IFS INSTITUTE FOR SOFTWARE

OST Ostschweizer Fachhochschule

City Trip Planner: Kurztrip-Planer für Fussgänger

Bachelorarbeit

Studiengang Informatik

OST – Ostschweizer Fachhochschule

Campus Rapperswil-Jona

Herbstsemester 2022

Autoren	Lukas Grigis
	Jan Ruch
Betreuer	Prof. Stefan Keller
Projektpartner	Schweiz Tourismus
	Estefan Justo
Experte	Claude Eisenhut
Gegenleserin	Mitra Purandare

Abstract

Die vorliegende Arbeit befasst sich mit der Thematik der Wegfindung für Fussgänger:innen in der Schweiz. Es wurde eine Applikation entworfen, welche Rundgänge und Einwegtrips unter Einbezug von kategorisierten Interessen in Schweizer Städten erstellen kann. Bei der Berechnung werden persönliche Präferenzen wie Distanz oder Dauer berücksichtigt, sodass ein individuelles Erlebnis geschaffen werden kann. In Zusammenarbeit mit Schweiz Tourismus wurden touristische Zwecke bei der Entwicklung berücksichtigt, damit auch Tourist:innen aus dem In- und Ausland die Anwendung nutzen können.

Als Ausgangslage dient die gleichnamige Studienarbeit. Die darin erworbenen Erkenntnisse sollen in dieser Bachelorarbeit vertieft und genutzt werden, um daraus eine weitere Iteration eines Minimal Viable Product zu schaffen. Dabei steht der Ausbau der Funktionalität und die Verbesserung der Benutzerfreundlichkeit im Vordergrund. Die Applikation City Trip Planner ermöglicht es, den Benutzer:innen individuelle Routen mit kategorisierten Interessen im städtisch-urbanen Raum zu planen. Sie kann sowohl am Computer wie auch am Smartphone genutzt werden. Es handelt sich um eine Webapplikation, welche ohne Installation zusätzlicher Software im Browser läuft. Um Tourist:innen einen zusätzlichen Mehrwert zu bieten und die Erkundung spannender zu gestalten, werden Hintergrundinformationen zu einzelnen Stationen auf der Route gesammelt und dargestellt.

Die Arbeit vereint agile Projektplanung, moderne Entwicklungsansätze, bekannte Entwurfsmuster und aktuelle Technologien. Folgende Technologien sind elementar wichtig für die Umsetzung des Projekts: Python, FastAPI, TypeScript, Angular, PostgreSQL, Keycloak, OpenStreetMap, OSRM, GraphHopper und openrouteservice.

Management Summary

Ausgangssituation

Die Planung von optimalen Routen findet in vielen Bereichen Anwendung. Dazu existieren zahlreiche Dienste von teils namhaften Anbietern, welche spezialisierte Lösungen erfolgreich zur Verfügung stellen. Dies sind vor allem Dienste, um individuelle Routen für motorisierte Fahrzeuge zu planen. Gleichartige Dienste für die Planung von Routen ohne Motorunterstützung gibt es vergleichsweise wenig. Mit diesem Thema setzt sich die vorliegende Arbeit auseinander.

Eine Aufarbeitung von existierenden Produkten hat ergeben, dass es bislang keine Softwarelösung auf dem Markt gibt, die es Benutzer:innen ermöglicht, eine Route mit kategorisierten Interessen im städtisch-urbanen Raum zu planen. Der City Trip Planner soll diese Lücke schliessen und Tourist:innen aus dem In- und Ausland eine Plattform zur Planung solcher Fussgängerrouten bieten. Diese Arbeit basiert auf der gleichnamigen Studienarbeit und erweitert den bestehenden Funktionsumfang. Ein besonderer Fokus liegt dabei auf der Funktionserweiterung für Einwegtrips, der Anbindung weiterer Routing Engines und der Einführung eines Benutzerprofils. Das Ziel der Bachelorarbeit ist es, die Erkenntnisse aus der Studienarbeit umzusetzen und in Zusammenarbeit mit Schweiz Tourismus ein weiteres Minimal Viable Product einer responsiven Webapp zu erstellen. Damit soll den Benutzer:innen ein individuelles Erlebnis geboten werden, welches auf ihre persönlichen Bedürfnisse zugeschnitten ist.

Ergebnissituation

Das ursprüngliche Minimal Viable Product konnte in der neuen Iteration überarbeitet und ausgebaut werden. Die erwarteten funktionalen Anforderungen der Aufgabenstellung konnten vollumfänglich implementiert werden. Weitere Anforderungen aus der Tourismusbranche, insbesondere Verbesserungen der Benutzerfreundlichkeit, flossen in die Arbeit ein und konnten zur Zufriedenheit unseres Industriepartners umgesetzt werden.

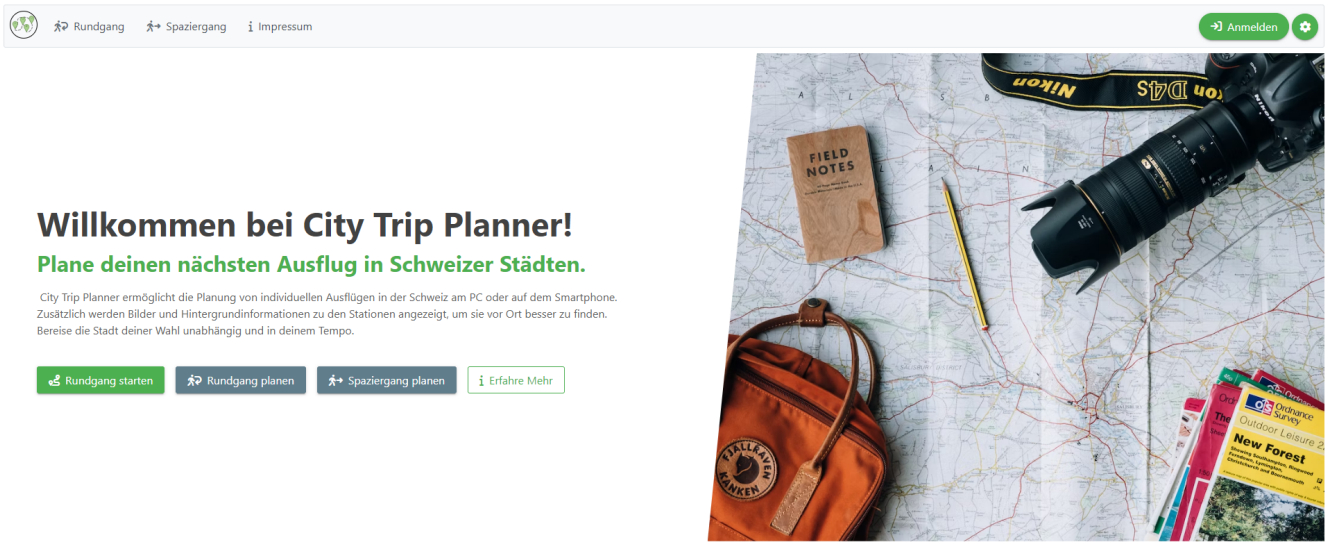

Abbildung 1. Startseite der Webapplikation City Trip Planner mit Wahl Rundgang oder Spaziergang

Die in der vorhergehenden Arbeit evaluierten Routing Engines wurden erneut betrachtet und zusätzliche Routing Engines konnten erfolgreich integriert werden. Für die virtuelle Erkundung einer Region wurde eine Suchfunktion eingeführt, um potenzielle Stationen einer Route zu finden. Über einen Dialog können sie der Route hinzugefügt oder entfernt werden.

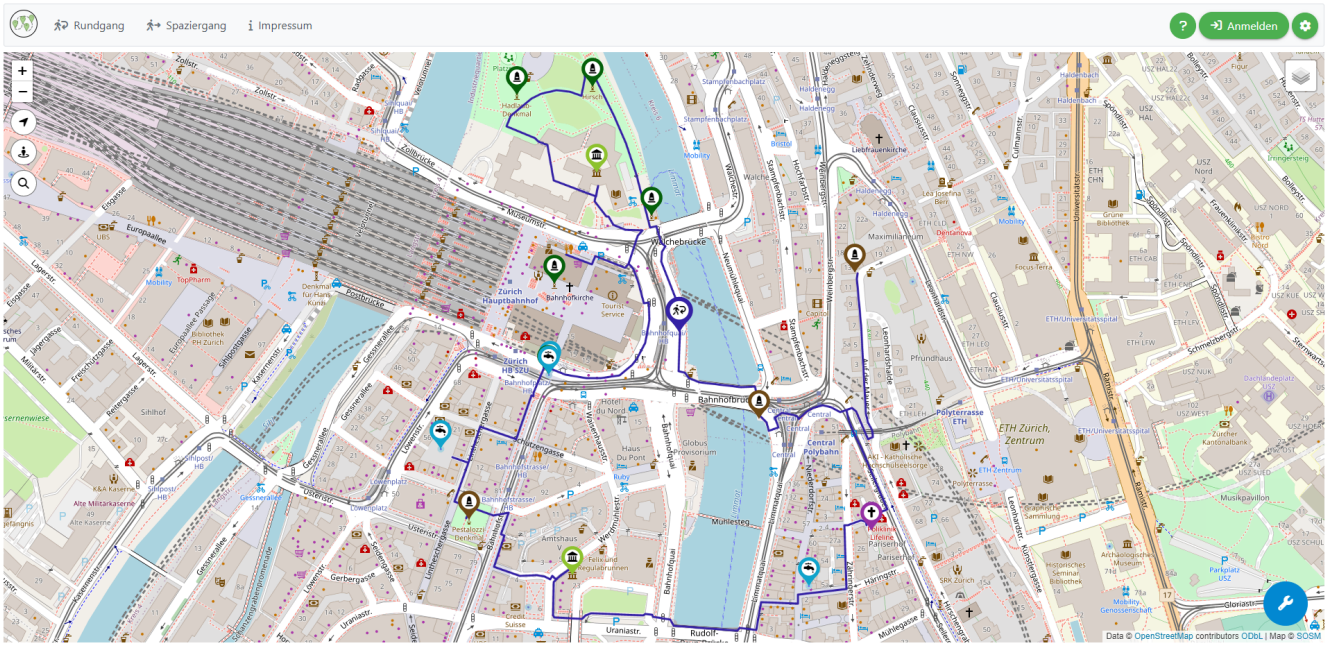

Abbildung 2. Beispiel eines touristischen Rundgangs um Zürich HB im Umkreis von 500 Meter

Der Hauptbestandteil der Erweiterung besteht aus der Implementation von Einwegtrips, den sogenannten Spaziergängen. Dadurch steht es den Benutzer:innen nun frei, ob sie an den Ausgangspunkt zurückkehren wollen und bietet ihnen weitere Flexibilität bei der Bedienung der Applikation. Ausserdem wurde die Darstellung der Informationsanzeige auf der Karte überarbeitet, sodass eine Unterscheidung zwischen Startpunkt und Station auf den ersten Blick ersichtlich ist. Zusätzlich wurde die in der Kartografie notwendige Generalisierung auch auf die Stationen angewandt. Dazu werden beieinanderliegende Stationen zusammengefasst und gruppiert dargestellt.

Abbildung 3. Beispiel eines Spaziergangs mit gruppierten Stationen für bessere Lesbarkeit

Um eine geräteübergreifende Synchronisation der individuellen Einstellungen der Applikation zu ermöglichen, wurde ein Benutzerprofil implementiert. Dies bedingte einen zusätzlichen Schutz der Applikation, denn sensitive Daten dürfen ausschliesslich verschlüsselt im öffentlichen Internet übertragen werden. Die neue Funktion wird ebenfalls genutzt, um die Speicherung und Verwaltung von individuellen Routen im Profil zu ermöglichen. Diese technischen Erweiterungen legten die Grundlage für die Zusammenarbeit mit Estefan Justo, einem Branchenexperten von Schweiz Tourismus. Seine Sichtweise hat zum Erfolg dieser Arbeit beigetragen. Besonders hinsichtlich Darstellung und Bedienung auf mobilen Geräten konnte das Produkt verbessert werden.

Für die Umsetzung dieser Arbeit wurden moderne Entwicklungsansätze, agile Methodiken und erprobte Technologien eingesetzt. Als Datengrundlagen dienen namentlich OpenStreetMap (Schweiz) und als Technologien unter anderem Python, FastAPI, PostgreSQL/PostGIS, TypeScript, Angular, Keycloak, sowie die Routing Engines OSRM, GraphHopper und openrouteservice.

Erweiterungen

Das Resultat der Arbeit bietet Potenzial für diverse Erweiterungen. Eine davon ist die Einführung einer zusätzlichen Meta-Ebene zur Bestimmung einer Routing Engine, welche sich am besten für den aktuellen Anwendungsfall eignet. Damit entfällt die Entscheidung der Benutzer:innen, welche Routing Engine sie für ihre Anfrage nutzen möchten. Beispielsweise stellen Treppen eine Herausforderung für Menschen im Rollstuhl oder Familien mit Kinderwagen dar. Dieses Feature wird nicht von allen Routing Engines unterstützt. Mithilfe zusätzlicher Logik könnte die passende Routing Engine bestimmt werden, welche fähig ist die Anforderung zu erfüllen.

Eine andere Erweiterung wäre die Ausarbeitung von zusätzlichen Routing-Profilen. So wäre es wünschenswert, dass ruhige und sichere Wege bevorzugt werden. Ebenfalls möglich ist es, dadurch Routen entlang von Gewässern und Naturparks zu erstellen. Die Erstellung der individuellen Profile wäre ausserdem kombinierbar mit der Meta-Ebene zur Findung einer passenden Routing Engine.

Jede berechnete Route wird in der Datenbank mit der aufgerufenen Spezifikation abgelegt. Mithilfe einer Datenanalyse und Machine Learning könnten daraus weitere Erkenntnisse und Annahmen zum Benutzerverhalten getroffen werden. Die resultierenden Modelle können für Vorschläge genutzt werden, welche Region für Benutzer:innen auch attraktiv sein könnten oder welche Top-Sehenswürdigkeiten pro Region existieren. Die Daten würden gegebenenfalls auch Erkenntnisse zur Optimierung eines Routing-Profils liefern. Damit dies möglich wird, müssen möglichst viele echte Benutzerdaten generiert werden. Allenfalls bräuchte es noch Ergänzungen am Domänenmodell, um gezielter Daten zu sammeln. Ausserdem wäre eine grossangelegte Kampagne notwendig, damit genügend Benutzer:innen die Platform nutzen.

Die Anzeige von Informationen zu verschiedenen Stationen könnte mithilfe von kantonalen und regionalen Tourismusorganisationen ebenfalls erweitert werden. Zürich Tourismus bietet, wie Schweiz Tourismus auch, eine offene Schnittstelle für solche Zwecke an. Dadurch kann die Informationsdichte erhöht werden.

Danksagungen

An dieser Stelle bedanken wir uns bei den Personen, welche uns während dieser Bachelorarbeit und des Studiums begleitet und unterstützt haben.

Als Erstes gilt unser Dank Prof. Stefan Keller. Er begleitete das Projekt bereits während der Studienarbeit und ermöglichte es uns, das erarbeitete Wissen in dieser Bachelorarbeit weiter zu vertiefen. Wir danken Prof. Stefan Keller für die etlichen Ideen und Anregungen, die investierte Zeit und die stetige sowie konstruktive Kritik. Wir profitierten von seiner Erfahrung und konnten unser Wissen erweitern.

Wir danken ebenfalls Estefan Justo von Schweiz Tourismus. Während unserer Zusammenarbeit konnten wir immer wieder Nutzen aus seiner Sichtweise und seinen Denkanstössen ziehen. Für seine aufgebrachte Zeit, den spannenden Austausch und das entgegengebrachte Interesse danken wir ihm herzlich.

Einen besonderen Dank möchten wir unseren Familien und Freunden aussprechen, welche uns während des ganzen Studiums unterstützt haben. Sie hatten stets ein offenes Ohr und Verständnis für unsere Anliegen. Wir danken euch von ganzem Herzen für eure Unterstützung in den vergangenen Jahren!

Inhaltsverzeichnis

Aufgabenstellung

City Trip Planner: Kurztrip-Planer für Fussgänger

Bachelorarbeit im Herbstsemester 2022/23, Bachelor-Studiengang Informatik

Hintergrund und Aufgabenstellung

Das Ziel der Arbeit ist es ein Minimal Viable Product (MVP) für einen Routen-Planer auf Grundlage der gleichnamigen Semesterarbeit der Autoren zu entwickeln. Mithilfe des Planers sollen Fussgänger:innen und Tourist:innen Routen zu kategorisierten Interessen im städtisch-urbanen Raum erstellen können. Erstellte Routen sollen in einer responsiven Webapp zugänglich gemacht werden.

Aufgaben

- Implementation eines weiteren MVP wie Einwegtrip als Erweiterung des Rundgangs
- Anschliessen von weiteren Open Source Routing Engines an Applikation
- User-Login z.B. für Personalisierung inkl. User-Management
- Responsive Webapp für Demo-Zwecke, zusätzlicher Ausbau im Frontend
- Evaluation der Ergebnisse

Lieferobjekte

1. Dokumentation, inkl. Textabstract, Management Summary, Anhänge (Literaturverzeichnis, Inhalt).
2. Lauffähige und webbasierte Applikation
3. Die vom Studiengang geforderten Lieferobjekte wie Poster (nur digital), Broschüren-Abstract.

Vorgaben – Technologien – Rahmenbedingungen

Analog der Semesterarbeit.

- Frontend: responsive Webapp mit HTML5, JavaScript; Angular
- Backend: Python
- Persistenz: Spatial SQL, PostgreSQL
- Daten: Open Data (OpenStreetMap)

Die Studierenden wählen nach Rücksprache ein Vorgehensmodell zur Softwareentwicklung.

Termine und Bewertung

Es gelten die üblichen Termine und Regelungen zum Ablauf und zur Bewertung der Arbeit des Bachelor-Studiengangs Informatik der OST. Zusätzlich wird Gewicht gelegt auf moderne Softwareentwicklung (Versionierung, Erweiterbarkeit, CI/CD) sowie auf ein lauffähiges MVP.

Beteiligte

Autoren: Jan Ruch und Lukas Grigis
Betreuer: Prof. Stefan Keller, Institut für Software OST
Industriepartner: Estefan Justo, Schweiz Tourismus

Vorarbeit

Dieser Bachelorarbeit ist die gleichnamige Studienarbeit [1] vorausgegangen. Sie dient als Grundlage für diese Arbeit und wurde im Herbstsemester 2021 an der Fachhochschule OST von Lukas Grigis und Jan Ruch eingereicht. Die darin entwickelte Software und der Source Code werden vollständig in dieser Arbeit übernommen und weiterentwickelt. In den folgenden Darstellungen wird aufgezeigt, welche Teile bereits existieren und welche für die vorliegende Arbeit entwickelt werden sollen.

Bereich Routing Engine

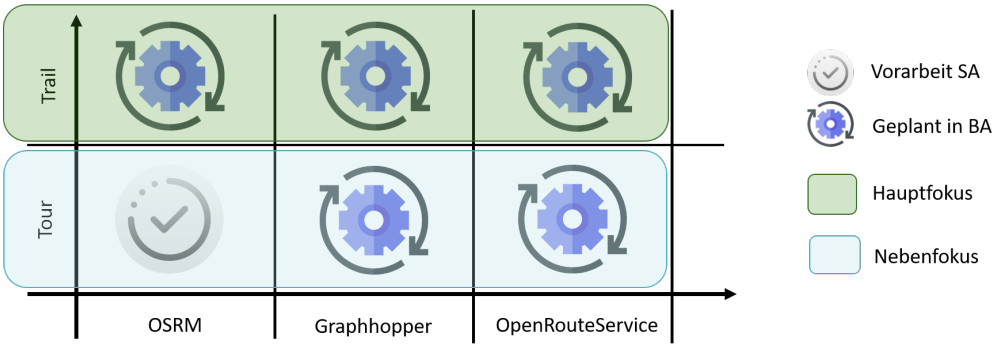

Abbildung 4. Geplante Erweiterungen in der Bachelorarbeit im Bereich Routing Engine [1]

In der Studienarbeit konnte OSRM erfolgreich für die Berechnung von Rundgängen eingesetzt werden. Diese Vorarbeit wird für die Bachelorarbeit übernommen. Darüber hinaus ist geplant, weitere Routing Engines anhand der Evaluation in der Studienarbeit anzubinden. Bei dieser Erweiterung steht primär die Entwicklung von Spaziergängen im Vordergrund. Die Berechnung von Rundgängen mithilfe der zusätzlichen Routing Engines steht im Nebenfokus.

Bereich DevOps

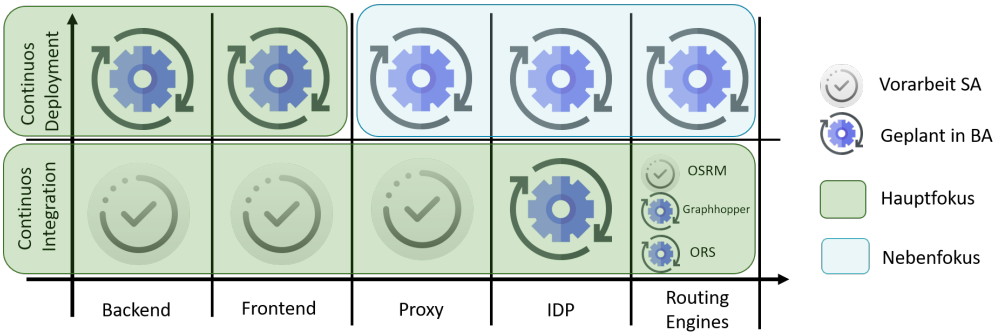

Abbildung 5. Geplante Erweiterung in der Bachelorarbeit im Bereich DevOps [1]

Im Bereich DevOps ist der Bau und die Konfiguration der Container bereits implementiert und kann übernommen werden. Die Umsetzung der fehlenden Teile ist für die Bachelorarbeit geplant. Das Deployment wurde in der Studienarbeit nur konzeptionell abgehandelt. Das Konzept wird in dieser Arbeit nochmals aufgegriffen und wo nötig angepasst, damit es den aktuellen Anforderungen entspricht. Es stellt die Grundlage für die Umsetzung des Continuous Deployment dar.

Bereich User-Login

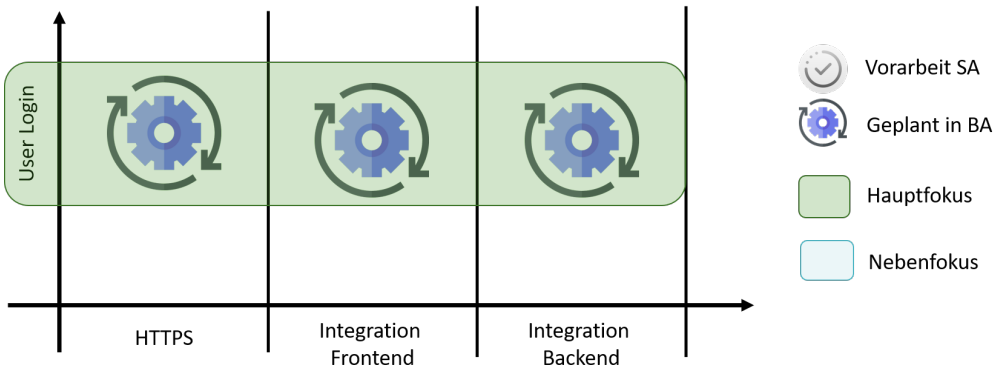

Abbildung 6. Geplante Erweiterung in der Bachelorarbeit im Bereich User-Login [1]

Im Bereich des User-Logins gibt es keine Vorarbeiten. Es wird in der Bachelorarbeit vollständig entworfen und implementiert.

Bereich Frontend

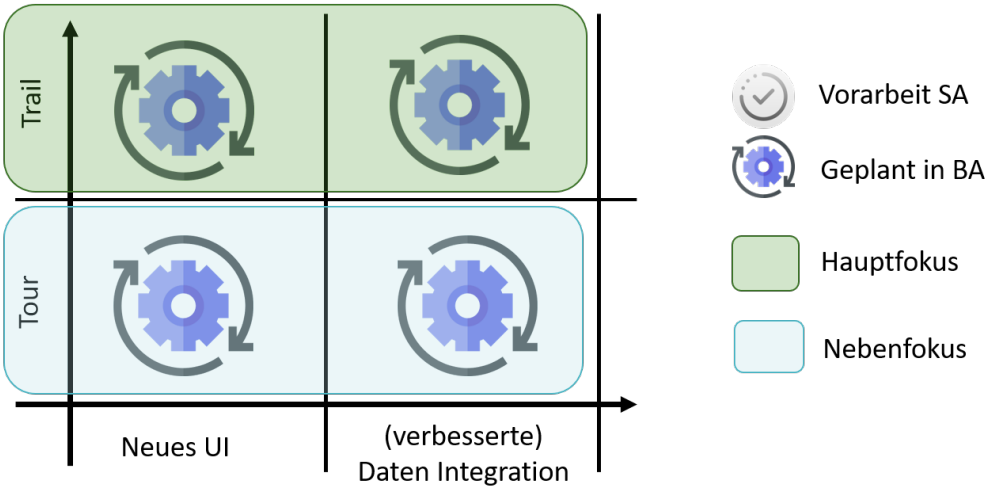

Abbildung 7. Geplante Erweiterung in der Bachelorarbeit im Bereich Frontend [1]

Die ursprüngliche Frontend-Applikation der Studienarbeit wird übernommen. Anhand der bekannten Defizite wird eine neue Benutzeroberfläche für Spaziergänge entworfen. Die existierenden Dienste zur Integration der Daten werden übernommen und überarbeitet. Dies steht im Hauptfokus dieser Bachelorarbeit.

Die Migration von Rundgängen auf die neue Benutzeroberfläche wird in einem zweiten Schritt erfolgen. Das übergeordnete Ziel ist es, die gesamte Funktionalität in einer einheitlichen Benutzeroberfläche zu integrieren.

[1] Dieses Cover wurde unter Verwendung von Ressourcen von Flaticon.com [2] erstellt.

Teil I: Technischer Bericht

1. Einführung

In dieser vorliegenden Arbeit geht es um einen Planer, mithilfe dessen Rundgänge und Spaziergänge in der Schweiz geplant werden können. Das Resultat der Planung ist eine individualisierte Route für Fussgänger:innen in Städten der Schweiz. Die Arbeit basiert auf der gleichnamigen Studienarbeit, welche im Herbstsemester 2021 von Lukas Grigis und Jan Ruch eingereicht wurde.

1.1. Industriepartner

Der Industriepartner für die Bachelorarbeit ist Schweiz Tourismus. Als öffentlich-rechtliche Körperschaft des Bundes, hat Schweiz Tourismus den Auftrag, die touristische Nachfrage der Schweiz als Destination für Ferien, Reisen und Kongresse im In- und Ausland zu fördern. Zu den Kernaufgaben von Schweiz Tourismus gehören [3]:

- Sorge um gesamtschweizerisches Marketing im Tourismus
- Partner für gemeinsame Marktauftritte zu gewinnen und zu koordinieren
- Proaktive Beratung von Mitgliedern durch erlangtes Fachwissen aus Marktforschung und -beobachtungen

Schweiz Tourismus bietet auf ihrer Webseite [4] Kurztrips in der Schweiz an. Interessierte können daraus einen vorgefertigten Kurztrip auswählen und erhalten Hintergrundinformationen dazu. Um in Zukunft individualisierbare Ausflüge anbieten zu können, besteht ein Interesse von Schweiz Tourismus an dieser Bachelorarbeit.

In der Studienarbeit wurde mittels eines Minimal Viable Product die prinzipielle Durchführbarkeit des Projekts anhand von Rundgängen bereits aufgezeigt. Die vorliegende Arbeit setzt sich mit dem Weiterausbau der geschaffenen Grundlage auseinander.

1.2. Problemstellung

Kurztrips erfreuen sich immer grösserer Beliebtheit in der Schweiz. Bislang gibt es jedoch keinen passenden Dienst, welcher die Planung von solchen Kurztrips für Fussgänger:innen anbietet. Die vorausgegangene Studienarbeit und die vorliegende Bachelorarbeit beschäftigen sich mit dem Thema, diesen fehlenden Dienst zu entwickeln. Dabei steht die Bedienbarkeit und Individualisierbarkeit durch die Benutzer:innen im Vordergrund.

1.3. Produktvision

Menschen sind von Natur aus individuell und einzigartig. Sie unterscheiden sich in ihrem Charakter, haben eigene Vorlieben und Interessen. Vorbereitete Kurztrips können je nach Mensch genau passen oder dem absoluten Gegenteil entsprechen. City Trip Planner beschäftigt sich mit dem Thema der Individualisierbarkeit beim Bereisen von Schweizer Städten. Dabei soll ein intuitiv und leicht bedienbarer Routenplaner für Fussgänger:innen entstehen, mit dem man in Schweizer Städten basierend auf den eigenen Interessen eine Route berechnen kann.

Unter Einbezug von Kriterien wie verfügbare Zeit, gewünschte Laufdistanz, Interessen und Standort können sich Tourist:innen einen attraktiven Ausflug als Rundgang oder Spaziergang zusammenstellen. Zu den ausgewählten Stationen werden Hintergrundinformationen und Bilder angezeigt, welche man vorher oder während des Ausflugs konsumieren kann und um die Station örtlich besser auszumachen. Diese Funktionen sollen es Tourist:innen ermöglichen, ein abgestimmtes Erlebnis zu schaffen und bei der Erkundung der Feriendestination Schweiz helfen.

1.4. Ziele

- Ausbau von bestehender Webapplikation für das Erstellen von Rundgängen und Spaziergängen in Schweizer Städten
- Fokus auf Bedienbarkeit auf mobilen Geräten und Desktop, sowie Individualisierbarkeit der persönlichen Interessen und Destinationen
- Erhalt von Zusatzinformationen mithilfe von Internetmedien an ausgewählten Stationen
- Ansprechende und moderne Applikation, welche sich zum Weiterausbau eignet

1.5. Rahmenbedingungen

- Das zu entwickelnde Produkt entspricht einem Minimal Viable Product, welches die konzeptionelle Umsetzbarkeit aufzeigt.
- Die verwendete Datenbasis wird durch die OpenStreetMap Community zur Verfügung gestellt und nicht durch die Absolventen erfasst.
- Das Deployment und der Betrieb der Applikation wird während der Entwicklung berücksichtigt und stellt einen wesentlichen Teil der Arbeit dar.

1.6. Vorgehen

In diesem Abschnitt wird das Vorgehen während der Bachelorarbeit grob skizziert. Diese Skizze dient als Grundlage für das Projektmanagement und die dazugehörigen Meilensteine.

1.6.1. Phase 0: Aktualisierungen der Software

Bevor die Weiterentwicklung des Resultats der Studienarbeit aufgenommen werden kann, muss die eingesetzte Software auf einen aktuellen Stand gebracht werden. Dabei ist der bestehende Funktionsumfang beizubehalten und wo nötig anzupassen. Diese Aktualisierung ist notwendig, um den Wartungsaufwand in der Zukunft möglichst gering zu halten und allfällige Schwachstellen zu mitigieren.

1.6.2. Phase 1: Umsetzung der Primärziele

Die Primärziele repräsentieren die technischen Voraussetzungen für die Erweiterung der Applikation. Erst nach dem Erreichen der Primärziele können weitere Anforderungen aus dem Business umgesetzt werden. Gleichzeitig soll die Umsetzung der Primärziele der Untermauerung oder Verwerfung der getroffenen Entscheidungen aus der Studienarbeit dienen.

Nach Abschluss dieser Phase müssen folgende Artefakte vorliegen:

- Anschluss einer oder mehrerer zusätzlicher Routing Engines
- Integration von geeignetem Identity Provider für User-Login
- Erfolgreiche Berechnung und Darstellung von Spaziergängen
- Vereinheitlichung der Benutzeroberfläche für Rundgang und Spaziergang

1.6.3. Phase 2: Umsetzung der Sekundärziele

Sobald die technische Machbarkeit mit der Umsetzung der Primärziele sichergestellt ist, kann mit dem Ausbau von zusätzlichen Features begonnen werden. Der Input für die Sekundärziele wird von Schweiz Tourismus geliefert. Das übergeordnete Thema ist die Bedienbarkeit und das Benutzerverständnis der entworfenen Applikation.

Nach Abschluss dieser Phase soll die Applikation:

- sich benutzerfreundlich verhalten
- intuitiv bedienbar sein
- zusätzliche Features enthalten, welche einen Mehrwert bieten

Die genauen Anforderungen können zum Zeitpunkt der Erstellung des Umsetzungsplans, aufgrund der agilen Vorgehensmethode und der technischen Abhängigkeit zur vorhergehenden Phase, nicht festgelegt werden.

1.6.4. Phase 3:

In der dritten und letzten Phase werden wir uns dem Abschluss des Projektes und der Dokumentation widmen. Nach Abschluss dieser Phase müssen folgende Artefakte zur Abgabe bereit sein:

- Lauffähiges Minimal Viable Product für Demonstrationen
- Geforderte Dokumente und Abgaben für Bachelorarbeit
- Präsentation und Poster

2. Stand der Technik

Dieses Kapitel beschäftigt sich mit dem Stand der Technik hinsichtlich der technischen Problemstellung, den existierenden Algorithmen und bestehenden Lösungsansätzen. Unter Einbezug der theoretischen Informatik wird die Problemstellung eingeordnet. Diese Einordnung dient dann als Orientierungshilfe, um die Relevanz von Algorithmen zu bestimmen. Anhand der betrachteten Gesichtspunkte werden ähnliche, bereits umgesetzte Projekte und Arbeiten beurteilt, um daraus Erkenntnisse für die vorliegende Arbeit abzuleiten.

2.1. Graphentheorie

Ein wesentlicher Bestandteil dieser Arbeit ist die Berechnung einer Route mithilfe einer Routing Engine. Routing Engines basieren auf Algorithmen der Graphentheorie. Sowohl die diskrete Mathematik als auch die theoretische Informatik beschäftigen sich mit der Graphentheorie. In diesem Abschnitt werden die Eigenschaften und Beziehungen betrachtet, um ein besseres Verständnis der Aufgabe zu erhalten.

2.1.1. Hamiltonkreisproblem

Eine wichtige Frage in der Graphentheorie ist, ob ein geschlossener Pfad in einem Graphen existiert, welcher jeden Knoten genau einmal enthält. Dabei haben die Gewichte der verwendeten Kanten keine Relevanz. Das bedeutet also, dass sich das Hamiltonkreisproblem mit der Frage beschäftigt, ob ein Pfad auf dem Graphen existiert, wobei die resultierende Distanz keine Rolle spielt. Dieses Problem ist NP-vollständig. Wenn ein Problem NP-vollständig ist, lässt es sich nichtdeterministisch in polynomieller Zeit lösen. [5] Dies ist eine wichtige Erkenntnis für die vorliegende Arbeit. Das zu entwickelnde Produkt soll als Internetdienst angeboten werden. Im Internetumfeld ist die Geschwindigkeit ein entscheidendes Kriterium, ob Benutzer:innen einen Dienst nutzen oder nicht. Somit lässt sich sagen, dass sich mögliche Algorithmen, das heisst bereits bekannte oder heute noch unbekannte, für die Berechnung eines Hamiltonkreisproblems für diese Bachelorarbeit nicht eignen.

2.1.2. Problem des Handlungsreisenden

Das Problem des Handlungsreisenden stellt eine Verallgemeinerung des Hamiltonkreisproblems dar. Bei diesem Problem geht es darum, den kürzesten Pfad in einem Graphen zu finden, welcher jeden Knoten eines Subsets genau einmal enthält. Dabei spielt, im Gegensatz zum Hamiltonkreisproblem, das Gewicht der Kanten eine entscheidende Rolle. Ausserdem müssen nicht alle Knoten innerhalb des Graphen besucht werden. Eine gemeinsame Eigenschaft des Hamiltonkreisproblems und des Problems des Handlungsreisenden ist, dass der Startknoten gleich dem Endknoten sein muss. Die resultierenden Graphen sind bei beiden Problemen zyklisch. Dadurch, dass nicht alle Knoten besucht werden müssen und es sich bei diesem Problem um eine Verallgemeinerung handelt, reduziert sich die Komplexität bei der Berechnung. Dieses Problem ist NP-schwer. [6]

Diese Eigenschaften entsprechen der Berechnung eines Rundgangs nach der Auffassung dieser Bachelorarbeit. Die Stationen repräsentieren das Subset *S* aus zu besuchenden Knoten im Graphen *G* des Fussgängerverkehrsnetzes der Schweiz, wobei die Distanz der Summe der Kantengewichte möglichst klein sein und der Startpunkt gleich dem Endpunkt sein soll.

2.1.3. Eulerkreisproblem

Das Eulerkreisproblem beschäftigt sich mit der Frage, ob es einen Pfad in einem Graphen gibt, welcher jede Kante genau einmal enthält. Dabei dürfen Knoten mehrmals besucht werden. Dieses Problem ist gelöst und in linearer Zeit mittels Tiefensuche berechenbar. [7]

Die bisherigen Probleme sind davon ausgegangen, dass in der Sequenz jeder Knoten genau einmal enthalten ist. Bei diesem Problem wird nach einem Pfad gesucht, welcher in der Sequenz jede Kante genau einmal enthält.

Für die vorliegende Arbeit brauchen wir jedoch keinen Eulerkreis, sondern einen Eulerpfad. Beim Eulerpfad handelt es sich um eine Verallgemeinerung des Eulerkreises, wobei Start- und Endknoten nicht gleich, sondern unterschiedlich sind.

Die Eigenschaften der Berechnung eines Eulerpfads entsprechen denjenigen eines Spaziergangs nach Auffassung dieser Bachelorarbeit. Die Stationen repräsentieren dabei das Subset *S* aus zu besuchenden Knoten im Graphen *G* des Fussgängerverkehrsnetzes der Schweiz, wobei jede gerichtete Kante genau einmal enthalten ist und Start- sowie Endknoten unterschiedlich sind.

2.2. Algorithmen

In diesem Abschnitt werden die Algorithmen zur Problemlösung der genannten Probleme aus der theoretischen Informatik betrachtet. Dabei steht die grundsätzliche Herangehensweise und der Ablauf im Vordergrund, welche bei der Einordnung des Vorhabens helfen.

2.2.1. Concorde TSP Solver

Einer der wohl bekanntesten Solver für das Problem des Handlungsreisenden ist *Concorde TSP Solver* [8]. Die Software ist in *C* (Programmiersprache) geschrieben und wird als einer der besten Implementationen gehandelt.

2.2.2. Self-Organizing Maps

Ein weiterer Ansatz zur Problemlösung sind *Self-Organizing Maps*. In seinem Blog [9] erklärt Diego Vincente wie er das Problem des Handlungsreisenden mithilfe dieser Technik gelöst hat. Auch bei dieser Variante zur Lösung des Problems wurden Annahmen getroffen, um innert nützlicher Frist eine passende Lösung zu finden.

2.2.3. Hierholzer Algorithmus

Der deutsche Mathematiker Carl Hierholzer entwickelte bereits im späten 19. Jahrhundert einen Algorithmus um das Eulerkreisproblem zu lösen. Obschon der Algorithmus lange vor dem ersten Computer entwickelt wurde, ist er bis heute von Relevanz. Der Algorithmus skaliert nämlich linear mit der Anzahl Knoten im Graph. Abhijit Purru erläutert und illustriert auf seinem Blog [10] schrittweise das Vorgehen und den Graphen.

2.2.4. Fazit zu Algorithmen

Bei der Lösungsfindung für das Problem des Handlungsreisenden kommen häufig Heuristiken zum Einsatz. Diese Methode kommt auch beim Trainieren von Modellen in der künstlichen Intelligenz und anderen Ingenieurswissenschaften zum Einsatz. Obschon durch den Einsatz von Heuristiken oftmals ein Geschwindigkeitsvorteil erzielt werden kann, muss man bedenken, dass das Resultat unter Umständen nicht allgemeingültig generalisiert.

> Bei der Suche des schnellsten Weges von Zürich nach Bern ist es eine voreilige Schlussfolgerung, davon auszugehen, dass dieser schnellste Weg auch der kürzeste und somit umweltfreundlichste ist.

Bei der Verkehrsführung werden beispielsweise Contraction Hierarchies eingesetzt. Dieser Vorgang besteht aus zwei Phasen. In der ersten Phase werden die Daten vorprozessiert, in dem man implizite Annahmen trifft. In der zweiten Phase werden die zuvor aufbereiteten Daten nun abgefragt. Dieser Vorgang würde ohne die Aufbereitung der Daten viel länger dauern.

Contraction Hierarchies können bei der Verkehrsführung deshalb eingesetzt werden, weil es auch verschiedene Klassen (*Hierarchien*) bei den Strassen gibt. Diese Klassen ermöglichen es, eine Autobahn einer Überlandstrasse vorzuziehen. Im Normalfall kann man auf einer Autobahn schneller fahren als auf einer Überlandstrasse. Unter dieser Annahme werden die Daten nun vorprozessiert. Das Modell wird auch die Abfrage der schnellsten Route von Zürich nach St. Gallen beantworten können. Es ist jedoch falsch, anzunehmen, dass dieselbe Route auch die kürzeste oder gar die umweltfreundlichste ist.

Ein weiterer Ansatz, das Travelling Salesman Problem (de: Problem des Handlungsreisenden) zu lösen, sind genetische Algorithmen. Sie orientieren sich an der natürlichen Evolution zur Lösung von Optimierungsproblemen. Mithilfe von dynamischer Programmierung werden verschiedene Evolutionen eines Resultats berechnet. Nach jeder Evolution wird das Zwischenresultat geprüft und beurteilt. Wenn es besser - oder *optimaler* - ist, wird die nächste Evolution gestartet. Falls nicht, wird das Zwischenresultat verworfen und sofern notwendig die Eingangsparameter modifiziert, um erneut eine Berechnung zu starten.

2.3. Bestehende Lösungsansätze

POI Tour [11]

Diese Arbeit beschäftigt sich mit einer sehr ähnlichen Fragestellung, beschränkt sich aber auf Rundgänge. Diese sogenannten Touren stellen sich aus Points of Interest zusammen, welche von der OpenStreetMap Community zur Verfügung gestellt werden. Für die Berechnung der Route wurden Project Open Source Routing Machine und pgRouting Project gegenüber gestellt. Der Source Code des Projekts ist auf GitHub [12] veröffentlicht.

The Trip Boutique [13]

Dieses kommerzielle Produkt bietet personalisierte Trips in europäischen Städten an. Anhand der von der Benutzer:in eingegebenen Interessen wird ein Profil errechnet, mithilfe dessen dann ein entsprechender Vorschlag unterbreitet wird.

MakeMyDriveFun [14]

MakeMyDriveFun bietet einen Dienst an, mithilfe dessen man entlang geplanter Autofahrten Sehenswürdigkeiten besichtigen kann. Dazu muss man den Ausgangspunkt und das gewünschte Ziel eingeben, woraufhin einem die Route und mögliche Points of Interest angezeigt werden.

A System for Generating Customized Pleasant Pedestrian Routes Based on OpenStreetMap Data [15]

In dieser Arbeit wird ein System präsentiert, welches Routen für Fussgänger:innen nur basierend auf von OpenStreetMap stammenden Daten erstellt. Dafür werden Wege durch Parks oder entlang von verkehrsarmen Strassen positiv in der Kostenfunktion des Routings berücksichtigt. Die Faktoren, welche die Kostenfunktion beeinflussen, können interaktiv von Benutzer:innen eingegeben werden.

Trail Router [16]

Trail Router bietet einen Webdienst an, welcher basierend auf Eingaben von Benutzer:innen nach Grünflächen, Gewässern oder Wanderrouten sucht und diese auf einer Route verbindet. Der Dienst unterscheidet zwischen sogenannten *Rundwegen* und *Einwegstrecken*.

Inspirock [17]

Inspirock ist eine kommerzielle Platform, auf der man sich mehrtägige Trips zusammenstellen kann. Dazu wird im Umkreis der gewählten Destination nach Sehenswürdigkeiten gesucht, welche dann wiederum auf die Tage verteilt, besucht werden können.

A Computational Model of Pedestrian Road Safety: the Long Way Round is the Safe Way Home [18]

In diesem Artikel wird eine Lösung präsentiert, wie die Sicherheit von Fussgänger:innen in britischen Städten verbessert werden kann. Diese Arbeit scheint auf den ersten Blick nicht mit dieser Bachelorarbeit verwandt zu sein. Allerdings ist das Thema Sicherheit auf der Strasse auch relevant für Benutzer:innen von City Trip Planner.

PlazaRoute [19]

Die Arbeit *PlazaRoute* adressiert den Umstand, dass Routing Engines nicht über offene Plätze führen, sondern deren Rand entlang. Darin werden verschiedene Möglichkeiten zur Vorprozessierung von Daten vorgestellt, womit eine Routing Engine Routen erstellen kann, welche dem natürlichen Verhalten eines Menschen entsprechen.

Computing Optimal Road Trips Using Operations Research [20]

In seinem Blog erklärt Nathan Brixius wie er mithilfe eines genetischen Algorithmus den kürzesten Pfad durch Hauptstädte von US-amerikanischen Bundesstaaten sucht. Er erklärt in seinem Post, welche Annahmen er getroffen hat und auf welche Probleme er gestossen ist.

plotaroute.com [21]

Auf der Seite www.plotaroute.com wird ein Dienst angeboten, mithilfe dessen sich Routen planen lassen. Dabei können viele Einstellungen vorgenommen werden, um beispielsweise Geschwindigkeit oder Bewegungsart anzupassen.

RouteYou [22]

RouteYou ist ein kommerzieller Dienst und bietet Routing für zahlreiche Bewegungsarten an. Dadurch können Strecken von A nach B und Rundwege erstellt werden. Zusätzlich gibt es die Möglichkeit, sich Routen generieren zu lassen und per Navigation der Route zu folgen. Dies ist allerdings erst nach Registration und Abonnement möglich.

Greco, F. (2008). Traveling Salesman Problem. IntechOpen. [23]

Federico Greco erklärt in seinem Buch *Traveling Salesman Problem* diverse Lösungsansätze und Herangehensweisen. Er geht dabei auch auf Aspekte der theoretischen Informatik ein und gibt einen vertieften Einblick in die angewandte Methode des Lösungsansatzes.

2.4. Beurteilungskriterien

Die gewonnenen Erkenntnisse aus dem Stand der Technik sollen in diese Arbeit einfliessen. Dafür werden folgende Kriterien festgelegt:

- Welche technischen Grenzen sind durch die theoretische Informatik gegeben und können nicht ohne Weiteres überwunden werden?
- Welche Bereiche sind gut erforscht und können für diese Arbeit angewendet werden?
- Anhand welcher Kriterien werden Points of Interests festgemacht und wie kann man effizient danach suchen?
- Welche Funktionen für Kartenapplikationen gibt es und wie werden sie dargestellt und bedient?
- Welche Dienste müssen neu entwickelt werden und können nicht bezogen werden?

2.5. Defizite

Die meisten Lösungen bieten fertige Ausflüge an, welche nach deren Erstellung oftmals gar nicht oder nur geringfügige Individualisierungen einer Route zulassen. Die Points of Interest können meist nicht aufgrund ihrer geografischen Lage ausgewählt werden. Für City Trip Planner ist es aber von Bedeutung, die Points of Interest aufgrund der Art und der geografischen Lage ausfindig zu machen, damit einerseits auch weniger bekannte Sehenswürdigkeiten berücksichtigt werden und andererseits, dass innerhalb eines Perimeters möglichst viel von einer Destination erlebt werden kann. Ein weiteres Defizit vieler Lösungen ist die Lauffähigkeit auf Mobilgeräten und Desktops. Es gibt Applikationen, welche ausschliesslich auf dem Smartphone oder dem Desktop nutzbar sind. Um Benutzer:innen ein umfassenderes Erlebnis zu bieten, erachten wir die Kompatibilität für Smartphones, Tablets und Desktops als besonders wichtig.

2.6. Schlussfolgerungen

2.6.1. Algorithmus

Dadurch, dass City Trip Planner als Webapplikation umgesetzt werden soll, welche auch auf Mobilgeräten funktioniert, ist die Geschwindigkeit von grosser Bedeutung. Dadurch müssen allerdings Abstriche bei der Exaktheit eines Resultats gemacht werden. Benutzer:innen erwarten in diesem Umfeld kurze Wartezeiten, ansonsten könnten sie das Interesse verlieren.

> Aufgrund dieses Umstands verzichten wir bei der Suche nach einer geeigneten Route darauf, diese maximal zu optimieren. Dies würde aufgrund der Problemkomplexität zu viel Zeit in Anspruch nehmen. Wir befürchten, dass bei zu langen Berechnungszeiten Benutzer:innen das Interesse an der Applikation verlieren. Wir nehmen deshalb in Kauf, dass berechnete Routen nicht optimal sind, solange sie schnell und die Kriterien von Benutzer:innen erfüllen.

Eine der Hauptherausforderungen der Bachelorarbeit wird es sein, einen passenden Algorithmus zu finden, welcher Rundgänge und Spaziergänge effizient berechnen kann. Dies betrifft die Suche nach möglichen Stationen und die Berechnung der Route gleichermassen. Die Optimierung spielt dabei eine untergeordnete Rolle. Um dies zu erreichen, müssen wir eigene Heuristiken definieren und die Daten so vorverarbeiten, dass eine Routing Engine sie anschliessend effizient und ohne Redundanzen verbinden kann.

2.6.2. Manifestierung von Stationen

Keine der betrachteten Arbeiten berichtet über eine Kategorisierung von Points of Interest. Die erfassten Daten von der OpenStreetMap Community sind sehr breit und werden anhand verschiedener Kriterien klassiert. Darunter gibt es aber auch viele Einträge, welche keine Relevanz für diese Bachelorarbeit haben. Um passende Stationen anbieten zu können, müssen wir in diesem Bereich tätig werden und einen eigenen Algorithmus entwerfen.

Dieser Algorithmus muss mindestens zwei verschiedene Strategien aufweisen. Wir unterscheiden in dieser Arbeit zwischen Rundgang und Spaziergang. Aufgrund deren Verschiedenheit in Erscheinung und Eigenschaften, gehen wir davon aus, dass sich die geografische Suche nach Points of Interest unterscheiden wird. Deshalb nehmen wir an, dass wir für jede der beiden Ausprägungen eine eigene Strategie entwerfen müssen.

2.6.3. Erkenntnisse aus bestehenden Lösungsansätzen

Aus den bestehenden Lösungsansätzen geht hervor, dass die Planung von fussgängerfreundlichen Routen ein weitverbreitetes Thema ist. Darunter gibt es verschiedene Aspekte, die es zu berücksichtigen gilt. Die Themen wie Green Routing, Verkehrssicherheit und Adressierung an bestimmte Zielgruppen sind mehrfach zur Sprache gekommen. Diese Themen können unter dem Überbegriff *Routing Profile* zusammengefasst werden. Dies sind sicherlich wichtige und interessante Aspekte bei der Entwicklung einer Routing Engine. Das Ziel dieser Bachelorarbeit ist es ein Minimal Viable Product zu schaffen, welches in der Lage ist, individualisierbare Routen für touristische Zwecke zu berechnen. Dazu werden wir aber vorerst existierende Routing Profile verwenden. Erst sobald die technische Machbarkeit bewiesen ist und Resultate angezeigt werden, können wir feststellen, welche Routing Profile sich eignen oder unbrauchbar sind. Ohne diese Erkenntnisse, macht es aus unserer Sicht noch keinen Sinn, ein eigenes Routing Profil zu entwickeln.

RouteYou bieten eine Vielzahl von Möglichkeiten an, eine Route zu planen. Es wird ebenso zwischen Einwegtrips und Rundgängen unterschieden. Die zahlreichen Routentypen und Routing-Arten bieten für jegliche Art der Fortbewegung Alternativen an, um ein bestmögliches Erlebnis zu schaffen. Es fehlt allerdings die Funktion, dass Points of Interest in die Route miteingebunden werden. Dadurch ist auch der Zugriff auf Hintergrundinformationen nicht möglich.

In unserer Studienarbeit haben wir entschieden, dass eine eigene Implementation dieses Funktionsumfangs, aufgrund des Fehlens eines geeigneten Produkts auf dem Markt, zielführend ist. Diese Entscheidung vertreten wir auch nach erneuter Analyse der bestehenden Lösungsansätze. Deshalb basiert diese Bachelorarbeit auf den damaligen Erkenntnissen und baut darauf auf. Der ursprüngliche Funktionsumfang des Rundgangs (Hinweis: in der Studienarbeit *Rundtrip* genannt) soll in dieser Arbeit um Spaziergänge ergänzt und weitere Anforderungen aus der Branche umgesetzt werden.

3. Umsetzungskonzept

Das Umsetzungskonzept orientiert sich an den vom Rational Unified Process definierten Phasen. Innerhalb dieser Phasen gilt es, die in der Aufgabenstellung geforderten Ziele zu erreichen. Als Projektgrundlage dient die vorausgehende Studienarbeit. Im Kapitel Projekt Management wird erläutert, wie sich der Entwicklungsprozess zusammenstellt und welche projektinternen Abhängigkeiten es gibt. Ausserdem werden in genanntem Kapitel die Meilensteine definiert.

Die Bachelorarbeit wird aus zwei Phasen bestehen. In einer ersten Phase der Bachelorarbeit konzentrieren wir uns auf technische Erweiterungen, die uns in einer zweiten Phase die Implementation von zusätzlichen, funktionalen Anforderungen ermöglichen sollen. Im Folgenden werden die Teilabschnitte des Rational Unified Process mit den vorgesehenen Arbeitsschritten umrissen.

3.1. Inception

In dieser ersten Phase beschäftigen wir uns mir der (Wieder-)Aufnahme der Tätigkeit. Diese Phase wird genutzt, um eine Vision zu definieren, welche während des Projekts verfolgt wird. Dazu werden die Primärziele dieser Arbeit in die bestehende Architektur gelegt und mögliche Risiken identifiziert. Zusätzlich wird ein rudimentärer Projektplan ausgearbeitet, der in der folgenden Phase detailierter spezifiziert wird. Neben diesen Tätigkeiten wird die bereits existierende Software aus der Studienarbeit auf den aktuellen Stand gebracht.

Beim Abschluss dieser (Re-)Initialisierungsphase müssen folgende Punkte erfüllt sein:

- Rudimentärer Projektplan ist erstellt und berücksichtigt die wesentlichen Punkte der Aufgabenstellung.
- Code Base, Entwicklungsumgebungen und andere Tools sind bereit für die weiteren Arbeitsschritte.
- Kontaktpersonen, Termine und Vorgehensweise sind definiert und aufgesetzt.

3.2. Elaboration

In dieser zweiten Phase werden die mentalen Modelle aus der Inception formalisiert und festgehalten. Dazu werden die bestehenden Konstrukte aufgenommen und analysiert, wie sich der Ausbau gestalten lässt. Ein wichtiger Bestandteil dieser Analyse ist es, die fehlenden Entitäten und Systeme auszumachen und gemäss der Vision zu entwerfen. Der Entwurf wird anschliessend genutzt, um Abhängigkeiten und Bedingungen aufzuzeigen, welche bei der Umsetzung gelöst werden müssen.

Bevor mit der Umsetzungsphase begonnen werden kann, müssen folgende Punkte entworfen sein:

- Integration der neuen Routing Engines
- Einführung von Security für die Implementation des User-Login
- Ausbau der Grundfunktionalität zur Berechnung von Einwegtrips (Spaziergängen)

Aufgrund der Verschiedenheit dieser drei Themengebiete haben wir beschlossen, dass die nächste Phase für jedes Themengebiet nach Abschluss der Elaboration individuell begonnen werden kann. Deshalb hat auch jedes dieser Primärziele einen eigenen Meilenstein im Projekt erhalten. Diese Meilensteine dienen wiederum als Synchronisationspunkt während des Projekts, welche erreicht worden sein müssen, damit mit den Sekundärzielen begonnen werden kann.

3.2.1. Anbinden von weiteren Routing Engines

In der Elaboration Phase der Studienarbeit wurde beim Entwurf der Architektur speziell darauf geachtet, dass für die Interaktion mit einer Routing Engine eine klar definierte Schnittstelle besteht. Dies sollte das Anbinden von alternativen Routing Engines vereinfachen, wobei es keine Rolle spielen soll, ob die Routing Engine self-hosted oder über einen Dienst bezogen wird. Diese aufgestellte Behauptung gilt es nun in der vorliegenden Arbeit zu beweisen.

Ebenso wurde in der Studienarbeit eine Evaluation von geeigneten Routing Engines durchgeführt. Diese Evaluation dient als Entscheidungsgrundlage für die Wahl der anzubindenden Routing Engines. Mit der Anbindung zusätzlicher Routing Engines soll nicht nur die gewählte Architektur unterstrichen werden, sondern sie bietet auch einen interessanten Blick auf die technische Sicht. Wir streben einen Vergleich zwischen den Resultaten der verschiedenen Routing Engines an.

3.2.2. Einführung von Security für Implementation von User-Login

Damit Benutzer:innen berechnete Routen und Einstellungen abspeichern können, braucht es ein Benutzerprofil. Der Zugriff auf dieses Benutzerprofil soll mittels Login geschützt werden. Damit dies möglich ist, muss die Kommunikation über HTTPS erfolgen, einem verschlüsselten Protokoll. Dies ist eine essenzielle Vorarbeit für das Benutzerlogin. In der heutigen Zeit ist es nicht mehr erdenklich, Personally Identifiable Information unverschlüsselt über das Internet zu übertragen. Wir müssen uns damit befassen, wie wir unsere Applikation mithilfe von TLS sichern.

Nach der erfolgreichen Einführung von TLS soll ein Benutzerlogin implementiert werden. Wir haben bereits zu Beginn dieser Arbeit entschieden, dass wir die Benutzerverwaltung und die damit verbundenen Sicherheitskonzepte nicht selbst realisieren werden. Zu gross ist das Risiko, dass dabei Fehler nicht bemerkt oder gar Schwachstellen eingebaut werden. Wir werden einen sogenannten Identity Provider zur Bereitstellung der gewünschten Funktionalitäten einsetzen. Dafür muss zuerst ein geeigneter Identity Provider mit einer geeigneten Spezifikation selektiert werden. Anschliessend müssen die Zugriffe aus der Frontend-Applikation im Browser der Benutzer:innen sowie aus der Backend-Applikation sichergestellt werden. Mithilfe der nun zur Verfügung stehenden Identität der Benutzer:innen können wir nun die Benutzereinstellungen in eigenen Datenbanktabellen abspeichern und beim Zugriff über REST-Controller die Authentizität der Benutzer:innen überprüfen.

3.2.3. Ausbau der Grundfunktionalität zur Berechnung von Einwegtrips (Spaziergängen)

Bereits in der Studienarbeit hatten wir festgestellt, dass es zwei grundverschiedene Anforderungen an Routen gibt. Die eine Ausprägung davon ist die Berechnung einer Route, welche wieder am Ausgangspunkt endet. In dieser Arbeit sprechen wir dabei von Rundgängen. Andere Begriffe dafür sind *Touren*, *Rundtrips* oder *Rundtouren*. Die andere Ausprägung ist die Berechnung einer Route, wobei der Startpunkt und der

Endpunkt verschieden sind. In dieser Arbeit wird diese Art einer Route Spaziergang genannt. In der Literatur finden sich auch dafür andere Begriffe wie *Einwegtrip*, *Trip (von A nach B)* oder *Pfad*. Beide Ausprägungen haben jedoch gemeinsam, dass sie möglichst kurz sein und möglichst viele Stationen beinhalten sollen.

Um die vorgestellten Spaziergänge zu realisieren, brauchen wir eine neue Entität. Die dazugehörigen Services werden als Vertical Slice, analog zu den existierenden Services der anderen Entitäten, implementiert. Sobald die Implementation im Backend abgeschlossen ist, werden wir mit der Entwicklung der notwendigen Komponenten im Frontend starten. Bei der Entwicklung der Frontend-Komponenten werden wir ausserdem eine neue Technologie für State Management einführen. Wenn die Technologie den erwarteten Nutzen bringt, können wir die bereits bestehenden Komponenten migrieren.

3.3. Construction

Die Construction Phase dient dazu, die ausgearbeiteten Entwürfe umzusetzen. Sie besteht im Wesentlichen darin, die zuvor erstellten Modelle zu einer lauffähigen Version zu bringen. Diese Phase hat kein definiertes Beginndatum, sondern beginnt für jedes der drei Primärziele individuell. Dies gibt uns die Freiheit unabhängig und dennoch zeitgleich am Projekt zu arbeiten und die Funktionalität umzusetzen. Nach Abschluss der drei genannten Primärziele beginnen wir mit den Sekundärzielen. Die genaue Spezifikation dieser Sekundärziele ist zu diesem Zeitpunkt nicht möglich. Anhand der definierten Vision lassen sich allerdings mögliche Anforderungen manifestieren.

3.3.1. Umsetzung der Anforderungen aus Business

Nun befinden wir uns in der angesprochenen zweiten Phase der Bachelorarbeit. Bis hierhin haben wir neue Anforderungen umgesetzt, um das Produkt technisch auszubauen und unserem Industriepartner Estefan Justo einige Funktionalitäten anzubieten, welche jetzt gemeinsam ausgearbeitet werden können.

Diese externe Sicht auf das Produkt ist enorm wünschenswert, da wir über kein branchenübliches Wissen verfügen. Ausserdem soll der Austausch früh Feedback einbringen, damit die Anpassungen zeitnah und mit geringen Kosten umgesetzt werden können. Unsere Erwartungshaltung ist, dass wir Rückmeldungen zu zusätzlichen Anforderungen und der Benutzbarkeit der Applikation erhalten.

> Die erwähnten Kosten entsprechen nicht einem monetären Gegenwert, sondern der zur Verfügung stehenden Zeit. Die Zeit ist die kritische Ressource in dieser Bachelorarbeit. Eine Änderung ist *teuer*, wenn sie lange dauert.

3.4. Transition

Der letzte Abschnitt befasst sich mit dem Abschluss der Arbeit. In diesem Teil werden wir das Ergebnis unserer Arbeit reflektieren, allfällige letzte Fehler beheben, die Dokumentation überarbeiten und unsere persönlichen Erfahrungsberichte schreiben. Sie endet mit der Einreichung der Bachelorarbeit.

4. Resultate

Die erzielten Resultate der Bachelorarbeit werden in diesem Kapitel festgehalten. Wir beziehen uns in den folgenden Abschnitten auf die Aufgabenstellung und beleuchten mögliche Weiterentwicklungen. Eine Evaluation der Resultate hinsichtlich der definierten Anforderungen, sowie eine ausführliche Auflistung von Weiterentwicklungen, finden sich im Kapitel [results-and-enhancements] in der Projektdokumentation.

4.1. Zielerreichung

Das Ziel der Arbeit war es ein Minimal Viable Product für einen Routenplaner zu entwickeln. Dazu haben wir die bestehende Vorarbeit weiterentwickelt und als lauffähige Webapplikation auf einer Demo-Umgebung veröffentlicht. Die Erreichung der konkreten Ziele umreissen wir nachfolgend.

- Der Funktionsumfang wurde um die Spaziergänge erweitert. Fussgänger:innen können nun auf dieselbe Weise Rundgänge und Spaziergänge erstellen. Bei der Implementierung der Spaziergänge wurden die bekannten Defizite aus der Studienarbeit überarbeitet und eine neue Benutzeroberfläche geschaffen.
- Im Architekturentwurf des Projekts wurde vorgesehen, unterschiedliche Routing Engines verwenden zu können. Die Entscheide der Architektur wurden durch das erfolgreiche Anschliessen zwei weiterer Routing Engines untermauert. Die Wahl der anzuwendenden Engine bleibt noch den Benutzer:innen überlassen, eine Erweiterung diesbezüglich ist naheliegend.
- Für die Personalisierung der Applikation und die geräteübergreifende Synchronisation der individuellen Präferenzen gibt es nun ein Benutzerprofil. Als Vorbedingung dieser Aufgabe galt es TLS und einen Identity Provider einzuführen.
- Die Applikation ist als responsive Webapp unter city-trip-planner.ch als Demo verfügbar.

Die Evaluation der erreichten Ergebnisse erfolgt an dieser Stelle und im Kapitel [results-and-enhancements] in der Projektdokumentation.

4.2. Zusätzliche Features

In diesem Abschnitt werden Funktionalitäten aufgeführt, die die erwarteten Leistungen übersteigen. Die Anregungen für die Erweiterungen entstanden aus Eigeninitiative, im Austausch mit Prof. Stefan Keller oder in Gesprächen mit unserem Partner Estefan Justo.

- Benutzer:innen können nun auch nach Bahnhöfen und nicht nur nach Ortschaften suchen.
- Mithilfe der Funktion *Suche in Region* können Tourist:innen Points of Interest für eine beliebige Ortschaft entdecken. Besteht bereits eine Route, können die neu entdeckten Punkte als Station in die Route aufgenommen werden.
- Möchte man eine Station nicht besuchen, kann diese über das Popup, die Routen-Details oder über die neue Funktion *Stationen bearbeiten* entfernt werden. Entfernte Points of Interest können wiederum der Route hinzugefügt werden.
- Stationen, die sich nicht auf der Route befinden, können für eine bessere Übersicht ausgeblendet werden.
- Neben Einstellungen können auch persönliche Routen im eigenen Profil gespeichert werden.

Zusätzliche Erweiterungen des Benutzerprofils sind denkbar.

- Die Darstellung der Marker für Points of Interest wurde überarbeitet. Ein eigener Service wurde implementiert und lässt vollständige Freiheit bei der Kombination von Icon und Farbe. Zusätzlich werden beieinanderliegende Punkte zusammengefasst und gruppiert dargestellt.
- Benutzer:innen können über die GPS-Lokation ihren eigenen Standort anzeigen lassen und zusätzlich eine Funktion aktivieren, die die Karte laufend auf den eigenen Standort fokussiert.

4.3. Weiterentwicklungen

Nachfolgend werden mögliche Erweiterungen des Projekts City Trip Planner aufgezählt. Die detaillierten Erläuterungen der einzelnen Punkte finden sich im Kapitel [enhancements] in der Projektdokumentation.

- Die Eigenheiten der Routing Engines können für konkrete Anwendungsfälle genutzt werden und in die Suchstrategie einfliessen.
- Neue Routing-Profile können eingeführt werden, um den Benutzer:innen spezifischere Resultate für ihren Anwendungsfall zu liefern.
- Weitere Informationsquellen können eingebunden werden, um zusätzliche Daten und Hintergrundinformationen anzuzeigen.
- Points of Interest können priorisiert und eingeschränkt werden. Durch diese Funktionalität kann das mehrfache Vorkommen einer irrelevanten Station unterbunden werden.
- Benutzer:innen können eigene Kategorien erfassen und Interessen individuell hinzufügen.
- Routen können vordefiniert und katalogisiert veröffentlicht werden. Tourist:innen können eine bestehende Route selektieren und diese besuchen, ohne die Stadt genauer zu kennen.
- Eine Navigation könnte den Benutzer:innen helfen das Ziel zu erreichen. Insbesondere an unbekannten Orten kann diese Funktionalität unterstützend wirken.

Teil II: Projektdokumentation

5. Identity Provider

In diesem Kapitel widmen wir uns der Integration eines geeigneten Identity Provider. Für dessen Ermittlung wurden verschiedene Prototypen gebaut, um die Integration praktisch zu erproben. Die daraus resultierenden Erkenntnisse sollen im Betrieb des Gesamtsystems zum Einsatz kommen und angewendet werden.

5.1. Was ist ein Identity Provider?

Ein Identity Provider (de: *Identitätsanbieter*) bietet einen Dienst für Authentisierung und Autorisierung an. Der Grundgedanke dabei ist, dass derselbe Identity Provider für mehrere Applikationen, sogenannte *Relying Applications*, verwendet werden kann. Dieser Vorgang ist auch bekannt als *Single Sign-On* und wird von verschiedenen, bekannten Anbietern zur Verfügung gestellt. Für diesen Vorgang gibt es verschiedene Protokolle zur Authentisierung. Die bekanntesten und in der Industrie verbreitetsten sind OAuth 2.0, OpenID Connect und SAML.

5.2. Weshalb verwenden wir einen Identity Provider?

Bei Software ist es ratsam auf bewährte Technologien und Standards zu setzten. Besonders hinsichtlich Sicherheit sollten Softwarebibliotheken und Abläufe nicht selber entwickelt werden. Zu gross ist das Risiko, dabei Fehler zu machen und damit Schwachstellen in die Applikation einzubauen. Wir setzen auf Produkte, deren Hauptaufgabe es ist, verschiedene Protokolle als Identity Provider zu unterstützen. Wir wenden die zur Verfügung gestellte Funktionalität eines Produkts an, ohne sie selber zu entwickeln.

Ein weiterer wichtiger Grund ist das Konzept der *User Federation*. Mit einem Identity Provider, welcher User Federation unterstützt, wird eine Schnittstelle geschaffen, um organisationsübergreifend Benutzerdaten zu integrieren. Benutzer:innen aus verschiedenen Systemen können somit auf die Applikation zugreifen, ohne dass das Ursprungssystem das eingesetzte Protokoll zur Autorisierung unterstützen muss.

5.3. Keycloak

Bei der praktischen Ermittlung eines geeigneten Identity Provider hat sich Keycloak vom Rest der getroffenen Auswahl abgesetzt. Obschon keine konkreten Softwarebibliotheken zur Integration mit FastAPI existieren, welche noch aktiv gewartet werden, konnte sich Keycloak durch folgende Aspekte durchsetzen:

- Sehr einfache Installation und Betrieb mit Docker
- Dokumentation mit vielen aktuellen Beispielen und Konfigurationsmöglichkeiten
- Sogenannte Realms welche eine Trennung von verschiedenen Applikationen innerhalb des Identity Provider ermöglichen
- Support diverser Protokolle für Autorisierung und User Federation

5.4. Interaktionsabläufe mit Keycloak

Die Abläufe der Interaktionen mit Keycloak als Identity Provider werden in diesem Abschnitt festgehalten. Sie entstanden bei der Erarbeitung des Prototyps zur Ermittlung der Eignung von Keycloak und wurden anschliessend ausgearbeitet.

5.4.1. Login

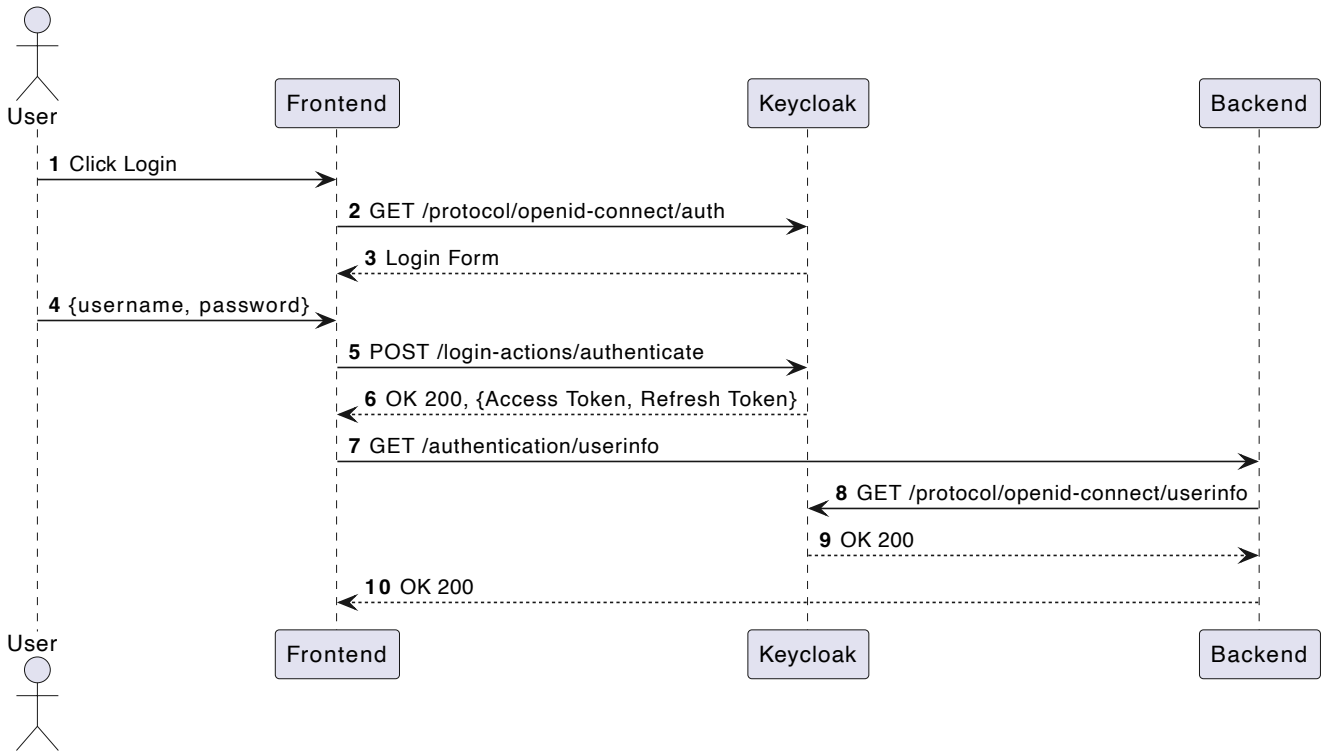

Abbildung 8. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Login

Der Login-Prozess wird durch die Frontend-Applikation und den Browser abgewickelt. Nach einem Klick auf Login leitet die Frontend-Applikation auf das Login-Formular von Keycloak weiter. Anschliessend findet die Authentifizierung durch die Eingabe von Benutzername und Passwort statt. Keycloak antwortet nach erfolgreicher Authentisierung mit den ausgestellten Tokens, welche im Browser zwischengespeichert werden. Bei Anfragen an die Backend-Applikation muss das Token im Authorization-Header übergeben werden. Die Backend-Applikation überprüft die Autorisierung anhand dieses Header.

5.4.2. Unauthorized Request Frontend

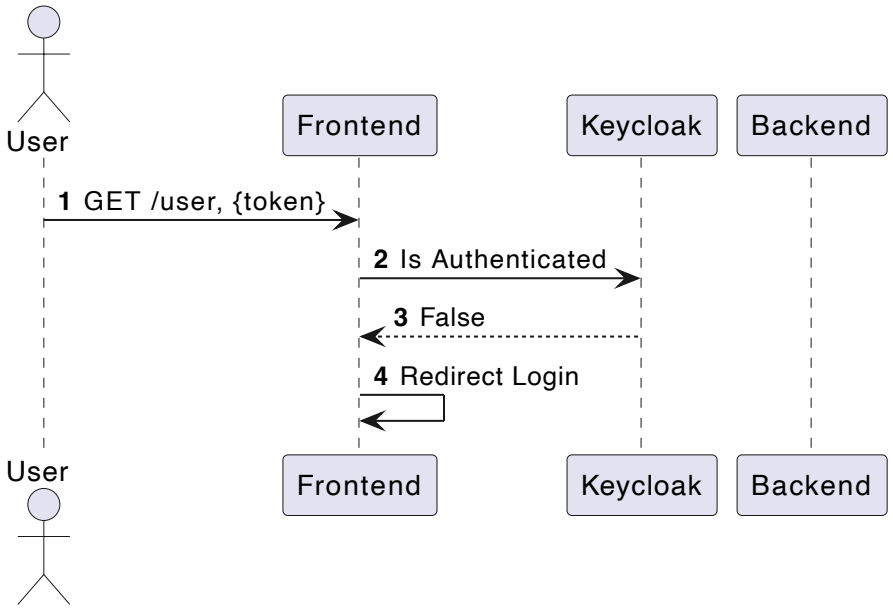

Abbildung 9. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Unauthorized Request Frontend

Die Frontend-Applikation leitet die unautorisierten Anfragen nicht an die Backend-Applikation weiter. Dies dient der Entlastung der Ressourcen des Servers.

Im Frontend besteht Vorwissen, für welche Aktionen man eingeloggt sein muss. Durch dieses Vorwissen können Anfragen, welche eine Authentisierung benötigen, präemptiv zum Login-Formular umgeleitet werden.

5.4.3. Unauthorized Request Backend

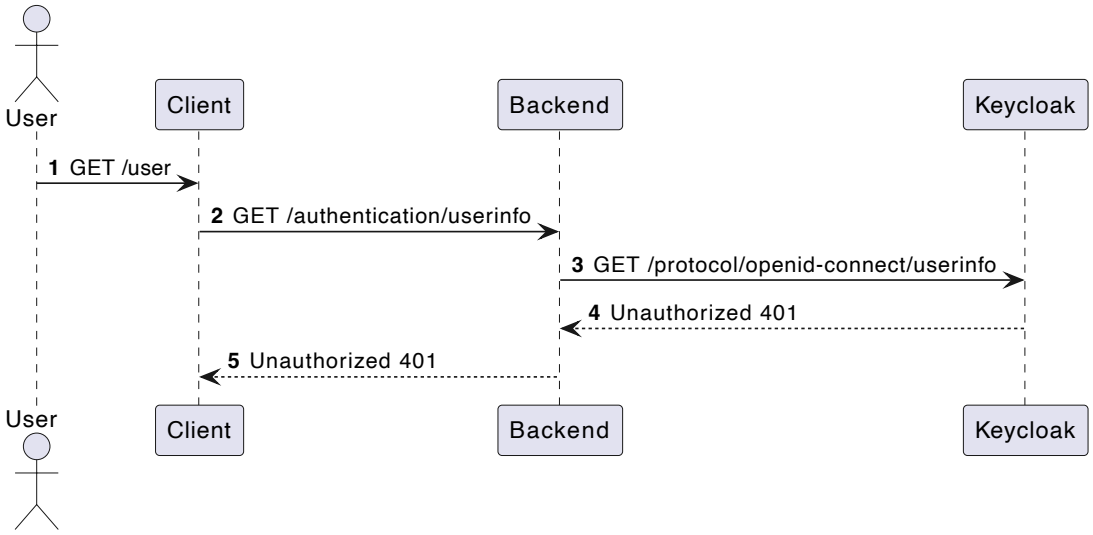

Abbildung 10. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Unauthorized Request Backend

Die Bedienung der Backend-API erfolgt hauptsächlich durch die Frontend-Applikation. Allerdings ist es nicht ausgeschlossen, dass von einem anderen Client Anfragen an die API gesendet werden. Aus diesem Grund muss die Backend-API ebenso einen Schutzmechanismus erhalten, welcher Unbefugten den Zugriff verwehrt.

5.4.4. Forbidden Request Frontend

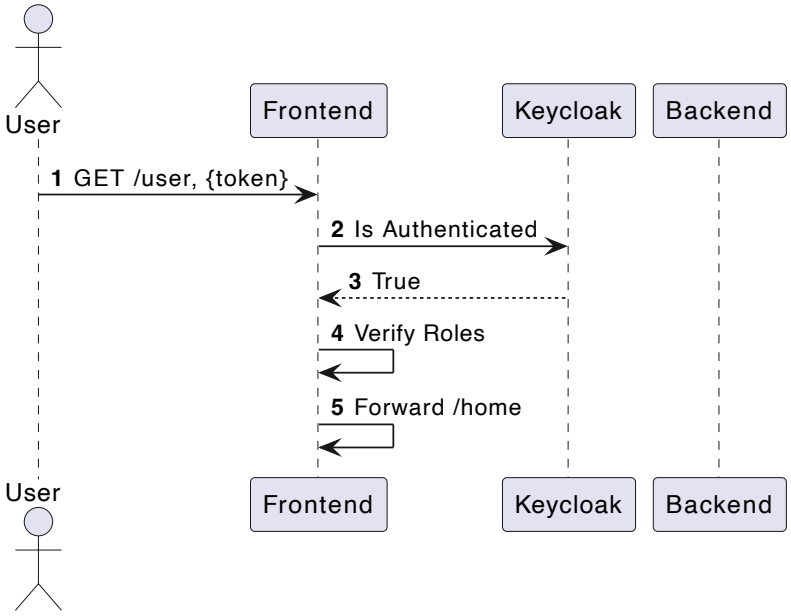

Abbildung 11. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Forbidden Request Frontend

Wie in einem oberen Abschnitt erwähnt, besteht Vorwissen in der Frontend-Applikation, ob eine Autorisierung stattgefunden haben muss und welche Rolle benötigt wird. Dieses Vorwissen wird hier ebenfalls genutzt, um Benutzer:innen auf die Startseite weiterzuleiten, wenn sie nicht die notwendigen Rollen besitzen.

5.4.5. Forbidden Request Backend

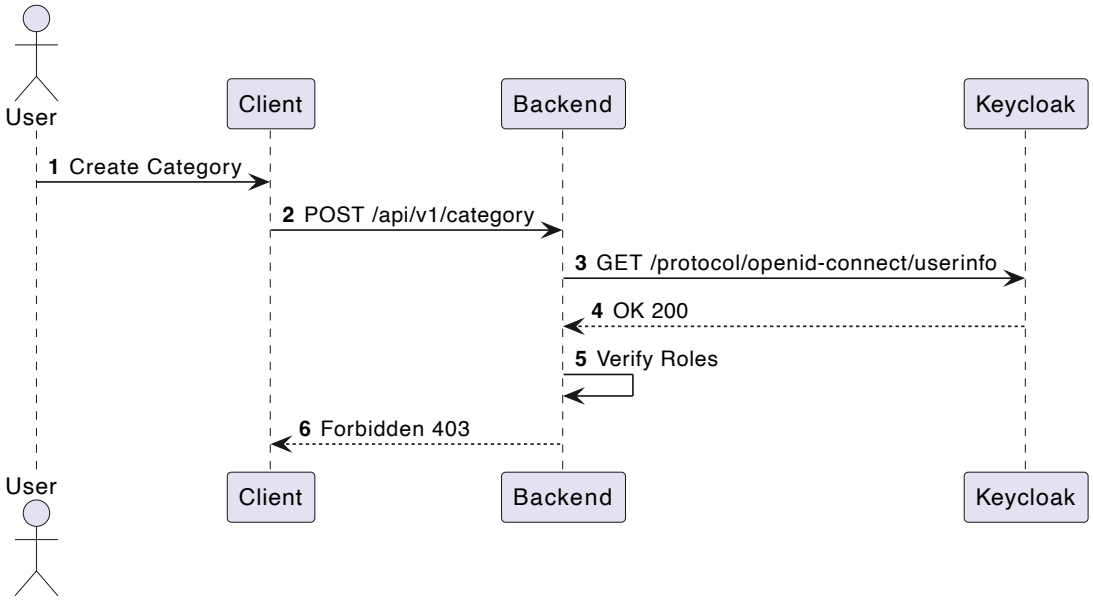

Abbildung 12. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Forbidden Request Backend

In der Backend-Applikation müssen Anfragen von anderen Clients ebenfalls geprüft werden. Dazu wird dem Identity Provider das Token zur Prüfung gesendet. Wenn die Echtheit des Tokens bestätigt ist, werden die darin enthaltenen Rollen überprüft. Fehlt die notwendige Rolle, dann antwortet die Applikation mit dem HTTP-Status 403 Forbidden.

5.4.6. Authorized Request

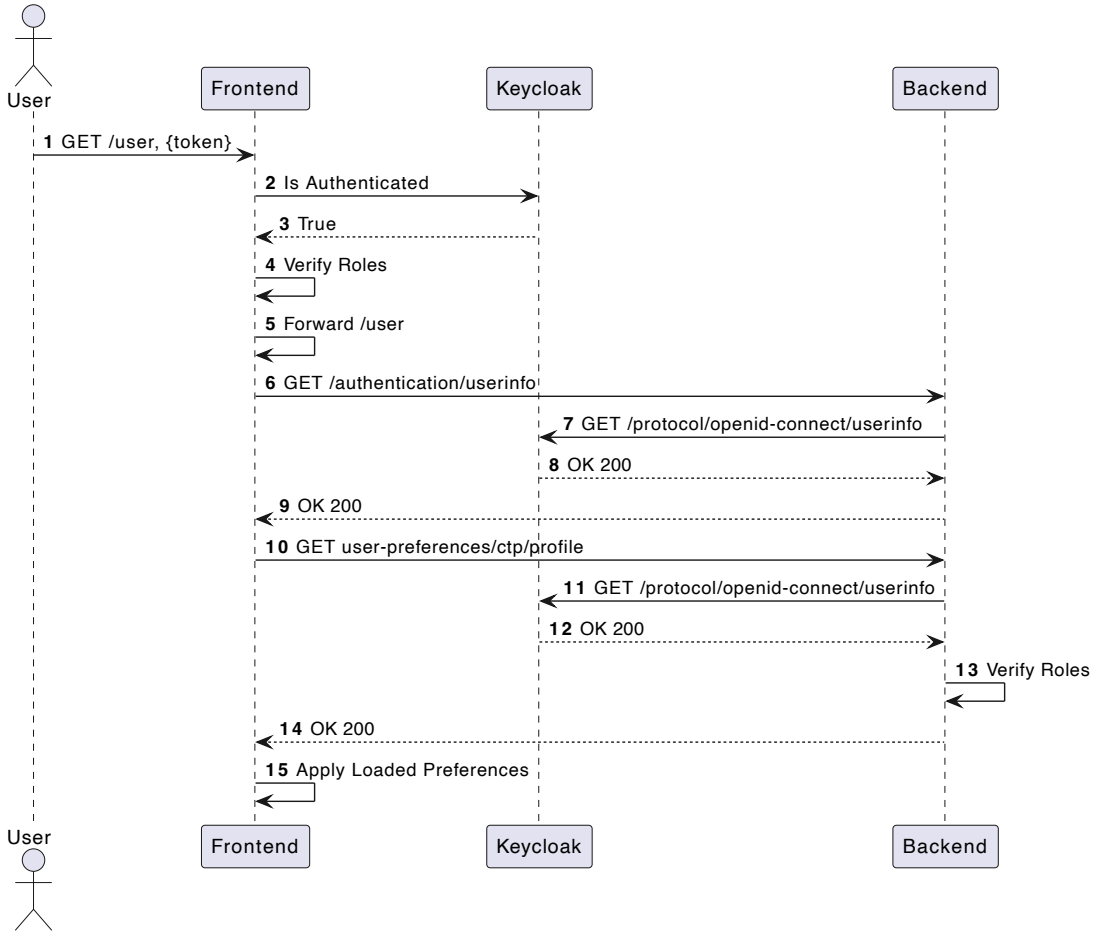

Abbildung 13. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Authorized Request

Dieses Diagramm zeigt den vollständigen Interaktionsablauf für autorisierte Anfragen am Beispiel des Ladens von Benutzerkonfigurationen.

5.4.7. Logout

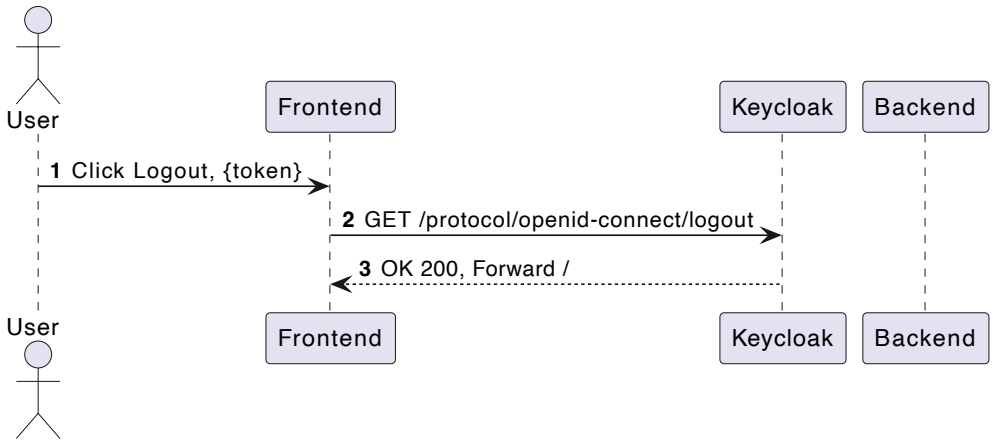

Abbildung 14. UML Sequenzdiagramm [24] Interaktionsablauf Keycloak für Logout

Ebenso wie Login findet Logout nur zwischen Frontend-Applikation, Browser und Keycloak statt. Das Backend erfährt durch den Logout-Prozess keine Interaktion mit der Frontend-Applikation.

6. Projektmanagement

6.1. Vorgehensmodell

Wir verwenden SCRUM+ als Vorgehensmodell für unsere Bachelorarbeit. SCRUM+ wurde uns im Studium vermittelt und ist eine Kombination aus SCRUM und Rational Unified Process. In Projekten, in welchen in einer vorgegebenen Zeit gewisse Ziele erreicht werden müssen, braucht es zusätzlich zu den Iterationen übergeordnete Phasen. Die Abgabe ist an einen fixen Termin gebunden, weshalb wir uns für dieses Vorgehensmodell entschieden haben.

Die in RUP vorgesehenen Iteration gestalten wir mit SCRUM. Diese agile Herangehensweise ermöglicht es, unsere Planung Woche für Woche anzupassen, auf Unvorhergesehenes zu reagieren und Feedback unseres Betreuers, Prof. Stefan Keller, oder unseres Industriepartners, Estefan Justo, einfliessen zu lassen. Die Dauer eines Sprints legen wir auf eine Woche fest und wir werden die wöchentlichen Meetings mit Prof. Stefan Keller und Estefan Justo für Demonstrationen nutzen.

6.2. Involvierte Personen

Person	**Rolle**	**Verantwortlichkeit**
Lukas Grigis	Student, Entwickler	Coding, Dokumentation, Kommunikation
Jan Ruch	Student, Entwickler	Coding, Dokumentation, Kommunikation
Prof. Stefan Keller	Betreuer FH OST	Betreuung, fachliche Unterstützung, Bewertung
Estefan Justo	Industriepartner Schweiz Tourismus	Anforderungen kommunizieren

Tabelle 1. Involvierte Personen

6.3. Aufwandsschätzung

Die Ostschweizer Fachhochschule (OST) vergibt für die Bachelorarbeit 12 ECTS-Punkte pro Student. Laut Hochschulreglement soll der Gegenwert eines ETCS-Punkts rund 30 Arbeitsstunden entsprechen. Daraus errechnet sich ein Total von 720 Stunden Arbeitszeit für die gesamte Arbeit.

Das Projekt ist so geplant, dass alle Hauptaufgaben und Primärziele innerhalb der zur Verfügung stehenden Zeit erreichbar sind. Um das Produkt weiterzubringen und unserer Vorstellung davon gerecht zu werden, sind wir bereit einen Mehraufwand zu leisten. Hierfür können wir aber keine konkrete Aufwandschätzung abgeben, da diese Sekundärziele noch nicht im Detail bekannt sind. Das gewählte Vorgehensmodell erlaubt diese Unsicherheit und stellt sicher, dass zum Abgabezeitpunkt alle notwendigen Artefakte eingereicht werden können.

6.4. Meilensteine

Die Meilensteine für diese Bachelorarbeiten definieren sich aus den Primärzielen der Aufgabenstellung und den vom Studiengang geforderten Artefakten. Die Inception und Elaboration Phase werden wir im Vergleich zur Studienarbeit eher kurz halten können, da wir an einem existierenden Projekt weiterarbeiten.

Die Meilensteine der Primärziele wurden so definiert, dass sie unabhängig und individuell erreicht werden können. Allerdings können wir mit den Sekundärzielen erst beginnen, wenn die technischen Voraussetzungen erfüllt sind. Diese Abhängigkeit wird in der angefügten Grafik nochmals verdeutlicht.

Die letzte Woche das Jahres 2022, konkret vom 24.12.2022 - 01.01.2023, gelten als Weihnachtsferien und fliessen somit nicht in unsere Projektplanung ein. Die beiden darauffolgenden Wochen werden mit Semesterwoche 15 und 16 bezeichnet.

Phase	Meilenstein	Fälligkeit
Inception	M1: Kick-Off	Sonntag, 25.09.2022
Elaboration	M2: Grobplanung des Projekts	Sonntag, 02.10.2022
Construction	M3: Anschluss neuer Routing Engines	Sonntag, 06.11.2022
Construction	M4: Security und User-Login	Sonntag, 20.11.2022
Construction	M5: Implementation Spaziergang	Sonntag, 20.11.2022
Construction	M6: Angleichung/Migration Rundgang	Sonntag, 27.11.2022
Construction	M7: Implementation Sekundärziele	Sonntag, 11.12.2022
Transition	M8: Dokumentation	Freitag, 23.12.2022
Transition	M9: Abgabe der Bachelorarbeit	Freitag, 13.01.2023

Tabelle 2. Zuordnung der Meilensteine in die Projektphasen

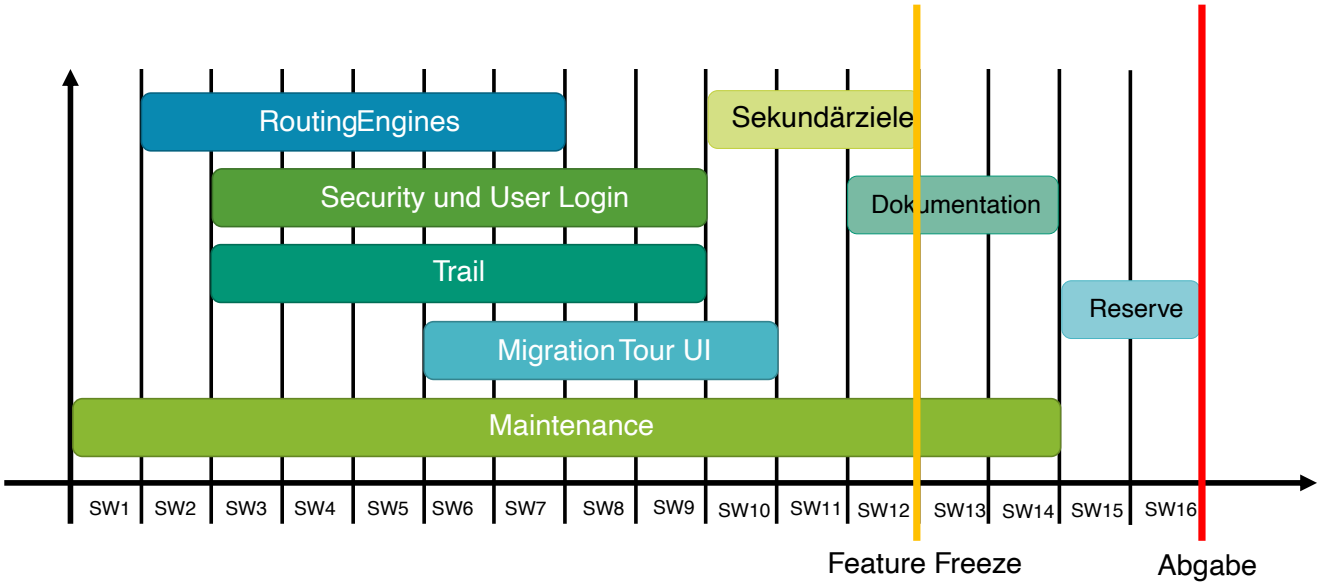

Abbildung 15. Planung der Meilensteine während des Semesters

6.5. Prototypen und Releases

Die Hochschule hat uns einen Server zur Verfügung gestellt, welchen wir für regelmässige Deployments nutzen. Sobald eine Änderung an der Software in den Master-Branch eingepflegt wird, gilt sie als release-fähig und kann ausgeliefert werden. Erfüllt eine Änderung diese Bedingung nicht, dann wird sie nicht in den Master eingepflegt und somit nicht veröffentlicht.

Die aktuelle Version der Software ist unter city-trip-planner.ch veröffentlicht. Für die Abgabe dieser Bachelorarbeit planen wir die Version `1.0.0`. Die Source Code Repositories werden mit derselben Version getaggt.

6.6. Risiken

Im folgenden Abschnitt widmen wir uns den möglichen Risiken für dieses Projekt. Einige der Risiken unterscheiden sich unwesentlich von denjenigen, welche wir bereits in der Studienarbeit festgehalten haben. Die Eintrittswahrscheinlichkeit ist jedoch geringer, da wir uns bereits mit dem Thema befasst haben. Deshalb können wir die Eintrittswahrscheinlichkeit von ähnlichen Risiken nach unten korrigieren.

Die aufgelisteten Risiken müssen fortlaufend neu beurteilt werden. Dies ist notwendig, um auf die möglichen Probleme frühzeitig zu reagieren und Alternativen auszuarbeiten. Das Resultat dieser Risikobeurteilung soll einen wesentlichen Teil zum Abschluss dieser Bachelorarbeit beitragen.

6.6.1. R1: Unterschätzte Komplexität

Beschreibung	Die Komplexität einzelner Anforderungen oder neuer Technologien wurde unterschätzt und die Arbeit wird deshalb nicht, oder nur fehlerhaft implementiert.
Eintrittswahr-scheinlichkeit	30%
Gewichteter Schaden	8
Vorbeugung	Modularisierung
Verhalten beim Eintreffen	Analysieren, Vereinfachen oder Weglassen der Anforderung und eine konzeptionelle Lösung erarbeiten

Tabelle 3. Beschreibung von R1: Unterschätzte Komplexität

6.6.2. R2: Fehlerhaftes Exception Handling

Beschreibung	Exceptions werden nicht korrekt abgearbeitet, das Mapping auf interne Fehlercodes ist fehlerhaft, oder das Mapping auf HTTP Statuscodes ist fehlerhaft.
Eintrittswahr-scheinlichkeit	20%
Gewichteter Schaden	4
Vorbeugung	Definition von Defaults, Testing
Verhalten beim Eintreffen	Exception Handling überarbeiten, Registration der korrekten Statuscodes

Tabelle 4. Beschreibung von R2: Fehlerhaftes Exception Handling

6.6.3. R3: Mangelhafte Kommunikation

Beschreibung	Die Kommunikation mit den Betreuern oder dem Industriepartner scheitert.
Eintrittswahr-scheinlichkeit	5%
Gewichteter Schaden	10
Vorbeugung	Frühzeitiges Eingreifen durch die Studenten bei Unklarheiten
Verhalten beim Eintreffen	Besprechung mit den Betreuern, notfalls ein Gespräch mit dem Studiengangleiter

Tabelle 5. Beschreibung von R3: Mangelhafte Kommunikation

6.6.4. R4: Personeller Ausfall

Beschreibung	Eine Person fällt aus gesundheitlichen Gründen aus, welche eine Verzögerung oder Verschmälerung der Arbeit mit sich zieht.
Eintrittswahr-scheinlichkeit	50%
Gewichteter Schaden	6
Vorbeugung	Gute Kommunikation im Voraus, da sich Risiko kaum verhindern lässt (höhere Gewalt)
Verhalten beim Eintreffen	Besprechung mit den Betreuern, Arbeitspakete und Anforderungen neu definieren

Tabelle 6. Beschreibung von R4: Personeller Ausfall

6.6.5. R5: Unflexible Software-Architektur

Beschreibung	Die bereits entworfene Software-Architektur bietet nicht die geforderte Flexibilität, um die Software zu erweitern.
Eintrittswahr-scheinlichkeit	20%
Gewichteter Schaden	6
Vorbeugung	Design Reviews
Verhalten beim Eintreffen	Problemanalyse, Aufwandschätzung, Korrektur, Neubeurteilung der Machbarkeit

Tabelle 7. Beschreibung von R5: Unflexible Software-Architektur

6.6.6. R6: Technologische Schnittstellen

Beschreibung	Die technischen Schnittstellen von eingesetzten Technologien ändern sich, was wiederum die Kommunikation verunmöglicht.
Eintrittswahr-scheinlichkeit	10%
Gewichteter Schaden	4
Vorbeugung	Frühe Upgrades von Softwarebibliotheken und Code Maintenance
Verhalten beim Eintreffen	Problemanalyse, Aufwandschätzung, Korrektur, Neubeurteilung der Machbarkeit

Tabelle 8. Beschreibung von R6: Technologische Schnittstellen

6.6.7. R7: Systemausfall

Beschreibung	Die Demo-Umgebung fällt aus.
Eintrittswahr-scheinlichkeit	10%
Gewichteter Schaden	5
Vorbeugung	*Infrastructure as Code* damit auf neue Umgebung gewechselt werden kann und Backup
Verhalten beim Eintreffen	Lokale Ausführung für Demonstrationen und Präsentationen

Tabelle 9. Beschreibung von R7: Systemausfall

6.6.8. R8: Gold Plating

Beschreibung	Eigene Anforderungen, um das Produkt zu verbessern, werden überbewertet, wodurch verlangte Anforderungen nicht umgesetzt werden.
Eintrittswahr-scheinlichkeit	20%
Gewichteter Schaden	5
Vorbeugung	Wöchentliche Repriorisierung während Sprint Planning
Verhalten beim Eintreffen	Abbruch der Tätigkeit und Auslegen des Fokus auf geforderte Funktionalität

Tabelle 10. Beschreibung von R8: Gold Plating

6.6.9. Neubewertung nach Meilenstein 2

Nach Abschluss der Grobplanung befürchten wir, dass die Eintrittswahrscheinlichkeit von Risiko R1 und R5 angestiegen ist. Besonders im Frontend-Bereich könnte die Einführung von NgRx und der Umbau der Karte zu grösseren Änderungen führen. Allerdings können durch diese Einführung die Eintrittswahrscheinlichkeit von Risiko R2 und R6 gesenkt werden.

6.6.10. Neubewertung nach Meilenstein 3

Mit dem Anschluss von zwei neuen Routing Engines steigt der gewichtete Schaden von R6 an. Wenn eine der Routing Engines GeoJSON nicht weiter unterstützt oder gar entfernt, muss die Applikation angepasst werden. Ausserdem besitzt jede Routing Engine ein proprietäres Format für die Antwort. Es sind gezwungenermassen Änderungen der Applikation notwendig, sobald diese Schnittstellen ändern.

6.6.11. Neubewertung nach Meilenstein 4

Die Eintrittswahrscheinlichkeit von Risiko R5 sinkt drastisch für diese Bachelorarbeit. Die Gesamtarchitektur erlaubt das Anbinden zusätzlicher Systemen für Security und Login.

6.6.12. Neubewertung nach Meilenstein 5

Mit dem Erreichen des 5. Meilensteins minimieren sich gleich mehrere Risiken. Die Eintrittswahrscheinlichkeit für R1, R3, und R5 nehmen weiter ab. Diese Beurteilung begründen wir damit, dass wir nun die Primärziele für diese Arbeit erreicht haben. Dies wiederum hat zur Folge, dass das Risiko R8 von grösserer Bedeutung für die Umsetzung der Sekundärziele wird.

6.6.13. Neubewertung vor Meilenstein 7

Aufgrund des ausbleibenden Kontakts mit Schweiz Tourismus beginnen wir allmählich mit der Ausarbeitung von eigenen Business-Anforderungen. Ausser der initialen Zusage für eine Zusammenarbeit, haben wir noch keine weiteren Angaben. Wir werden diese Anforderungen mit dem Industriepartner besprechen, sobald der Kontakt steht. Sollte der Kontakt wider Erwarten abbrechen, würden wir unsere eigenen Anforderungen umsetzen. Die Eintrittswahrscheinlichkeit des Risikos R3 erhöht sich.

6.6.14. Neubewertung nach Meilenstein 7

Estefan Justo von Schweiz Tourismus hat sich in den vergangenen Wochen bei uns gemeldet und sich sehr stark um eine aktive Zusammenarbeit bemüht. Wir haben sein Engagement und seine konstruktiven Rückmeldungen sehr geschätzt. Entsprechend hat sich die Eintrittswahrscheinlichkeit von Risiko R3 wieder normalisiert.

6.6.15. Anmerkung nach Meilenstein 8

Jan Ruch ist krankheitsbedingt für zwei Tage ausgefallen, was einen Einfluss auf Risiko R4 hatte. Allerdings hatten die beiden Tagen keinen schwerwiegenden Einfluss auf den Projektfortschritt.

7. Projektmonitoring

Dieses Kapitel befasst sich mit Auswertungen der Projektarbeit. Im ersten Abschnitt wird der effektive Arbeitsaufwand offengelegt und Details erläutert. Im zweiten Abschnitt geht es um Codestatistiken für die von uns erstellten Komponenten.

7.1. Effektiver Aufwand

Die gesamte Projektplanung haben wir in YouTrack [25] geführt. Der initiale Mehraufwand für das Aufsetzen dieses Tools war in den darauffolgenden Wochen eine gute Investition. Dadurch konnten wir alle offenen Aufgaben als Tickets erfassen und unsere agile Projektplanung fortführen.

YouTrack stellt ebenfalls eine Möglichkeit zur Verfügung, um Zeitaufwände zu verbuchen. Allerdings haben wir die betroffenen Arbeitstypen manchmal unterschiedlich und nicht konsequent genug genutzt. Aus diesem Grund haben wir eine Excel Datei erstellt und die totalen Aufwände übersichtlich gegliedert.

Totale Dauer	897h 45m
Development	**521h 45m**
Jan Ruch	197h 10m
Lukas Grigis	324h 35m
Investigation	**148h 20m**
Jan Ruch	101h 45m
Lukas Grigis	46h 35m
Testing	**54h 05m**
Jan Ruch	29h 45m
Lukas Grigis	24h 20m
Documentation	**173h 35m**
Jan Ruch	94h 15m
Lukas Grigis	79h 20m

Abbildung 16. Zeiterfassung nach Kategorie und Student

Den zeitlichen Mehraufwand von rund 180 Stunden, gegenüber den veranschlagten 720 Stunden, haben wir für die Umsetzung von zusätzlichen Funktionalitäten aufgewendet. Die Freude an der Arbeit und unser persönlicher Ehrgeiz haben uns dazu bewogen, mehr Zeit zu investieren und einen zusätzlichen Aufwand zu leisten.

Unter der Kategorie *Investigation* haben wir rund 20 Stunden pro Student für wöchentliche Meetings verbucht.

7.2. Codestatistik

In diesem Abschnitt werten wir die Codestatistiken der einzelnen Teile der Applikation aus. Für die Auswertung nutzen wir das *Statistic-Plugin* [26].

7.2.1. Backend

Extension	Count	Lines	Lines AVG	Lines CODE
md *(MD files)*	1x	105	105	62
py *(PY files)*	551x	20332	36	16901
sh *(SH files)*	3x	82	27	52
txt *(Text files)*	3x	152	50	82
yml *(YML files)*	3x	590	196	453

Abbildung 17. Statistische Auswertung der Dateien und Codezeilen im Backend

Die wichtigsten Dateien im Backend sind die Python-Dateien mit der Endung *.py*. Die Applikation umfasst 551 Dateien mit insgesamt 16'901 Zeilen Code. Darüber hinaus wichtig sind die drei YML-Dateien, die die gesamte Konfiguration beinhalten.

7.2.2. Frontend

Extension	Count	Lines	Lines AVG	Lines CODE
html *(HTML files)*	32x	1431	44	1309
jpg *(JPG images)*	2x	11767	5883	11537
md *(MD files)*	1x	81	81	45
png *(PNG images)*	9x	1544	171	1498
scss *(SCSS files)*	31x	414	13	337
sh *(SH files)*	1x	46	46	27
svg *(SVG files)*	8x	169	21	169
ts *(TS files)*	239x	16126	67	12810
txt *(Text files)*	1x	28	28	20

Abbildung 18. Statistische Auswertung der Dateien und Codezeilen im Frontend

Die wichtigsten Dateien im Frontend sind die üblichen Web-Dateien mit den Endungen *.html*, *.ts* und *.scss*. Die Applikation umfasst 302 Dateien mit insgesamt 14'456 Zeilen Code. Weitere wichtige Dateien für die Darstellung der Webseite sind die Bilder mit den Endungen *.jpg*, *.png* und *.svg*, davon gibt es nochmals 19.

8. Softwaredokumentation

In diesem Kapitel werden die wichtigsten Technologien und Umsysteme dokumentiert, welche für dieses Projekt eingesetzt und verwendet werden.

8.1. Eingesetzte Technologien

8.1.1. Datenbank

Technologie	Version	Beschreibung
PostgreSQL [27]	14.6	Relationale Datenbank für Datenhaltung
PostGIS [28]	3.3	Erweiterung für PostgreSQL
HSTORE [29]	14.6	Erweiterung für PostgreSQL

Tabelle 11. Eingesetzte Technologien für Datenbank

8.1.2. Backend

Technologie	Version	Beschreibung
Python [30]	3.11	Programmiersprache in Backend
FastAPI [31]	0.88.0	Application Framework
SQLAlchemy [32]	1.4.46	Library für objektrelationales Mapping
alembic [33]	1.9.1	Library für Migrationen
OpenAPI [34]	3.0.2	Spezifikation der API

Tabelle 12. Eingesetzte Technologien für Backend

8.1.3. Frontend

Technologie	Version	Beschreibung
TypeScript [35]	4.8.4	Programmiersprache in Frontend
Angular [36]	15.0.4	Application Framework
PrimeNG [37]	15.0.0	UI Library
FontAwesome [38]	6.2.1	Icon Library
RxJs [39]	7.8.0	Library für Reactive Programming
NgRx [40]	15.1.0	Library für State Management
Leaflet [41]	1.9.3	Library für interaktive Karten

Tabelle 13. Eingesetzte Technologien für Frontend

8.1.4. Routing Engines

Name	Version
Project Open Source Routing Machine [42]	5.25.0
GraphHopper [43]	6.2
openrouteservice [44]	6.8.0

Tabelle 14. Eingesetzte Routing Engines

8.1.5. Andere Technologien

Name	Version	Beschreibung
Keycloak [45]	20.0.2	Identity Provider
Traefik [46]	2.9.6	Reverse Proxy
Docker [47]	20.10.22	Containervirtualisierung
Geofabrik [48]	Tagesaktuell	Download-Server für OpenStreetMap -Daten

Tabelle 15. Weitere Technologien, welche im Projekt eingesetzt werden.

8.2. Installation

Die aktuellsten Installationsinstruktionen befinden sich git-Repository Docker.

8.3. Setup

Um das Projekt weiterzuentwickeln, befindet sich in jedem Repository eine README-Datei mit den benötigten Befehlen.

- Backend
- Frontend
- Docker

Appendix A: Abgabeumfang

Die in der Aufgabenstellung und die des Studiengangs geforderten Abgaben werden folgendermassen abgegeben:

A.1. Archivierung

- Vom Studiengang geforderte Archivdatei als ZIP
- Eigenständigkeitserklärung als PDF
- Urheber- und Nutzungsrechte als PDF
- A0 Poster als PDF

A.2. E-Prints

- Plain-Text Abstract als TXT-Datei
- Gekürzter Bericht als PDF
- Einverständniserklärung zur Veröffentlichung des gekürzten Berichts

A.3. Broschüre

- Abstract für Broschüre über absolvierte Studien- und Bachelorarbeiten

A.4. Code Repositories

- https://gitlab.ost.ch/city-trip-planner/frontend
- https://gitlab.ost.ch/city-trip-planner/backend
- https://gitlab.ost.ch/city-trip-planner/docker
- https://gitlab.ost.ch/city-trip-planner/documentation

A.5. Lauffähige Applikation

Die Webapplikation kann wie in https://gitlab.ost.ch/city-trip-planner/docker beschrieben installiert werden.

Die lauffähige Applikation kann öffentlich über die Domain city-trip-planner.ch erreicht werden.

> Aktuell haben nur Prof. Stefan Keller, Jan Ruch und Lukas Grigis Zugriff auf die Code Repositories der lauffähigen Applikation. (Stand 13.01.2023)

A.6. Physische Version

Die vorliegende Bachelorarbeit wird als gebundene Version an Prof. Stefan Keller abgegeben.

Appendix B: Persönliche Kurzberichte zu Prototypen für Ermittlung von Identity Provider

Für die Ermittlung des zu integrierenden Identity Provider wurden verschiedene Prototypen erstellt. Mit der Anwendung eines jeweiligen Identity Provider werden praktische Erfahrungen gesammelt, welche bei der konkreten Umsetzung wiederverwendet werden können. Die Erkenntnisse und aufgetretenen Probleme werden im folgenden in der Form von persönlichen Berichten festgehalten. Sie beschreiben die persönliche Sicht und beziehen sich auf die Integration mit den in dieser Arbeit verwendeten Technologien.

> Wir möchten mit Nachdruck darauf hinweisen, dass wir die verschiedenen Anbieter und ihre Produkte nicht allgemeingültig bewerten oder beurteilen. Die Berichte sind nicht repräsentativ und eine Beurteilung der Eignung muss für andere Projekte individuell vorgenommen werden.

B.1. Gluu

B.1.1. Lizenz

Gluu [49] ist ein Open-Source-Projekt, welches unter MIT eigene Projekte veröffentlicht hat [50].

B.1.2. Dokumentation

Gluu präsentiert sich mit einer umfangreichen Dokumentation und vielen Erklärungen. Die Installation und Inbetriebnahme ist gut beschrieben, allerdings hatten wir im Verlaufe der Evaluation einige Probleme, aufgrund derer wir uns nicht weiter mit Gluu auseinandersetzten. Die Dokumentation gab zu wenig Aufschluss darüber, wie Gluu als Identity Provider, mit den im Projekt eingesetzten Technologien, zu verwenden ist.

B.1.3. Features

Folgende Features sind uns auf den ersten Blick überaus positiv aufgefallen:

- OpenID Connect und OAuth 2.0 Schnittstellen
- Umfangreiche On-Premise Installation

B.1.4. Implementation und Schwierigkeiten

Die Installation von Gluu schien denkbar einfach, entpuppte sich allerdings als problematisch. Wir haben insgesamt drei Versuche unternommen und viel Zeit in die Fehlersuche investiert, bis wir uns gegen einen Einsatz von Gluu entschieden haben. Nachfolgend gehen wir auf die unterschiedlichen Ansätze und ihre jeweiligen Schwierigkeiten kurz ein.

Den ersten Versuch haben wir auf einem Windows-PC unternommen. Dieser PC wird seit Beginn der Arbeit für die Entwicklung und das lokale Testen der Applikation verwendet. Auch Docker respektive Docker Compose funktioniert, für unsere Projektarchitektur, einwandfrei auf diesem Gerät. Auf eben diesem Gerät konnte die Docker Compose Datei von Gluu gar nicht erst ausgeführt werden. Was genau zu diesem

Fehlverhalten geführt hat, konnten wir nicht mit Bestimmtheit eruieren. Mutmasslich war die Syntax von Dateien, Ordnern oder Volumes inkorrekt. Dies hat dazu geführt, dass Parameter nicht gelesen werden konnten und ein Eintrag für `ds-config-attribute` gefehlt hat.

Aufgrund der Komplexität des Fehlers haben wir uns entschieden die Installation auf einem zweiten Gerät zu starten. Die Wahl fiel auf ein Windows-Notebook, ebenfalls mit Docker, dass als Zweitgerät für die Entwicklung der Arbeit fungierte. Der Server konnte nun erfolgreich gestartet werden, war dann allerdings nicht über den Browser erreichbar. Konfigurationsänderungen und Anpassungen an der URL brachten jedoch nicht den gewünschten Erfolg.

Angespornt von diesem Zwischenerfolg haben wir uns mit einem Dual-Boot Ubuntu-Notebook hingesetzt. Um allfällige Komplikationen mit Docker auszuschliessen, installierten wir den Gluu Server direkt auf das Ubuntu Betriebssystem. Die Installation war einfach und hat ohne weitere Probleme funktioniert. Dann ergab sich das gleiche Problem wie bereits zuvor auf Windows, der Server konnte erfolgreich gestartet, jedoch nicht über den Browser erreicht werden. Auch unterschiedliche Konfigurationen der IP-Adresse bewirkten keinen Unterschied.

Aufgrund dieser Schwierigkeiten haben wir uns gegen die Verwendung von Gluu entschieden und nach Alternativen gesucht.

B.2. Keycloak

B.2.1. Lizenz

Keycloak ist ein Open-Source-Projekt und kann unter der Apache-Lizenz 2.0 genutzt werden [51].

B.2.2. Dokumentation

Die Dokumentation bietet eine Schritt-für-Schritt-Anleitung, um Keycloak mit Docker lokal in Betrieb zu nehmen. Darüber hinaus gibt es auch einige Beschreibungen und Beispiele für die Integration in eine Applikation. Aus unserer Erfahrung sind kleine, wichtige Details allerdings manchmal schwer zu finden oder einfach zu übersehen.

B.2.3. Features

Folgende Features sind uns auf den ersten Blick überaus positiv aufgefallen:

- Ein Admin Panel für die einfache Bewirtschaftung der Client- und Benutzerdaten
- Konfigurierbare Realms für die Anbindung mehrerer Instanzen auf einen Identity Provider
- Ein Client-Secret für den sicheren Datenaustausch zwischen Backend und Identity Provider

B.2.4. Implementation und Schwierigkeiten

Für die Integration von Keycloak in Python gibt es diverse Software-Bibliotheken. Dieses sind allerdings schlecht gewartet und funktionieren nur teilweise. Eine der grossen Herausforderungen war es, funktionierende Beispiele zu finden und zu testen. Die Dokumentationen sind leider nicht aktuell und somit unbrauchbar.

Daraufhin haben wir die Schnittstellendokumentation von Keycloak so aufgearbeitet, dass wir in der Lage waren, eine eigene Implementation für die Interaktionen mit Keycloak umzusetzen. Wir verwenden dazu die von Keycloak zur Verfügung gestellten OpenID Connect-Schnittstelle. Die Details der Integration werden im Kapitel Identity Provider ausgeführt.

Die Implementation ist ausreichend für die aktuelle Verwendung von Keycloak. Sie muss allerdings selbst gewartet und getestet werden. Eine gewartete Software-Bibliothek würde dies vereinfachen und wir hätten viel Zeit einsparen können. Mit den getroffenen Massnahmen konnte ein Proof of Concept erstellt und für die Arbeit wiederverwendet werden.

B.3. OpenIAM

B.3.1. Lizenz

OpenIAM [52] bietet eine gratis Community Edition an, ist aber auch Anbieter eines kostenpflichtigen Modells für Firmen.

B.3.2. Dokumentation

Die Dokumentation selbst ist leicht auffindbar, einfach zu verstehen und sehr umfangreich.

B.3.3. Features

Folgende Features sind uns auf den ersten Blick überaus positiv aufgefallen.

- OpenID Connect und OAuth 2.0 Schnittstellen
- Ein einfaches Setup
- Umfangreiche Business Integration

B.3.4. Implementation und Schwierigkeiten

Begonnen haben wir auch bei OpenIAM mit dem Einlesen in die Dokumentation. Geschuldet dem grossen Umfang von OpenIAM dauerte das Einlesen und das Erstellen eines ersten Proof of Concept massiv länger, als zu Beginn erwartet. Um den Proof of Concept zu erstellen, haben wir als Erstes eine Installation mittels Docker vorgenommen. Die notwendigen Schritte waren sehr gut dokumentiert und denkbar einfach. Zusätzlich musste noch ein Verzeichnis und ein Shell-Skript heruntergeladen werden.

Nach erfolgreicher Installation musste das Shell-Skript genutzt werden, um OpenIAM zu starten. Den Logs nach zu urteilen hat das Starten der Services erfolgreich funktioniert, leider war aber auch bei diesem Produkt das Web-Interface nicht zu erreichen. Zu diesem Zeitpunkt haben wir ernsthaft an unserem Docker-Setup und an den lokalen Einstellungen gezweifelt. Zusätzlich wurde uns nun auch bewusst, dass OpenIAM ein massives, umfangreiches Produkt ist, welches die Komplexität unseres kleinen Projektes weit übersteigt. Bereits die Vielzahl an erstellten Docker-Containern bringt zum Ausdruck, welche Möglichkeiten diese Software bietet. Schlussendlich wurde uns nun auch bewusst, weshalb OpenIAM eine derart umfangreiche Business Integration anbietet und oftmals in Grossprojekten vertreten ist. Aus den eben genannten, aber auch aus zeitlichen Gründen haben wir uns gegen weitere Integrationsversuche entschieden und direkt nach alternativen Lösungen gesucht.

B.4. SuperTokens

B.4.1. Lizenz

SuperTokens [53] ist ein Open-Source-Projekt und kann unter Apache 2.0 verwendet werden [54].

B.4.2. Dokumentation

SuperTokens bietet eine hervorragende Dokumentation. Die Beispiele auf deren Webseite sind bereits vorkonfiguriert für verschiedenste Programmiersprachen, respektive Frameworks, und gut beschrieben. Ein Proof of Concept war somit schnell erstellt.

B.4.3. Features

Folgende Features sind uns auf den ersten Blick überaus positiv aufgefallen:

- SuperTokens bietet vorgefertigte Recipes an
- Backend und Frontend Integration sind sehr einfach gestaltet
- SuperTokens nutzt einen API-Key für den sicheren Datenaustausch zwischen Backend und Identity Provider

B.4.4. Implementation und Schwierigkeiten

Aufgrund der einsteigerfreundlichen Dokumentation haben wir uns schnell zu SuperTokens hinreissen lassen. Nach kurzem Einlesen haben wir unser Backend und Frontend entsprechend aufgesetzt und SuperTokens lokal in einem Docker Container gestartet. Zu diesem Zeitpunkt haben wir in Kauf genommen, dass SuperTokens ebenfalls eine eigene Implementation der OAuth 2.0 und OpenID Connect Protokolle nutzt. Somit wird allerdings der Identity Provider kaum austauschbar (Vendor Lock-In).

Während der Implementation sind einige Unklarheit aufgetreten, die wir mit dem Support besprochen haben. Dieser hat schnell und professionell reagiert, alle Probleme konnten innerhalb eines halben Tages beseitigt werden. Wir haben erfahren, dass ein initialer Benutzer über die Sign-Up Funktion programmatisch erstellt werden kann. Dafür kann die Sign-Up Methode überschrieben und eine Benutzerrolle zugewiesen werden. Leider gibt es zurzeit noch keine Admin-Konsole, diese ist jedoch in Entwicklung.

Ein Nachteil der vielen Recipes wurde uns während der Implementation bewusst. Wir hatten keine funktionierende Verbindung zwischen Frontend und Backend. Grund dafür war, dass wir im Frontend die Konfiguration *ThirdPartyEmailPassword* und im Backend die Konfiguration *EmailPassword* verwendet hatten. Dieser Fehler ist uns wohl beim Öffnen der verschiedenen Dokumentation unterlaufen. Wichtig ist es, ein Recipe von Anfang bis zum Schluss einheitlich zu verwenden.

Nach der Fertigstellung des Prototyps und dem erfolgreichen Testen des Proof of Concept stellten wir fest, dass SuperTokens Multi-Tenancy nicht unterstützt und unterschiedliche Sessions verwendet. Für die von uns angestrebte horizontale Skalierung sind zustandslose Autorisierungsmethoden besser geeignet. Diese Erkenntnis war für uns untragbar, deshalb haben wir entschieden auf SuperTokens zu verzichten.

B.4.5. Möglicher Authentication Flow

In diesem Abschnitt werfen wir einen kurzen Blick auf den Ablauf von Anfragen innerhalb von SuperTokens. Die nachfolgenden Diagramme wurden anhand des Proof of Concept, der SuperTokens Dokumentation und persönlicher Auffassung erstellt. Diese Diagramme dienten in erster Linie dem eigenen Verständnis für diesen Identity Provider.

Die Diagramme sind aus Gründen der Vollständigkeit und Nachvollziehbarkeit angehängt. Es kann sein, dass die Diagramme von einer tatsächlichen Integration abweichen. Als Diskussionsgrundlage für einen Gedankenaustausch waren die Diagramme ausreichend.

Login

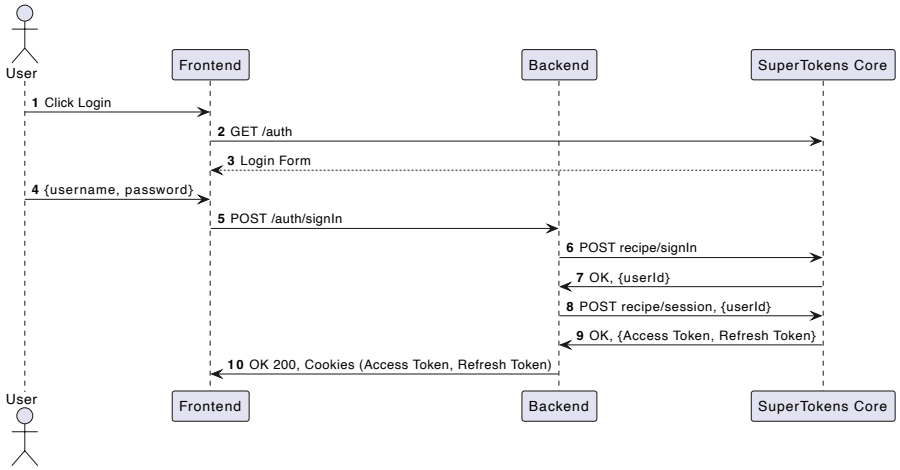

Abbildung 19. UML Sequenzdiagramm [24] Login SuperTokens

Nicht-authentifizierte Anfrage

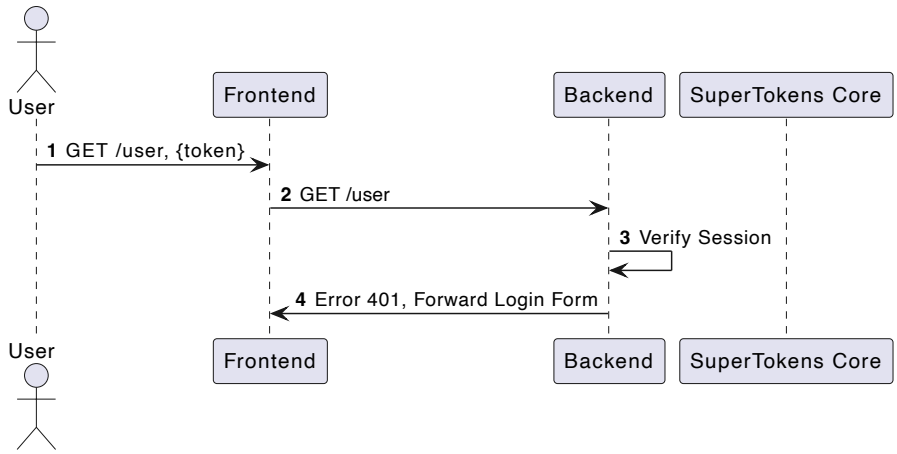

Abbildung 20. UML Sequenzdiagramm [24] Not Authenticated Request SuperTokens

Nicht-autorisierte Anfrage

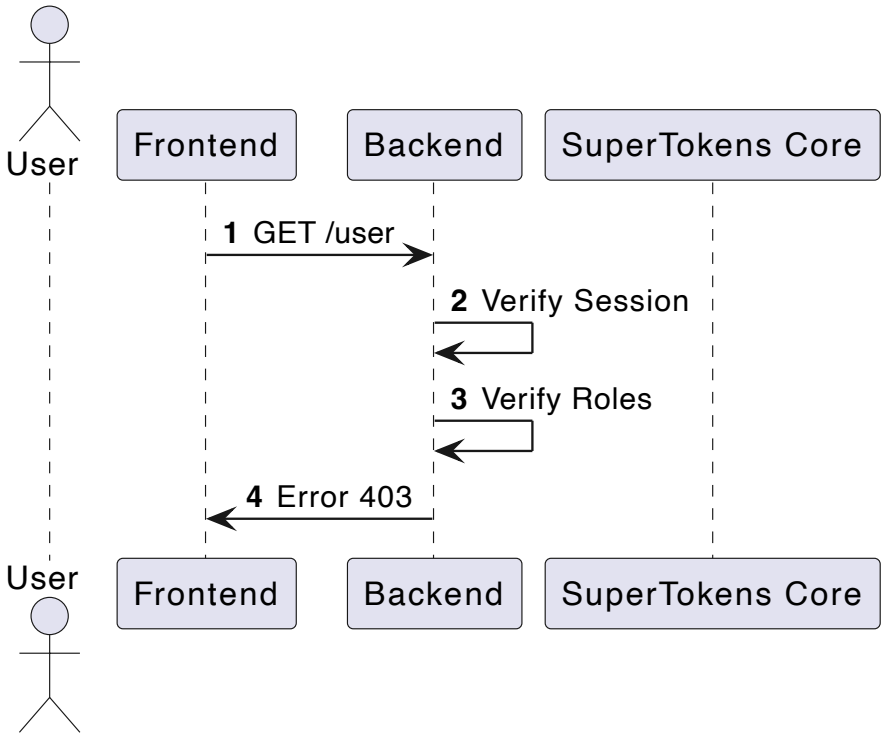

Abbildung 21. UML Sequenzdiagramm [24] Not Authorized Request SuperTokens

Valide Anfrage

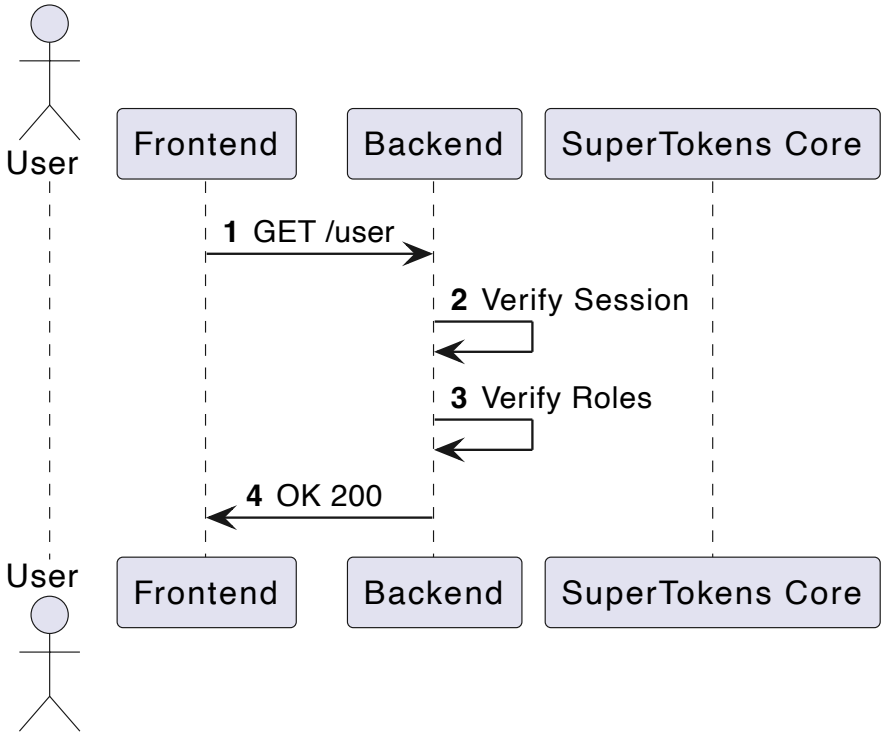

Abbildung 22. UML Sequenzdiagramm [24] Valid Request SuperTokens

Logout

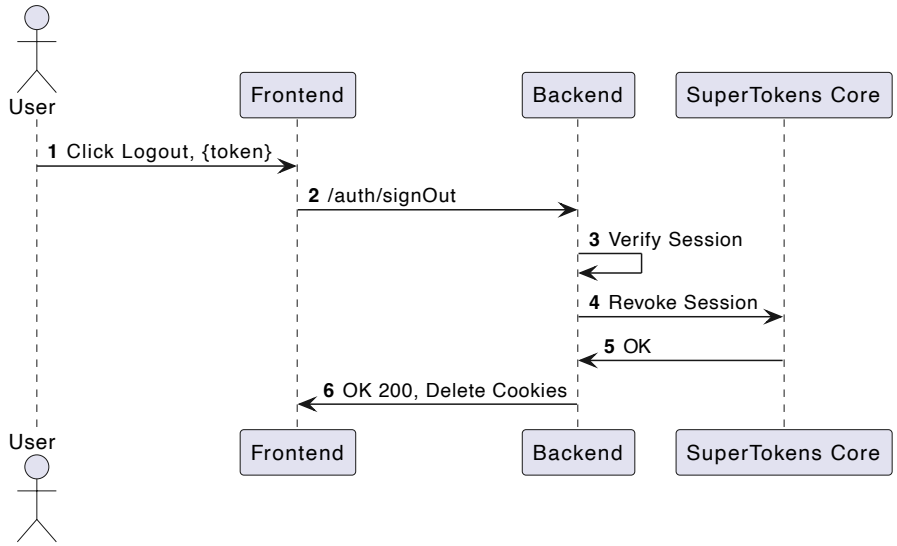

Abbildung 23. UML Sequenzdiagramm [24] Logout SuperTokens

B.5. Traefik

B.5.1. Lizenz

Traefik entwickelt Open-Source-Produkte unter MIT [55], ist aber auch Anbieter eines kostenpflichtigen Modells für Firmen.

B.5.2. Dokumentation

Die Webseite von Traefik ist übersichtlich gestaltet und das Produkt ist gut dokumentiert. Wichtige Sachverhalte werden mit Bildern anschaulich vermittelt. Mögliche Konfigurationen werden mit Code-Beispielen in verschiedenen Sprachen zur Verfügung gestellt, was Traefik zu einem sehr einsteigerfreundlichen Produkt macht.

B.5.3. Features

Folgende Features sind uns auf den ersten Blick überaus positiv aufgefallen.

- Reverse-Proxy
- Load Balancing
- TLS Integration

B.5.4. Implementation und Schwierigkeiten

Bei der Recherche nach einer geeigneten Lösung für die Authentisierung und Autorisierung haben wir die Möglichkeit in Betracht gezogen, eine Middleware von Traefik einzusetzen. Traefik wird bereits erfolgreich als Reverse-Proxy eingesetzt und sollte erweitert werden. Dadurch ist diese Erweiterung naheliegend und könnte mit vergleichbar geringem Aufwand umgesetzt werden. Der grosse Vorteil dieser Middleware ist es, dass ein Identity Provider unabhängig vom Backend konfiguriert werden kann. Wohingegen der klare Nachteil einer Abhängigkeit gegenüber Traefik herrscht.

Nach kurzer Recherche haben wir herausgefunden, dass die Middleware [56] ausschliesslich in der kostenpflichtigen Enterprise Variante verfügbar ist. Aus Gründen des Open-Source-Ansatzes dieser Bachelorarbeit ist uns deshalb diese Option verwehrt, abgesehen davon wäre die Enterprise-Variante auch sehr kostenintensiv und in unserer Situation kaum lohnend.

B.5.5. Traefik OIDC Middleware

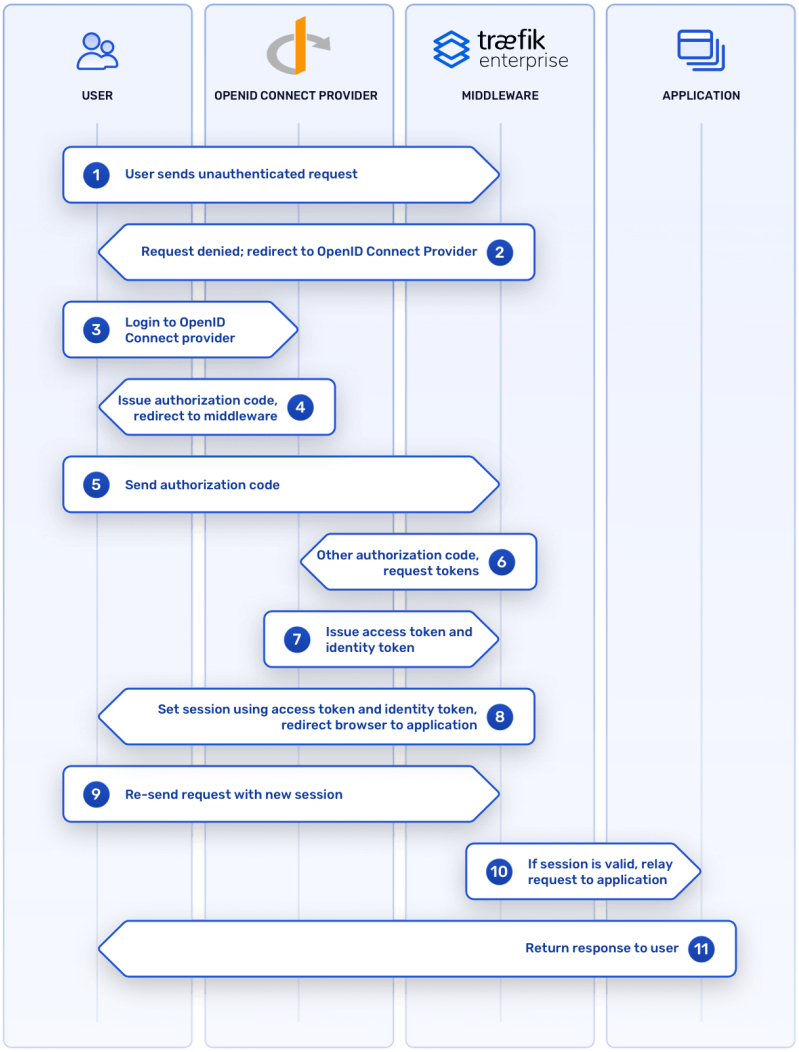

Abbildung 24. Traefik OIDC Middleware [56]

Appendix C: Testreporte

Testing ist ein wichtiges Mittel zur Qualitätssicherung einer Software. Dabei gibt es viele Faktoren, welche es zu berücksichtigen gilt. In diesem Kapitel werden die ausgeführten Tests und deren Resultate festgehalten. Die Testresultate werden anschliessend beurteilt und eingeordnet.

C.1. Backend Test Coverage

Zur Feststellung der Testabdeckung im Backend haben wir einen Job in der Build-Pipeline integriert. Dieser Job wird vor jeder Vereinigung in den Master-Branch ausgeführt, um jederzeit eine Auswertung zur Testabdeckung zu erhalten. In den Merge-Requests sind die getesteten Abschnitte farblich hinterlegt, damit ersichtlich ist, ob neuer Code getestet ist.

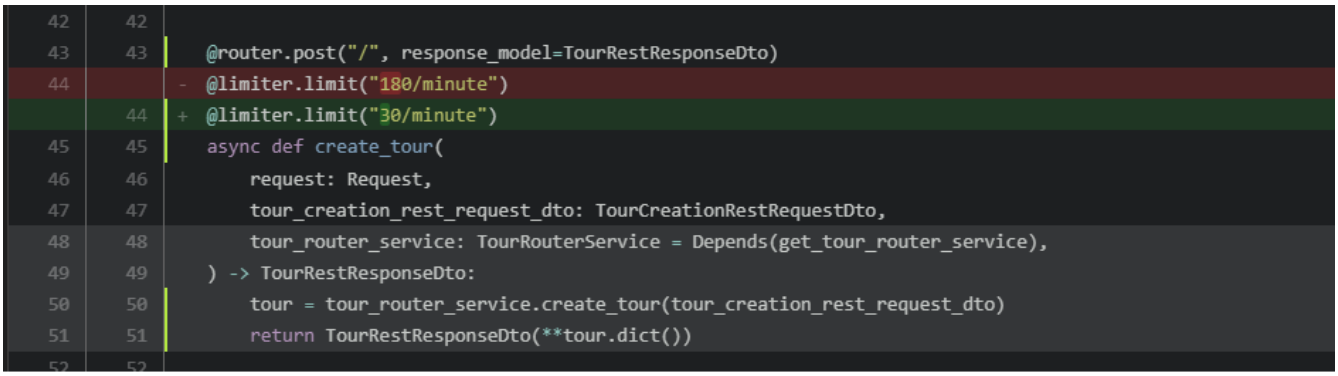

Abbildung 25. Beispiel von getestetem Code in Merge-Request von GitLab

Dazu werden sämtliche Tests ausgeführt. Die Auswertung wird anschliessend an GitLab übergeben. Die Codebasis der Backend-Applikation ist zu 88 % mit Tests abgedeckt.

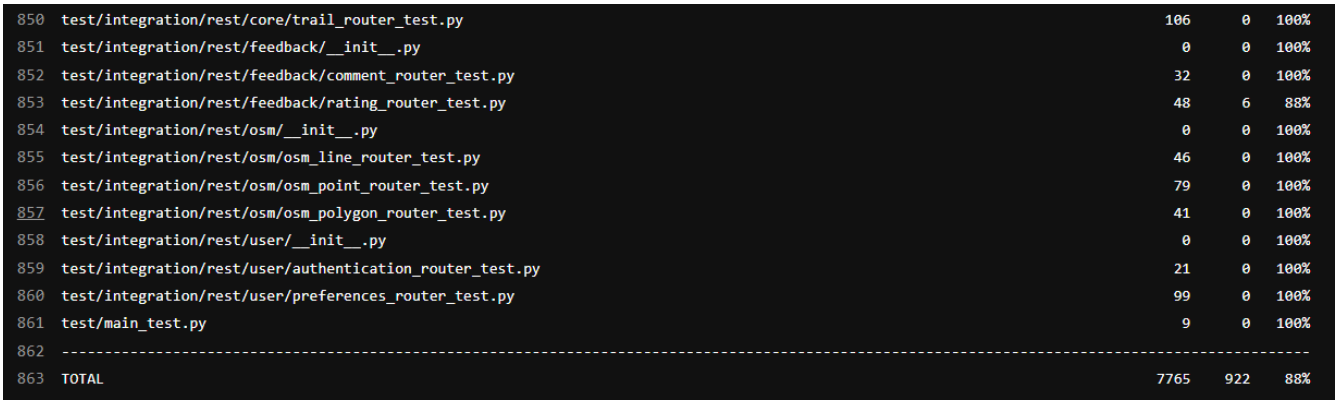

Abbildung 26. Auszug des Testreports zur Abdeckung der Codebasis mit Tests

C.2. Backend Load/Performance Test

Um die spezifizierten nicht-funktionalen Anforderungen im Bereich Performanz zu überprüfen, haben wir Tests spezifiziert, die in diesem Kapitel festgehalten werden.

C.2.1. Load-Test Backend für Rundgang

Für die Überprüfung der Anforderung TOUR-NFR-1 wurde folgender Test spezifiziert:

ID	TOUR-NFR-1-TEST-1
Anzahl Backend Instanzen	1
Anzahl Anfragen	1000
Ramp-up Periode	500s

Request Body	<pre>{
	  "category_ids": [2],
	  "start_coordinate": {
	  "latitude": "${latitude}",
	  "longitude": "${longitude}"
	  },
	  "perimeter": 500,
	  "required_tags": [],
	  "max_distance_in_meters": 0,
	  "routing_engine": "osrm-engine-tour"
	}</pre>

Latitude/Longitude	Random Samples um Zürich, Bern und Luzern

Tabelle 16. Load-Test-Spezifikation TOUR-NFR-1-TEST-1

Dieser Test wurde mithilfe von Apache JMeter ausgeführt und hat dabei folgendes Resultat ergeben.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	1000	868	133	4151	647.26	10.70%	2.0/sec	28.12	0.89	14408.5
TOTAL	1000	868	133	4151	647.26	10.70%	2.0/sec	28.12	0.89	14408.5

Abbildung 27. Testresultat TOUR-NFR-1-TEST-1 Summary

Die durchschnittliche Ausführungszeit ist mit 868ms deutlich unter der in TOUR-NFR-1 geforderten 3s. Dabei resultierten 10,70 % der Anfragen in einem Fehler. Dies ist auf den Umstand zurückzuführen, dass die Daten für Latitude und Longitude randomisiert erstellt wurden. Darunter sind auch Koordinaten, für die es nicht möglich war, einen Rundgang zu erstellen. Die kürzeste Dauer von 133ms ist auf einen dieser Fehler zurückzuführen. Die längste Ausführung dauerte 4151ms. Dieser liegt deutlich über dem Durchschnitt und ist zu Beginn des Tests ausgeführt worden. Auf dem Graph sind keine nennenswerten Anomalien zu erkennen. Die Standardabweichung beträgt 647ms und der Median liegt bei 657ms.

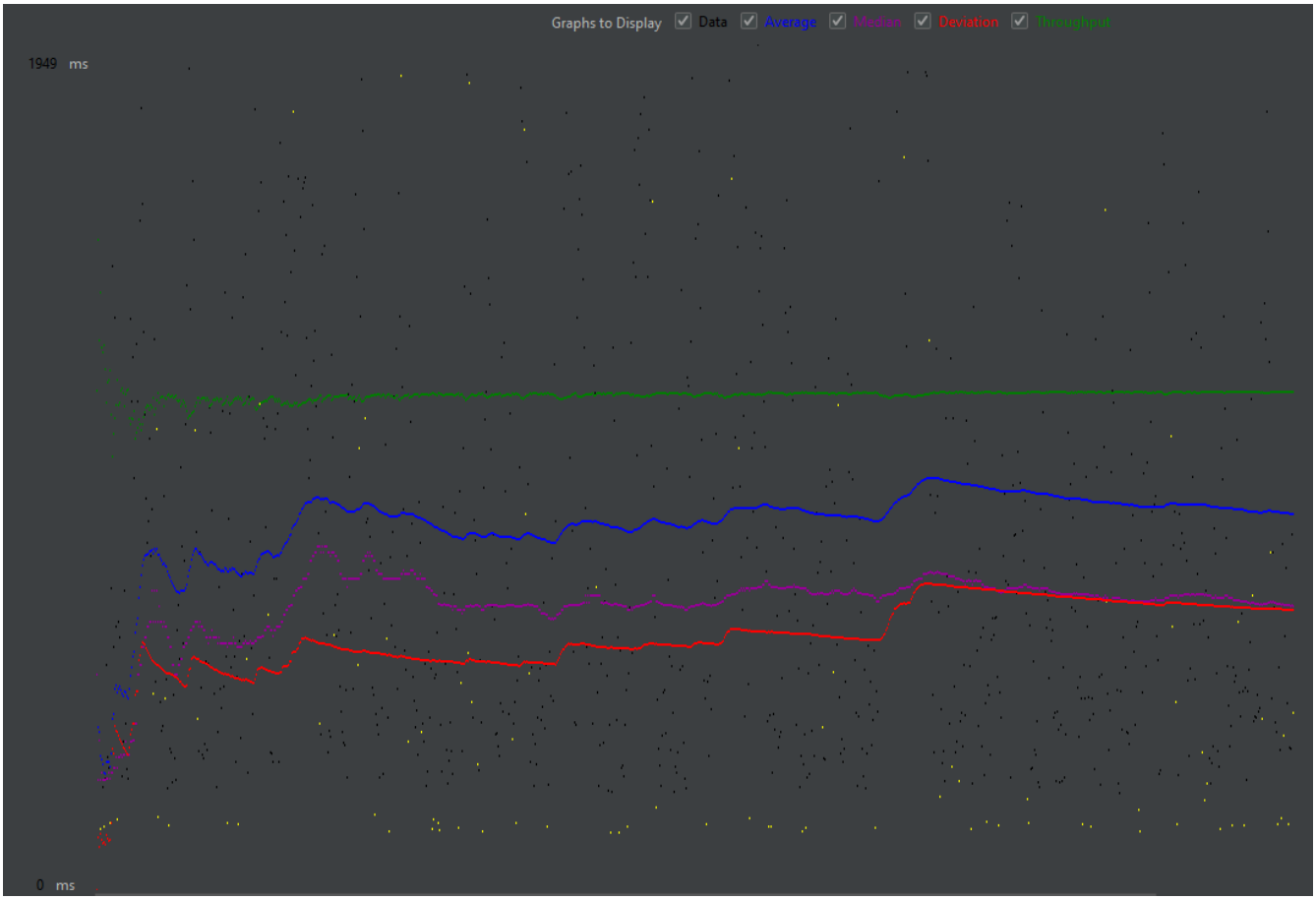

Abbildung 28. Testresultat TOUR-NFR-1-TEST-1 Graph

Die Gesamtausführungsdauer beträgt 8:20 min. Dies ergibt einen durchschnittlichen Durchsatz von rund 2 Anfragen pro Sekunde. In TOUR-NFR-3 rechneten wir mit rund 30 Anfragen pro Minute oder einer Anfrage alle 2 Sekunden. Wir beurteilen diese Anforderung als ebenfalls erfüllt. Zum Schutz der Applikation vor Lastspitzen wurde eine Limitierung von 150 Anfragen pro Minute eingebaut. Im nachfolgenden Test wird die Einhaltung von TOUR-NFR-1 mit dieser Obergrenze geprüft.

ID	TOUR-NFR-1-TEST-2
Anzahl Backend Instanzen	1
Anzahl Anfragen	500
Ramp-up Periode	200s
Request Body	`{ "category_ids": [2], "start_coordinate": { "latitude": "${latitude}", "longitude": "${longitude}" }, "perimeter": 500, "required_tags": [], "max_distance_in_meters": 0, "routing_engine": "osrm-engine-tour" }`
Latitude/Longitude	Random Samples um Zürich, Bern und Luzern

Tabelle 17. Load-Test-Spezifikation TOUR-NFR-1-TEST-2

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	6211	138	10670	3004.70	11.40%	2.4/sec	33.82	1.06	14535.3
TOTAL	500	6211	138	10670	3004.70	11.40%	2.4/sec	33.82	1.06	14535.3

Abbildung 29. Testresultat TOUR-NFR-1-TEST-2 Summary

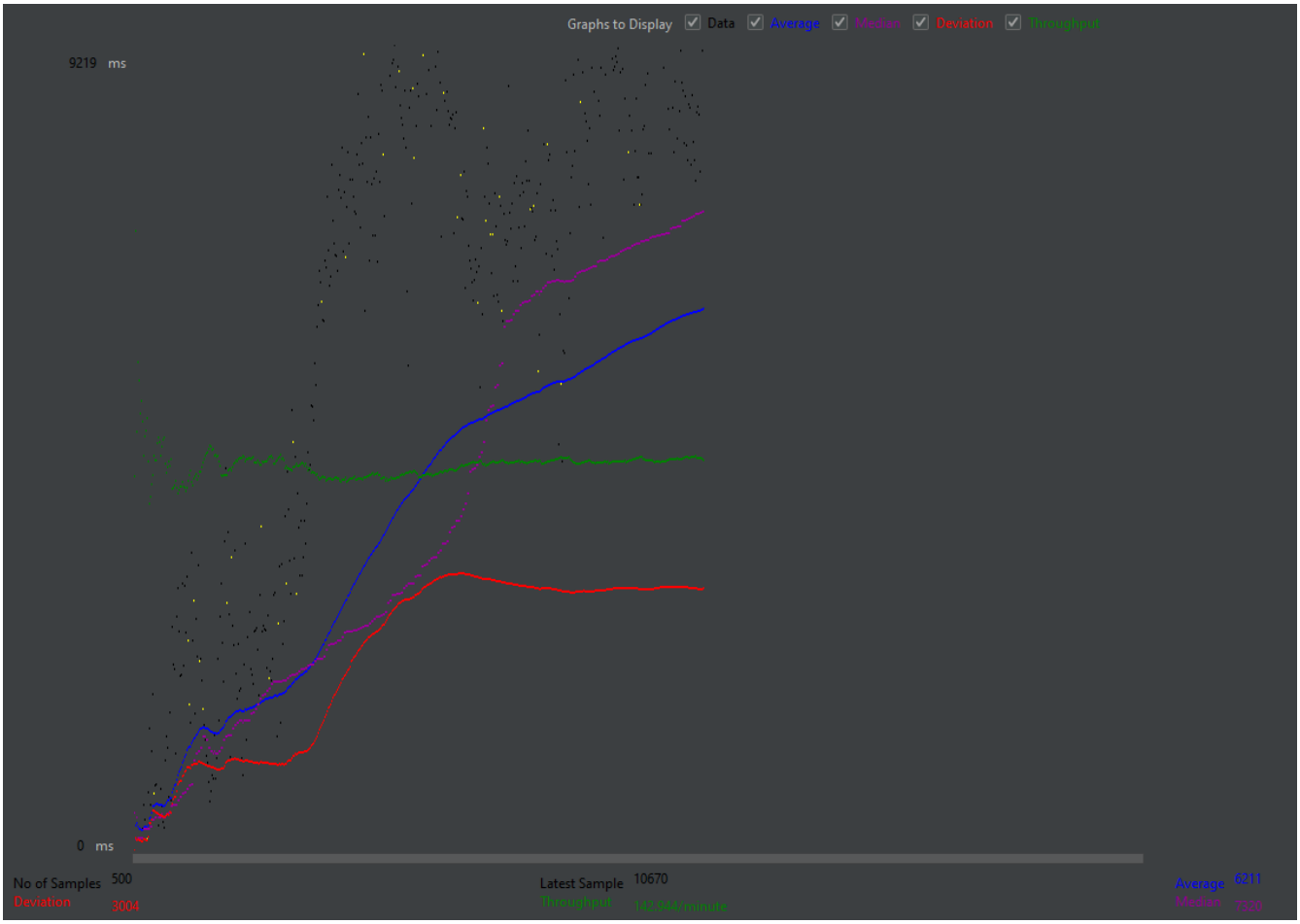

Abbildung 30. Testresultat TOUR-NFR-1-TEST-2 Graph

Bei diesem Test wird deutlich, dass die Anforderung TOUR-NFR-1 nach rund 1:30 min nicht mehr eingehalten werden kann. Die durchschnittliche Dauer über den gesamten Test liegt mit 6211ms über den spezifizierten 3s. Aus diesem Grund skalieren wir die Backend-Applikation und erhöhen die Anzahl der Instanzen um 1.

ID	TOUR-NFR-1-TEST-3
Anzahl Backend Instanzen	2
Anzahl Anfragen	500
Ramp-up Periode	200s

Request Body	{
	  "category_ids": [2],
	  "start_coordinate": {
	  "latitude": "${latitude}",
	  "longitude": "${longitude}"
	  },
	  "perimeter": 500,
	  "required_tags": [],
	  "max_distance_in_meters": 0,
	  "routing_engine": "osrm-engine-tour"
	}

Latitude/Longitude	Random Samples um Zürich, Bern und Luzern

Tabelle 18. Load-Test-Spezifikation TOUR-NFR-1-TEST-3

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	534	133	1716	317.64	11.40%	2.5/sec	35.33	1.10	14535.3
TOTAL	500	534	133	1716	317.64	11.40%	2.5/sec	35.33	1.10	14535.3

Abbildung 31. Testresultat TOUR-NFR-1-TEST-3 Summary

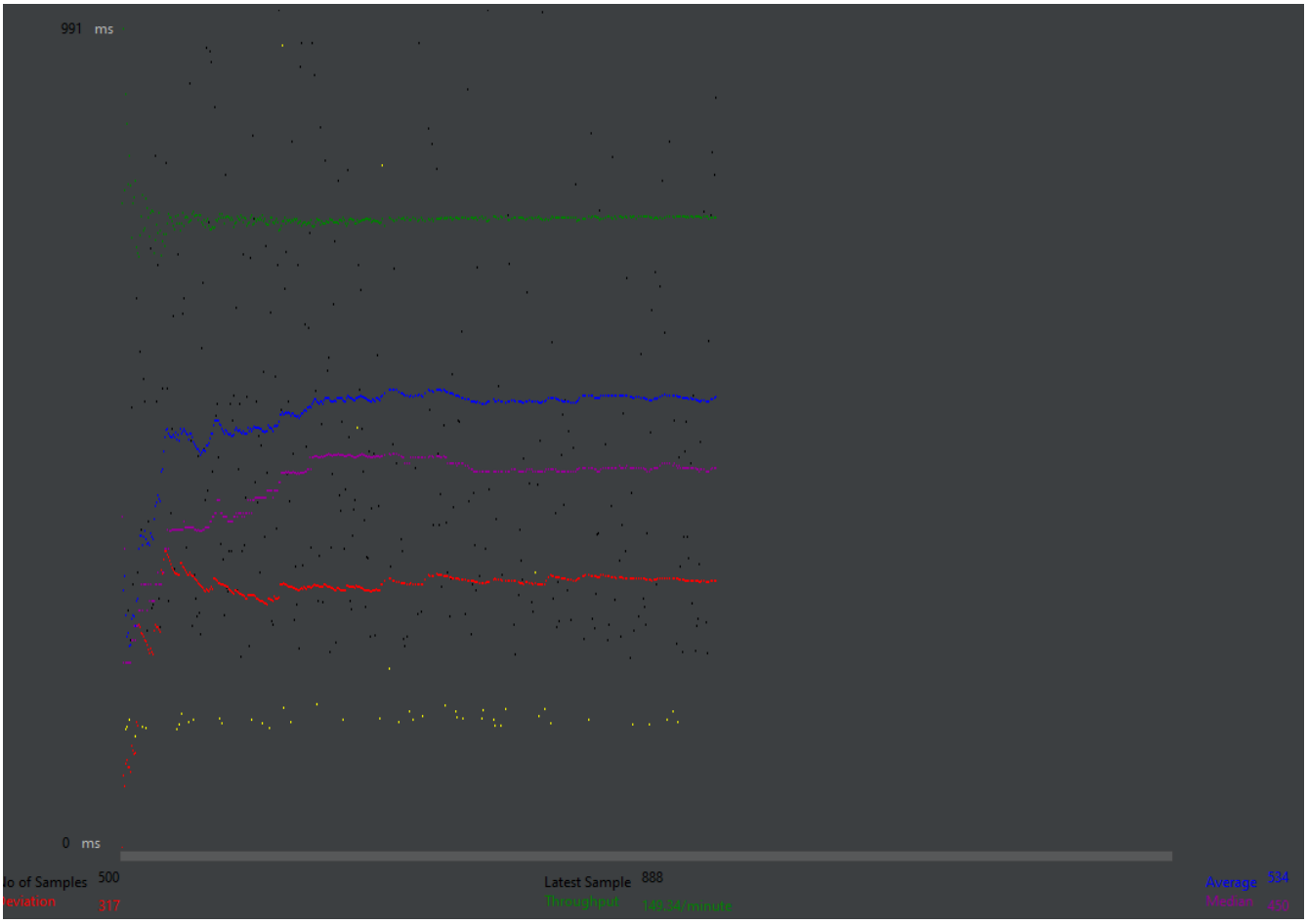

Abbildung 32. Testresultat TOUR-NFR-1-TEST-3 Graph

Die Resultate dieses Tests liefern den Beweis, dass die horizontale Skalierung der Backend-Applikation funktioniert. Die Anfragen werden alle innerhalb der spezifizierten Dauer abgearbeitet, wobei die durchschnittliche Dauer mit 534ms deutlich unter 3s liegt. Der Median liegt bei 450ms und die Standardabweichung bei 317ms.

Diese Skalierung kann theoretisch beliebig oft angewandt werden. Allerdings wird man an den Punkt kommen, wo der Server an seine physischen Grenzen kommen wird oder eine andere Komponente der Flaschenhals im System ist. An diesem Punkt müssen wirtschaftliche und technische Faktoren nochmals beurteilt werden, um eine geeignete Lösung zu finden.

ID	TOUR-NFR-1-TEST-4
Anzahl Backend Instanzen	2
Anzahl Anfragen	2000
Ramp-up Periode	800s

Request Body	<pre>{
	  "category_ids": [2],
	  "start_coordinate": {
	  "latitude": "${latitude}",
	  "longitude": "${longitude}"
	  },
	  "perimeter": 500,
	  "required_tags": [],
	  "max_distance_in_meters": 0,
	  "routing_engine": "osrm-engine-tour"
	}</pre>

Latitude/Longitude	Random Samples um Zürich, Bern und Luzern

Tabelle 19. Load-Test-Spezifikation TOUR-NFR-1-TEST-4

Anhand dieses letzten Tests für TOUR-NFR-1 wollen wir feststellen, ob die Anforderung auch über längere Dauer eingehalten werden kann. Dazu wurde ein neues Datenset erstellt und der Test über eine längere Dauer gestartet.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	2000	546	131	2200	341.09	11.55%	2.5/sec	35.89	1.11	14708.3
TOTAL	2000	546	131	2200	341.09	11.55%	2.5/sec	35.89	1.11	14708.3

Abbildung 33. Testresultat TOUR-NFR-1-TEST-4 Summary

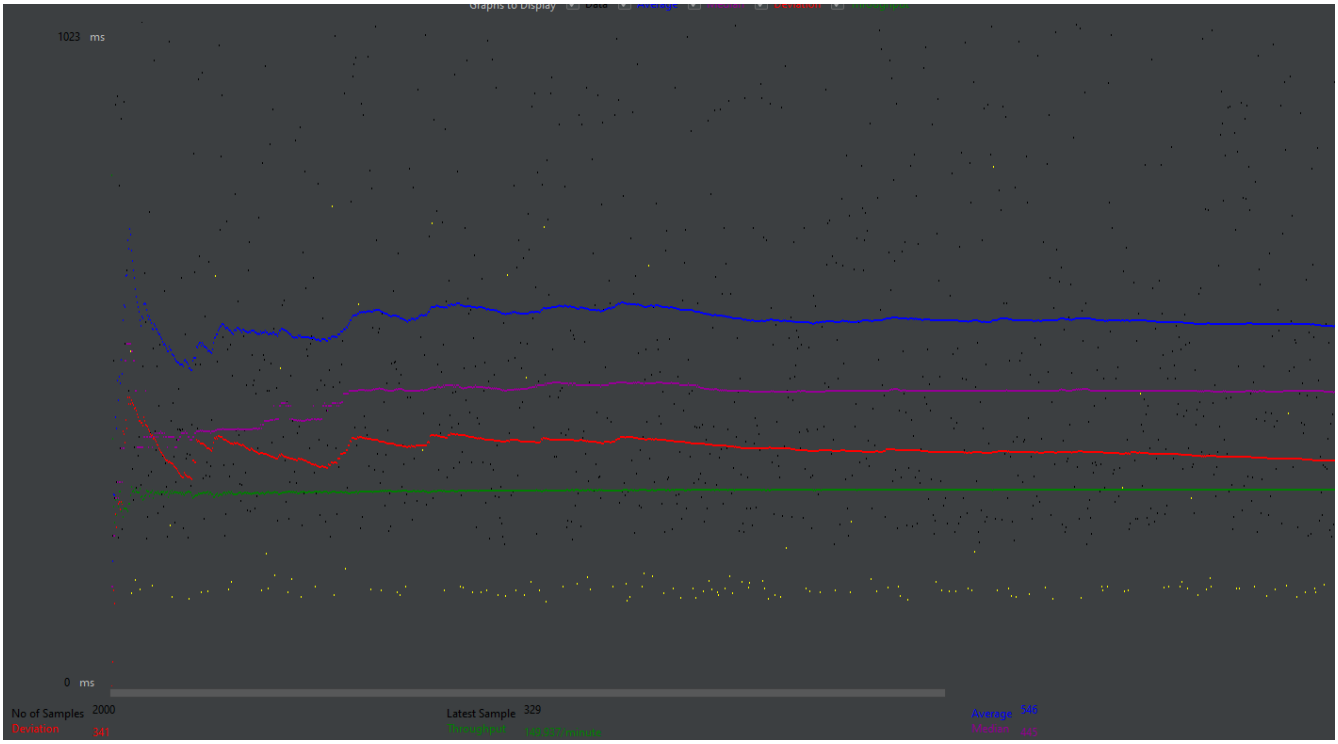

Abbildung 34. Testresultat TOUR-NFR-1-TEST-4 Graph

Dem Resultat ist zu entnehmen, dass das System über eine Dauer von 13:20 min mit der Last umgehen kann.

C.2.2. Load-Test Backend für Spaziergang

Zur Überprüfung von TRAIL-NFR-1 werden ebenfalls Load-Tests vorgenommen.

ID	TRAIL-NFR-1-TEST-1
Anzahl Backend Instanzen	2
Anzahl Anfragen	1000
Ramp-up Periode	500s

Request Body	{
	"category_ids": [2],
	"start_coordinate": {
	"latitude": "${latitude-start}",
	"longitude": "${longitude-start}"
	},
	"end_coordinate": {
	"latitude": "${latitude-end}",
	"longitude": "${longitude-end}"
	},
	"required_tags": [],
	"max_duration_in_seconds": 0,
	"routing_engine": "osrm-engine-trail"
	}

Latitude/Longitude	Random Samples um Zürich, Bern und Luzern

Tabelle 20. Load-Test-Spezifikation TRAIL-NFR-1-TEST-1

Bei diesem Test werden 2 Anfragen pro Sekunde an die Backend-Applikation geschickt. Mit diesem initialen Test soll festgestellt werden, ob die Werte überhaupt in den angestrebten Bereich fallen.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	1000	640	137	3591	490.34	15.80%	2.0/sec	29.53	1.06	15142.7
TOTAL	1000	640	137	3591	490.34	15.80%	2.0/sec	29.53	1.06	15142.7

Abbildung 35. Testresultat TRAIL-NFR-1-TEST-1 Summary

Das arithmetische Mittel der Ausführungsdauer liegt für Spaziergängen bei 640ms. Der Median des Tests liegt bei 481ms und die Standardabweichung bei 490ms. Die Anforderung TRAIL-NFR-1 ist damit deutlich erfüllt.

Abbildung 36. Testresultat TRAIL-NFR-1-TEST-1 Graph

Beim Betrachten des resultierenden Graphen ist ersichtlich, dass die Applikation keine Probleme mit der getesteten Last hat. Es sind keine Anomalien erkennbar, welche weitere Schlüsse zulassen würden. Aus diesem Grund reduzieren wir die Ramp-up Periode und führen den Test nochmals aus.

ID	TRAIL-NFR-1-TEST-2
Anzahl Backend Instanzen	2
Anzahl Anfragen	1000
Ramp-up Periode	400s

Request Body	{
	  "category_ids": [2],
	  "start_coordinate": {
	  "latitude": "${latitude-start}",
	  "longitude": "${longitude-start}"
	  },
	  "end_coordinate": {
	  "latitude": "${latitude-end}",
	  "longitude": "${longitude-end}"
	  },
	  "required_tags": [],
	  "max_duration_in_seconds": 0,
	  "routing_engine": "osrm-engine-trail"
	}

Latitude/Longitude	Random Samples um Zürich, Bern und Luzern

Tabelle 21. Load-Test-Spezifikation TRAIL-NFR-1-TEST-2

Die Reduktion der Ramp-up Periode hat keinen nennenswerten Einfluss auf das Verhalten der Applikation. Dies wird durch die Resultate bestätigt.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	1000	718	139	3061	540.57	15.80%	2.5/sec	36.85	1.32	15142.7
TOTAL	1000	718	139	3061	540.57	15.80%	2.5/sec	36.85	1.32	15142.7

Abbildung 37. Testresultat TRAIL-NFR-1-TEST-2 Summary

Bei diesem erneuten Test liegt das arithmetische Mittel bei 718ms. Der Median liegt mit 533ms etwas über dem vorherigen Resultat und die Standardabweichung mit 540ms ebenfalls. Der erhöhte Durchsatz dieses Testfalls bestätigt ebenfalls TRAIL-NFR-3. Es können weitaus mehr Anfragen als 30 pro Minute beantwortet werden.

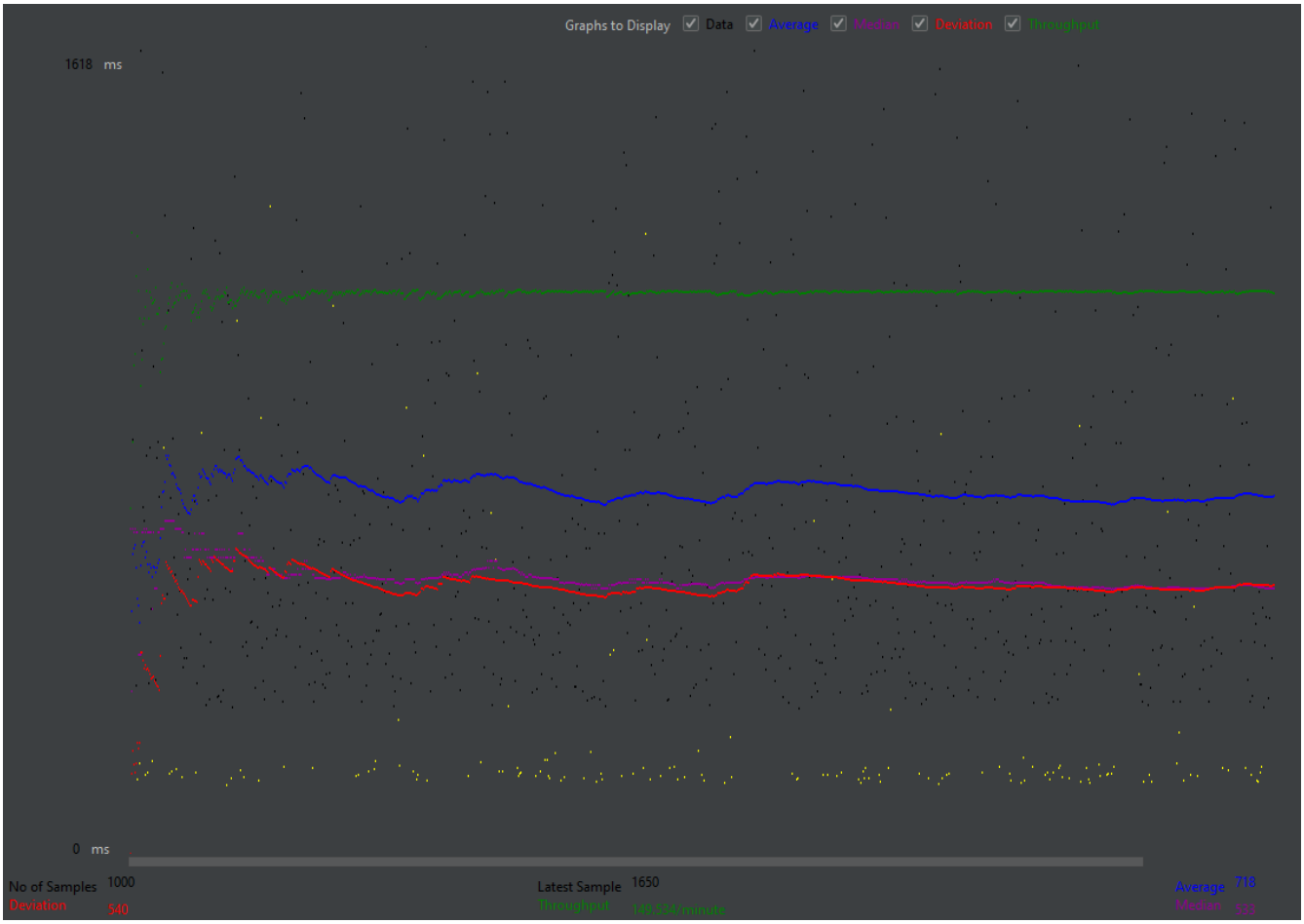

Abbildung 38. Testresultat TRAIL-NFR-1-TEST-2 Graph

Bei der Beurteilung der Graphen sind keine neuen Erkenntnisse anhand von Anomalien erkennbar. Damit unterscheidet sich das Laufzeitverhalten der Berechnung von Rundgang und Spaziergang unter Last nicht wesentlich voneinander.

C.2.3. Beurteilung und Einordnung

Die ausgeführten Tests geben einen Eindruck, mit welchen Zeiten für die Berechnung von Rundgängen und Spaziergängen gerechnet werden muss. Die generierten Daten für die Tests waren alle sehr ähnlich und im selben Grössenbereich. Ausserdem wurden, wo überall möglich, die Standardwerte übernommen. Grundsätzlich lässt sich sagen, dass die Anforderungen erfüllt sind und die Applikation bereit für den Einsatz ist. Natürlich lassen sich Szenarien konstruieren, in denen es unmöglich ist, die Anforderungen zu erfüllen. So wird die Suche nach einer Route mit Einschränkungen der Benutzer:innen immer länger dauern, als diejenige ohne. Dies liegt aber in der Natur des Problems, wie zu Beginn dieser Arbeit im Kapitel Stand der Technik erläutert wurde.

Zusätzlich haben andere Faktoren wie Netzwerkverbindung und daraus resultierende Latenzen ebenfalls einen Einfluss auf die Testergebnisse. Für die vorliegenden Tests wurden zudem synthetische Testdaten verwendet, welche sehr homogen sind. Aufgrund der Tatsache, dass dieses Produkt noch nicht von Benutzer:innen angewendet wird, gibt es keine echten Daten, welche für Testzwecke wiederverwendet werden könnten. Diese selbst zu erstellen ist sehr aufwändig und zeitintensiv. Jedoch lässt sich die Performanz der Applikation als akzeptabel und tauglich beurteilen. Es hat sich gezeigt, dass die horizontale Skalierung der Backend-Applikation ebenfalls den gewünschten Effekt erzielt. Aufgrund dessen halten wir an Architektur, Design und Deployment fest.

C.3. Frontend Lighthouse Test

Die Frontend-Applikation kann nicht anhand derselben Kriterien getestet werden wie die Backend-Applikation. Die Bedingungen auf jedem Gerät sind unterschiedlich und auch die Anforderungen unterscheiden sich deutlich. Wir haben uns daher für Testing mit Google Lighthouse entschieden.

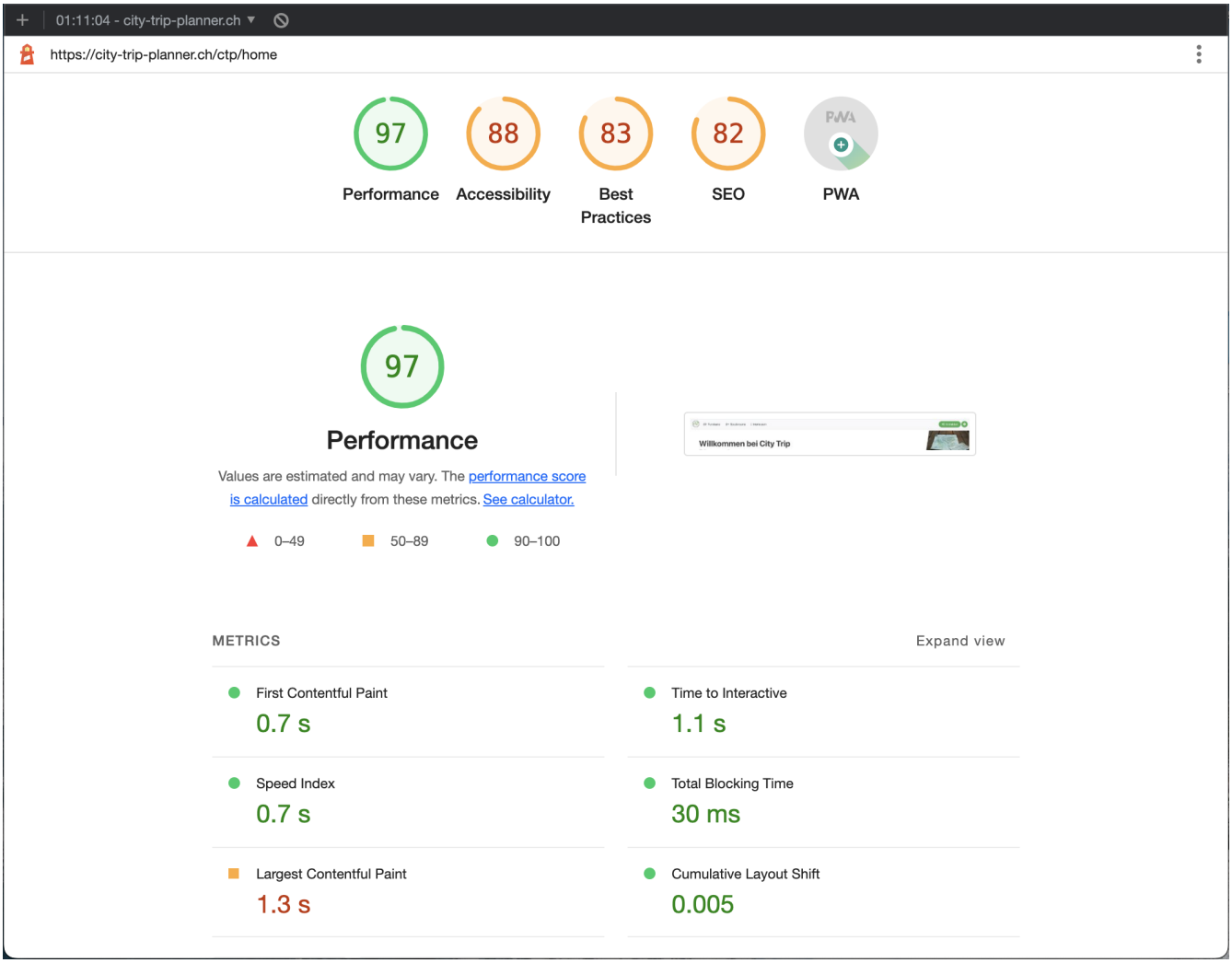

Abbildung 39. Testresultat der Frontend-Applikation mit Google Lighthouse

Aus dem Resultat sticht die Performanz der Applikation besonders hervor. Dies ist ein erfreuliches Resultat, da wir in dieser Arbeit immer wieder viel Wert auf die Performanz gelegt haben. Die anderen Bereiche haben ebenfalls gute Werte. Abzüge gibt es durch die Wahl der Farbkombinationen, welche durch das Framework jedoch vorgegeben sind, und dass die Links nicht maschinell abgesucht werden können.

Der Test bestätigt ausserdem folgende Anforderungen:

- FE-REQ-10
- DEPL-REQ-3
- DEPL-NFR-1

Appendix D: Persönliche Berichte

D.1. Jan Ruch

Nach der erfolgreichen Abgabe der vorangegangenen Studienarbeit war klar, dass wir unsere Bachelorarbeit im selben Themenbereich schreiben würden. Bereits zu Beginn war mir bewusst, dass wir auch in dieser Arbeit mit einigen neuen Technologien konfrontiert werden würden. Obwohl wir in der Studienarbeit das Wissen über die Programmiersprachen erarbeitet hatten und mir die Konzepte und Methodiken vertraut waren, schwang ein Gefühl von Respekt gegenüber dieser Arbeit mit. Um jedoch eines vorwegzunehmen, wenn ich mich nochmals vor die Wahl der Themen für die Abschlussarbeit stellen müsste, würde ich mich erneut für diese Aufgabe entscheiden.

Mich persönlich hat die Architektur der Applikation sehr beeindruckt. Der ursprüngliche Ansatz aus der Studienarbeit hat sich in dieser Bachelorarbeit erst richtig bestätigt. Durch die nahtlose Anbindung zusätzlicher Routing Engines ist mir bewusst geworden, dass eine saubere Architektur über Erfolg und Misserfolg eines Projektes entscheiden kann und wie wichtig die korrekte Ausarbeitung einer geeigneten Architektur ist. Ein Projekt kann mit einer schlechten Architektur entwickelt und möglicherweise vollendet werden. Wird die Applikation allerdings weiterentwickelt oder abgeändert, kann dies zu grossem zeitlichem Aufwand und meist schwerwiegenden strukturellen Umbauten führen. Zu meiner Zufriedenheit konnten wir die geplanten Änderungen mit beschränktem Aufwand umsetzen.

Neben den technischen Herausforderungen hat mich der Kontakt mit unserem Partner Estefan Justo von Schweiz Tourismus motiviert. Die Rückmeldungen aus einer unternehmerischen Perspektive haben mein Verständnis für kundenorientierte Verbesserungen gefördert und meine technische Sicht etwas durchbrochen. Ich bin dankbar, dass diese Zusammenarbeit stattgefunden hat und wir das Produkt massgeblich verbessern konnten.

Die vielen Arbeitsstunden und anstrengenden Wochen haben sich bezahlt gemacht. Einige der Technologien, beispielsweise einen Identity Provider, habe ich das erste Mal kennengelernt und angewendet. Dies führte zu teils steilen Lernkurven und ungeahnten Herausforderungen. An anderen Orten konnte ich wiederum bereits bekanntes Wissen auffrischen und vertiefen. Insbesondere die methodischen Kompetenzen werde ich für mein zukünftiges Berufsleben mitnehmen. Wie ich zu Beginn erwähnt habe, würde ich mich erneut für diese Arbeit entscheiden. Es hat mir grossen Spass gemacht und ich durfte einige wertvolle Erfahrungen sammeln. Zum Schluss möchte ich mich nochmals bei allen involvierten Personen für die gute Zusammenarbeit und bei meiner Familie für die Unterstützung bedanken. Ganz besonderer Dank gilt Lukas für die vielen intensiven Stunden und den unglaublich guten Austausch den wir hatten!

D.2. Lukas Grigis

In der Studienarbeit konnten wir uns eine gute Ausgangslage für diese Bachelorarbeit schaffen. Trotz einiger Hürden erarbeiten Jan und ich eine Grundlage, auf der wir weiter aufgebaut haben. Ich durfte in das mir bislang unbekannte Gebiet der Geoinformationssysteme eintauchen und konnte am praktischen Beispiel die Wichtigkeit von Architektur und Design erfahren.

Es freut mich, dass das geplante Vorhaben in dieser Bachelorarbeit umgesetzt werden konnte. Ich möchte dabei das Deployment der Applikation besonders hervorheben. Für mich persönlich ist ein Entwicklungsschritt erst abgeschlossen, wenn die Applikation lauffähig auf einem System installiert und den Benutzer:innen zugänglich ist. Dabei zeigt sich, ob die getroffenen Massnahmen korrekt waren und ob ein funktionierendes System gebaut wurde. Durch die Anwendung der bekannten DevOps-Prinzipien ist uns dieser erste entscheidende Schritt gelungen. Mit dem Betrieb einer Applikation entstehen automatisch Wartungsarbeiten. Ausserdem erwarten Benutzer:innen laufend neue Features, welche ebenfalls integriert werden sollen. Die DevOps-Prinzipien lieferten uns auch hierzu passende Lösungsansätze, welche wir realisieren konnten. Ich bin stolz, ein System erschaffen zu haben, welches diesen Herausforderungen gewachsen ist.

FastAPI ist ein relativ neues und populäres Framework, mithilfe dessen eine Applikation mit geringem Aufwand erstellt werden kann. Aus meiner Sicht ist FastAPI noch nicht ausgereift genug, um damit eigenständige Backend-Applikationen umzusetzen. Besonders im Bereich Security und Testing gibt es noch Verbesserungsmöglichkeiten. So wäre es wünschenswert, dass die Interaktion mit einer Datenbank trotz Dependency Injection einfacher testbar wäre. Im Bereich Security müssen eigene Implementationen gemacht werden, was sehr fehleranfällig ist. Eine Standardisierung wäre auch hier wünschenswert. Aus diesem Grund würde ich FastAPI eher in einer Microservice-Architektur hinter einem API-Gateway einsetzen, sodass die Überprüfung der Autorisierung ausgelagert werden kann. Für die Weiterentwicklung von City Trip Planner würde ich in Betracht ziehen, das Framework auszutauschen. Dies hätte einen Einfluss auf den Infrastruktur-Layer der Applikation, jedoch nicht auf die Business-Logik.

Mit dem Einreichen dieser Bachelorarbeit endet mein Bachelorstudium, welches ich im September 2018 begonnen habe. Während den vergangenen Jahre konnte ich mir viel neues Wissen im Bereich Software-Engineering aneignen und spannende Erfahrungen und Erkenntnisse in unterschiedlichen Bereichen sammeln. Dabei ist meine Begeisterung für Software stets gestiegen und wurde gar zum Enthusiasmus. Mit diesem Enthusiasmus im Rucksack - wie es im OST-Jargon heisst - freue ich mich auf viele neue Herausforderungen, bei dessen Bewältigung ich das Erlernte anwenden kann.

Zu guter Letzt möchte ich mich bei Jan bedanken. In diesem Studium hatten wir unzählige Diskussionen, regen Austausch und viele wertvolle Gespräche. Jan, du weisst wofür und du weisst wieso: Danke!

Appendix E: Glossar und Abkürzungsverzeichnis

Apache JMeter

Open-Source-Java-Applikation für Load-Tests und Performanzmessungen. [57]

City Trip Planner

City Trip Planner ist der Name des Produkts, welches im Rahmen der vorliegenden Arbeit entwickelt wurde.

Data Transfer Object

Ein Data Transfer Object steht ist ein simples Objekt ohne Logik, das für die reine Datenübertragung eingesetzt wird.

Git

Git [58] ist ein Tool zur verteilten Versionsverwaltung von Software-Projekten.

GitLab

GitLab [59] ist eine Webanwendung zur Verwaltung von Projekten, welche auf Git basieren.

Google Lighthouse

Lighthouse [60] ist ein Open-Source-Tool um die Qualität von Webseiten zu überprüfen und zu verbessern.

GraphHopper

GraphHopper [43] ist eine Open-Source Routing Engine.

Identity Provider

Ein Identity Provider ist ein Anbieter für einen Dienst zur Authentifizierung und Autorisierung von Benutzern. Einer der wichtigsten Anwendungsfälle ist der sogenannte Single Sign-On. Dafür werden verschiedene Protokolle unterstützt, beispielsweise SAML, OAuth 2.0 und OpenID Connect.

JSON Web Token

Ein JSON Web Token ist JSON-basiert und ermöglicht den Austausch von sicherheitsrelevanten Benutzerinformationen. Das Token wird verschlüsselt und kann nicht unbemerkt verändert werden, um sich beispielsweise selbst Administrator-Rechte einzuräumen.

Keycloak

Keycloak [45] ist ein Open-Source-Projekt eines Identity Provider.

Minimal Viable Product

Ein Minimal Viable Product beschreibt die erste funktionsfähige Iteration eines Softwareprodukts mit dem Ziel, möglichst rasch Feedback generieren zu können.

NgRx

NgRx [40] ist ein Framework, welches Software-Bibliotheken für den Bau von reaktiven Angular Applikationen zur Verfügung stellt.

OAuth 2.0

Open Authorization 2.0 [61] ist ein Standardprotokoll für Authentifizierung und Autorisierung von Benutzern.

OpenID Connect

OpenID Connect [62] basiert auf OAuth 2.0 und fügt eine zusätzliche Schicht mit Informationen über den Benutzer hinzu.

openrouteservice

openrouteservice [44] ist eine Open-Source Routing Engine.

OpenStreetMap

OpenStreetMap [63] ist ein Projekt, welches frei nutzbare Geodaten zur Verfügung stellt.

osm2pgsql

osm2pgsql [64] ist ein Commandline-Tool für den Import von OpenStreetMap-Daten in PostgreSQL.

Personally Identifiable Information

Unter *PII* versteht man alle Informationen, welche Rückschlüsse auf eine Person erlauben, womit diese identifiziert werden kann. Solche Personen-identifizierende Daten unterliegen besonderem, gesetzlich verankertem Schutz.

pgRouting Project

Das Projekt pgRouting [65] stellt Routing-Funktionen basierend auf PostGIS und PostgreSQL zur Verfügung.

Point of Interest

Ein Point of Interest (de: Ort von Interesse) stellt einen Punkt auf der Landkarte dar, welcher für den Betrachter von Bedeutung sein könnte.

PostGIS

PostGIS [28] ist eine Extension für PostgreSQL-Datenbanken, welche es ermöglicht geografische Objekte in einer objektrelationalen Datenbank abzuspeichern und abzufragen.

PostgreSQL

PostgreSQL [27] ist eine offene objektrelationale SQL Datenbank.

PrimeNG

PrimeNG [37] ist eine Software-Bibliothek für UI-Komponenten.

Project Open Source Routing Machine

Das Projekt Open Source Routing Machine [42] stellt eine freie und offene Routing Engine zur Verfügung.

Proof of Concept

Ein Proof of Concept ist ein Machbarkeitsnachweis, der die Durchführbarkeit belegt.

Rational Unified Process

RUP [66] ist ein kommerzielles Produkt, welche ein Vorgehensmodell definiert und die dazugehörigen Entwicklungsprogramme liefert.

Rundgang

Der Begriff *Rundgang* steht für einen Ausflug ausgehend von einem Startpunkt über diverse Stationen und endet wieder am Ausgangspunkt.

SAML

Security Assertion Markup Language [67] ist ein XML-basiertes Protokoll, dass für die Authentifizierung von Benutzer:innen verwendet wird.

SCRUM

Scrum [68] ist ein Vorgehensmodell zur agilen Softwareentwicklung.

Single Page Application

Eine Single Page Application ist eine Webanwendung, die mit nur einem einzigen HTML-Dokument auskommt und Inhalte dynamisch nachlädt.

Single Sign-On

Single Sign-On bedeutet, dass Benutzer:innen nach einer einmaligen Authentifizierung beim Identity Provider, nun auf alle Systeme Zugriff haben, die denselben Identity Provider nutzen.

Spaziergang

Als *Spaziergang* wird in dieser Arbeit ein Ausflug bezeichnet, welcher an einem Startpunkt A startet und an einem definierten Endpunkt B endet. Zwischen Startpunkt und Endpunkt liegen diverse Stationen.

Station

Als *Station* wird ein Punkt auf der Route eines Rundgangs oder Spaziergangs bezeichnet. Eine Station unterscheidet sich von einem Point of Interest dadurch, dass Stationen aus einem Point of Interest bestehen, ein Point of Interest aber nicht automatisch eine Station ist.

Studienarbeit

Die Studienarbeit [1] wurde im Herbstsemester 2021 von Lukas Grigis und Jan Ruch geschrieben und dient als Grundlage für die vorliegende Bachelorarbeit.

Traefik

Traefik [46] ist ein Open-Source Reverse-Proxy.

Universally Unique Identifier

Ein UUID steht für einen Wert, welche zur Identifikation eines Objekts verwendet wird.

Vertical Slice

Ein Vertical Slice repräsentiert einen vertikalen Schnitt durch sämtliche Schichten der Applikation. Die Breite eines Vertical Slices kann aus einer einzelnen oder mehreren Entitäten bestehen.

Appendix F: Literatur- und Quellenverzeichnis

Quellen

[1] Leuenberger, L. & Ruch, J. (2022, März). City Trip Planner: Kurztrip-Planer für Fussgänger. ePrints OST. Abgerufen am 4. Januar 2023, von https://eprints.ost.ch/id/eprint/984/

[2] Flaticon. (o. D.). Kostenlose Icons und Sticker – Millionen von Ressourcen zum Herunterladen. Abgerufen am 4. Januar 2023, von https://www.flaticon.com/de/

[3] Facts & Figures. (2023, 3. Januar). Schweiz Tourismus. Abgerufen am 9. Januar 2023, von https://www.stnet.ch/de/portrait/facts-figures/

[4] Tourismus, S. (o. D.). Schweizer Städte-Tipps. Schweiz Tourismus. Abgerufen am 9. Januar 2023, von https://www.myswitzerland.com/de-ch/erlebnisse/staedte-kultur/staedtereisen/schweizer-staedte-tipps/

[5] Wikipedia-Autoren. (2002, 22. November). Hamiltonkreisproblem. Wikipedia. Abgerufen am 9. Januar 2023, von https://de.wikipedia.org/wiki/Hamiltonkreisproblem

[6] Wikipedia-Autoren. (2002, 20. Dezember). Problem des Handlungsreisenden. Wikipedia. Abgerufen am 9. Januar 2023, von https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden

[7] Wikipedia-Autoren. (2002, 22. November). Eulerkreisproblem. Wikipedia. Abgerufen am 9. Januar 2023, von https://de.wikipedia.org/wiki/Eulerkreisproblem

[8] Cook, W. (o. D.). Concorde TSP Solver. math.uwaterloo.ca. Abgerufen am 4. Januar 2023, von https://www.math.uwaterloo.ca/tsp/concorde.html

[9] Using Self-Organizing Maps to solve the Traveling Salesman Problem. (2018, 21. Januar). Diego Vicente. Abgerufen am 4. Januar 2023, von https://diego.codes/post/som-tsp/

[10] Purru, A. (2022, 25. März). The Hierholzer's Algorithm Explained. Abhijit Purru. Abgerufen am 4. Januar 2023, von https://www.abhijitpurru.com/blog/hierholzer-algorithm

[11] Scala, F. (2014). POI Tour – Personalisierter Tourenplaner für Fussgänger. ePrints OST. Abgerufen am 4. Januar 2023, von https://eprints.ost.ch/id/eprint/408/

[12] Scala, F. (2014, Dezember). GitHub - fabio-scala/poitour: POI Tour - Personalisierter Tourenplaner für Fussgänger. GitHub. Abgerufen am 4. Januar 2023, von https://github.com/fabio-scala/poitour

[13] The Trip Boutique. (o. D.). The Trip Boutique. Abgerufen am 4. Januar 2023, von https://www.thetripboutique.co/

[14] MAKE MY DRIVE FUN. (o. D.). Abgerufen am 4. Januar 2023, von https://makemydrivefun.com/

[15] Novack,T.,Wang,Z. & Zipf,A. (2018, 8. November). A System for Generating Customized Pleasant Pedestrian Routes Based on OpenStreetMap Data. MDPI. Abgerufen am 4. Januar 2023, von https://www.mdpi.com/1424-8220/18/11/3794

[16] Powerful route planner that prefers greenery and can generate round trip routes of a specified distance. (2020). Trail Router. Abgerufen am 4. Januar 2023, von https://trailrouter.com/

[17] Inspirock. (o. D.). Trip Planner: Plan & manage your vacation itinerary on •. Abgerufen am 4. Januar 2023, von https://www.inspirock.com/

[18] Hannah,C.,Spasic,I. & Corcoran,P. (2018, Juni). A computational model of pedestrian road safety: the long way round is the safe way home - ORCA. Abgerufen am 4. Januar 2023, von https://orca.cardiff.ac.uk/id/eprint/112872/

[19] Matter,J. & Suter,R. (2017). PlazaRoute: Fussgänger-Routing über offene Flächen im urbanen Raum

- ePrints OST. Abgerufen am 4. Januar 2023, von https://eprints.ost.ch/id/eprint/625/
[20] Brixius, N. (2016, 15. Juni). Computing Optimal Road Trips Using Operations Research. Nathan Brixius. Abgerufen am 4. Januar 2023, von https://nathanbrixius.wordpress.com/2016/06/09/computing-optimal-road-trips-using-operations-research/
[21] Walking Route Planner. (o. D.). Abgerufen am 4. Januar 2023, von https://www.plotaroute.com/walkingrouteplanner
[22] Erleben Sie die schönsten Routen von der Schweiz. (o. D.). RouteYou. Abgerufen am 4. Januar 2023, von https://www.routeyou.com/de-ch
[23] Greco, F. (2008). Traveling Salesman Problem. IntechOpen.
[24] PlantUML. (o. D.). Sequence Diagram syntax and features. PlantUML.com. Abgerufen am 9. Januar 2023, von https://plantuml.com/sequence-diagram
[25] JetBrains. (2021, 9. Juni). YouTrack: Project management for all your teams. Abgerufen am 8. Januar 2023, von https://www.jetbrains.com/youtrack/
[26] Statistic - IntelliJ IDEs Plugin | Marketplace. (o. D.). JetBrains Marketplace. Abgerufen am 8. Januar 2023, von https://plugins.jetbrains.com/plugin/4509-statistic
[27] PostgreSQL. (o. D.). PostgreSQL. Abgerufen am 6. Januar 2023, von https://www.postgresql.org/
[28] PostGIS. (o. D.). PostGIS. Abgerufen am 6. Januar 2023, von https://postgis.net/
[29] HSTORE. (2022, 10. November). PostgreSQL Documentation. Abgerufen am 6. Januar 2023, von https://www.postgresql.org/docs/14/hstore.html
[30] Home. (2022, 22. Dezember). Python.org. Abgerufen am 6. Januar 2023, von https://www.python.org/
[31] Ramírez, S. (o. D.). FastAPI. FastAPI. Abgerufen am 6. Januar 2023, von https://fastapi.tiangolo.com/
[32] SQLAlchemy - The Database Toolkit for Python. (o. D.). SQLAlchemy. Abgerufen am 6. Januar 2023, von https://www.sqlalchemy.org/
[33] Welcome to Alembic's documentation! — Alembic 1.9.1 documentation. (o. D.). Abgerufen am 6. Januar 2023, von https://alembic.sqlalchemy.org/en/latest/
[34] Home. (2022, 17. Oktober). OpenAPI Initiative. Abgerufen am 6. Januar 2023, von https://www.openapis.org/
[35] JavaScript With Syntax For Types. (o. D.). TypeScript. Abgerufen am 6. Januar 2023, von https://www.typescriptlang.org/
[36] Angular. (o. D.). Abgerufen am 6. Januar 2023, von https://angular.io/
[37] PrimeFaces. (o. D.). PrimeNG. Abgerufen am 6. Januar 2023, von https://www.primefaces.org/primeng/
[38] Font Awesome. (o. D.). Font Awesome. Abgerufen am 6. Januar 2023, von https://fontawesome.com/
[39] RxJS. (o. D.). Abgerufen am 6. Januar 2023, von https://rxjs.dev/
[40] Docs. (o. D.). NgRx. Abgerufen am 6. Januar 2023, von https://ngrx.io/
[41] Agafonkin, V. (o. D.). Leaflet — an open-source JavaScript library for interactive maps. Abgerufen am 6. Januar 2023, von https://leafletjs.com/
[42] Project OSRM. (o. D.). Abgerufen am 6. Januar 2023,von https://project-osrm.org/
[43] GraphHopper Directions API with Route Optimization. (2022, 19. Oktober). GraphHopper Directions API. Abgerufen am 6. Januar 2023, von https://www.graphhopper.com/
[44] openrouteservice. (o. D.). Abgerufen am 6. Januar 2023, von https://openrouteservice.org/

[45] Team, K. (o. D.). Keycloak. Abgerufen am 6. Januar 2023, von https://www.keycloak.org/
[46] Traefik, The Cloud Native Application Proxy | Traefik Labs. (o. D.). Traefik Labs: Makes Networking Boring. Abgerufen am 6. Januar 2023, von https://traefik.io/traefik/
[47] Ratliff, J. (2022, 25. Oktober). Docker: Accelerated,Containerized Application Development. Docker. Abgerufen am 6. Januar 2023,von https://www.docker.com/
[48] Geofabrik Download Server. (o. D.). Abgerufen am 6. Januar 2023, von https://download.geofabrik.de/
[49] Gluu Inc. (2022, 7. Juli). Gluu 4 | Open Web Standards | SSO, 2FA, MFA. Gluu Identity and Access Management. https://gluu.org/gluu-4/
[50] Gluu Server 4.0 Docs. (o. D.). Gluu. Abgerufen am 9. Januar 2023, von https://gluu.org/docs/gluu-server/4.0/
[51] Keycloak. (o. D.). keycloak/LICENSE.txt at main · keycloak/keycloak. GitHub. Abgerufen am 9. Januar 2023, von https://github.com/keycloak/keycloak/blob/main/LICENSE.txt
[52] OpenIAM. (2021, 10. September). Home. OpenIAM - Open Source Identity Governance & Administration, Web Access Management, MFA and CIAM Platform. https://www.openiam.com/
[53] SuperTokens. (o. D.). SuperTokens, Open Source User Authentication. Abgerufen am 25. Oktober 2022, von https://supertokens.com/
[54] Supertokens. (o. D.). supertokens-core/LICENSE.md at master · supertokens/supertokens-core. GitHub. Abgerufen am 9. Januar 2023, von https://github.com/supertokens/supertokens-core/blob/master/LICENSE.md
[55] Traefik. (o. D.). traefik/LICENSE.md at master · traefik/traefik. GitHub. Abgerufen am 9. Januar 2023, von https://github.com/traefik/traefik/blob/master/LICENSE.md
[56] Traefik Labs. (o. D.). OpenID Connect Authentication - Traefik Enterprise. Abgerufen am 8. Januar 2023, von https://doc.traefik.io/traefik-enterprise/middlewares/oidc/
[57] Apache JMeter. (o. D.). Apache JMeter. Abgerufen am 11. Januar 2023, von https://jmeter.apache.org/
[58] Git. (o. D.). Git. Abgerufen am 9. Januar 2023, von https://git-scm.com/
[59] The DevSecOps Platform. (o. D.). GitLab. Abgerufen am 9. Januar 2023, von https://about.gitlab.com/
[60] Overview. (o. D.). Chrome Developers - Lighthouse. Abgerufen am 11. Januar 2023, von https://developer.chrome.com/docs/lighthouse/overview/
[61] OAuth 2.0 — OAuth. (o. D.). Abgerufen am 9. Januar 2023, von https://oauth.net/2/
[62] OpenID. (2022, 30. Dezember). OpenID Connect | OpenID. OpenID - The Internet Identity Layer. Abgerufen am 9. Januar 2023, von https://openid.net/connect/
[63] OpenStreetMap. (o. D.). OpenStreetMap. openstreetmap.com. Abgerufen am 9. Januar 2023, von https://www.openstreetmap.org/
[64] Home - osm2pgsql. (o. D.). osm2pgsql. Abgerufen am 9. Januar 2023, von https://osm2pgsql.org
[65] pgRouting Project — Open Source Routing Library. (o. D.). pgRouting.org. Abgerufen am 9. Januar 2023, von https://pgrouting.org/
[66] Wikipedia-Autoren. (2004, 16. Juni). Rational Unified Process. Abgerufen am 9. Januar 2023, von https://de.wikipedia.org/wiki/Rational_Unified_Process
[67] Campbell, B. (2005, Mai). RFC 7522: Security Assertion Markup Language (SAML) 2.0. RFC Editor. Abgerufen am 9. Januar 2023, von https://www.rfc-editor.org/rfc/rfc7522
[68] Home. (o. D.). Scrum.org. Abgerufen am 9. Januar 2023, von https://www.scrum.org/

Appendix G: Abbildungsverzeichnis

Appendix H: Tabellenverzeichnis