

Bachelorarbeit, Abteilung Informatik

Aufgaben-Coach

Hochschule für Technik Rapperswil

Herbstsemester 2019

16. September 2019

<i>Autoren:</i>	Luca Gubler & Alessandro Bonomo
<i>Betreuer:</i>	Frank Koch
<i>Experte:</i>	Stephan Meier
<i>Gegenleser:</i>	Laurent Metzger
<i>Projektpartner:</i>	INS Institute for Networked Solutions
<i>Arbeitsperiode:</i>	16.09.2019 - 10.01.2020
<i>Arbeitsumfang:</i>	360 Stunden, 12 ECTS pro Student

Abstract

Die Schüler der Deutschschweizer Kantone werden mit der Einführung des Lehrplans 21 mit weitgehend gleichen Lerninhalten konfrontiert werden. Ein bestimmtes Thema wird dann also in der gleichen Art und Weise an eine wesentlich grössere Zahl von Schülern vermittelt. Gleichzeitig werden immer mehr Lerninhalte nach der Flipped-Classroom Methode vermittelt, wobei Schüler sich das Wissen weitgehend eigenständig erarbeiten und dabei von Lehrern betreut werden. Bei dieser Wissenserarbeitung spielen digitale Medien eine zentrale Rolle; die Rolle der Lehrenden wandelt sich dabei zu Coaches.

Es gibt bereits heute einige Programme oder Open-Source Projekte, welche auf diesen Trend aufspringen und Lernende beim Erarbeiten des Schulstoffs unterstützen wollen. Diese richten sich jedoch hauptsächlich direkt an Lernende, indem zum Beispiel Nachhilfeunterricht für einzelne Fächer angeboten wird.

In dieser Bachelorarbeit wird eine Anwendung entwickelt, welche Lernende und Lehrpersonen enger miteinander verbindet. Lernende können ein Thema auf der Plattform selbstständig und in ihrem eigenen Tempo erlernen. Lehrpersonen stellen dazu die passenden Aufgaben zur Verfügung. Hat der Lernende Schwierigkeiten beim Lösen einer Aufgabe, so kann er mit Hilfestellungen Schritt für Schritt durch die Aufgabe geführt werden. Gibt es dennoch Unklarheiten, können allfällige Fragen in einem Forum gestellt werden, bei welchem sich alle Fragen punktuell rund um diese spezifische Aufgabe drehen.

Die Lehrperson kann mit Hilfe von Statistiken jederzeit kontrollieren, wie gut die Lernenden ein spezifisches Thema verstehen. Somit muss die Lehrperson nicht bis zur Prüfung warten, um ein Feedback über den Wissensstand der Lernenden zu erhalten, sondern kann schon während der Lernphase gezielt auf jene Schüler eingehen, welche das Thema noch nicht ganz verstanden haben. Andere Schüler werden so auch nicht aufgehalten und können bereits mit dem nächsten Thema beginnen. Die Lehrperson behält so auch den Überblick über den Wissensstand der Lernenden und kann kontrollieren, dass alle Lernenden die einzelnen Themengebiete in der vorgegebenen Zeit abschliessen.

Lay Summary

Mit der, im Rahmen dieser Bachelorarbeit, entwickelten Webanwendung, soll die Zusammenarbeit von Schülern und Lehrern verbessert werden. Dabei spielt die Unterrichtsmethode "Flipped Classroom", also "Umgedrehter Unterricht", eine wichtige Rolle. Hierbei handelt es sich um eine Unterrichtsmethode, bei welcher die Schüler ihr Wissen zu einem grossen Teil selbstständig erarbeiten und sich die Rolle des Lehrers mehr und mehr in die eines Coaches verwandelt. Doch auch in der Rolle eines Coaches muss die Lehrperson über Möglichkeiten verfügen, den Wissensstand der Schüler zu ermitteln um sicherzustellen, dass niemand den Anschluss verliert.

Mit Hilfe der entwickelten Anwendung können Schüler selbstständig vordefinierte Theorieinhalte durcharbeiten und lernen. Um die Schüler beim Lernen so gut wie möglich zu unterstützen sowie zu überprüfen, ist es Lehrpersonen möglich, selber Aufgaben zu definieren, welche sie den Schülern als Hausaufgabe aufgeben können. Zu jeder Aufgabe können Hilfestellungen erfasst werden, welche den Schülern beim Lösen, helfen sollen. Sollte dies nicht genügen, kann im aufgabenspezifischen Forum um Hilfe gebeten werden. Die von den Schülern gelösten Aufgaben können von der Lehrperson korrigiert und bewertet werden. Anhand der bewerteten Abgaben kann die Lehrperson Statistiken einsehen, welche ihr den aktuellen Wissensstand der einzelnen Schüler übermittelt. Dadurch können schwächere Schüler erkannt und individuell unterstützt werden.

1 Management Summary

1.1 Ausgangslage

Mit der Umsetzung des Lehrplans 21 gibt es in allen deutsch- und mehrsprachigen Kantonen den selben Lehrplan. Unter den Zielen des Lehrplans 21 findet man folgenden Punkt¹:

Ein gemeinsamer Lehrplan ist eine Grundlage für die Koordination der Lehrmittel und erleichtert die gemeinsame Entwicklung von Lehrmitteln für die deutschsprachige Schweiz.

Das heutige Unterrichtsmodell hat verschiedene Nachteile. Kann zum Beispiel ein Lernender krankheitshalber nicht in die Schule, so wird der gesamte Schulstoff verpasst. Es bleibt dem Lernenden nichts anderes übrig, als zu Hause zu lernen, um so den verpassten Stoff nachzuholen. Da ein wichtiger Bestandteil des Unterrichts darin besteht, dass eine Lehrperson den Schülern die Theorie beibringt, kann die Lehrperson nicht individuell auf die Bedürfnisse einzelner Schüler eingehen. Manche haben die Theorie bereits verstanden, während andere längst den Faden verloren haben.

Ein weiteres Problem der Lehrpersonen ist, dass diese erst sehr spät ein Feedback über den Wissensstand der Schüler bekommen. Erst wenn eine Prüfung durchgeführt wurde, sieht der Lehrer wie gut die einzelnen Schüler der Klasse ein Thema verstanden haben. Bemerkt eine Lehrperson zu diesem Zeitpunkt, dass eine Wissenslücke besteht, ist meist nicht mehr genügend Zeit vorhanden, um dieses Thema nochmals in Ruhe zu erklären.

Die Idee von Lernplattformen für Schulen ist nicht ganz neu. An der Hochschule für Technik Rapperswil (HSR) wird zum Beispiel Moodle² eingesetzt, ein Open-Source Learning Management System (LMS). Es gibt aber auch kommerzielle Lösungen wie Sofatutor³ oder EF Class⁴. Diese richten sich aber entweder an die Schüler direkt oder sind nur für einzelne Fächer konzipiert.

¹ *Lehrplan21 - Ziele*. URL: <https://lehrplan21.ch/ziele> [Besucht am 4. Januar 2020].

² *Moodle - Open-source learning platform*. URL: <https://moodle.org/> [Besucht am 20. Oktober 2019].

³ *Mit Spass lernen und Noten verbessern*. URL: <https://www.sofatutor.ch/> [Besucht am 20. Oktober 2019].

⁴ *English lessons, designed for you*. URL: <https://class.ef.com/> [Besucht am 20. Oktober 2019].

Mit dieser Bachelorarbeit möchte man eine Anwendung schaffen, welche direkt im Unterricht verwendet werden kann. Lernende sollen in der Lage sein, unabhängig und individuell einzelne Themen zu erlernen, während eine Lehrperson dennoch in der Lage ist, den aktuellen Wissensstand einzelner Lernenden zu überprüfen. So ist die Lehrperson kein Wissensvermittler im klassischen Sinne, sondern er coacht die Lernenden und unterstützt diese beim Lernen.

1.2 Vorgehen / Technologien

Die Anwendung soll den Schülern rund um die Uhr zur Verfügung stehen und plattformübergreifend verwendbar sein. So kann die Anwendung auch zu Hause verwendet werden. Um diesen Anforderungen gerecht zu werden, wird eine Webanwendung mit dem Django-Framework und der Programmiersprache Python entwickelt.

Es wurde besonders darauf geachtet, dass die Anwendung modular aufgebaut ist. In Zukunft können so auf der bestehenden Plattform weitere Features implementiert werden.

1.3 Ergebnisse

Während dieser Bachelorarbeit ist eine Anwendung mit dem Titel "Aufgaben-Coach" entstanden. Diese Anwendung erlaubt es, Theoriezusammenfassungen mit Videos und Bildern bereitzustellen. Lehrer sind in der Lage, neue Aufgaben zu erstellen und diese den Schülern zur Verfügung zu stellen. Haben die Schüler die Aufgaben gelöst, kann der Lehrer diese einsehen und korrigieren. Anhand der gelösten Aufgaben werden Statistiken generiert, mit welchen eine Lehrperson jederzeit den Wissensstand der Klasse einsehen kann. Bemerkt die Lehrperson, dass ein einzelner Lernender Schwierigkeiten in einem spezifischen Aufgabengebiet hat, so kann die Lehrperson direkt und individuell auf diese Person zugehen, ohne die anderen Lernenden vom Lernen abzuhalten.

1.4 Ausblick

Da für die Bachelorarbeit nur ein begrenzter Zeitraum zur Verfügung steht, konnten nicht alle Features im gewünschten Umfang umgesetzt werden. Es ist jedoch geplant, dass diese Arbeit nach dem Studium in Form eines Startups verbessert und weiter entwickelt wird.

Inhaltsverzeichnis

Abstract	3
1 Management Summary	5
1.1 Ausgangslage	5
1.2 Vorgehen / Technologien	6
1.3 Ergebnisse	6
1.4 Ausblick	6
2 Aufgabenstellung	9
3 Technischer Bericht	15
3.1 Einleitung	15
3.1.1 Hintergrund	15
3.1.2 Problemstellung / Vision	15
3.1.3 Aufgabenstellung	16
3.1.4 Zielgruppen	16
3.2 Stand der Technik	17
3.2.1 Existierende Produkte	17
3.2.2 Flipped Classroom	19
3.3 Eigener Lösungsansatz	21
3.3.1 Konzeption	21
3.3.2 Zentrale Elemente	21
4 Anforderungen	23
4.1 Personas	23
4.1.1 Persona 1	23
4.1.2 Persona 2	24
4.1.3 Persona 3	24
4.2 Funktionale Anforderungen	25
4.2.1 Use Case Diagramm	25
4.2.2 Aktoren	26
4.2.3 Use Case Beschreibung	26
4.3 Nicht Funktionale Anforderungen	29
5 Architektur	30
5.1 System Übersicht	30
5.2 Domain Model	31
5.3 Tools und Frameworks	37
5.4 Deployment	40
6 Design	42
6.1 UI Design	42
6.1.1 UI Design Anforderungen	42

6.1.2	UI Design Entscheidungen	44
6.2	Mockups	45
6.3	Frontend Framework	47
7	Implementation und Tests	48
7.1	Django Architektur	48
7.2	Testing	50
7.2.1	Unit Tests	50
7.2.2	Integration Tests	51
7.2.3	Manuelle Tests	51
7.2.4	Usability Tests	51
7.2.5	Performance Testing	55
7.3	Continuous Integration	56
7.4	Code Statistiken	58
7.5	Ergebnisse	59
7.5.1	Resultate	59
7.5.2	Verbesserung	60
7.5.3	Weiterentwicklung	61
7.5.4	Nicht Funktionale Anforderungen	61
8	Projektplan	63
8.1	Inhalt	63
8.2	Projektübersicht	63
8.3	Organisation	64
8.4	Management Abläufe	65
8.5	Projektverwaltung	67
8.6	Risikomanagement	68
8.7	Qualitätsmanagement	71
8.8	Zeitauswertung	73
9	Software Dokumentation	75
10	Dankausgang	83
11	Fazit	84
A	Mockups	91

2 Aufgabenstellung

Auf den nachfolgenden Seiten befindet sich die vom Betreuer unterschriebene Aufgabenstellung.

Aufgabenstellung Bachelor Thesis „Aufgaben-Coaching“

1. Personen

Diplomanden: Alessandro Bonomo und Luca Gubler, Studiengang Informatik

Betreuer: Prof. Frank Koch, Dozent für Wirtschaftsinformatik

Experte: Stephan Meier

Gegenleser: Prof. Laurent Metzger

2. Ausgangslage, Problembeschreibung

In vielen Schulfächern wird immer dasselbe unterrichtet. Dinge, wie der Satz des Pythagoras, werden stetig neu erklärt. Da sich die Lehrer immer wieder wiederholen, bleibt ihnen weniger Zeit, um sich auf den wesentlichen Punkt zu konzentrieren - die Schüler.

Die Schüler sitzen im Unterricht und hören dem Lehrer zu. Schüler sind aber keine Maschinen, manchmal sind sie krank oder unaufmerksam. Den verpassten Stoff müssen sie dann irgendwann nachholen, spätestens wenn die Prüfung bevorsteht. Aber auch die Repetition ist mit viel Aufwand verbunden. Vergessenes wird meist mit einem grossen Zeitaufwand wieder aufgeholt.

3. Aufgabenstellung

Es wird eine Lernplattform erstellt, welche dieses Problem angeht. Auf dieser Lernplattform wird die Theorie der einzelnen Fächer in Form von Videos, Theoriezusammenfassungen und Übungen gemäss Lehrplan21 (siehe www.lehrplan21.ch) zur Verfügung gestellt. Die Schüler können mit Hilfe dieser Plattform unabhängig von ihrem Standort lernen, z.B. während einer Projektstunde in der Schule oder zu Hause in Form von Hausaufgaben.

Der Lehrer kann dazu einzelne Aufgaben im Tool erfassen. Die Schüler können diese Aufgaben selbstständig lösen. Anschliessend sieht der Lehrer eine Statistik, ob die Aufgaben gelöst wurden und ob diese auch korrekt sind. Falls eine Aufgabe überdurchschnittlich schlecht gelöst wurde, kann er diese Aufgabe nochmals im Plenum mit den Schülern diskutieren.

In dieser Bachelorarbeit soll dieser Punkt umgesetzt werden.

3.1 Funktionale Anforderung

Allgemeine Funktionalität

Admin

Pro Schule gibt es einen Administrator. Dieser Administrator ist für das Erfassen der Lehrer und Schüler verantwortlich. Des Weiteren kann er Klassen erfassen und einer Klasse die Lehrer und Schüler zuordnen. Falls die Schüler aus der Schule austreten, muss der Administrator diese Klassen auflösen und die Schüler aus dem System löschen können.

Lehrer

Der Lehrer kann multimedialen Content für die Wissensvermittlung freischalten sowie Aufgaben definieren und diese freischalten. Diese Aufgaben können einem Wochenplan zugeteilt werden. In diesem Wochenplan sind alle Aufgaben ersichtlich, welche die Schüler lösen müssen.

Der Lehrer sieht eine Statistik, wie viele Schüler Aufgaben gelöst haben und kann sehen, wie gut die einzelnen Aufgaben gelöst wurden. Wurde eine Mathematik Aufgabe überdurchschnittlich schlecht gelöst, kann der Lehrer diese Aufgabe nochmals mit der Klasse besprechen.

Bei der Auswertung der Statistik sieht der Lehrer, wie gut ein einzelner Schüler ein Thema versteht. Sieht der Lehrer, dass ein Schüler ein Thema überhaupt nicht versteht, kann der Lehrer auf diesen einzelnen Schüler zugehen und mit ihm persönlich die Aufgaben besprechen.

Schüler

Der Schüler sieht auf dem Dashboard alle Aufgaben. Er kann sich aber selber dafür entscheiden, in welcher Reihenfolge er diese Aufgaben lösen möchte. Hat er eine Aufgabe gemacht, kann er diese abschliessen. Solange die Deadline aber noch nicht abgelaufen ist, kann er Aufgaben immer noch bearbeiten.

Übungsaufgaben

Ein zentraler Punkt der Bachelorarbeit wird das Erstellen und Lösen von Aufgaben sein.

Nachdem die Schüler in einer ersten Phase einen Theorie Input angesehen haben, können sie in der folgenden Übungsphase Aufgaben zu diesem Thema lösen. Falls der Schüler das Resultat nicht sofort herausfindet, kann er mit Hilfe einer Schritt-für-Schritt Anleitung die Aufgabe lösen. Dazu gibt es einen Button, mit welchem er Hilfe anfordern kann.

Falls dem Schüler bei einer Übungsaufgabe trotzdem etwas unklar ist, kann er Fragen in das aufgaben-spezifische Forum stellen. Befinden sich unbeantwortete Fragen im Forum, erhält der Lehrer nach einer gewissen Zeit eine Meldung, so dass er sich der Frage des Schülers widmen kann.

Optional

Diese Plattform ist beliebig erweiterbar. Falls Zeit vorhanden ist, können noch mehrere Punkte implementiert werden:

- Ein Aufgaben-spezifisches Forum, bei welchem nur Fragen rund um eine Aufgabe gestellt werden.
- Lehrer kann Quizze erstellen.
- Schüler kann Aufgaben bewerten. Ist eine Aufgabe zum Beispiel unklar formuliert, kann der Schüler dies melden.
- Lehrer sieht Bewertung / Kommentierung seiner Aufgaben und kann diese dementsprechend anpassen
- Schüler kann Frage direkt an Lehrer stellen (Chat Funktionalität)
- Eine Lehrperson kann selber definierte Aufgaben für andere Lehrpersonen freigeben.

Bei den schülerspezifischen Ansichten wird darauf geachtet, dass Mobile-first unterstützt ist. Die Ansichten, die nur den Admin und die Lehrer betreffen, wird aus zeitlichen Gründen in einem ersten Schritt nur die Desktop-Ansicht unterstützt.

3.2 Nicht Funktionale Anforderung

Usability

Bei der Applikation soll es sich um eine Mobile First Applikation handeln. Die Plattform soll sowohl auf Desktop PCs, an Tablets oder auf Mobiles bedienbar sein. Die Applikation soll ein intuitives User Interfaces besitzen, damit sich die Benutzer auf Anhieb zurechtfinden.

Reliability

Die Applikation soll so robust sein, dass sie auch reibungslos läuft, wenn mehrere Schüler mit der Plattform verbunden sind.

Scalability

Das System soll von 5 Schulen mit je ca. 300 Benutzern einwandfrei verwendet werden können.

4. Zur Durchführung

Mit dem Betreuer finden Besprechungen gemäss Absprache statt. Die Besprechungen sind von den Diplomanden mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das dem Betreuer per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsergebnisse in vorläufigen Versionen abzugeben.

5. Dokumentation und Abgabe

Das gesamte Projekt muss dokumentiert werden. Zu dokumentieren ist der gesamte Projektverlauf.

Die Dokumentation zur Projektplanung und –Verfolgung ist gemäss den Richtlinien der Abteilung Informatik anzufertigen. Die Detailanforderungen an die Dokumentation der Recherche- und Entwicklungsergebnisse werden entsprechend dem konkreten Arbeitsplan festgelegt.

Die Dokumentation ist vollständig digital sowie ausgedruckt in drei Exemplaren abzugeben.

Neben der Dokumentation sind abzugeben:

- ein Poster zur Präsentation der Arbeit
- alle zum Nachvollziehen der Arbeit notwendigen Ergebnisse und Daten (Quellcode, Buildskripte, Testcode, Testdaten usw.)
- Material für eine Abschlusspräsentation (ca. 20')

6. Termine

Es gelten folgende Termine:

- **16.09.19** Beginn der Bachelorarbeit
- Bis **06.01.20** Erfassung des Abstracts im Online-Tool <https://abstract.hsr.ch/> Die Studierenden geben den Abstract für die Diplomarbeitsbroschüre zur Kontrolle an ihren Betreuer/Examinator frei.
- Bis **06.01.20** Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract der Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei.
- **10.01.20 17:00** Abgabe des Berichts an den Betreuer
- **10.01.20 17:00** Hochladen aller Dokumente auf archiv-i.hsr
- Bis **14.02.20** Mündliche Prüfung
- Bis **16.02.20** Abgabe BA-Note

7. Beurteilung

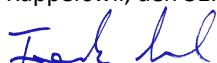
Eine erfolgreiche Bachelorarbeit erhält 12 ECTS-Punkte (1 ECTS Punkt entspricht einer Arbeitsleistung von ca. 25 bis 30 Stunden). Die Bewertung erfolgt gemäss der nachstehenden Kriterien.

Gesichtspunkt	Gewicht
1. Organisation, Durchführung (Projektplanung u. Nachführung Arbeit gemäss Projektplan, Selbstständigkeit, Einsatz, Zusammenarbeit mit Auftraggeber, Betreuer)	1/6
2. Bericht (Inhalt des Projektschlussberichts, Gliederung, Darstellung, Sprache der gesamten Dokumentation)	1/6
3. Inhalt	1/6
3.1 Problemanalyse (Vorstudie, Literaturstudium, Anforderungsspezifikation, Anforderungsanalyse, Domainanalyse)	

3.2 Lösungsentwurf (Lösungsvarianten und deren Beurteilung, Variantenentscheid, Konzept, Entwurf)	1/6
3.3 Realisierung und Test	1/6
4. Präsentation und Mündliche Prüfung zur Bachelorarbeit	1/6

Im Übrigen gelten die Abläufe und Regelungen Studien- und Bachelorarbeiten im Studiengang Informatik. Abläufe und Regelungen, Anleitung Dokumentation, Vorlagen und Erklärungen finden sich auf dem Skriptserver \\hsr.ch\root\alg\skripte\Informatik\Fachbereich\Bachelor-Arbeit_Informatik

Rapperswil, den 31.12.2019



Frank Koch

3 Technischer Bericht

3.1 Einleitung

3.1.1 Hintergrund

Luca Gubler arbeitete nebst dem Studium als IT-Supporter in der Schule Gossau ZH. Während eines Gespräches mit seinem Vorgesetzten bemängelte dieser, dass es keine guten E-Learning Plattformen für Schulen gibt. Zwar gebe es vereinzelt Lehrer oder Schulen, welche dies selber in die Hand nehmen und ein Open-Source-Tool wie Moodle verwenden, jedoch wird viel Zeit benötigt, um den gesamten Schulstoff digital zu erfassen. Solche Projekte werden oft auch nur halbherzig umgesetzt und geraten schnell in Vergessenheit. Meistens lässt auch die Qualität zu wünschen übrig.

3.1.2 Problemstellung / Vision

Aus diesem Gespräch heraus entstand die Idee, eine All-In-One Lernplattform für Schulen zu entwickeln. In einem ersten Schritt richtet sich die Plattform an die Sekundarstufe 1 und soll den ganzen Schulstoff in Form von Videos und Theorie-Zusammenfassungen zur Verfügung stellen. Diese Anwendung soll es Lernenden ermöglichen, sich ein neues Thema komplett selbstständig beizubringen. Um die erlernte Theorie mit praktischen Beispielen zu vertiefen, sollen Übungen zur Verfügung stehen. Den Lernenden soll es auch möglich sein, diese Plattform zu Hause zu verwenden.

Dies erlaubt es einer Lehrperson, die sogenannte Flipped-Classroom-Unterrichtsmethode anzuwenden. Mit dieser Methode erlernen Lernende die Theorie zu einem Thema selbstständig, was üblicherweise in Form von Hausaufgaben stattfindet. In der Schule werden dann Übungen oder Projektarbeiten gelöst. Die Lehrperson ist nicht mehr im klassischen Sinne als Wissensvermittler zuständig, sondern sie nimmt die Rolle eines Coaches ein. Lernende, welche Hilfe benötigen, können individuell unterstützt werden. Die anderen Lernenden werden dadurch aber nicht abgelenkt und können an einem anderen Thema weiter arbeiten.

Da die Lernenden aber weitgehend selbstständig arbeiten, kann die Lehrperson schnell den Überblick über den Stand der einzelnen Lernenden verlieren. Ein zentraler Punkt in der Schule sind jedoch die Prüfungen. Der Lehrperson sollen Statistiken zur Verfügung stehen, mit welchen der aktuelle Wissensstand der Lernenden fortlaufend überprüft werden kann. Dies erlaubt es einer Lehrperson, die Lernenden da zu unterstützen, wo sie am meisten Hilfe benötigen. So besteht auch die Möglichkeit, die gesamte Klasse auf ein Niveau zu heben, auf welchem der Wissensstand effektiv geprüft werden kann. Die Lehrperson hat einen Überblick über die Lernenden, auch wenn nicht genau vorgegeben wird, an welchem Thema diese arbeiten sollen.

Mit dieser Lernplattform hat man sich das Ziel gesetzt, Lehrperson und Lernende enger miteinander zu verbinden. Um diese Zusammenarbeit weiter zu fördern, soll ein aufgabenspezifisches Forum bereitgestellt werden. In diesem können Lernende Fragen zu einzelnen Aufgaben stellen und sich so gegenseitig helfen. Die Lehrperson muss nur dann eingreifen, wenn sich die Lernenden etwas falsch erklären oder niemand reagiert.

3.1.3 Aufgabenstellung

Bei dieser Problemstellung handelt es sich um eine grobe Beschreibung der gesamten Idee. Da die Zeit dieser Bachelorarbeit begrenzt ist, können nicht alle Punkte in vollem Umfang umgesetzt werden. Die Idee für eine solche Applikation ist auch nicht ganz neu und es gibt Ähnlichkeiten zu bereits existierenden Tools. Aus diesem Grund wurde nach einem Bereich gesucht, in welchem man sich von den existierenden Tools abheben kann.

In Zusammenarbeit mit Frank Koch, dem Betreuer dieser Bachelorarbeit und dem Moodle Experten der HSR, kam man zum Schluss, dass man sich mit dem aufgabenspezifischen Teil der Anwendung von solchen Tools abgrenzen kann. Lernende sollen in der Lage sein, sich ein Thema selbstständig beizubringen. Zu der bereitgestellten Theorie sollen Übungen bereit stehen, mit welchen die Theorie vertieft werden kann. Damit die Lernenden auch schwierige oder neue Aufgaben selbstständig erarbeiten können, soll eine Schritt-für-Schritt-Anleitung zur Verfügung stehen. So können neue wie auch schwierige und komplizierte Aufgaben selbstständig erarbeitet werden.

3.1.4 Zielgruppen

Für den Moment ist die Aufgaben-Coach für die Sekundarstufe 1 ausgelegt. Später soll es jedoch auch möglich sein, den Aufgaben-Coach in anderen Schulstufen einzusetzen.

3.2 Stand der Technik

3.2.1 Existierende Produkte

Es gibt bereits einige Tools, welche Lerninhalte für Schüler zur Verfügung stellen. Diese lassen sich grundsätzlich in die beiden Kategorien "kommerzielle Anbieter" und "Open-Source-Projekte" einteilen.

Kommerzielle Anbieter

Sofatutor gehört wohl zu den grössten und bekanntesten Anbietern von Lerninhalten in Deutschland, Österreich und der Schweiz (DACH). Sofatutor richtet sich jedoch hauptsächlich an Schüler, welche Nachhilfe in einem bestimmten Fach benötigen. In einem begrenzten Rahmen ist es auch möglich, Sofatutor in den Schulalltag einzubinden⁵. Lehrer können ihren Schülern zum Beispiel einzelne Videos oder Übungen freischalten. Dazu muss aber ein Link generiert und an die Schüler verschickt werden, welcher nur 14 Tage lang aktiv ist. Somit sind die Schüler nicht frei im Lernen und die Zusammenarbeit der Schüler untereinander wird nicht gefördert.

EF Class ist ein weiterer Anbieter von Lerninhalten. Dieses Tool ist aber speziell für den Englischunterricht ausgelegt⁶. Möchte eine Lehrperson die Flipped-Classroom-Methode anwenden, müssten gleich mehrere solcher Tools eingekauft werden. Zum einen ist das aus finanzieller Sicht gesehen ein negativer Punkt und zum anderen ist es schwer, einen Gesamtüberblick zu behalten, da die Informationen über den Wissensstand der Lernenden auf mehrere Plattformen verteilt ist.

Open-Source-Projekte

Open-Source-Anwendungen, wie zum Beispiel Moodle, fallen in die Kategorie der LMS, welche eine Unterkategorie von Content Management System (CMS) sind. Unter einem CMS versteht man folgendes⁷:

A content management system (CMS) is a software application or set of related programs that are used to create and manage digital content. CM-Ses are typically used for enterprise content management (ECM) and web content management (WCM). An ECM facilitates collaboration in the workplace by integrating document management, digital asset management and records retention functionalities, and providing end users with role-based access to the organization's digital assets. A WCM facilitates collaborative authoring for websites. ECM software often includes a WCM publishing

⁵Erfolgreich unterrichten mit Sofatutor. URL: <https://www.sofatutor.ch/lehrer-und-schulen> [Besucht am 20. Oktober 2019].

⁶English lessons, designed for you.

⁷content management system (CMS). URL: <https://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS> [Besucht am 29. Dezember 2019].

functionality, but ECM webpages typically remain behind the organization's firewall.

Ein Beispiel für ein bekanntes Enterprise Content Management (ECM) ist SharePoint⁸. Dieses Tool ermöglicht es, Dateien zu speichern oder zu teilen und ermöglicht einen rollenbasierten Zugriff. Wie der Name jedoch sagt, ist es hauptsächlich auf Firmen ausgelegt, welche intern Dateien verwalten möchten.

Ein Web Content Management (WCM) unterscheidet sich, weil damit vor allem Websites für den Internetauftritt erstellt werden können. Dabei braucht man auch praktisch kein Vorwissen, um Websites zu erstellen. Üblicherweise stellen solche Plattformen What You See Is What You Get (WYSIWYG) Editoren zur Verfügung, mit welchen eine Webseite zusammen geklickt werden kann. Gemäss einer Statistik vom Internet Live Stats⁹ gibt es über 1.7 Milliarden unique Websites. Unter "unique Websites" wird eine Webseite mit einzigartigem Hostnamen verstanden. Diese Anzahl ist aber hauptsächlich wegen dem Gebrauch unterschiedlicher CMS so hoch, da auch Laien sehr schnell Webseiten erstellen können. Unter allen CMS ist WordPress das mit Abstand meist eingesetzte Tool und besitzt mit über 27 Millionen Webseiten einen Marktanteil von 53.3%. Der zweite Platz in der Rangliste belegt Joomla! mit gerade einmal 3.8 Millionen Webseiten und einem Marktanteil von 7.5%¹⁰.

Ein LMS grenzt sich von WCM ab, da es speziell auf die Bereitstellung von Online-Trainings ausgelegt ist. Ein grosser Vorteil solcher Tools ist, dass sie sehr flexibel eingesetzt werden können. Eine Firma kann zum Beispiel interne Weiterbildungen anbieten, während eine Schule den Schülern die Theorie zu einzelnen Themen vermittelt.

Eines der wohl bekanntesten LMS ist Moodle¹¹, welches auch an der HSR zum Einsatz kommt. Moodle belegt jedoch nur Platz 19 von 20 der besten LMS Systeme, basierend auf User Experience¹². Moodle schneidet mit einer Bewertung von 70% ab, während Loop¹³ den 1. Platz mit einer Bewertung von 90% belegt. Es ist jedoch nicht ersichtlich, warum Moodle schlechter abschnitt, denn 88% der Benutzer würden Moodle weiterempfehlen. Moodle ist auch das bekannteste LMS mit etwa 68 Millionen Benutzer und

⁸ *What is SharePoint?* URL: <https://support.office.com/en-us/article/what-is-sharepoint-97b915e6-651b-43b2-827d-fb25777f446f> [Besucht am 29. Dezember 2019].

⁹ *Total number of Websites.* URL: <https://www.internetlivestats.com/total-number-of-websites/> [Besucht am 4. Januar 2020].

¹⁰ *CMS Usage Distribution on the Entire Internet.* URL: <https://trends.builtwith.com/cms/traffic/Entire-Internet> [Besucht am 29. Dezember 2019].

¹¹ *Moodle - Open-source learning platform.*

¹² *The Best Learning Management Systems based on User Experience.* URL: <https://elearningindustry.com/directory/software-categories/learning-management-systems/best/user-experience?version=2018-02-22> [Besucht am 22. Dezember 2019].

¹³ *Loop - The LMS that drives performance.* URL: <https://www.loop.co/> [Besucht am 22. Dezember 2019].

55'000 deployten Webseiten. Dafür gibt es mehrere Gründe. Zum einen ist Moodle Open Source und somit gratis. Das ist sicherlich ein grosser Pluspunkt, vorallem für Schulen, welche nur über ein begrenztes Budget verfügen. Ein weiterer Grund ist, dass Moodle konfigurierbar und flexibel ist und sehr viele Features anbietet. So gibt es zum Beispiel über 500 Plugins, welche installiert werden können.

3.2.2 Flipped Classroom

Plattformen wie Moodle erlauben es einer Lehrperson, das Flipped Classroom Prinzip in den Unterricht zu integrieren. In der klassischen Unterrichtsmethode besteht der Unterricht aus einer Phase, in welcher die Lehrperson den Schülern die Theorie zu einem neuen Thema beibringt. In der zweiten Phase lösen die Schüler basierend auf der erlernten Theorie Übungen. Diese finden dann meistens zu Hause in Form von Hausaufgaben statt. Flipped Classroom vertauscht nun diese beiden Phasen. Die Lerninhalte werden zu Hause von den Schülern erarbeitet. Dies kann in Form von Videos oder Theoriezusammenfassungen stattfinden. Die Anwendung, also das Lösen von Aufgaben, findet dann während dem Unterricht statt¹⁴.

Da die Schüler die Theorie zu Hause lernen, hat eine Lehrperson während des Unterrichts viel mehr Zeit, sich direkt und individuell mit den Schülern zu befassen. Die Lehrperson kann jene Schüler coachen, welche ein Thema noch nicht ganz verstehen. Die Schüler, welche das Thema aber verstehen, können bereits mit dem nächsten Thema beginnen und werden nicht aufgehalten.

¹⁴ *The Flipped Class Manifest*. URL: <http://www.thedailyriff.com/articles/the-flipped-class-manifest-823.php> [Besucht am 20. Oktober 2019].

Auf den ersten Blick bietet Flipped Classroom viele Vorteile, hat jedoch auch ein paar Nachteile. Nachfolgend befindet sich eine kurze Übersicht mit möglichen Vor- und Nachteilen¹⁵.

Vorteile

- Die Schüler haben mehr Kontrolle über ihr eigenes Lernen und können das Lern-tempo selber festlegen. Fragen oder Unklarheiten können dann im Unterricht mit einer Lehrperson besprochen werden. Jene Schüler, welche ein Konzept nicht auf Anhieb verstehen, können sich die Theorie in Ruhe nochmals anschauen, ohne den Anschluss an den Unterricht zu verlieren.
- Während des Unterrichts gibt es weniger Frontalunterricht, dafür mehr Übungen und Projektarbeiten. Die Schüler können sich so gegenseitig unterstützen und beim Lernen helfen. Die Lehrperson ist nur noch als Coach tätig und unterstützt einzelne Schüler.
- Die Lektionen sind immer verfügbar. Kann ein Schüler krankheitshalber nicht in den Unterricht kommen, so kann die Theorie dennoch erlernt werden.
- Weil die Lektionen praktisch 24/7 verfügbar sind, können die Eltern auch zu Hause schauen, was genau die Kinder im Unterricht machen. So können sie sich auch selbst besser vorbereiten, falls sie ihre Kinder unterstützen wollen.

Nachteile

- Ein grosser Nachteil ist, dass die Infrastruktur vorhanden sein muss. In der Schule werden den Schülern unter Umständen Tablets verteilt, auf welchen sie die Videos anschauen können. Zu Hause braucht der Schüler aber ein eigenes Tablet und einen Internet Anschluss. Es kann durchaus vorkommen, dass diese Infrastruktur nicht in jedem Haushalt vorhanden ist.
In der Schule sollte dies jedoch kein Problem sein, da die Digitalisierung der Schulen immer weiter voran schreitet¹⁶.
- Die Schüler werden dazu aufgefordert, sich zu Hause vorzubereiten. Es kann aber nicht garantiert werden, dass die Schüler dies auch tatsächlich tun.
- Bei Flipped Classroom verbringen die Schüler automatisch mehr Zeit vor einem Bildschirm. Es kann Eltern geben, welche dies nicht als positiv erachten, wenn ihre Kinder längere Zeit vor einem Bildschirm sitzen.

¹⁵ 10 Pros And Cons Of A Flipped Classroom. URL: <https://www.teachthought.com/learning/10-pros-cons-flipped-classroom/> [Besucht am 20. Oktober 2019].

¹⁶ Jörg Krummenacher. *Die Städte rüsten die Schulen digital auf*. URL: <https://www.nzz.ch/schweiz/staedte-und-digitale-schule-ld.1489721> [Besucht am 20. Oktober 2019].

3.3 Eigener Lösungsansatz

3.3.1 Konzeption

Zu Beginn der Bachelorarbeit tauchte die Frage auf, ob ein Plugin für Moodle entwickelt werden soll, oder ob eine komplett neue Anwendung erstellt wird. Es gab jedoch mehrere Gründe, warum man sich für die Entwicklung einer neuen Anwendung entschieden hat. Ausschlaggebend war vorallem ein Post auf reddit¹⁷ des Benutzers "greenek_". Seiner Aussage zufolge besteht Moodle aus fast 600'000 Zeilen PHP Code. Zudem ist der Moodle Core 15 Jahre alt und wurde nie wirklich refactored. Des Weiteren gibt viel duplizierten Code. Für Klassen gibt es zum Beispiel 20 Methoden, welche alle das gleiche machen.

Um sicher zu gehen, dass die neue Arbeit zum Erfolg wird, wollte man kein grösseres Risiko eingehen. Aus diesem Grund entschied man sich für die Entwicklung einer neuen Anwendung. Mit Aufgaben-Coach möchte man sich von bereits existierenden Anwendungen abgrenzen. Aufgaben-Coach richtet sich direkt an Schulen, welche so die Flipped Classroom Methode in den Unterricht integrieren können.

Diese Anwendung wurde mit den Bedürfnissen von Lehrenden und Lernenden entworfen. Für die Konzeption war es hilfreich, dass die HSR selber eine Moodle Seite betreibt. So konnte bereits während dem Studium Erfahrung mit einem LMS gesammelt werden. Dies diente teilweise als Inspiration für den Entwurf der Mockups.

3.3.2 Zentrale Elemente

Benutzer und Rechte

Eine Person kann einer der drei Rollen "Administrator", "Lehrer" oder "Schüler" zugewiesen werden. Je nach Rolle ist der Zugriff auf einzelne Seiten eingeschränkt.

Der Administrator ist für das Erfassen der einzelnen Personen und der Zuweisung in die Klassen zuständig.

Lehrpersonen können ihre eigenen Klassen verwalten und Aufgaben erstellen oder bearbeiten. Zudem sind sie berechtigt, die Statistiken ihrer Klassen anzuschauen.

Schüler sind in der Lage, den freigegebenen Inhalt zu sehen und die dazugehörigen Aufgaben zu lösen. Zusätzlich haben sie auch Zugriff auf das Forum. Webseiten zum einsehen der Statistiken, bearbeiten von Klassen oder anpassen von Aufgaben sind gesperrt.

¹⁷ "Moodle. Not even once." *What is your the worst project in career?* URL: <https://www.reddit.com/r/webdev/comments/2l3hy9/> [Besucht am 20. Oktober 2019].

Lerninhalte

Die Lehrer sind in der Lage, einzelne Fächer für die Schüler einer Klasse freizuschalten. Im momentanen Zustand können Lehrpersonen nicht selbst Anpassungen an den Inhalten vornehmen. Nur die Betreiber von Aufgaben-Coach sind in der Lage, neue Theorieinhalte zu erstellen oder zu bearbeiten.

Aufgaben

Lehrpersonen können neue Aufgaben erstellen, welche anschliessend von den Schülern gelöst werden können. Pro Aufgabe können mehrere Fragen erstellt werden. Für jede dieser Fragen können Hilfestellungen erfasst werden. Diese sollen es dem Schüler ermöglichen, auch schwierige Fragen selber zu beantworten.

Hat ein Schüler eine Aufgabe gelöst und abgegeben, kann diese von einer Lehrperson korrigiert werden. Dazu kann mit Filteroptionen nach abgegebenen Aufgaben gesucht werden. Wählt die Lehrperson die abgegebene Aufgabe eines Schülers aus, kann er für jede einzelne Frage sagen, ob diese richtig oder falsch beantwortet wurde und dafür Punkte vergeben. Sollte die Aufgabe jedoch falsch gelöst worden sein, kann diese mit einem Feedback an den Schüler zurückgewiesen werden. Dieser hat dann nochmals die Möglichkeit, die Aufgabe zu verbessern.

Statistiken

Mit allen abgegebenen Aufgaben wird eine Statistik erstellt. Anhand dieser Statistiken kann eine Lehrperson erkennen, wie gut die Klasse oder ein einzelner Schüler ein Thema versteht. Dies ermöglicht der Lehrperson auch, gezielt auf diese Schüler einzugehen und die Wissenslücken zu schliessen.

Forum

Einem Schüler sollte es anhand der Hilfestellungen möglich sein, eine Aufgabe selbstständig zu lösen. Sollten dennoch Fragen oder Unklarheiten auftreten, kann er diese in einem Forum stellen, welches sich rund um diese einzelne Aufgabe dreht. In diesem Forum befinden sich auch bereits früher gestellte Fragen, welche die Frage des Schülers vielleicht gleich beantwortet.

4 Anforderungen

4.1 Personas

Unter der Definition von Personas¹⁸ versteht man:

Personas (lat. Maske) sind Nutzermodelle, die Personen einer Zielgruppe in ihren Merkmalen charakterisieren. Sie können z. B. einem Entwicklungsteam aufgrund ihrer umfangreichen Beschreibung helfen, sich in die Lage der potenziellen Nutzer zu versetzen und diese Perspektive während des gesamten Designprozesses leicht zu vertreten. Sie werden mit einem Namen, einem Gesicht, einer Funktion, einem Werdegang und einem Privatleben versehen. Personas verfügen über Ziele und Verhaltensweisen, haben Vorlieben und Erwartungen.

Diese Personas wurden erstellt, um ein klareres Bild vermitteln zu können, welche Funktionalitäten durch den Aufgaben-Coach ermöglicht werden und wie diese Anwendung eingesetzt werden könnte.

4.1.1 Persona 1

Persönliches Profil

Abirsana ist Lehrerin an der Schule Gossau und unterrichtet die Fächer Mathematik, Deutsch und Englisch.

Situation

Um den Unterricht interessanter zu gestalten, greift Abirsana oft auf digitale Medien, wie zum Beispiel YouTube Videos, zurück. Oftmals ist jedoch das Material nicht genau auf den Schulstoff angepasst oder die Qualität lässt zu wünschen übrig. Zudem braucht sie sehr lange, um passende Videos zu finden.

Szenario

Im Rahmen des Unterrichts soll es Abirsana möglich sein, die Aufgaben-Coach Webanwendung zu verwenden. Sie soll in der Lage sein, einzelne Fächer und Theorieinhalte ihrer Klasse zur Verfügung zu stellen. Zudem kann sie eigene Übungen erfassen, welche die Schüler direkt über die Webanwendung lösen können. Sind die Aufgaben abgegeben, kann sie die einzelnen Aufgaben anschauen und bewerten. Falls ihr auffällt, dass ein gewisser Schüler oder die ganze Klasse etwas nicht versteht, kann sie eingreifen und Wissenslücken schliessen.

¹⁸ *Definition Personas*. URL: <https://www.onlinemarketing-praxis.de/glossar/personas> [Besucht am 13. November 2019].

4.1.2 Persona 2

Persönliches Profil

Salina ist Schülerin an einer Schule. Seit längerer Zeit hat sie jedoch Mühe mit dem Schulstoff und ist nicht zufrieden mit ihrer Schulleistung.

Situation

Salina versteht den Lernstoff oft nicht, wenn ihr Lehrer ihr etwas erklärt. Zu Hause verbringt sie dann viel Zeit im Internet, um einzelne Themen zu lernen. Sie hat aber weder die Zeit noch die Lust dazu, zu Hause nochmals die gesamte Theorie anschauen zu müssen und nach guten Videos zu suchen.

Szenario

Nach kurzer Zeit hat Salina die Lehrer an ihrer Schule dazu überredet, die Aufgaben-Coach Plattform einzusetzen. Somit ist Salina freier im Lernen. In der Schule kann sie sich selbstständig ein Thema beibringen. Falls sie dennoch etwas nicht versteht, kann sie direkt auf eine Lehrperson zugehen, welche ihr dann das Kapitel vielleicht noch etwas genauer erklärt.

Zudem fällt es Salina einfacher Fragen zu stellen. Zuvor hatte sie immer das Gefühl, die ganze Klasse aufzuhalten. Nun kann sie in Ruhe ihre Fragen stellen, während die anderen Schüler selbstständig weiter lernen können. Des Weiteren verbringt sie zu Hause auch nicht mehr so viel Zeit mit dem Suchen von Videos, da sich alle benötigten Theorieinhalte auf Aufgaben-Coach befinden.

4.1.3 Persona 3

Persönliches Profil

Liam ist ebenfalls Lehrer an einer Schule. Er mag es am liebsten, wenn er genau weiss, was vor sich geht. Überraschungen kann er gar nicht leiden und besonders dann nicht, wenn es sich dabei um schlechte Resultate seiner Schüler bei einer Prüfung handelt.

Situation

Immer wieder kommt es vor, dass seine Schüler eine Prüfung schlechter abschneiden, als er sich das erhofft. Dem will er nun ein Ende setzen und sucht nach einer Lösung.

Szenario

Um in Zukunft unangenehme Überraschungen wie diese zu vermeiden, will er eine Möglichkeit haben, genauere Details über den aktuellen Wissensstand seiner Schüler zu sammeln. Um dieses Ziel zu erreichen, möchte er keinen grossen Mehraufwand auf sich nehmen und will den Unterricht so wie bisher weiterführen.

4.2 Funktionale Anforderungen

4.2.1 Use Case Diagramm

In der Abbildung 1 ist das Use Case Diagramm abgebildet. Dieses Diagramm soll die Abhängigkeiten zwischen Aktoren, den einzelnen Use Cases und externen Systemen visualisieren.

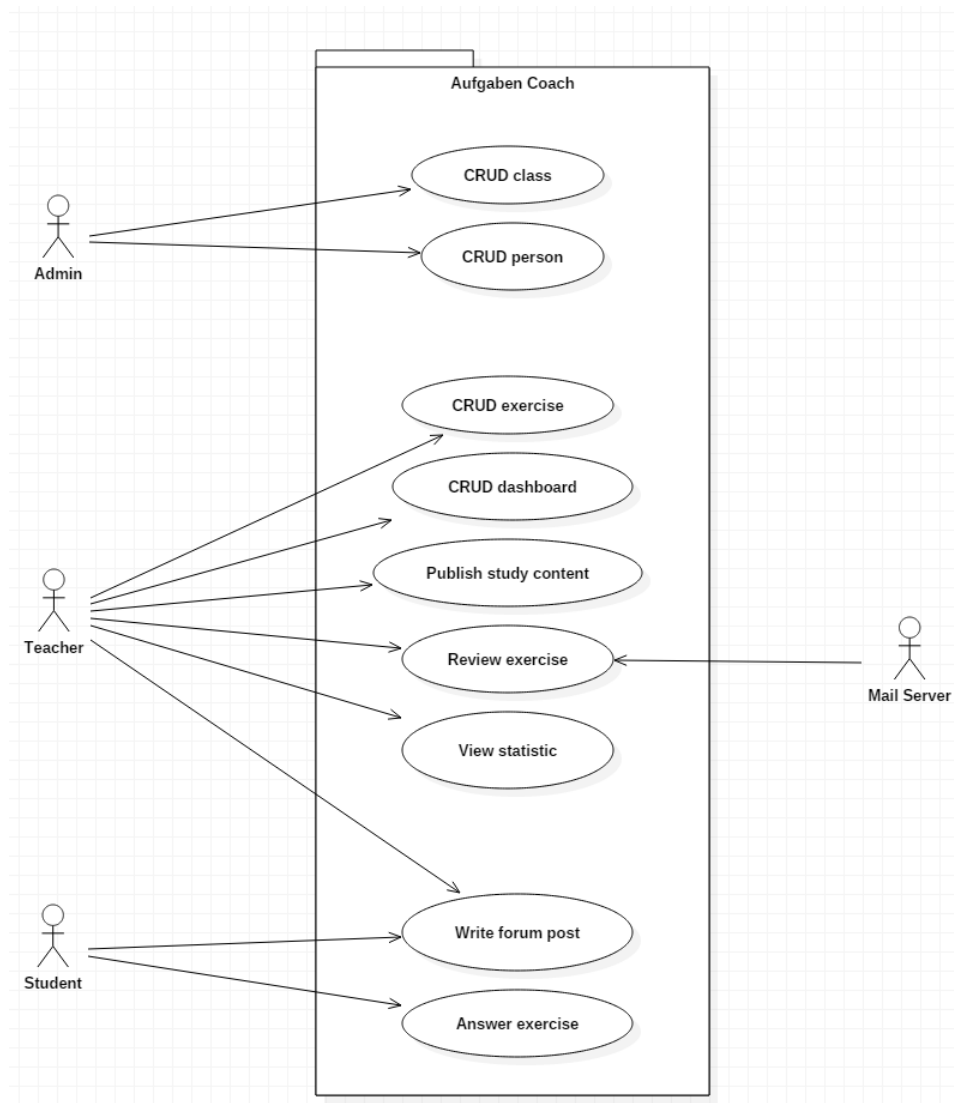


Abbildung 1: Use Case Diagramm

4.2.2 Aktoren

Die Aktoren, welche mit der Anwendung interagieren, sind in der Tabelle 1 beschrieben.

Aktor	Beschreibung
Administrator	Der Administrator ist für die Verwaltung der Klassen, Lehrer und Schüler zuständig. Er kann Klassen erstellen sowie Lehrpersonen und Schüler diesen Klassen zuweisen.
Lehrer	Der Lehrer verwaltet die ihm zugewiesenen Klassen. Er kann Fächer für seine Klassen freischalten sowie Aufgaben erstellen und diese der Klasse zuweisen. Zusätzlich kann er Statistiken einsehen, die ihn über den aktuellen Wissensstand seiner Klasse informieren.
Schüler	Die Schüler haben Zugriff auf freigeschaltete Lerninhalte und können diese anschauen. Falls Fragen auftreten, können diese im aufgaben-spezifischen Forum gestellt werden. Zusätzlich können Aufgaben gelöst werden.
Mail Server	Der Mail Server dient dem Versenden von Nachrichten, falls zum Beispiel eine Lehrperson ein Feedback an einen Lernenden schicken möchte.

Tabelle 1: Aktoren

4.2.3 Use Case Beschreibung

Nachfolgend werden die einzelnen Use Cases stichwortartig beschrieben.

CRUD person

- Hauptszenario:
 - Der Administrator kann neue Benutzer erstellen. Diese können entweder die Rolle "Schüler" oder "Lehrer" haben.
- Alternatives Szenario:
 - Der Administrator entfernt einen Benutzer aus dem System.
 - Der Administrator ändert Angaben eines Benutzers.

CRUD class

- Hauptszenario:
 - Der Administrator erstellt eine neue Klasse, welcher Lehrpersonen und Schüler zugewiesen werden können.
- Alternatives Szenario:
 - Nach Ablauf eines Schuljahres kann der Administrator Klassen aus dem System entfernen.
 - Der Administrator kann einzelne Schüler aus einer Klasse entfernen und diese einer anderen Klasse zuweisen.

CRUD exercise

- Hauptszenario:
 - Der Lehrer kann neue Aufgaben erstellen und diese einem Thema zuweisen. Pro Aufgabe gibt es mehrere Fragen, welche wiederum mehrere Hilfestellungen enthalten können. Für jede Frage kann auch eine maximale Punktzahl angegeben werden.
- Alternatives Szenario:
 - Der Lehrer passt einzelne Fragen oder Hilfestellungen an oder fügt neue Fragen einer Aufgabe hinzu.

CRUD dashboard

- Hauptszenario:
 - Der Lehrer passt den Wochenplan einer Klasse an und fügt neue Aufgaben hinzu.
- Alternatives Szenario:
 - Der Lehrer verschiebt den Abgabetermin einer Aufgabe auf einen anderen Tag.

Publish study content

- Hauptszenario:
 - Der Lehrer kann Fächer für einzelne Klassen freischalten.
- Alternatives Szenario:
 - Freigeschaltene Fächer können wieder entfernt werden.

View statistics

- Hauptszenario:
 - Der Lehrer hat Einsicht in die Statistiken einzelner Klassen oder Schüler.

Write forum post

- Hauptszenario:
 - Treten beim Lösen von Aufgaben Unklarheiten auf, kann ein Schüler eine entsprechende Frage direkt im aufgabenspezifischen Forum stellen.
- Alternatives Szenario:
 - Ein Schüler sieht eine Frage im Forum, zu welcher er die Lösung weiss. Er kann auf den Beitrag seines Schulkameraden antworten und ihm so helfen.

Review exercise

- Hauptszenario:
 - Der Lehrer hat die Möglichkeit, gelöste Aufgaben der Schüler zu korrigieren und zu bewerten. Nach der Bewertung kann die Abgabe akzeptiert werden.
- Alternatives Szenario:
 - Der Lehrer stellt beim Korrigieren der Abgabe zu viele Fehler fest und entscheidet sich die Abgabe an den Schüler zurückzuweisen. Er kann für den Schüler ein Feedback schreiben, welches per Mail versendet wird.

Answer exercise

- Hauptszenario:
 - Der Schüler kann die Aufgaben von freigegebenen Fächern lösen. Sind alle Fragen beantwortet, kann die Übung abgegeben und vom Lehrer korrigiert werden.
- Alternatives Szenario:
 - Falls der Lehrer die Übung eines Schülers zurückweist, kann dieser die Fragen nochmals überarbeiten.

4.3 Nicht Funktionale Anforderungen

Bei der Erstellung der einzelnen Non Functional Requirements (NFR) wird auf FURPS zurückgegriffen. FURPS ist ein Akronym für Functionality, Usability, Reliability, Performance und Supportability. Dieses Modell wurde im Jahre 1992 von HP entwickelt und dient zur Priorisierung der Software Requirements¹⁹.

FURPS	Titel	Beschreibung
Functionality	Reusability	Die Anwendung soll so entwickelt werden, dass sie an mehreren Schulen eingesetzt werden kann.
	Security	Die Rechte der einzelnen Benutzer sind eingeschränkt. Niemand hat per Default Zugriff auf alles. Die Webseite ist nur über HTTPS erreichbar.
Usability	Responsiveness	Für Schüler soll die Applikation sowohl auf Tablets wie auch auf Mobile Phones verfügbar sein.
Reliability	Availability	Die Anwendung ist stabil und stürzt nicht ab.
	Fault Tolerance	Die Anwendung soll auch bei unerlaubten Anfragen weiterhin verfügbar sein und die korrekten Ergebnisse liefern.
Performance	Availability	Die Anwendung soll ohne abzustürzen von 300 Benutzern zeitgleich verwendet werden können.
	Response Time	Wechselt der Benutzer zwischen einzelnen Seiten, soll die neue Seite in maximal einer Sekunde geladen werden.
Supportability	Maintainability	Die Applikation soll so gebaut werden, dass sie auch in Zukunft gewartet und ausgebaut werden kann.
	Installability	Die Applikation wird so gebaut, dass sie innerhalb einer Stunde auf einem neuen System installiert werden kann.

Tabelle 2: Non Functional Requirements

¹⁹Deborah L. Grady Robert B. und Caswell. *Software metrics: establishing a company-wide program*. URL: <https://archive.org/details/softwaremetricse00grad/page/159> [Besucht am 23. Oktober 2019]. 1987.

5 Architektur

5.1 System Übersicht

Für die Umsetzung der Anwendung gibt es zwei Optionen. Man kann zwischen einer Native App, also einer klassischen Client-Server Anwendung, oder einer Web Anwendung, entscheiden.

Native Apps haben zwar einige Vorteile. Die Integration mit anderen Anwendungen ist so zum Beispiel einfacher. Zudem wäre die Monetarisierung auch um einiges einfacher, denn eine Native App kann sehr gut im App Store platziert werden. So steigt auch gleich die Bekanntheit der Applikation²⁰.

Die Entscheidung bei der Entwicklung von Aufgaben-Coach fiel jedoch sehr schnell auf eine Web Anwendung. Viele Schulen haben zwar Apple iPads im Einsatz. Trotzdem kann es Schulen geben, welche mit Android Tablets arbeiten. Mit einer Native App müsste für jedes Betriebssystem ein separates Programm erstellt werden.

Hinzu kommt noch die Anforderung der Lehrer. Da diese hauptsächlich auf Notebooks oder Desktop PCs arbeiten, müsste für sie auch eine separate Anwendung entwickelt werden. Mit einer Web Anwendung hat man den Vorteil, nur eine Anwendung entwickeln zu müssen, auf welche alle Benutzer Zugriff haben.

Ein weiterer Vorteil von Web Anwendungen ist, dass im vorhinein kein Programm auf dem Device installiert werden muss. Alles, was ein Benutzer braucht, ist ein Internetzugang und ein Tablet. Dies senkt zugleich die Supportkosten der Schule, da so keine Zeit für die Installation oder das Updaten verbraucht wird.

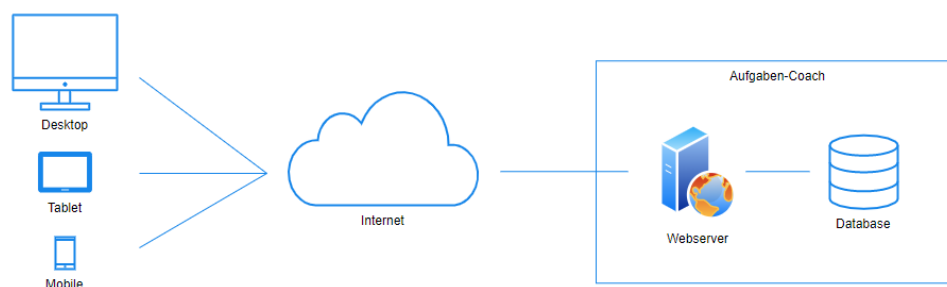


Abbildung 2: System Overview

²⁰ *Native app vs. progressive web app (PWA): Everything you need to know.* URL: <https://www.adjust.com/blog/native-app-vs-progressive-web-app/> [Besucht am 13. Dezember 2019].

5.2 Domain Model

In der Abbildung 3 ist das vollständige Domain Model abgebildet. Dem Leser soll so ein Überblick über die einzelnen Teilgebiete der Anwendung gegeben werden.

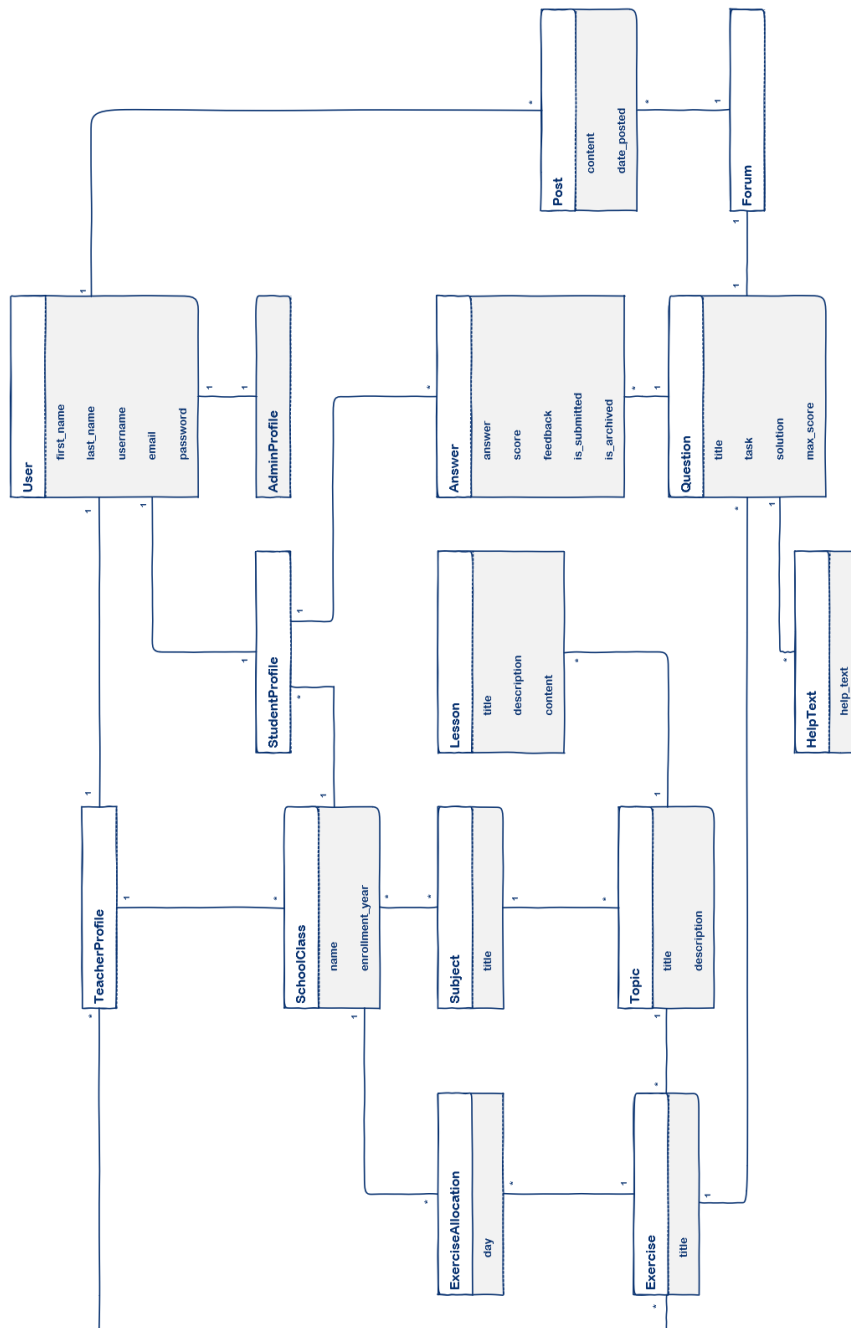


Abbildung 3: Vollständiges Domain Model

School

Ein zentraler Punkt der Anwendung ist die Schule und das User Management. Die User sind einer der drei Rollen Student, Teacher oder Admin zugeteilt. Da jede Rolle unterschiedliche Rechte hat, konnte nicht das Standard User Model von Django selbst verwendet werden.

Es stehen drei Möglichkeiten zur Verfügung, wie unterschiedliche User Rollen implementiert werden können.

1. Unterscheidung der User anhand eines Boolean Flags
2. Unterscheidung der User anhand eines Choices Field
3. Erstellung eines Profile Models mit einer 1 zu 1 Beziehung zum User Model

Die erste Variante hat den Vorteil, dass ein User mehrere Rollen gleichzeitig haben kann. Im Gegensatz zur ersten Variante kann ein User bei der zweiten Variante nur eine Rolle annehmen. Diese beiden Varianten haben jedoch den Nachteil, dass jeder User die gleichen Daten hat. Möchte man einem Schüler eine Matrikelnummer zuweisen, existiert dieses Feld implizit auch für alle Lehrer und Admins.

Um diesem Problem aus dem Weg zu gehen, entschied man sich für die dritte Variante. Pro Rolle existiert ein Profile Model. Möchte man nur einer bestimmten Rolle ein Feld zuweisen, so kann dieses Feld im entsprechenden Profile Model hinzugefügt werden²¹.

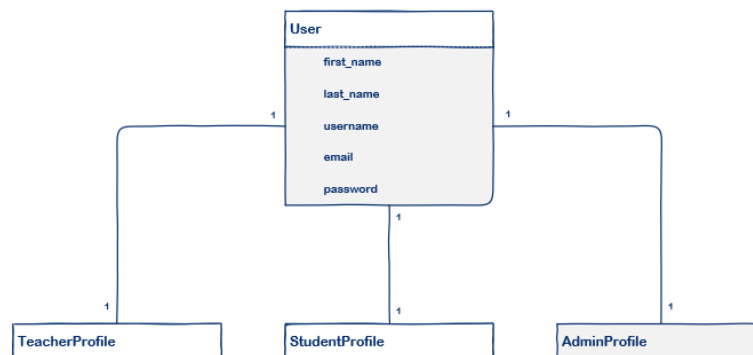


Abbildung 4: Domain Model - User Abschnitt

²¹ How to Implement Multiple User Types with Django. URL: <https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html> [Besucht am 20. November 2019].

Der Admin kann als einziger User Typ für sich alleine existieren. Schüler und Lehrer werden Klassen zugewiesen. Ein Lehrer kann gleichzeitig mehreren Klassen zugewiesen sein, ein Schüler kann sich jedoch nur in einer Klasse befinden. Für den aktuellen Stand reicht diese Funktionalität aus. Es gibt jedoch den Use Case, in welchem Schüler aus unterschiedlichen Klassen Fächer zusammen belegen. Dies ist zum Beispiel der Fall, wenn Migrationskinder aus unterschiedlichen Klassen zusammen kommen und gemeinsam ein Deutsch Nachhilfe Fach besuchen. Für diesen Fall kann man dem Student Profile ein Boolean Flag zuweisen. Falls dieses Flag gesetzt ist, hat der Schüler Zugriff auf Nachhilfe Fächer.

Wenn dieses Projekt als Startup weiter verfolgt wird, dann würde pro Schule eine neue Instanz der Anwendung deployed werden. So entstehen keine Multi Tenancy Probleme. Möchte man in Zukunft jedoch einen Multi Tenancy Ansatz verfolgen, müsste das Domain Model dementsprechend angepasst werden. Ein möglicher Lösungsansatz wäre, wenn jeder Schule eine eigene Subdomain, wie zum Beispiel schule-gossau.aufgaben-coach.ch, vergeben würde. Anhand der Subdomain können dann die einzelnen User identifiziert werden²².

²² *Building Multi Tenant Applications with Django*. URL: <https://books.agiliq.com/projects/django-multi-tenant/en/latest/shared-database-shared-schema.html> [Besucht am 22. Dezember 2019].

Study Content

Ein zentraler Punkt dieser Anwendung sind die Schulfächer. Pro Subject gibt es mehrere Topics. Topics wiederum können mehrere Lessons haben.

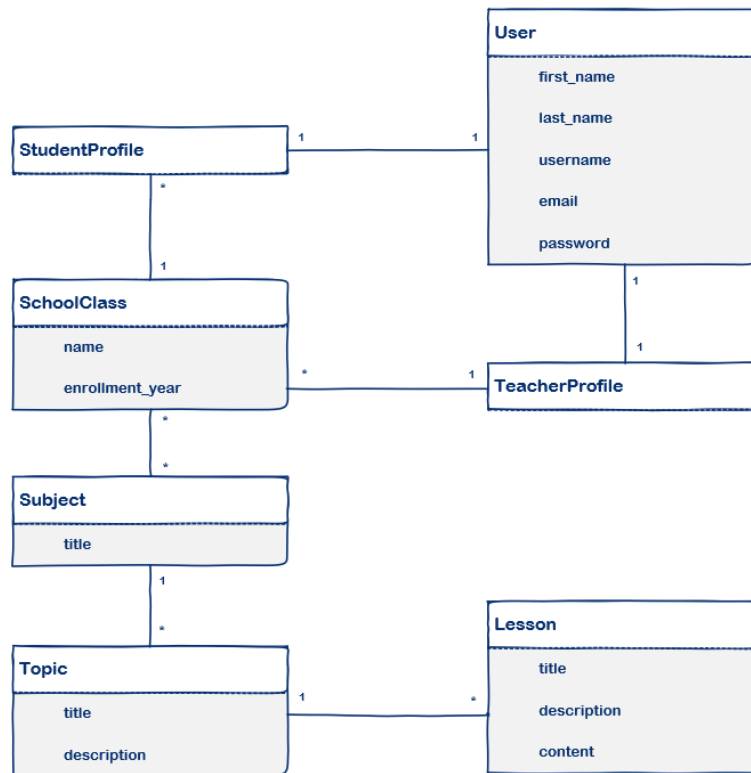


Abbildung 5: Domain Model - Schulfächer

Ein Subject kann einer Klasse zugewiesen werden. Erst dann hat die Klasse Zugriff auf Topics und Lessons. Im Moment ist es nicht möglich, einzelne Topics oder Lessons freizuschalten. Möchte man diese Funktionalität in Zukunft jedoch haben, muss eine neue Zwischentabelle in der Datenbank erstellt werden.

Exercise

Das Herzstück der Anwendung sind die Exercises. Lehrer können Übungen erstellen, welche aus mehreren Teilaufgaben bestehen, welche wiederum mehrere Hilfestellungen haben.

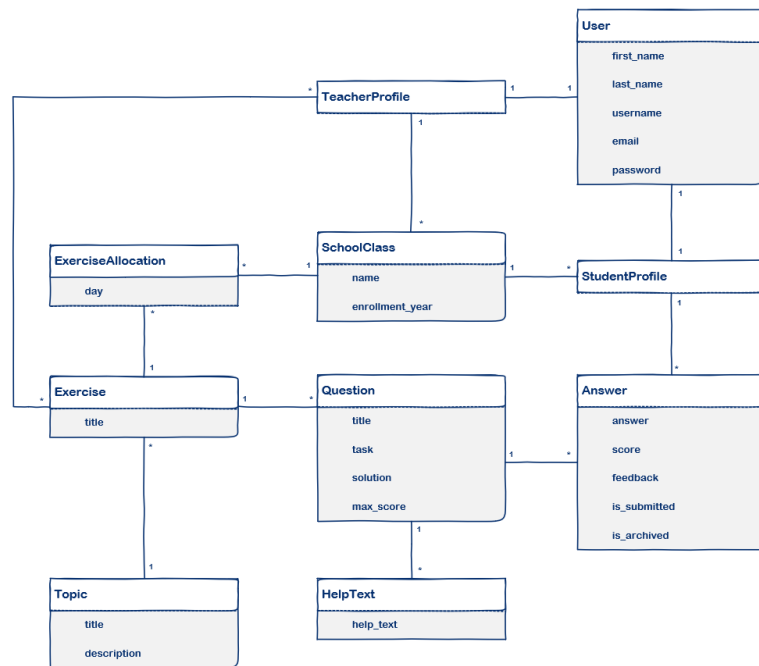


Abbildung 6: Domain Model - Aufgaben

Die Tabelle Answer dient zur Speicherung der Antworten von Schülern. Die beiden Felder "is_submitted" und "is_archived" sind für den Status der Aufgaben. Ist eine Aufgabe submitted, also abgegeben, kann sie nicht weiter vom Schüler bearbeitet werden. Erst wenn Aufgaben abgegeben sind, können sie vom Lehrer angeschaut werden. Wurden die Aufgaben vom Lehrer korrigiert, bekommen sie den Status "is_archived".

Forum

Für jede Frage in einer Übung gibt es ein Forum. Die Schüler haben einen Ort, in welchem themenspezifische Fragen gestellt werden können. Da es für jede Frage ein Forum gibt, werden die Schüler auch nicht von anderen Forumsbeiträgen abgelenkt.

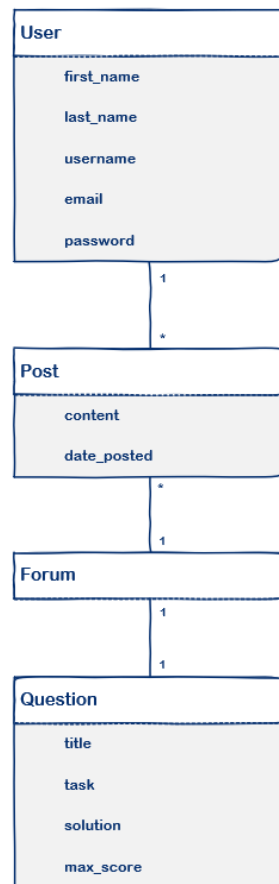


Abbildung 7: Domain Model - Forum

5.3 Tools und Frameworks

Bei der Wahl von Technologien, Tools und Frameworks achtete man besonders darauf, dass die entsprechenden Tools auch in der Zukunft noch supported werden. Dazu wurde abgeklärt, ob und wie gross die Community hinter einzelnen Tools ist und wie gut die Tools dokumentiert sind.

Programmiersprache

Für das Web Development gibt es mehrere Programmiersprachen, die zur Auswahl stehen. Die wohl bekanntesten darunter sind:

- Python (Django / Flask)
- C# (ASP.NET Core)
- Java (Spring)
- JavaScript (Express.js / Node.js / Sails.js)
- PHP (CakePHP / CodeIgniter / Laravel)
- Perl (Catalyst / Mojolicious)
- Ruby (Ruby on Rails)

Gemäss Developer Survey Results²³ von Stack Overflow ist Python (41.7%) etwas beliebter als Java (41.1%). Nur JavaScript (67.8%) ist noch bekannter. In den letzten Jahren ist die Bekanntheit von Python noch weiter gestiegen.

Die Wahl der Programmiersprache fiel schnell auf Python, da bereits etwas KnowHow vorhanden ist. Man geht aber davon aus, dass Python in Zukunft noch an Beliebtheit steigt und man weiterhin damit arbeiten kann.

²³ Developer Survey Results 2019. URL: <https://insights.stackoverflow.com/survey/2019>
[Besucht am 25. Dezember 2019].

Web Framework

Mit Python hat man die Auswahl von verschiedenen Web Frameworks. Gemäss einer Umfrage von JetBrains²⁴ wird die Frage "What web frameworks / libraries do you use in addition to Python?" mit dem Kommentar "Django and Flask continue to be by far the most popular Python web frameworks." zusammengefasst. Laut dieser Umfrage verwenden 43% der Entwickler Django als Web Framework. Flask folgt dicht darauf mit 41%. Das nächst bekannteste Framework ist Tornado mit nur noch 6%.

Bei der Evaluation konzentrierte man sich deshalb hauptsächlich auf Django und Flask. Mit bekannteren Frameworks hat man den Vorteil, dass es einfacher ist, Hilfe und gute Dokumentationen zu finden. Mit weit verbreiteten Technologien kann man auch davon ausgehen, dass diese noch längere Zeit bestehen bleiben.

Flask²⁵ beschreibt sich selber mit den Worten:

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Der grosse Vorteil von Flask ist, dass man als Entwickler relativ frei ist, wie man etwas umsetzen möchte. So kommt Flask standardmässig ohne Database Abstraction Layer, Form Validation oder anderen Tools²⁶. Da bereits Libraries existieren, welche dies erledigen, werden diese nicht nochmals in Flask umgesetzt. Flask folgt dabei der UNIX Philosophie "Do one thing, and do it well". Gerade bei Entwicklern, welche noch nicht so viel Erfahrung mit diesem Framework haben, kann dies aber schnell Verwirrung stiften. Oftmals stehen für ein Problem mehrere Lösungsmöglichkeiten zur Verfügung. Ohne Erfahrung ist es jedoch schwierig, die Beste Lösungsmöglichkeit zu erkennen.

Im Gegensatz zu Flask verfolgt Django den Ansatz eines "batteries included" Framework. Das bedeutet, dass viele Funktionalitäten vom Framework bereits mitgeliefert werden. So ist zum Beispiel ein Object Relational Mapper (ORM), Middlewares, Authentication, ein Admin Interface und eine Templating Engine in Django integriert und kann out of the box verwendet werden²⁷.

²⁴ Python 2019 - The state of Developer Ecosystem in 2019 Infographic. URL: <https://www.jetbrains.com/lp/devecosystem-2019/python/> [Besucht am 25. Dezember 2019].

²⁵ Flask | The Pallets Projects. URL: <https://palletsprojects.com/p/flask/> [Besucht am 25. Oktober 2019].

²⁶ Design Decisions in Flask. URL: <https://flask.palletsprojects.com/en/1.1.x/design/> [Besucht am 25. Oktober 2019].

²⁷ Django at a glance. URL: <https://docs.djangoproject.com/en/3.0/intro/overview/>

Dies kann jedoch auch als Nachteil angesehen werden. Da Django vielfach vorgibt, wie etwas umgesetzt werden muss, ist es sehr monolithisch. Die Entwickler sind nicht nicht mehr ganz so flexibel und müssen sich an Django anpassen²⁸.

Bei der Bachelorarbeit entschied man sich für Django als Web Framework. Besonders die Features, welche von Django angeboten werden, konnten überzeugen. Zudem ist Django das am weitesten verbreitete Framework. Die Dokumentation ist dementsprechend gut und konnte ebenfalls überzeugen.

Datenbank

Laut Deployment Statistics²⁹ von Django Sites verwenden 47.7% aller deployten Sites MySQL als Datenbank. Postgres folgt mit 40.7%. Das nächst grössere Datenbanksystem ist sqlite mit nur noch 7.6%.

MySQL wird zwar etwas öfters verwendet als Postgres. Die Hauptoperationen der Anwendung sind jedoch einfache Schreib- und Lese-Operationen. Da man während den Modulen Datenbanksysteme 1 & 2 Erfahrungen mit Postgres sammeln konnte, weiss man, dass dieses Datenbanksystem die Anforderungen erfüllt, weshalb man sich für Postgres entschied.

Sollte man in Zukunft jedoch ein anderes Datenbanksystem bevorzugen, kann die Migration mit dem Django Management Tool durchgeführt werden³⁰.

[Besucht am 25. Oktober 2019].

²⁸ *Django Advantages and Disadvantages – Why You Should Choose Django?* URL: <https://data-flair.training/blogs/django-advantages-and-disadvantages/> [Besucht am 25. Oktober 2019].

²⁹ *Django Deployment Statistics*. URL: <https://djangosites.org/stats/> [Besucht am 27. Dezember 2019].

³⁰ *Django Database Migration*. URL: <https://docs.djangoproject.com/en/3.0/topics/migrations/> [Besucht am 22. November 2019].

5.4 Deployment

Zusätzlich zur System Overview in Abbildung 2, wird hier das Deployment Diagramm gezeigt. Dieses soll die einzelnen Komponenten aufzeigen.

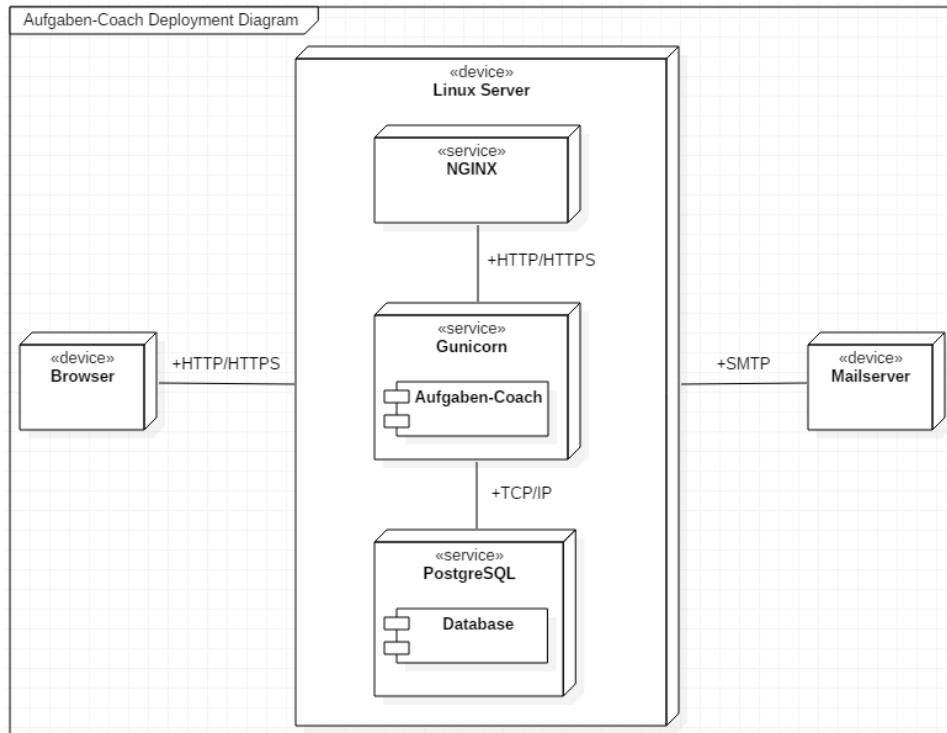


Abbildung 8: Deployment Diagramm

Web Server

Bei der Evaluation eines Web Servers beschränkte man sich auf NGINX und Apache. Wie man auf einer Statistik von w3techs³¹ entnehmen kann, wird in 42% der Web Server Apache eingesetzt. NGINX kommt mit 31.3% etwas weniger zum Einsatz.

Apache kommt öfters zum Einsatz und ist flexibler als NGINX³². Dennoch entschied man sich für NGINX. Bei dieser Entscheidung achtete man hauptsächlich auf die Performance der beiden Web Server, bei welcher NGINX in den Kategorien "Requests per second", "Time per request" und "Transfer rate" durchschnittlich besser abschnitt³³.

³¹ Usage of web servers. URL: https://w3techs.com/technologies/overview/web_server [Besucht am 05. Januar 2020].

³² NGINX vs. Apache (Pro/Con Review, Uses and Hosting for Each). URL: <https://www.hostingadvice.com/how-to/nginx-vs-apache/> [Besucht am 05. Januar 2020].

³³ Apache vs nginx - performance comparison. URL: <https://theorganicagency.com/blog/apache-vs-nginx-performance-comparison/> [Besucht am 05. Januar 2020].

Web Server Gateway Interface (WSGI)

Während der Entwicklungsphase wurde der Development Server, welcher direkt mit dem Django Framework mitgeliefert wird, verwendet. In der produktiven Umgebung reicht dies jedoch nicht mehr aus, weshalb ein WSGI eingesetzt wird. Dieser beschreibt, wie der Web Server mit einer Web Anwendung kommuniziert. Anfragen an den Web Server werden zur Web Anwendung geleitet. Sobald der Request bearbeitet wurde, wird die Response an den Web Server zurück geschickt³⁴.

Docker

Bei der Entwicklung hat man sich dafür entschieden, dass jeder Service in einem eigenen Docker Container läuft. Jeder dieser Container hat dabei seine eigene virtuelle Umgebung. Somit laufen die Container immer gleich, egal auf welchem System sie deployed werden.

Containerisierung bietet mehrere Vorteile gegenüber der Hardware Virtualisierung. Bei der Hardware Virtualisierung wird ein komplett neues Guest OS auf dem laufenden Host hochgefahren. Damit der Guest mit dem Host kommunizieren kann, wird ein Hypervisor dazwischen geschaltet. Container können jedoch direkt auf den vom Host-System verwendeten Kernel zugreifen. Somit wird kein Guest OS und Hypervisor mehr benötigt, was viel Ressourcen spart.

Für die Entwickler hat es den Vorteil, dass praktisch gleich auf der Entwicklungsumgebung programmiert werden kann. Dabei muss auch nicht jeder Entwickler zuerst die gesamte Umgebung bei sich aufbauen, sondern kann mit einem einfachen Befehl die ganze Umgebung hoch- oder herunterfahren.

Für jeden der einzelnen Services in Abbildung 8 wird ein eigener Docker Container erstellt. Requests werden vom NGINX-Service, auf welchem der Web Server läuft, entgegen genommen und an den web-Service weitergeleitet, auf welchem der Aufgaben-Coach läuft. Dieser hat wiederum eine Verbindung zum db-Service, falls Daten aus der Datenbank geladen werden müssen. Somit ist es einem Benutzer auch nicht möglich, direkt mit der Datenbank eine Verbindung aufzubauen.

³⁴ *What Is WSGI?* URL: <https://www.liquidweb.com/kb/what-is-wsgi/> [Besucht am 05. Januar 2020].

6 Design

6.1 UI Design

Bevor mit der Entwicklung des Frontends begonnen werden kann, muss geplant werden, wie dieses aussehen soll. Um ein möglichst gutes Ergebnis zu erzielen, ist es wichtig sich folgende Fragen zu stellen³⁵:

- Welche Anforderungen werden an die Webseite gestellt?
- Welche Benutzergruppen werden die Applikation verwenden?
- Was sind die Bedürfnisse der einzelnen Gruppen?

6.1.1 UI Design Anforderungen

Durch das Beantworten der einzelnen Fragen kann schnell herausgefunden werden, welche Anforderungen an das Frontend gestellt werden. Da der Aufgaben-Coach von mehreren Benutzergruppen mit unterschiedlichen Anforderungen verwendet wird, entschied man sich, das User Interface möglichst simpel und übersichtlich zu halten. Folgende Anforderungen werden an das User Interface gestellt.

Usability

Schüler sind die Hauptnutzer dieser Anwendung. Aus diesem Grund möchte man die Anwendung so gestalten, dass Schüler auch gerne Zeit auf der Webseite verbringen. Laut einer Statistik zur Handynutzung von Primarschülern³⁶ nutzen 34% der 6-/7-Jährigen regelmässig ein Handy. Die Nutzung steigt jedoch mit dem Alter weiter an. Bei den 12-/13-Jährigen nutzen 77% regelmässig ein Handy. Es kann davon ausgegangen werden, dass die Handynutzung mit fortschreitendem Alter noch weiter ansteigt.

Um die Bedürfnisse der Schüler möglichst gut abzudecken, soll der für sie gedachte Teil der Anwendung auch auf Handies erreichbar und bedienbar sein.

³⁵ 100+ Questions You Must Ask When Developing A Website. URL: <https://marketingland.com/100-questions-you-must-ask-when-developing-web-site-86342> [Besucht am 20. Oktober 2019].

³⁶ Jugendliche und Medien - Fakten und Zahlen. URL: <https://www.jugendundmedien.ch/digitale-medien/fakten-zahlen.html> [Besucht am 20. Oktober 2019].

Struktur

Kommen neue Benutzer auf die Webseite, bleibt in der Regel wenig Zeit, um einen positiven ersten Eindruck zu hinterlassen. Gemäss der Statistik des Statistic Brain Research Institute³⁷ liegt die Aufmerksamkeitsspanne bei Nutzern bei knapp 8.25 Sekunden. Es bleibt also nicht viel Zeit, um einen positiven Eindruck zu hinterlassen.

Um bei den Benutzern einen positiven Eindruck zu hinterlassen wurde besonders die Struktur der Webseite angepasst. Die Webseite soll klar und übersichtlich dargestellt werden. Es soll genügend Information auf der Startseite vorhanden sein, aber trotzdem nicht überladen wirken.

Navigation

Die Navigation der Webseite soll ebenfalls einfach und simpel gehalten werden. Trotzdem müssen die Hauptseiten über die Navigation erreichbar sein.

Orientierung

Die Benutzer der Webseite sollen sich gut zurechtfinden und jederzeit wissen, wo sie sich genau befinden. Benutzer sollen sich nie auf einer Seite befinden, bei welcher sie nicht genau wissen, wie sie dahin gelangt sind und wie sie wieder zurück kommen.

Kontrast

Die einzelnen Elemente sollen gut erkennbar sein, so dass man sie auf Anhieb findet. Zudem sollen die Elemente über die ganze Webseite konstant sein. Buttons der selben Kategorie sollen alle das gleiche Design und die selbe Farbe haben.

³⁷ *Attention Span Statistics*. URL: <https://www.statisticbrain.com/attention-span-statistics/> [Besucht am 06. Januar 2020].

6.1.2 UI Design Entscheidungen

Basierend auf den oben aufgelisteten UI-Design Anforderungen, wurden folgende Entscheidungen getroffen:

Usability

Die Webseite soll niemals mit Informationen überladen sein. Pro Webseite sollen nur diese Informationen angezeigt werden, welche vom Benutzer auch erwartet werden. Zudem soll auf lange Texte verzichtet werden³⁸. Auf dem Desktop sind diese oft gut lesbar. Wird die Webseite aber über Mobile Phones aufgerufen, geht die Übersicht schnell verloren.

Um die Übersicht besser gestalten zu können, sollen die einzelnen Seiten nur die für sie gedachte Aufgabe erledigen.

Struktur

Um die Struktur benutzerfreundlich zu gestalten, muss ein Kompromiss zwischen Übersicht und Funktionalität gemacht werden. Einzelne Webseiten sollen nur die für sie gedachten Aufgaben erledigen. Die Nutzer sollen die Webseite aber dennoch so bedienen, dass nicht ständig zwischen den einzelnen Webseiten gewechselt werden muss.

Navigation

Um die Navigation auf der Webseite zu erleichtern, wird die Anwendung in fünf Teilbereiche aufgeteilt. Jeder dieser Teilbereiche ist über die Navigationsleiste erreichbar.

Orientierung

Auf der Webseite sollen Breadcrumbs verwendet werden, um dem Benutzer eine Orientierungshilfe zu geben. Anhand dieser Breadcrumbs ist auf jeder Webseite ersichtlich, wo man sich genau befindet. Diese dienen auch gleich als Überschrift für die einzelnen Webseiten.

Kontrast

Knallige Farben sollen so gut wie möglich vermieden werden. Wichtige Elemente sollen aber dennoch durch Farben hervorgehoben werden. Es wurde jedoch darauf geachtet, dass diese nicht zu fest hervorstechen.

³⁸ *MOBILE FIRST – NUR EIN TREND ODER SCHON BALD DIE ZUKUNFT?* URL: <https://www.cubetech.ch/mobile-first-nur-ein-trend-oder-schon-bald-die-zukunft/> [Besucht am 21. Oktober 2019].

6.2 Mockups

Anhand dieser Anforderungen wurden Mockups erstellt. Mockups bieten mehrere Vorteile beim Entwerfen einer Webseite. So kann zum Beispiel früh erkannt werden, ob wichtige Punkte vergessen oder vernachlässigt wurden. Mit Mockups können auch gleich die ersten Tests durchgeführt werden. Es kann geprüft werden, ob die geforderten funktionalen Anforderungen erfüllt werden und Usability-Tests können durchgeführt werden³⁹.

Nachfolgend werden einige Beispiele gezeigt, wie die getroffenen Entscheidungen in die Entwicklung der Mockups eingeflossen sind. Die vollständigen Mockups befinden sich im Anhang.

Usability

Auf dem Dashboard werden die freigeschalteten Fächer und der Wochenplan eines Schülers aufgelistet. Da dies die erste Seite ist, welche ein neu eingeloggter Benutzer sieht, wurde besonders auf Übersichtlichkeit geachtet.

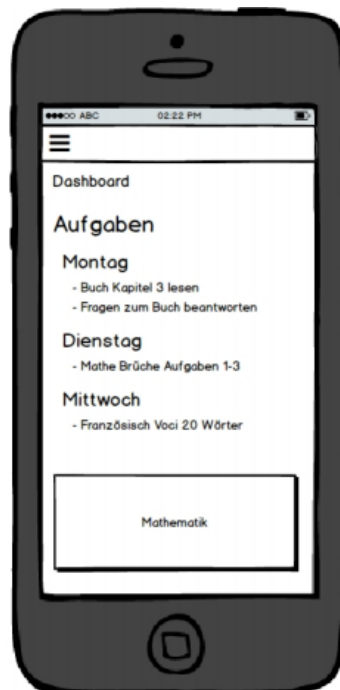


Abbildung 9: Mockup Smartphone Dashboard

³⁹ Mockups. URL: <https://www.igual.ch/de/internet-glossar/mockups> [Besucht am 21. Oktober 2019].

Struktur

Die Aufgabe dieser Seite ist es, neue Aufgaben und Fragen zu definieren. Mit dem Risiko die Seite zu überladen, dem Benutzer jedoch eine bessere Übersicht zu bieten, wurde ein Kompromiss eingegangen und entschieden, dass auch die dazugehörigen Hilfestellungen erfasst werden können.

The mockup shows a web browser window titled 'Aufgaben-Coaching' with the URL 'https://aufgaben-coaching.ch'. The navigation bar at the top contains buttons for 'Dashboard', 'Statistiken', 'Klassenverwaltung', 'Admin', and 'Logout'. The main content area is titled 'Mathematik > Brüche > Aufgabe erfassen'. It is divided into two columns. The left column, 'Aufgabe erfassen', contains a dropdown menu for 'Mathematik' (with 'Physik' and 'Deutsch' as options), a dropdown menu for 'Brüche' (with 'Pythagoras' as an option), a text input for 'Titel', a large text area for 'Aufgabenstellung erfassen', a text input for 'Lösung', a checkbox for 'Aufgabe für andere Lehrpersonen freigeben', and a 'Speichern' button. The right column, 'Hilfestellungen erfassen', contains a section for 'Hilfestellung 1' with a 'Hilfertext' input, a 'Zwischenresultat' input, and a 'Hinzufügen' button. There is a close button (X) next to the 'Hilfestellung 1' section. At the bottom right of the form is an 'Abbrechen' button.

Abbildung 10: Mockup Aufgabe erfassen Desktop

Navigation

In der Navigationsbar sind die fünf definierten Bereiche ersichtlich. Dashboard stellt den Schulstoff Bereich dar, Statistiken die Statistiken, Klassenverwaltung die Klassenverwaltung, Admin das Adminpanel und Logout/Login die User.

Damit die Navigationsbar auch in der Mobile Version gut benutzbar ist, wird ein Hamburger Menu erstellt. Klickt ein Benutzer darauf, klappt sich die Navigationsbar auf.

The mockup shows a horizontal navigation bar with five buttons: 'Dashboard', 'Statistiken', 'Klassenverwaltung', 'Admin', and 'Logout'.

Abbildung 11: Mockup Navigationsbar Desktop

Orientierung

Durch die Breadcrumbs wird hierarchisch dargestellt, wo sich ein Benutzer aktuell auf der Webseite befindet. Mit einem Klick auf die übergeordneten Bereiche navigiert man auch auf diese.

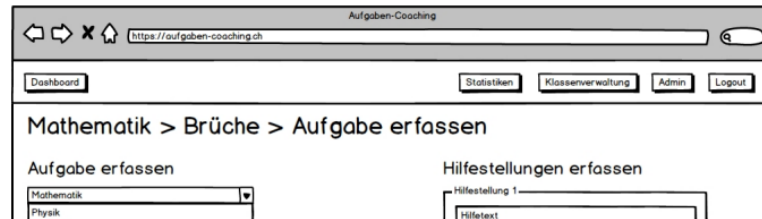


Abbildung 12: Mockup Breadcrumbs Desktop

Kontrast

Die Mockups wurden in schwarz und weiss erstellt, weshalb man hierfür kein Beispiel machen kann. Die Entscheidungen wurden dennoch berücksichtigt und und kann in der effektiven Webseite gesehen werden.

6.3 Frontend Framework

Bei der Wahl des Frontend Frameworks wurde zwischen Semantic-UI und Bootstrap entschieden. Ein Vorteil von Semantic-UI gegenüber Bootstrap ist, dass die Ladezeiten teilweise deutlich kürzer sind als bei Bootstrap. Dies hat den Grund, da nur die Komponenten geladen werden, welche auch wirklich benötigt werden. Bootstrap besitzt jedoch auch gewisse Vorteile, so wird zum Beispiel Responsive Web Design explizit unterstützt. Ausserdem ist Bootstrap das am meisten verbreitete Frontend Framework und bietet daher umfassende Dokumentationen und Beispiele an. Da die Unterstützung von Responsive Web Design eine wichtige Anforderung Web Anwendung ist, hat man sich für Bootstrap entschieden.

7 Implementation und Tests

7.1 Django Architektur

Bei der Implementation der Anwendung befolgte man die vom Django Framework vorgegebene Architektur. Für die einzelnen Apps entschied man sich jedoch dafür, diese in einem separaten "apps" Verzeichnis zu speichern, um das Home-Verzeichnis übersichtlicher zu halten.

Django ist nach dem Model Template Views (MTV) Pattern aufgebaut, was eine leichte Abänderung des Model View Controller (MVC) Pattern ist. Beim MVC Pattern dient das Model als Vermittler zwischen dem Website Interface und der Datenbank. Die View ist das eigentliche User Interface und enthält zum Beispiel das HTML oder CSS der Website. Der Controller ist die Hauptkomponente dieses Patterns und mappt anhand der vom Benutzer gestellten Anfrage das entsprechende Model mit der entsprechenden View⁴⁰.

Django jedoch basiert auf dem MTV Pattern. Es kann ein wenig verwirrend sein, denn die einzelnen Rollen im MTV Pattern haben eine andere Bezeichnung als die im MVC Pattern. Der Controller zum Beispiel wird als View bezeichnet und ist nun eigentlich die Templating-Engine⁴¹. Wird im Browser eine Uniform Resource Locator (URL) aufgerufen, entscheidet Django anhand des URL Patterns, an welche View die Anfrage gesendet wird. Diese View generiert dann schlussendlich die Webseite mit Hilfe des Templates. Im Django Architektur Diagramm⁴² können diese Abläufe erkannt werden.

⁴⁰ *Django Architecture – 3 Major Components of MVC Pattern*. URL: <https://data-flair.training/blogs/django-architecture/> [Besucht am 3. Januar 2020].

⁴¹ *Django-Architektur*. URL: <https://wagtailtagging.readthedocs.io/en/latest/architektur.html> [Besucht am 4. Januar 2020].

⁴² *Django-Architektur Schnitt*. URL: https://wagtailtagging.readthedocs.io/en/latest/_images/mtv_schnitt.png [Besucht am 4. Januar 2020].

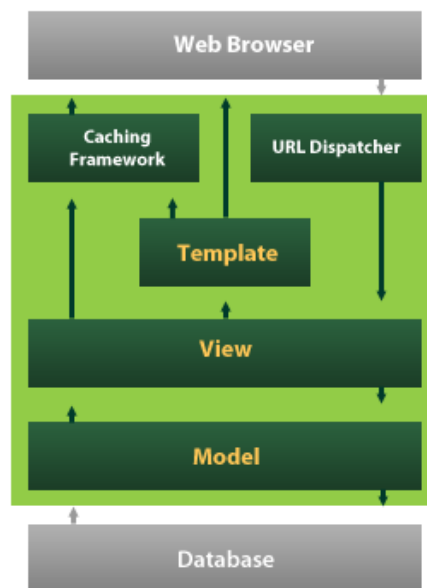


Abbildung 13: Django Architektur Diagramm

Model

In den Models wird das Datenbankmodell mit den benötigten Attributen definiert⁴³. Django stellt auch gleich einen ORM bereit, mit welchem die Tabellen in der Datenbank erstellt werden können. Dies bietet gleich mehrere Vorteile. Zum einen nimmt der ORM viel Arbeit ab, da man keine eigenen Queries schreiben muss. Zum anderen ist er sehr hilfreich, wenn man Migrationen an der Datenstruktur durchführen möchte. Der ORM erkennt, was an der Datenbank geändert wurde und wie die Daten migriert werden müssen⁴⁴. Als Entwickler können so auch ohne ausgeprägte Datenbank Kenntnisse Migrationen durchgeführt werden.

Templates

Mit Templates können dynamische HTML Seiten generiert werden. Die Templates bestehen aus dem statischen HTML Code, welcher mit dynamischen Inhalten ergänzt werden kann. Dazu verwendet Django die Django Template Language (DTL)⁴⁵.

⁴³ *Models | Django documentation*. URL: <https://docs.djangoproject.com/en/3.0/topics/db/models/> [Besucht am 3. Januar 2020].

⁴⁴ *django-admin and manage.py | Django documentation*. URL: <https://docs.djangoproject.com/en/3.0/ref/django-admin/> [Besucht am 3. Januar 2020].

⁴⁵ *Templates | Django documentation*. URL: <https://docs.djangoproject.com/en/3.0/topics/templates/> [Besucht am 3. Januar 2020].

Views

Eine View ist nichts anderes als eine Python Funktion, welche einen Request entgegen nimmt und eine Response zurück schickt. Die Response kann alles mögliche enthalten, von einfachem Hypertext Markup Language (HTML) über 404 Fehlerseiten bis hin zu Redirects. Die View an sich enthält die Logik, welche nötig ist, um den Request entgegennehmen zu können und eine Response zu senden⁴⁶.

Middlewares

Hinzu kommen jedoch noch die Django Middlewares, welche Requests und Responses anpassen können. Jede dieser Middlewares ist für bestimmte Funktionen verantwortlich. Die "AuthenticationMiddleware" wird zum Beispiel verwendet, um die Benutzer einer Session zuordnen zu können. Es werden auch mehrere Security Middlewares angeboten, welche praktisch "out-of-the-box" verwendet werden können. Diese können die Sicherheit der Webseite, zum Beispiel die Integration von HTTP Strict Transport Security (HSTS) oder Cross-Site Scripting (XSS) Schutz, erhöhen⁴⁷.

7.2 Testing

Es gibt mehrere Arten von Tests, welche durchgeführt wurden, um die Qualität dieser Arbeit zu gewährleisten.

Test Framework

In der Django Dokumentation wird erwähnt, dass das Framework "unittest" das beliebteste Test Framework für Django Applikationen ist⁴⁸. Dies widerspiegelt auch die Dokumentation von Django. Unzählige Beispiele zu allen möglichen Arten von Tests sind dokumentiert und erklärt. Um also davon zu profitieren, wurde entschieden das Framework "unittest" zu verwenden.

7.2.1 Unit Tests

Anhand der Unittests werden die erstellten Models getestet. Mit diesen Tests wird sichergestellt, dass die Models korrekt funktionieren. Es kann zum Beispiel getestet werden, ob die Objekte als String abgespeichert werden oder ob duplizierte Objekte abgelegt werden können. Wenn in der Datenbank bereits ein Subject mit dem Titel "Mathematik" existiert, darf kein weiteres Subject mit dem selben Namen erstellt werden können.

⁴⁶ *Writing views | Django documentation*. URL: <https://docs.djangoproject.com/en/3.0/topics/http/views/> [Besucht am 3. Januar 2020].

⁴⁷ *Middleware | Django documentation*. URL: <https://docs.djangoproject.com/en/3.0/topics/http/middleware/> [Besucht am 3. Januar 2020].

⁴⁸ *Testing in Django*. URL: <https://docs.djangoproject.com/en/3.0/topics/testing/> [Besucht am 2. Dezember 2019].

7.2.2 Integration Tests

Mit Integration Tests kann getestet werden, ob die Applikation korrekt auf eingehende Requests reagiert. Diese Anwendung wird von mehreren Benutzern mit unterschiedlichen Rechten verwendet. Beim Erstellen der Tests wurde besonders viel Wert darauf gelegt, dass geprüft wird, dass jeder Benutzer nur Zugriff auf die erlaubten Webseiten hat. Aus Erfahrung weiss man, dass Schüler sehr experimentierfreudig sind und sicherlich versuchen werden, die URL des Browsers anzupassen. Es darf jedoch nicht vorkommen, dass ein Schüler Zugriff auf Statistiken hat oder die Punktezahl von Aufgaben anpassen kann.

7.2.3 Manuelle Tests

Manuelle Tests wurden da eingesetzt, wo es sehr schwierig und zeitaufwändig gewesen wäre, Tests zu schreiben. Vor jedem Merge in den Master wurde geprüft, ob diese Funktionen noch funktionieren. Ein Beispiel für den Fall von manuellen Tests ist, ob das Filtering und Pagination richtig funktioniert. Zu Beginn hatte man da das Problem, dass das Filtern zwar funktionierte, aber das Pagination basierte auf dem gesamten Queryset. Die Anzahl der Seiten änderte sich also nicht, auch wenn der Filter kein einziges Element gefunden hat. Ein weiterer Fall für manuelle Tests sind die Drag&Drop Seiten. Bei diesen wurde geprüft, ob man das Item einem Feld zuweisen kann und ob es anschliessend richtig in der Datenbank gespeichert wird.

7.2.4 Usability Tests

Nachdem man die Mockups erstellt hat, konnten damit die Usability Tests durchgeführt werden. Ziel dieser Tests ist es, zu prüfen, ob Benutzer das Design der Webseite verstehen und wissen, wo man was machen kann.

Jeder der 3 Probanden hat die Tests für alle drei Benutzerrollen durchgeführt. Bei jeder Durchführung sollten sie jedoch eine andere Rolle und andere Aufgaben übernehmen. So konnte verhindert werden, dass die Tests verfälscht werden. Auch wenn einer der Probanden die Webseite gesehen hat, musste er jedes Mal andere Aufgaben lösen. Zudem löste jeder Proband die Tests in einer anderen Reihenfolge. Somit hat man zumindest für den ersten Test eine Person, welcher die Webseite noch nie gesehen hat. In der Tabelle 3 ist ein Beispiel, wie die Durchführung der Tests gemacht wurde.

Beim Durchführen der Usability Tests wurden die Probanden gebeten, beim Lösen der Aufgaben laut zu denken. Während der Durchführung wurden anhand der Gedanken und Anmerkungen der Probanden Notizen gemacht. Da die verschiedenen Benutzergruppen die Applikation später auf unterschiedlichen Geräten verwenden werden, wurde dies in die Tests miteinbezogen. Es wird davon ausgegangen, dass die Schüler die Webapplikation später meistens per Tablet besuchen werden. Bei den Lehrern und den

Proband	Durchführung 1	Durchführung 2	Durchführung 3
Proband 1	Schüler	Lehrer	Administrator
Proband 2	Lehrer	Administrator	Schüler
Proband 3	Administrator	Lehrer	Schüler

Tabelle 3: Usability Test Rollen

Administratoren wird davon ausgegangen, dass diese die Web Anwendung hauptsächlich per Notebook Desktop-PC verwenden.

Nachfolgend sind die Aufgaben und Fragen aufgelistet, welche die Probanden pro Rolle lösen mussten:

Schüler

- Welche Aufgabe/Aufgaben muss/müssen bis am Dienstag erledigt werden?
- Navigiere zum ersten Theorieteil des Themas Brüche.
- Löse die Aufgabe 1 des Themas Brüche.
- Navigiere ins Forum der Aufgabe "Aufgabe 1" des Themas Brüche.
- Wie kann eine Hilfestellung bei einer Aufgabe angefordert werden?
- Wo befindet sich das Quiz 1 des Themas Brüche?

Lehrer

- Wie sieht der Wochenplan der Klasse "S2" aus?
- Wie kann der Wochenplan angepasst werden?
- Navigiere ins aufgabenspezifische Forum der Aufgabe "Aufgabe 1" des Themas Brüche.
- Erfasse eine neue Aufgabe für das Thema Brüche.
- Wie und wo kann eine bereits erstellte Aufgabe für andere Lehrpersonen freigeschalten werden?
- Lösche das Quiz "Quiz Basics" des Themas Brüche.
- Erstelle ein neues Quiz für das Thema Brüche mit zwei bereits erstellten Fragen.
- Erfasse eine neue Frage für ein Quiz.
- Schaue dir die Statistiken an.

- Wie gut hat der Schüler "Philipp Muster" das Quiz "Quiz 1" des Themas Brüche gelöst?
- Bei welcher Frage des Quiz 1 des Themas Brüche hatte der Schüler "Philipp Muster" die grössten Probleme?

Admin

- Erstelle eine neue Klasse.
- Erstelle einen neuen Schüler oder Lehrer.
- Lösche einen Schüler, Lehrer oder eine Klasse.
- Wie kann ein Lehrer oder Schüler einer Klasse zugewiesen werden?

Feedback

Anhand dem Verhalten und dem persönlichen Feedback der Probanden konnten folgende Probleme erkannt werden:

Schüler

- Es fiel den Probanden schwer, die Quizze zu finden. Es wurde erwartet, dass die Quizze zusammen mit den Theorieteilern und Aufgaben aufgelistet werden.

Lehrer

- Beim Anpassen des Wochenplans einer Klasse, wurde angemerkt, dass nicht sofort klar sei, dass man die Klasse, das Fach und das Thema in einem Drop-Down-Menü auswählen muss, damit die gesuchten Aufgaben dargestellt werden. Es wurde erwartet, dass dies einfacher funktionieren würde.
- Die Funktion eine erstellte Aufgabe mit anderen Lehrern zu teilen sei schwer zu finden.

Beim Implementieren der Webapplikation wurden die aus den Usability Tests gewonnenen Kenntnisse miteinbezogen und verbessert. Ausserdem haben sich die Anforderungen an die Arbeit verändert, weshalb die Quizze kein Bestandteil der Applikation mehr sind. Folgende Anpassungen und Entscheidungen wurden gemacht:

- Um das Verwalten einer Klasse zu vereinfachen, wurde entschieden, dass im Gegensatz zu der ursprünglichen Version sich die entsprechende Seite dynamisch erweitert. Zu Beginn sollen nur die dem Lehrer zugewiesenen Klassen aufgelistet werden.

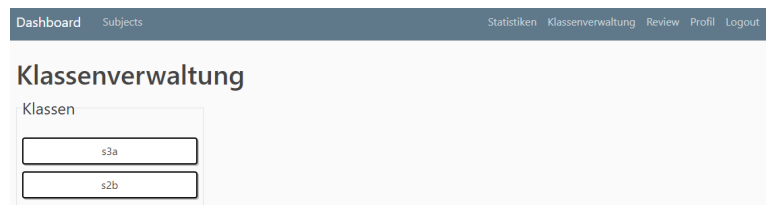


Abbildung 14: Anpassung Klassenverwaltung

Nachdem eine Klasse gewählt wurde, werden die klassenspezifischen Informationen geladen und ebenfalls dargestellt.

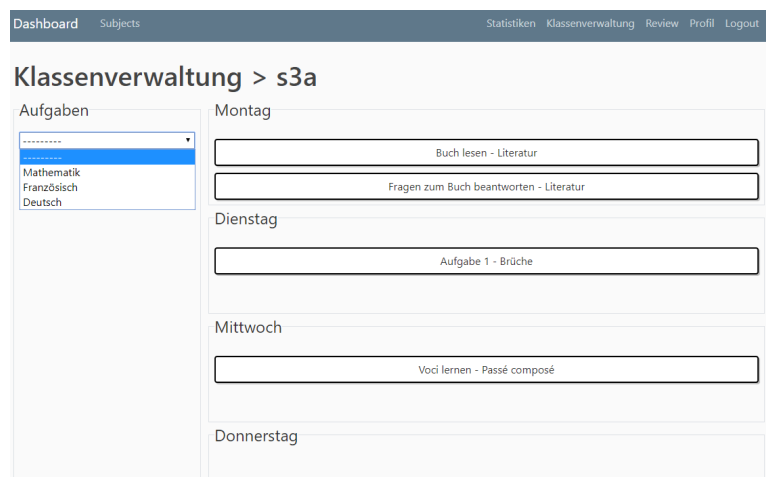


Abbildung 15: Informationen einer Klasse

- Die Funktion um eigene Aufgaben mit anderen Lehrpersonen zu teilen wurde entfernt. Standardmässig sind nun alle erstellten Aufgaben für alle Lehrpersonen ersichtlich.

7.2.5 Performance Testing

Um sicherzustellen, dass die nichtfunktionale Anforderung Response Time erfüllt wird, wurden Performance Tests durchgeführt. Dafür wurde das Tool Gatling⁴⁹ verwendet. Um so realitätsgetreu und dementsprechend auch sinnvoll wie möglich zu testen, wurden 300 Schüler, 20 Lehrer und 1 Administrator simuliert, welche die Applikation zeitgleich verwendeten.

Auswertung

Erste Durchführung:

Bei der ersten Durchführung wurde festgestellt, dass noch einige Probleme vorliegen. Wie in der Abbildung 16 zu sehen ist, benötigte die Applikation bei mehr als einem Drittel der eingegangenen Requests mehr als eine Sekunde um eine Response zu schicken. Bei knapp einem Drittel konnte sogar gar keine Response zurückgesendet werden.

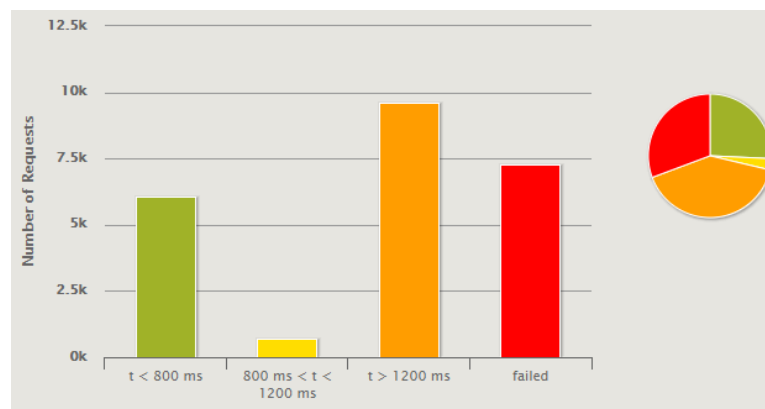


Abbildung 16: Performance Test Übersicht vorher

Bei der Analyse der gefailten Requests konnte festgestellt werden, dass bei jedem Request die Bootstrap Files geladen wurden. Diese Files befinden sich jedoch nicht in den statischen Files des Web Servers, sondern wurden jedes Mal direkt von Bootstrap geladen.

Als Massnahme wurden die Bootstrap Files heruntergeladen und in den statischen Files des Web Servers gespeichert. Somit können diese Files direkt vom Web Server bereitgestellt werden.

Bei der zweiten Durchführung wurden über einen Zeitraum von 1 Minute und 51 Sekunden insgesamt 38'909 Requests abgesetzt. In der Abbildung 17 ist ersichtlich, wie die

⁴⁹Load test as code. URL: <https://gatling.io/> [Besucht am 28. Dezember 2019].

Requests über den erwähnten Zeitverlauf verteilt gesendet wurden.

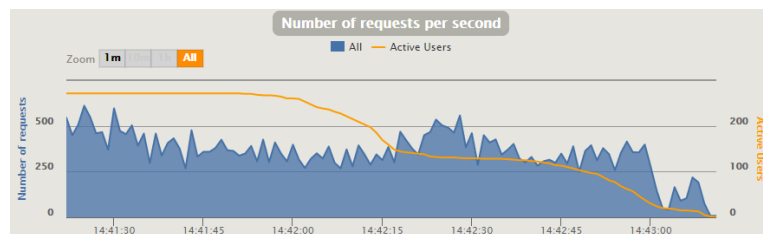


Abbildung 17: Anfragen pro Sekunde

Wie in der Abbildung 18 ersichtlich ist, benötigte die Applikation bei etwa 5% der eingegangenen Anfragen etwas länger als eine Sekunde, um die Response zu schicken. Im Vergleich zur ersten Durchführung ist dies eine enorme Verbesserung.

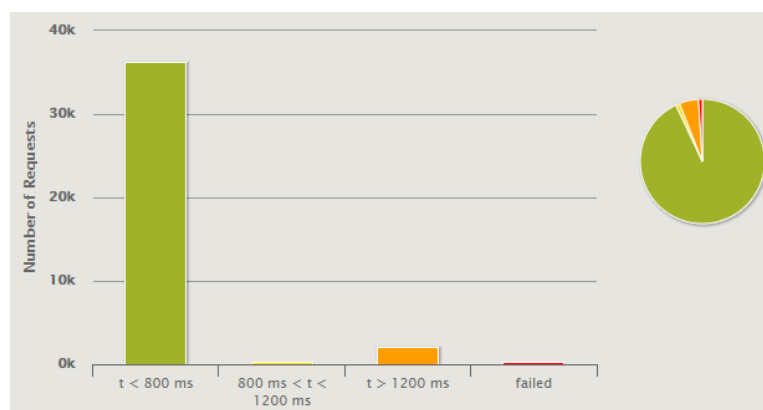


Abbildung 18: Performance Test Übersicht nachher

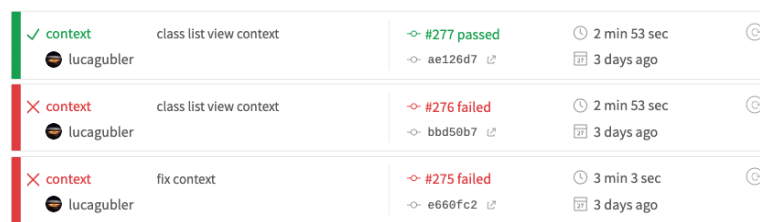
7.3 Continuous Integration

Continuous Integration (CI) beschreibt das Prinzip, wenn mehrere Entwickler an einem Projekt arbeiten und ihre Änderungen am Code regelmässig in den Master Branch mergen. Dies bietet gleich mehrere Vorteile für die Entwickler. Zum einen werden kleinere Änderungen gemerged. Kleine Änderungen sind einfacher zu handhaben, als wenn grosse Änderungen im Code vorkommen. Bei jedem Merge mit dem master werden die Tests automatisch gestartet. Somit erhalten die Entwickler sehr schnell Feedback, falls Tests fehlgeschlagen sind. Da vorallem kleinere Änderungen gemerged werden, sind die Ursachen der fehlgeschlagenen Tests meist schnell gefunden. Somit lässt sich auch die Mean Time To Repair (MTTR) verkürzen.

Anbieter

Bei dieser Arbeit wurde Travis als CI Anbieter ausgewählt. Bei jeder Änderung am Code auf Github wird Travis mittels Web-Hook benachrichtigt. Der Code wird dann heruntergeladen und gestartet. Die genaue Anweisung, was gestartet und geprüft werden soll, befindet sich im ".travis.yml" File. In diesem Fall werden zuerst die drei Docker Container gestartet, dann wird die Code Qualität geprüft und zum Schluss werden die Tests durchgeführt. Somit sehen die Entwickler schnell, ob die Tests erfolgreich durchgelaufen sind oder nicht.

Zusätzlich wurde das Git Repository so konfiguriert, dass ein Pull Request nur in den Master Branch gemerged werden kann, wenn alle Tests erfolgreich durchgelaufen sind und wenn ein Code Review durchgeführt wurde. Somit kann das Risiko vermindert werden, dass nicht lauffähiger Code in den Master gemerged wird.



✓ context	class list view context	→ #277 passed	⌚ 2 min 53 sec	🔗
👤 lucagubler		→ ae126d7	📅 3 days ago	
✗ context	class list view context	→ #276 failed	⌚ 2 min 53 sec	🔗
👤 lucagubler		→ bbd50b7	📅 3 days ago	
✗ context	fix context	→ #275 failed	⌚ 3 min 3 sec	🔗
👤 lucagubler		→ e660fc2	📅 3 days ago	

Abbildung 19: Failed Travis Build

In der Abbildung 19 kann man ein Beispiel eines fehlgeschlagenen Builds erkennen. Grund dafür war ein fehlende leere Zeile an Ende eines Python Files. Nach dem Einfügen einer leeren Zeile konnte Travis erfolgreich ausgeführt werden.

Zu Beginn überlegte man sich, ob man Travis oder Jenkins als CI Tool auswählt. Jenkins ist mit über 600 Plugins zwar sehr flexibel und weiter verbreitet als Travis⁵⁰. Da man aber nur die Basics braucht, ist man nicht auf diese Features angewiesen. Der Hauptgrund, warum man sich für Travis entschied, ist, weil es für öffentliche Git Repositories gratis verwendet werden kann. Jenkins ist zwar auch frei erhältlich, doch man hätte einen eigenen Build Server konfigurieren müssen.

⁵⁰ Jenkins vs Travis-CI: What is the difference? URL: <https://www.guru99.com/jenkins-vs-travis.html> [Besucht am 25. Oktober 2019].

7.4 Code Statistiken

Die Werte aus Tabelle 4 wurden mit dem Tool cloc⁵¹ (Count Lines of Code) generiert und zeigen, wie viele Zeilen Code im Rahmen dieser Bachelorarbeit geschrieben wurden. In die Statistik aufgenommen wurde nur selbst geschriebener Code. Alles was automatisch generiert wurde, wurde ausgeschlossen. Der Code der geschriebenen Tests ist in diesen Zahlen inbegriffen.

Sprache	Blank	Kommentar	Code
Python	1456	108	5501
HTML	256	0	1087
JavaScript	12	51	168
Summe	1727	159	6761

Tabelle 4: Code-Statistik

⁵¹ *Count Lines of Code*. URL: <https://github.com/AlDanial/cloc> [Besucht am 05. Januar 2020].

7.5 Ergebnisse

Nach Abschluss der Bachelorarbeit konnte ein Grossteil der gewünschten Funktionalität implementiert werden. Das Hauptziel, welches wir uns gesetzt haben, ist die Entwicklung einer Anwendung, welche die Flipped Classroom Unterrichtsmethode ermöglicht. Aufgaben-Coach erfüllt diesen Punkt und kann so, wie es jetzt existiert verwendet werden. Wird die Idee als Startup weiter verfolgt, gibt es einige Punkte, welche verbessert oder neue Features, welche noch implementiert werden können.

7.5.1 Resultate

In den folgenden Abschnitten sind die Funktionalitäten der einzelnen Benutzergruppen aufgelistet.

Schüler

- Einzelne Lektionen anschauen
- Aufgaben lösen
- Gelöste Aufgaben verbessern
- Fragen im Forum stellen

Lehrer

- Übungen erfassen und bearbeiten
 - Dynamisches Hinzufügen von Fragen und Hilfestellungen.
 - Punktevergabe pro Frage.
- Eigene Klassen verwalten
 - Fächer einer Klasse zuweisen
 - Aufgaben dem Wochenplan einer Klasse zuweisen
- Gelöste Aufgaben korrigieren
 - Filtern von allen gelösten Aufgaben seiner Klasse oder einzelner Schüler
 - Pagination Ansicht der gefilterten Aufgaben
 - Einzelne Aufgaben akzeptieren oder zurückweisen
 - Feedback für einzelne Aufgaben erfassen
 - Mail an Schüler mit dem Feedback senden
- Statistiken einsehen

Admin

Im Prinzip hat der Admin die Rechte, um alles auf der Website zu tun. Nachfolgend sind nur die Features, welche sich direkt an den Admin richten:

- Schüler und Lehrer erfassen und bearbeiten
- Klassen erstellen und User diesen Klassen zuweisen

7.5.2 Verbesserung

Die Anwendung war viel komplexer als zu Beginn erwartet wurde. Bei gewissen Features mussten Kompromisse eingegangen werden um im Zeitplan bleiben zu können. Folgende Punkte müssen noch verbessert werden:

- Im Moment wurde beim Frontend hauptsächlich Bootstrap eingesetzt. Vereinzelt wurde auch AJAX oder jQuery eingesetzt, um die Website dynamischer zu gestalten. Für die Zukunft muss man evaluieren, ob man nicht besser ein anderes Frontend Framework, wie zum Beispiel React oder Angular, verwendet.
- Lehrpersonen können Aufgaben einem Wochenplan zuweisen. Dieses Feature wurde aber nur sehr grob implementiert und kann noch verbessert werden. Der Lehrer sollte den Wochenplan für mehrere Wochen vorgeben können. Wird eine Aufgabe dem Wochenplan zugewiesen, soll eine Deadline hinterlegt werden. Nach Ablauf dieser Deadline sollen keine weiteren Abgaben gemacht werden können.
- Das Filtern der Aufgaben kann noch verbessert werden. Lehrer sollen auch nach Klassen oder nach archivierten Aufgaben filtern können.
- Im Moment sind alle Aufgaben global. Erstellt ein Lehrer eine Aufgabe, so kann diese von anderen Lehrern auch bearbeitet oder sogar gelöscht werden. In Zukunft soll ein Lehrer nur Zugriff auf seine eigenen Aufgaben haben und auch nur diese bearbeiten können.
- Wird der Klasse ein Fach zugewiesen, haben die Schüler automatisch Zugriff auf alle Themen, Lessons und Aufgaben. Der Lehrer soll jedoch auch einzelne Themen, Lessons oder Aufgaben freischalten können.
- Erstellen die Schüler einen Forumsbeitrag, erhält der Lehrer keine Benachrichtigung. Er soll jedoch eine Nachricht erhalten, so dass er auf unbeantwortete Fragen von Schülern eingehen kann.

7.5.3 Weiterentwicklung

In einem nächsten Schritt sollten die unter Abschnitt 7.5.2 genannten Punkte verbessert werden. Es gibt jedoch viele interessante Features, welche noch zusätzlich implementiert werden können.

- Ein globales Forum, in welchem die Schüler Fragen stellen können, welche nicht direkt auf eine Aufgabe bezogen sind.
- Quizze, welche die Schüler während des Unterrichts lösen können. So sieht der Lehrer sehr schnell, was die Schüler noch nicht verstanden haben.
- Das Dashboard könnte so erweitert werden, dass auch allgemeine Informationen, wie kommende Events oder Anlässe, dargestellt werden können.
- Intelligentes Zuweisen von Aufgaben. Im Moment können die Schüler selber sagen, an welchen Fächern oder Übungen sie arbeiten. Es könnte ein System implementiert werden, welches erkennt, in welchen Themen der Schüler Mühe hat und automatisch solche Übungen zuweisen, dass sich der Schüler verbessern kann.
- In einer Französisch Lektion sprechen deutschsprachige Kinder miteinander auf französisch. Mit einer solchen Plattform wäre es möglich, dass ein deutschsprachiger Lehrer eine Lektion zusammen mit einem Lehrer aus der Westschweiz plant. Während der Lektion wird jedes deutschsprachige Kind einem französisch sprechenden Kind zugewiesen, welche dann untereinander sprechen können. Dies macht vielleicht den Französischunterricht noch ein wenig interessanter.

7.5.4 Nicht Funktionale Anforderungen

Functionality

Die Funktionalität der Anwendung ist gegeben. Man kann die Anwendung an mehreren Schulen deployen, ohne dabei Änderungen am Backend vornehmen zu müssen. Die Security Aspekte konnten auch umgesetzt werden. Die Benutzergruppen haben unterschiedliche Rechte und es kann nicht auf Inhalte zugegriffen werden, wenn man die Rechte dazu nicht hat. Zudem kann man sich nur per HTTPS mit dem Server verbinden.

Usability

Die Responsiveness der Anwendung ist leider nicht gegeben. Zu Beginn wollte man Schüler die Anwendung auch auf Mobile Phones oder Tablets verwenden können. Man merkte jedoch schnell, dass man zu wenig Zeit hat, um die geforderte Funktionalität zu implementieren und gleichzeitig gutes User Interface zu erstellen. Daher entschied man sich, dass man sich auf das Backend konzentriert. Natürlich wurde darauf geachtet, dass die Anwendung gut bedienbar und ansprechend ist. Auf kleineren Displays kann es aber vorkommen, dass die Buttons am falschen Ort sind oder nicht korrekt formatiert sind.

Reliability

Die gestellten Anforderungen an die Availability der Anwendung wurden erfüllt. Die Applikation war während mehrerer Wochen auf dem Web Server deployed und war ununterbrochen verfügbar. Die Fault Tolerance ist auch gegeben. Bei jedem Request wird geprüft, ob diese Person zu dieser Aktion berechtigt ist. So wird sicher gestellt, dass die URL im Browser nicht angepasst werden kann. Auch während solcher Abfragen blieb der Web Server ständig erreichbar.

Performance

Dieser Punkt wurde bereits im Kapitel 7.2.5 angesprochen. Ca. 6% der 38'000 Requests lagen über dem gesetzten Grenzwert von einer Sekunden. Man geht aber davon aus, dass sich dieses Problem mit einem Server mit mehr Performance löst.

Supportability

Bei der Implementation wurde darauf geachtet, dass die Anwendung modular aufgebaut ist. So ist es ohne Probleme möglich, weitere Apps hinzuzufügen und so die Funktionalität zu erweitern.

Zudem befindet sich die gesamte Anwendung in einem Docker Container. So kann die Anwendung ohne Probleme auf einem neuen Server installiert werden. Dieser Punkt konnte also auch erreicht werden.

8 Projektplan

Während der Bachelorarbeit wurde der Projektplan als eigenständiges Dokument geführt. Falls Änderungen vorgekommen sind, wurde der Projektplan dementsprechend angepasst. Gegen Ende der Bachelorarbeit wurde der Projektplan in die Dokumentation eingefügt.

8.1 Inhalt

Zweck

Dieses Dokument beschreibt die Planung der Bachelorarbeit, in welchem eine Lernplattform für Schulen entwickelt wird.

Gültigkeitsbereich

Dieses Dokument ist während der gesamten Laufzeit der Bachelorarbeit gültig. Die Änderungsgeschichte kann in Github nachverfolgt werden.

Referenzen

Dieses Dokument wurde mit dem Wissen erstellt, welches in den Modulen Software Engineering 1 & 2, der Studienarbeit sowie in gewissen Grenzen in Cloud Infrastructure und Web Engineering & Design, vermittelt wird.

8.2 Projektübersicht

Bei dieser Bachelorarbeit wird eine Lernplattform erstellt, mit welchem Lehrer und Schüler zusammenarbeiten können. Auf dieser Plattform kann die Theorie der einzelnen Fächer in Form von Videos oder Theoriezusammenfassungen vermittelt werden. Zudem können Übungen erstellt und zur Verfügung gestellt werden.

Der Lehrer kann eigene Aufgaben erfassen, welche die Schüler lösen können. Nachdem die Aufgaben gelöst wurden, sieht der Lehrer eine Statistik, ob und wie gut die einzelnen Aufgaben gelöst wurden. Falls eine Aufgabe überdurchschnittlich schlecht gelöst wurde, kann er diese direkt im Unterricht ansprechen und allfällige Fragen klären.

Die Kernaufgabe dieser Bachelorarbeit ist der aufgabenspezifische Teil. Ferner kann die Plattform aber noch um zusätzliche Features wie einen Chat oder ein Forum erweitert werden.

Lieferumfang

Folgende Dokumente werden am Ende der Bachelorarbeit abgeliefert:

- Abstract
- Aufgabenstellung
- Projektplan
- Projektdokumentation
- Installationsanleitung
- Bedienungsanleitung
- Einverständniserklärung
- Erklärung zur Urheberschaft
- Passwörter
- Persönliche Berichte
- Protokolle
- Source Code

8.3 Organisation

Personen und Rollen

Nachfolgend werden alle an dieser Bachelorarbeit beteiligten Personen und ihre Rollen aufgelistet.

Luca Gubler

Die Idee für das Thema dieser Bachelorarbeit stammt von Luca. Aus diesem Grund überwacht er den groben Projektverlauf der Arbeit. Bei der Arbeit selber kümmert er sich hauptsächlich um die Infrastruktur, die Datenbank und das Backend der Applikation.

Alessandro Bonomo

Alessandro erstellt das User Interface und kümmert sich um die Entwicklung des Frontends. Er hilft jedoch auch bei der Implementation von Features im Backend.

Externe Personen

Bei dieser Bachelorarbeit übernimmt Professor Frank Koch die Rolle des Betreuers und Stephan Meier die Rolle des Experten. Zusätzlich wird Professor Laurent Metzger diese Bachelorarbeit als interner Co-Examinator betreuen.

8.4 Management Abläufe

Zeitbudget

Die Bachelorarbeit begann in der Woche vom 16. September 2019 und dauert insgesamt 17 Wochen. Für das Erreichen der 12 ECTS ist geplant, dass jedes Teammitglied 360 Stunden arbeitet. Daraus resultiert eine durchschnittliche Arbeitszeit von knapp 24 Stunden pro Woche.

Projektstart	13.10.2019
Projektdauer	13 Wochen
Arbeitsstunden pro Person	23h pro Woche, Total 300h
Arbeitsstunden Total	600h
Projektende	10.01.2020

Tabelle 5: Übersicht Zeitaufwand

Für die Bachelorarbeit stehen total 720 Stunden zur Verfügung. Da jedoch für das erste Thema ca. 60 Arbeitsstunden pro Person aufgewendet wurden, stehen für die neue Arbeit noch total 600 Stunden zur Verfügung. Mit dem definierten Projektumfang wird diese Arbeitszeit voraussichtlich vollständig ausgenutzt. Sollte der Umfang jedoch früher als erwartet abgeschlossen werden können, kann das Projekt um weitere Funktionalitäten erweitert werden.

Projektmanagement

Als Projektmanagement Methode wurde SCRUM+ gewählt. Bei dieser Methode handelt es sich um einen Mix aus SCRUM und Unified Process und wird von Daniel Keller im Modul "Software Engineering" unterrichtet.

Phasen / Sprints

Bei SCRUM+ wird das gesamte Projekt in die vier Phasen Inception, Elaboration, Construction und Transition eingeteilt. Pro Phase gibt es wiederum einzelne Sprints. Zudem wurden einzelne Meilensteine definiert, welche auf der Tabelle 6 entnommen werden können.

Iterationsplanung

Die Dauer eines Sprints wurde auf 2 Wochen festgelegt. Zu Beginn jedes Sprints setzt sich das Team zusammen, um den nächsten Sprint zu planen. Dabei wird jeweils besprochen, welche Aufgaben des vergangenen Sprints nicht vollständig abgeschlossen werden konnten. Die nicht abgeschlossenen Arbeiten werden mit neu definierten Aufgaben in den neuen Sprint übernommen und jeweils zeitlich abgeschätzt und priorisiert. Da sich im Team nur 2 Mitglieder befinden, wird darauf verzichtet, die Arbeitspakete unter den Teammitgliedern zuzuweisen. Die gesamte Planung und Verwaltung der Aufgaben wird in Jira erledigt.

Meilensteine

Folgende Meilensteine wurden gesetzt. Da der Projektumfang aber relativ gross war und durch den Wechsel der Projektarbeit weniger Zeit zur Verfügung stand, konnten nicht alle Meilensteine eingehalten werden.

Der Meilenstein "Feature Freeze" musste um 2 Wochen nach hinten verschoben werden. Die Zeit reichte schlicht und einfach nicht aus, um alle geforderten Funktionalitäten in der vorgenommenen Zeit umzusetzen.

SW	Meilenstein	Beschreibung
4	M0: Kickoff	Start des Projektes
5	M1: Abschluss Projektplan	Projektplan erstellt und mit Betreuer besprochen
7	M2: End of Elaboration	Use Cases und funktionale sowie nicht funktionale Anforderungen sind erfasst. Mockups Domainanalyse und Konzept für die Architektur sind erstellt.
12	M3: Feature Freeze	Entwicklung der Features ist abgeschlossen, damit man sich auf Bugfixes und Code Qualität konzentrieren kann.
15	M4: Code Freeze	Entwicklung an der Applikation ist abgeschlossen. End of Construction.
17	M5: Projektende	Abgabe der Bachelorarbeit

Tabelle 6: Übersicht Meilensteine

Termine

Folgende Termine sind von der HSR oder dem Betreuer vorgegeben und müssen eingehalten werden.

Datum	Beschreibung
6.1.1010	Erfassung des Abstracts im Online-Tool https://abstract.hsr.ch/ Die Studierenden geben den Abstract für die Diplomarbeitsbroschüre zur Kontrolle an ihren Betreuer/Examinator frei. Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen auf dem Skripteserver zur Verfügung. Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract der Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei.
10.1.2020	Abgabe des Berichts an den Betreuer und Hochladen aller Dokumente auf archiv-i.hsr bis 17 Uhr
10.2.2020	Präsentation der Bachelorarbeit und mündliche Prüfung.

Tabelle 7: Übersicht Termine

8.5 Projektverwaltung

Als Projektverwaltungstool wird Jira verwendet. Man hat sich für dieses Tool entschieden, da es die gewünschte Funktionalität mit sich bringt. Um den Stand der einzelnen Arbeitspakete möglichst genau darzustellen, wurde ein Workflow definiert, welcher jedes Arbeitspaket durchlaufen muss.

Wie im Bild ersichtlich ist, muss jedes Arbeitspaket folgende Status durchlaufen: Open, In Progress, Review und Done. Ein Arbeitspaket muss jeden Schritt im Workflow durchlaufen, ausser Review. Dieser Schritt ist optional und muss nicht bei jedem Arbeitspaket durchgeführt werden.

Zeiterfassung

Die Zeiterfassung wird mit Jira verwaltet. Beim Einfügen von Arbeitspaketen in den Sprint wird die Zeit geschätzt, welche für das Arbeitspaket aufgewendet werden muss. Jede Person, welche an diesem Arbeitspaket gearbeitet hat, kann Zeit auf dieses Arbeitspaket buchen. Am Schluss kann so eine Zeitauswertung über die einzelnen Sprints oder das gesamte Projekt gemacht werden.

Meetings

Die Teammitglieder arbeiten an mindestens zwei Tagen pro Woche zusammen im Bachelorarbeitszimmer. So können Fragen schnell geklärt und sich gegenseitig geholfen werden. Im Normalfall findet jeden Donnerstag ein Meeting mit dem Betreuer Frank Koch statt, bei dem der aktuelle Stand vorgestellt, Probleme besprochen und das weitere Vorgehen geplant wird. Da der Betreuer jedoch immer wieder unterwegs war, wurde oft eine Telefonkonferenz geschaltet. Zudem wurden die im Meeting besprochenen Punkte protokolliert

8.6 Risikomanagement

Risiken

Folgende Risiken sind beim Beginn der Bachelorarbeit erkannt worden:

Nr	Beschreibung	Schaden total [h]	Eintritts- wahrsch.	Gwichteter Schaden [h]
R1	Unterschätzen des Aufwandes	60	20 %	12
R2	Fehlendes KnowHow von Python oder Frameworks	60	20 %	12
R3	Konflikte im Team	8	5 %	0.4
R4	Schlechtes UI Design	32	25 %	8
R5	Technische Fehlkonfigurationen	60	20 %	12
R6	Probleme beim Deployment	24	25 %	6
R7	Probleme mit der Architektur	32	25 %	8

Tabelle 8: Risikoübersicht

Daraus resultiert folgender Risikograph:

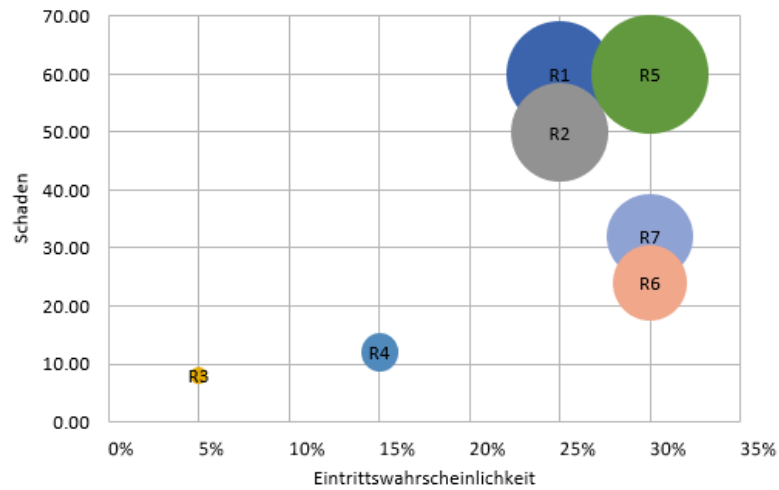


Abbildung 20: Risikograph

In der Inception Phase wurde ein total gewichteter Schaden von 70.7 Arbeitsstunden geschätzt. Durch gezielte Massnahmen konnte dieser Schaden auf 56.2 Arbeitsstunden vermindert werden.

Aufgrund dieser Annahme und da der Zeitraum durch die Verzögerungen zu Beginn der Arbeit recht knapp bemessen ist, wurde zum Schluss der Bachelorarbeit eine Woche Bufferzeit einberechnet.

Massnahmen

Risiken lassen sich in einem grösseren Projekt leider nicht ganz vermeiden. Für die erfassten Risiken wurden Massnahmen definiert, um das Risiko weitgehend zu minimieren. In der nachfolgenden Tabelle sind diese Massnahmen aufgelistet.

Nr	Vorbeugung	Verhalten beim Eintreten
R1	Reserven einplanen und Abgrenzung des Scopes	Reserven nutzen und notfalls den Scope anpassen
R2	In der Elaboration Phase Zeit einplanen, um sich in die einzelnen Phasen einarbeiten zu können	Fehlendes KnowHow aufbauen, gegebenenfalls mit Unterstützung des anderen Teammitgliedes
R3	Zusammenarbeit im Team und gegenseitige Absprache	Notfalls Gespräch mit dem Betreuer
R4	Mockups erstellen und Usability Tests mit externen Personen durchführen	User Interface anpassen
R5	KnowHow in der Elaboration Phase aufbauen und für den Notfall Backups erstellen	Restore der Backups und redeployment des Systems
R6	Vorläufig testen, wie eine Applikation deployed wird	Genauere Recherche, wie das Deployment funktioniert
R7	Architektur so planen, dass sie einfach erweitert und angepasst werden kann	Architektur anpassen

Tabelle 9: Massnahmen

Auswertung

Während der gesamten Projektdauer sind eigentlich nur zwei der oben genannten Risiken aufgetreten. Ein grosses Problem war das fehlende KnowHow des Django Frameworks. Besonders zu Beginn der Projektarbeit gab es oft Fragen und Unklarheiten, welche zuerst geklärt werden mussten.

Dies führte unweigerlich zum zweiten Risiko, nämlich des sehr knapp bemessenen Zeitraums. Durch den Wechsel des Themas der Bachelorarbeit standen 4 Wochen weniger Zeit zur Verfügung. Als immer deutlicher wurde, dass man nicht den gesamten Scope der Arbeit abdecken kann, wurden die einzelnen Aufgaben neu priorisiert. Aufgrund des fehlenden KnowHows beim Entwickeln von User Interfaces entschied man sich, mehr Fokus auf die Funktionalitäten im Backend zu legen. Zudem wurde der Scope dementsprechend angepasst. Da sich die Quizze und Aufgaben sehr ähneln, hat man sich entschieden, auf die Quizze zu verzichten.

8.7 Qualitätsmanagement

Dokumentation

Die Dokumentation der Bachelorarbeit wird mit \LaTeX und in einem eigens dafür eingerichteten Git Repository erstellt. Somit ist es möglich, jede Änderung an der Dokumentation zu verfolgen. Wird ein Kapitel geschrieben, wird es vom anderen Teammitglied gegengelesen und falls nötig angepasst. Somit stammt die Dokumentation von beiden Teammitgliedern und repräsentiert nicht die Meinung einer einzelnen Person.

Entwicklung

Der gesamte Source Code befindet sich auf Github. Der Master Branch wird als protected Branch eingerichtet. Ein Branch kann also nicht direkt in den Master Branch gemerged werden, sondern muss mittels eines Pull Requests gemerged werden. Dies hat den Vorteil, dass sämtlicher Code das Vier-Augen Prinzip durchlaufen hat.

Zudem wird Travis CI verwendet, welches nach jedem Push die selbst geschriebenen Tests durchführt. Somit kann man sich auch sicher sein, dass sämtlicher Code auch wirklich alle Tests besteht.

Code Reviews

Bei jedem Pull Request wird auch gleich ein Code Review durchgeführt. Dies hat den Vorteil, dass der Code, welcher in den Master Branch gemerged wird, auch den Anforderungen beider Teammitglieder entspricht.

Code Qualität

Black⁵² ist ein "Code Formatter", welcher es ermöglicht, den kompletten Code einer Anwendung automatisch zu formatieren. Dies ist besonders nützlich, wenn in einem Team gearbeitet wird. Auch wenn man sich auf einen Code Style geeinigt hat, kann es trotzdem zu Unterschieden kommen. Black formatiert den Code automatisch. Die Entwickler haben über die ganze Anwendung hinweg den gleichen Code Style und können sich besser auf den Inhalt des Codes, anstatt auf die Formatierung, konzentrieren.

⁵² *The Uncompromising Code Formatter*. URL: <https://github.com/psf/black> [Besucht am 23. Dezember 2019].

Unter dem Begriff Linting⁵³ versteht man, wenn der Code auf Fehler untersucht wird. Solche Tools zeigen auch gleich an, wie man den Code verbessern kann. Dies hilft dem Entwickler, besseren Code zu schreiben. So wird der Code zum Beispiel nach Missachtung von Code Standards, Syntax Fehlern oder schlechter Formatierung untersucht. Flake8⁵⁴ wird als Linting Tool eingesetzt, um die Entwickler beim Schreiben von Code zu unterstützen.

Tools und Infrastruktur

Für die Infrastruktur werden lediglich 2 Notebooks und ein Server benötigt. Die Notebooks stammen von den Diplomanden und der Server wird durch die HSR zur Verfügung gestellt.

Als IDE wurde entschieden, sich auf PyCharm von JetBrains zu beschränken. Dies hat den Vorteil, dass man sich beim Pair-Programming schnell zurecht findet.

Als Projektmanagement Tool wird Jira eingesetzt.

Die Dokumentation und der Projektplan werden mit \LaTeX erstellt. Kleinere Dokumente oder das Sitzungsprotokoll werden mit Word erfasst.

Travis wird als CI Tool eingesetzt. Dabei werden auch gleich die Tests durchgeführt und die Code Qualität überprüft.

⁵³ *What is Flake8 and why we should use it?* URL: <https://medium.com/python-pandemonium/what-is-flake8-and-why-we-should-use-it-b89bd78073f2> [Besucht am 23. Dezember 2019].

⁵⁴ *Your Tool For Style Guide Enforcement.* URL: <http://flake8.pycqa.org/en/latest/> [Besucht am 23. Dezember 2019].

8.8 Zeitauswertung

Wie der Abbildung 21 entnommen werden kann, war der Zeitaufwand beider Projektmitglieder beinahe identisch. Es gab eine kleine Differenz, jedoch hat diese keine Auswirkung auf den prozentualen Anteil eines Studenten. Luca Gubler arbeitete 323 Stunden und Alessandro Bonomo arbeitete 318 Stunden an der Bachelorarbeit.



Abbildung 21: Zeitaufwand nach Mitgliedern

In den Abbildungen 22 und 23 ist der prozentuale Aufwand pro Phase, beziehungsweise der effektive Aufwand pro Projektmitglied und Phase, abgebildet.

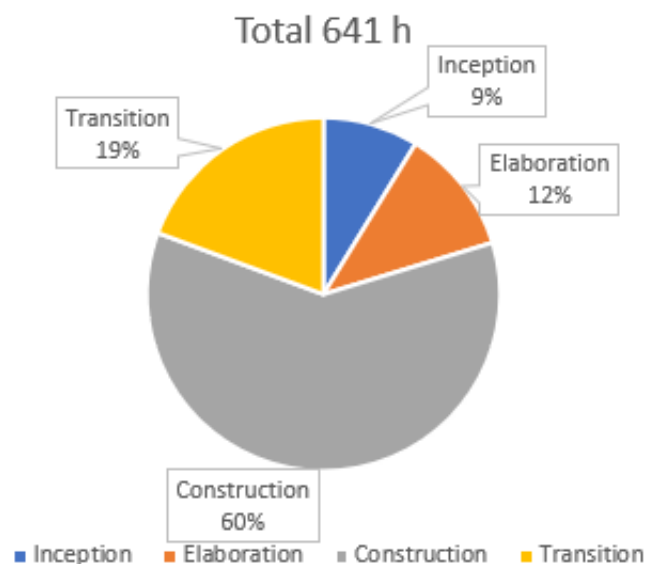


Abbildung 22: Zeitaufwand nach Phasen

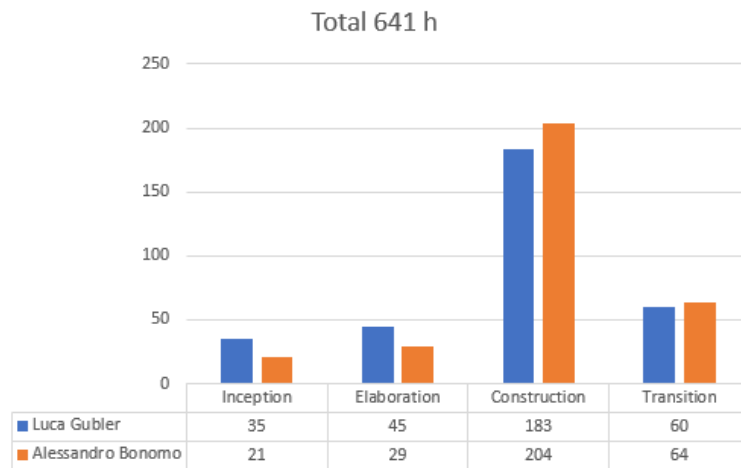


Abbildung 23: Zeitaufwand nach Phasen und Person

Die Abbildung 24 zeigt, wie viel Zeit jedes Projektmitglied pro Woche in die Arbeit investiert hat. Durch Job und andere Module, entstanden gewisse Schwankungen über das Semester verteilt. Dennoch ist es beiden Projektmitgliedern gelungen, den geforderten Zeitaufwand von je 300 Stunden zu leisten.

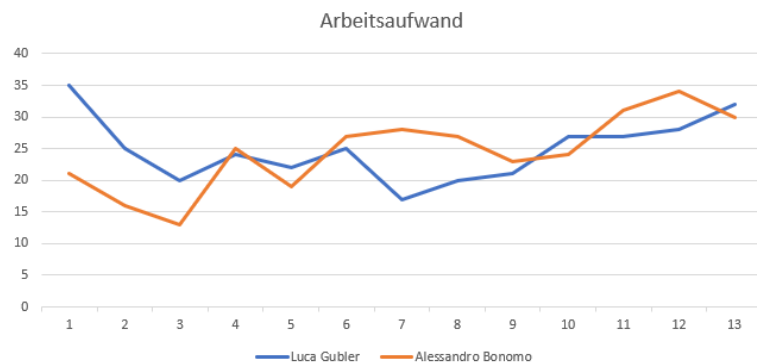


Abbildung 24: Zeitaufwand pro Woche

9 Software Dokumentation

Die Software Dokumentation wurde mit Sphinx erstellt. Die HTML Version befindet sich in der Anwendung im Verzeichnis "docs" und kann direkt im Browser angesehen werden. Zudem wurde ein PDF generiert, welches auf den nachfolgenden Seiten eingefügt wurde.

GETTING STARTED

1.1 Installation von Aufgaben-Coach

Aufgaben-Coach ist ein normales Python Package und kann direkt von Github heruntergeladen werden:

```
git clone https://github.com/lucagubler/lern_plattform.git
virtualenv venv
source venv/bin/activate
pip install -r requirements.txt
```

1.2 Anwendung starten

Aufgaben-Coach wurde mit Docker entwickelt. Ist Docker installiert, kann die Anwendung als Docker Container gestartet werden:

```
docker-compose up -d --build
```

Wird die Anwendung zum ersten Mal gestartet, muss die Datenbank erstellt und ein neuer Benutzer hinzugefügt werden. Dazu muss man sich zuerst mit dem Docker Container verbinden und anschliessend folgende Befehle eingeben:

```
# Anzeigen alles Docker Container
docker ps -a

# Verbinden zum Docker Container
docker exec -it lern_plattform_web_1 /bin/bash

# Datenbank und User erstellen
python3 lern_plattform/manage.py migrate
python3 lern_plattform/manage.py createsuperuser
```

Note: Dieser User wird als Administrator erstellt. Damit er die benötigten Rechte hat, muss er einem Admin Profil und der Gruppe Admin hinzugefügt werden. Dazu muss die URL 0.0.0.0/admin aufgerufen werden. Bei Groups muss eine neue Gruppe mit dem Namen 'admin' erstellt werden. Wird nun der User ausgewählt, kann ihm die zuvor erstellte Gruppe zugeteilt werden. Unter AdminProfile kann dann mit diesem User ein neues Profil angelegt werden.

KONFIGURATION VON AUFGABEN-COACH

Im Aufgaben-Coach Repository existiert ein `aufgaben_coach_settings_example.py` File. Dies ist eine Default Konfigurationsdatei für die Anwendung. Um Aufgaben-Coach mit einer eigenen Konfiguration zu starten, kann diese Datei kopiert und unter dem Namen `aufgaben_coach_settings.py` gespeichert werden.

Folgende Einstellungen können angepasst werden:

- **DATABASE:** (dict) Die Einstellungen für die Datenbank. Genauere Informationen können unter [DATABASES](#) in der Dokumentation von Django gefunden werden.
- **LANGUAGE_CODE:** (str) Einstellung für den Sprach Code, welcher verwendet werden kann. Genauere Informationen können unter [LANGUAGE_CODE](#) in der Dokumentation von Django gefunden werden. Für den Moment ist Aufgaben-Coach nur auf Deutsch `de-ch` verfügbar.
- **TIME_ZONE:** (str) Einstellung der Time Zone. Genauere Informationen können unter [TIME_ZONE](#) in der Dokumentation von Django gefunden werden. Aufgaben-Coach verwendet `Europe/Zurich` als Time Zone.
- **EMAIL:** (dict) Hier können die E-Mail Einstellungen angepasst werden.
- **SECRET_KEY:** (str) Der Secret Key wird für die kryptographische Signatur verwendet, weshalb es ein zufälliger und unvorhersehbarer Wert sein muss. Genauere Informationen können unter [SECRET_KEY](#) in der Dokumentation von Django gefunden werden.
- **DEBUG:** (bool) Dieser Wert gibt an, ob die Anwendung im Debug Modus gestartet wird oder nicht. In einer produktiven Umgebung sollte Debug auf `False` gesetzt sein.
- **ALLOWED_HOSTS:** (list) Die Hostnamen der Webseiten, welche über Django erreichbar sind. Genauere Informationen können unter [ALLOWED_HOSTS](#) in der Dokumentation von Django gefunden werden.
- **BLEACH_ALLOWED_TAGS:** (list) Liste aller erlaubten HTML tags.
- **BLEACH_ALLOWED_ATTRIBUTES:** (list) Liste aller erlaubten HTML Attribute.
- **BLEACH_ALLOWED_STYLES:** (list) Liste aller erlaubten CSS Attribute. Dies ist jedoch nur möglich, wenn `style` in `BLEACH_ALLOWED_ATTRIBUTES` vorkommt.
- **BLEACH_ALLOWED_PROTOCOLS:** (list): Liste aller erlaubten Protokolle innerhalb `src`. Dies ist jedoch nur möglich, wenn `src` in `BLEACH_ALLOWED_ATTRIBUTES` vorkommt.
- **BLEACH_STRIP_TAGS:** (bool) Nicht erlaubte tags werden entfernt, respektive mit HTML escaped characters abgespeichert.
- **BLEACH_STRIP_COMMENTS:** (bool) Entferne HTML Kommentare.

ERWEITERUNG VON AUFGABEN-COACH

Django erlaubt es, neue Apps zu erstellen und in ein Projekt zu integrieren. Nachfolgend wird erklärt, wie ein neues App erstellt wird und wie man es in Aufgaben-Coach einbinden kann.

Warning: Auf dieser Seite befindet sich eine kurze Dokumentation über die Django Views und wie diese bei unserem Projekt eingesetzt werden könnten. Diese Dokumentation ist jedoch nicht vollständig. Die vollständige Dokumentation wird von [Django](#) bereitgestellt.

Note: Im Django Admin Interface, erreichbar unter der URL */admin* kann auf die Dokumentation der einzelnen Views oder Models zugegriffen werden. Um zur Dokumentation zu gelangen, muss man auf *Documentation* am rechten oberen Rand des Browser Fensters drücken.

3.1 Neues App erstellen

Zuerst muss ein neues App erstellt werden:

```
python3 manage.py startapp my_test_app
```

Dieses App ist im selben Verzeichnis erstellt worden, in welchem sich *manage.py* befindet. Um die Apps übersichtlicher zu halten, werden alle erstellten Apps nach *lern_plattform/apps/* verschoben. So bleiben die Verzeichnisse übersichtlich.

Als nächstes muss die erstellte App in den *INSTALLED_APPS* in *lern_plattform/settings.py* hinzugefügt werden. Dazu muss einfach eine neue Zeile mit dem Namen der App hinzugefügt werden.

Damit diese App über URLs erreichbar ist, muss ein Eintrag in *lern_plattform/urls.py* hinzugefügt werden. Dieser könnte folgendermassen aussehen:

```
path('my_simple_app/', include('lern_plattform.apps.my_simple_app.urls'), namespace=
↳ 'my_simple_app')
```

Bei der neuen App muss noch ein leeres *urls.py* File erstellt werden. Später wird angeschaut, wie genau die URLs definiert werden.

3.2 Model

Möchte man etwas in der Datenbank speichern, muss das *models.py* File angepasst werden, welches sich im *my_test_app* directory befindet. In diesem Beispiel wird eine einfache Tabelle *Car* erstellt. Der Code dafür sieht folgendermassen aus:

```
from django.db import models

class Car(models.Model):
    manufacturer = models.CharField(max_length=30)
```

Diese Tabelle ist jedoch noch nicht in der Datenbank. Um dies zu tun, muss der Befehl *python3 manage.py makemigrations* eingegeben werden. Dies schaut, ob sich etwas in den Models geändert hat und erstellt gleich auch das migrations File im directory *migrations*. Dieses File enthält die Informationen, wie die Migration genau durchgeführt werden soll. Die Migration wurde jedoch noch nicht durchgeführt. Dies kann mit dem Befehl *python3 manage.py migrate* gemacht werden. Im Terminal sollte nun folgendes stehen:

```
Migrations for 'my_test_app':
  lern_plattform/apps/my_test_app/migrations/0001_initial.py
    - Create model Car
```

Damit dieses Model im Admin Panel von Django ersichtlich ist, muss es im *admin.py* File erfasst werden:

```
from django.contrib import admin
from .models import Car

admin.site.register(Car)
```

Wenn man nun die auf die Seite */admin* geht, wird Car da aufgelistet.

3.3 ListView

Mit einer ListView können die einzelnen Daten aus der Datenbank geladen und als Liste dargestellt werden. Folgender Code muss ins File *views.py* geschrieben werden:

```
from django.views.generic import ListView

class CarListView(ListView):
    model = Car
    template_name = 'my_test_app/car_list.html'
    context_object_name = 'cars'
```

Die ListView lädt alle Daten aus einer Datenbank. Das `model = Car` gibt dabei an, von welchem Model die Daten geladen werden sollen. Mit dem `template_name` kann angegeben werden, welches Template gerendert werden soll. Befolgt man die Django Konvention und benennt alle Templates nach dem vorgegebenen Schema, könnte man in der View das Template weglassen und Django würde selber wissen, welches Template verwendet werden muss. Da dies jedoch sehr unübersichtlich wird, hat man das Template immer angegeben. Das letzte Statement ist der `context_object_name`. Dieser wird so direkt dem Template übergeben und kann verwendet werden.

3.4 Template

Die Templates befinden sich im Directory der `my_test_app` unter `templates/my_test_app/`. Für die ListView kann nun das entsprechende `car_list.html` File erstellt werden. Der Inhalt könnte folgendermassen aussehen:

```
{% extends "base.html" %}
{% block content %}
    {% for car in cars %}
        {{ car }}
    {% endfor %}
{% endblock %}
```

Dieses Template würde nun alle Cars auf der Webseite anzeigen. Zuerst muss aber noch eine URL erstellt werden, auf welche man zugreifen kann.

3.5 URL

Im File `urls.py` kann nun ein Eintrag erstellt werden, welche die URL auf die korrekte View leitet. Dies könnte folgendermassen aussehen:

```
from django.urls import path
from . import views as my_test_app_views

urlpatterns = [
    path('cars/', my_test_app_views.CarListView.as_view(), name='car-list'),
]
```

Gibt man nun die URL `my_test_app/cars` im Browser ein, kommt man auf die Webseite, auf welcher alle Cars dargestellt wurden. Die Seite ist aber noch leer, da es noch keine Möglichkeit gibt, neue Cars zu erstellen.

3.6 CreateView

Um Cars erstellen zu können, wird eine Create View erstellt. Diese befindet sich unterhalb der ListView im *views.py* File:

```
class CarCreateView(CreateView):
    model = Car
    fields = ['manufacturer']
    template_name = 'my_test_app/car_form.html'
```

Note: *CreateView* muss noch importiert werden!

Anschliessend muss erneut ein Eintrag im *urls.py* File hinzugefügt werden. Dieses File sollte nun folgendermassen aussehen:

```
from django.urls import path
from . import views as my_test_app_views

urlpatterns = [
    path('cars/', my_test_app_views.CarListView.as_view(), name='car-list'),
    path('car/create/', my_test_app_views.CarCreateView.as_view(), name='car-create'),
]
```

Zusätzlich muss auch noch ein neues Template mit dem Namen *car_form.html* erstellt werden:

```
{% extends "base.html" %}
{% load crispy_forms_tags %}

{% block content %}
<form method="post">
    {% csrf_token %}
    {{ form|crispy }}
    <input type="submit" value="Save">
</form>
{% endblock %}
```

Öffnet man die neu erstellte Seite und speichert ein neues Element, wird dieses zwar gespeichert, jedoch erscheint folgende Fehlermeldung: *No URL to redirect to*. Grund dafür ist, dass Django den User nach dem Erstellen eines Elements auf eine andere Seite weiterleiten möchte. Dies wurde aber noch nicht angegeben. Das Problem wird gelöst, indem man der CreateView die Funktion *get_success_url()* hinzufügt:

```
class CarCreateView(CreateView):
    model = Car
    fields = ['manufacturer']
    template_name = 'my_test_app/car_form.html'

    def get_success_url(self):
        return reverse('car-list')
```

Note: *from django.shortcuts import reverse* muss noch importiert werden.

Erstellt man nun ein neues Car Objekt, wird man automatisch wieder zur ListView weitergeleitet.

3.7 DetailView

Die DetailView lädt einzelne Objekte aus der Datenbank und stellt diese auf einer Webseite dar. Mit folgendem Code können die Details einzelner Views dargestellt werden:

```
class CarDetailView(DetailView):
    model = Car
    template_name = 'my_test_app/car_detail.html'
    context_object_name = 'car'
```

Note: Dazu muss wiederum die *DetailView* importiert werden.

Das *car_detail.html* Template kann hier sehr simpel gehalten werden:

```
{% extends "base.html" %}

{% block content %}
    {{ car }}
{% endblock %}
```

Bei der URL muss nun ein *pk* angegeben werden, anhand welchem ein Eintrag aus der Datenbank geladen wird:

```
path('car/<int:pk>/', my_test_app_views.CarDetailView.as_view(), name='car-detail')
```

Geht man nun auf die Seite */my_test_app/car/1/* wird das Car Objekt mit der ID 1 geladen und dargestellt.

3.8 UpdateView

Es kann aber auch vorkommen, dass man existierende Objekte anpassen möchte. Dafür eignet sich die UpdateView:

```
class CarUpdateView(UpdateView):
    model = Car
    fields = ['manufacturer']
    template_name = 'my_test_app/car_form.html'
```

Note: Dazu muss wiederum die *UpdateView* importiert werden. Die UpdateView verwendet das gleiche Template wie die Createview.

Die URL dazu muss wiederum einen *pk* enthalten, damit Django weiss, welches Objekt verändert werden soll:

```
path('car/<int:pk>/update', my_test_app_views.CarUpdateView.as_view(), name='car-
↪update')
```

10 Dankausgang

Wir möchten uns ganz herzlich bei Professor Frank Koch bedanken, welcher uns ermöglicht hat, diese Bachelorarbeit vom Konzept in die Praxis umzusetzen.

Zudem möchten wir uns bei Daniela und Jürg Gubler, Abirsana Sreetharan, Daniela Bonomo und Vangelis Al-Sghir für die Korrekturlesung der Arbeit bedanken.

Luca Gubler möchte einen speziellen Dank an seine Eltern ausrichten, welche ihn durchs Studium begleiteten und bei Problemen immer zur Seite standen.

Ein spezieller Dank gebührt Abirsana Sreetharan, welche während der gesamten Bachelorarbeit mit viel Geduld, Interesse und Hilfsbereitschaft zur Seite stand.

Auch Alessandro Bonomo möchte sich bei seinen Eltern bedanken, welche ihn sein ganzes Leben bedingungslos unterstützt haben.

11 Fazit

Der Anfang der Bachelorarbeit war ein wenig chaotisch. Während vier Wochen arbeiteten wir an einem anderen Thema. Aufgrund mehrerer Vorkommnisse waren wir aber gezwungen, das Thema der Arbeit zu wechseln. Dadurch ging jedoch knapp 20% der Zeit verloren. Da das neue Thema aber einen relativ grossen Umfang hatte, war es während der Arbeit nicht immer einfach. Viele der eingesetzten Technologien waren neu und man musste sich zuerst einarbeiten. Dies führte schlussendlich zu einem relativ strikt getakteten Zeitplan.

Nach Abschluss der Arbeit können wir aber sagen, dass wir stolz auf die erreichten Ergebnisse sind. Besonders toll war, dass wir unsere eigene Idee als Bachelorarbeit umsetzen konnten. Wir waren auch beim Erstellen der Aufgabenstellung involviert und konnten unsere eigenen Ideen einbringen. Zudem ist es toll, dass wir eine Arbeit erstellen konnten, welche nicht einfach zum erhalten des Bachelor Diploms dient, sondern nach Abschluss des Studiums weiter entwickelt werden soll.

Abbildungsverzeichnis

1	Use Case Diagramm	25
2	System Overview	30
3	Vollständiges Domain Model	31
4	Domain Model - User Abschnitt	32
5	Domain Model - Schulfächer	34
6	Domain Model - Aufgaben	35
7	Domain Model - Forum	36
8	Deployment Diagramm	40
9	Mockup Smartphone Dashboard	45
10	Mockup Aufgabe erfassen Desktop	46
11	Mockup Navigationsbar Desktop	46
12	Mockup Breadcrumbs Desktop	47
13	Django Architektur Diagramm	49
14	Anpassung Klassenverwaltung	54
15	Informationen einer Klasse	54
16	Performance Test Übersicht vorher	55
17	Anfragen pro Sekunde	56
18	Performance Test Übersicht nachher	56
19	Failed Travis Build	57
20	Risikograph	69
21	Zeitaufwand nach Mitgliedern	73
22	Zeitaufwand nach Phasen	73
23	Zeitaufwand nach Phasen und Person	74
24	Zeitaufwand pro Woche	74

Tabellenverzeichnis

1	Aktoren	26
2	Non Functional Requirements	29
3	Usability Test Rollen	52
4	Code-Statistik	58
5	Übersicht Zeitaufwand	65
6	Übersicht Meilensteine	66
7	Übersicht Termine	67
8	Risikoübersicht	68
9	Massnahmen	70

Akronyme

CI Continuous Integration. 56, 57, 72

CMS Content Management System. 17, 18

DACH Deutschland, Österreich und der Schweiz. 17

DTL Django Template Language. 49

ECM Enterprise Content Management. 18

HSR Hochschule für Technik Rapperswil. 5, 16, 18, 21, 67, 72

HSTS HTTP Strict Transport Security. 50

HTML Hypertext Markup Language. 50, 75

LMS Learning Management System. 5, 17, 18, 21

MTTR Mean Time To Repair. 56

MTV Model Template Views. 48

MVC Model View Controller. 48

NFR Non Functional Requirements. 29

ORM Object Relational Mapper. 38, 49

URL Uniform Resource Locator. 48, 51, 62

WCM Web Content Management. 18

WSGI Web Server Gateway Interface. 41

WYSIWYG What You See Is What You Get. 18

XSS Cross-Site Scripting. 50

References

- Lehrplan21 - Ziele*. URL: <https://lehrplan21.ch/ziele> [Besucht am 4. Januar 2020].
- Moodle - Open-source learning platform*. URL: <https://moodle.org/> [Besucht am 20. Oktober 2019].
- Mit Spass lernen und Noten verbessern*. URL: <https://www.sofatutor.ch/> [Besucht am 20. Oktober 2019].
- English lessons, designed for you*. URL: <https://class.ef.com/> [Besucht am 20. Oktober 2019].
- Erfolgreich unterrichten mit Sofatutor*. URL: <https://www.sofatutor.ch/lehrer-und-schulen> [Besucht am 20. Oktober 2019].
- content management system (CMS)*. URL: <https://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS> [Besucht am 29. Dezember 2019].
- What is SharePoint?* URL: <https://support.office.com/en-us/article/what-is-sharepoint-97b915e6-651b-43b2-827d-fb25777f446f> [Besucht am 29. Dezember 2019].
- Total number of Websites*. URL: <https://www.internetlivestats.com/total-number-of-websites/> [Besucht am 4. Januar 2020].
- CMS Usage Distribution on the Entire Internet*. URL: <https://trends.builtwith.com/cms/traffic/Entire-Internet> [Besucht am 29. Dezember 2019].
- The Best Learning Management Systems based on User Experience*. URL: <https://elearningindustry.com/directory/software-categories/learning-management-systems/best/user-experience?version=2018-02-22> [Besucht am 22. Dezember 2019].
- Looop - The LMS that drives performance*. URL: <https://www.looop.co/> [Besucht am 22. Dezember 2019].
- The Flipped Class Manifest*. URL: <http://www.thedailyriff.com/articles/the-flipped-class-manifest-823.php> [Besucht am 20. Oktober 2019].
- 10 Pros And Cons Of A Flipped Classroom*. URL: <https://www.teachthought.com/learning/10-pros-cons-flipped-classroom/> [Besucht am 20. Oktober 2019].
- Krummenacher, Jörg. *Die Städte rüsten die Schulen digital auf*. URL: <https://www.nzz.ch/schweiz/staedte-und-digitale-schule-ld.1489721> [Besucht am 20. Oktober 2019].
- "Moodle. Not even once." What is your the worst project in career?* URL: <https://www.reddit.com/r/webdev/comments/2l3hy9/> [Besucht am 20. Oktober 2019].
- Definition Personas*. URL: <https://www.onlinemarketing-praxis.de/glossar/personas> [Besucht am 13. November 2019].
- Grady Robert B. und Caswell, Deborah L. *Software metrics: establishing a company-wide program*. URL: <https://archive.org/details/softwaremetricse00grad/page/159> [Besucht am 23. Oktober 2019]. 1987.

Native app vs. progressive web app (PWA): Everything you need to know. URL: <https://www.adjust.com/blog/native-app-vs-progressive-web-app/> [Besucht am 13. Dezember 2019].

How to Implement Multiple User Types with Django. URL: <https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html> [Besucht am 20. November 2019].

Building Multi Tenant Applications with Django. URL: <https://books.agiliq.com/projects/django-multi-tenant/en/latest/shared-database-shared-schema.html> [Besucht am 22. Dezember 2019].

Developer Survey Results 2019. URL: <https://insights.stackoverflow.com/survey/2019> [Besucht am 25. Dezember 2019].

Python 2019 - The state of Developer Ecosystem in 2019 Infographic. URL: <https://www.jetbrains.com/lp/devecosystem-2019/python/> [Besucht am 25. Dezember 2019].

Flask | The Pallets Projects. URL: <https://palletsprojects.com/p/flask/> [Besucht am 25. Oktober 2019].

Design Decisions in Flask. URL: <https://flask.palletsprojects.com/en/1.1.x/design/> [Besucht am 25. Oktober 2019].

Django at a glance. URL: <https://docs.djangoproject.com/en/3.0/intro/overview/> [Besucht am 25. Oktober 2019].

Django Advantages and Disadvantages – Why You Should Choose Django? URL: <https://data-flair.training/blogs/django-advantages-and-disadvantages/> [Besucht am 25. Oktober 2019].

Django Deployment Statistics. URL: <https://djangosites.org/stats/> [Besucht am 27. Dezember 2019].

Django Database Migration. URL: <https://docs.djangoproject.com/en/3.0/topics/migrations/> [Besucht am 22. November 2019].

Usage of web servers. URL: https://w3techs.com/technologies/overview/web_server [Besucht am 05. Januar 2020].

NGINX vs. Apache (Pro/Con Review, Uses and Hosting for Each). URL: <https://www.hostingadvice.com/how-to/nginx-vs-apache/> [Besucht am 05. Januar 2020].

Apache vs nginx - performance comparison. URL: <https://theorganicagency.com/blog/apache-vs-nginx-performance-comparison/> [Besucht am 05. Januar 2020].

What Is WSGI? URL: <https://www.liquidweb.com/kb/what-is-wsgi/> [Besucht am 05. Januar 2020].

100+ Questions You Must Ask When Developing A Website. URL: <https://marketingland.com/100-questions-you-must-ask-when-developing-web-site-86342> [Besucht am 20. Oktober 2019].

Jugendliche und Medien - Fakten und Zahlen. URL: <https://www.jugendundmedien.ch/digitale-medien/fakten-zahlen.html> [Besucht am 20. Oktober 2019].

Attention Span Statistics. URL: <https://www.statisticbrain.com/attention-span-statistics/> [Besucht am 06. Januar 2020].

MOBILE FIRST – NUR EIN TREND ODER SCHON BALD DIE ZUKUNFT? URL: <https://www.cubetech.ch/mobile-first-nur-ein-trend-oder-schon-bald-die-zukunft/> [Besucht am 21. Oktober 2019].

Mockups. URL: <https://www.igual.ch/de/internet-glossar/mockups> [Besucht am 21. Oktober 2019].

Django Architecture – 3 Major Components of MVC Pattern. URL: <https://data-flair.training/blogs/django-architecture/> [Besucht am 3. Januar 2020].

Django-Architektur. URL: <https://wagtailtagging.readthedocs.io/en/latest/architektur.html> [Besucht am 4. Januar 2020].

Django-Architektur Schnitt. URL: https://wagtailtagging.readthedocs.io/en/latest/_images/mtv_schnitt.png [Besucht am 4. Januar 2020].

Models | Django documentation. URL: <https://docs.djangoproject.com/en/3.0/topics/db/models/> [Besucht am 3. Januar 2020].

django-admin and manage.py | Django documentation. URL: <https://docs.djangoproject.com/en/3.0/ref/django-admin/> [Besucht am 3. Januar 2020].

Templates | Django documentation. URL: <https://docs.djangoproject.com/en/3.0/topics/templates/> [Besucht am 3. Januar 2020].

Writing views | Django documentation. URL: <https://docs.djangoproject.com/en/3.0/topics/http/views/> [Besucht am 3. Januar 2020].

Middleware | Django documentation. URL: <https://docs.djangoproject.com/en/3.0/topics/http/middleware/> [Besucht am 3. Januar 2020].

Testing in Django. URL: <https://docs.djangoproject.com/en/3.0/topics/testing/> [Besucht am 2. Dezember 2019].

Load test as code. URL: <https://gatling.io/> [Besucht am 28. Dezember 2019].

Jenkins vs Travis-CI: What is the difference? URL: <https://www.guru99.com/jenkins-vs-travis.html> [Besucht am 25. Oktober 2019].

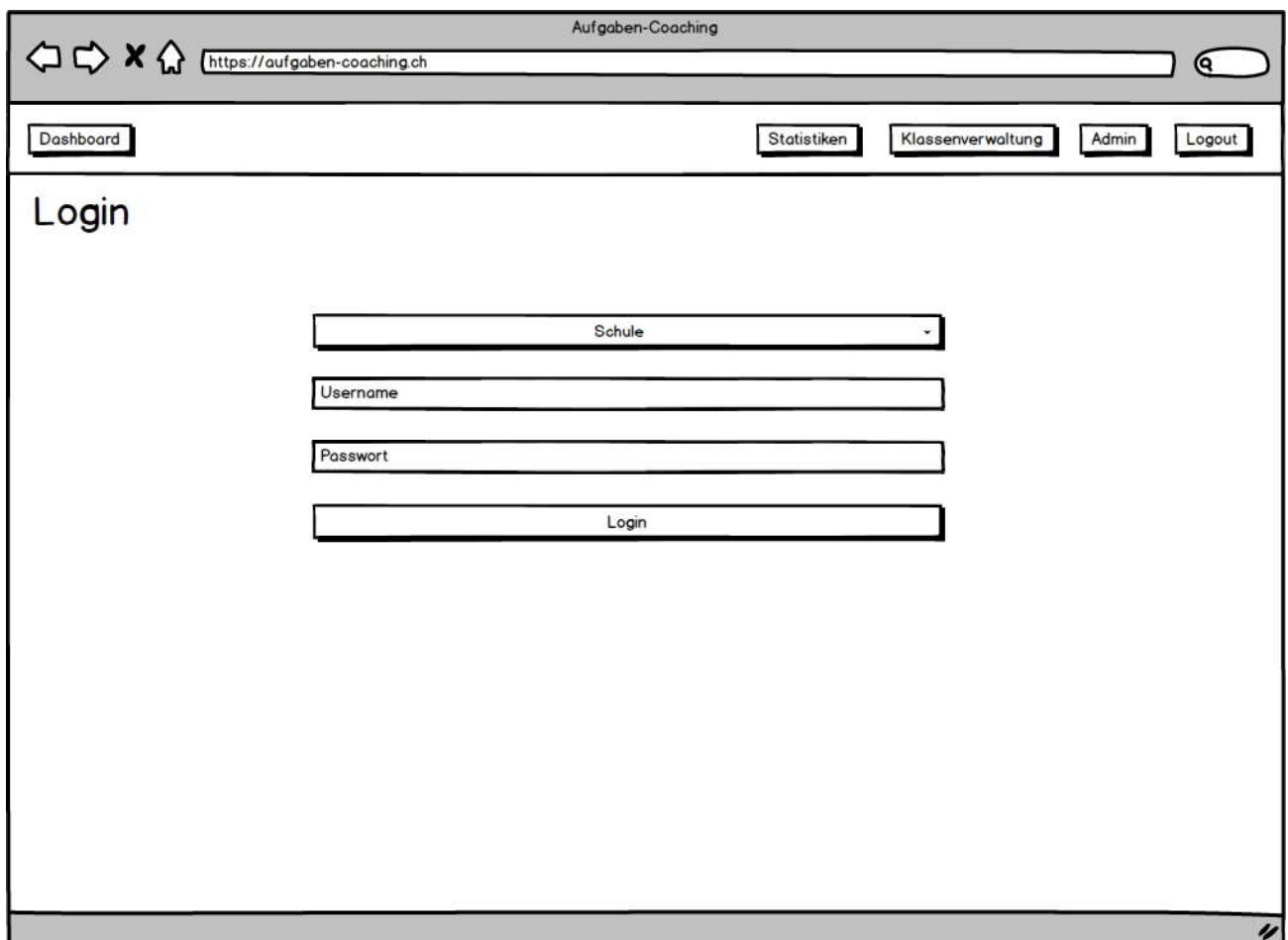
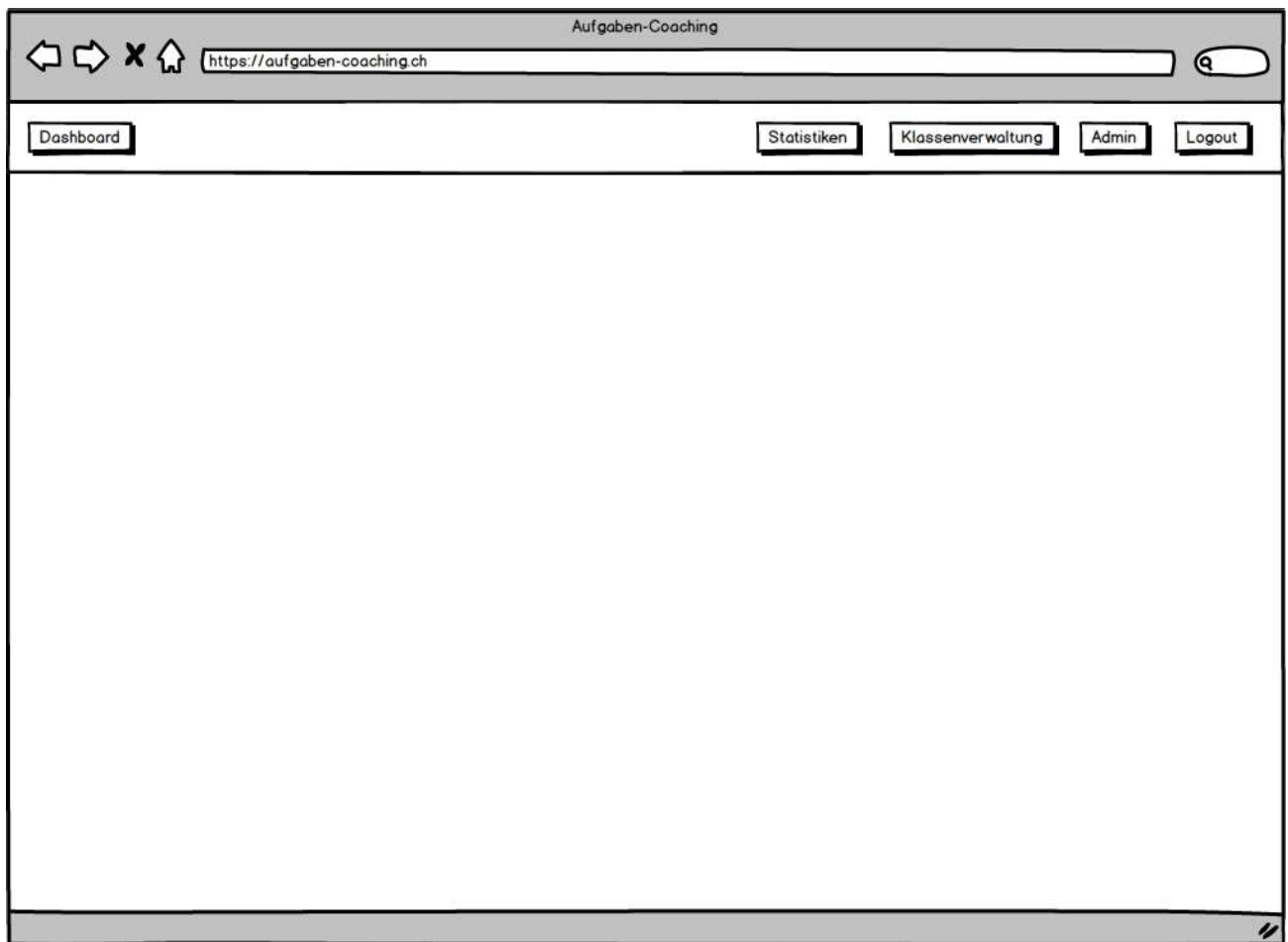
Count Lines of Code. URL: <https://github.com/AlDanial/cloc> [Besucht am 05. Januar 2020].

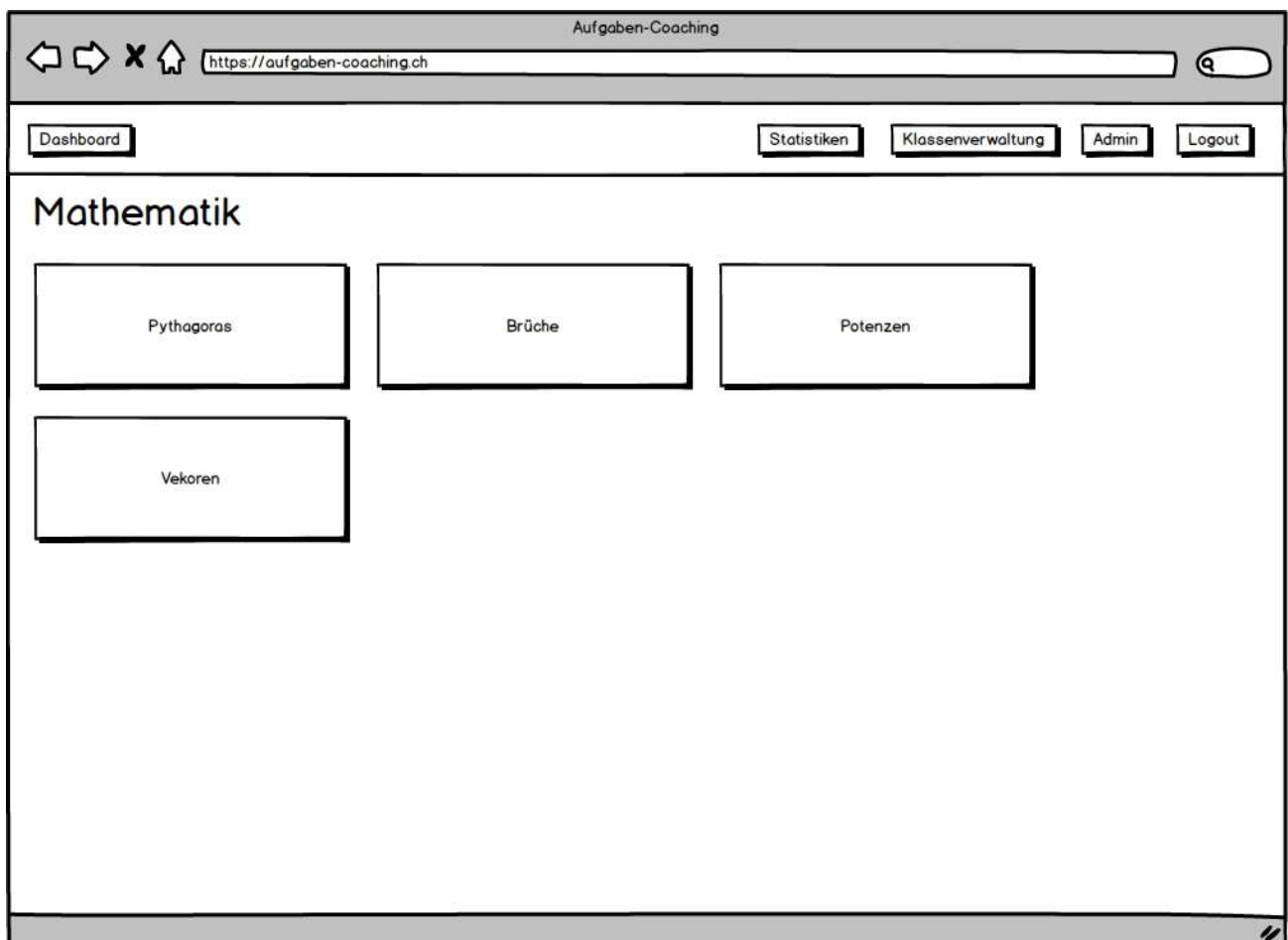
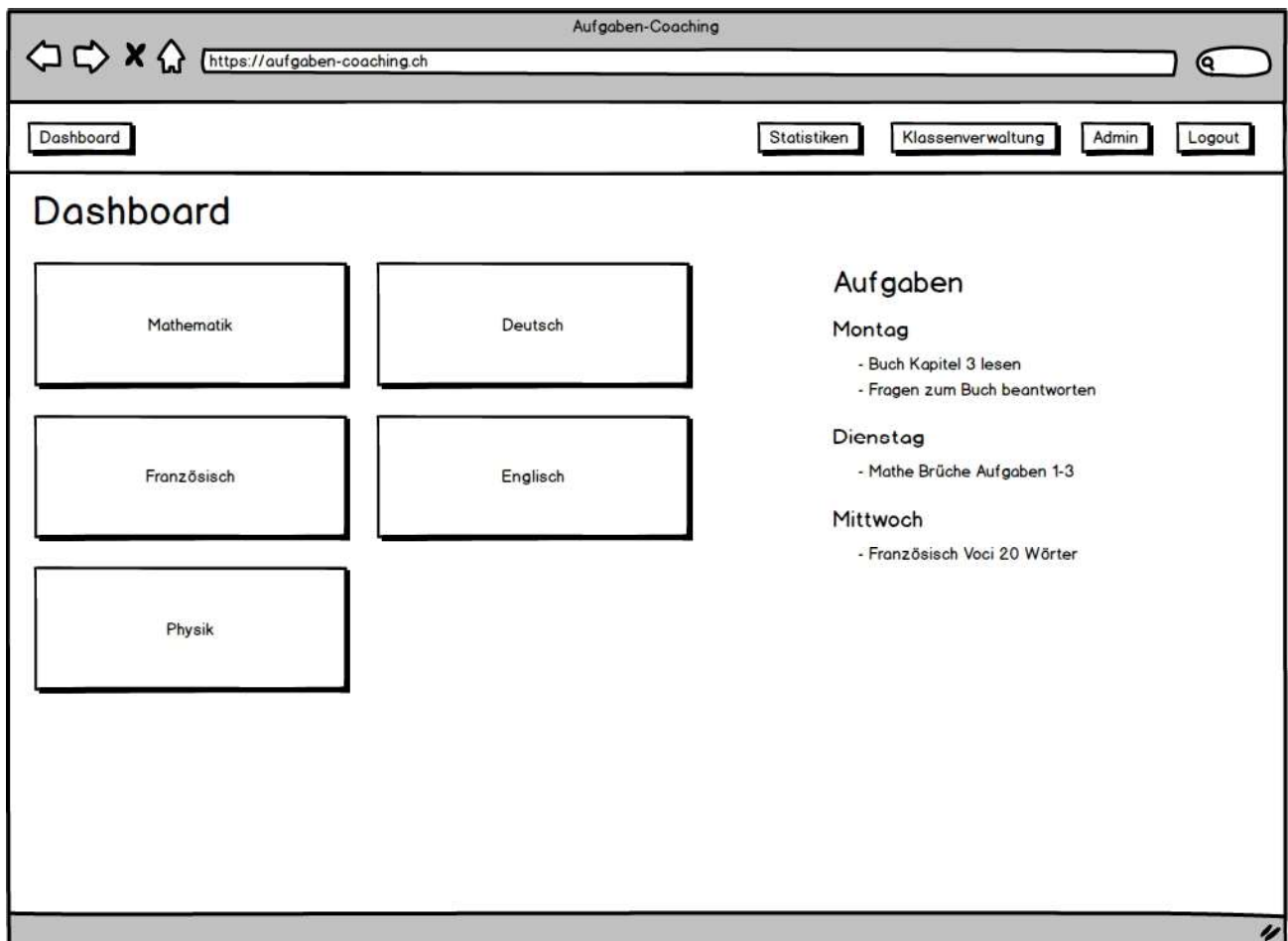
The Uncompromising Code Formatter. URL: <https://github.com/psf/black> [Besucht am 23. Dezember 2019].

What is Flake8 and why we should use it? URL: <https://medium.com/python-pandemonium/what-is-flake8-and-why-we-should-use-it-b89bd78073f2> [Besucht am 23. Dezember 2019].





Your Tool For Style Guide Enforcement. URL: <http://flake8.pycqa.org/en/latest/> [Besucht am 23. Dezember 2019].

A Mockups






Aufgaben-Coaching



https://aufgaben-coaching.ch



Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Mathematik > Brüche

Theorie

- Kapitel 1
- Kapitel 2
- Kapitel 3





Aufgaben einfach

- Aufgabe 1
- Aufgabe 2
- Aufgabe 3


Aufgaben mittel

- Aufgabe 1
- Aufgabe 2

Aufgaben-Coaching



https://aufgaben-coaching.ch



Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Mathematik > Brüche > Quiz 1

Was ergibt $\frac{2}{3} + \frac{1}{6}$?

☒ $\frac{5}{6}$

☐ 1

☐ $\frac{2}{3}$

Ist $\frac{5}{4}$ grösser als 1?

☐ Ja

☐ Nein

Speichern

Abbrechen

Aufgaben-Coaching
https://aufgaben-coaching.ch

Dashboard
Statistiken
Klassenverwaltung
Admin
Logout

Mathematik > Brüche > Theorie 1

Einführung ins Bruchrechnen

Hier werden die Grundlagen des Bruchrechnens behandelt. Es werden die verschiedenen Arten von Brüchen (eigentliche, uneigentliche, gemischte) und die Regeln für das Addieren, Subtrahieren, Multiplizieren und Dividieren von Brüchen erklärt. Die Aufgabenstellungen sind so gestaltet, dass sie das Verständnis der Regeln fördern und die Fähigkeit, Brüche zu vereinfachen und zu addieren/subtrahieren, zu trainieren.

Quiz 1
Quiz 2
Quiz 3

Aufgaben-Coaching
https://aufgaben-coaching.ch

Dashboard
Statistiken
Klassenverwaltung
Admin
Logout

Mathematik > Brüche > Aufgabe 1

Aufgabe 1

Fasse folgenden Bruch zusammen

$$\frac{1}{6} + \frac{1}{2}$$

Lösung:

Benötigst du Hilfe?

Forum

- Mach den Bruch gleichnamig

$\frac{1}{6} + \frac{3}{6}$
- Addiere die Brüche

$\frac{4}{6}$
- Kürze den Bruch

$\frac{2}{3}$

Aufgaben-Coaching
https://aufgaben-coaching.ch

Dashboard
Statistiken
Klassenverwaltung
Admin
Logout

Mathematik > Brüche > Aufgabe 1 > Forum

Luca, 12.10.2019 16:05

Ich verstehe nicht, wie man diesen Bruch kürzen kann.

Alessandro, 12.10.2019 16:37

Du musst den Bruch zuerst gleichnamig machen.

Meine Antwort

Senden
Abbrechen

Aufgaben-Coaching
https://aufgaben-coaching.ch

Dashboard
Statistiken
Klassenverwaltung
Admin
Logout

Klassenverwaltung

Klasse S2

Mathematik
Brüche

Aufgaben

Aufgabe 1
Aufgabe 2

Elemente per Drag & Drop an gewünschten Tag hinzufügen

Aufgabe erfassen

Quizzes

Quiz Basics
Quiz Fortgeschritten

Quizzes verwalten

Wochenplan

Montag

- M, Brüche, Aufgabe 2

Dienstag

- M, Pythagoras, Aufgabe 2

Mittwoch

Donnerstag

Freitag

Aufgaben-Coaching

https://aufgaben-coaching.ch

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Mathematik > Brüche > Aufgabe erfassen

Aufgabe erfassen

Mathematik

Physik

Deutsch

Brüche

Pythagoras

Titel

Aufgabenstellung erfassen

Lösung

☐ Aufgabe für andere Lehrpersonen freigeben

Speichern

Hilfestellungen erfassen

Hilfestellung 1

Hilfetext

Zwischenresultat

Hinzufügen

Abbrechen

Aufgaben-Coaching

https://aufgaben-coaching.ch

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Mathematik > Brüche > Quizzes verwalten

Quizzes

Quiz Basics

Quiz Fortgeschritten

Quiz erfassen

Aufgaben-Coaching
https://aufgaben-coaching.ch

Dashboard
Statistiken
Klassenverwaltung
Admin
Logout

Mathematik > Brüche > Quiz erfassen

Titel

zugewiesene Quizfragen

Frage 1

nicht zugewiesene Quizfragen

Frage 2

Frage 3

Quizfrage erfassen

Speichern
Abbrechen

Aufgaben-Coaching
https://aufgaben-coaching.ch

Dashboard
Statistiken
Klassenverwaltung
Admin
Logout

Mathematik > Brüche > Quizfrage erfassen

Titel

Frage
☐ sinlge choice
☐ multiple choice

Was ergibt $\frac{2}{3} + \frac{1}{6}$?

☐ Antwortmöglichkeit 1
☐ Antwortmöglichkeit 2

Hinzufügen

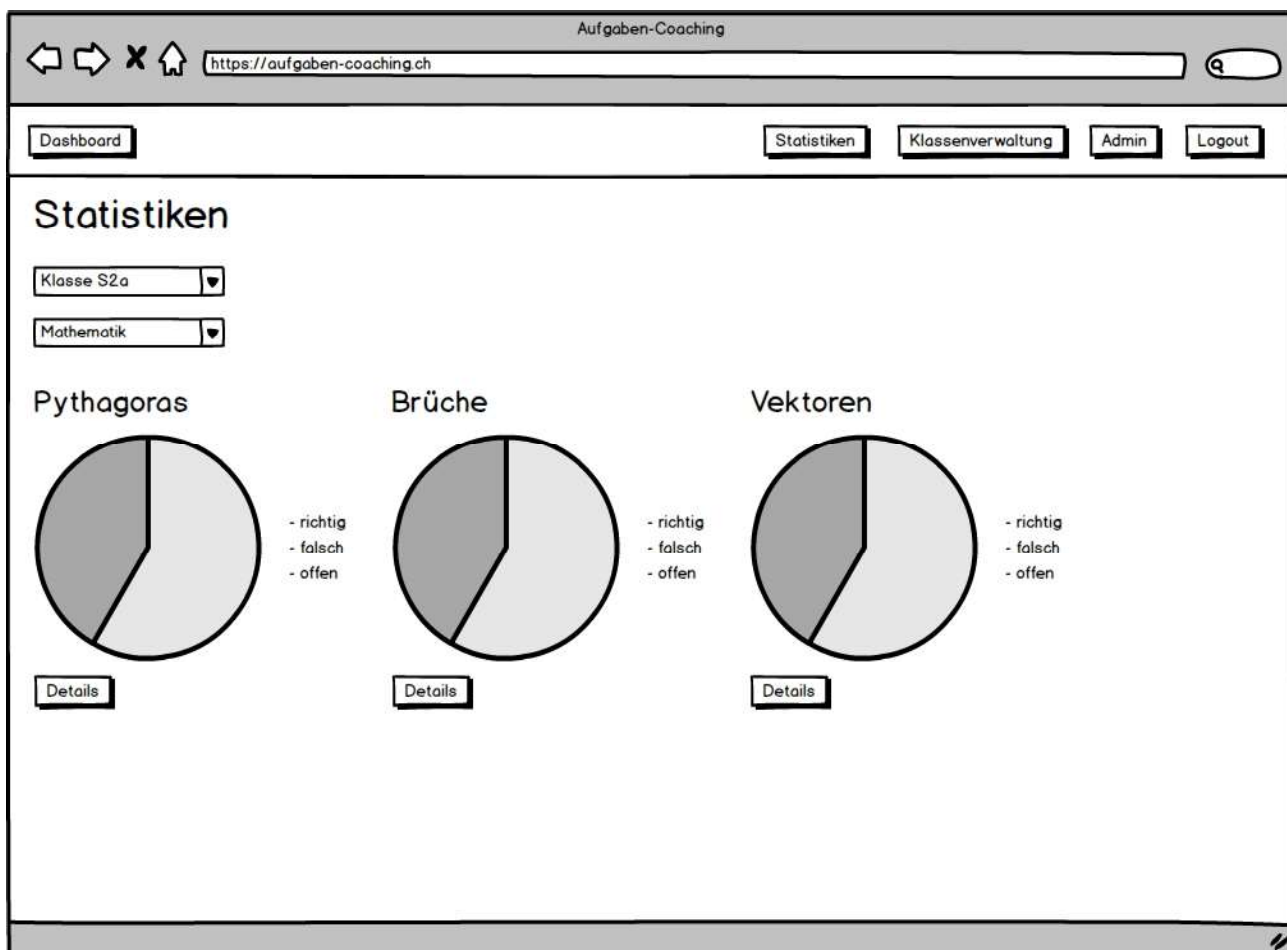
Frage
☒ sinlge choice
☐ multiple choice

Was ergibt $\frac{2}{3} + \frac{1}{6}$?

☐ Antwortmöglichkeit 1
☐ Antwortmöglichkeit 2

Hinzufügen

Speichern
Abbrechen



Aufgaben-Coaching

https://aufgaben-coaching.ch

Dashboard Statistiken Klassenverwaltung Admin Logout

Statistiken > Brüche

Schüler	Quiz 1	Quiz 2
Muster Philipp	70%	100%
Muster Carmen	100%	90%

Details Details

Aufgaben-Coaching

← → ✕ 🏠

https://aufgaben-coaching.ch

🔍

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Statistiken > Brüche > Quiz 1

Schüler	Frage 1	Frage 2	Frage 3	Frage 4
Muster Philipp	100%	75%	100%	0%
Muster Carmen	100%	25%	100%	66%

Aufgaben-Coaching

← → ✕ 🏠

https://aufgaben-coaching.ch

🔍

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Admin

Klassen

S2a

✕

S3a

✕

S2b

✕

nicht zugewiesene Schüler

19763 Bonoma, Alessandro

✕

81520 Gubler, Luca

✕

Lehrer

92154 Knobel, Marcel

✕

91373 Noser, Emma

✕

15383 Frey, André

✕

+ Neue Klasse

+ Neuer Schüler

+ Neuer Lehrer

Aufgaben-Coaching

←

→

✕

🏠

https://aufgaben-coaching.ch

🔍

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Admin > Klassenverwaltung

Klasse S2a

Name kann geändert werden

Lehrer

91373 Noser, Emma

Schüler

61592 Muster, Philipp

46230 Muster, Carmen

Lehrer

92154 Knobel, Marcel

15383 Frey, André

nicht zugewiesene Schüler

19763 Bonomo, Alessandro

81520 Gubler, Luca

Durch einen einfachen Klick auf die Schüler oder Lehrer können diese der Klasse zugewiesen oder entwiesen werden

Speichern

Abbrechen

Aufgaben-Coaching

←

→

✕

🏠

https://aufgaben-coaching.ch

🔍

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Admin > Klasse erstellen

Eintrittsjahr

11 /02/ 2008

📅

Klassenname

Speichern

Abbrechen

Aufgaben-Coaching

https://aufgaben-coaching.ch

Dashboard

Statistiken

Klassenverwaltung

Admin

Logout

Admin > Person erfassen

Eintrittsjahr

11 /02/ 2008

Schüler

Lehrer

Vorname

Nachname

Speichern

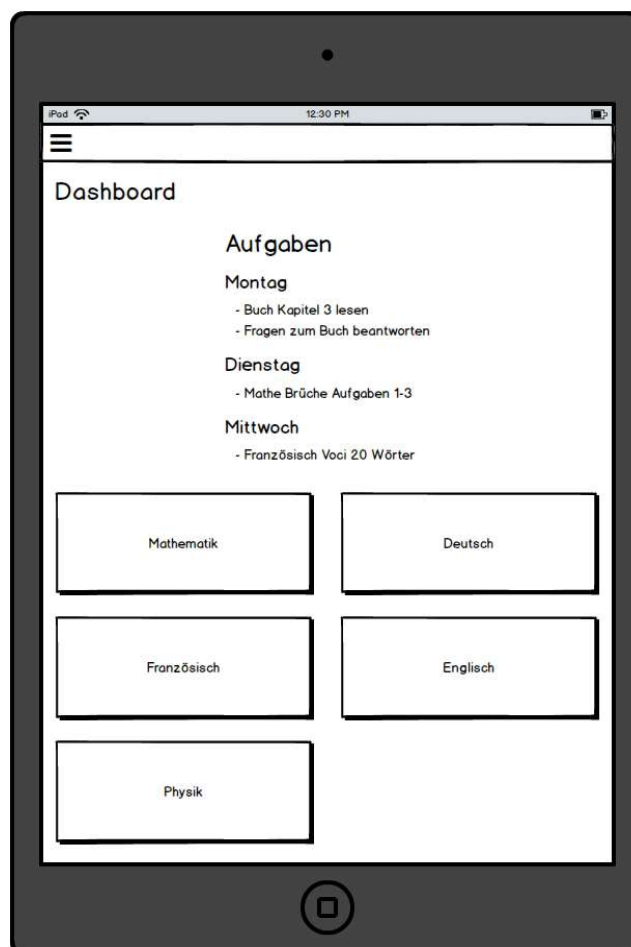
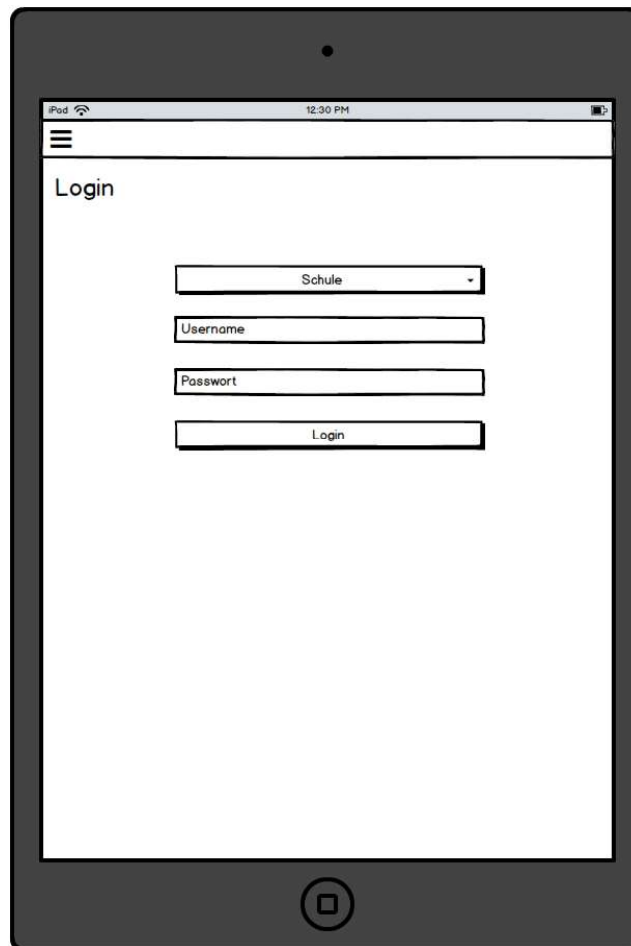
Abbrechen

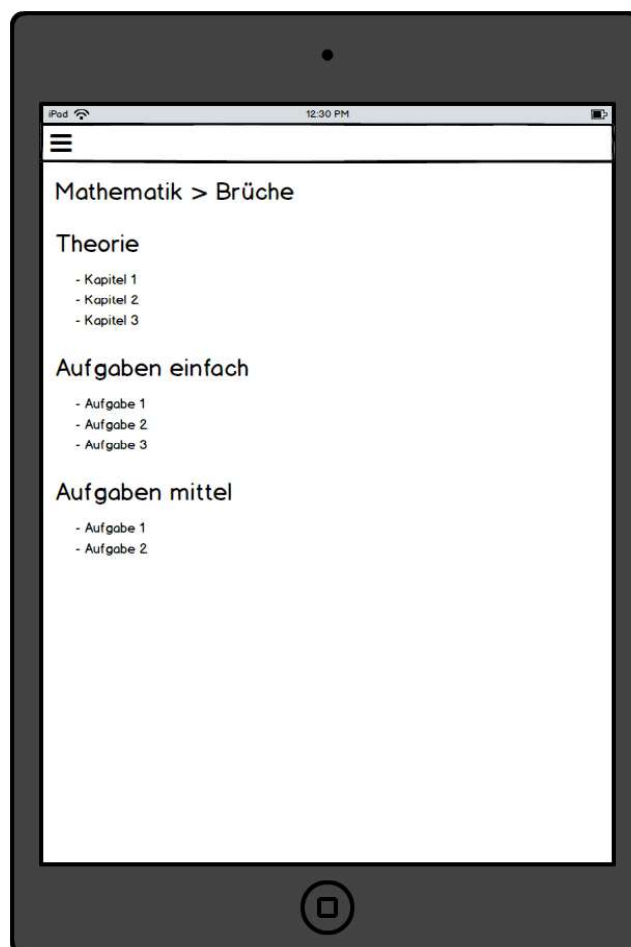
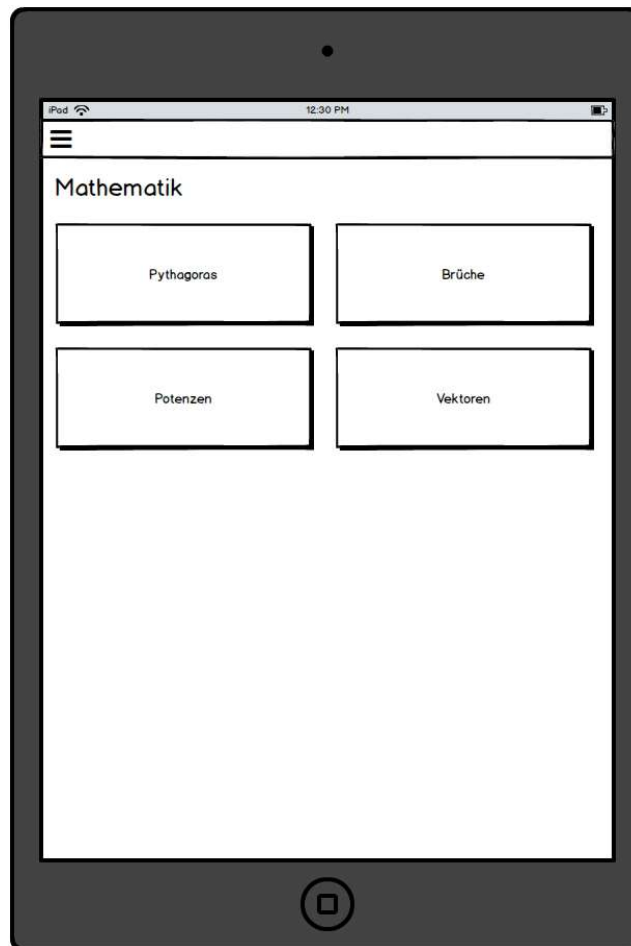
Statistiken

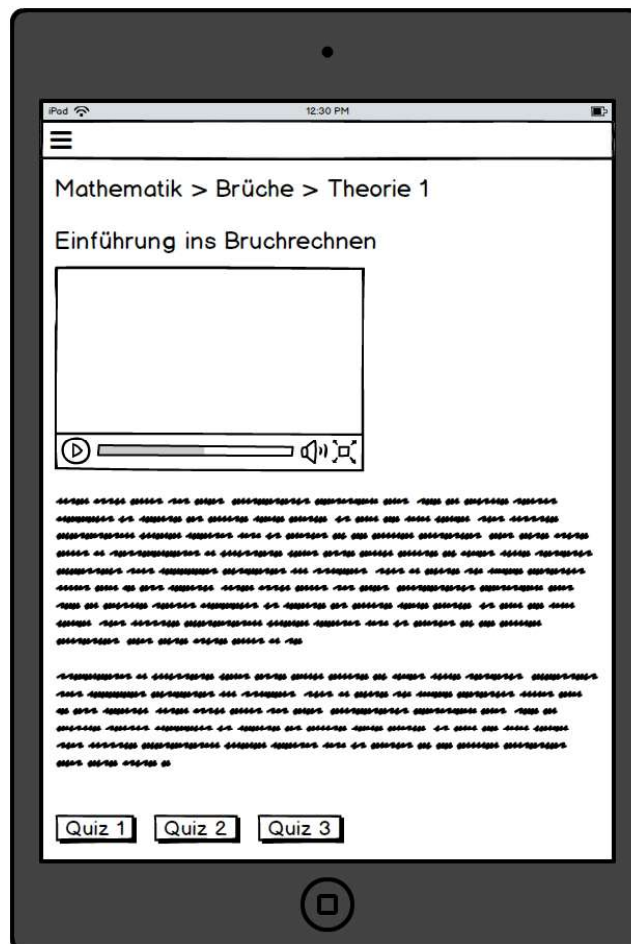
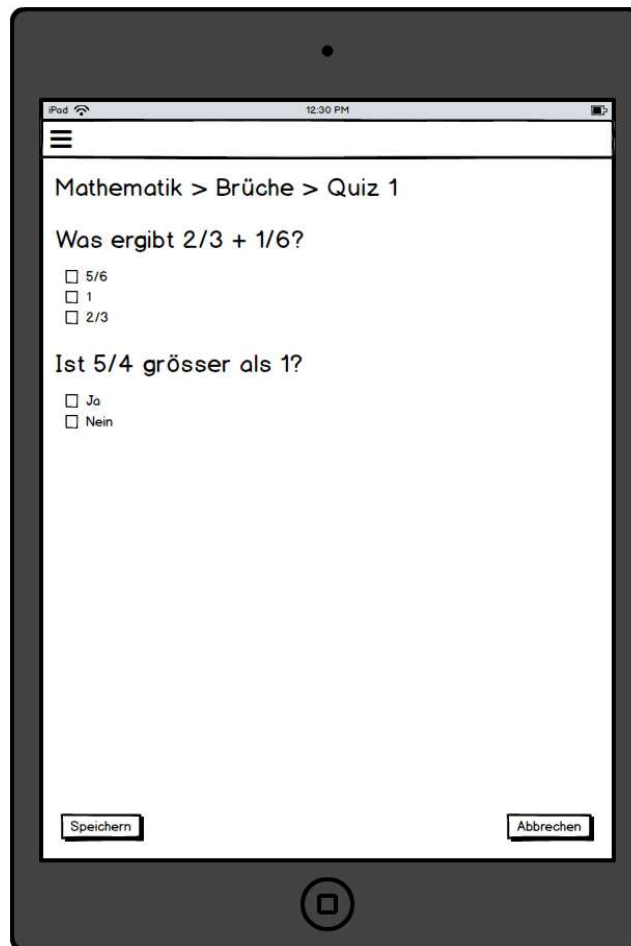
Klassenverwaltung

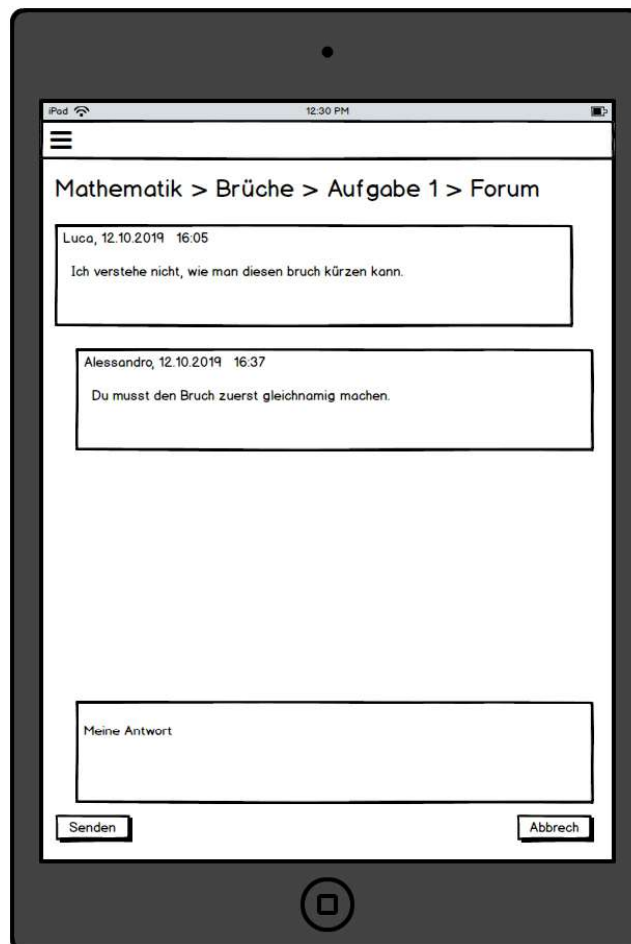
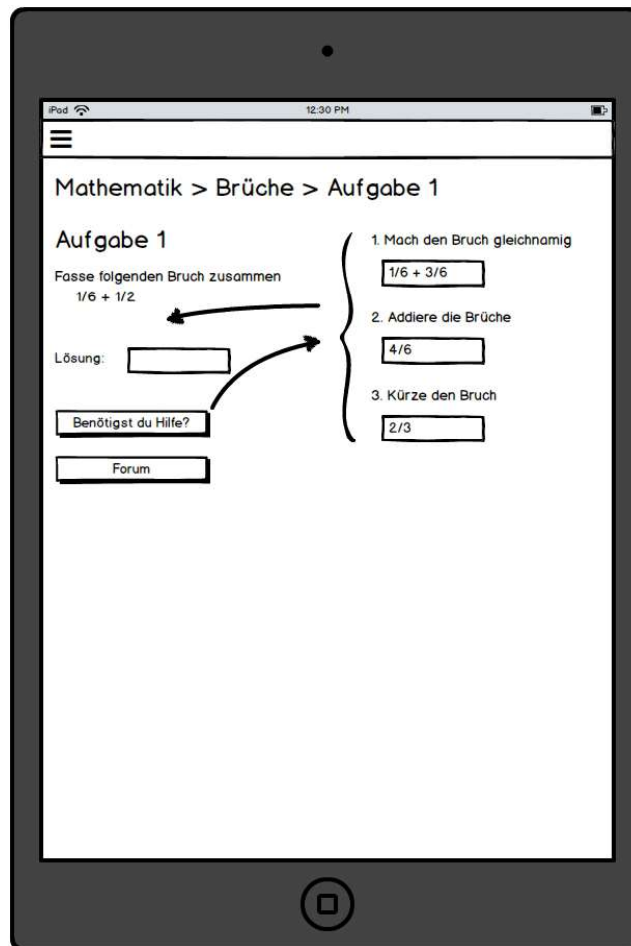
Admin

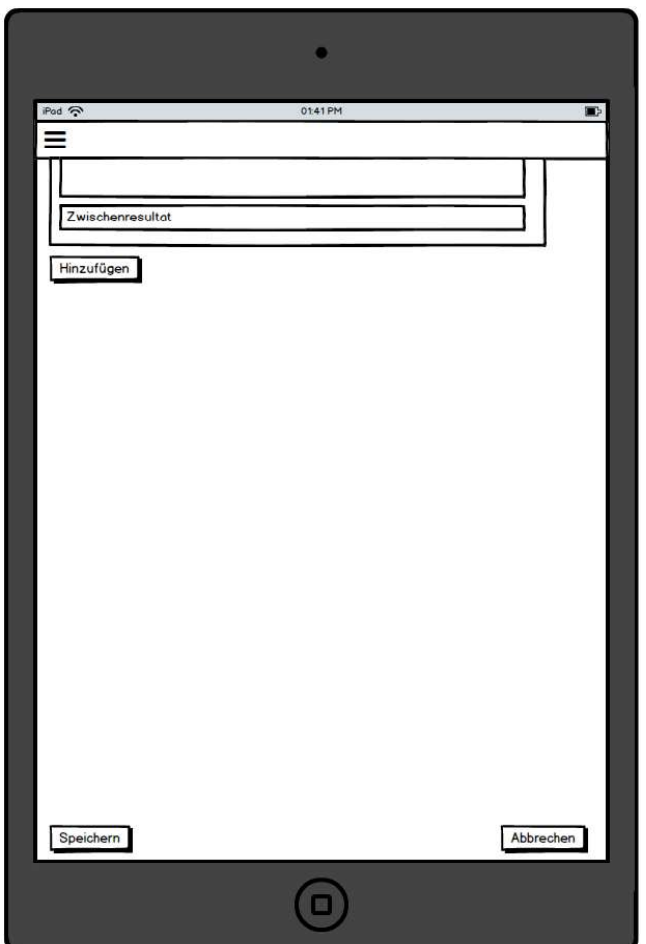
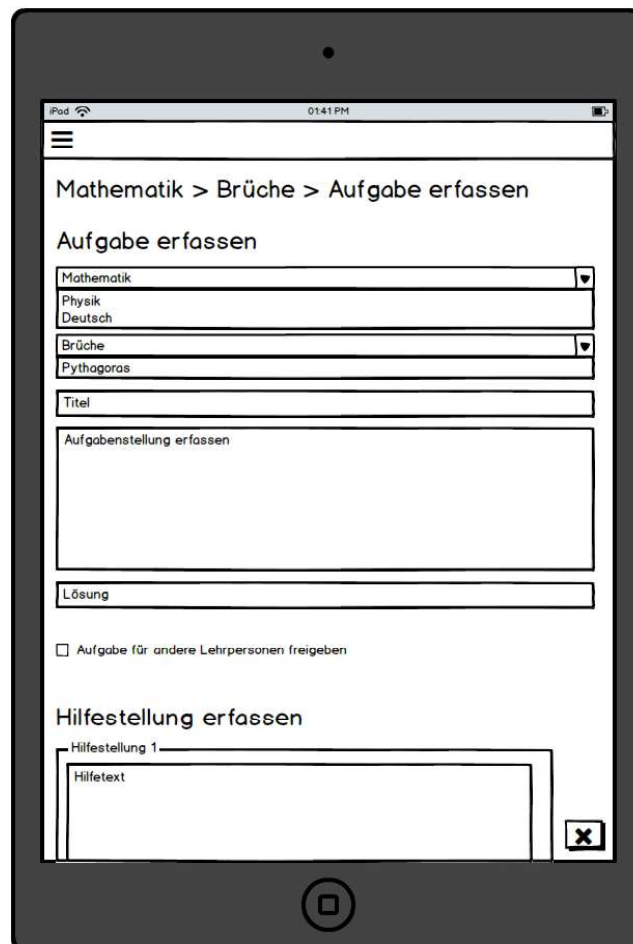
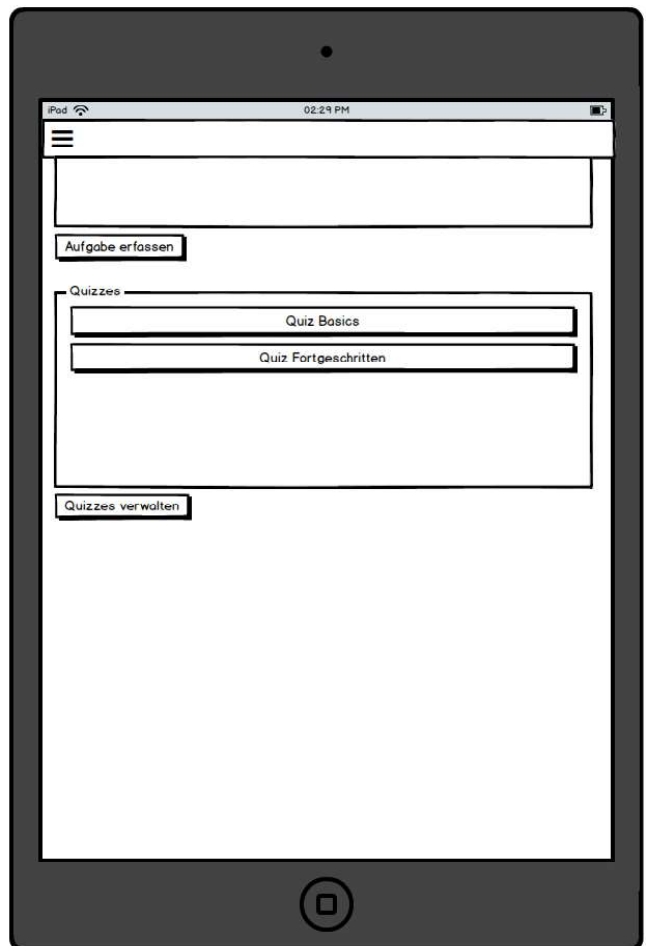
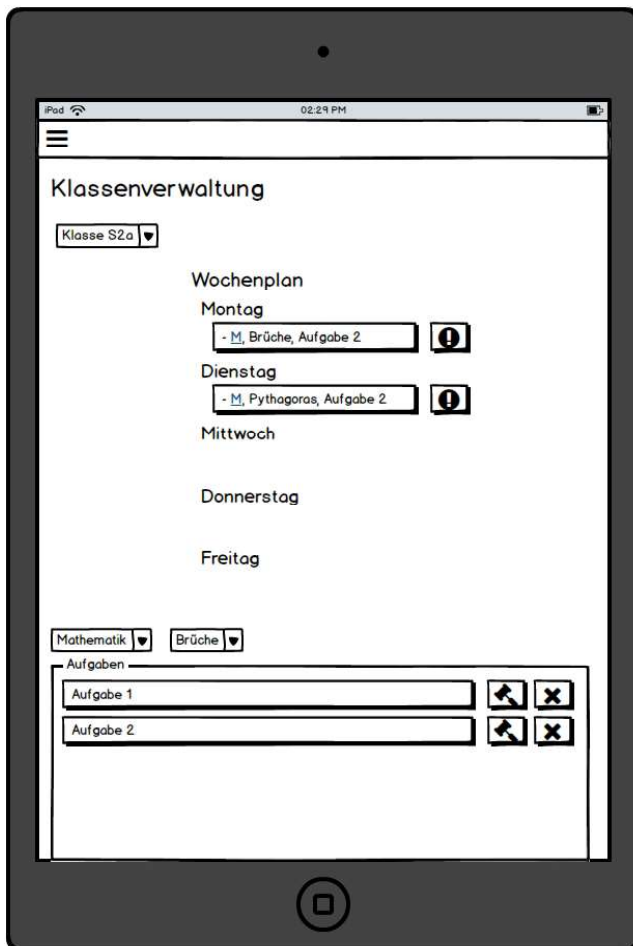
Logout

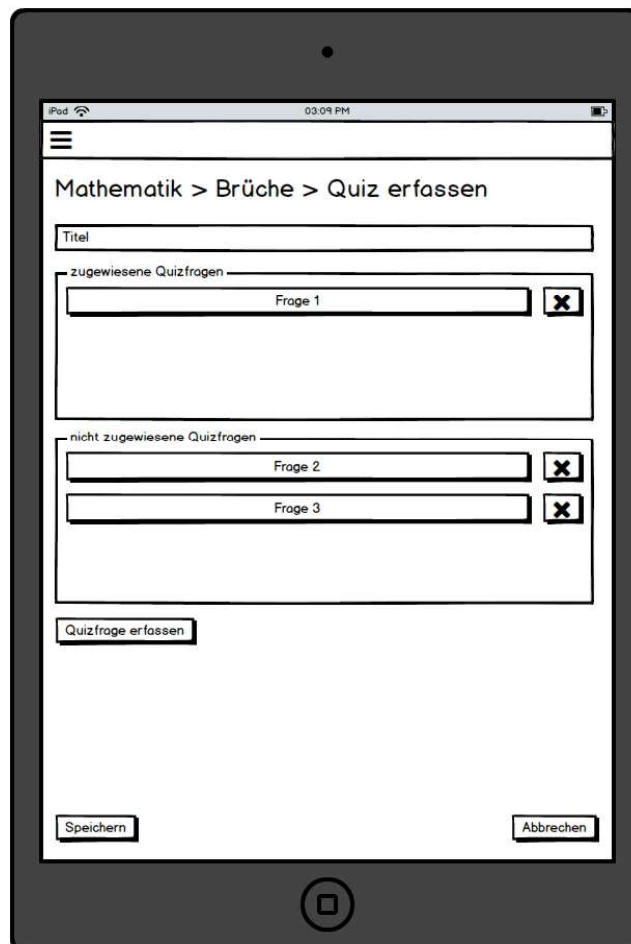
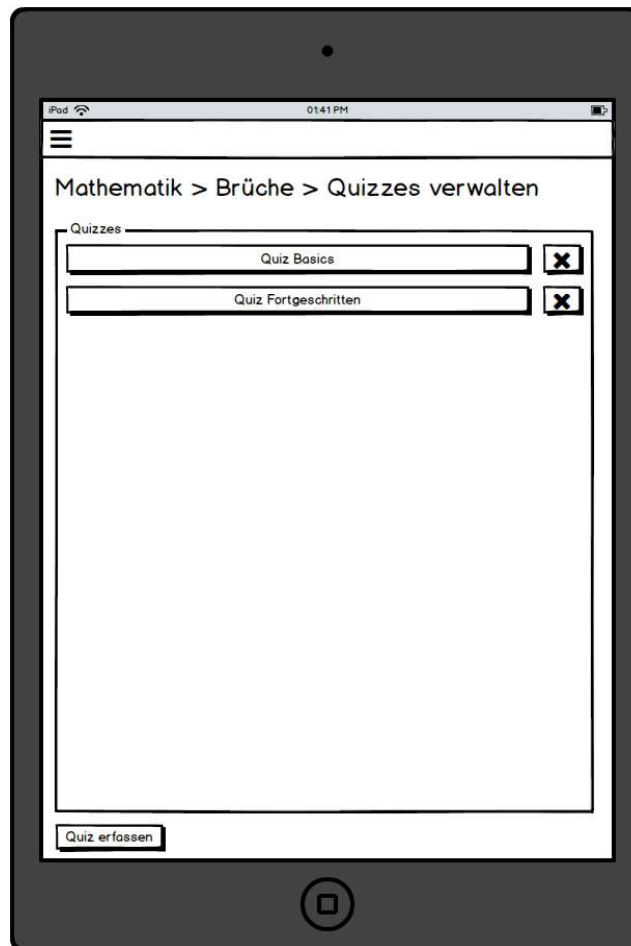


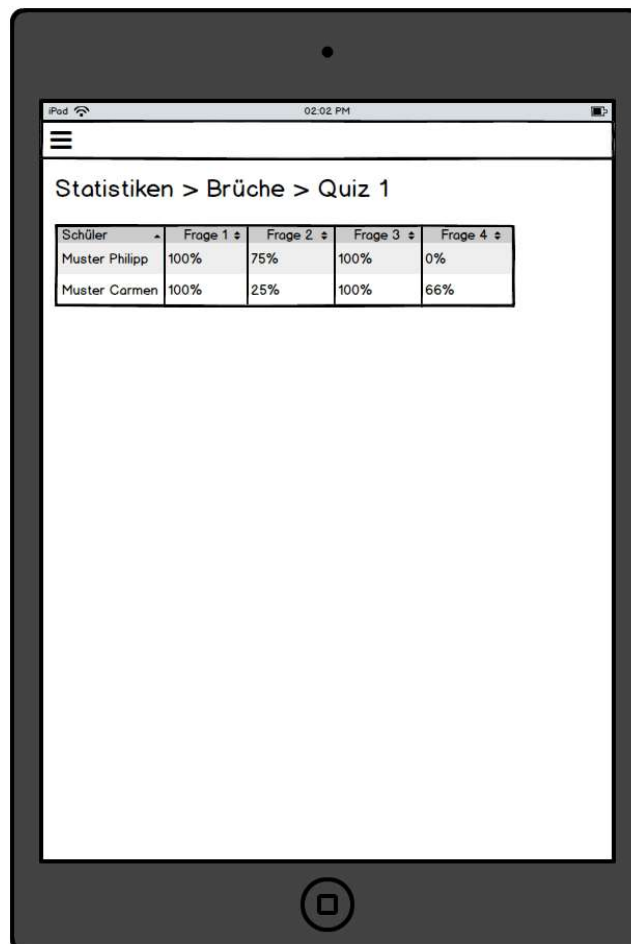
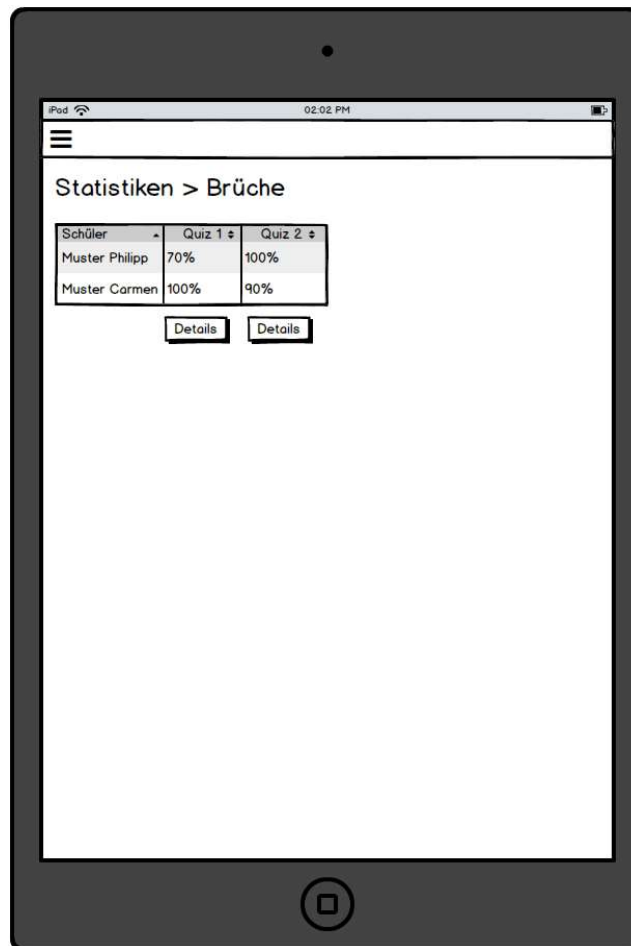


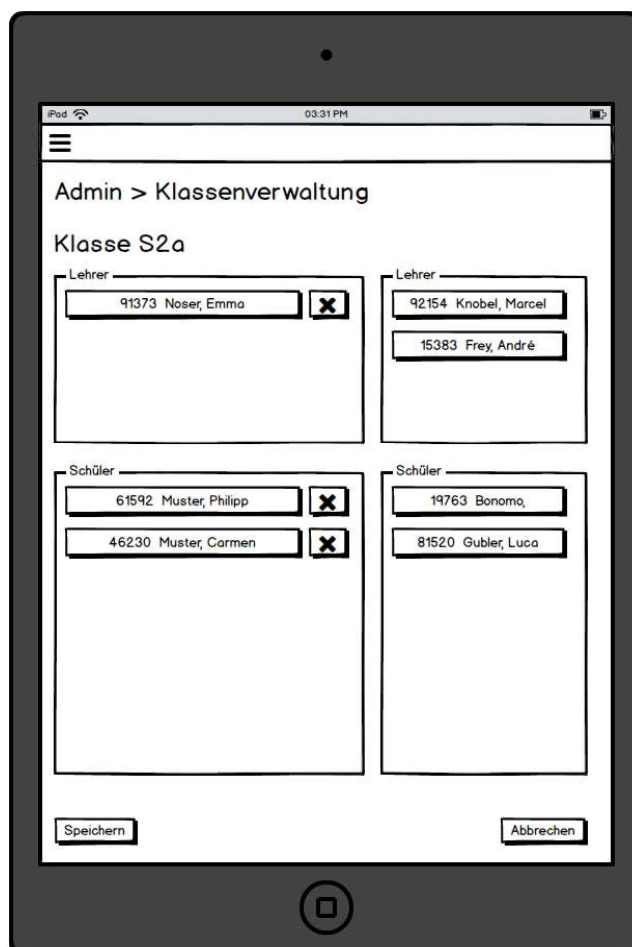
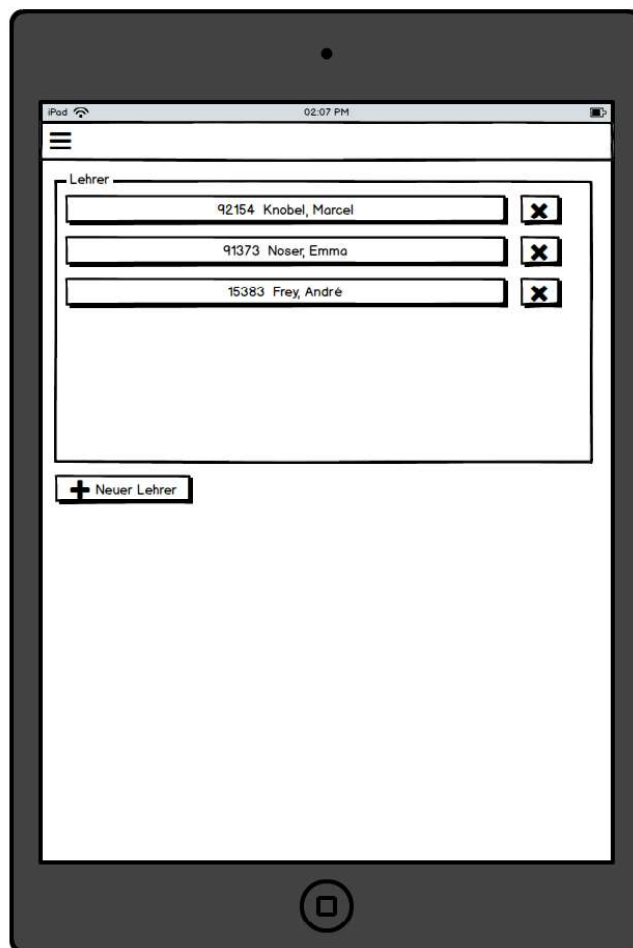
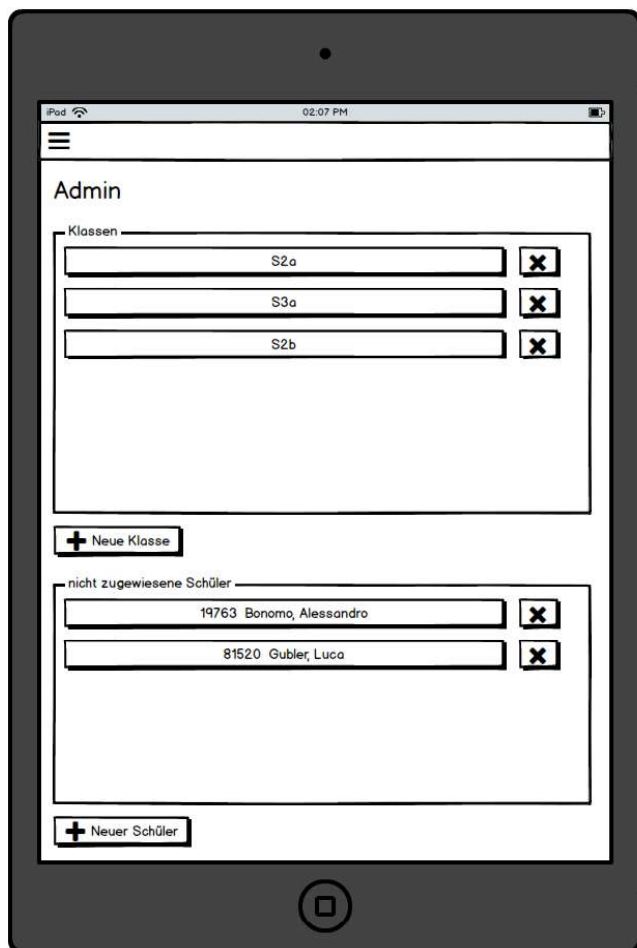














iPad 03:31 PM


Admin > Klasse erstellen

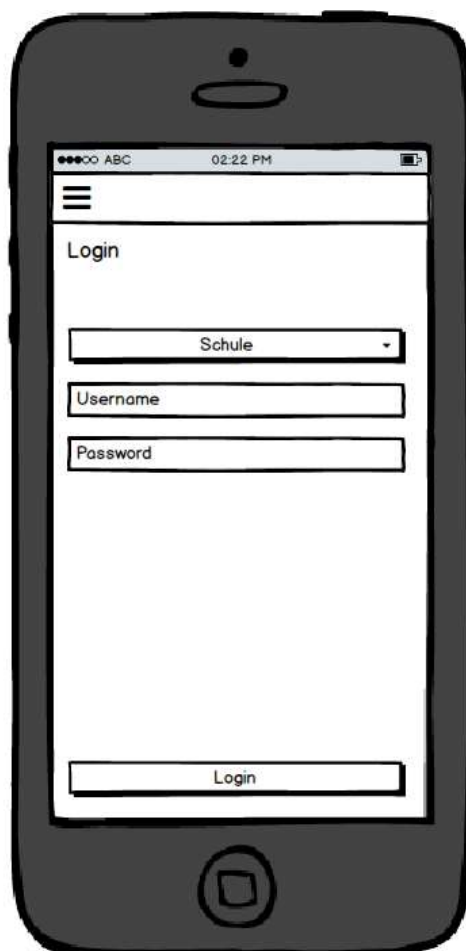
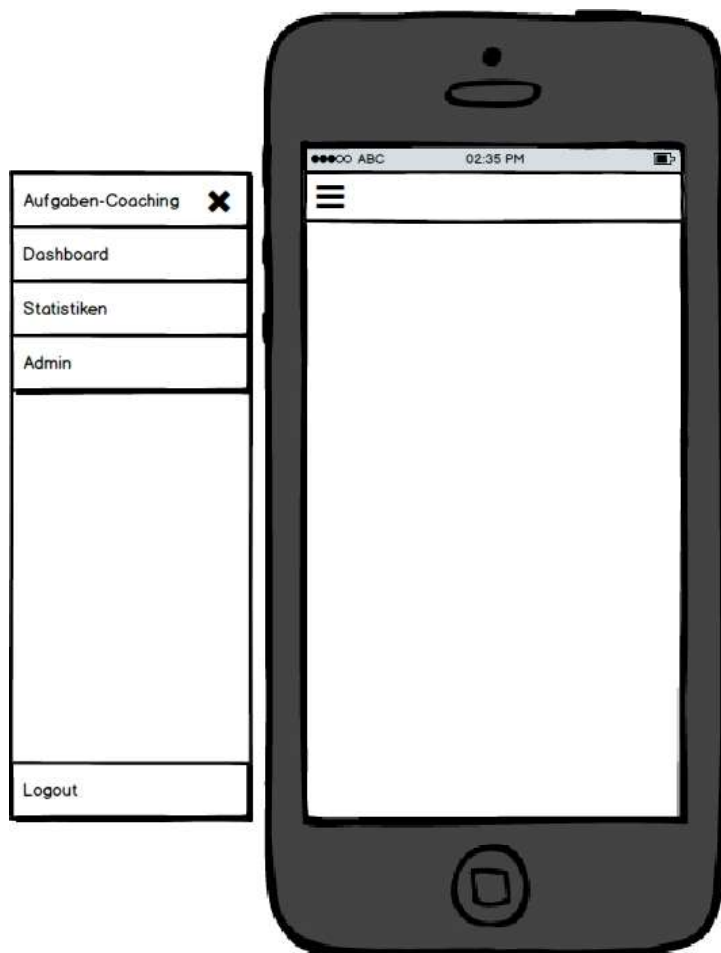
Eintrittsjahr 

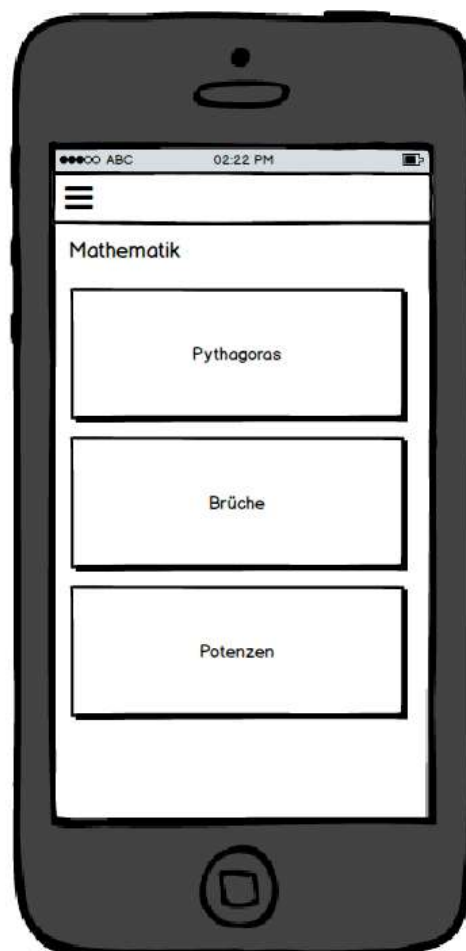
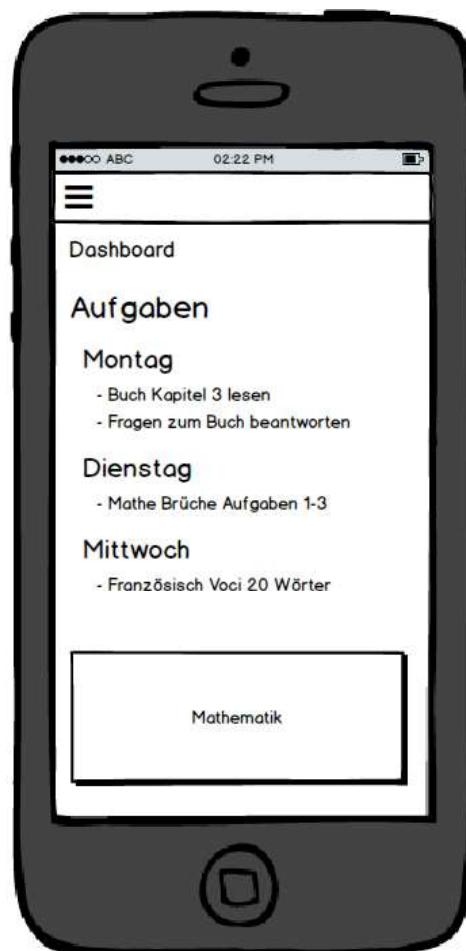
iPad 03:31 PM

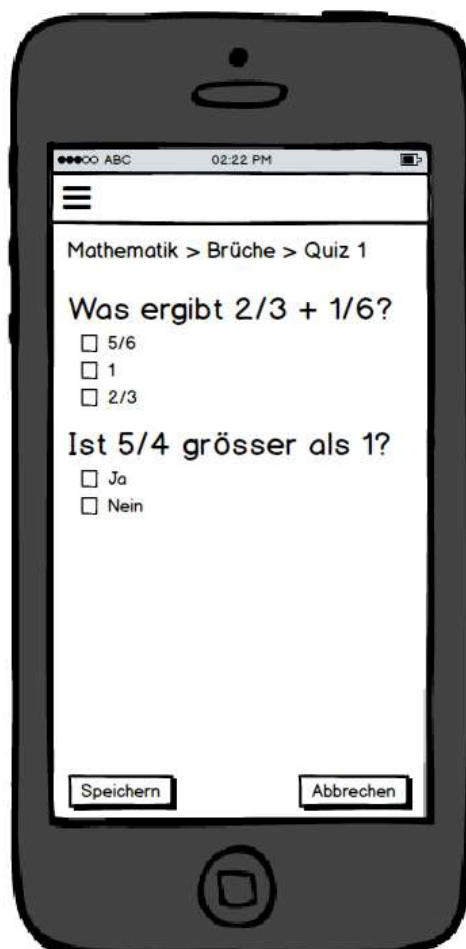
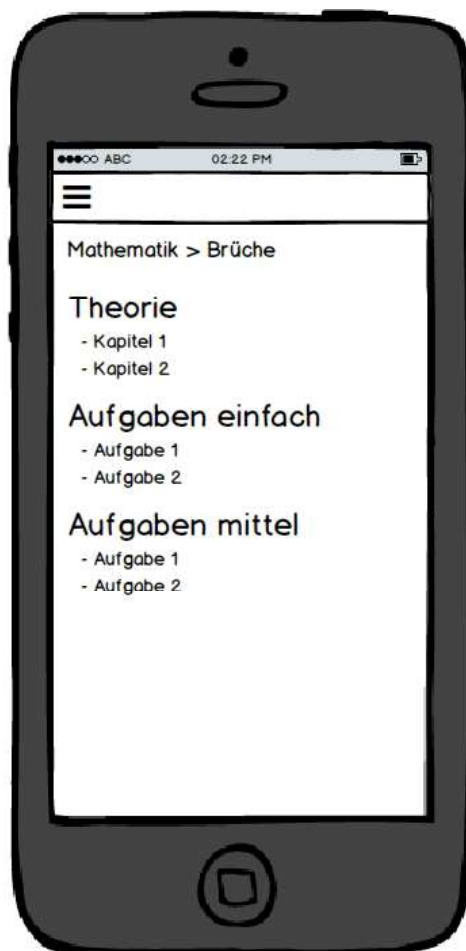
Admin > Person erfassen

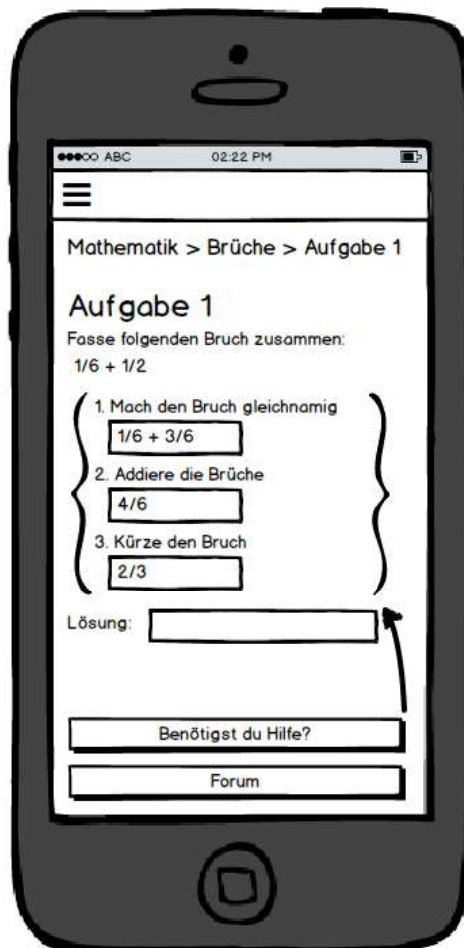
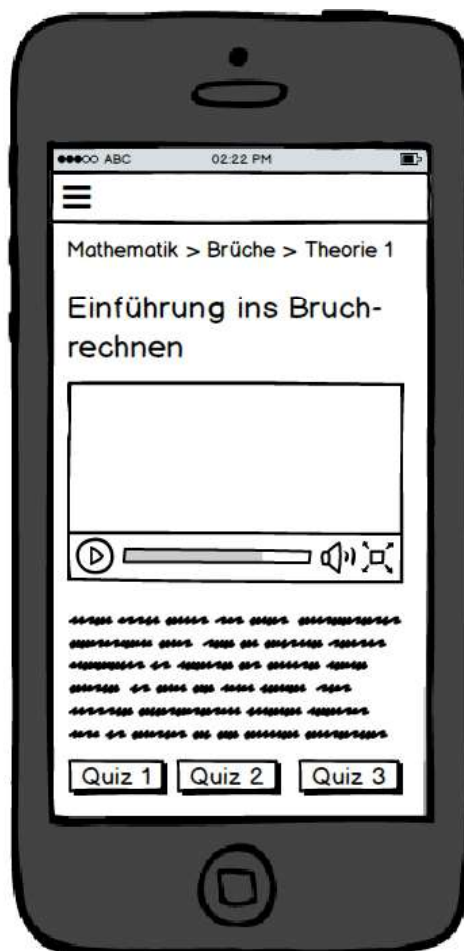
Eintrittsjahr 

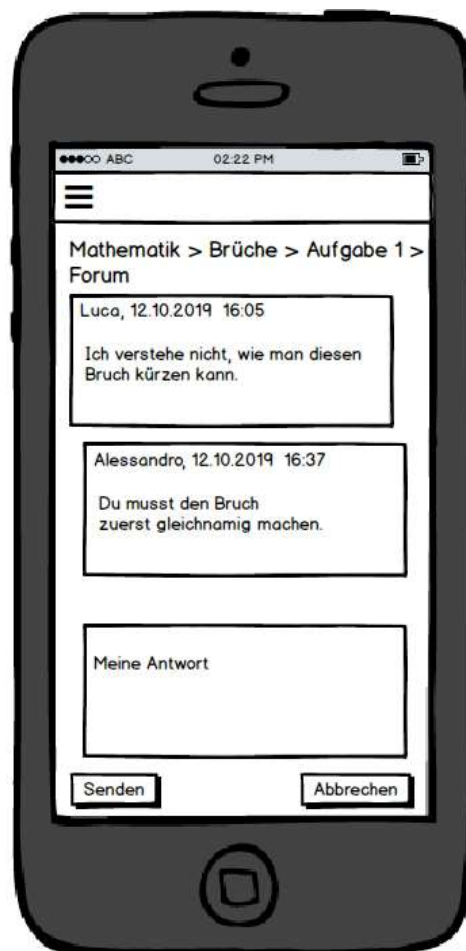


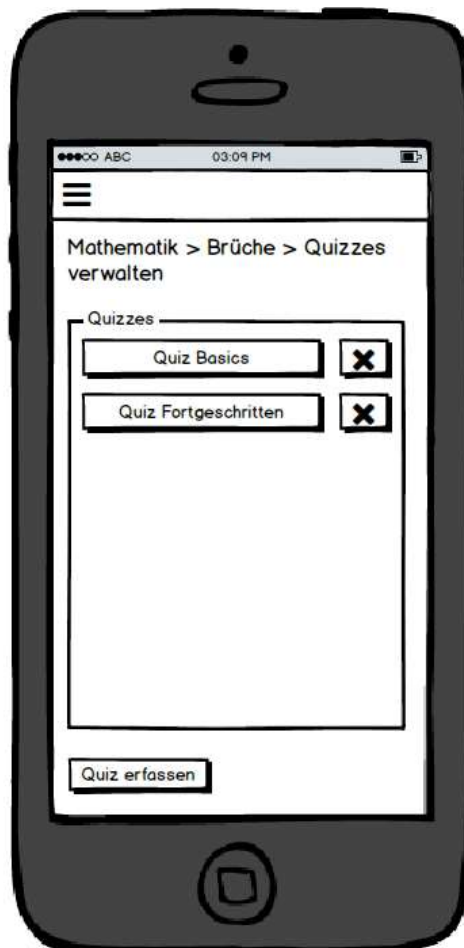
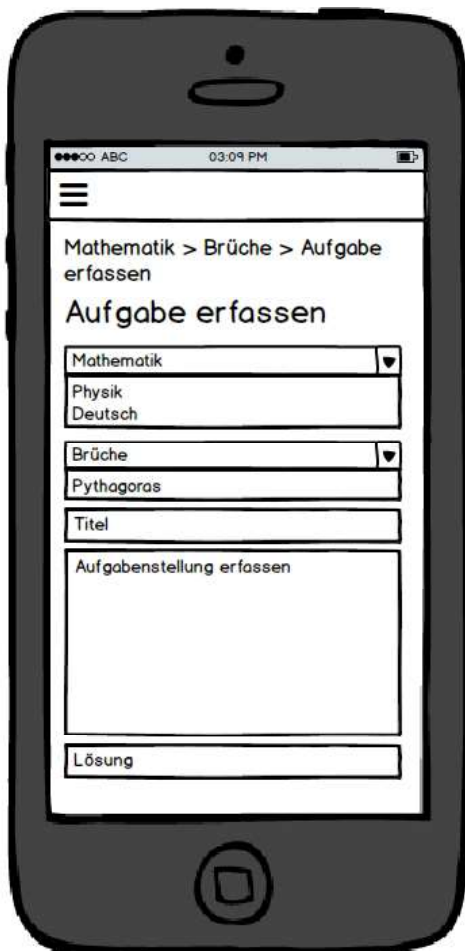


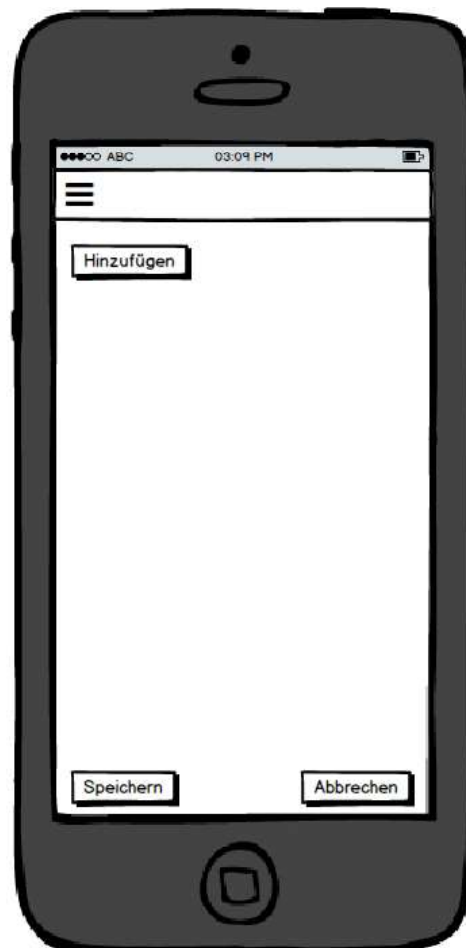
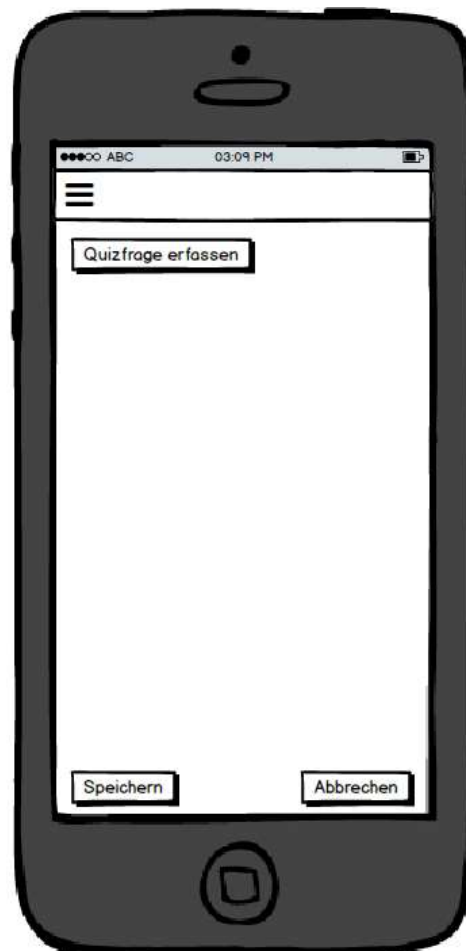
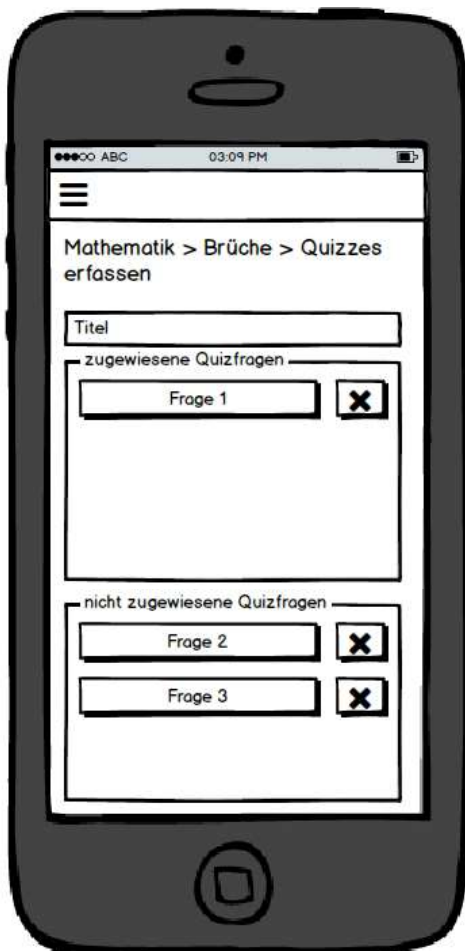


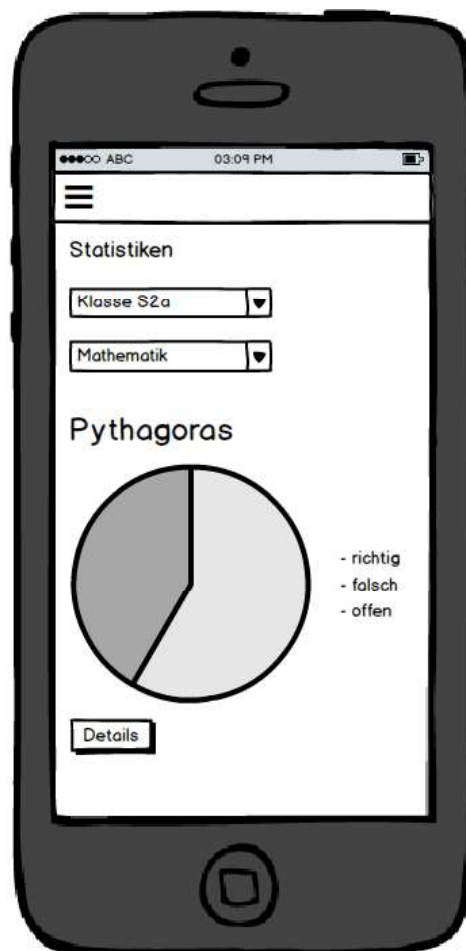












The screen displays a mobile application interface. At the top, there is a status bar with signal strength, 'ABC', time '03:09 PM', and battery level. Below is a hamburger menu icon. The main content area is titled 'Statistiken > Brüche'. It contains a table with three columns: 'Schüler', 'Quiz 1', and 'Quiz 2'. Below the table are two 'Details' buttons, one under each quiz column.

Schüler	Quiz 1	Quiz 2
Muster Philipp	70%	100%
Muster Carmen	100%	90%

