

A. Testprotokolle

A.1. Usability Tests

A.1.1. Jonas

Usability Test

Testprotokoll Prototyp

Teilnehmer

Name: *Jonas Hager*

Alter: *19*

Technische Kenntnisse:

gut, kann sowohl Smartphone als auch Computer für seine Bedürfnisse uneingeschränkt nutzen

Bezug zum Eishockey: *besucht mind. 50% aller Heimspiele und auch Auswärtsspiele*

Ergebnisse

Task1: Erstelle einen neuen Account mit deinem Google Konto.

	++	+	-	--	Bemerkung
Navigierung	X				<i>Abwand konnte direkt fehlerfrei alle Benützungspunkte ihrem Inhalt und Zweck zuordnen</i>
Präsentation		X			
Bedienung		X			

Task2: Finde deine Platzierung in der öffentlichen Rangliste und notiere diese.

	++	+	-	--	Bemerkung
Navigierung	X				
Präsentation			X		<i>Tabelle nach nicht schön</i>
Bedienung		X			

Task3: Wähle Melvin Nyffeler bei der Frage im Tippspiel, bei der du noch keine Auswahl getroffen hast.

	++	+	-	--	Bemerkung
Navigation			X		Wolte fälschlicherweise auf Auswahl ändern klicken um zu speichern und verlassen (Speichern ist unklar)
Präsentation			X		Begriff "Angelöst" als Resultat bereits vorhanden interpretiert dafür "Tippen" unklar, hat nicht ausgefüllte Challenges nicht gefunden "Geschieben" heißt Du hast bereits ausgefüllt
Bedienung		X			Typ "Geschieben" ev. zu beendet ändern Offen: Angefangen aber noch nicht selber ausgefüllt

Task4: Ändere deine Auswahl bei "Wer macht am meisten Sprints?" zu Roman Cervenka.

	++	+	-	--	Bemerkung
Navigation			X		Nicht auf Anhieb bestehenden Tipp gefunden
Präsentation		X			Man hat eine gute Übersicht, es sei ein schönes Layout
Bedienung		X			

Task5: Suche das Resultat bei einer Frage, bei welcher das Resultat bekannt ist. Hast du dort richtig getippt?

	++	+	-	--	Bemerkung
Navigation		X			Sucht direkt mit Filter
Präsentation		X			
Bedienung		X			

Task6: Wo im Stadion kannst du den Gutschein für einen Softdrink abholen, den du gewonnen hast?

	++	+	-	--	Bemerkung
Navigierung		X			
Präsentation			X		Preise unklar, da es persönliche oder allgemeine sind Erster Gedanke zur Kasse: Getränkeausgabe ins Stadion Zweiter Gedanke zur Kasse: Ticketcenter
Bedienung		X			

Task7: Welche Platzierung hat "zindable" in der öffentlichen Rangliste?

	++	+	-	--	Bemerkung
Navigierung		X			
Präsentation		X			
Bedienung		X			

Feedback:

	Bemerkung
Optik	Als großer Sponsor der Lakers ist das Design sehr passend, er fühlt sich direkt mit dem Verein verbunden
Beschriftungen	Begriffe "Offen" und "Geschlossen" unklar
Platzierungen	Ist gut
Allgemein	Auf der Startseite war für Proband alles sehr klar

A.1.2. Prisca

Usability Test

Testprotokoll Prototyp

Teilnehmer

Name: Prisca Hager

Alter: 50+

Technische Kenntnisse: beschränkt, nutzt täglich Smartphone und Laptop für einfache Aufgaben

Bezug zum Eishockey: verfolgt es mit Distanz mit (im Fernsehen, Zeitung), nie selber im Stadion

Ergebnisse

Task1: Erstelle einen neuen Account mit deinem Google Konto.

	++	+	-	--	Bemerkung
Navigierung		X			Sie hat den Einstiegspunkt nicht gefunden
Präsentation		X			
Bedienung			X		Account-Button könnte etwas größer sein

Task2: Finde deine Platzierung in der öffentlichen Rangliste und notiere diese.

	++	+	-	--	Bemerkung
Navigierung			X		Der Probandin war unklar, wie sie wieder zurück kommt
Präsentation			X		Tabellarisch macht visuell noch nicht befriedigend
Bedienung		X			

Task3: Wähle Melvin Nyffeler bei der Frage im Tippspiel, bei der du noch keine Auswahl getroffen hast.

	++	+	-	--	Bemerkung
Navigation	X				Navigation ist sehr eindeutig
Präsentation			X		"Offen" wird mit "Nein" / "Nein nicht getippt"
Bedienung			X		Möglichkeit zum Scrollen unclear Unclear, das Eingabe / Auswahl

bereits übermittelt wurde

Task4: Ändere deine Auswahl bei "Wer macht am meisten Sprints?" zu Roman Cervenka.

	++	+	-	--	Bemerkung
Navigation			X		Hatte Schwierigkeiten, existierenden Tipp zu erkennen
Präsentation		X			Geschlossen wird mit "Nein nicht entschieden" assoziiert
Bedienung		X			

Task5: Suche das Resultat bei einer Frage, bei welcher das Resultat bekannt ist. Hast du dort richtig getippt?

	++	+	-	--	Bemerkung
Navigation		X			Nutzt selbständig Filtermöglichkeit
Präsentation			X		Die Punkte sind etwas gar schlecht dargestellt / unscheinbar
Bedienung	X				

Task6: Wo im Stadion kannst du den Gutschein für einen Softdrink abholen, den du gewonnen hast?

	++	+	-	--	Bemerkung
Navigation	X				Schnell gefunden, war eindeutig
Präsentation			X		Mit "Kasae" wird der Ort der Getränkeausgabe verstanden
Bedienung			X		Die Pöbardin hat Slider ansprechen gewünscht und damit den Gutschein eingelöst

Task7: Welche Platzierung hat "zindable" in der öffentlichen Rangliste?

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation	X				
Bedienung		X			

Feedback:

	Bemerkung
Optik	Die Farben sind sehr ansprechend, es wirkt sehr angenehm, es ist nicht zu überladen
Beschriftungen	Rangliste ist etwas klein
Platzierungen	-
Allgemein	Passet

A.1.3. Mario

Usability Test

Testprotokoll Prototyp

Teilnehmer

Name: Mario Zindel

Alter: 24

Ergebnisse

Task1: Erstelle einen neuen Account mit deinem Google Konto.

	++	+	-	--	Bemerkung
Navigierung	X				
Präsentation	X				
Bedienung	X				

Task2: Finde deine Platzierung in der öffentlichen Rangliste und notiere diese.

	++	+	-	--	Bemerkung
Navigierung	X				=> Resultate
Präsentation		X			ult, zu Rangliste umbenennen
Bedienung	X				

Task3: Wähle Melvin Nyffeler bei der Frage im Tippspiel, bei der du noch keine Auswahl getroffen hast.

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation			X		keine Auswahl unklar, Auswahl Bestätigung
Bedienung	X				

Task4: Ändere deine Auswahl bei "Wer macht am meisten Sprints?" zu Roman Cervenka.

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation		X			
Bedienung	X				

Task5: Suche das Resultat bei einer Frage, bei welcher das Resultat bekannt ist. Hast du dort richtig getippt?

	++	+	-	--	Bemerkung
Navigation	X				Filter genutzt
Präsentation	X				
Bedienung	X				

Task6: Wo im Stadion kannst du den Gutschein für einen Softdrink abholen, den du gewonnen hast?

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation			X		Abschort nicht gefunden => Ort klarer formulieren
Bedienung	X				

Task7: Welche Platzierung hat "zindable" in der öffentlichen Rangliste?

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation	X				
Bedienung	X				

Feedback:

	Bemerkung
Optik	Punktestand auf Home störend
Beschriftungen	Resultate => Rangliste, Icon für Challenge-Status oder anders visuell hervorheben
Platzierungen	
Allgemein	Was kann ich gewinnen? Wie funktioniert das Tippspiel?

A.1.4. Rolf

Usability Test

Testprotokoll Prototyp

Teilnehmer

Name: Rolf Emmenegger

Alter: 53

Technische Kenntnisse: Anwender

Bezug zum Eishockey: Grundkenntnisse

Ergebnisse

Task1: Erstelle einen neuen Account mit deinem Google Konto.

	++	+	-	--	Bemerkung
Navigation		X			
Präsentation	X				
Bedienung	X				

Task2: Finde deine Platzierung in der öffentlichen Rangliste und notiere diese.

	++	+	-	--	Bemerkung
Navigation	X				über Resultate
Präsentation	X				Total Anzahl in Rangliste
Bedienung	X				

Task3: Wähle Melvin Nyffeler bei der Frage im Tippspiel, bei der du noch keine Auswahl getroffen hast.

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation	X				
Bedienung		X			keine Bestätigung der Auswahl

Task4: Ändere deine Auswahl bei "Wer macht am meisten Sprints?" zu Roman Cervenka.

	++	+	-	--	Bemerkung
Navigation	X				
Präsentation	X				
Bedienung		X			keine Bestätigung der Auswahl

Task5: Suche das Resultat bei einer Frage, bei welcher das Resultat bekannt ist. Hast du dort richtig getippt?

	++	+	-	--	Bemerkung
Navigation	X				Filter genutzt
Präsentation	X				
Bedienung	X				

Task6: Wo im Stadion kannst du den Gutschein für einen Softdrink abholen, den du gewonnen hast?

	++	+	-	--	Bemerkung
Navigierung	X				
Präsentation	X				
Bedienung	X				

Task7: Welche Platzierung hat "zindable" in der öffentlichen Rangliste?

	++	+	-	--	Bemerkung
Navigierung	X				
Präsentation	X				
Bedienung		X			Versuch zu filtern / suchen

Feedback:

	Bemerkung
Optik	
Beschriftungen	Resultate ⇒ Rangliste
Platzierungen	
Allgemein	Verfügbare Preise anzeigen

A.2. Crowd Test

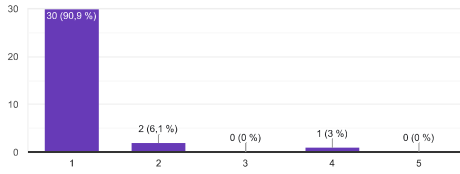
Crowd / Usability Test

33 Antworten

Vorinterview

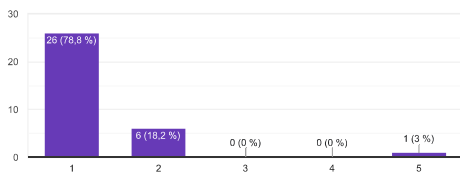
Wie häufig bist du in einem Hockeystadion während einer Saison

33 Antworten



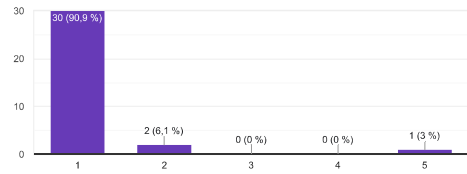
Verfolgst du Icehockey im Fernsehen/Vor Ort?

33 Antworten



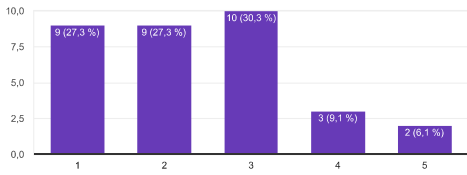
Welchen Bezug hast du zu den SCRJ Lakers?

33 Antworten



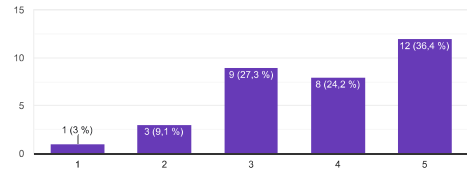
Wie fühlst du dich gerade?

33 Antworten



Bist du eher aufgeregt oder entspannt?

33 Antworten

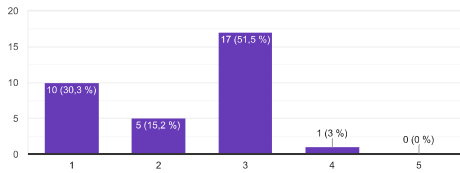


Crowd / Usability Test

Nachinterview 1/3

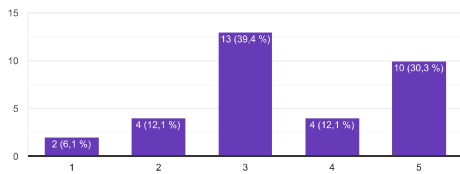
Wie fühlst du dich gerade?

33 Antworten



Bist du eher aufgeregt oder entspannt?

33 Antworten

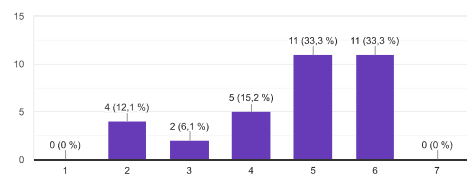


Nachinterview 2/3

Bitte geben Sie Ihre Beurteilung ab.

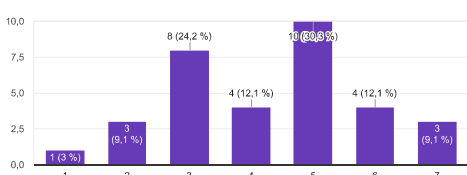
1

33 Antworten



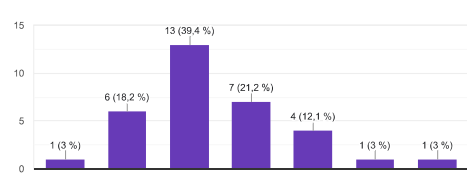
2

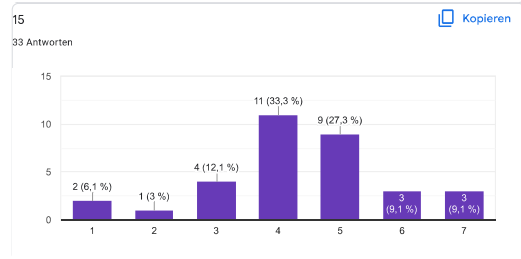
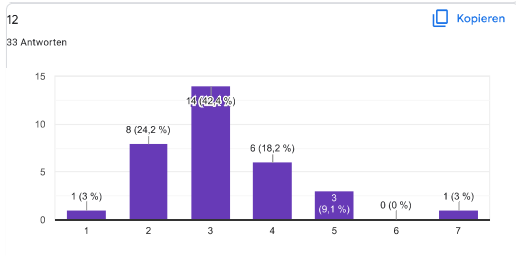
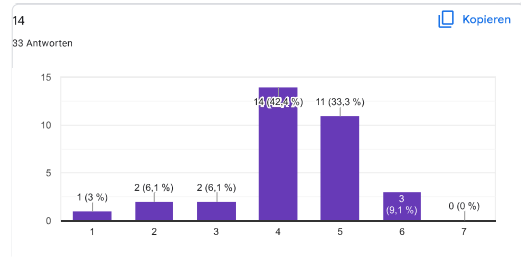
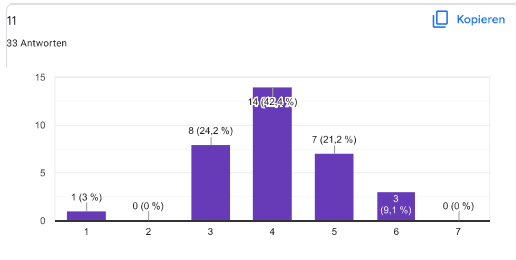
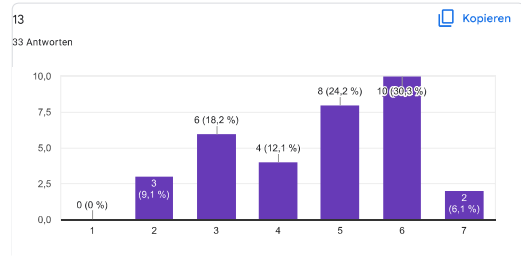
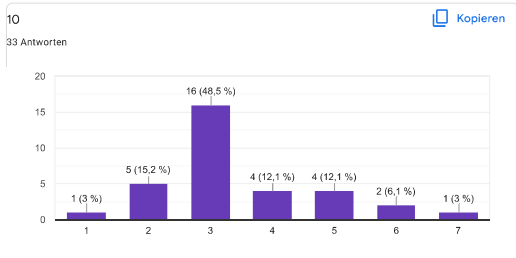
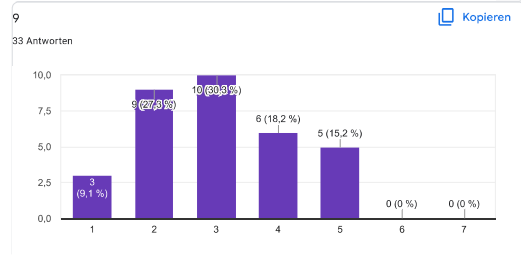
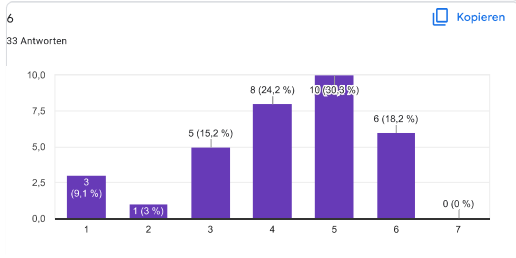
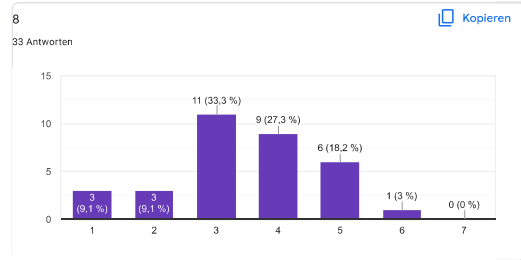
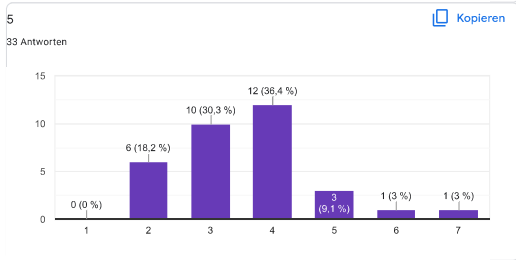
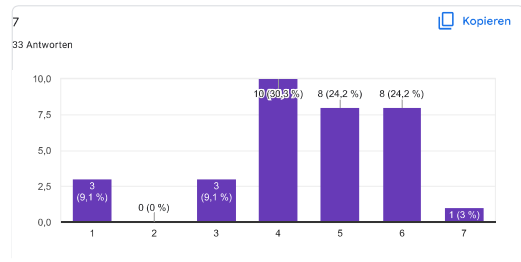
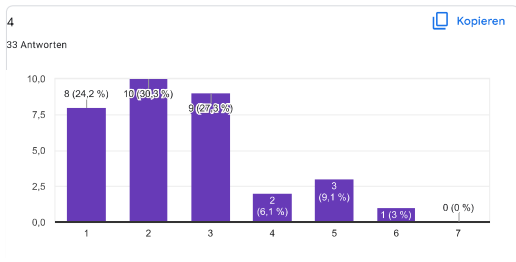
33 Antworten

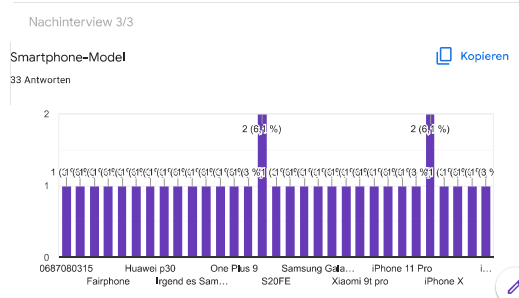
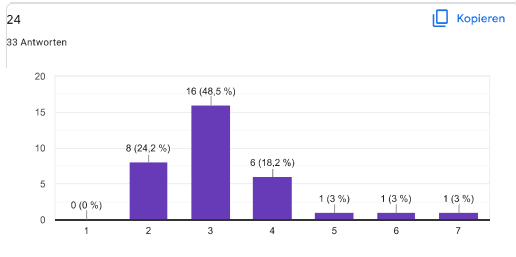
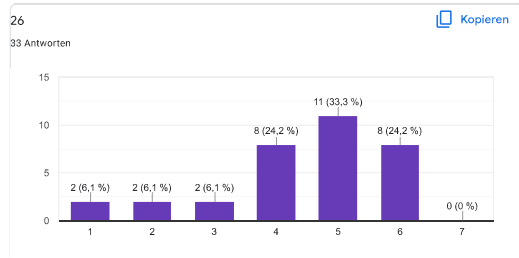
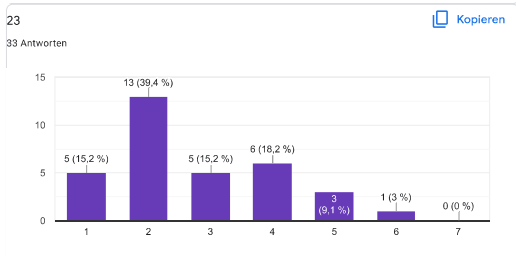
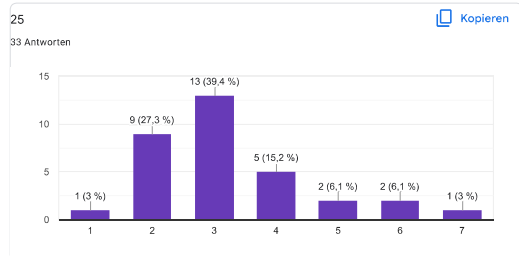
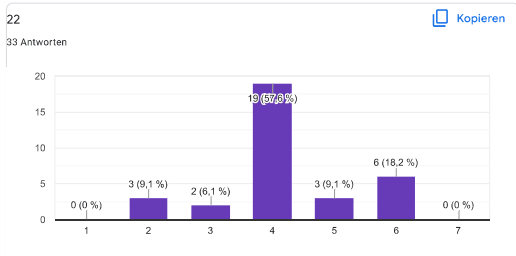
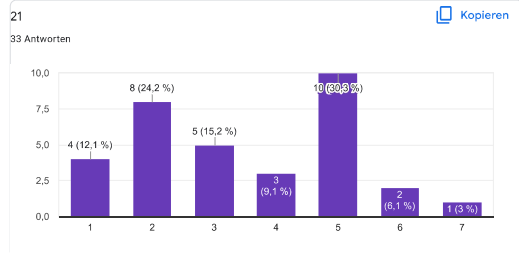
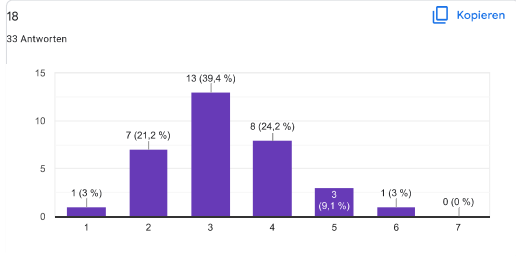
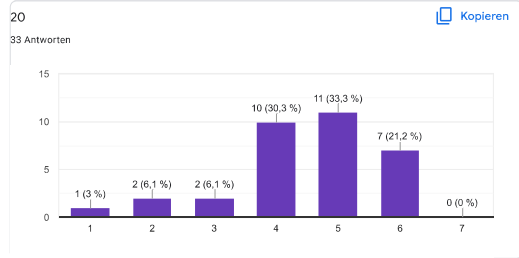
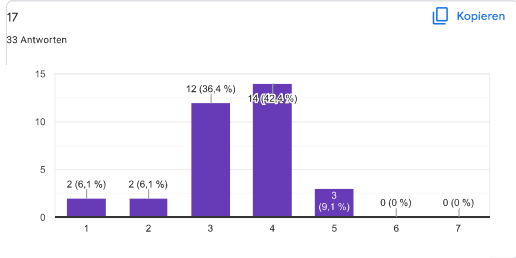
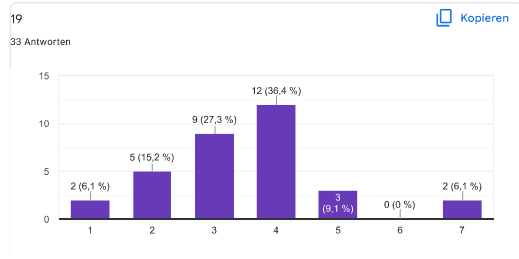
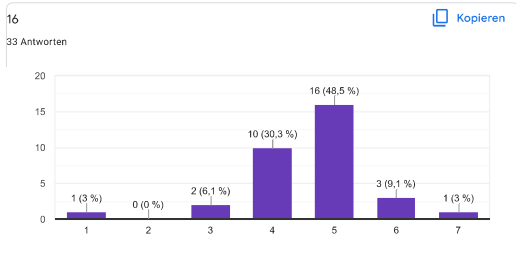


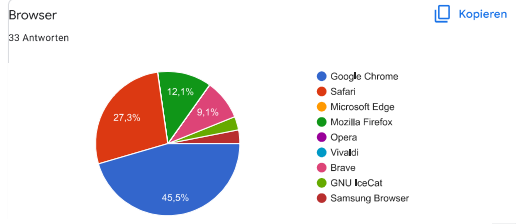
3

33 Antworten









Das ist mir positiv aufgefallen.

22 Antworten

Design

Schön gestaltet

einfach zu erlernen, updated zeitnah (poll sperren, öffnen, ...)

Finde ich noch eine nette idee

Live Aktualisierung der App.

UI irgendwie komisch und broken

einfach zu verstehen und zu verwenden

Die Web-App ist übersichtlich und intuitiv gestaltet.

Schnelle Auswertung der Resultate

Live-updates

Einfache Bedienung, ansprechendes Design

Schnelle Updates auf der Seite

Design war gut und einfach

sieht ok aus

Tippspiel selbst aktualisiert automatisch und scheint sehr zuverlässig zu funktionieren.

kompetitiv

Das Design ist clean und ansprechend. Die Bedienung ist sehr intuitiv.

Geschwindigkeit

Design originell

Übersichtlich gestaltet

Die Updates der fragen werden schnell auf dem app übernommen, app ist übersichtlich und ansprechend. Coole Idee!

Das ist mir negativ aufgefallen.

27 Antworten

Erklärung hat ein wenig gefehlt

Tippspiel geschlossene Tippspiele sind in umgekehrter reihenfolge, neustes ist nicht zuoberst

Viel neu laden der Seite

Polls waren nachher nicht mehr in der gleichen Reihenfolge sichtbar wie sie stattgefunden haben.

Aktualisierung von den punkten nicht dynamisch sonder man muss es selber refreshen. Am anfang wusste ich nicht wo klicken

Hattet ihr nöd gseid dasmer sich aamelde mun zum en Priis abstaube het ich als user mich nöd registriert / es wer mer nöd uufgafalle dasi mi registriere muess. Vlt das nochli hervorhebe :)

die Umfrage war recht wirr. Wusste nicht wirklich was anklicken und hab einfach irgendwas angeklickt.

aktualisiert nicht von alleine

Vieles ist nicht sortiert und deshalb ein wenig verwirrend (Spieler, Antworten).

.

Sortierung in der ganzen app willkürlich

Keine Suchfunktion bei den Spielern macht es unhandlich.

Bei der Spielerauswahl wurde der Goal teilweise verdeckt von der oberen Navigationsleiste, man musste scrollen damit man den Goal ganz erkennen konnte.

Rangliste hat nicht aktualisiert, Ob ein Tippspiel noch aktiv ist, hat sich auch erst aktualisiert, als ich darauf geklickt habe

Bei der Auswahl von einem Quiz wurde mein Bildschirm weiss und allgemein wurde die App langsamer. Es gab kein Suchfeld/Sortierung für die Liste der Spieler. Da die app langsamer wurde hatte ich kaum eine Chance die Spieler in der vorgegebenen Zeit auszuwählen. Die Challenges wurden nicht immer direkt aktualisiert und brauchten manchmal einen reload. Die Bilder der Spieler wurden nicht mehr geladen. Ich fände eine native App einiges besser, als eine Browser Webseite, die man immer wieder mal reloaden muss.

rangliste ist nicht korrekt, ui nicht sehr responsive wenn man drüber hovert/klickt

Damit die Rangliste aktualisiert wird, muss jeweils die Seite neugeladen werden. Die Ränge der Rangliste werden nicht immer korrekt angezeigt.

Verfügbare und abgeschlossene Fragen noch übersichtlicher anzeigen. Countdown anzeigen, wie lange man Zeit hat, um die Frage zu beantworten.

leicht Buggy (Rangliste)

Die Auswahl in der Rangliste gelistet zu werden kann nach der Erstanmeldung nicht mehr geändert werden (zuerst Häkchen nicht gesetzt, nachträglich nicht möglich). Bei der Auswahl der Spieler werden noch zu viele Klicks benötigt -> Vorschlag: beim Tippen auf den jeweiligen Spieler diesen direkt einloggen und zum Beispiel mit einer Umrandung aus der Liste abheben. Dieser Vorgang wäre mir aktuell zu "fummelig"/zu kleine Buttons" wenn ich mit Bier und guter Laune im Stadion stehe ;)

Format, Übersicht

zu viele Spieler bei der Auswahl bei den Fragen

Umfrage: zu viele Fragen und viele Adjektive waren synonyme.
App: Eine Suche bei der Spielerauswahl fehlt. Nach dem Anmelden hat es meine Tipps zurück gesetzt.

Ordnung der Rangliste nicht nachvollziehbar

die App sieht sehr langweilig aus, es hat keinerlei Animationen.

Rangliste falsche Nummerierung

Die Auswahlmöglichkeiten sind zu viele und nicht übersichtlich sortiert. Ich würde nach nummern sortieren. Ranglistensortierung funktioniert noch nicht richtig, bzw. sortierung stimmt, aber rang nummer stimmt nicht

Das hat sich für mich komisch angefühlt.

18 Antworten

Ich mache meine Spielerauswahl und muss nachher immernoch Auswahl speichern klicken. Unnötiger step. Habe meine Auswahl gemacht als ich auf den Namen geklickt habe.

Als App für Fans wahrscheinlich noch einfacher als Webseite

Wenn Filter nach Offene/Abgeschlossene Tippspiel denn hetti erwartet das es die andere ausblendet und ned nur det ane scrollt.

als Antwort hatte man mehr Möglichkeiten als in auf den Folien zu sehen waren.

Sortierung

Die Auswahl des Spielers muss zuerst bestätigt werden, kann etwas umständlich wirken. Die Sortierung der bereits ausgewerteten Spiele sollte umgekehrt sortiert werden. Ebenfalls könnten die Spieler sortiert werden

am Anfang etwas verwirrend, da zwei Abstimmungen offen waren

Für vor dem Spiel oder den Pausen eine gute Idee. Nur möchte ich mich während dem Spiel auf das Spielgeschehen konzentrieren und nicht unbedingt noch irgendwelche Tippspiele ausfüllen.

Das der Filter auf geschlossen, offen etc. sich wie ein One-Pager verhält und die anderen Kategorien nicht ausblendet

Die Ansicht der Preise wird nicht im Popup angezeigt sondern muss extra gesucht werden.

Punkteverteilung, Challenges sind irgendwo, nicht klar welche Challenge aktiv ist wenn 2 aktiv waren. Fragen sind komisch formuliert,

Die Anzeige der Challenges hat sich etwas komisch angefühlt. Die ältesten Spiele sollten zuunterst angezeigt werden, und erst kürzliche oben. Ansonsten muss man um das Resultat der aktuellen Challenge erst ganz nach unten scrollen.

Einloggen der getippten Spieler. (siehe negative Anmerkung) -> mehr straight-forward pts.

Aufbau

Hat einen Speicher Button bei der Spielerauswahl, Auswahl wird aber auch ohne gespeichert -> Unerwartetes Verhalten.

Ordnung der vergangenen Tippspiele nicht nachvollziehbar, Preis abholen Wischen Animation fehlt

Ich finde es komisch, dass wir in der Klasse, das Testing für euch übernehmen müssen.

Antwort wählen ist etwas mühsam zum Suchen

Das habe ich nicht ganz verstanden.

14 Antworten

-

Punkteverteilung

Wenn mehrere Challenges auf einmal

UI, dass man auf Tippspiel klicken muss.

mehr Spieler zur Auswahl als vorne angezeigt wurden

Am Anfang war es schwierig nachzuvollziehen was man machen muss.

Wie wird die Auflösung im Stadion gezeigt? Ohne Video Auflösung wäre es mittelmässig spannend...

Wieso so viele Antwortmöglichkeiten? ui könnte besser sein

Ablauf, wird sich aber mit einem Spiel dazwischen verbessern.

Bei ersten mal öffnen wusste ich nicht direkt welches Tippspiel das offene ist

Ich war unsicher, ob ich die Seite neuladen muss, um die neuesten Daten zu erhalten.

Vollständiger Name (für Rückfragen)

12 Antworten

Michael Enzler

Kevin Kempf

Lieber nicht.

-

Dejan Bogdanovic

Mischa Binder

Simon Weibel

Mario Burger

Sascha Häring

Dominik Rügge

Stefanie Jäger

Jan Preisig

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. [Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Google

Formulare

A.3. Engagement Test

Results

You can interpret the means of the scales. The UEQ does not produce an overall score for the user experience (you need to use the KPI extension for this). Because of the construction of the questionnaire it does make no sense to built such an overall score (for example by calculating the mean over all scales), since this value can not be interpreted properly. The values for the single items are listed to allow you to detect outliers in the evaluations. If an item shows big deviations to the evaluations of the other items of the same scale this can be a hint that the item is misinterpreted (for example, because of a special context in your evaluation) by a higher number of participants.

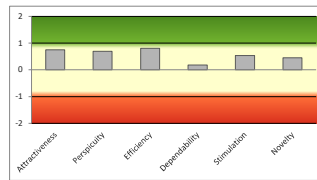
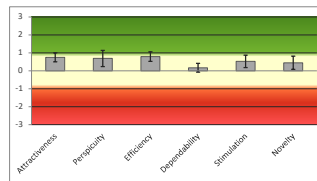
Values between -0.8 and 0.8 represent a more or less *neutral evaluation* of the corresponding scale, values > 0.8 represent a *positive evaluation* and values < -0.8 represent a *negative evaluation*.

The range of the scales is between -3 (horribly bad) and +3 (extremely good). But in real applications, in general, only values in a restricted range will be observed. It is due to the calculation of means over a range of different persons with different opinions and answer tendencies (for example the avoidance of extreme answer categories) extremely unlikely to observe values above +2 or below -2.

Thus, even a quite good value of +1.5 for a scale looks from the purely visual standpoint on a scale range of -3 to +3 not as positive as it really is. For this reason this sheet contains two variants for the figure that depicts the scale means. Use the figure with the reduced scale -2 to +2 if you communicate the results to persons that have not much knowledge on the interpretation of this type of data and in situations where you don't want to explain in detail how building mean values and answer tendencies influence the observed data.

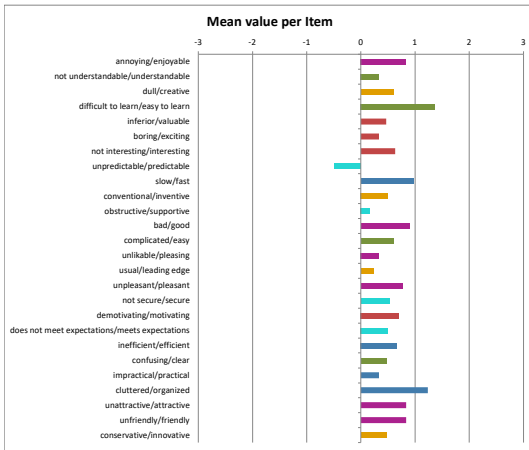
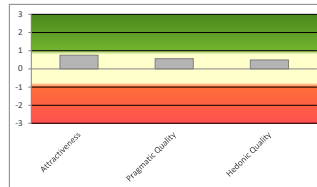
Item	Mean	Variance	Std. Dev.	No.	Left	Right	Scale
1	0.8	1.4	1.2	30	annoying	enjoyable	Attractiveness
2	0.3	2.5	1.6	30	not understandable	understandable	Perspicuity
3	0.6	1.1	1.1	30	creative	dull	Novelty
4	1.4	1.9	1.4	30	easy to learn	difficult to learn	Perspicuity
5	0.5	1.0	1.0	30	valuable	inferior	Stimulation
6	0.3	1.5	1.2	30	boring	exciting	Stimulation
7	0.6	1.6	1.2	30	not interesting	interesting	Stimulation
8	-0.5	1.4	1.2	30	unpredictable	predictable	Dependability
9	1.0	1.3	1.2	30	fast	slow	Efficiency
10	0.5	1.9	1.4	30	inventive	conventional	Novelty
11	0.2	1.1	1.1	30	obstructive	supportive	Dependability
12	0.9	1.0	1.0	30	good	bad	Attractiveness
13	0.5	2.0	1.4	30	complicated	easy	Perspicuity
14	0.3	0.9	0.9	30	unlikable	pleasing	Attractiveness
15	0.2	1.8	1.4	30	usual	leading edge	Novelty
16	0.8	0.7	0.8	30	unpleasant	pleasant	Attractiveness
17	0.5	0.7	0.9	30	secure	not secure	Dependability
18	0.7	1.3	1.1	30	demotivating	motivating	Stimulation
19	0.5	1.4	1.2	30	does not meet expectations	meets expectations	Dependability
20	0.7	1.3	1.2	30	inefficient	efficient	Efficiency
21	0.5	2.8	1.7	30	confusing	clear	Perspicuity
22	0.3	1.0	1.0	30	impractical	practical	Efficiency
23	1.2	1.7	1.3	30	organized	cluttered	Efficiency
24	0.8	0.8	0.9	30	attractive	unattractive	Attractiveness
25	0.8	1.5	1.2	30	friendly	unfriendly	Attractiveness
26	0.5	1.6	1.3	30	conservative	innovative	Novelty

UEQ Scales (Mean and Variance)		
Attractiveness	0.750	0.47
Perspicuity	0.692	1.62
Efficiency	0.800	0.54
Dependability	0.175	0.45
Stimulation	0.533	0.94
Novelty	0.450	1.08



Pragmatic and Hedonic Quality	
Attractiveness	0.75
Pragmatic Quality	0.56
Hedonic Quality	0.49

The scales of the UEQ can be grouped into pragmatic quality (Perspicuity, Efficiency, Dependability) and hedonic quality (Stimulation, Originality). Pragmatic quality describes task related quality aspects, hedonic quality the non-task related quality aspects. Below the mean of the three pragmatic and hedonic quality aspects is calculated.



A.4. API Tests

A.4.1. 125 Benutzer

```

data_received.....: 1.3 MB 70 kB/s
data_sent.....: 314 kB 17 kB/s
group_duration.....: avg=530.66ms min=64.3ms med=253.38ms max=5.1s p(90)=1.99s p(95)=2.32s
http_req_blocked.....: avg=7.26ms min=527ns med=1.06µs max=71.49ms p(90)=28.79ms p(95)=58.8ms
http_req_connecting.....: avg=1.98ms min=0s med=0s max=18.51ms p(90)=8.18ms p(95)=18.1ms
http_req_duration.....: avg=344.64ms min=57.76ms med=234.09ms max=2.67s p(90)=382.54ms p(95)=2.01s
  { expected_response:true }...: avg=344.64ms min=57.76ms med=234.09ms max=2.67s p(90)=382.54ms p(95)=2.01s
http_req_failed.....: 0.00% ✓ 0 ✗ 386
http_req_receiving.....: avg=131.7µs min=58.3µs med=117.2µs max=1.74ms p(90)=175.96µs p(95)=204.09µs
http_req_sending.....: avg=274.11µs min=108.18µs med=211.18µs max=12.18ms p(90)=355.92µs p(95)=444.42µs
http_req_tls_handshaking.....: avg=5.15ms min=0s med=0s max=44.7ms p(90)=20.32ms p(95)=40.38ms
http_req_waiting.....: avg=344.24ms min=57.45ms med=233.67ms max=2.67s p(90)=381.93ms p(95)=2.01s
http_reqs.....: 386 21.332037/s
iteration_duration.....: avg=1.06s min=493.23ms med=611.53ms max=5.1s p(90)=2.44s p(95)=2.52s
iterations.....: 126 6.963307/s
vus.....: 0 min=0 max=50
vus_max.....: 50 min=50 max=50

```

API Test Resultat Zusammenfassug mit 125 Benutzer, 1 Challenge and 1 Reward

```

data_received.....: 1.4 MB 64 kB/s
data_sent.....: 437 kB 19 kB/s
group_duration.....: avg=841.9ms min=62.97ms med=599.54ms max=7.99s p(90)=2.61s p(95)=2.97s
http_req_blocked.....: avg=5.17ms min=397ns med=1.04µs max=238.7ms p(90)=12.03ms p(95)=29.84ms
http_req_connecting.....: avg=1.34ms min=0s med=0s max=42.6ms p(90)=2.12ms p(95)=8.32ms
http_req_duration.....: avg=328.36ms min=62.09ms med=239.77ms max=2.91s p(90)=346.58ms p(95)=1.84s
  { expected_response:true }...: avg=328.36ms min=62.09ms med=239.77ms max=2.91s p(90)=346.58ms p(95)=1.84s
http_req_failed.....: 0.00% ✓ 0 ✗ 641
http_req_receiving.....: avg=138.98µs min=39.35µs med=114.87µs max=3.3ms p(90)=170.68µs p(95)=208.93µs
http_req_sending.....: avg=249.86µs min=71.59µs med=213.14µs max=2.66ms p(90)=392.45µs p(95)=450.95µs
http_req_tls_handshaking.....: avg=3.2ms min=0s med=0s max=41.98ms p(90)=8.73ms p(95)=21.12ms
http_req_waiting.....: avg=327.97ms min=61.76ms med=239.3ms max=2.91s p(90)=345.89ms p(95)=1.84s
http_reqs.....: 641 28.320991/s
iteration_duration.....: avg=1.68s min=922.91ms med=1.15s max=7.99s p(90)=3.06s p(95)=3.36s
iterations.....: 125 5.522814/s
vus.....: 0 min=0 max=50
vus_max.....: 50 min=50 max=50

```

API Test Resultat Zusammenfassung mit 125 Benutzer, 3 Challenges and 3 Rewards

A.4.2. 500 Benutzer

```

data_received.....: 3.1 MB 105 kB/s
data_sent.....: 990 kB 34 kB/s
dropped_iterations.....: 16 0.542286/s
group_duration.....: avg=386.65ms min=55.77ms med=242ms max=6.62s p(90)=598.02ms p(95)=1.93s
http_req_blocked.....: avg=2.07ms min=353ns med=1µs max=74.75ms p(90)=1.36µs p(95)=9.98ms
http_req_connecting.....: avg=586.71µs min=0s med=0s max=33.48ms p(90)=0s p(95)=1.96ms
http_req_duration.....: avg=254.68ms min=54.92ms med=222.51ms max=3.37s p(90)=279.69ms p(95)=360ms
  { expected_response:true }...: avg=254.68ms min=54.92ms med=222.51ms max=3.37s p(90)=279.69ms p(95)=360ms
http_req_failed.....: 0.00% ✓ 0 ✗ 1463
http_req_receiving.....: avg=142.11µs min=31.76µs med=111.21µs max=17.86ms p(90)=162.64µs p(95)=188.96µs
http_req_sending.....: avg=212.07µs min=66.14µs med=188.85µs max=718.26µs p(90)=300.6µs p(95)=362.68µs
http_req_tls_handshaking.....: avg=1.46ms min=0s med=0s max=52.33ms p(90)=0s p(95)=7.68ms
http_req_waiting.....: avg=254.33ms min=54.58ms med=222.19ms max=3.37s p(90)=279.31ms p(95)=359.7ms
http_reqs.....: 1463 49.585275/s
iteration_duration.....: avg=774.27ms min=432.85ms med=562.4ms max=6.62s p(90)=2.24s p(95)=2.41s
iterations.....: 485 16.438044/s
vus.....: 0 min=0 max=50
vus_max.....: 50 min=50 max=50

```

API Test Resultat Zusammenfassung mit 500 Benutzer, 1 Challenge and 1 Reward


```

data_received.....: 3.1 MB 105 kB/s
data_sent.....: 990 kB 34 kB/s
dropped_iterations.....: 16 0.542286/s
group_duration.....: avg=386.65ms min=55.77ms med=242ms max=6.62s p(90)=598.02ms p(95)=1.93s
http_req_blocked.....: avg=2.07ms min=353ns med=1µs max=74.75ms p(90)=1.36µs p(95)=9.98ms
http_req_connecting.....: avg=586.71µs min=0s med=0s max=33.48ms p(90)=0s p(95)=1.96ms
http_req_duration.....: avg=254.68ms min=54.92ms med=222.51ms max=3.37s p(90)=279.69ms p(95)=360ms
{ expected_response:true }...: avg=254.68ms min=54.92ms med=222.51ms max=3.37s p(90)=279.69ms p(95)=360ms
http_req_failed.....: 0.00% ✓ 0 ✗ 1463
http_req_receiving.....: avg=142.11µs min=31.76µs med=111.21µs max=17.86ms p(90)=162.64µs p(95)=188.96µs
http_req_sending.....: avg=212.07µs min=66.14µs med=188.85µs max=718.26µs p(90)=300.6µs p(95)=362.68µs
http_req_tls_handshaking.....: avg=1.46ms min=0s med=0s max=52.33ms p(90)=0s p(95)=7.68ms
http_req_waiting.....: avg=254.33ms min=54.58ms med=222.19ms max=3.37s p(90)=279.31ms p(95)=359.7ms
http_reqs.....: 1463 49.585275/s
iteration_duration.....: avg=774.27ms min=432.85ms med=562.4ms max=6.62s p(90)=2.24s p(95)=2.41s
iterations.....: 485 16.438044/s
vus.....: 0 min=0 max=50
vus_max.....: 50 min=50 max=50

```

API Test Resultat Zusammenfassung mit 500 Benutzer, 3 Challenges and 3 Rewards

A.4.3. 2000 Benutzer

```

data_received.....: 11 MB 114 kB/s
data_sent.....: 3.8 MB 40 kB/s
dropped_iterations.....: 19 0.201312/s
group_duration.....: avg=301.52ms min=55.04ms med=234.65ms max=11.48s p(90)=555.51ms p(95)=606.48ms
http_req_blocked.....: avg=503.37µs min=299ns med=966ns max=96.9ms p(90)=1.1µs p(95)=1.22µs
http_req_connecting.....: avg=140.58µs min=0s med=0s max=18.48ms p(90)=0s p(95)=0s
http_req_duration.....: avg=199.78ms min=54.2ms med=207.66ms max=7.01s p(90)=270.92ms p(95)=296.87ms
{ expected_response:true }...: avg=199.78ms min=54.2ms med=207.66ms max=7.01s p(90)=270.92ms p(95)=296.87ms
http_req_failed.....: 0.00% ✓ 0 ✗ 5954
http_req_receiving.....: avg=125.35µs min=33.32µs med=101.71µs max=20.15ms p(90)=148.8µs p(95)=172.98µs
http_req_sending.....: avg=197.17µs min=65.21µs med=176.34µs max=1.48ms p(90)=269.82µs p(95)=317.08µs
http_req_tls_handshaking.....: avg=355.76µs min=0s med=0s max=78.29ms p(90)=0s p(95)=0s
http_req_waiting.....: avg=199.46ms min=53.96ms med=207.29ms max=7.01s p(90)=270.63ms p(95)=296.55ms
http_reqs.....: 5954 63.084904/s
iteration_duration.....: avg=603.56ms min=436ms med=535.51ms max=11.48s p(90)=643.69ms p(95)=754.98ms
iterations.....: 1982 21.000047/s
vus.....: 0 min=0 max=50
vus_max.....: 50 min=50 max=50

```

API Test Resultat Zusammenfassung mit 2000 Benutzer, 1 Challenge and 1 Reward

```

data_received.....: 13 MB 122 kB/s
data_sent.....: 5.6 MB 51 kB/s
dropped_iterations.....: 31 0.285549/s
group_duration.....: avg=520.49ms min=51.9ms med=469.01ms max=24.33s p(90)=997.4ms p(95)=1.09s
http_req_blocked.....: avg=278.9µs min=303ns med=988ns max=68.11ms p(90)=1.16µs p(95)=1.29µs
http_req_connecting.....: avg=78.16µs min=0s med=0s max=19.08ms p(90)=0s p(95)=0s
http_req_duration.....: avg=207.07ms min=51.23ms med=203.79ms max=7.24s p(90)=269.49ms p(95)=308.75ms
{ expected_response:true }...: avg=207.07ms min=51.23ms med=203.79ms max=7.24s p(90)=269.49ms p(95)=308.75ms
http_req_failed.....: 0.00% ✓ 0 ✗ 9866
http_req_receiving.....: avg=154.64µs min=38.12µs med=111.63µs max=29.81ms p(90)=170.61µs p(95)=212.38µs
http_req_sending.....: avg=204.09µs min=63.06µs med=177.79µs max=7.94ms p(90)=288.31µs p(95)=353.35µs
http_req_tls_handshaking.....: avg=195.88µs min=0s med=0s max=49.66ms p(90)=0s p(95)=0s
http_req_waiting.....: avg=206.71ms min=50.99ms med=203.43ms max=7.24s p(90)=268.99ms p(95)=308.43ms
http_reqs.....: 9866 90.878288/s
iteration_duration.....: avg=1.04s min=736.55ms med=958.74ms max=24.33s p(90)=1.16s p(95)=1.29s
iterations.....: 1970 18.146182/s
vus.....: 0 min=0 max=50
vus_max.....: 50 min=50 max=50

```

API Test Resultat Zusammenfassung mit 2000 Benutzer, 3 Challenges and 3 Rewards

A.5. Kostenanalyse

Konfiguration		Test 1 - 09.05.22 0700-0800			Test 2 - 09.05.22 2300-2400		
Users							
Challenges							
AWS Services							
Name	Position	Unit	Snapshot before	Snapshot after	Unit expenditures	Snapshot before	Snapshot after
AWS Amplify	DataStorage usage in EU (Frankfurt)	Gigabytes	0,41	0,41	0,00	0,405	0,45
AWS Amplify	DataTransferOut usage in EU (Frankfurt)	Gigabytes	1,84	1,88	0,00	2,778	2,93
AWS AppSync	ConnectionDuration in EU (Frankfurt)	Minutes	4024,83	4260,64	4,72	4334,692	5017,95
AWS AppSync	GraphQLInvocation Query and data modification operations in EU (Frankfurt)	Requests	11828,00	12265,00	8,74	16416,000	18820,00
AWS AppSync	GraphQLNotification Real-time updates in EU (Frankfurt)	Messages	6644,00	6931,00	5,74	9783,000	11381,00
AWS CloudWatch	Requests	Requests	3,00	3,00	0,00	3,000	3,00
AWS CloudWatch	Metrics requested using GetMetricData API - EU (Frankfurt)	Metrics	33,00	33,00	0,00	33,000	33,00
AWS CloudWatch	TimedStorage-BytesHrs	GB-Mo	0,00	0,00	0,00	0,000	0,00
AWS CloudWatch	PutLogEvents: Log data ingested	GB	0,00	0,00	0,00	0,004	0,01
AWS CloudWatch	StartQuery: Log data scanned by CloudWatch Logs insights queries	GB	0,45	0,45	0,00	0,449	0,47
AWS Cognito	Amazon Cognito CognitoUserPoolsOperation	CognitoUserPoolMAU	338,00	344,00	0,12	438,000	480,00
AWS Data Transfer	AWS Data Transfer EUCL-USE1-AWS-In-Bytes	GB	0,00	0,00	0,00	0,000	0,00
AWS Data Transfer	AWS Data Transfer EUCL-USE1-AWS-Que-Bytes	GB	0,00	0,00	0,00	0,000	0,00
AWS Data Transfer	Bandwidth Data transfer in	GB	7,00	7,06	0,00	7,058	7,08
AWS Data Transfer	Bandwidth Data transfer out	GB	2,07	2,17	0,00	2,439	2,86
AWS DynamoDB	Amazon DynamoDB EUCL-TimedStorage-BytesHrs	GB-Mo	0,00	0,00	0,00	0,000	0,00
AWS DynamoDB	Read Request Units (RRU)	ReadRequestUnits	26740,00	27452,00	14,24	37716,500	42262,00
AWS DynamoDB	Write Request Units (WRU)	WriteRequestUnits	8420,00	8964,00	10,88	11888,000	13083,00
Key Management Service (KMS)	Requests	Requests	203,00	209,00	0,12	386,000	474,00
AWS Lambda	Compute	Lambda-GB-Second	232,37	230,88	0,21	296,139	322,25
AWS Lambda	Requests	Request	2784,00	2882,00	1,96	3848,000	4385,00
AWS Route 53	DNS-Queries	Queries	1192,00	1229,00	0,74	1304,000	1506,00
AWS Route 53	HostedZone	HostedZone	1,00	1,00	0,00	1,000	1,00
AWS Route 53	Intra-AWS DNS-Queries	Queries	303,00	305,00	0,04	313,000	361,00
Simple Email Service (SES)	SendEmail or SendRawEmail EC2 EUW1	Count	109,00	112,00	0,06	204,000	239,00
AWS Simple Notification Service (SNS)	Amazon SNS API Requests	Requests	6,00	6,00	0,00	4,000	6,00
AWS Simple Storage Service (S3)	PUT, COPY, POST, or LIST requests	Requests	23257,00	23810,00	11,06	23810,000	25389,00
AWS Simple Storage Service (S3)	GET and all other requests	Requests	19084,00	18988,00	16,08	19846,000	20845,00
AWS Simple Storage Service (S3)	TimedStorage-BytesHrs	GB-Mo	0,58	0,58	0,00	0,720	0,72

A.6. Code reviews

Für die Code Reviews wurden zwei Termine mit Michael Gfeller festgelegt. An diesen Tagen setzte er sich mit einem Fokus auf das Frontend mit dem Code auseinander und hielt seine Bemerkungen in Form von Notizen fest. Diese wurden anschliessend mit dem Projektteam besprochen und ihnen übergeben, damit sie für allfällige Verbesserungen genutzt werden konnten.

A.6.1. 12.04.2022 - Michael Gfeller

Fragen	Antwort
<p>Gewisse CSS Klassen wurden platziert ohne das diese genutzt werden z.B. open-challenges / closed-challenges. Gibt es dafür einen Grund? → Ev. wäre auch eine weitere Komponente sinnvoll.</p>	<p>Ist für die Weiterentwicklung angenehm und wurde beim stylen auch verwendet.</p>
<p>Ihr habt kein spezifisches State Management eingesetzt. z.B. <i>Redux</i> / <i>Mobx</i> / ... Wie habt ihr dies entschlossen?</p>	<p>Das State Management ist im Kapitel 5.1.2 beschrieben und begründet.</p>
<p>Ihr habt sehr viele h1-h3 verwendet. Aus meiner Sicht zu viele. z.B. Players.tsx: <h2>Wer schießt das erste Tor?</h2> <h2 className='divider'>Goalkeeper</h2></p>	<p>Die hier erwähnten h1-h3 wurden mit div Elementen ersetzt. Generell werden noch weitere Titel reduziert.</p>
<p>Ihr habt semantic-ui-react eingebunden aber nutzt es (fast) nur für Icons und erstellt eigene Komponente. Begründung? → Aufpassen, dass die erwartete Funktionalitäten vorhanden sind</p>	<p>Neben den Icons wird Semantic UI auch für Formulare, Messages, Tables, Modals und dem Loader verwendet</p>

Tabelle 80: Code Review 1 Fragen und Antworten

Feedback	Massnahmen
<p>(Teilweise) Die Root Readmes sind leer oder beinhalten nichts sinnvolles.</p>	<p>Readmes wurden mit sinnvollen Informationen ergänzt</p>
<p>Wie bringe ich die Entwicklungsumgebung zum laufen: → npm install & npm start hat funktioniert.</p>	<p>Readmes beinhalten Abschnitt mit lokaler Deploymentanleitung</p>
<p>Ich hab die App über https://dev.d39wkpnt1m68yq.amplifyapp.com/ ausprobiert. → Fokus auf Frontend Code</p>	<p><i>Ist eine Anmerkung</i></p>

<p>Viel Logik in den Komponenten z.B. Game.tsx → Filtern von Challenges → Laden der Daten anhand vom State z.B. Players.tsx → Das Handling von Usern wird dieser Komponente überlassen. z.B. AdminTheme.tsx → user.signInUserSession.accessToken.payload ['cognito:groups']?.includes('Admin')</p>	<p>Die Logik in Komponenten wurden best Möglich in Hilfsfunktionen ausgelagert</p>
<p>Keine Tests vorhanden → Wichtig um zu Beweisen ob die Architektur sinnvoll gewählt wurde.</p>	<p>Ausgelagerte Logik, sowie Actions wurden mit Unit Tests getestet.</p>
<p>Auf Details achten → Namen von Dateien und Ordner sind nicht konsistent. → File (z.B. gameActions.ts) ist komisch formatiert.</p>	<p>Nicht konsistente Namen wurden angepasst.</p>
<p>type: 'FETCH_ACTIVE_GAME_SUCCEEDED'. → String-Constanten im Code verstreut.</p>	<p>String-Konstanten wurden in den Enum Typ ActionTypes ausgelagert</p>
<p>if (currentTip?.answer === `\${player.firstname} `\${player.lastname}`) → Das fühlt sich sehr falsch an wenn man Namen als Key braucht.</p>	<p>Zukünftig wird die Player Id in diesem Feld gesetzt anstelle des Namens. Das war geplant wurde bisher jedoch nicht umgesetzt.</p>
<p> → Komponente zum extrahieren. Alt Text ist falsch.</p>	<p>Der Sponsor wird in einem geplanten Feature dynamisch festgelegt. Entsprechend wird auch der Alt Text gesetzt.</p>
<p>Auth.configure() → Diese Settings sollten ausgelagert werden.</p>	<p>Settings wurden ausgelagert.</p>
<p>page: any, ...rest: any action?: any; → any sollten vermieden werden. Eslint etwas zu stark reduziert.</p>	<p>Bei GraphQL Results konnten Typen definiert werden.</p>
<p>Copy and Paste → Auth.currentAuthenticatedUser()</p>	<p>Wurde in eine eigene Action ausgelagert.</p>
<p>Unused Code → challengeTipReducer</p>	<p>Code wird nun gebraucht (war nur zum Stand der Entwicklung)</p>
<p>reducers / actions → Aktuell beinhaltet die Action einfach alle Properties für alle möglichen Actions. Es gibt bessere möglichkeiten z.B. https://www.sumologic.com/blog/react-hook-typescript/ https://medium.com/codex/typescript-and-react-usereducer-943e4f8d1ad4</p>	<p>Die Möglichkeiten wurden in Betracht bezogen. Es gibt nun den typ AppStateAction.</p>

Der Slider funktioniert https://dev.d39wkpnt1m68yq. amplifyapp.com/rewards/redeem nur mit Touch	Vorsichtig funktioniert der Slider auch am Computer. Zudem wird ein Preis praktisch jedem Fall über das Mobile eingelöst. Das Problem ist uns bekannt und wird tief priorisiert angegangen.
"Warum isch Amplify so scheisse" kann ich verstehen aber schlecht für Demo Zwecke :)	Unangebrachte Datenfelder wurden entfernt.
"Zurzeit gibt es keine aktive Challenge" wird dargestellt obwohl die Daten noch geladen werden. → Spinner oder ähnliches darstellen und nicht den Fallback Text.	Loader wurden eingebaut
Tests für die Funktions definiert	War ein positiver Aspekt
Finde den Import schöner als über die "globalen" Objekte React.Dispatch<IAppStateAction> → Dispatch<IAppStateAction>	Wurde so umgesetzt weil wir dem zustimmen.
import { Dispatch } from 'react';	

Tabelle 81: Code Review 1 Feedback und Massnahmen

Fazit

Ich fand mich im Code gut zu recht. Code ist in Ordnung hat aber Verbesserungspotential

A.6.2. 02.05.2022 - Michael Gfeller

Feedback	Massnahmen
Viele Punkte behoben z.B. Strings bei Actions / AppStateAction / isAdmin / .env / ...	Feedback im Vergleich zu Code Review 1
Schöne Dialoge	Feedback im Vergleich zu Code Review 1
Eigene Hooks können den Code stark verbessern. Siehe Copy and Paste	Feedback im Vergleich zu Code Review 1
Ich habe keine NOGOs gefunden. Stellt sicher das die UX stimmt und gut bedienbar ist. Es gab doch einige komische Phänomene.	Feedback im Vergleich zu Code Review 1
setLoading(true); Tipp: Hooks definieren	Custom Hook wurde integriert und in Kapitel 5.1.5 beschrieben.
dispatch({ type: ActionTypes.SET_ERROR, error: err }); const [error, setError] = useState<string undefined>(undefined);	Custom Hook wurde integriert und in Kapitel 5.1.5 beschrieben.
Copy und Paste im CSS: → z.B. 400px / border-radius: 8px; / margin-left: 16px → Nicht erste Prio → Aufgrund von der Menge von CSS. Wie stellt ihr sicher das der CSS-Code wartbar bleibt?	Variablen für Schriftgrößen, Abständen und Breakpoints wurden in _variables.scss definiert. Mixins für Media Queries und Browser-spezifische Präfixe für Transitions wurden in _mixins.scss definiert.

<pre>useEffect(() => {}, [state.user]);</pre> <p>→ Warum?</p>	<p>Für API Requests ist ein JWT Token nötig, welcher aus state.user gelesen wird. Durch ein Refactoring am User Management konnte diese Dependency an vielen Orten entfernt werden.</p>
<p>Kommentare wenn nötig z.B. Weshalb nötig? Und weshalb setzt setMigrationFlag nur ein Token.</p> <pre>if (user?.pool.userPoolId === "eu-central-1_FllUdSaCo") { setMigrationFlag(user.signInUserSession. accessToken.jwtToken);}</pre>	<p>Bei einer Anmeldung kann dem Backend der Token des anonymen Users übergeben werden. Wenn der Token vorhanden ist, werden die anonym abgegebenen Tipps dem neu eingeloggtem User übertragen. Das Flag wird im LocalStorage abgelegt für den Fall, dass sich der Benutzer über einen SocialProvider anmeldet. Bei diesem Prozess wird zuerst der anonyme User abgemeldet und danach durch den Social Provider ein Page Reload erzwingt. Ohne dem LocalStorage hätte das Frontend keinen Zugriff mehr auf den Token.</p>
<p>Promise<any> noch 3 vorhanden</p>	<p>Typ wurde definiert.</p>
<p>Error-Handling</p> <p>→ Sinnvolle Fehler dem Benutzer darstellen</p> <p>→ AppState: Error undefined wird aktuell nicht angezeigt</p> <p>→ Fehler an den Server senden, um Client Fehler zu erhalten</p>	<p>Error Handling wurde implementiert und ist im Kapitel 5.1.7 beschrieben</p>
<p>Testing</p> <p>→ Snapshot Tests:</p> <p>https://jestjs.io/docs/snapshot-testing</p> <p>→ Sinnvoll da viel CSS programmiert habt</p>	<p>Snapshot Tests wurden angeschaut, konnte aber wegen dem zeitlichem Rahmen nicht umgesetzt werden.</p>
<p>Bugs:</p> <p>→ Ab und zu werden die Items (z.B. Sponsoren / Preise) angezeigt und ab und zu nicht. Erfassen ging nicht.</p> <p>→ Benutzername geht bei mir jedes Mal verloren und muss ich neu eingeben.</p> <p>→ Bei Zahlenfelder können auch andere Buchstaben eingegeben werden.</p>	<p>Bugs wurden behoben.</p>
<p>React 18.0 ist verfügbar</p>	<p>Das Projektteam hat ein Update in Betracht gezogen.</p>
<p>Google/FB/Apple Login funktionieren (aktuell?) nicht.</p>	<p>Die angebotenen Social Provider waren noch nicht komplett implementiert.</p>
<p>Grossteil vom UI konnte ich nicht ausprobieren (Preise / Tippspiel / Rangliste funktionieren nicht so richtig)</p>	<p>Durch die Umstellung des AWS-Accounts hatten sich einige Bugs eingeschlichen. → Wurden behoben</p>
<p>Seiten Refresh loggt den Benutzer aus.</p>	<p>Bug wurde behoben</p>
<p>Console-Output Errors, davon einige "gefährlich".</p>	<p>Consolen Outputs wurden aus dem Code entfernt.</p>

<pre>const position = location.pathname. lastIndexOf('/'); navigate(location.pathname.substring(0, posi- on)); → Interessante Lösung. Auch möglich: navigate(-1)</pre>	Die Zurück-Navigation wurde überarbeitet, aber im Ansatz so gelassen.
--	---

Ihr nutzt die Subscriptions von <i>GraphQL</i> aber keine Push Nachrichten. Wäre für diese Art von App super: https://docs.aws.amazon.com/sns/latest/dg/sns-mobile-application-as-subscriber.html	Push-Nachrichten ist ein Feature mit weniger hoher Priorität als andere.
--	--

Tabelle 82: Code Review 2 Feedback und Massnahmen

A.7. System Test 15.06.2022

system_test.md

6/15/2022

System Test Protokoll

Vorbereitung

- Aktuellste Versionen der Repositories sind deployed.
- Ein Account mit Administrator Rechten ist bekannt.
- Die Startseite ist auf einem Smartphone geöffnet und das Admin-Portal an einem Desktop.Client. In beiden Browsern sind sowohl Cache als auch Local Storage gelöscht.

Allgemein

Admin-Portal

- Lakers-Logo wird dargestellt.
- Login-Formular wird dargestellt.
- Mit korrekten Anmeldeinformationen führt der Anmelde Button zur Admin-Übersicht. Ansonsten wird ein Fehler dargestellt.

Start-Seite

- Lakers Logo wird dargestellt.
- Hintergrundbild wird dargestellt.
- Fragezeichen und Benutzer Symbol werden im Header dargestellt.
- Preise, Tippspiel und Rangliste werden im Footer dargestellt.
- Bei ungültigen URLs wird die PageNotFound-Seite dargestellt.

Admin-Panel Konfiguration

Challenges

- Eine Übersicht mit den vorhandenen Challenges wird dargestellt.
- Existierende Challenges können bearbeitet und gelöscht (archiviert) werden.
- Weitere Challenges können erfasst werden.

Sponsoren

- Eine Übersicht mit den vorhandenen Sponsoren wird dargestellt.
- Existierende Sponsoren können bearbeitet und gelöscht werden.
- Weitere Sponsoren können erfasst werden.

Preise

- Eine Übersicht mit den vorhandenen Preisen wird dargestellt.
- Existierende Preise können bearbeitet und gelöscht (archiviert) werden.
- Weitere Preise können erfasst werden.

Kader

- Eine Übersicht mit den vorhandenen Spielern wird dargestellt.
- Existierende Spieler können bearbeitet und gelöscht werden.
- Weitere Spieler können erfasst werden.

Tippspiel Vorbereitung

Tippspiel erstellen

- Falls noch kein Tippspiel vorhanden ist, kann über den Button Spiel erstellen eines erstellt werden.

Beschreibung

- Das Textfeld für die Beschreibung des Tippspieles wird angezeigt.
- Mit dem Button Speichern wird die Eingabe übernommen

Challenges

- Der Button Challenges bearbeiten wird dargestellt.
- Die im Menüpunkt Challenges existierenden Fragen werden mit den zu vergebenen Punkten dargestellt.
- Mit der Checkbox können Challenges selektiert und über den Button Speichern dem Tippspiel hinzugefügt werden.
- Bereits benutzte und zudem aufgelöste Challenges sind ausgegraut und können nicht mehr hinzugefügt/entfernt werden.
- Neu hinzugefügte Challenges haben den Status versteckt.
- Der Button Abbrechen macht die Änderungen rückgängig.

Sponsoren

- Der Button Sponsor bearbeiten wird dargestellt.
- Die im Menüpunkt Sponsoren existierenden Einträge werden dargestellt.
- Ein Sponsor kann per Klick selektiert werden und wird hervorgehoben. Bei einem weiteren Klick auf den Sponsor, wird die Auswahl gelöscht.
- Der Button Speichern übernimmt die Änderungen.
- Es kann nur ein Sponsor ausgewählt sein.
- Der Button Abbrechen macht die Änderungen rückgängig.

Preise

- Der Button Preise bearbeiten wird dargestellt.
- Die im Menüpunkt Preise existierenden Gewinne werden falls vorhanden mit ihrem Wert dargestellt.
- Mit der Checkbox können Preise selektiert und über den Button Speichern dem Tippspiel hinzugefügt werden.
- Die Preise sind nach Reihenfolge des Auswählens priorisiert.
- Ausgewählte Preise haben eine Stückzahl, welche mit einem Plus, respektiv Minus Button bearbeitet werden kann
- Der Button Abbrechen macht die Änderungen rückgängig.

Aufgebot

- Der Button Aufgebot bearbeiten wird dargestellt.
- Die im Menüpunkt Kader existierenden Spieler werden dargestellt.
- Ein Spieler kann per Klick selektiert werden und wird hervorgehoben. Bei einem weiteren Klick auf den Spieler, wird die Auswahl gelöscht.
- Der Button Speichern übernimmt die Änderungen.
- Der Button Abbrechen macht die Änderungen rückgängig.

Tippspiel starten

- Es wird darauf hingewiesen, wenn die Konfiguration des Sponsors und der Preise fehlt.
- Es wird gefragt, ob das Aufgebot korrekt ist und es nicht mehr bearbeitet werden kann. Ohne Aufgebot lässt sich das Tippspiel nicht starten.
- Geräte, welche die Push-Notification aktiviert haben bekommen eine Benachrichtigung, dass ein Tippspiel gestartet wurde.

Tippspiel Admin-Bedienung

- Das Kader kann nicht mehr angepasst werden. Challenges, Sponsor und Preise schon.

Challenge hinzufügen

- Challenges können wie in der Konfiguration hinzugefügt/entfernt werden.

Challenge starten

- Die Frage der Challenge wird dargestellt.
- Mit dem Button Starten kann eine versteckte Challenge veröffentlicht werden. Sie hat dann den Status Offen.

Challenge sperren

- Die Frage der Challenge wird dargestellt.
- Mit dem Button Schliessen kann eine offene Challenge gesperrt werden. Sie hat dann den Status Geschlossen.

Challenge auflösen

- Die Frage der Challenge wird dargestellt.
- Mit dem Button Auflösen kann das Resultat einer geschlossenen Challenge angegeben werden. Sie hat dann den Status Aufgelöst.

Tippspiel beenden

- Der Button Spiel beenden wird dargestellt.
- Es gibt einen Hinweis, falls es noch Challenges gibt die selektiert, aber nicht aufgelöst sind.
- Das beenden eines Tippspieles muss bestätigt werden mit dem Hinweis, dass die Preise dann verlost werden.

About Seite

- Ein Q&A wird dargestellt.
- Weiterführende Links zu Impressum, AGB, Datenschutzerklärung, Entwicklung und Kontakt werden dargestellt.

Weiterführende Links

- Beim Klick auf einen weiterführenden Link wird die entsprechende Page mit ihrem Inhalt dargestellt.

Preise anzeigen

- Dem eingeloggtem Benutzer werden sowohl nicht abgeholte, sowie eingelöste Preise dargestellt.
- Dem anonymen Benutzer wird der Hinweis dargestellt, dass er angemeldet sein muss, um Preise gewinnen zu können.

Preis Detail

- Offene Preise können mit einem Slider eingelöst werden.
- Eingelöste Preise sind mit ihrem Einlösedatum versehen.

Rangliste anzeigen

- Rangliste mit eigenem Profil wird angezeigt.
- Falls nicht eingeloggt steht kein Profil.

Profilübersicht

- Der Button Einstellungen und Abmelden wird dargestellt.
- Eingeloggten Benutzern werden ihre Punkte, Rang, Anzahl beteiligte Tippspiele, Anzahl abgegebene Tipps und die Prozentzahl der richtigen Tipps dargestellt.

Abmelden

- Der Button Abmelden loggt den Benutzer aus.

Einstellungen

- Der Button Einstellungen öffnet eine neue Seite, auf der das Profil bearbeitet und geändert werden kann.

Tippspiel Teilnahme

- Beim erstmaligem Betreten des Tippspieles wird die Frage gestellt, ob das App Pushnachrichten senden darf. Dies ist jedoch nur in den gängigsten Browsern der Fall (Safari ausgeschlossen).
- Das Tippspiel und dessen Challenges wird durch live updates auf dem aktuellem Stand gehalten.

Challenge Detail

- Die Frage und Beschreibung der Challenge wird dargestellt.
- Ein Feld für die Antwortmöglichkeit wird dargestellt.

Tippabgabe

- Bei Tippabgabe wird der Tipp gespeichert.
- Falls ein anonymer Benutzer einen Tipp abgibt wird der Migrationstoken im Local Storage gespeichert.

B. Projektplan

B.1. Einführung

B.1.1. Zweck

Der Projektplan bietet einen Überblick über die Bachelorarbeit mit dem "Thema SCRJ Lakers Fan Engagement App". Er beinhaltet Informationen zum zeitlichen Ablauf der Arbeit, der Organisation, den Risiken und dem Qualitätsmanagement. Mit ihm wird die Basis für eine erfolgreiche Projektabwicklung gelegt.

B.1.2. Gültigkeitsbereich

Der Gültigkeitsbereich erstreckt sich über die ganze Projektdauer innerhalb des Frühjahrssemesters 2022. Das Dokument wird laufend mit neuen Informationen ergänzt und im Versionsverlauf festgehalten.

B.1.3. Produkt

- Anforderungsspezifikation
- Domainanalyse
- Testspezifikation und Testprotokolle
- Wireframes und Design-Dokumentation
- Softwarearchitektur
- Deployment-Dokumentation
- Quellcode

B.2. Projektorganisation

Alle wichtigen Entscheidungen werden gemeinsam im Projektteam getroffen. Aus diesem Grund wird auf eine flache Hierarchie gesetzt, bei welcher alle Mitglieder gleichgestellt sind und sich gleichermaßen am Projekt beteiligen. Prof. Dr. Markus Stolze steht als Betreuer zur Seite und nimmt die Bewertung vor.

B.2.1. Verwendung von Tools

Im Verlauf des Projektes werden diverse Tools für die Erfüllung von Aufgaben eingesetzt. Einige davon sind dem Projektteam bereits von bisherigen Arbeiten bekannt, andere benötigen für die erfolgreiche Verwendung eine Einarbeitungszeit.

Tool	Bezug/Zweck
Heimdall	Heimdall liefert eine Übersicht der wichtigsten online verfügbaren Dienste innerhalb dieses Projektes.
GitLab	Codespeicher- und versionierung sowie Projektmonitoring mit Zeiterfassung
Miro	Visualisierung des Storyboards
Visual Paradigm	Zeichnen von Modellen
Diagrams.net	Zeichnen von Visualisierungen zu Unterstützung des Verständnisses
SonarQube	SonarQube führt statische Analysen durch und bewertet die technische Qualität von Code. Dafür analysiert es den Code auf verschiedene Qualitätsmerkmale und bereitet die Resultate visuell auf.
Uptime	Uptime Kuma ist ein sehr schlankes Monitoring Tool für die Überwachung der Verfügbarkeiten externer Schnittstellen und eigener Umgebung. Es bietet eine übersichtliche Statusseite mit sämtlichen registrierten Endpunkten.

Google Lighthouse	Kontroller der Accessibility
Figma	Entwicklung des Prototypen
Grafana	Visualisierung von Performance-/Lasttests
Moon Modeler	Visualisierung von Datenbankschema

Tabelle 83: Tooling/Utils

Der konkrete Einsatz gewisser aufgelisteter Tools wird in der Dokumentation genauer beschrieben.

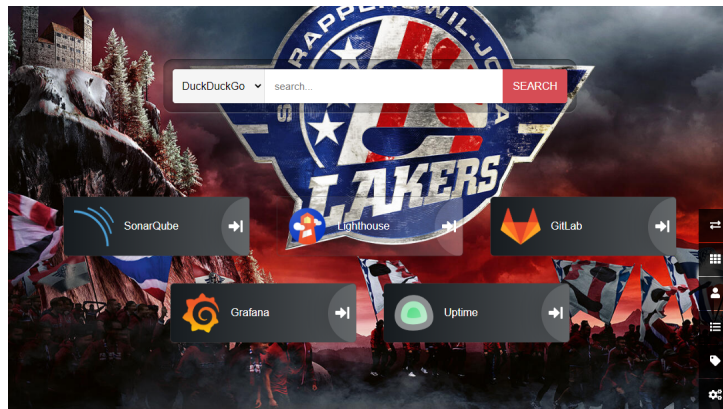


Abbildung 62: Heimdall Dashboard

B.2.2. Organisationsstruktur

Name	E-Mail	Rolle
Simon Hager	simon.hager@ost.ch	Teammitglied
Philipp Emmenegger	philipp.emmenegger@ost.ch	Teammitglied
Joel Schaltegger	joel.schaltegger@ost.ch	Teammitglied
Prof. Dr. Markus Stolze	markus.stolze@ost.ch	Betreuer / Product Owner

Tabelle 84: Organisationsstruktur

B.2.3. Externe Schnittstellen

Name	E-Mail	Rolle
Michael Bart	michael.bart@lakers.ch	Ansprechperson Lakers Sport AG
Rene Schmid	rene.schmid@lakers.ch	IT-Verantwortlicher Lakers Sport AG
Timo Schaffner	tschaffner@detail.ch	Kontakt Hosting/Webseite lakers.ch
Alain Habegger	alain@marketingmonkeys.ch	Kontakt Newsletter-Verwaltung
Werner Schuller	werner.schuller@sihf.ch	IT-Verantwortlicher Verband

Tabelle 85: Externe Schnittstellen

B.3. Zeitmanagement

B.3.1. Zeitraum

Der offizielle Projektstart war am 21. Februar 2022 und endet am 17. Juni 2022 um 17 Uhr mit der Abgabe der Bachelorarbeit.

Projektdauer	15 Wochen
Projektstart	Montag, 21. Februar 2022
Projektende	Freitag, 17. Juni 2022 17 Uhr

Tabelle 86: Zeitraum

B.3.2. Zeitbudget

Jedes Teammitglied ist mit insgesamt 360 Arbeitsstunden gleichermassen an der erfolgreichen Entwicklung des Produktes beteiligt. Insgesamt stehen dem Projekt somit 1080 Stunden zur Verfügung.

Anzahl Projektmitarbeiter	3 Personen
Arbeitsstunden pro Woche und Person	24h
Arbeitsstunden insgesamt	1080h

Tabelle 87: Zeitbudget

B.3.3. Zeitplanung

Die Zeitplanung und die Arbeitspakete werden direkt im GitLab verwaltet. Sie unterliegt einem agilen Projektmanagement und wird kontinuierlich auf neue Anforderungen und Erkenntnisse angepasst. Veranschlagte und effektive Zeitaufwände jedes Arbeitspaketes werden auf 15 Minuten genau im GitLab erfasst.

In diesem Projekt wird Scrum+ eingesetzt, eine Kombination aus Rational Unified Process (RUP) und Scrum. Mit RUP wird das Projekt zeitlich grob in die Phasen Inception, Elaboration, Construction und Transition unterteilt. Innerhalb dieser Phasen kommt das agilere Projektmanagement Scrum zum Einsatz. Der Projektzeitraum wird in zweiwöchige Sprints unterteilt. In diesem Projekt wird jeder Sprint mit einem Meilenstein und anschliessendem Review abgeschlossen. Das Projektteam macht vor jedem neuen Sprint ein Sprint Planning und führt nach jedem abgeschlossenen Sprint ein internes Sprint Review durch.

Phase	Start	Ende
Inception	21.02.2022 (SW01)	27.02.2022 (SW01)
Elaboration	28.02.2022 (SW02)	13.03.2022 (SW03)
Construction	14.03.2022 (SW04)	22.05.2022 (SW13)
Transition	23.05.2022 (SW14)	12.06.2022 (SW16)
Reserve	13.06.2022 (SW17)	17.06.2022 (SW17)

Tabelle 88: Projektphasen

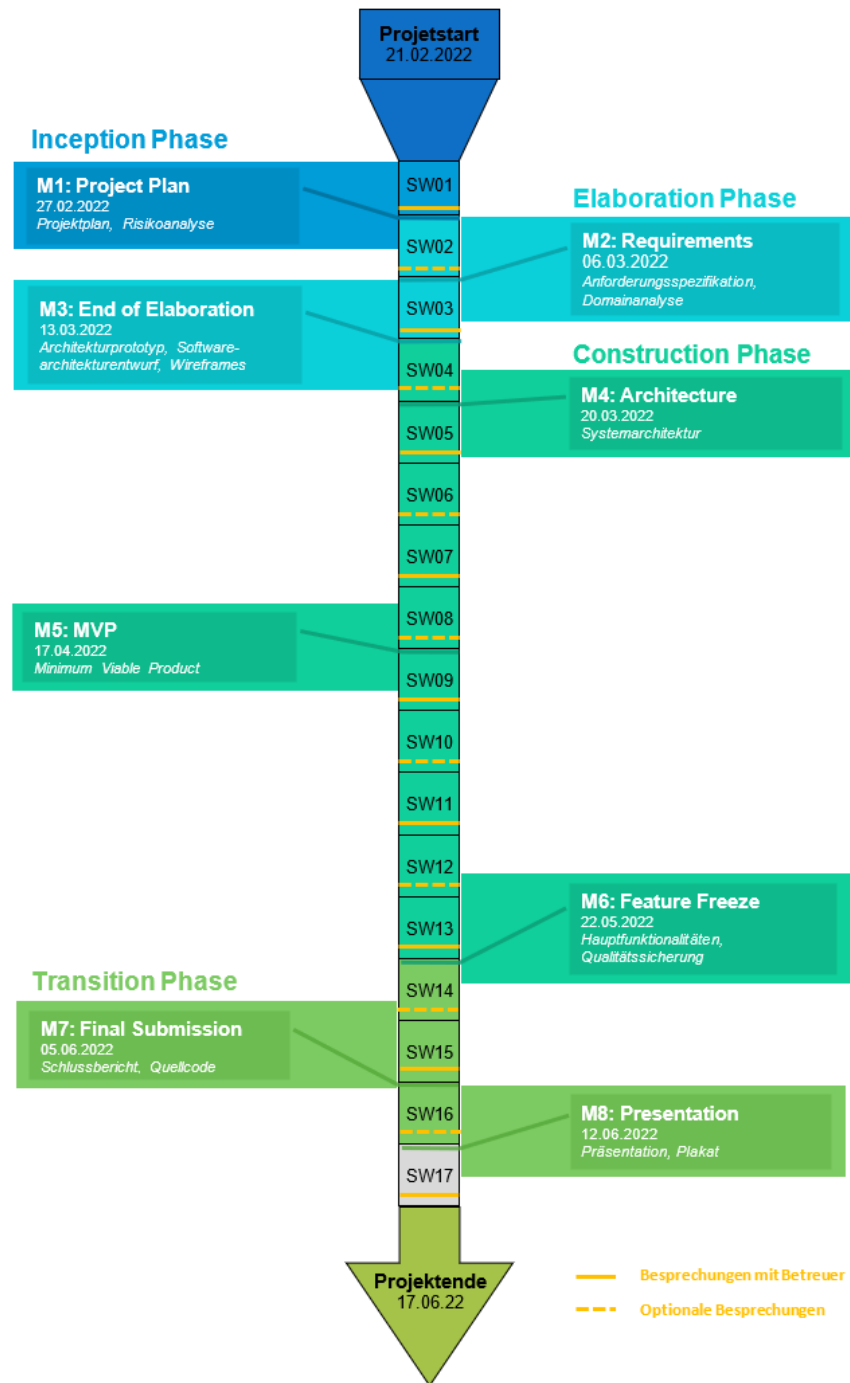


Abbildung 63: Zeitmanagement Übersicht

B.4. Meilensteine

M0: Kick off - 10.02.2022

Eine erste Sitzung mit dem Betreuer und dem Praxispartner wurde durchgeführt. Dabei wurden die Möglichkeiten des Teams und Absichten des Auftraggebers besprochen.

Arbeitsprodukte:

- Ideensammlung

M1: Project plan - 27.02.2022

Der Projektplan gibt dem Projekt die Struktur für einen erfolgreichen und termingerechten Abschluss. Er beinhaltet die Organisation und das Projektmanagement im Team. Ausserdem werden im Rahmen dieses Meilensteins alle Informationen und Vorgaben des Betreuers analysiert und wenn nötig mit ihm besprochen.

Arbeitsprodukte:

- Projektplan
- Risikoanalyse

M2: Requirements - 06.03.2022

Anforderungen an das Produkt sind spezifiziert und eine Domainanalyse wurde erstellt. Stakeholder, insbesondere der Kunde, wurden in die Spezifizierung einbezogen.

Arbeitsprodukte:

- Functional Requirements
- Non-Functional Requirements
- Domainanalyse
- Product Backlog (Epics, User Stories)

M3: End of Elaboration - 13.03.2022

Arbeitsumgebung ist vollständig eingerichtet. Ein lauffähiger Prototyp (vertikale Integration) wurde erstellt und geprüft (Proof of Concept). Die meisten technischen Risiken wurden minimiert oder vollständig eliminiert.

Arbeitsprodukte:

- Systemarchitekturentwurf
- Prototyp (Proof of Concept)
- Deploymentstrategie

M4: Architecture - 20.03.2022

Die physische und logische Architektur/das Design ist stabil und vollständig dokumentiert. Designentscheidungen für entwickelte Schnittstellen, Klassenstrukturen sowie eingesetzte Technologien, Persistenz-, UI- und UX-Lösungen sind begründet.

Arbeitsprodukte:

- Systemarchitektur
- Wireframes

M5: MVP - 29.03.2022

Das MVP, ein minimal brauchbares Endprodukt, steht. Eine mit den nötigsten Kernfunktionen ausgestattete erste Version ist online (im Web verfügbar). Systemtests sind durchgeführt und Coderichtlinien eingehalten. Das Produkt wird anschliessend im Rahmen eines Pilot-Tests während einem Playoff-Spiel von ausgewählten Zuschauern getestet.

Arbeitsprodukte:

- MVP
- Systemtest Protokoll
- Pilot-Test Protokoll

M6: Feature Freeze - 22 .05.2022

Das Produkt hat seine finale Grösse innerhalb dieses Projektes erreicht. Es werden keine neuen Funktionalitäten mehr hinzugefügt sondern nur noch die Implementierung der bestehenden Funktionen verbessert und Fehler behoben.

Arbeitsprodukte:

- Beta-Release des Produktes
- Usability-Test Protokoll

M7: Final Submission - 05.06.2022

Das Projekt und der Bericht sind abgeschlossen. Alle während der Arbeit erstellten Dokumente und der gesamte Quellcode werden eingereicht indem diese auf avt.i.ost.ch hochgeladen werden.

Arbeitsprodukte:

- Dokumentation (Schlussbericht)
- Eigenständigkeitserklärung
- Software Repository mit ganzem Code und allen Versionen
- Deployment-Dokumentation

M8: Presentation - 12.06.2022

Die Präsentation der Bachelorarbeit ist vorbereitet und das Team ist bereit diese zu halten. Das Plakat ist gemäss den Vorgaben der Hochschule erstellt.

Arbeitsprodukte:

- Präsentation (PowerPoint)
- Plakat

B.5. Iterationen (Sprints)

Sprints sind iterativ aufgebaut. Dadurch hat der letzte Sprint einen starken Einfluss auf den nächsten Sprint. Der Inhalt des neuen Sprints wird immer aus dem Produkt und Projekt Backlog zusammengestellt. Die Task eines Sprints werden als Issues in GitLab geführt. Ein Auszug davon befindet sich im Anhang.

B.6. Besprechungen

B.6.1. Besprechungen mit Betreuer

Alle zwei Wochen, jeweils am Montagmorgen trifft sich das Team am Campus mit dem Projekt-Betreuer für eine einstündige Besprechung. Darin wird sowohl der Fortschritt im Projekt, sowie das weitere Vorgehen besprochen. In den übrigen Wochen findet die Besprechung nur statt, wenn diese einen Mehrwert produziert. Ansonsten wird darauf verzichtet.

B.6.2. Sprint Planning

Zu Beginn jedes Sprints trifft sich das Team, um das Ziel für die aktuelle Iteration zu definieren. Ausserdem wird der Vortschritt des letzten Sprints reflektiert. Es werden Items im Backlog definiert, welche im aktuellen Sprint umgesetzt werden sollen. Somit initiiert die Besprechung einen neuen Sprint und legt die Arbeiten fest, die innerhalb der aktuellen Ettape umgesetzt werden sollen.

B.6.3. Daily Scrum

An allen Arbeitstagen wird der Informationstand über den Fortschritt im aktuellen Sprint, die Problemherde und offene Pendenzen synchronisiert. Dies erfolgt mittels einem kurzen, informellen Meeting zu Beginn des Arbeitstages.

B.7. Risikomanagement

B.7.1. Risiken

Um die Risiken und ihren potentiellen Schaden besser einschätzen zu können, sind diese anhand ihrem maximalen Schadenspotential [h] und ihrer möglichen Eintrittswahrscheinlichkeit aufgelistet und visualisiert. Alle Zahlen sind Schätzungen und sind nicht als verbindlich zu werten.

Nr	Beschreibung	[h]	%	Gewichtung
R1	Mangelnde Performance der Infrastruktur	15	60%	9
R2	Mangelnde Retention-Rate	30	40%	12
R3	Bedürfnisse der Stakeholder nicht ausreichend erfüllt	40	30%	12
R4	Komplikationen bei Schnittstelle für Spieldaten	30	50%	15
R5	Pilot Test während Spiel in Stadion kann nicht durchgeführt werden	20	60%	12
R6	Unpassende Auswahl der Softwarearchitektur	30	20%	6
R7	Zu hohe Komplexität der Funktionalität	25	50%	12.5
R8	Anforderungen der Coding-Guidelines nicht erfüllt	10	40%	4
R9	Fehlende / fehlerhafte Tests	20	40%	8
R10	Probleme beim Aufsetzen der Softwarekomponenten	20	30%	6
R11	Probleme beim Aufsetzen der CI / CD Pipelines	12	50%	6
R12	Nichteinhalten von Meilensteinen / Deadlines	20	25%	5
R13	Unterbrüche in der Verfügbarkeit der OST Infrastruktur	10	50%	5
R14	Mangelnde Kommunikation im Team führt zu Missverständnissen	30	60%	18
R15	Sicherheitslücken in der Software	20	20%	4
R16	Mangelnde Technologie-Kenntnisse im Team	30	30%	9
				Total: 143.5

Tabelle 89: Risiken

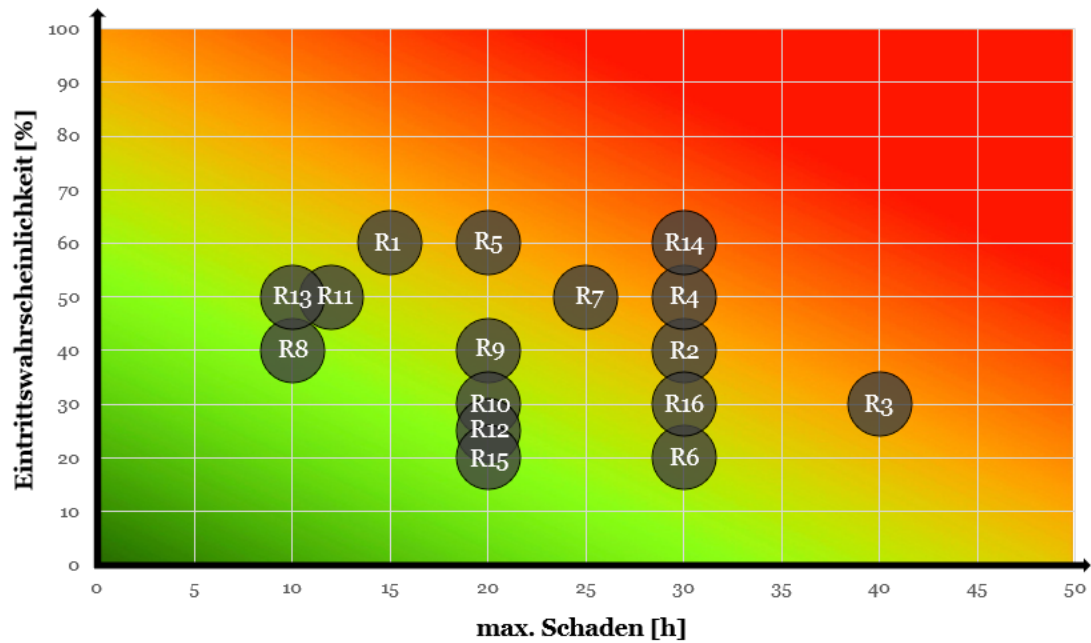


Abbildung 64: Risikoanalyse

B.7.2. Umgang mit Risiken

Nr	Vorbeugung	Verhalten beim Eintreten
R1	Einfaches Deployment ermöglichen	Produkt auf bessere Infrastruktur mit mehr Performance mitigieren (AWS / Azure)
R2	Fokus auf Usability im Requirement Engineering	Usability verbessern und interessante Features entwickeln
R3	Bedürfnisse der Stakeholder als Anforderungen definieren	Verbesserungen zusammen mit Stakeholder erarbeiten
R4	Ansprechperson kontaktieren und Ausgangslage analysieren	Ansprechperson und Experten als Unterstützung nutzen
R5	Pilot Test als Fokus im Projektmanagement definieren	Alternativer Test des MVP durchführen
R6	Umfassende Planung bei der Wahl von Architektur-Bestandteilen	Kompromiss / Alternative für unpassenden Bestandteil finden
R7	Komplexität der Funktionen in der Planung realistisch einschätzen	Funktionalität einschränken, auf MVP konzentrieren
R8	Coding Guidelines festlegen, Code-Reviews durchführen	In Code-Review ansprechen, aus den Fehlern lernen
R9	Umfassende, strukturierte Testpläne schreiben	Testfälle ergänzen, Testcoverage erhöhen, Test-Reviews durchführen
R10	Nur verbreitete Software mit gutem Support verwenden	Komponente ersetzen
R11	Dokumentation von GitLab CI / CD studieren	Unterstützung von Kommilitonen und Experten anfragen

R12	Agiles Projektmanagement, Probleme frühzeitig kommunizieren	Prioritäten setzen, unwichtiges vernachlässigen
R13	Lokale Backups aller Daten	Eigener GitLab Runner entwickeln
R14	Periodische Meetings und Informationskanal (MS Teams) definieren	Zusätzliche Besprechungen festlegen
R15	Aktuelle Versionen von Komponenten verwenden, bekannte Sicherheitslücken meiden	Sicherheitslücke adressieren und beheben
R16	Fähigkeiten trainieren in Testumgebung, Einlesen in neue Technologien	Expertenmeinung einholen

Tabelle 90: Umgang mit Risiken

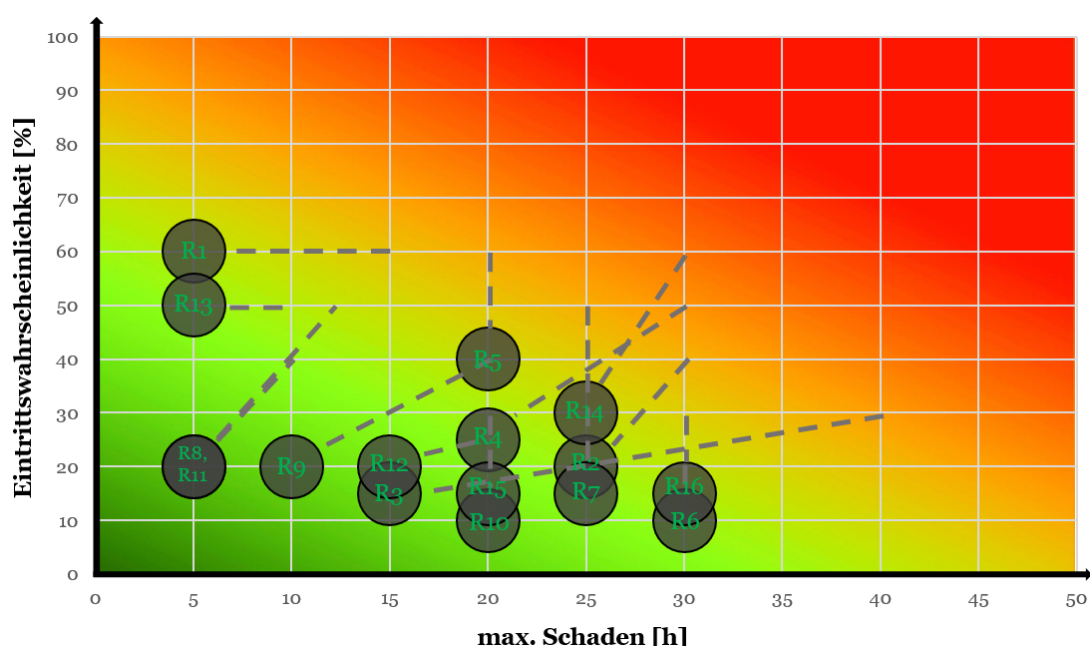


Abbildung 65: Risikoanalyse Mitigiert (stand M1)

B.7.3. Einflüsse auf Entscheide während M2

R1:

Damit die Performance sichergestellt ist, haben wir uns entschieden auf AWS zu setzen. Somit ist das Risiko R1 praktisch auf null gesunken.

R3:

Um R3 zu mitigieren wurden Storyboards erstellt und mit den Stakeholdern besprochen. Unsere Vorstellung des Produkt deckt sich sehr gut mit den Bedürfnissen von Michael Bart, weshalb die Eintrittswahrscheinlichkeit nun tiefer ist.

R5:

Bezüglich R5, dem Pilot Test haben wir entschieden, dass dieser nicht zwingend umgesetzt werden muss. Wir fokussieren uns auf eine gute Basis und möchten in der Entwicklung keine überstürzten Entscheidungen treffen. Falls der Pilot Test nicht durchgeführt werden kann, werden wir das App auf eine andere Art testen, wie zum Beispiel Performance Tests.

R7:

Der MVP soll möglichst einfach gehalten werden und nur die minimalen Anforderungen enthalten. Wir haben aber auch schon viele Ideen zu weiteren Features gesammelt, weshalb sich die Eintrittswahrscheinlichkeit von R7 ein wenig erhöht hat.

R14:

Die Kommunikation im Team funktioniert gut und wir haben sehr ähnliche Vorstellungen was das gesamte Projekt angeht. Somit haben sich die Eintrittswahrscheinlichkeit und der Schaden gesenkt.

R15:

Mangelnde Technologie-Kenntnisse im Bereich AWS sind ein hohes Risiko. Wir haben uns jedoch gut über die angebotenen Services informiert und werden uns noch mit Prof. Mirko Stocker, einem Expertem in Bereich Cloud Solutions austauschen.

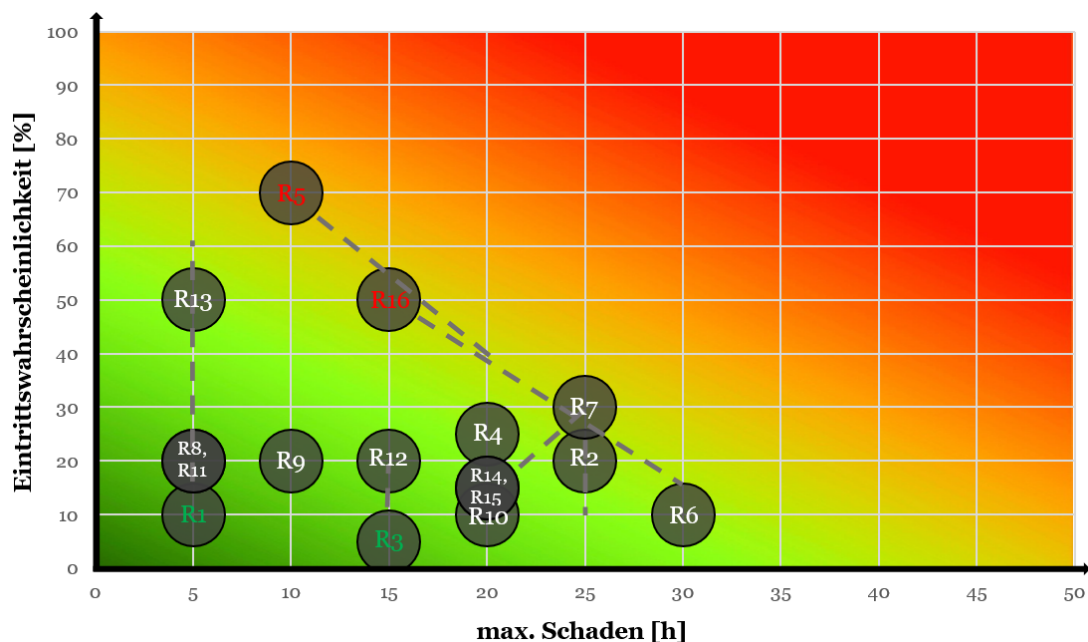


Abbildung 66: Risikoanalyse Mitigiert (stand M2)

B.7.4. Einflüsse auf Entscheide während M3

R4:

Die Kommunikation mit dem IT Verantwortlichen der SCRJ Lakers ist sehr träge. Telefonisch ist er bisher nicht erreichbar und der E-Mail Verkehr sehr langsam. Aus diesem Grund erhöht sich die Wahrscheinlichkeit von Komplikationen bezüglich der Schnittstelle für Spielerdaten.

R5:

Im Meilenstein 3 wurde entschieden, dass wenn der Pilot Test nicht stattfinden kann, die App in einem anderen Rahmen getestet wird. Da dies kaum einen Mehraufwand für das Projekt erfordert, hat sich der maximale Schaden verkleinert.

R7:

Der Funktionsumfang für den MVP wurde nun klar definiert und auf einem Minimum gehalten. Aus diesem Grund verringert sich die Eintrittswahrscheinlichkeit des Risikos.

R11:

Das CI/CD im Gitlab wurde nun in Kombination mit AWS Amplify erstmals aufgesetzt und getestet. Durch dessen Komplexität hat sich aber der maximale Schaden bei möglichen, künftigen Problemen

etwas erhöht.

R16:

Das Frontend wurde zusammen mit AWS Amplify initialisiert. Zusätzlich wurden bereits API Calls und Realtime-Updates erfolgreich getestet. Somit hat sich die Eintrittswahrscheinlichkeit des Risikos mitigiert.

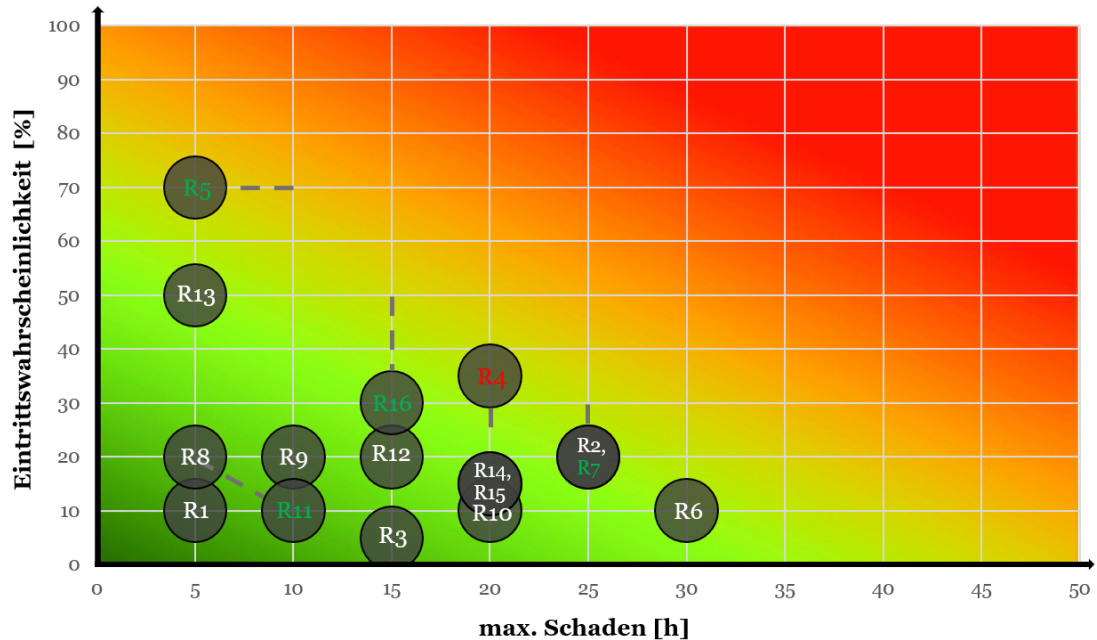


Abbildung 67: Risikoanalyse Mitigiert (stand M3)

B.7.5. Einflüsse auf Entscheide während M4

R2:

Durch Usability Tests konnten die Bereiche Navigation, Präsentation und Bedienung validiert werden und kleinere Unklarheiten behoben werden. Die Testpersonen haben positives Feedback auf das Konzept und das Design gegeben und meinten, dass sie das App nutzen würden. Dadurch haben wir die Eintrittswahrscheinlichkeit des Risikos gesenkt.

R17:

Im aktuellen Meilenstein ist durch die mangelnde Kommunikation mit verschiedenen Instanzen der SCRJ Lakers ein neues Risiko entstanden. Erst nach vielen Bemühungen konnten Informationen beschaffen werden, von welchen der weitere Fortschritt des Projektes abhängt.

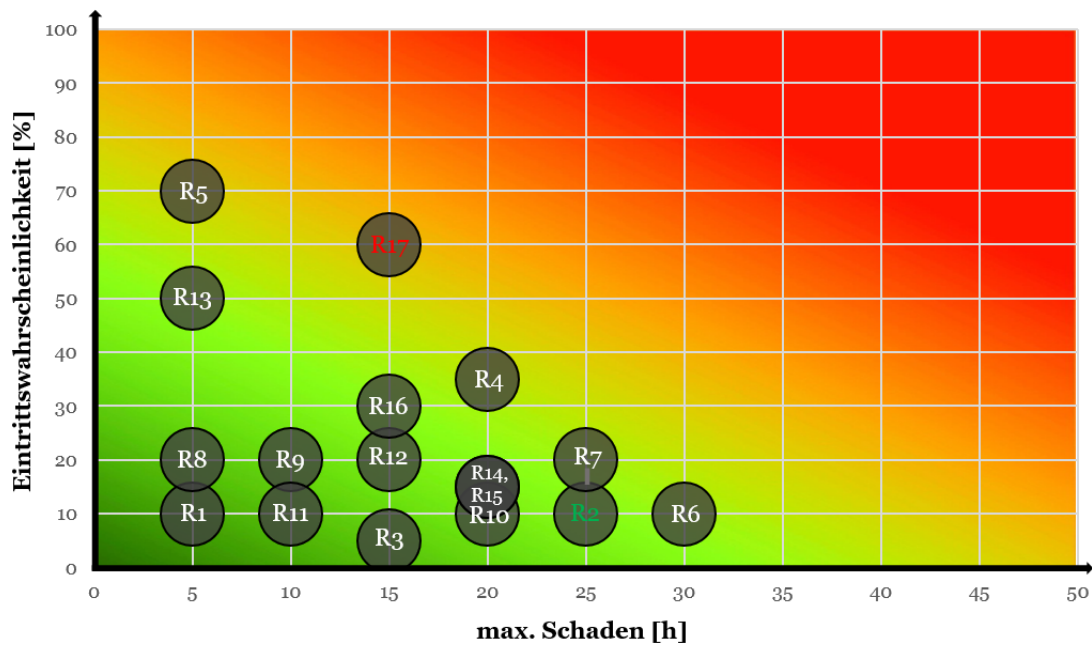


Abbildung 68: Risikoanalyse Mitigiert (stand M4)

B.8. Arbeitspakete

Die Arbeitspakete wurden anhand der Produkte in den jeweiligen Meilensteinen grob definiert und als Issues auf GitLab erfasst. Mit Labels werden sie den verschiedenen Sprints (1 - 8) zugeordnet und mit relativen Aufwänden eingeschätzt. Während den Sprint Planning Besprechungen werden die groben Arbeitspakete des nächsten Sprints in kleinere, spezifischere Pakete aufgeteilt und diese mit einer absoluten Aufwandsschätzung versehen.

B.9. Qualitätsmassnahmen

Für das Team ist es wichtig, dass das Produkt eine hohe Qualität aufweist. Deshalb werden folgende Massnahmen getroffen:

Massnahme	Zeitraum	Ziel
Sprint Reviews	Nach jedem Sprint	Das Team tauscht sich über den aktuellen Stand aus, um Engpässe und gegebenenfalls Probleme frühzeitig zu erkennen.
Meilensteine	In der Zeitplanung definierter Tag	Meilensteine werden gesetzt, um ein erfolgreiches projekt zu garantieren.
Code Reviews	Bei jedem Merge Request	Mit Code Reviews wird Clean Coding garantiert.
Automatisierte Tests	Bei jedem Push in einen Branch	Automatisierte Tests garantieren das korrekte Verhalten von Code.
Definition of Done	Beim Schliessen eines Issues	Für jedes Issue wird die DoD eingehalten.

Tabelle 91: Definition of Done

B.9.1. Definition of Done

Die Definition of Done beinhaltet folgende Punkte:

- CI erfolgreich durchlaufen
- Coding Guidelines sind eingehalten
- Bei Merge: Code Review umgesetzt
- Sinnvolle Unit Test wurden zu neuem Code ergänzt
- Wenn nötig: Dokumentation aktualisiert

B.9.2. Dokumentation

Die Dokumentation wird während dem Projekt fortlaufend aktualisiert und ist auf dem Gitlab Server der OST abgelegt. Bei jedem Merge auf den Master Branch wird das PDF aus dem Source Code kompiliert und als Artefakt der Pipeline abgelegt.

B.9.3. Projektmanagement

Als Projektmanagement wird Scrum+ verwendet was eine Kombination aus Scrum und RUP ist. Von RUP wird die Langzeitplanung und somit die Aufteilung in die Phasen Inception, Elaboration, Construction und Transition übernommen. Zusätzlich arbeitet das Team mit Sprints, welche in Scrum definiert sind. Dazu gehören Product Backlog, Sprint Planning, Sprint Backlog, Daily Scrum, Sprint Review und Sprint Retrospective.

Für jeden Task wird ein Issue erstellt und der geleistete Aufwand jedes Teammitglieds erfasst. Um einen Überblick über die geleisteten Stunden zu haben werden die Zeiten jede Nacht neu ausgewertet und in Plots dargestellt. Die Plots zeigen die Aufwände pro Person, Meilenstein und Kategorie.

B.9.4. Unit Testing

Über die Gitlab CI/CD wird bei jedem Push die Pipeline durchlaufen. Somit wird geprüft, ob sich der neue Code bauen lässt und ob die Unit Tests erfolgreich waren.

B.9.5. Code Reviews

Bei einem Code Review schaut sich ein zweites Teammitglied den veränderten Code an. Sind die Qualitätsmassnahmen sichergestellt kann der Merge Request angenommen werden, ansonsten muss es nochmals überarbeitet werden.

B.9.6. Code Style Guides

Für die Einhaltung der Code Style Guides werden Code Formatting Tools eingesetzt. Durch sie wird anhand von vordefinierten Regeln automatisch ein stilistisch einheitlicher Code geschrieben.

B.9.7. Namenskonvention

Die Issues (Tasks), insbesondere dessen Titel, unterliegen folgender Struktur:

Titel	Beschreibung	Beispiele
Sprache	Die Titel der Issues werden in englischer Sprache verfasst	Evaluate use cases
Satzbau	Die Titel sind immer nach dem Schema: beschreibendes Verb + Nomen + Übriges aufgebaut. Ausnahme Besprechungen. Es werden keine Modalverben eingesetzt.	Design Wireframes, Evaluate Interface, Develop Service, Test Prototype
Gross-/Kleinschreibung	Die Titel, das Verb, der Issues beginnen immer mit einem Grossbuchstaben	Setup Server

Zuweisung	Da Issues in unserer GitLab-Version nicht an mehrere Teammitglieder zugewiesen werden können, werden Titel von Gruppenaufgaben mit dem Präfix <All: > ergänzt	All: Kickoff Meeting
Abkürzungen	Wörter im Titel werden vollständig ausgeschrieben	
Textlänge	Der Titel ist so kurz wie möglich und so lange wie nötig, um das Issue eindeutig zu identifizieren	Write Section Projectplan

Tabelle 92: Issue Namenskonvention

C. Projektmonitoring

C.1. Projektstruktur

Das Projekt ist in insgesamt sechs Repositories unterteilt. Diese liegen auf der GitLab Instanz der OST. Die konkrete Nutzung der einzelnen Repositories ist im entsprechenden README detaillierter beschrieben.

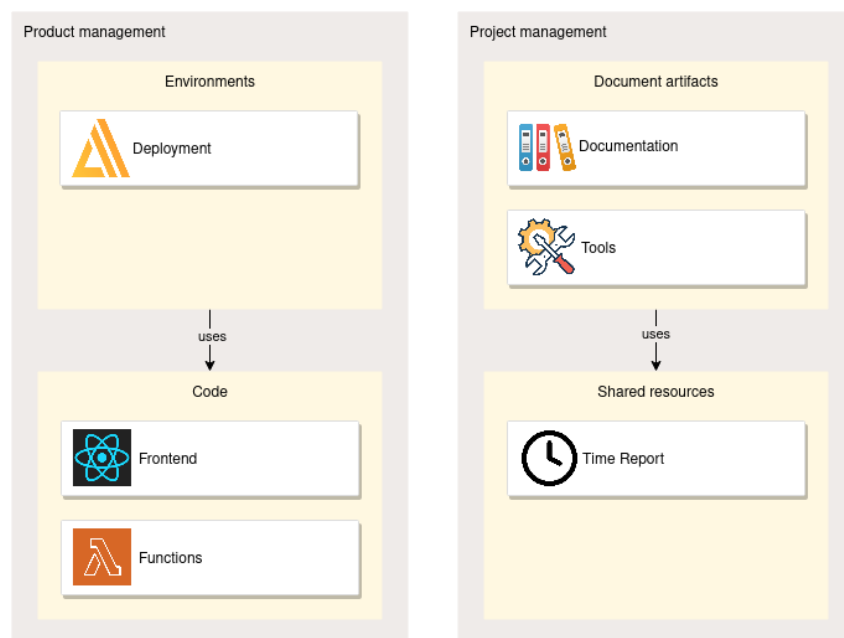


Abbildung 69: Projektstruktur

C.1.1. Produktmanagement

Diese Gruppe an Repositories bildet das resultierende Produkt ab. Damit kann das Endprodukt hergestellt und veröffentlicht werden.

Deployment

In diesem Repository wird das aufgesetzte Backend as a Service (BaaS) beschrieben. Dazu zählen die gesamte Konfiguration jedes eingesetzten Services für die beiden Environments `<Prod>` und `<Dev>`. Nur damit lässt sich das Tippspiel schlussendlich veröffentlichen. Eine erfolgreiche Aktualisierung von Front- oder Backend in ein veröffentlichtes Environment bedingt die Angaben aus diesem Repository. Zusätzlich finden sich die Systemtests in diesem Repository.

Der hauptsächliche Programmcode liegt in den zwei ausgelagerten Repositories Frontend und Functions. Darin liegen auch die Unit Tests der jeweiligen Systemkomponenten.

In diesen Code Repositories werden folgende Aufgaben erledigt:

- Code Building
- Unit Testing
- Quality control

Frontend

Das Frontend Repository beinhaltet alles, was für die Entwicklung und den Betrieb von der entwickelten *PWA* vonnöten ist. Zu erwähnen sind die Konfiguration der *PWA* und Testsuite, die eingesetzten Packages und Libraries, sämtliche React Komponenten und die eigentliche Businesslogik.

Functions

In Functions sind die Konfiguration und Businesslogik aller eigens entwickelten Backend-Funktionen fürs Tippspiel abgelegt. Die einzelnen Funktionen nutzen auch gemeinsame Ressourcen und Konfigurationen. Dessen Benützung ist ebenfalls hier dokumentiert.

Jedes Produkt-Repository hat zwei beständige Branches `<Dev>` und `<Main>`. Entwickelt wird in weiteren Branches, welche auf ein Issue (Task) referenzieren.

Branch	Beschreibung
main	Der Code in diesem Branch ist stabil und kann jederzeit für den produktiven Einsatz verwendet werden. Auch die Interoperabilität mit anderen Systemkomponenten und Umsystemen ist gewährleistet. Der Entwicklungsstand enthält nahezu keine Errors und Defects mehr.
dev	Enthält den zusammengetragenen Code aus den verschiedenen Issues. Dieser ist für sich lauffähig und kann im Gesamtsystem eingesetzt werden. Schwerwiegende Errors und Defects sind behoben. Werden die Qualitätsansprüche vom <code><main></code> Branch erfüllt, soll der aktuelle Commit in den <code><main></code> Branch verschoben werden.
<code>ISSUE_ID-ISSUE_NAME</code>	Diese Branches werden für die tägliche Weiterentwicklung des Systems verwendet. Sie sind in der Regel nicht stabil und erhalten regelmässig Aktualisierungen. Werden die Qualitätsansprüche vom <code><dev></code> Branch erfüllt, soll der aktuelle Commit in den <code><dev></code> Branch verschoben werden.

Tabelle 93: Produkt-Repository Branches

C.1.2. Projektmanagement

Die Dokumentation und Zeiterfassung sind ebenfalls versioniert.

Documentation

Der Projektfortschritt ist in diesem Repository dokumentiert. Die Projektdokumentation ist in LaTeX geschrieben und steht als Artefakt im PDF-Format zur Verfügung.

Tools

Beschreibt den Aufbau des gemieteten OST-Servers und die Konfiguration von Entwicklertools sowie Anleitungen für dessen Benutzung. Es lagert auch die CI/CD Konfigurationsvorlagen für Code und Environment Repositories. Damit wird eine möglichst einheitliche Pipeline garantiert.

Time Report

Analysiert die Issues von GitLab und generiert täglich um 2 Uhr Auswertungen zur geleisteten Arbeit. Die Ergebnisse stehen als Artefakt im PNG-Format zur Verfügung.

C.2. Zeitauswertung

Die Zeitauswertung generiert drei Grafiken. Eine den Aufwand pro Teammitglied, eine weitere für den Aufwand pro 2-Wochen Sprint und eine letzte mit zu Beginn definierten Labels.

C.2.1. Zeitaufwand pro Teammitglied

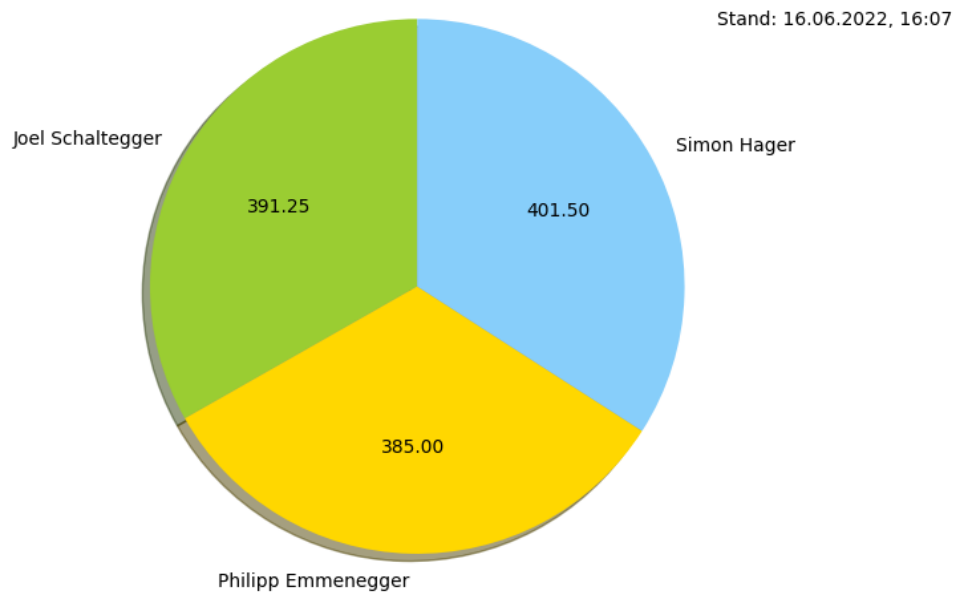


Abbildung 70: Zeitaufwand pro Teammitglied

C.2.2. Zeitaufwand pro Sprint

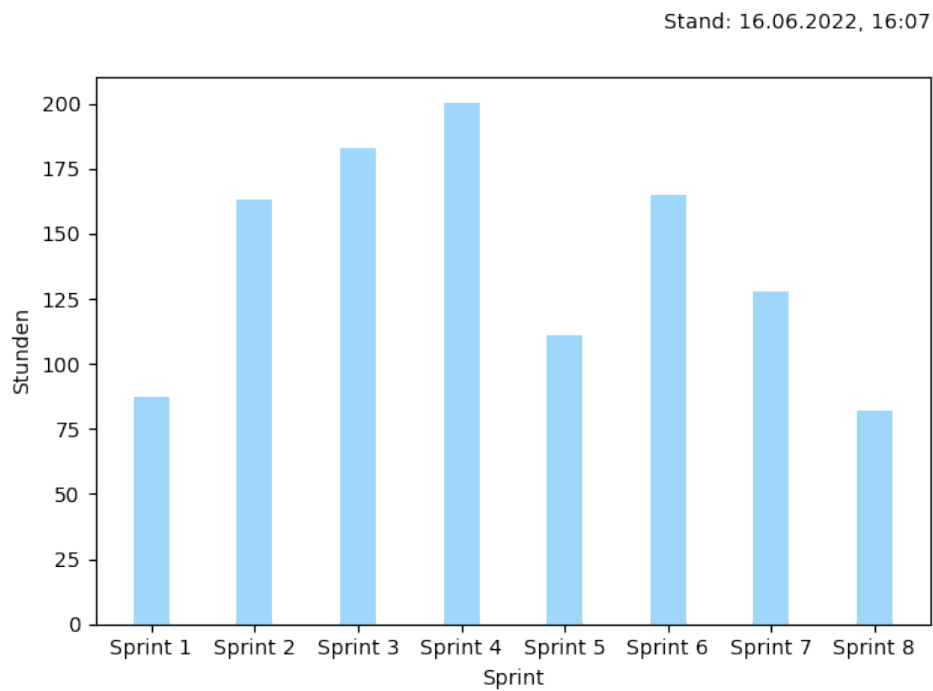


Abbildung 71: Zeitaufwand pro Sprint

C.2.3. Zeitaufwand nach Kategorie

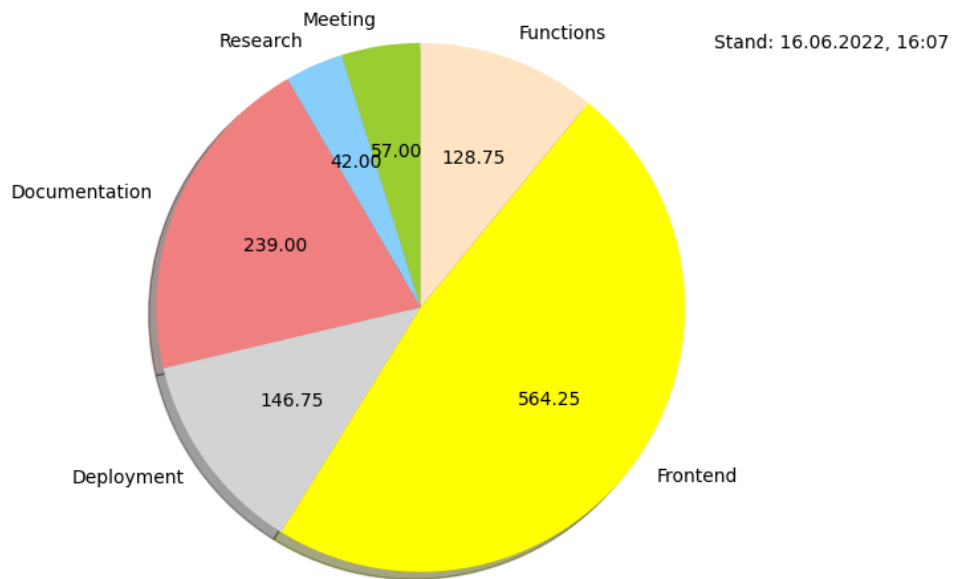


Abbildung 72: Zeitaufwand nach Kategorie

D. Verzeichnisse

D.1 Glossar

Actions Funktionen die aufgerufen werden um den Applikationszustand zu verändern. 52

Amazon Web Services Ein US-amerikanischer Cloud-Computing-Anbieter. 2, 160

Atomicity-Consistency-Isolation-Durability Atomarität, Konsistenzerhaltung, Isolation und Dauerhaftigkeit sind erwünschte Eigenschaften von Transaktionen in Datenbankmanagementsystemen. 41, 160

Boilerplate Boilerplate ist die Bezeichnung für Code-Segmente, die an mehreren Stellen mit geringen bis keinen Änderungen wiederholt werden. 52

Command Line Interface Befehlszeilenschnittstelle. 2, 160

Continuous Integration/Continuous Delivery die kombinierte Praxis von Continuous Integration und Continuous Delivery oder Continuous Deployment. 30, 160

Cypress Junges Open-Source-End-To-End-Test-Framework für Web-basierte, grafische Benutzeroberflächen. 97, 163

Datenschutz-Grundverordnung der Europäischen Union Ist seit dem 25. Mai 2018 in Kraft. Sie kann für Akteure, die auf dem Gebiet der EU tätig sind, unmittelbar anwendbar sein. 56, 160

Don't repeat yourself Besagt, Redundanz zu vermeiden oder zumindest zu reduzieren. Es handelt sich hierbei auch um ein Prinzip von Clean Code. 73, 160

Figma eine kollaborative Software zum Erstellen von Prototypen im Bereich des UX bzw. UI Design. 47, 82, 87

Font Awesome eine Icon-Sammlung, das skalierbare Vektorbilder anbietet, die mit CSS angepasst werden können. 46

GitLab Eine komplette DevOps-Plattform, die in Form einer einzelnen Anwendung bereitgestellt wird. 30, 71, 72, 76, 77

GraphQL Eine Open-Source-Datenabfrage- und Manipulationssprache. 2, 41, 54, 56, 58, 59, 61, 72, 135

High-Fidelity-Prototyping ein Prototyp mit einem hohen Detailgrad an Funktionalität und Design. 47

Hooks Mit React Hooks lassen sich State und andere React Features nutzen, ohne dass eine Klasse geschrieben werden muss. 52

Institut für Software Institut an der OST mit dem Schwerpunkt Software. 8, 160

Marketingmonkeys Marketing Monkeys kümmert sich um Newsletter-Marketing in die Neuzeit und implementierten Automation Monkey. 49

Minimum Viable Product Erste minimal funktionsfähige Iteration eines Produkts. 15, 160

Miro Ein «unendliches» Online-Whiteboard. 13

Net Promoter Score Kennzahl, die Aufschluss über die Zufriedenheit von Kunden eines Unternehmens gibt. 10

Nichtfunktionale Anforderungen Die nichtfunktionalen Anforderungen beschreiben, wie gut das System die Leistung erbringen soll. Sie werden vielfach als Randbedingungen und Qualitätseigenschaften verstanden.. 27, 34, 160

On-Premise Inhouse, ein Software-Nutzungsmodell, bei dem ein Lizenznehmer eine serverbasierte Software auf eigener oder gemieteter Hardware im eigenen Rechenzentrum (Inhouse) betreibt. 35

Progressive Web Application Progressive Web Apps können wie eine Webseite mit HTML5, CSS3 und JavaScript erstellt werden. Zusätzlich dienen sogenannte Service Worker durch optimiertes Caching den Offline-Funktionalitäten und der Installierbarkeit. 2, 51, 160

Property Drilling Bezeichnet das Übergeben derselben Daten über mehrer Hierarchie-Level via Parameter. 52

- Quality Gates** Punkte im Ablauf eines Entwicklungsprojekts, bei denen anhand von im Voraus eindeutig bestimmten Qualitätskriterien über die Freigabe des nächsten Projektschrittes entschieden wird. 77
- README** eine Datei, die üblicherweise mit Software mitgeliefert wird und diverse, oft wichtige Informationen über die Software enthält. 49, 51, 71
- Reducer** Eine Funktion, welche Veränderungen in einem Zustand erkennt und diese persistiert. 52, 77
- Redux** Redux ist eine JavaScript-Bibliothek zur Verwaltung von Zustandsinformationen in einer Webanwendung. 52, 53, 131
- Representational State Transfer** eine einfache, zustandslose Architektur, die in der Regel über HTTP läuft. 41, 160
- Search Engine Optimization** Bezeichnet Maßnahmen, die dazu dienen, die Sichtbarkeit einer Website und ihrer Inhalte für Benutzer einer Websuchmaschine zu erhöhen. 78, 161
- Self-Assessment Manikin** Ein sprachfreies Verfahren, um affektive (gefühlbetonte) reaktionen durch die selbsteinschätzung der teilnehmer zu erfassen. 101, 160
- SemanticUI** ein Entwicklungsframework, das die Erstellung schöner, responsiver Layouts mit menschenfreundlichem HTML unterstützt. 46
- Serverless** ein cloudnatives Entwicklungsmodell, bei dem Entwickler Anwendungen erstellen und ausführen können, ohne Server verwalten zu müssen. 34, 51, 52, 59
- Service Worker** Moderne Browsertechnologie, die mittels JavaScript einen Proxy zwischen dem Webbrowser und dem Server bereitstellt, der vielfältig genutzt werden kann. Der bekannteste Einsatzzweck ist es, bestimmte Seiten auch offline verfügbar zu machen. 2, 51, 52, 55
- Software Development Kit** eine Sammlung von Programmierwerkzeugen und Programmbibliotheken, die zur Entwicklung von Software dient. 42, 161
- Spezifische, Messbare, Attraktive, Realistische und Terminierte** SMART steht für spezifisch, messbar, attraktiv, realistisch und terminiert. 10, 161
- Syntactically Awesome Style Sheets** eine Stylesheet-Sprache. 51, 161
- Transport Layer Security** Verschlüsselungsprotokoll zur sicheren Datenübertragung im Internet. 29, 161
- Uniform Resource Locator** identifiziert und lokalisiert eine Ressource, beispielsweise eine Webseite. 39, 161
- User Experience Questionnaire** Ein Fragebogen, der eine schnelle Messung verschiedener Kriterien der Softwarequalität erlaubt. 101, 161
- User-centered-Design** Iterativer Design-Prozess, welcher einen starken Fokus auf den Benutzern und ihren Bedürfnissen hat. 2, 4, 46, 52

D.2. Akronyme

- ACID** *Atomicity-Consistency-Isolation-Durability*. 41
- AWS** *Amazon Web Services*. 2, 52
- CI/CD** *Continuous Integration/Continuous Delivery*. 30
- CLI** *Command Line Interface*. 2, 58, 76, 109
- DRY** *Don't repeat yourself*. 73
- DSGVO** *Datenschutz-Grundverordnung der Europäischen Union*. 56
- IFS** *Institut für Software*. 8
- MVP** *Minimum Viable Product*. 15
- NFA** *Nichtfunktionale Anforderungen*. 27, 34
- PWA** *Progressive Web Application*. 2, 34, 51, 52, 78, 155
- REST** *Representational State Transfer*. 41, 49, 50, 56, 61
- SAM** *Self-Assessment Manikin*. 101, 102, 104

SASS *Syntactically Awesome Style Sheets*. 51, 53, 54

SDK *Software Development Kit*. 42, 52

SEO *Search Engine Optimization*. 78

SMART *Spezifische, Messbare, Attraktive, Realistische und Terminierte*. 10

TLS *Transport Layer Security*. 29

UEQ *User Experience Questionnaire*. 101, 102, 103

URL *Uniform Resource Locator*. 39

D.3. Abbildungsverzeichnis

1.	Screenshots Fan-Engagement-App	8
2.	Screenshot Admin-Bereich	8
3.	Net Promoter Score[7]	10
4.	Use Case Diagramm	15
5.	Domain Model	32
6.	Makroarchitektur	34
7.	State Diagram eines Games	36
8.	Benutzermigration	38
9.	Passwordless-Login-Prozess	39
10.	Schema der Datenbank	40
11.	State Diagram eines Games	44
12.	Farbschema der SCRJ Lakers	46
13.	Schriftart Rubik	46
14.	Font Awesome Piktogramme[14]	46
15.	User Centered Design Prozess [15]	47
16.	Startbildschirm im Prototyp	48
17.	Tippspiel-Ansicht im Prototyp	48
18.	Challenge-Ansicht im Prototyp	48
19.	Preis-Ansicht im Prototyp	48
20.	Typ des Zustands im Frontend	53
21.	Error Popup	55
22.	AWS Cloud Systemübersicht	57
23.	E-Mail-Registrierungsprozess	62
24.	Login Provider Registrierungsprozess	63
25.	E-Mail-Anmeldeprozess	64
26.	E-Mail-Anmeldeprozess	65
27.	Login Provider Anmeldeprozess	67
28.	Gleiche Attribute für alle Challengearten	69
29.	Challenge States	70
30.	Simple Factory Evaluator	70
31.	GitLab Pipelines	72
32.	Aufbau Webpack	73
33.	SonarQube Dashboard	77
34.	Google Lighthouse Audit Resultat	78
35.	Navigationsleiste vorher	82
36.	Navigationsleiste nachher	82
37.	Preisübersicht vorher	83
38.	Preisübersicht nachher	83
39.	Ergänzende Verknüpfung für neue Preis-Seite	83
40.	Preis einlösen	84
41.	Antwortauswahl vorher	85
42.	Antwortauswahl angepasste Variante	85
43.	Challengeübersicht vorher	86
44.	Registrierung vorher	87
45.	Registrierung nachher	88
46.	Sortierung Spielerauswahl	89
47.	Menüpunkt für Informationen	90
48.	Informationsbereich	90
49.	Loader nach der Auswahl eines Spielers	91
50.	Architektur der Api Tests	93
51.	Grafana Dashboard von TestszENARIO 6	93
52.	Latenzen von TestszENARIO 6	94
53.	Zeiten von setChallengeTip	94
54.	Grafana Dashboard von TestszENARIO 6 mit sichtbaren Cold-Starts	95

55.	Erfolgreiche E2E Testdurchführung mit <i>Cypress</i>	97
56.	Preiszusammensetzung eines Spiels gemäss End-to-end tests	98
57.	Preiszusammensetzung eines Spiels gemäss Crowd Test	100
58.	SAM Skala [40]	101
59.	SAM und UEQ im Vergleich	101
60.	UEQ Resultate	103
61.	UEQ Vergleich mit Benchmark	103
62.	Heimdall Dashboard	142
63.	Zeitmanagement Übersicht	144
64.	Risikoanalyse	148
65.	Risikoanalyse Mitigiert (stand M1)	149
66.	Risikoanalyse Mitigiert (stand M2)	150
67.	Risikoanalyse Mitigiert (stand M3)	151
68.	Risikoanalyse Mitigiert (stand M4)	152
69.	Projektstruktur	155
70.	Zeitaufwand pro Teammitglied	157
71.	Zeitaufwand pro Sprint	157
72.	Zeitaufwand nach Kategorie	158

D.4. Tabellenverzeichnis

1.	Zuschaueraufkommen an Heimspielen der Saison 2021/22[5]	9
2.	Aktoren	12
3.	User Journey eines Besuchers	13
4.	Schritte des Administrators zur Erstellung eines Spiels	13
5.	Schritte des Administrators zur Durchführung einer Challenge	14
6.	Schritte des Besuchers zur Durchführung einer Challenge	14
7.	Schritte des Besuchers zur Einlösung einer gewonnenen Belohnung	14
8.	Use Case 1.1 CRU Game	16
9.	Use Case 1.2 Assign challenges	17
10.	Use Case 1.3 Publish challenge	17
11.	Use Case 1.4 Close challenge	17
12.	Use Case 1.5 Assign rewards	18
13.	Use Case 1.6 Assign sponsor	18
14.	Use Case 1.7 Assign squad	19
15.	Use Case 1.8 Close game	19
16.	Use Case 2.1 CRU challenge	20
17.	Use Case 2.2 Archive challenge	21
18.	Use Case 3.1 CRU reward	21
19.	Use Case 3.2 Archive reward	22
20.	Use Case 4 CRUD sponsor	22
21.	Use Case 5 CRUD player	23
22.	Use Case 6 Login	24
23.	Use Case 7 Register	24
24.	Use Case 8 RUD User profile	25
25.	Use Case 9.1 Solve challenge	26
26.	Use Case 9.2 Display solution	26
27.	Use Case 10.1 Read community results	26
28.	Use Case 10.2 Share results	27
29.	Use Case 11 Read rewards	27
30.	NFA Suitability	28
31.	NFA Performance efficiency	28
32.	NFA Compatibility	28
33.	NFA Usability	29
34.	NFA Reliability	29
35.	NFA Security	30
36.	NFA Maintainability	30
37.	NFA Portability	30
38.	NFA Fan-Engagement	30
39.	NFA Kosten	31
40.	NFA Styleguide	31
41.	Objekte im Domain Model	33
42.	Zustände eines Games	37
43.	Zustände einer Challenge	37
44.	Zustände eines ChallengeTips	37
45.	Beispiel zur Auswertung auf Gleichheit	43
46.	Beispiel zur Auswertung auf Resultat	43
47.	Beispiel zur Auswertung auf Schätzung	44
48.	Mögliche Gewinne in einem Beispiel-Spiel	45
49.	Gewinnverteilung in einem Beispiel-Spiel	45
50.	Inhalte der National League Schnittstelle	49
51.	Vergleich von AWS und Google Cloud [21, 22]	56
52.	Buckets	59
53.	Secrets	61
54.	Benutzerlöschung	67

55.	Profil-Mutationen	68
56.	ProfilStats-Mutationen	69
57.	Challengentypen und ihre Datenformate in den dynamischen Attributen	69
58.	Challengentypen und ihre Auswertungsstrategie	70
59.	Infrastruktur	71
60.	Entwicklungs- und Betriebsumgebung	73
61.	AWS Dienste deaktivieren	77
62.	Konfiguration Crowd Test	79
63.	Usability Testziele	80
64.	Usability Testszenarios	81
65.	Usability Test Vorinterview	81
66.	Organisationsstruktur	81
67.	Usability Testpersonen	82
68.	Testszenarien der Api Tests	92
69.	Konfiguration der Api Tests	92
70.	Vorbedingungen einzelner Testphasen	96
71.	Konfiguration End-to-end Tests	96
72.	Kosten gemäss End-to-end Tests	98
73.	Kosten gemäss End-to-end Tests	99
74.	Kosten gemäss Crowd Test	99
75.	UEQ Skalen	102
76.	Engagement Test Vorinterview	102
77.	Engagement Test Nachinterview	102
78.	Engagement SAM Resultate	103
79.	Aufwandschätzung	108
80.	Code Review 1 Fragen und Antworten	131
81.	Code Review 1 Feedback und Massnahmen	133
82.	Code Review 2 Feedback und Massnahmen	135
83.	Tooling/Utils	142
84.	Organisationsstruktur	142
85.	Externe Schnittstellen	143
86.	Zeitraum	143
87.	Zeitbudget	143
88.	Projektphasen	144
89.	Risiken	147
90.	Umgang mit Risiken	149
91.	Definition of Done	152
92.	Issue Namenskonvention	154
93.	Produkt-Repository Branches	156

D.5. Literaturverzeichnis

- [1] “Scrj lakers app,” 23.02.2022. [Online]. Available: https://play.google.com/store/apps/details?id=es.treenovum.swisscehockey.scrjlakers&hl=de_CH&gl=US
- [2] M. Yoshida, B. Gordon, M. Nakazawa, and R. Biscaia, *Conceptualization and Measurement of Fan Engagement: Empirical Evidence From a Professional Sport Context*. Journal of Sport Management, Juli 2014. [Online]. Available: https://www.researchgate.net/profile/Masayuki-Yoshida-3/publication/258421243_Conceptualization_and_Measurement_of_Fan_Engagement_Empirical_Evidence_From_a_Professional_Sport_Context/links/53d711900cf228d363eac9b0/Conceptualization-and-Measurement-of-Fan-Engagement-Empirical-Evidence-From-a-Professional-Sport-Context.pdf
- [3] “Facetten des customer engagements,” 05.05.2022. [Online]. Available: <https://www.zenloop.com/de/ressourcen/retention/facetten-des-customer-engagements>
- [4] “Scb verzeichnet grössten zuschauerrückgang fribourg dank neuem stadion im plus,” 05.06.2022. [Online]. Available: <https://www.watson.ch/sport/eishockey/172840863-zuschauer-zahlen-in-der-national-league-scb-im-minus-gottlron-im-plus>
- [5] M. Bart, “Zuschauerzahlen saison 2021/22,” 2022.
- [6] “Nps for events: why its crucial and how to start using,” 05.06.2022. [Online]. Available: <https://inevent.com/blog/others/nps-for-events-why-its-crucial-how-to-apply.html>
- [7] “Net promoter score,” 05.05.2022. [Online]. Available: <http://www.digitalwiki.de/nps-net-promoter-score/>
- [8] “Nps calculator,” 05.05.2022. [Online]. Available: <https://www.surveymonkey.de/mp/nps-calculator/>
- [9] P. Bitrián, I. Buil, and S. Catalán, *Gamification in sport apps: the determinants of users’ motivation*. European Journal of Management and Business Economics, Mai 2020.
- [10] “Iso 25000 portal,” 10.03.2022. [Online]. Available: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- [11] “Google lighthouse,” 31.05.2022. [Online]. Available: <https://developers.google.com/web/tools/lighthouse>
- [12] “GraphQL is the better rest,” 05.05.2022. [Online]. Available: <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/>
- [13] “Open font license,” 05.06.2022. [Online]. Available: https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
- [14] “Font awesome icons,” 05.06.2022. [Online]. Available: <https://fontawesome.com/icons>
- [15] S. Peuker, “Hcd vorlesung - prototyping,” 2022.
- [16] “Create react app,” 31.05.2022. [Online]. Available: <https://create-react-app.dev/>
- [17] “Aws sdk v2 or v3 - which one should you use?” 05.06.2022. [Online]. Available: <https://dev.to/dvddpl/aws-sdk-v2-or-v3-which-one-should-you-use-3kaj#:~:text=when%20you%20use%20v2%20you,need%2C%20so%20bundle%20is%20smaller.>
- [18] “Redux,” 31.05.2022. [Online]. Available: <https://redux.js.org/introduction/getting-started>
- [19] “Sass 7-1 pattern,” 31.05.2022. [Online]. Available: <https://www.learnhowtoprogram.com/user-interfaces/building-layouts-preprocessors/7-1-sass-architecture>
- [20] “Cypress,” 31.05.2022. [Online]. Available: <https://docs.cypress.io/>

-
- [21] "Google firebase," 05.05.2022. [Online]. Available: <https://firebase.google.com>
- [22] "Aws amplify," 05.05.2022. [Online]. Available: <https://aws.amazon.com/amplify/>
- [23] "Typescript project references," 05.06.2022. [Online]. Available: <https://www.typescriptlang.org/docs/handbook/project-references.html>
- [24] "Webpack," 05.06.2022. [Online]. Available: <https://webpack.js.org>
- [25] "Npm workspaces," 05.06.2022. [Online]. Available: <https://docs.npmjs.com/cli/v7/using-npm/workspaces>
- [26] "Dynamo db pagination," 19.05.2022. [Online]. Available: <https://www.dynamodbguide.com/filtering#breakdown-of-a-dynamodb-api-call>
- [27] "Dynamo db secondary index," 19.05.2022. [Online]. Available: <https://www.dynamodbguide.com/secondary-indexes/>
- [28] "Dynamo db filtering," 19.05.2022. [Online]. Available: <https://www.dynamodbguide.com/filtering>
- [29] "Implementing passwordless email authentication with amazon cognito," 05.06.2022. [Online]. Available: <https://aws.amazon.com/blogs/mobile/implementing-passwordless-email-authentication-with-amazon-cognito/>
- [30] "validationdata not passed for loginwithoutpassword", 05.06.2022. [Online]. Available: <https://github.com/aws-amplify/amplify-js/issues/9136>
- [31] "Quotas in amazon cognito," 05.06.2022. [Online]. Available: <https://docs.aws.amazon.com/cognito/latest/developerguide/limits.html#resource-quotas>
- [32] "Error handling in appsync," 05.06.2022. [Online]. Available: <https://advancedweb.hu/error-handling-in-appsync/>
- [33] "Dynamic group authorization," 05.06.2022. [Online]. Available: <https://docs.amplify.aws/cli/graphql/authorization-rules/#user-group-based-data-access>
- [34] "Add support for self-hosted gitlab(ce) or github enterprise," 05.06.2022. [Online]. Available: <https://github.com/aws-amplify/amplify-hosting/issues/14>
- [35] "What is amplify," 06.03.2022. [Online]. Available: https://youtu.be/HkbjHtG_d7w?t=379
- [36] "Team workflows with amplify backend environments," 05.06.2022. [Online]. Available: <https://docs.aws.amazon.com/amplify/latest/userguide/team-workflows-with-amplify-cli-backend-environments.html>
- [37] "Aws budget killswitch," 05.06.2022. [Online]. Available: <https://dev.to/leonti/aws-budget-killswitch-disable-aws-services-when-budget-is-exceeded-36oc>
- [38] "Grafana," 05.06.2022. [Online]. Available: <https://grafana.com>
- [39] "Operating lambda performance optimization," 05.06.2022. [Online]. Available: <https://aws.amazon.com/blogs/compute/operating-lambda-performance-optimization-part-1/>
- [40] M. Lombard, R. D. Reich, M. E. Grabe, C. C. Bracken, and T. B. Ditton, *Presence and Television - The Role of Screen Size*. International Communication Association, Januar 2000. [Online]. Available: https://www.researchgate.net/publication/227603901_Presence_and_television
- [41] "Sam self-assessment manikin," 05.06.2022. [Online]. Available: <https://germanupa.de/wissen/fragebogenmatrix/sam>
- [42] "Werkzeugkasten sam," 05.06.2022. [Online]. Available: https://www.kompetenzzentrum-usability.digital/angebote/materialien/werkzeugkasten?pagenr=48#compilationObject_1870
- [43] "User experience questionnaire," 05.06.2022. [Online]. Available: <https://www.ueq-online.org>
- [44] "Batchwriteitem," 05.06.2022. [Online]. Available: https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchWriteItem.html
-

E. Weiterführende Dokumente

In diesem Abschnitt werden ergänzende Produkte integriert, die im Rahmen der Arbeit erstellt wurden.

E.1. Admin Panel Anleitung

Gemäss den Nichtfunktionalen Anforderungen wurde eine Anleitung für den Regiebereich des Tippspiels entwickelt. Diese wurde an Michael Bart übergeben und beginnt auf der nächsten Seite.



SCRJ Lakers Tippspiel

Benutzeranleitung Administrator

Autoren:	Simon Hager Philipp Emmenegger Joel Schaltegger
Version:	1.0
Datum:	13. Juni 2022

Inhaltsverzeichnis

1	Vorwort	3
2	Zugang	4
3	Challenges	5
3.1	Challenge erstellen	5
3.2	Challenge bearbeiten	5
3.3	Challenge löschen	6
4	Sponsoren	7
4.1	Sponsor erstellen	7
4.2	Sponsor bearbeiten	7
4.3	Sponsor löschen	7
5	Preise	8
5.1	Preis erstellen	8
5.2	Preis bearbeiten	8
5.3	Preis löschen	8
6	Kader	9
6.1	Spieler erstellen	9
6.2	Spieler bearbeiten	9
6.3	Spieler löschen	9
7	Tippspiel	10
7.1	Tippspiel vorbereiten	10
7.2	Challenge Status	15

1 Vorwort

Mit dieser Anleitung wird dem Leser einen Überblick über die Funktionen und die Bedienung des Admin Panels gegeben. Zu jeder Benutzeroberfläche sind die einzelnen Bedienungsmöglichkeiten beschrieben. Zudem soll sich der Leser seiner Verantwortung als Administrator bewusst werden und erfahren, was er mit seinen Aktionen bewirkt.

2 Zugang

Das Admin Portal kann über die URL <https://game.lakers.ch/admin> geöffnet werden. Es empfiehlt sich dafür einen Desktop-Computer zu nutzen, da die Bedienung nicht auf Smartphones ausgelegt ist.

3 Challenges

Unter dem Menüpunkt «Challenges» werden alle Fragen aufgelistet, welche später für ein Tippspiel verwendet werden können. Jede Challenge benötigt sowohl eine Frage, als auch Punkte, welche bei korrekter Tippabgabe vergeben werden. Schwierigere Fragen sollten entsprechend mehr Punkte geben als einfachere.

Einschränkung: Aktuell können Challenges nur mit einem Spieler als Antwort aufgelöst werden.

3.1 Challenge erstellen



1. Unterhalb der Tabelle auf den Button «Challenge erstellen» klicken

Challenge hinzufügen

Beschreibung:
 2.

Frage: *
 3.

Punkte: *
 4.

5.

2. Optionale Beschreibung hinzufügen (Wird im Tippspiel auf der Detail-Seite der Challenge dargestellt)
 Beispiel: Die Abstimmung schliesst bei Spielstart
3. Frage eintippen
4. Punkte definieren (Gemäss Schwierigkeit der Frage)
5. Erstellen

3.2 Challenge bearbeiten



1. Auf den Button «Bearbeiten» klicken

Challenge bearbeiten

Beschreibung:
 2.

Frage: *

Punkte: *
 2.

3.

2. Gewünschte Angaben bearbeiten
3. Speichern

3.3 Challenge löschen

Verfügbare Challenges

- Welcher Spieler bricht den ersten Stock? Bearbeiten
- Welcher Spieler erzielt das nächste Tor? **2.** Bearbeiten
- Welcher Spieler bekommt die erste Strafe? Bearbeiten

[+ Challenge erstellen](#)

1. Auf den Button «Bearbeiten» klicken

Challenge bearbeiten

Beschreibung:

Frage: *

Punkte: *

2.

Abbrechen Löschen Speichern

2. Löschen

4 Sponsoren

Für die Sponsoren gilt das gleiche Prinzip wie bei den Challenges. Sie können im Menüpunkt ‹Sponsoren› erfasst und dann beim Tippspiel ausgewählt werden. Besonders zu beachten ist hier das Hochladen von Bildern: Das Logo eines Sponsors wird mit einem Seitenverhältnis von 5:2 dargestellt. Wenn das ausgewählte Bild ein anderes Verhältnis hat, wird es automatisch darauf zugeschnitten. Die Bildvorschau zeigt an, wie es im Tippspiel dargestellt wird. Die Logos werden maximal mit einer Breite von 300 Pixel verwendet. Sind sie beim Upload grösser, werden sie komprimiert.

4.1 Sponsor erstellen



1. Auf die Kachel ‹Sponsor erstellen› klicken

Sponsor hinzufügen

Sponsor Name: * 2.

Webseite: * 3.

Bildvorschau 4.
Bild hochladen
Per Klick oder Drag&Drop

Abbrechen 5. Erstellen

2. Name des Sponsors angeben
3. Webseite des Sponsors eintragen
4. Bild hochladen (Seitenverhältnis 5:2 empfohlen)
5. Erstellen

4.2 Sponsor bearbeiten



1. Auf den entsprechenden Sponsor klicken

Sponsor bearbeiten

Sponsor Name: * 2.

Webseite: * 2.

Bildvorschau 2.
GEBERIT

Bild ändern

Abbrechen 3. Löschen Speichern

2. Gewünschte Angaben bearbeiten
3. Speichern

4.3 Sponsor löschen



1. Auf den entsprechenden Sponsor klicken

Sponsor bearbeiten

Sponsor Name: *

Webseite: *

Bildvorschau
GEBERIT

Bild ändern

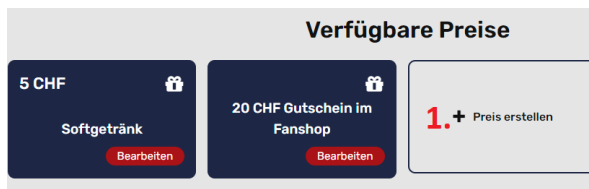
Abbrechen 2. Löschen Speichern

2. Löschen

5 Preise

Im Menüpunkt <Preise> werden die vorhandenen Preise erfasst. Ein Preis kann optional mit einem Wert gekennzeichnet werden, welcher aber nur dem Administrator angezeigt wird. Dies hilft dem Administrator zum späteren Zeitpunkt, wenn er die Preise eines Tippspiels konfiguriert. Für die Vergabe der Preise im Tippspiel, muss der Administrator eine Reihenfolge angeben, wobei er den Wert als Anhaltspunkt nehmen kann.

5.1 Preis erstellen



1. Auf die Kachel <Preis erstellen> klicken

Preis hinzufügen

Name: **2.**

Wert: **3.**

4.

Abbrechen Erstellen

2. Name des Preises definieren
3. Optionaler Wert angeben
4. Erstellen

5.2 Preis bearbeiten



1. Auf den entsprechenden Preis klicken

Preis bearbeiten

Name: **2.**

Wert: **2.**

3.

Abbrechen Löschen Speichern

2. Gewünschte Angaben bearbeiten
3. Speichern

5.3 Preis löschen



1. Auf den entsprechenden Preis klicken

Preis bearbeiten

Name: *

Wert:

2.

Abbrechen Löschen Speichern

2. Löschen

6 Kader

Im Menüpunkt ‹Kader› können die Spieler der aktuellen Saison erfasst werden. Sie sind nach Position (Goalkeeper, Verteidiger, Stürmer) und danach aufsteigend nach deren Nummer sortiert. Wie bei den Sponsoren wird ein Bild benötigt. In diesem Fall sollte das Portrait ein Seitenverhältnis von 4:5 haben. Auch hier wird das hochgeladene Bild automatisch zugeschnitten. Die Bildvorschau zeigt dem Administrator, wie es später im Tippspiel dargestellt wird. Die Portraits werden falls nötig auf eine Breite 80 Pixel komprimiert.

6.1 Spieler erstellen

The screenshot shows the 'Kader' management interface. At the top, there are three categories: Goalkeeper, Verteidiger, and Stürmer. Below these, there is a list of players. The first player shown is Cervenka Roman, #10. A red button labeled 'Spieler hinzufügen' with a red '1' is positioned below the player list.

1. Zuerst auf der Seite auf ‹Spieler hinzufügen› klicken

Spieler hinzufügen

The screenshot shows the 'Spieler hinzufügen' form. It includes input fields for 'Vorname' (Andrew), 'Nachname' (Rowe), and 'Nummer' (19). There are radio buttons for 'Position' (Torhüter, Verteidiger, Stürmer) with 'Stürmer' selected. A 'Bildvorschau' section shows a preview of the player's portrait. A red button 'Bild ändern' is next to the preview. At the bottom, there are 'Abbrechen' and 'Erstellen' buttons. Red numbers 2 through 6 are overlaid on the form to indicate the steps: 2. Vor- und Nachname eintragen, 3. Rückennummer angeben, 4. Position bestimmen, 5. Bild hochladen (Format: 80x100 Pixel), 6. Erstellen.

2. Vor- und Nachname eintragen
3. Rückennummer angeben
4. Position bestimmen
5. Bild hochladen (Format: 80x100 Pixel)
6. Erstellen

6.2 Spieler bearbeiten

The screenshot shows the 'Kader' management interface. At the top, there are three categories: Goalkeeper, Verteidiger, and Stürmer. Below these, there is a list of players. The first player shown is Cervenka Roman, #10. The second player shown is Rowe Andrew, #19. A red button labeled 'Spieler bearbeiten' with a red '1' is positioned below the player list.

1. Auf den entsprechenden Spieler klicken

Spieler bearbeiten

The screenshot shows the 'Spieler bearbeiten' form. It includes input fields for 'Vorname' (Andrew), 'Nachname' (Rowe), and 'Nummer' (54). There are radio buttons for 'Position' (Torhüter, Verteidiger, Stürmer) with 'Verteidiger' selected. A 'Bildvorschau' section shows a preview of the player's portrait. A red button 'Bild ändern' is next to the preview. At the bottom, there are 'Abbrechen', 'Löschen', and 'Speichern' buttons. Red numbers 2 and 3 are overlaid on the form to indicate the steps: 2. Gewünschte Angaben bearbeiten, 3. Speichern.

2. Gewünschte Angaben bearbeiten
3. Speichern

6.3 Spieler löschen

The screenshot shows the 'Kader' management interface. At the top, there are three categories: Goalkeeper, Verteidiger, and Stürmer. Below these, there is a list of players. The first player shown is Cervenka Roman, #10. The second player shown is Rowe Andrew, #19. A red button labeled 'Spieler löschen' with a red '1' is positioned below the player list.

1. Auf den entsprechenden Spieler klicken

Spieler bearbeiten

The screenshot shows the 'Spieler bearbeiten' form. It includes input fields for 'Vorname' (Andrew), 'Nachname' (Rowe), and 'Nummer' (19). There are radio buttons for 'Position' (Torhüter, Verteidiger, Stürmer) with 'Stürmer' selected. A 'Bildvorschau' section shows a preview of the player's portrait. A red button 'Bild ändern' is next to the preview. At the bottom, there are 'Abbrechen', 'Löschen', and 'Speichern' buttons. A red number 2 is overlaid on the form to indicate the step: 2. Löschen.

2. Löschen

7 Tippspiel



Beim der Verwaltung eines Tippspiels hat man eine grosse Verantwortung. Wenn ein Tipp-
spiel gestartet wird, wird eine Push-Nachricht ausgelöst. Zudem wird jede Challenge-Status
Änderung sofort bei allen Geräten aktualisiert. Dieser Bereich ist also nicht zum experi-
mentieren gedacht!

7.1 Tippspiel vorbereiten

Als erster Schritt muss der Administrator das Tippspiel vorbereiten. Während dieser Vorbereitung ist
das Tippspiel noch nicht aktiv und wird im App noch nicht dargestellt.

7.1.1 Tippspiel erstellen



1. Falls noch kein Tippspiel aktiv ist, auf den But-
ton <Spiel erstellen> klicken

7.1.2 Tippspiel Übersicht



Erklärung:

1. Optionale Spielbeschreibung wie z.B. (SCRJ vs. ZSC), wird auf der Tippspiel-Seite dargestellt
2. Challenges des Tippspiels hinzufügen/löschen
3. Sponsor des Tippspiels festlegen
4. Zu gewinnende Preise auswählen (werden am Ende des Tippspiels verlost)
5. Spieleraufgebot angeben
6. Button um das Tippspiel zu starten/beenden
7. Ausgewählte Challenges auf ihren Status verteilt

Hinweis: Challenges, Sponsor und Preise können auch während einem laufendem Tippspiel bearbeitet
werden. Das Aufgebot nicht.

7.1.3 Spielbeschreibung

Die Spielbeschreibung ist optional und wird im Tippspiel bei der Übersicht der Challenges angezeigt.

TODO: evtl Screenshot der Beschreibung im Tippspiel ergänzen

Beschreibung hinzufügen/ändern:

1. Beschreibung eintippen/anpassen
2. Speichern

Beschreibung löschen:

1. Text löschen (leerer Text)
2. Speichern

7.1.4 Challenges bearbeiten

Wählt man eine Challenge neu aus, wird sie dem Tippspiel hinzugefügt und beim Status versteckt aufgelistet. Versteckt bedeutet, dass sie nur im Admin Panel angezeigt wird. Im Tippspiel werden nur die offenen, geschlossenen und aufgelösten dargestellt.

Einschränkung: Jede Challenge kann nur einmal verwendet werden.

Wichtig: Entfernt man den Haken einer bereits ausgewählten Challenge, wird diese aus dem Tippspiel entfernt und bereits abgegebene Tipps gelöscht.

Dem Tippspiel eine Challenge hinzufügen:

1. Auf den Button <Challenges bearbeiten> klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Fragen aus dem Menüpunkt <Challenges>

Challenges bearbeiten **2.**

Welcher Spieler bricht den ersten Stock?	100 Punkte	<input type="checkbox"/>
Welcher Spieler erzielt das nächste Tor?	250 Punkte	<input checked="" type="checkbox"/>
Welcher Spieler bekommt die erste Strafe?	170 Punkte	<input checked="" type="checkbox"/>

Abbrechen Speichern

3. Eine Challenge ohne Haken anklicken
4. Speichern

Eine Challenge aus dem Tippspiel entfernen:

1. Auf den Button <Challenges bearbeiten> klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Fragen aus dem Menüpunkt <Challenges>

Challenges bearbeiten **2.**

Welcher Spieler bricht den ersten Stock?	100 Punkte	<input type="checkbox"/>
Welcher Spieler erzielt das nächste Tor?	250 Punkte	<input checked="" type="checkbox"/>
Welcher Spieler bekommt die erste Strafe?	170 Punkte	<input checked="" type="checkbox"/>

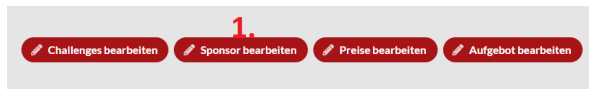
Abbrechen Speichern

3. Eine Challenge mit Haken anklicken
- Achtung:** Bereits abgegebene Tipps von Fans werden dabei gelöscht
4. Speichern

7.1.5 Sponsor bearbeiten

Der Sponsor ist optional und wird im Tippspiel bei jeder Challenge angezeigt.

Dem Tippspiel einen Sponsor hinzufügen/ändern:



1. Auf den Button «Sponsor bearbeiten» klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Sponsoren aus dem Menüpunkt «Sponsoren»

Einen Sponsor aus dem Tippspiel entfernen:



1. Auf den Button «Sponsor bearbeiten» klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Sponsoren aus dem Menüpunkt «Sponsoren»

Sponsor bearbeiten 2.

Hostpoint AG 3.

St.Galler Kantonalbank

Geberit



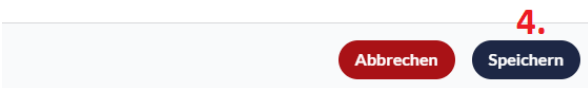
3. Einen Sponsor auswählen
4. Speichern

Sponsor bearbeiten

Hostpoint AG 3. ✓

St.Galler Kantonalbank

Geberit

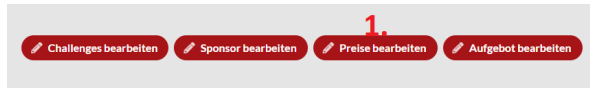


3. Bestehenden Sponsor anklicken und somit deaktivieren
4. Speichern

7.1.6 Preise bearbeiten

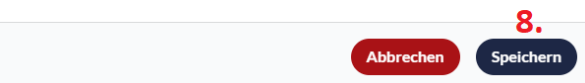
Die Preise im Tippspiel sind optional. Ausgewählte Preise haben eine Priorität und werden entsprechend verteilt. Der Preis mit der höchsten Priorität (1. Prio) wird gemäss seiner Stückzahl an die Fans mit den meisten Punkten verteilt. Darauf folgen die Preise mit den tieferen Prioritäten.

Dem Tippspiel Preise hinzufügen/bearbeiten



1. Auf den Button <Preise bearbeiten> klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Preisen aus dem Menüpunkt <Preise>

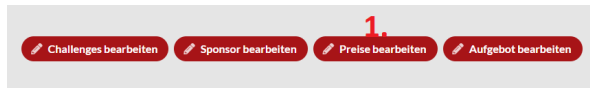
Preise bearbeiten 2.



3. Preis mit Priorität 1 auswählen
4. Anzahl festlegen mit den <Plus/Minus> Buttons
5. Preis mit Priorität 2 auswählen
6. Anzahl festlegen mit den <Plus/Minus> Buttons
7. Analog mit weiteren Preisen
8. Speichern

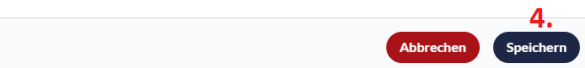
Hinweis: Wird der Haken bei einem Preis entfernt, rücken die Prioritäten der anderen Preisen nach

Einen Preis aus dem Tippspiel entfernen:



1. Auf den Button <Preise bearbeiten> klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Preisen aus dem Menüpunkt <Preise>

Preise bearbeiten

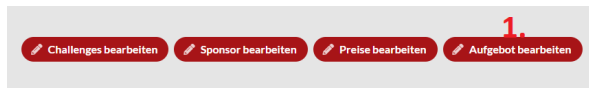


3. Eine Challenge mit Haken anklicken
4. Speichern

7.1.7 Aufgebot bearbeiten

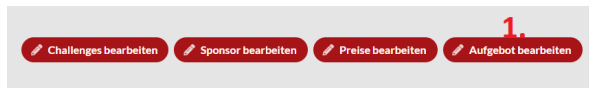
Das Aufgebot muss zwingend in der Spielvorbereitung definiert werden. Sobald das Tippspiel gestartet wurde, kann es nicht mehr angepasst werden.

Spieler dem Aufgebot hinzufügen:



1. Auf den Button <Aufgebot bearbeiten> klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Spielern aus dem Menüpunkt <Kader>

Einen Spieler aus dem Aufgebot entfernen:



1. Auf den Button <Aufgebot bearbeiten> klicken
2. Es öffnet sich eine Liste mit allen vorhandenen Spielern aus dem Menüpunkt <Kader>

Aufgebot bearbeiten 2.

3. Spieler anklicken
4. Speichern

Aufgebot bearbeiten 2.

3. Zu entfernende Spieler anklicken und somit de-
elektieren
4. Speichern

7.1.8 Tippspiel starten:

Wird das Tippspiel gestartet, zeigt es dieses in der App an. Es empfiehlt sich zu diesem Zeitpunkt bereits eine offene Challenge zu haben. - Ansonsten ist das Tippspiel beim Fan leer.

Um das Tippspiel zu starten muss man oben rechts auf den blauen Button <Spiel starten> klicken. Falls das Aufgebot nicht ausgewählt ist, wird dieser Vorgang mit einer Meldung verhindert. Fehlen andere Konfigurationen, wird mit einer Warnung darauf hingewiesen. Das Spiel kann durch eine Kenntnisnahme dennoch gestartet werden.

7.1.9 Tippspiel beenden:

Mit einem Klick auf den <Spiel beenden> Button kann das Tippspiel beendet werden. Diese irreversible Aktion muss bestätigt werden. Dadurch werden alle Preise an die Teilnehmer verteilt und die Darstellung des Tippspiels verschwindet sowohl im Admin Portal und in der Fan Ansicht.

7.2 Challenge Status

7.2.1 Versteckt

Versteckte Challenges werden nur im Admin Portal dargestellt. Wenn im Tippspiel eine Challenge hinzugefügt wird, bekommt sie diesen Status. - So können versehentlich aktivierte Challenges unbemerkt wieder entfernt werden.

Klickt man bei einer versteckten Challenge auf den Button <Starten>, wird sie offen und für die Fans sichtbar.

7.2.2 Offen

Bei offenen Challenges können die Fans Tipps abgeben und ihre Auswahl ändern. Sobald der Administrator auf den Button <Schliessen> klickt ändert sie der Status zu Geschlossen.

7.2.3 Geschlossen

Geschlossene Challenges sind Fragen, wofür keine Tipps mehr abgegeben werden können, aber die Lösung noch nicht gesetzt ist.

Wenn man auf den Button <Auflösen> klickt muss vom Administrator die Lösung ausgewählt werden.

Besonderheit: Falls das gefragte Ereignis nicht eingetroffen ist (wie z.B. Wer schießt das erste Tor? bei einem zu Null Spiel), muss als Lösung <Niemand> ausgewählt werden.

7.2.4 Aufgelöst

Aufgelöste Challenges können ihren Status nicht mehr ändern. Sie sind bereits abgeschlossen und die Punkte wurden an die Fans verteilt.

E.2. Storyboard Visualisierung

Storyboards

Ziel:

- Konzept vorstellen
- Diskussion anregen
- Abläufe visualisieren

**Definieren nicht:**

- Design
- Inhalte



Storyboards

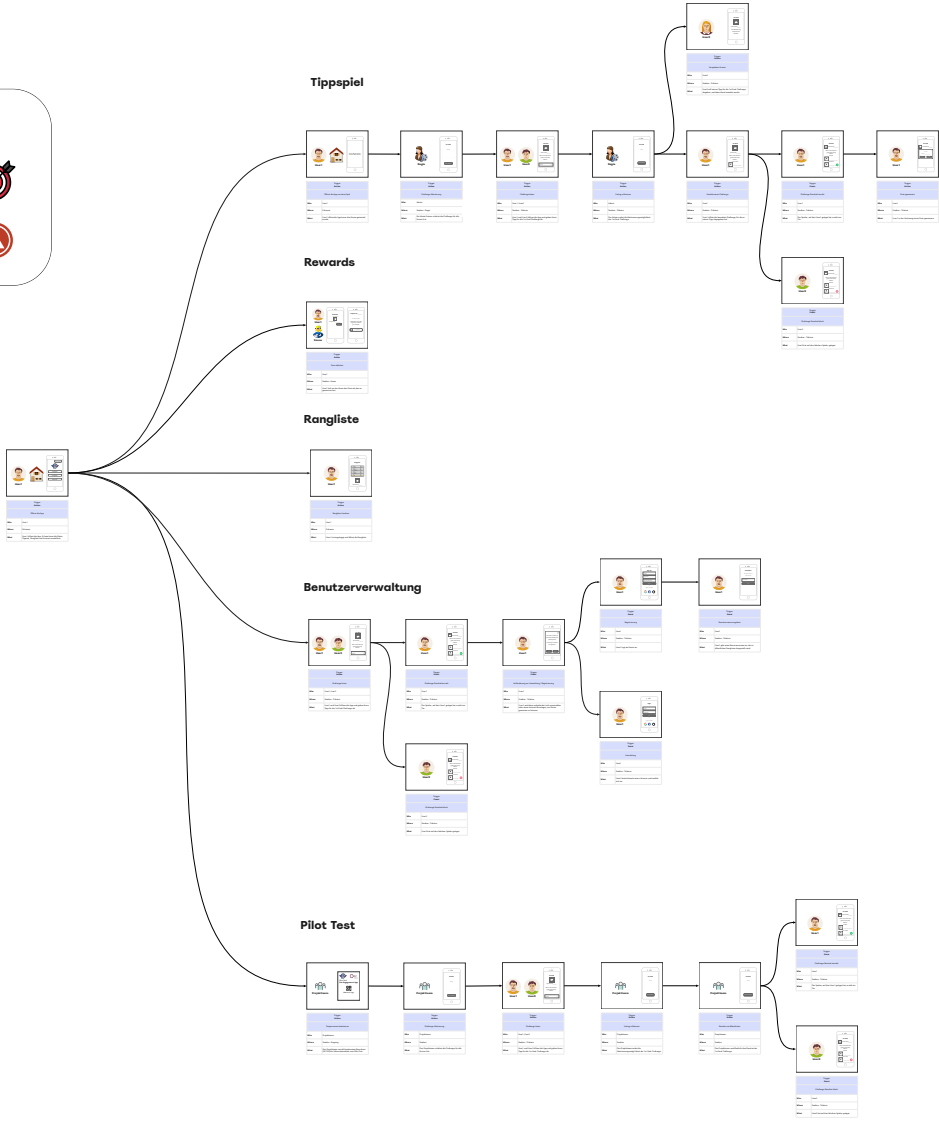
Ziel:

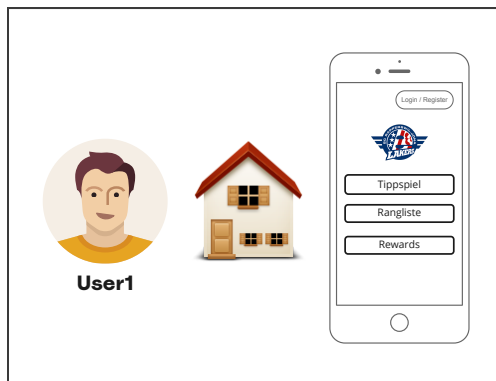
- Konzept vorstellen
- Diskussion anregen
- Abläufe visualisieren

Definieren nicht:

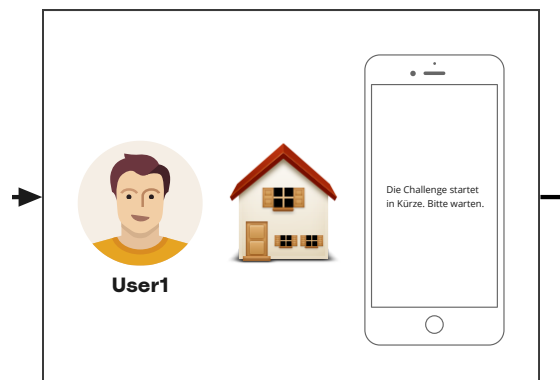
- Design
- Inhalte







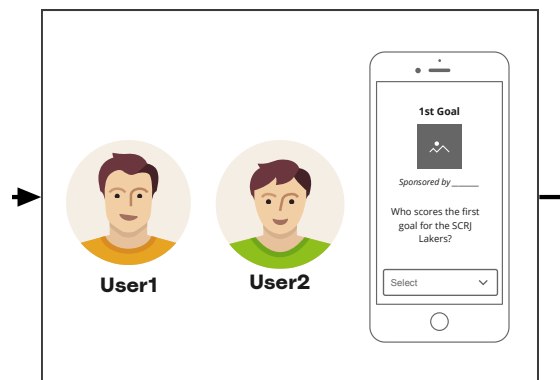
Trigger Action	
Öffnet die App	
Who	User1
Where	Zuhause
What	User1 öffnet die App. Er kann kann die Reiter Tippspiel, Rangliste und Account auswählen.



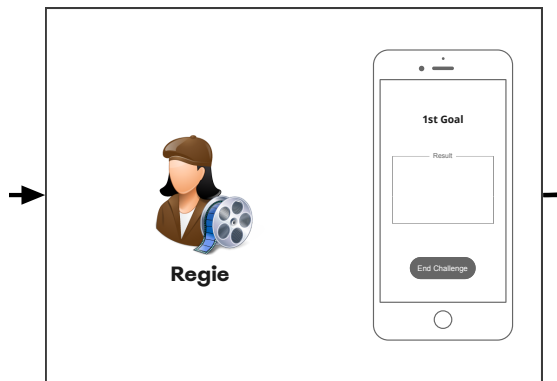
Trigger Action	
Öffnet die App vor dem Spiel	
Who	User1
Where	Zuhause
What	User1 öffnet die App bevor das Game gestartet wurde.



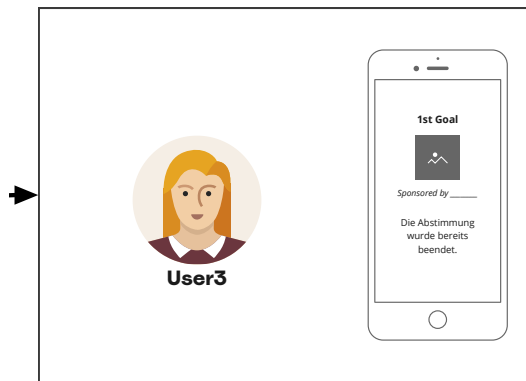
Trigger Action	
Challenge Aktivierung	
Who	Admin
Where	Stadion - Regie
What	Ein Admin Nutzer schaltet die Challenge für alle Nutzer frei.



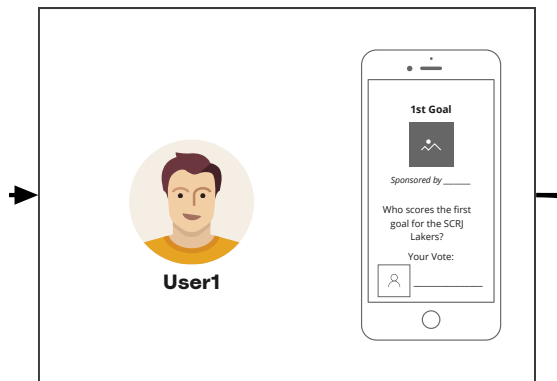
Trigger Action	
Challenge lösen	
Who	User1, User2
Where	Stadion - Tribüne
What	User1 und User2 öffnen die App und geben ihren Tipp für die 1st Goal Challenge ab.



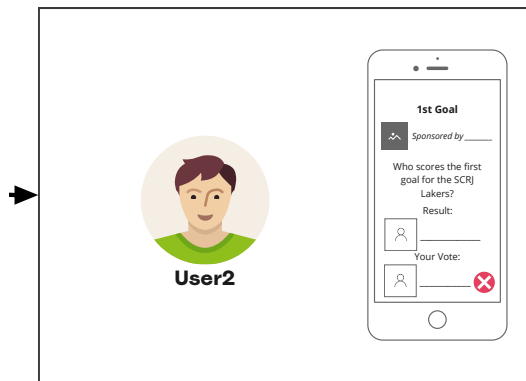
Trigger Action	
Voting schliessen	
Who	Admin
Where	Stadion - Tribüne
What	Der Admin endet die Abstimmungsmöglichkeit der 1st Goal Challenge.



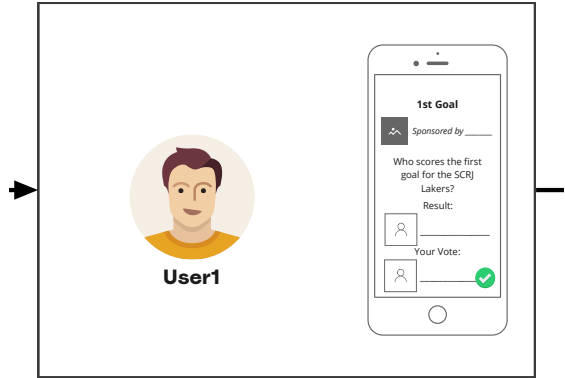
Trigger Action	
Verspäteter Nutzer	
Who	User3
Where	Stadion - Tribüne
What	User3 will seinen Tipp für die 1st Goal Challenge abgeben, nachdem diese beendet wurde.



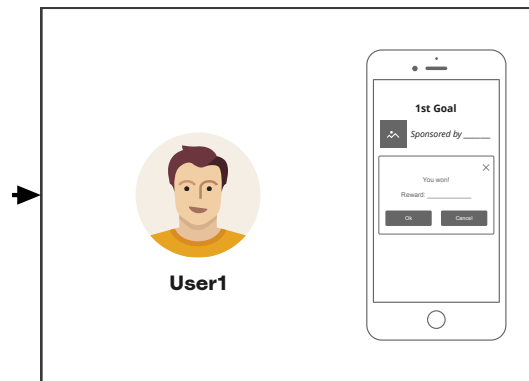
Trigger Action	
Geschlossene Challenge	
Who	User1
Where	Stadion - Tribüne
What	User1 öffnet die beendete Challenge, für die er seinen Tipp abgegeben hat.



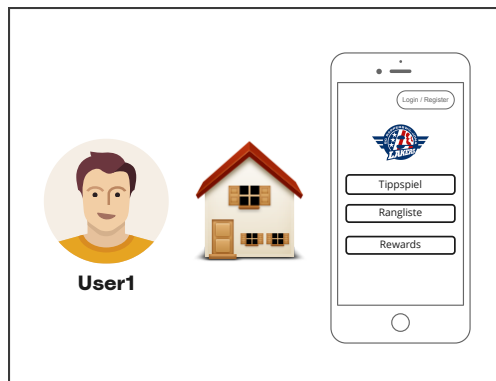
Trigger Event	
Challenge Resultat falsch	
Who	User2
Where	Stadion - Tribüne
What	User2 hat auf den falschen Spieler getippt.



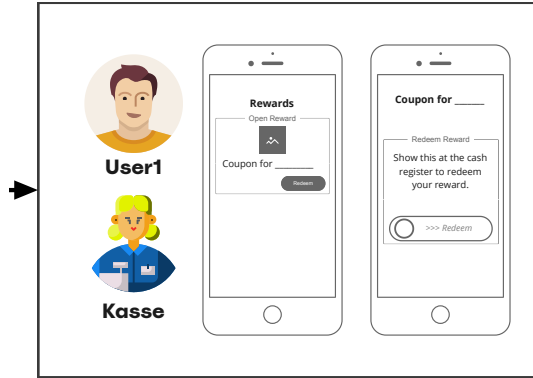
Trigger Event	
Challenge Resultat korrekt	
Who	User1
Where	Stadion - Tribüne
What	Der Spieler, auf den User1 getippt hat, erzielt ein Tor.



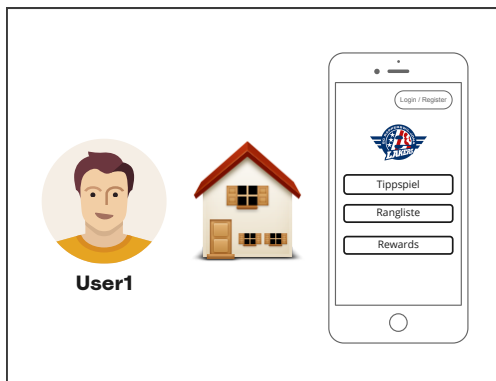
Trigger Action	
Preis gewonnen	
Who	User1
Where	Stadion - Tribüne
What	User1 in der Verlosung einen Preis gewonnen.



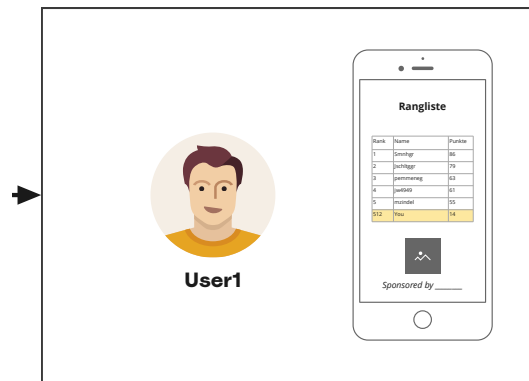
Trigger Action	
Öffnet die App	
Who	User1
Where	Zuhause
What	User1 öffnet die App. Er kann kann die Reiter Tippspiel, Rangliste und Account auswählen.



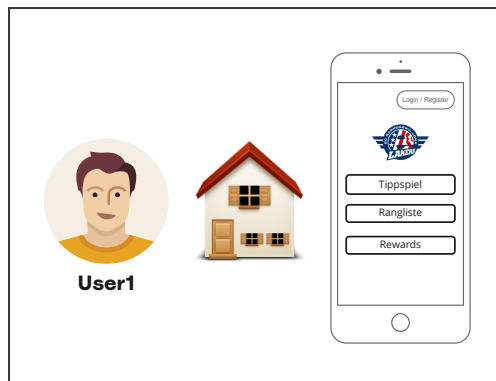
Trigger Action	
Preis abholen	
Who	User1
Where	Stadion - Kasse
What	User1 holt an der Kasse den Preis ab, den er gewonnen hat.



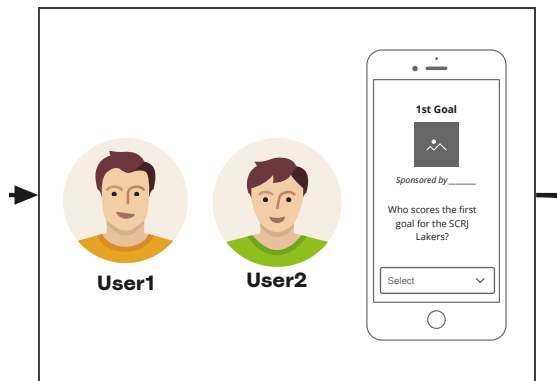
Trigger Action	
Öffnet die App	
Who	User1
Where	Zuhause
What	User1 öffnet die App. Er kann kann die Reiter Tippspiel, Rangliste und Account auswählen.



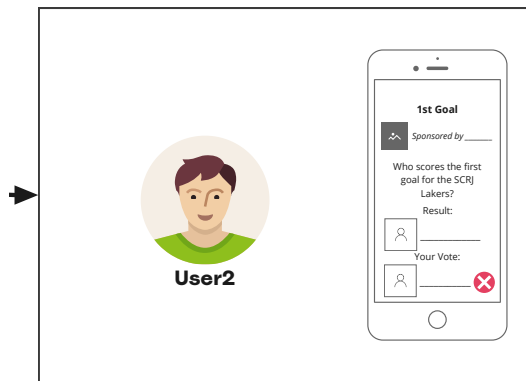
Trigger Action	
Rangliste Ansehen	
Who	User1
Where	Zuhause
What	User1 ist eingeloggt und öffnet die Rangliste



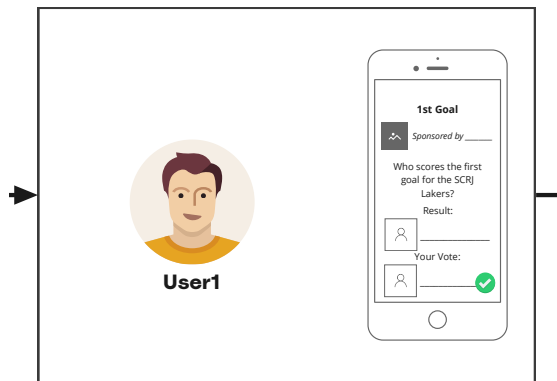
Trigger Action	
Öffnet die App	
Who	User1
Where	Zuhause
What	User1 öffnet die App. Er kann kann die Reiter Tippspiel, Rangliste und Account auswählen.



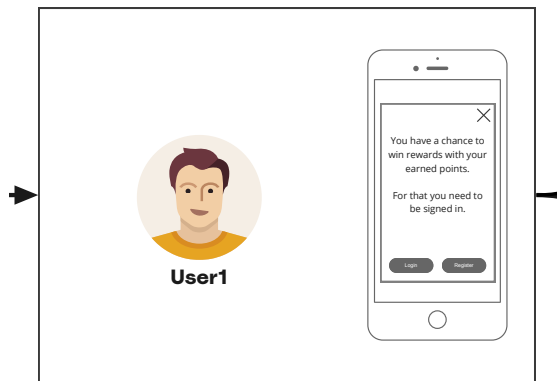
Trigger Action	
Challenge lösen	
Who	User1, User2
Where	Stadion - Tribüne
What	User1 und User2 öffnen die App und geben ihren Tipp für die 1st Goal Challenge ab.



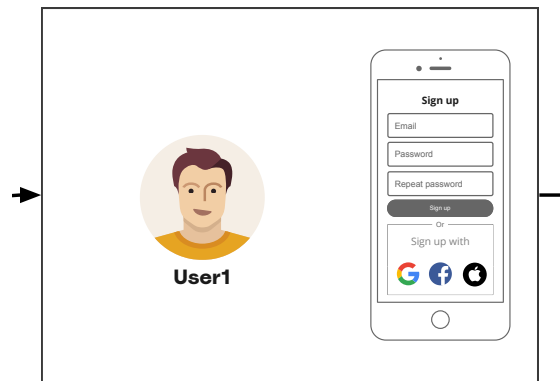
Trigger Event	
Challenge Resultat falsch	
Who	User2
Where	Stadion - Tribüne
What	User2 hat auf den falschen Spieler getippt.



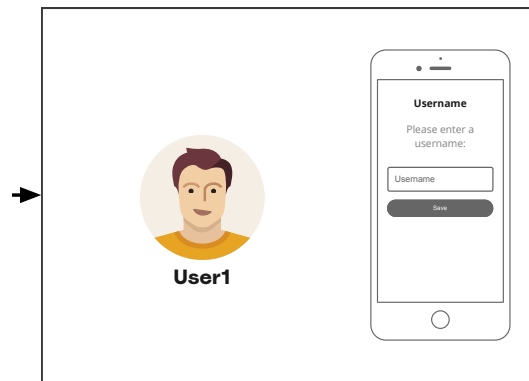
Trigger Event	
Challenge Resultat korrekt	
Who	User1
Where	Stadion - Tribüne
What	Der Spieler, auf den User1 getippt hat, erzielt ein Tor.



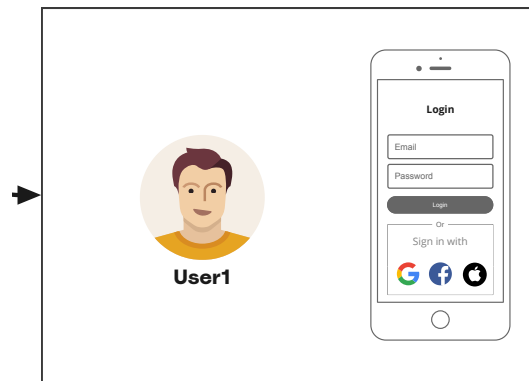
Trigger Event	
Aufforderung zur Anmeldung / Registrierung	
Who	User1
Where	Stadion - Tribüne
What	User1 wird dazu aufgefordert sich anzumelden oder einen Account anzulegen, um Preise gewinnen zu können.



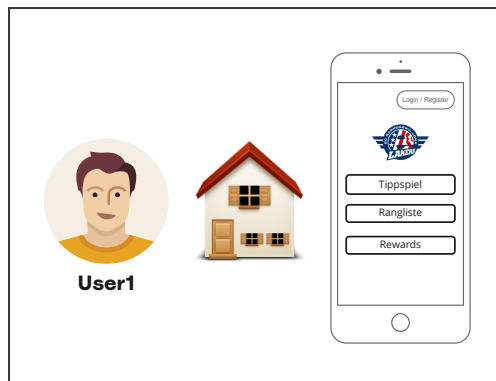
Trigger Event	
Registrierung	
Who	User1
Where	Stadion - Tribüne
What	User1 legt ein Konto an.



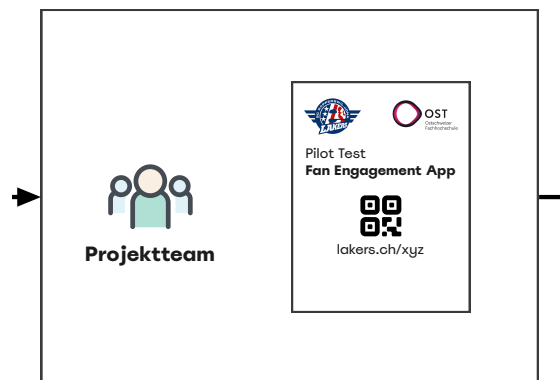
Trigger Event	
Benutzername angeben	
Who	User1
Where	Stadion - Tribüne
What	User1 gibt einen Benutzernamen an, der in öffentlichen Ranglisten dargestellt wird.



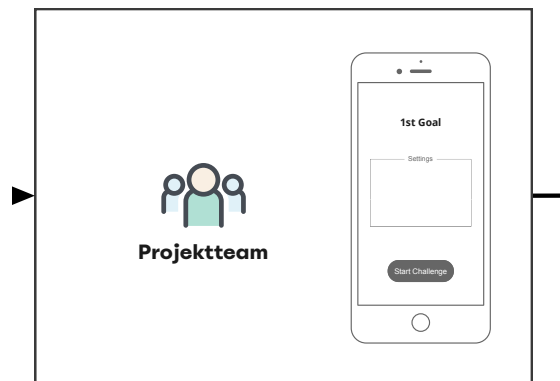
Trigger Event	
Anmeldung	
Who	User1
Where	Stadion - Tribüne
What	User1 besitzt bereits einen Account und meldet sich an.



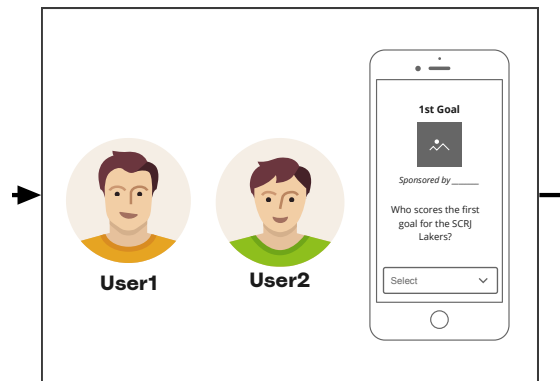
Trigger Action	
Öffnet die App	
Who	User1
Where	Zuhause
What	User1 öffnet die App. Er kann kann die Reiter Tippspiel, Rangliste und Account auswählen.



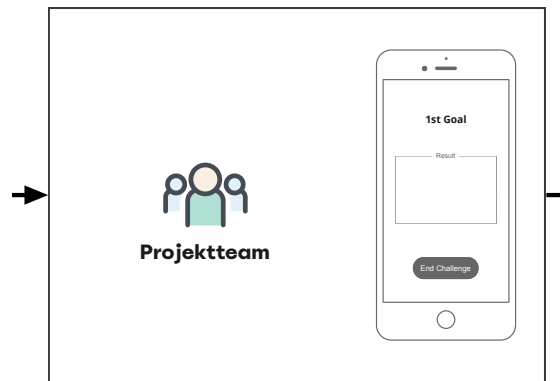
Trigger Action	
Testpersonen bestimmen	
Who	Projektteam
Where	Stadion - Eingang
What	Das Projektteam verteilt bestimmten Besuchern (50-100) ein Informationsblatt zum Pilot Test.



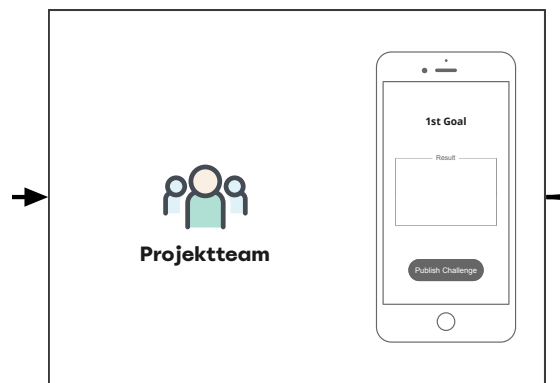
Trigger Action	
Challenge Aktivierung	
Who	Projektteam
Where	Stadion
What	Das Projektteam schaltet die Challenge für alle Nutzer frei.



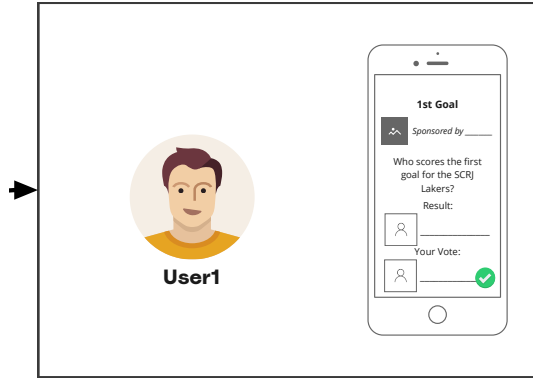
Trigger Action	
Challenge lösen	
Who	User1, User2
Where	Stadion - Tribüne
What	User1 und User2 öffnen die App und geben ihren Tipp für die 1st Goal Challenge ab.



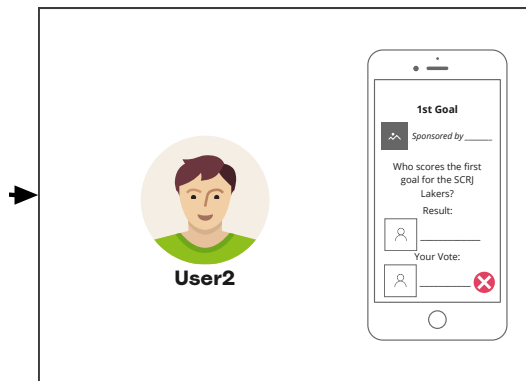
Trigger Action	
Voting schliessen	
Who	Projektteam
Where	Stadion
What	Das Projektteam endet die Abstimmungsmöglichkeit der 1st Goal Challenge.



Trigger Action	
Resultat veröffentlichen	
Who	Projektteam
Where	Stadion
What	Das Projektteam veröffentlicht das Resultat der 1st Goal Challenge.

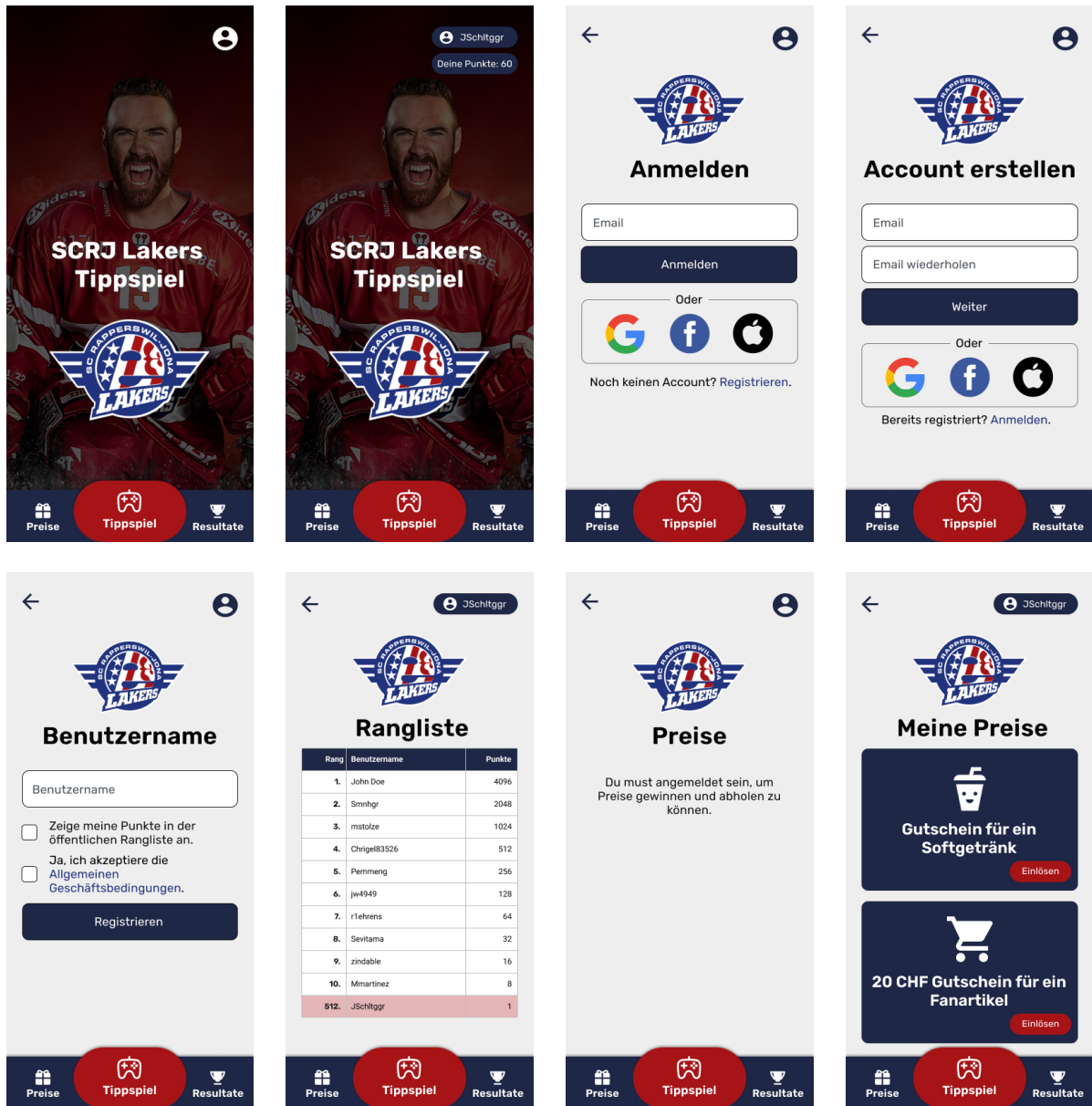


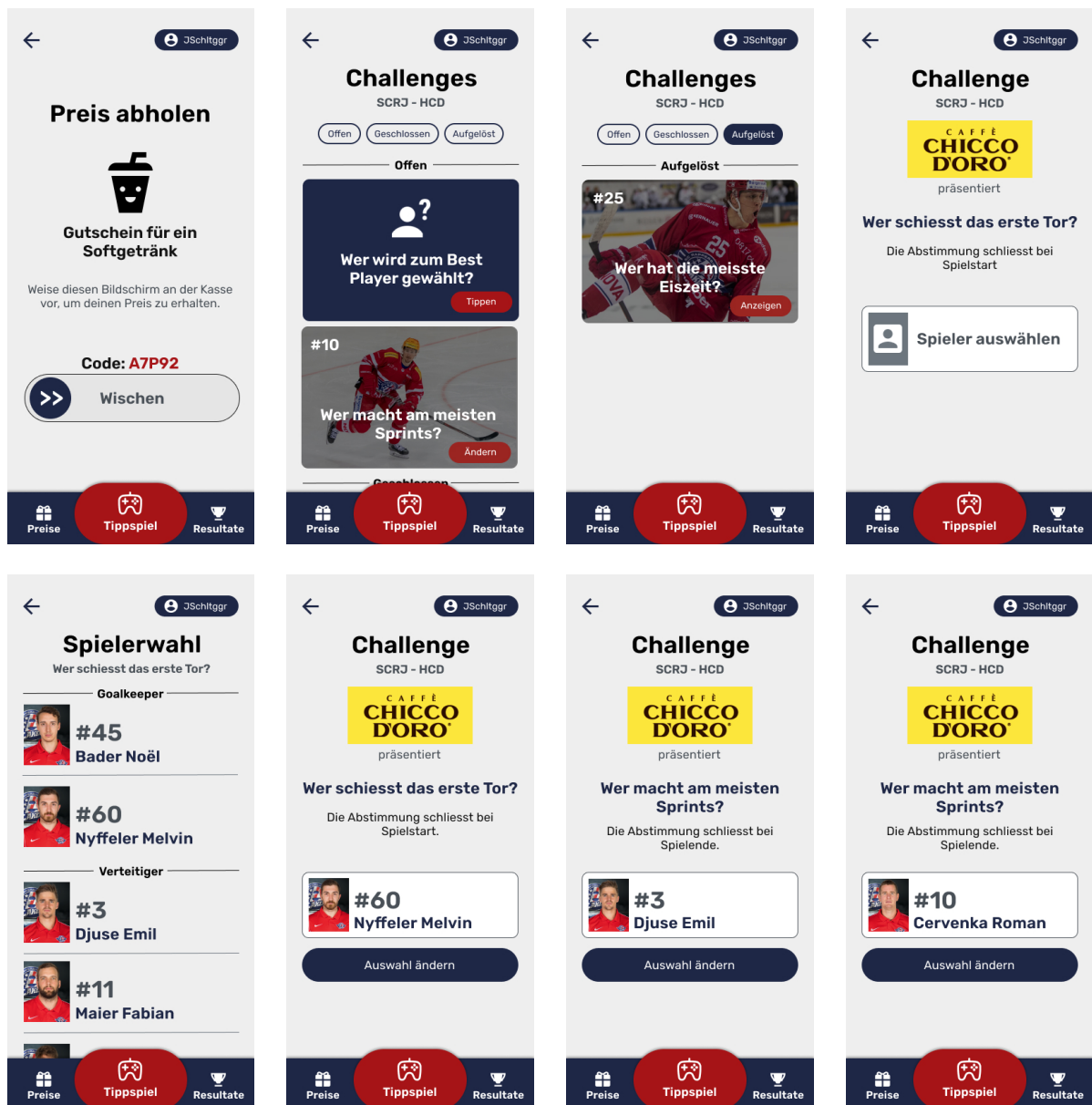
Trigger Event	
Challenge Resultat korrekt	
Who	User1
Where	Stadion - Tribüne
What	Der Spieler, auf den User1 getippt hat, erzielt ein Tor.

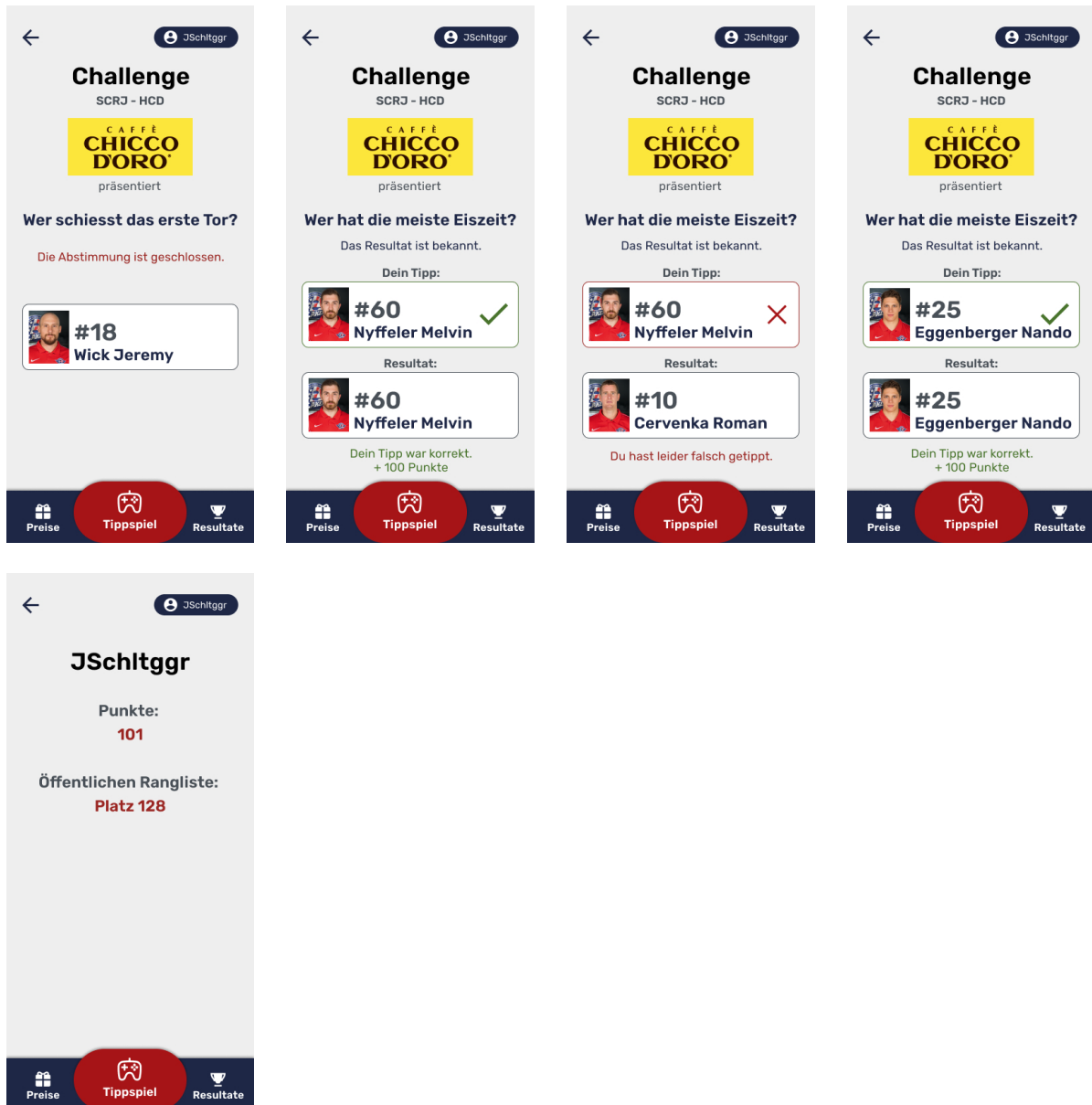


Trigger Event	
Challenge Resultat falsch	
Who	User2
Where	Stadion - Tribüne
What	User2 hat auf den falschen Spieler getippt.

E.3. Figma Prototyp Entwurf







E.4. Aufgabenstellung

Aufgabenstellung Bachelorarbeit FS22 Informatik OST «SCRJ Lakers Fan Engagement App»

1. Betreuer

Prof. Dr. Markus Stolze, OST, IFS
markus.stolze@ost.ch

2. Praxispartner

Lakers Sport AG
Walter-Denzler-Strasse 3
8640 Rapperswil

Kontaktperson
Michael Bart
michael.bart@lakers.ch
Tel: +41 76 337 75 39

3. Studierende

- Joel Schaltegger <joel.schaltegger@ost.ch>
- Philipp Emmenegger <philipp.emmenegger@ost.ch>
- Simon Hager <simon.hager@ost.ch>

4. Ausgangslage

Die OST und Lakers Sport AG haben eine Zusammenarbeit vereinbart. In diesem Zusammenhang sollen verschiedene Projekte im gegenseitigen Interesse realisiert werden. Eins der identifizierten Projekte ist die Erstellung einer App mit der das Engagement von Fans gesteigert werden kann.

5. Ziel der Arbeit

Analyse und prototypische Implementation einer SCRJ Lakers Fan Engagement App. Implementationsdetails können von den Studierenden bestimmt werden.

6. Auftrag

Die folgenden Teilaufgaben sind zu bearbeiten

- Analyse: Erhebung und Dokumentation der Anforderungen
 - Beschreibung der Ausgangssituation.
 - Analyse und Dokumentation von «guten» ähnlichen Anwendungen und «best practices».
 - Dokumentation der «Technology Constraints» und verwendeter externer APIs
 - Funktionale Anforderungen (notwendig, optional)
 - Nicht-Funktionale Anforderungen (notwendig, optional)
- Entwurf: Entwicklung Architektur und Vorschlag UX
 - Auswahl der Implementationsmethode der App (z.B. PWA mit React integriert in SCRJ Website)
 - Analyse sinnvoller Architektur und (soweit sinnvoll) unterstützender Libraries und Frameworks

- Beschreibung der Nutzer (Hauptzielgruppe) und Rollen
- Prototyping UX (Paper, Scenario Walkthrough für alle Nutzerrollen, ...)
- Definition eines MVP + Hypothesen zu Ausbausritten
- Demonstration Architektur-Prototyp und validiertem UX Vorschlag ca. zur Mitte des Projektes.
- Implementation und Test (mit Referenz auf FA/NFA)
 - Planung und Durchführung von Browser/Geräte-, Usability-, Load- und Pilot-Test
 - Dokumentation des Erreichungsgrads der FA/NFA (inkl. Unterstützung von Browsern und Geräten)
 - Beschreibung weiter Limitation
 - Empfehlungen und Beschreibungen zu weiteren Ausbausritten mit Aufwandschätzung
- Sinnvolle Dokumentation des Projektes und der prototypischen Implementation
 - Developer-Dokumentation
 - Projekt-Dokumentation inkl. Dokumentation von Browser/Geräte-, Usability-, Load- und Pilot-Test.

7. Rahmenbedingungen

Die von den Studierenden erstellte Software wird von den Studierenden mit einer MIT Lizenz versehen. Der Zeitpunkt und die Modalitäten der öffentlichen Verfügbarmachung des Codes wird zwischen den Studierenden und Lakers Sport AG abgesprochen. Die Lakers Sport AG und OST können den Code auch vor der Veröffentlichung durch die Studierenden in Projekten einsetzen.

8. Schlussbestimmungen

Im Weiteren gelten die Bedingungen zu den Informatik Bachelorarbeiten entsprechend dem aktuellen «Leitfaden für Bachelor- und Studienarbeiten Bachelorstudiengang Informatik an der OST»

Das Kriterien-Raster zur Bewertung wird den Studierenden zusammen mit dieser Aufgabenstellung verteilt. Allfällige Anpassungen müssen bis Mitte der Arbeit einvernehmlich beschlossen werden.

Rapperswil, 21 Feb 2022, Prof. Dr. Markus Stolze

(Dokument ohne Unterschrift)

E.5. Besprechungsbeschlüsse

E.5.1. Kickoff 24.02.2022

- Sitzungsprotokoll mit Entscheidungen führen und Aktualisierungen dem Betreuer mitteilen
- Kommunikation mit Betreuer: schnelle Kommunikation über Teams, ansonsten Mail
- GitLab Issues am Ende des Projektes exportieren
- Code Abgabe als GitLab Repositories ist in Ordnung
- Projekt- und Produkt Dokumentation trennen
- Developer Instruktionen in README okay, in Dokumentation vermerken.
- Wir/Ich Form in Dokumentation erlaubt
- Wissenschaftliche Arbeiten zitieren erwünscht, aber nur wenn sinnvoll
- Schnittstellen dokumentieren mit OpenAPI o. Ä.
- Sonarqube als Code Statistik okay
- JavaDok nur relevant wo API extern zur Verfügung gestellt wird
- Micheal Bart kontaktieren
- Erstellen eines visuellen User Szenarios dringend

E.5.2. Update Meeting mit Michael Bart 03.03.2022

- Szenarien und Konzept für App ist so in Ordnung
- Umsetzung als PWA
- Nächstes Meeting sobald Prototyp vorhanden ist
- Michael Bart schaut, dass uns der Multimediaverantwortlicher (Rene Schmid) bis nächsten Dienstag antwortet

E.5.3. Update Meeting mit Betreuer 03.03.2022

- Storyboard als PDF zustellen
- Code Reviews: Michael Gfeller meldet sich bei uns um Termine in KW 14 und KW18 abzumachen
- Wir gehen bezüglich der Architektur auf Mirko Stocker zu
- Markus Stolze klärt ab, ob wir einen AWS Account von der OST bereitgestellt bekommen

E.5.4. Expertenmeinung von Prof. Mirko Stocker 07.03.2022

- AWS Amplify nochmals in Betracht ziehen
- Projekt eignet sich gut für eine Serverless Umsetzung
- Empfehlung: Bereits bei Development einen AWS Account der Lakers nutzen
- Kosten sollten sehr gering sein
- Cloud Formation für einfaches Aufsetzen der Umgebung

E.5.5. Update Meeting mit Betreuer 10.03.2022

- Prototyp ist in Ordnung
- AWS Account der OST ist unwahrscheinlich, AWS Account der Lakers einfordern
- MVP zwei Wochen später
- Michael Bart fragen: Preise je Challenge oder je Tippspiel
- AWS Nutzung bringt grösseres Risiko mit sich
- Zwischenpräsentation wird abgehalten, Markus Stolze definiert Termin noch

E.5.6. Update Meeting mit Michael Bart 11.03.2022

- Michael Bart schaut, ob die Lakers einen AWS Account bereitstellen können
- Michael Bart sendet uns Medien von den Lakers
- Preise pro Tippspiel (nicht pro Challenge)

E.5.7. Update Meeting mit Betreuer 17.03.2022

- Zwischenpräsentation ist am Montag, 4. April 2022, um 0900-1000 oder 1000-1100

E.5.8. Zwischenpräsentation 04.04.2022

- Abschlusspräsentation ist am 28.06.2022 von 1330 bis 1430, vor Ort

E.5.9. Update Meeting mit Betreuer 05.05.2022

- Mitteilen, ob wir ein kleines Video oder eine kleine Präsentation für das Meeting "SCRJ meets OST" am 13.05.2022 vorbereiten
- Zusenden eines Grobkonzepts für die Dokumentation

E.5.10. Update Meeting mit Betreuer 10.06.2022

- Aufwandschätzung: Schätzungen höher stellen und gemäss Arbeitseffizienz von Simon
- Kein Meeting nächste Woche (Donnerstag)

E.5.11. Abschlussbesprechung 14.06.2022

- Michael Bart hat Arbeit abgenommen
- Michael Bart wird sich um Rechtliches (AGB, Datenschutzerklärung) kümmern
- App wird von Institut weiterbetrieben
- Release 1: Im Sommertraining
- Release 2: Im Testspiel am 13.08.2022

E.6.4. Functions Issues

Title	Description	Issue ID	URL	State	Author	Author Organization	Assignee	Assignee Organization	Confidential	Public	Start Date	Closed Date	Updated Date	Created Date	Resolution	Weight	Labels	Final Release	Open Type
Issue 1	Issue 1 description	1	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-01	2019-02-01	2019-02-01	2019-02-01	2019-02-01	1	Issue 1	0	0
Issue 2	Issue 2 description	2	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-02	2019-02-02	2019-02-02	2019-02-02	2019-02-02	1	Issue 2	0	0
Issue 3	Issue 3 description	3	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-03	2019-02-03	2019-02-03	2019-02-03	2019-02-03	1	Issue 3	0	0
Issue 4	Issue 4 description	4	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-04	2019-02-04	2019-02-04	2019-02-04	2019-02-04	1	Issue 4	0	0
Issue 5	Issue 5 description	5	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-05	2019-02-05	2019-02-05	2019-02-05	2019-02-05	1	Issue 5	0	0
Issue 6	Issue 6 description	6	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-06	2019-02-06	2019-02-06	2019-02-06	2019-02-06	1	Issue 6	0	0
Issue 7	Issue 7 description	7	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-07	2019-02-07	2019-02-07	2019-02-07	2019-02-07	1	Issue 7	0	0
Issue 8	Issue 8 description	8	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-08	2019-02-08	2019-02-08	2019-02-08	2019-02-08	1	Issue 8	0	0
Issue 9	Issue 9 description	9	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-09	2019-02-09	2019-02-09	2019-02-09	2019-02-09	1	Issue 9	0	0
Issue 10	Issue 10 description	10	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-10	2019-02-10	2019-02-10	2019-02-10	2019-02-10	1	Issue 10	0	0
Issue 11	Issue 11 description	11	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-11	2019-02-11	2019-02-11	2019-02-11	2019-02-11	1	Issue 11	0	0
Issue 12	Issue 12 description	12	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-12	2019-02-12	2019-02-12	2019-02-12	2019-02-12	1	Issue 12	0	0
Issue 13	Issue 13 description	13	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-13	2019-02-13	2019-02-13	2019-02-13	2019-02-13	1	Issue 13	0	0
Issue 14	Issue 14 description	14	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-14	2019-02-14	2019-02-14	2019-02-14	2019-02-14	1	Issue 14	0	0
Issue 15	Issue 15 description	15	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-15	2019-02-15	2019-02-15	2019-02-15	2019-02-15	1	Issue 15	0	0
Issue 16	Issue 16 description	16	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-16	2019-02-16	2019-02-16	2019-02-16	2019-02-16	1	Issue 16	0	0
Issue 17	Issue 17 description	17	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-17	2019-02-17	2019-02-17	2019-02-17	2019-02-17	1	Issue 17	0	0
Issue 18	Issue 18 description	18	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-18	2019-02-18	2019-02-18	2019-02-18	2019-02-18	1	Issue 18	0	0
Issue 19	Issue 19 description	19	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-19	2019-02-19	2019-02-19	2019-02-19	2019-02-19	1	Issue 19	0	0
Issue 20	Issue 20 description	20	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-20	2019-02-20	2019-02-20	2019-02-20	2019-02-20	1	Issue 20	0	0
Issue 21	Issue 21 description	21	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-21	2019-02-21	2019-02-21	2019-02-21	2019-02-21	1	Issue 21	0	0
Issue 22	Issue 22 description	22	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-22	2019-02-22	2019-02-22	2019-02-22	2019-02-22	1	Issue 22	0	0
Issue 23	Issue 23 description	23	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-23	2019-02-23	2019-02-23	2019-02-23	2019-02-23	1	Issue 23	0	0
Issue 24	Issue 24 description	24	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-24	2019-02-24	2019-02-24	2019-02-24	2019-02-24	1	Issue 24	0	0
Issue 25	Issue 25 description	25	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-25	2019-02-25	2019-02-25	2019-02-25	2019-02-25	1	Issue 25	0	0
Issue 26	Issue 26 description	26	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-26	2019-02-26	2019-02-26	2019-02-26	2019-02-26	1	Issue 26	0	0
Issue 27	Issue 27 description	27	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-27	2019-02-27	2019-02-27	2019-02-27	2019-02-27	1	Issue 27	0	0
Issue 28	Issue 28 description	28	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-28	2019-02-28	2019-02-28	2019-02-28	2019-02-28	1	Issue 28	0	0
Issue 29	Issue 29 description	29	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-02-29	2019-02-29	2019-02-29	2019-02-29	2019-02-29	1	Issue 29	0	0
Issue 30	Issue 30 description	30	https://github.com/...	Closed	Simon Hager	Simon Hager	Simon Hager	Simon Hager	No	No	2019-03-01	2019-03-01	2019-03-01	2019-03-01	2019-03-01	1	Issue 30	0	0

E.6.5. Setup Issues

Title	Description	Issue ID	URL	Status	Author	Author Username	Assignee	Assignee Username	Confidential	Locked	Due Date	Created At [UTC]	Updated At [UTC]	Closed At [UTC]	Milestone	Weight	Labels	Time Estimate	Time Spent
Setup project infrastructure	[-] Setup servers	1	https://gitlab.com/ibw-wir/setup/-/issues/1	Closed	Simon Hager	emmenegger	Simon Hager	simon.hager	No	No		23/02/2022 14:14	02/05/2022 12:59	16/05/2022 08:59	Mid-kick-off		Documentation,Sprint 1,Task	0	8000

E.6.6. Time Report Issues

Title	Description	Issue ID	URL	Status	Author	Author Username	Assignee	Assignee Username	Confidential	Locked	Due Date	Created At [UTC]	Updated At [UTC]	Closed At [UTC]	Milestone	Weight	Labels	Time Estimate	Time Spent
Setup Time Report		1	http://gitlab.eol.ch/bee-org/time-report/-/issues/2	Closed	Joel Schaltegger	joel.schaltegger	Joel Schaltegger	joel.schaltegger	No	No		23/02/2022 13:37	24/02/2022 14:45	25/02/2022 14:45	Mid-EOL Off		Documentation,Sprint 1,Task	0	23:00

F. Präsentationen

Im Verlauf der Bachelorarbeit wurden diverse Präsentation gehalten. Die dafür entwickelten Präsentationen werden hier aufgeführt.

F.1. Crowd Test

16.06.2022



1

Ablauf

- Vorinterview
- Testdurchführung
- Nachinterview

2 | BA – SCRJ Fan Engagement App

16. Juni 2022

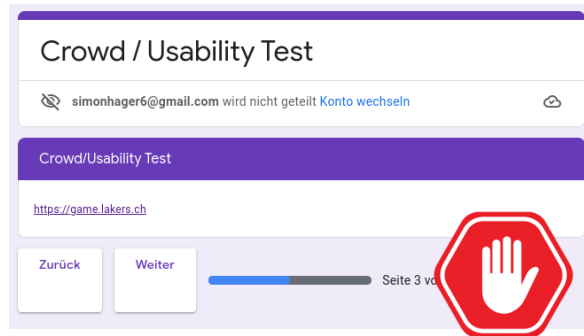


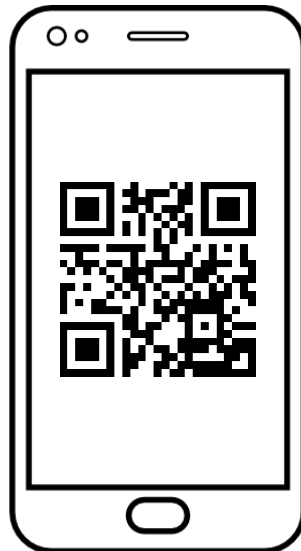
2

1



<https://bit.ly/3L1uMD1>





• <https://game.lakers.ch>

5 | BA – SCRJ Fan Engagement App

16. Juni 2022



5

Heimspiel



VS



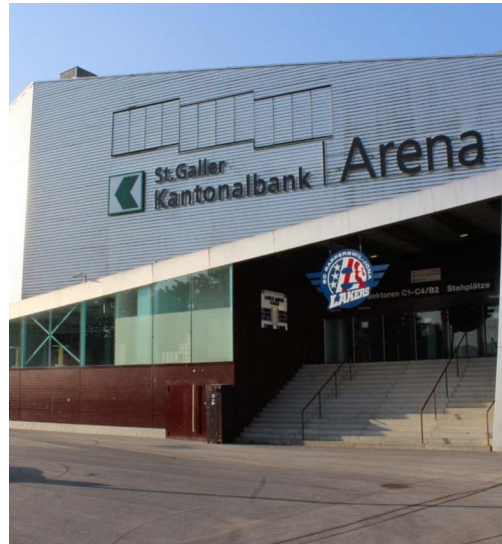
6 | BA – SCRJ Fan Engagement App

16. Juni 2022



6

Vor dem Spiel



7 | BA – SCRJ Fan Engagement App

16. Juni 2022



7

Liga-Topscorer 2021/22



#10 Roman Cervenka



#72 Gian-Marco Wetter

8 | BA – SCRJ Fan Engagement App

16. Juni 2022



8

Liga-Topscorer 2021/22

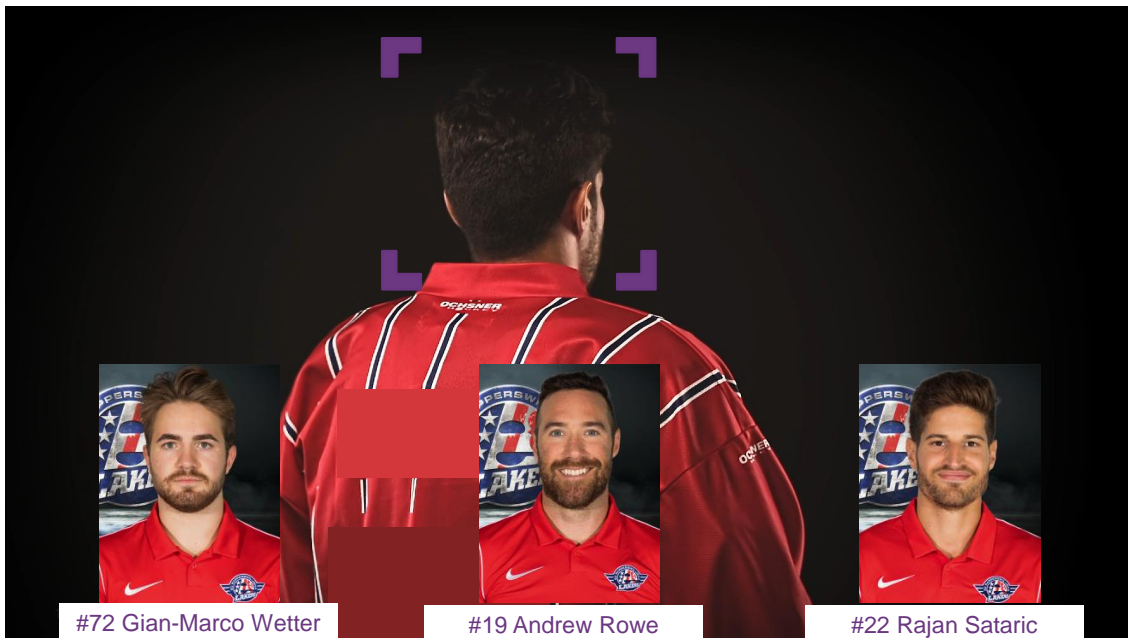


9 | BA – SCRJ Fan Engagement App

16. Juni 2022



9

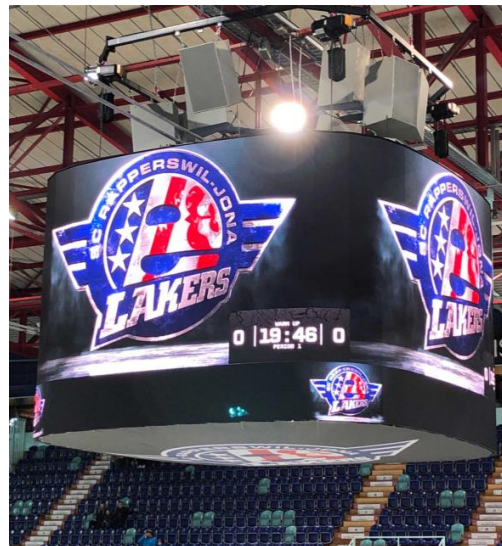


10



11

Anpiff

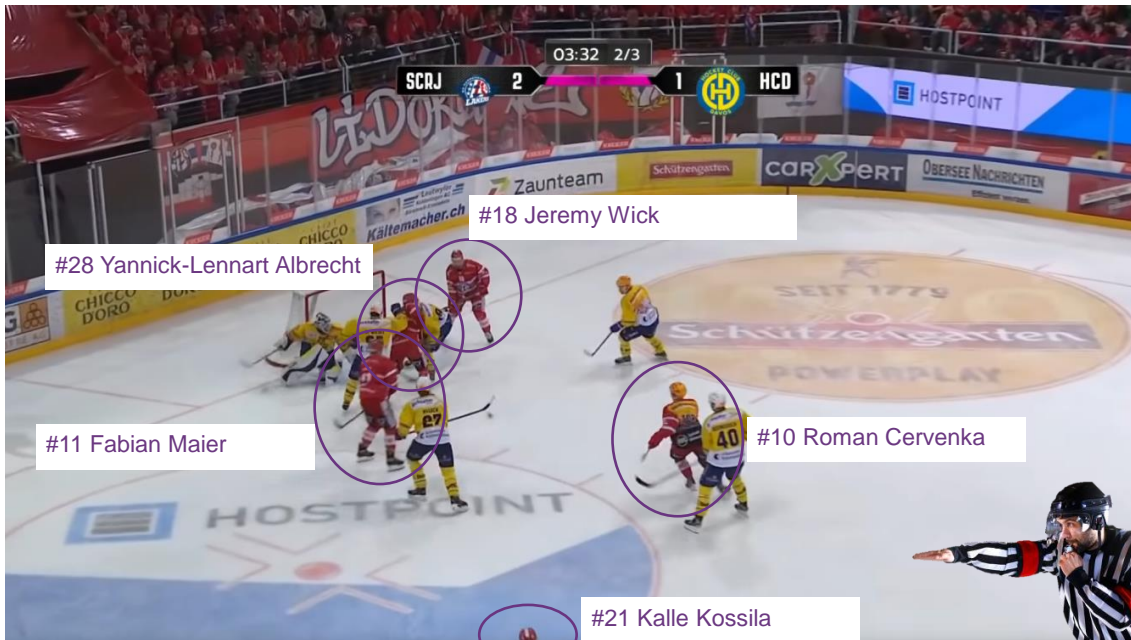


16. Juni 2022



12 | BA – SCRJ Fan Engagement App

12



15



16

1. Drittelspause

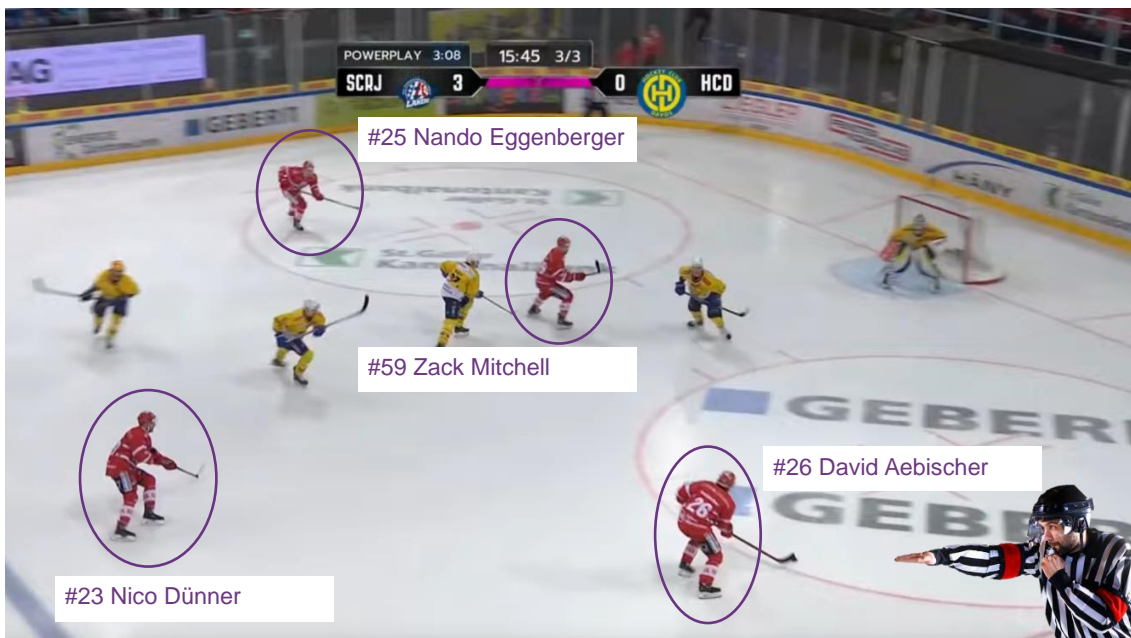


17 | BA – SCRJ Fan Engagement App

16. Juni 2022



17



22



23

2. Drittelspause



27



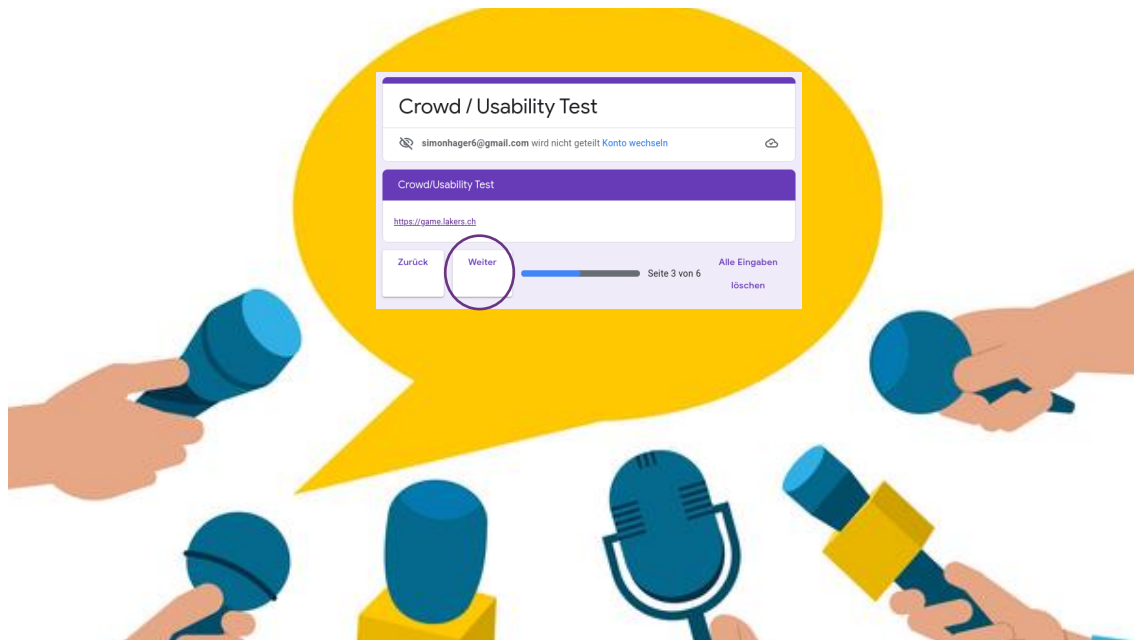
28

<https://www.srf.ch/play/tv/redirect/detail/b286bfac-cd51-4f20-bce9-cbadb4c2122e?startTime=803>

29

10

Schlusspfeif



16.06.2022



32

12

F.2. Zwischenpräsentation

16.06.2022



1

Agenda

- Domäne (Joel)
- Aktueller Stand (Philipp)
- Herausforderungen und Risiken (Hager)
- Weiteres Vorgehen (Joel / Hager)

2



1

Domäne

- Ausgangssituation
 - Analyse und prototypische Implementation einer Fan Engagement App
- Funktionale Anforderungen
 - Mit Marketing Leiter der Lakers
- Storyboard

3 | BA – SCRJ Fan Engagement App

16. Juni 2022



3

Aktueller Stand

- Prototyp inkl. Usability Tests
- AWS Services
- Frontend

4 | BA – SCRJ Fan Engagement App

16. Juni 2022



4

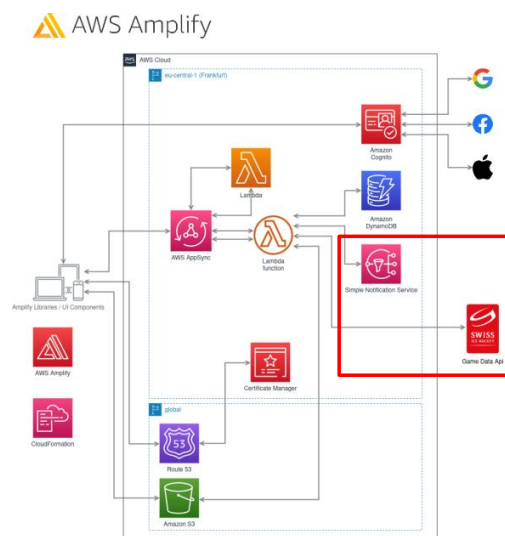
Aktueller Stand

Prototyp und Usability Tests

- Prototyp
 - Entwickelt mit Figma
 - Style von Lakers.ch
- Usability Test
 - Durchführung mit 4 Testpersonen
 - Ergebnisse
 - Label Resultate -> Rangliste
 - Verfügbare Preise darstellen
 - Ablauf vom Preis einlösen
 - Bestätigung der Tippauswahl



Aktueller Stand



Aktueller Stand

Frontend

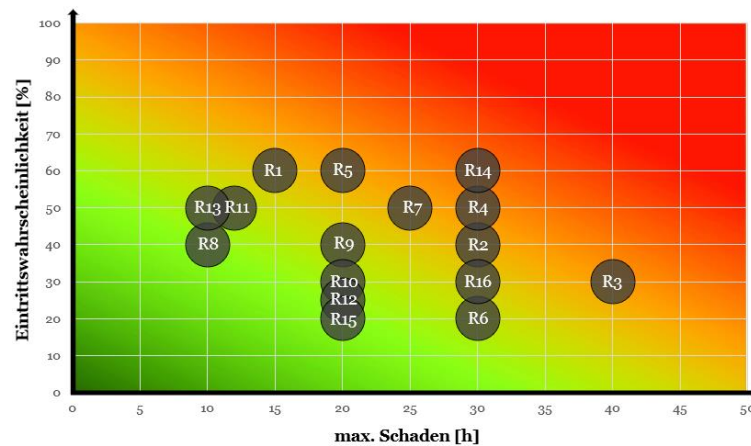
- Umsetzung Design mit User / Admin Theme
- Integration von AppSync / GraphQL
- Authentisierung mit AWS Cognito
- PWA



<https://dev.aws.ba.smnhgr.ch/>

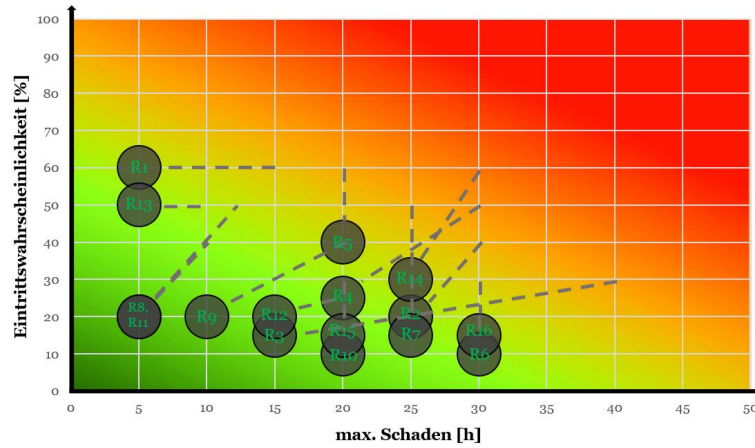
Herausforderungen und Risiken

Risikomanagement - M0



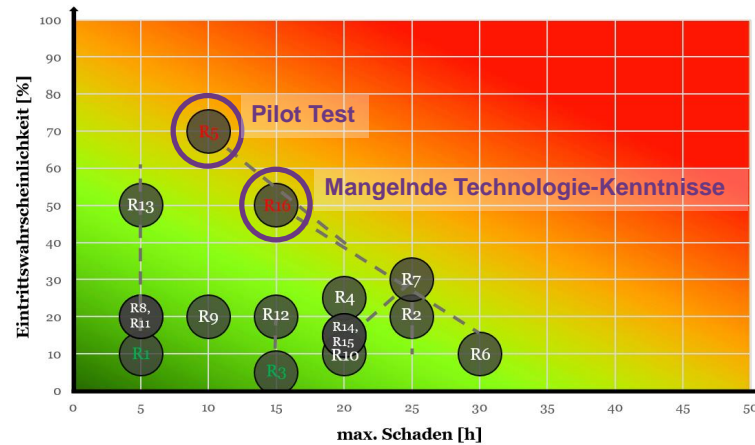
Herausforderungen und Risiken

Risikomanagement - M1



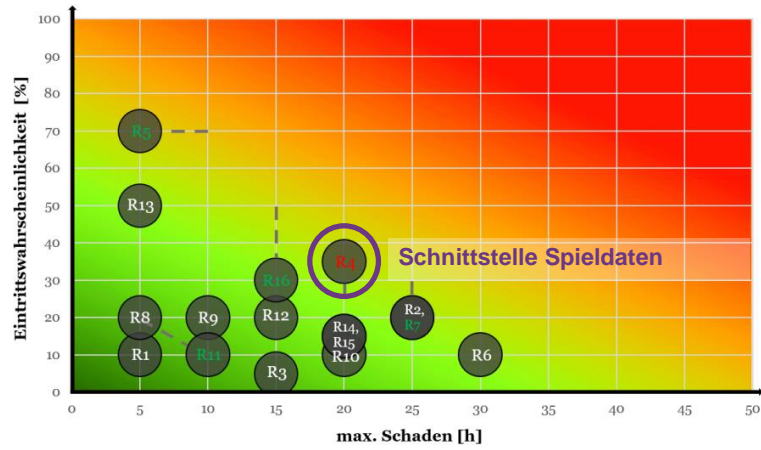
Herausforderungen und Risiken

Risikomanagement - M2



Herausforderungen und Risiken

Risikomanagement - M3



11 BA – SCRJ Fan Engagement App

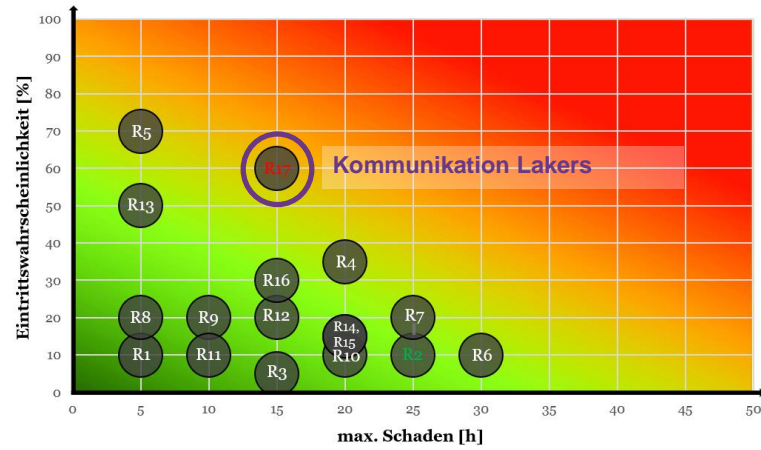
16. Juni 2022



11

Herausforderungen und Risiken

Risikomanagement - M4



12 BA – SCRJ Fan Engagement App

16. Juni 2022



12

Herausforderungen und Risiken

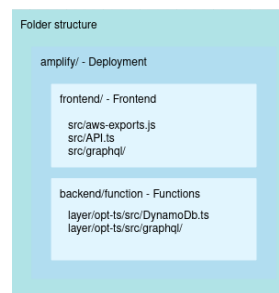
Gelöste Herausforderung: TypeScript - GitLab - Amplify

Probleme

- Strikte Orderstruktur von Amplify
- Erforderliche/Ignorierbare Dateien
- Authentifizierung gegenüber AWS
- GraphQL Artifacts
- Schlechte Unterstützung von Typescript

Lösung

- Provider Repositories Frontend und Function
- Consumer Repository Deployment



13 | BA – SCRJ Fan Engagement App

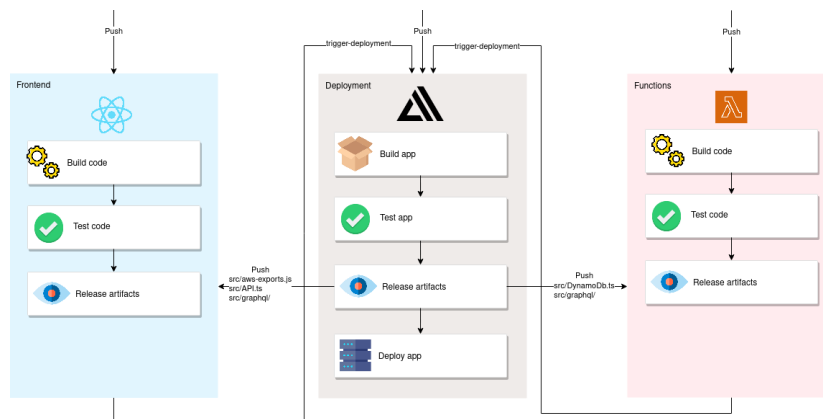
16. Juni 2022



13

Herausforderungen und Risiken

Gelöste Herausforderung: TypeScript - GitLab - Amplify



14 | BA – SCRJ Fan Engagement App

16. Juni 2022



14

Herausforderungen und Risiken

Gelöste Herausforderung: Amplify

Probleme

- Verhalten bei Push/Pull Aufrufen
- Schlechte Unterstützung von Teamarbeit

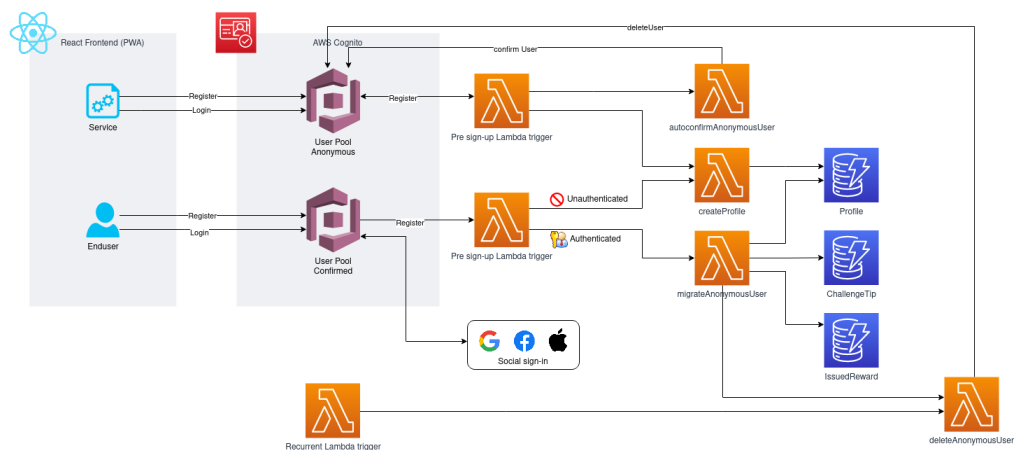
Lösung

- Research Amplify



Herausforderungen und Risiken

Gelöste Herausforderung: Anonymer Benutzer



Herausforderungen und Risiken

Gelöste Herausforderung: Frontend State Management

Probleme

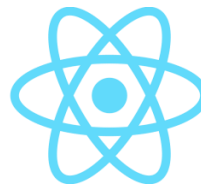
- AppSync / DynamoDB hält State
- Benötigen lokalen State
- Synchronisierung React und AppSync
- Ansatz: Redux, Apollo

Lösung

- React Hooks
 - useReducer
 - useContext (global)
 - useState (lokal)



Redux



React Hooks

17 | BA – SCRJ Fan Engagement App

16. Juni 2022



17

Herausforderungen und Risiken

Anstehende Herausforderung: Pilot test / Kostenoptimierung

Probleme

- Quantitative Zahlen zur tatsächlichen Nutzung
- Unklarheiten zur Belastung/Stabilität

Lösung

- Pilot Test durchführen
- Performance Tests durchführen

18 | BA – SCRJ Fan Engagement App

16. Juni 2022



18

Herausforderungen und Risiken

Anstehende Herausforderung: Einbindung Spieldaten

Probleme

- Langwierige Kommunikation
- Aktuell keine Testmöglichkeiten
- Kein Qualitätsnachweis über Daten

Lösung

- Möglichst bald Schnittstellenzugang erhalten
- Konzeptionell als Add-on behandeln



Herausforderungen und Risiken

Anstehende Herausforderung: Challengediversität

Probleme

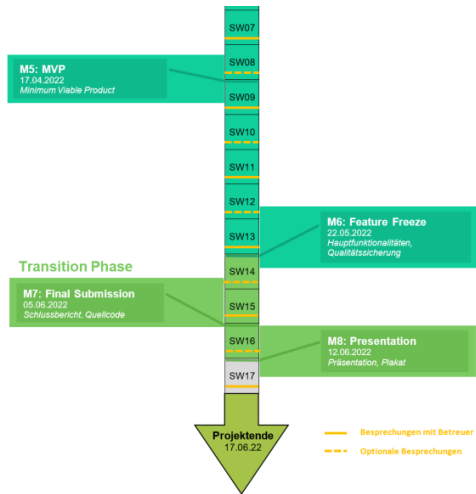
- Gleiche Datenstruktur für verschiedene Challenge-Formulare
- Umgang mit standardisierten Feldern
 - tipOptions: [String]
 - solution: String
 - answer: String

Lösung

- Standard festlegen
- Design Patterns berücksichtigen

Weiteres Vorgehen

- Kurzfristig:
 - Responsive Design
 - Fortsetzen der Integration
 - Fertigstellen des MVP
- Langfristig:
 - Wenn möglich: Pilot Test
 - Features hinzufügen
 - Weitere Fragentypen



F.3. SCRJ meets OST

16.06.2022



1

Agenda

- Vorstellung
- Konzept
- Aktueller Stand
- Weiteres Vorgehen

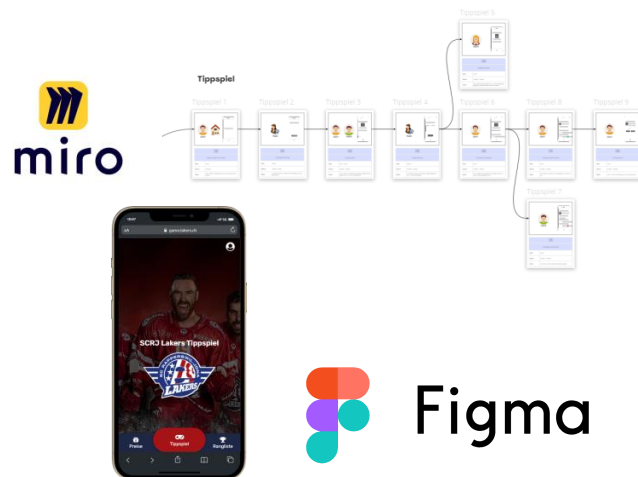
2



1

Konzept

- Ausgangssituation
 - Implementation einer Fan Engagement App
- Vorgehen
 - Idee für Tippspiel
 - Storyboard
 - Entwicklung eines Prototyps



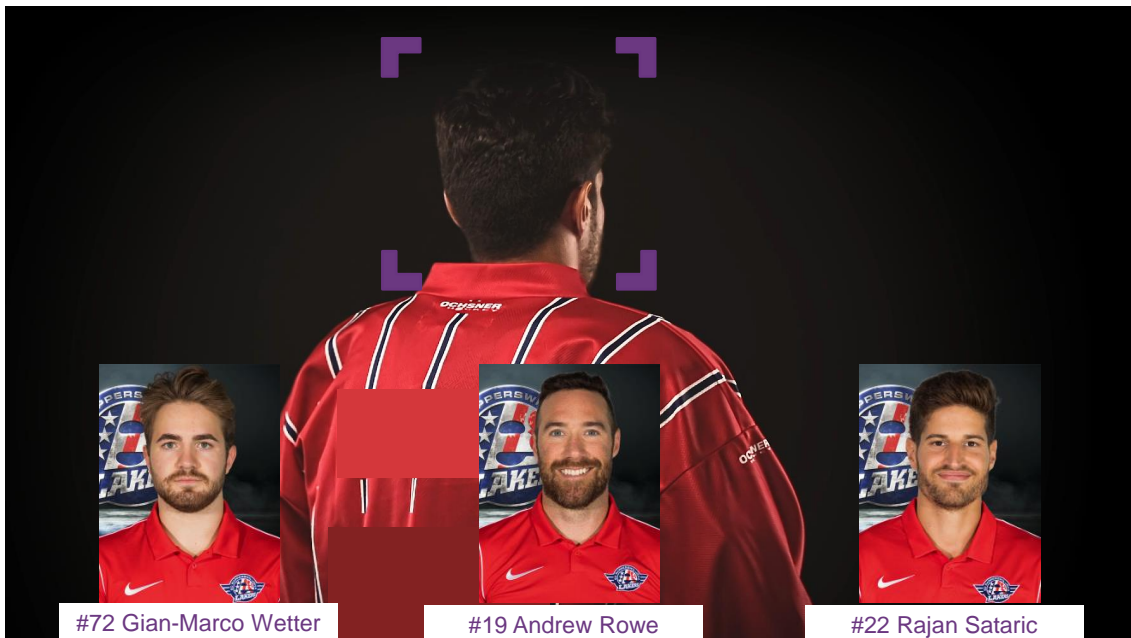
Aktueller Stand

- Stabile Version auf game.lakers.ch
- Entwicklungs-Version auf dev.game.lakers.ch für neue Features
- Usability und Performance getestet

Demonstration



<https://dev.game.lakers.ch>



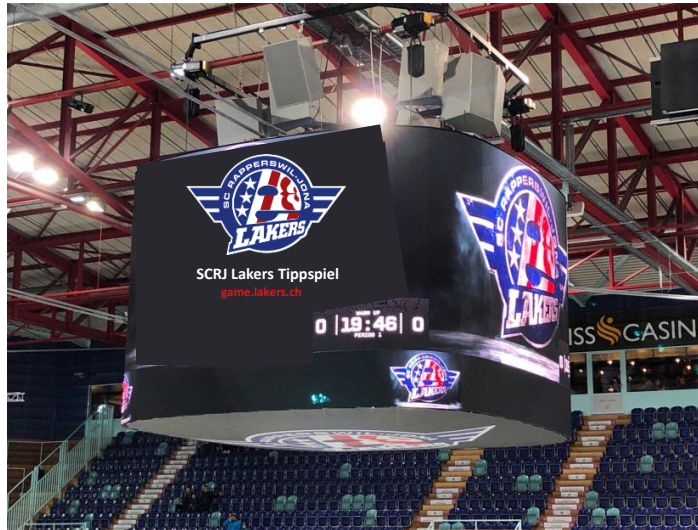


7

Weiteres Vorgehen

- Kurzfristig:
 - Verbesserung der Stabilität / Performance
 - Spiel-Statistiken der Benutzer
 - Google Analytics für Marketing
 - Erkenntnisse der Usability Tests umsetzen
 - Dokumentation
- Weiterentwicklung:
 - Neue Fragentypen
 - Login-Zugang per SMS statt Email
 - Anbindung an Verbands-API

16.06.2022



9 | BA – SCRJ Fan Engagement App

16. Juni 2022

