Digitalisierung von Reisen durch Entwicklung einer intelligenten Reiseführer App

Studienarbeit

im Rahmen der Prüfung zum

Bachelor of Science (B.Sc.)

des Studienganges Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Joshua Schulz und Raphael Müßeler

2020

-Sperrvermerk-

Abgabedatum:	18. Mai 2020
Bearbeitungszeitraum:	01.10.2019 - 18.05.2020
Matrikelnummer, Kurs:	4508858, 6801150, TINF15B1
Ausbildungsfirma:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Gutachter der Dualen Hochschule:	Thorsten Schlachter

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:

Digitalisierung von Reisen durch Entwicklung einer intelligenten Reiseführer App

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, den 16. Februar 2020

Schulz, Joshua; Müßeler, Raphael

Sperrvermerk

Die nachfolgende Arbeit enthält vertrauliche Daten der:

SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf, Deutschland

Sie darf als Leistungsnachweis des Studienganges Angewandte Informatik 2017 an der DHBW Karlsruhe verwendet und nur zu Prüfungszwecken zugänglich gemacht werden. Über den Inhalt ist Stillschweigen zu bewahren. Veröffentlichungen oder Vervielfältigungen der Studienarbeit - auch auszugsweise - sind ohne ausdrückliche Genehmigung der SAP SE nicht gestattet.

SAP und die SAP Logos sind eingetragene Warenzeichen der SAP SE. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in dieser Arbeit berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedem benutzt werden dürfen.

Abstract

- English -

This is the starting point of the Abstract. For the final bachelor thesis, there must be an abstract included in your document. So, start now writing it in German and English. The abstract is a short summary with around 200 to 250 words.

Try to include in this abstract the main question of your work, the methods you used or the main results of your work.

Abstract

- Deutsch -

Dies ist der Beginn des Abstracts. Für die finale Bachelorarbeit musst du ein Abstract in deinem Dokument mit einbauen. So, schreibe es am besten jetzt in Deutsch und Englisch. Das Abstract ist eine kurze Zusammenfassung mit ca. 200 bis 250 Wörtern.

Versuche in das Abstract folgende Punkte aufzunehmen: Fragestellung der Arbeit, methodische Vorgehensweise oder die Hauptergebnisse deiner Arbeit.

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
POI	Point of interest

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

1 Einleitung

(Joshua)

In der heutige Zeit unterliegt die Welt der Medien einer sehr rasanten und starken Veränderung. Es werden immer mehr neuartige und innovative Techniken entwickelt, wie Medien konsumiert werden können, z.B. argumented und virtual reality. Durch diese Techniken wird versucht die Mediennutzung effektiver, intensiver und moderner zu machen. In vielen Bereichen haben diese Techniken bereits Einzug gehalten und sind für eine große Masse von Konsumenten verfügbar, z.B. virtual reality gaming mit Hilfsmitteln wie Oculus rift und anderen Produkten.

Allerdings gibt es ebenfalls Bereiche bei denen die Digitalisierung und Nutzung neuer Techniken kaum genutzt und somit große Vorteile verschenkt werden, wie z.B. beim Reisen: Viele Menschen nutzen Reiseführer, um sich einen Überblick über ihre Reisedestination zu verschaffen und einen grundlegenden Plan zu erstellen, allerdings findet man diese Reiseführer fast ausschließlich in gedruckter Buchform. Das bedeutet, dass immer schwere Printmedien mit in den Urlaub genommen werden müssen, dass Informationen in den Reiseführern nicht aktualisiert werden können, ohne eine neue Auflage herauszugeben und eine interaktive Gestaltung der Medien nicht möglich ist. Außerdem sind die Seiten in einem Buch begrenzt, d.h. es muss für jede Reise ein neuer Reiseführer erworben werden, der spezielle Informationen zum Reiseziel enthält. All dies sind Nachteile, die durch eine Digitalisierung der Inhalte ausgeglichen werden könnten: Das Smartphone ist heutzutage ein ständiger Begleiter, der ausgenutzt werden kann um Echtzeitinformationen schnell und immer aktuell zur Verfügung zu stellen. Außerdem können interaktive Elemente wie Karten, Navigation, Audiogiudes uvm. direkt integriert werden. Es wäre möglich einen Anwendung zu schaffen, die in der Lage ist für jede beliebige Stadt und Region auf der Welt Informationen bereit zu stellen, ohne dass neue Inhalte erworben werden müssen. Damit könnte eine Reise entstehen, die unkomplizierter und trotzdem viel aktueller ist als bei Benutzung eines herkömmlichen Reiseführers.

Es könnten viele Services, die aktuell parallel zum Reiseführer genutzt werden (z.B. verschiedene Bewertungsportale und Karten) direkt integriert werden, um alle Informationen auf einen Blick zur Verfügung zu stellen. Ebenso könnte eine Personalisierung der verfügbaren Daten umgesetzt werden. Im Gegensatz zum herkömmlichen Reiseführer, welcher allgemeine und damit u.U. viele für den einzelnen irrelevant Informationen enthält, werden Vorschläge anhand der vom Nutzer gesetzten Vorlieben gemacht und somit nur nützliche Informationen zur Verfügung gestellt.

Insgesamt soll ein digitaler Reiseführer entstehen, der das Reiseerlebnis auf eine bessere, digitalere und einfachere Ebene hebt und noch mehr Spaß am Reisen erzeugen kann.

1.1 Motivation

(Joshua) Wir möchten mit dieser Arbeit einen Beitrag zu einer digitalisierten und technisch geprägten Gesellschaft leisten, indem wir eine Anwendung schaffen, die mit den Nachteilen von gedruckten Reiseführern aufräumt und eine neue innovative Art des Reisen schafft. Damit soll das Reiseerlebnis der Menschen verbessert werden und ihnen die Möglichkeit geben sich noch mehr auf das Erlebte zu konzentrieren, ohne Gedanken an eine komplizierte Planung, bei der viele Medien parallel genutzt werden, zu verschwenden. Außerdem ist diese Arbeit Teil unserer Prüfung zum Bachelor of Science und dient zur praktischen Anwendung der bisher im Studium erlernten Fähigkeiten und zum Ausprobieren neuer innovativer Techniken um unser Wissen zu erweitern.

1.2 Ausblick auf die Arbeit

2 Voraussetzungen

2.1 Machine Learning

2.2

2.3 Frameworks

2.3.1 Swagger

2.3.2 Android

2.4 Qualitätssichernde Maßnahmen

2.4.1 Code Review

2.4.2 Continuous Integration

2.4.3 Testing Frameworks

3 Anforderungen

(Joshua)

In diesem Kapitel sollen die Anforderungen an die zu erstellende Applikation beschrieben werden, um den zu erreichenden Scope festzulegen und eine abschließende Bewertung durchführen zu können, die das Erreichte mit den Zielen vergleicht.

Zuerst wird die Begrifflichkeit „Anforderung“ geklärt. In der Softwareentwicklung gibt es einige verschiedene Definitionen von „Anforderungen“. In dieser Arbeit soll der Definition gefolgt werden, welche von Helmut Balzert in seinem Buch zu den Basiskonzepten und des Requirements Engineering erarbeitet wurden [**Balzert.2009**]. In seinem Werk werden Anforderungen wie folgt definiert.

> **Anforderungen:** „Anforderungen (requirements) legen fest, was man von einem Definition Softwaresystem als Eigenschaften erwartet.“ Mit der Annahme, dass „man“ alle Stakeholder (Personen, die ein Interesse an der Entwicklung und/oder der erstellten Software haben) beinhaltet. [**Balzert.2009**]

Des Weiteren sollten vor der Festlegung der Anforderungen „Visionen und Ziele“ und die Rahmenbedingungen, in denen die Software existieren sollen, festgelegt werden. Erst danach sollten Eigenschaften, welche in „funktionale“ und „nichtfunktionale“ Eigenschaften unterteilt werden können, definiert werden. In den folgenden Unterkapiteln wird nach diesem Prinzip vorgegangen, um eine konsistente und sinnvolle Anforderungslage zu schaffen.

3.1 Visionen und Ziele

An erster Stelle der Anforderungen steht eine Vision für das zu erstellende Produkt. In diesem Fall wurden bereits einige wichtige Punkte in der Einleitung der Arbeit zusammen gefasst, die zu einer kurzen und Prägnanten Vision führen:

„Durch *Travlyn* sollen Nutzer in der Lage sein, ihre Städtereisen ohne das Mitführen von papierbasierten Reiseführern oder die Nutzung von multiplen mobilen Diensten zu bewältigen, ohne dabei einen Informationsverlust oder eine Beeinträchtigung des Reisespaßes hinnehmen zu müssen."

Anhand dieser Version, die beschreibt, was erreicht werden soll aber nicht wie, werden konkrete Ziele abgeleitet. Es wurde entschieden, dass diese Ziele dem „SMART" Prinzip folgen sollten.

> **SMART Ziele:** Die Art und Weise messbare Ziele zu setzen kann einer festgelegten Struktur folgen. In diesem Fall soll die Struktur, welche Peter Drucker in seinem Buch „The Practice of Management" (1954) erarbeitet hat, genutzt werden. Allerdings hat Drucker nie eine genaue Erklärung zu der Bedeutung des Akronyms „SMART" abgegeben, deswegen wird folgende allgemein akzeptierte Vaiante gewählt[**Lawlor.2012**]:
>
> - Specific (Spezifisch)
> - Measurable (Messbar)
> - Attainable (Erreichbar)
> - Realistic (Realistisch)
> - Timely (Rechtzeitig)

Folgend diesem Prinzip sind folgende Ziele entstanden:

- Ein Nutzer soll sich vor einer Reise über *Travlyn* folgende Informationen zu seiner Zielstadt einholen können: Name, Lage, Beschreibung und ein Bild, welches einen ersten Eindruck der Stadt vermittelt.
- Vor und während der Reise wird *Travlyn* den Nutzer entlang einer vorher festlegten Route durch die Stadt leiten und Informationen, wie Beschreibungen, Bilder und Kosten anzeigen. Die Route beinhaltet Stops an interessanten Orten der Stadt, wie Sehenswürdigkeiten und spannenden Aktivitäten.
- Während der Führung durch *Travlyn* werden einzelne Texte per machine learning zu einer zusammenhängenden Führung zusammengefasst und von Android vorgelesen um den Eindruck eines realen Guides zu schaffen.

- Der Nutzer kann Reisepläne teilen und Pläne von anderen Nutzern einsehen können, um sich inspirieren zu lassen und seinen eigenen Plan an den existierenden Plänen orientieren zu können.

3.2 Rahmenbedingungen

Laut Balzert stellen Rahmenbedingungen „organisatorische und/oder technische Restriktionen für das Softwaresystem und/oder den Entwicklungsprozess“ da [**Balzert.2009**].

3.2.1 Organisatorisches

Für „Travyln“ liegen folgende organisatorische Rahmenbedingungen vor: Der Anwendungsbereich der Software liegt im privaten Umfeld, genauer gesagt im Bereich des privaten Reisens. Die Zielgruppe sind alle Personen, die für relativ kurze Zeiträume in größere Städte reisen und diese mit ihren Sehenswürdigkeiten entdecken wollen und sich zu diesen weiter informieren wollen. Damit liegt eine mobile Benutzung unter ständiger Beobachtung des Nutzers vor. Während der Reise/der durch *Travyln* geführten Tour wird die Anwendung ohne Unterbrechung laufen.

3.2.2 Technisches

Die technischen Anforderungen werden an dieser Stelle in zwei Abschnitte aufgeteilt, da sich die Rahmenbedingungen für den Server und den Client stark unterscheiden.

Für den Server wird festgelegt, dass er in einem Docker-Container[**TODO**] läuft, welcher unabhängig von dem darunterliegendem Betriebssystem ist. Peripherie wird es an diesem Rechner keine geben, da er nur per remote Zugriff von außen gesteuert werden wird. Die wichtigste Rahmenbedingung ist, dass die Hardware auf der der Server läuft ständig mit dem Internet verbunden sein muss, um eine dauerhafte Verfügbarkeit zu gewährleisten

Für den Client wird festgelegt, dass er auf einem mobilen Smartphone, welches im Akkubetrieb operiert, läuft. Auf dem Gerät muss Android als Betriebssystem laufen und die Version soll 7.0.0 (TODO) nicht unterschreiten. Das Gerät stellt eine klassische mobile Peripherie zur Verfügung, welche z.B. eine virtuelle Tastatur, einen GPS-Sensor und einen Lautsprecher/Kopfhöreranschluss beinhaltet. Auch für den Client muss eine konstante Internetverbindung existieren, um sicher zu stellen, dass der Server ständig erreicht werden kann und die entsprechenden Informationen abgefragt werden können.

3.3 Geforderte Eigenschaften

Nachdem die Ziele und die Rahmenbedingungen definiert sind, können die erwarteten Eigenschaften erarbeitet werden. Bei jeder Eigenschaft sollte geprüft werden, ob diese im Sinne eines der gesteckten Ziele sind und zum Erreichen der Vision beitragen und ob diese im Sinne der Rahmenbedingungen erreichbar und realistisch ist. Wenn eine dieser beiden Prüfungen nicht positiv erfüllt wird sollte über eine Redefinition der Eigenschaft nachgedacht werden, denn in der aktuellen Form ist sie nicht zielführend und kann mit dem vorliegendem Rahmen ggf. nicht umgesetzt werden.

Wie im vorherigen Verlauf beschreiben werden Eigenschaften häufig in „funktional“ und „nicht-funktional“ aufgeteilt [**Balzert.2009**]. Im Folgenden wird dieser Trennung gefolgt.

3.3.1 Funktionale Eigenschaften

Unter funktionalen Eigenschaften wird alles spezifiziert, was ein System tun und explizit nicht tun/können soll. Laut Balzert können diese Eigenschaften in statische, dynamische und logische Eigenschaften aufgeteilt werden[**Balzert.2009**]. Wir haben uns explizit gegen eine solche Gliederung entschieden, um einen zu großen Mehraufwand zu sparen und den Rahmen dieses Kapitels nicht zu sprengen.

Zur Visualisierung der erwarteten funktionalen Eigenschaften wurde ein Use Case Diagramm erstellt.

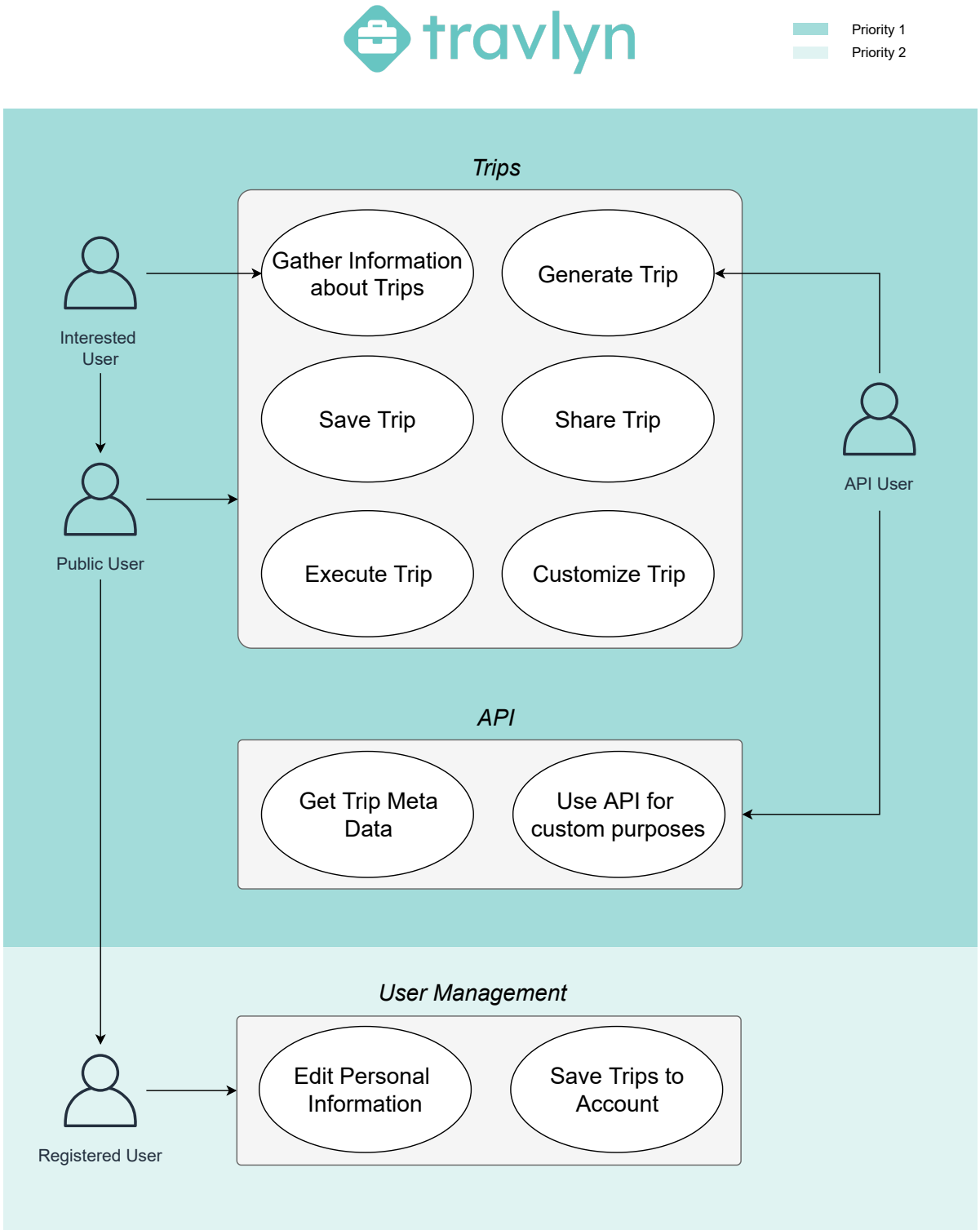

Abbildung 3.1: Darstellung aller geplanten use cases mit den assoziierten Nutzerprofilen.

In der vorliegenden Grafik wird der Begriff „Trip“ geprägt, welcher sich durch die gesamte Softwareentwicklung und Arbeit ziehen wird.

> **Trip:** Ein Trip ist eine durch den Server erstellte Abfolge von sehenswerten Punkten in einer Stadt. Damit ist jeder Trip eindeutig einer Stadt zugeordnet. Neben den Punkten und der Route um sich entlang der Punkte zu bewegen, werden weitere Metainformationen zu den Punkten geliefert, die einen Besuch noch interessanter machen.

Abbildung 3.1 zeigt neben den use cases auch die Nutzerprofile, denen die use cases zugeordnet sind. Für *Travyln* sind vier Nutzerprofile geplant. Der „interessierte Nutzer“ benutzt die App eher zufällig und ohne die Intention sie aktiv für eine Reise einzusetzen. Sein Ziel ist es sich über die Funktionalität und bereits bestehende Trips zu informieren. Sollte der Nutzer in dieser Phase ansprechende Informationen finden und sich entscheiden [Travyln] für die nächste Reise einzusetzen entwickelt er sich zum „Öffentlichen Nutzer“, welcher ohne sich anzumelden oder registrieren alle Aktionen auf Trips ausführen kann und die volle Funktionalität zur Durchführung von Trips nutzt. Darauf aufbauend ist eine freiwillige Registrierung möglich, um persönliche Daten zu hinterlegen und alle eigenen Trips in an einem persönlichen Platz zu speichern. Alle Nutzer die sich dafür entscheiden liegen im Profil „registrierter Nutzer“.

Da die RESTAPI, die im Zuge dieser Softwareentwicklung erstellt wird öffentlich angeboten wird, ist es möglich diese API in fremden Applikationen zu eigenen Zwecken zu Nutzen, in dieser Form bildet sich das Profil „API Nutzer“.

Alle in Abbildung 3.1 dargestellten use cases wurden auf ihre Kompatibilität zu Zielen und Rahmenbedingungen geprüft und für passend befunden. Zusätzlich wurden sie bereits grob priorisiert. Es ist wichtig zu erwähnen, dass viele der dargestellten use cases im Laufe der Entwicklung zu kleinen Einheiten aufteilt werden, die hier im Sinne der Übersichtlichkeit nicht dargestellt sind.

Abschließend ist zu erwähnen, dass die funktionalen Anforderungen der in der Einführung geschilderten Funktionalität folgen soll.

3.3.2 Non-Funktionale Eigenschaften

Neben den geschilderten funktionalen Eigenschaften gibt es weitere Anforderungen, welche sich nicht direkt auf die Funktionalität und den Funktionalitätsumfang auswirken. Diese Eigenschaften werden auch „Quality of Service“ (QoS) genannt und beinhalten häufig Eigenschaften wie Genauigkeit, Verfügbarkeit und Konsumierbarkeit. Es ist zu erwähnen, dass einige dieser Anforderungen miteinander in Konflikt stehen und ein geeigneter Mittelweg gefunden werden muss bzw. bestimmte Tradeoffs eingegangen werden müssen, z.B. schränkt die Eigenschaft der möglichst hohen Sicherheit meist die Eigenschaft der Benutzbarkeit oder der Speichereffizienz ein [**Balzert.2009**].

Für diese Softwareentwicklung soll für die non-funktionalen Eigenschaften als Orientierung die internationale ISO/IEC 25010 Norm gelten. Diese Norm setzt Standards für die Qualitätskriterien und -bewertung von Software und ist in drei Bereiche aufgeteilt[**ISO.2011**] [**Braun.2016**]:

- **Quality In Use Model**: Dieser Teil beschreibt alle Merkmale, welche die Interaktion zwischen Mensch und System beschreiben, wie z.B. Effektivität und Freiheit von Risiken.
- **Product Qualitiy Model**: Der zweite Teil der Norm beschreibt acht Charakteristiken, welche ein Softwareprodukt erfüllen sollte, u.a. Wartbarkeit, Sicherheit und Benutzbarkeit.
- **Data Quality Model**: Der dritte und letzte Teil beschreibt Ansprüche, welche an das Datenmodel gestellt werden sollten um eine möglichst konsistente Benutzung der Daten zu ermöglichen.

Diese Norm ist sehr umfangreich und für ein Studienprojekt nicht vollständig umsetzbar, ohne den Zeit- und Aufwandsrahmen zu sprengen, deshalb wurde einige wichtige Punkte ausgewählt auf die ein besonderes Augenmerk gelegt werden soll. Zusätzlich wurden einige themenspezifische Anforderungen hinzugefügt. Daraus ergibt sich folgende Liste:

- **Benutzbarkeit**: Durch die intensive mobile Benutzung der *Travyln* App sollte diese für alle gängigen Gerätetypen eine gute Nutzungserfahrung bieten, die

Nutzer überzeugen kann und die Reise angenehm verlaufen lässt. Explizit soll an dieser Stelle die Performance der Software genannt werden, welche bei anderen Applikationen häufig für eine schlechte Benutzbarkeit sorgt. Außerdem sollen ansprechende User Interfaces designed werden, die den Nutzer zur Nutzung einladen und z.B. über Spracheinstellungen für möglichst viele Personen anpassbar sind.

- **Sicherheit**: Der Schutz persönlicher Daten wir immer wichtiger und soll auch bei *Travyln* nicht vernachlässigt werden. Sowohl die API als auch der Client sollen nicht anfällig für gängige Angriffe, wie Injections oder Denail of Service sein und sicherstellen, dass persönliche Daten nur an authentifizierte Nutzer ausgeliefert/angezeigt werden.

- **Wartbarkeit**: Die Software sollte über die bereits beschriebenen qualitätssichernden Maßnahmen und durch eine entsprechende Architektur gut wart- und erweiterbar sein. An dieser Stelle ist der Verzicht auf „code ownership“ besonders hervorzuheben, d.h. alle Teilnehmer des Entwicklungsteams haben Einblicke in alle Teile des Codes und können im Notfall Korrekturen und Erweiterungen vornehmen. Es ist nicht erwünscht, dass einzelne Code Teile nur von einer Person geschrieben, gepflegt und gewartet werden. Außerdem soll durch den Server eine ausführliche Dokumentation der Benutzung angefertigt werden, um auftretende Fehler identifizieren und beheben zu können. Hierzu sollen alle Aktionen und ihr Ergebnis in geeigneter Weise persistiert werden.

- **Datenquellen**: Da diese Software im Rahmen eines Studienprojekts mit sehr begrenzten Ressourcen erstellt wird, können keine kostenpflichtige Services eingebunden werden. Außerdem sollen schwierige Lizenzfragen vermieden werden, indem komplett auf Opensource Dienste gesetzt wird, bei denen die Nutzung der Daten vollständig erlaubt und frei ist. In den folgenden Kapiteln wird genauer auf dieses Thema eingegangen und die verschiedenen Alternativen um Daten zu beschaffen genauer vorgestellt.

- **Zuverlässigkeit**: Der Nutzer soll sich auf *Travyln* verlassen können. Es ist wichtig, dass vor allem während einer Reise alle Funktionen zur Verfügung stehen und verhindern, dass der Nutzer alleine gelassen bzw. gezwungen wird auf an-

dere Diensten zurückgreifen zu müssen. Außerdem sollten die downtimes für die API und den Client so gering wie möglich sein.

4 Datengrundlage

(Joshua)

Die Datengrundlage ist für einen Reiseführer sehr wichtig, da der gesamte Sinn eines Reiseführers darauf basiert, Informationen aufzubereiten und an den Leser/Nutzer weiter zu geben. Aus diesem Grund wurden für diese Arbeit mehrere Datenquellen zu unterschiedlichen Themen herausgesucht und verglichen. Im Folgenden sollen diese Alternativen und die angestellten Überlegungen sowie die endgültige Entscheidung, welche Daten für *travlyn* verwendet werden sollen, aufgezeigt.

4.1 Points of Interest

Travlyn soll laut Spezifikation Trips erstellen können, welche eine Abfolge von interessanten und sehenswerten Punkten in einer Stadt ist. Diese Point of interest (POI) sollen über ein Application Programming Interface (API) in die App integriert werden. Folgende Anforderungen sind an die Informationen und die API gestellt:

- Die abgefragten Daten sollten möglichst für einen kommerziellen Nutzen zugelassen sein, damit während der Entwicklung keine schwierigen Lizenzfragen auftreten können und ggf. höhere Datenvolumen durch Nachfragen erreicht werden können.
- Da diese Arbeit ein Studienprojekt ist, für welches sehr begrenzte Ressourcen zur Verfügung stehen sollte die API kostenfrei benutzbar sein.
- Die API sollte Daten für möglichst viele Städte/Orte zur Verfügung stellen, damit *travlyn* möglichst überall eingesetzt werden kann.
- Zu den einzelnen POIs sollten neben dem Namen und der Position weitere Daten wie Beschreibungen, Öffnungszeiten und ggf. Bilder bereitgestellt werden.

4.1.1 Google Places API

Google ist einer der größten Anbieter von Ortsbasierten Services/Diensten und stellt eine API für POIs zu Verfügung [**Google.01.02.2020**]. Diese API hat sehr weit gefächerte Funktionen, die von einer einfachen Suche über ausführliche Details zu interessanten Orten bis hin zu „user check-in“ an einzeln Orten reichen. Diese große Funktionalität wäre für *travlyn* sehr wertvoll. Allerdings sind die Google APIs nicht frei zugänglich und die Anzahl der Requests ist u.U. stark eingeschränkt [**Singhal.2012**]. Außerdem ist die Nutzung der erhaltenen Daten nur in Verbindung mit anderen von Google bereitgestellten Services erlaubt [**Google.02.12.2019**], somit wäre die ganze *travlyn* Applikation an Google gebunden.

4.1.2 Openroute service

Openroute service [**TheHeidelbergInstituteforGeoinformationTechnology.**] wird vom Heidelberger Institut für Geoinformationstechnik angeboten. Es handelt sich um eine Crowd Sourced API, d.h. sie wird durch Benutzer über OpenStreetMap (OSM) [**OpenStreetMap.**] gespeist und ist damit frei zugänglich. Durch die Nutzung von OSM ergibt sich der weitere Vorteil, dass die API für Orte weltweit nutzbar ist. Leider sind die gelieferten Informationen nicht sehr umfangreich und beinhalten häufig keine genauere Beschreibung und keine Bewertung o.Ä.. Weiterhin sind Crowd Sourced Informationen meist nicht offiziell verifiziert und könnten u.U. falsch sein. Die Beschränkungen für diese API sind relativ gering (500 POIs requests pro Tag), allerdings können diese auf Nachfrage erhöht werden (z.B. für Bildungszwecke).

> **Crowdsourcing:** Beim Crowdsouring wird das Wissen, die Kreativität oder die Arbeitskraft der Masse ausgenutzt. Jeder leistet einen kleinen Teil und zusammen ergibt sich ein großes Ganzes. Typische Beispiele sind z.B. Wikipedia oder die Klassifikation von Daten zum Machine Learning. Allerdings können diese Daten von jedem bewusst oder unbewusst verfälscht werden und sie sind sehr schwer zu verifizieren [**Winkler.2009**].

4.1.3 Foursquare

Die Firma Foursquare bietet ebenfalls eine API an [**Foursquare.**], über die Informationen zu interessanten Orten gelesen werden können. Die API ist weltweit einsetzbar und liefert sehr viele Informationen zu einzelnen Orten, wie Ratings, kurze Beschreibungen oder Adressen von denen Bilder in der gewünschten Auflösung abgefragt werden können. Die Anzahl der möglichen Requests kann durch die Registrierung einer Kreditkarte (trotz der kostenlosen Nutzung) auf ca. 100.000 pro Tag gesteigert werden. Allerdings können die kostenfreien Varianten dieser API nicht für kommerzielle Zwecke genutzt werden und die abgefragten Daten dürfen nicht länger als 24 Stunden persistiert werden.

4.1.4 Evaluation der Alternativen

Für das Projekt wurde aus den obigen Alternativen gewählt. Die endgültige Entscheidung fiel auf den OpenRoute service, da es für unser relativ kurzes Projekt ein sehr großes Risiko darstellt, dass langwierige Nachforschungen zu Lizenzfragen und erlaubten Einsatzmöglichkeiten angestellt müssen. Openroute service ist selbst für kommerziellen Nutzen freigegeben und aus rechtlicher Perspektive damit sehr einfach zu nutzen. Im Gegensatz dazu stellen sowohl Google als auch Foursquare starke Einschränkungen auf, die ein nicht abzusehendes Risiko darstellen, z.B. sind bestimmte Funktionen unter diesen Einschränkungen umsetzbar. Trotz diesem entscheidenden Vorteil muss der Nachteil in Kauf genommen werden, dass nur mäßig ausführliche Informationen gelesen werden können. Im weiteren Verlauf dieser Arbeit werden weitere zusätzliche Datenquellen aufgezeigt, um diesen Nachteil möglichst gut aufzufangen.

4.2 Weiterführende Informationen zu POIs und Städten

Um die eingeschränkte API Openroute service auszugleichen und dem Nutzer weitere Informationen zu seinem Reiseziel und zu besuchenden Orten zu bieten müssen weitere APIs angefragt werden.

Die wahrscheinlich bekannteste und ausführlichste Datenquelle ist Wikipedia. Dies ist einen Crowd Sourced Enzyklopädie die von allen Nutzern gespeist werden kann. Aus diesem Grund ist die Nutzung direkt im Internet aber auch per API Zugriff kostenlos und frei nutzbar. Allerdings ist zu beachten, dass die enthaltenen Inforationen durch jeden verändert und ggf. gefälscht werden können und Wikipedia deshalb keine sichere Quelle für wissenschaftliche Arbeiten o.Ä. darstellt. Für den in dieser Arbeit vorliegenden Use Case wurde entscheiden, dass dieses Risiko annehmbar ist und der Vorteil der sehr großen Wissensbasis das Risiko überwiegen.

Für den Zugriff auf Wikipedia gibt es unterschiedliche Möglichkeiten, im Folgenden werden zwei APIs beschreiben, die ausprobiert worden sind:

- **MediaWiki**: Hinter Wikipedia und vielen anderen Wiki-Seiten steht die selbe Software: MediaWiki [**MediaWiki.24.01.2020**]. Diese Software bietet eine sogenannte *MediaWiki action API*, die viele Informationen zu allen Artikeln eines Wiki zurückliefern kann. Leider sind die Daten nicht über einen zentralen Aufruf abrufbar sondern es werden mehrere Abfragen in folge benötigt, um z.B. die URL eines der Bilder des Artikels zu ermitteln.
- **DBpedia**: Die zweite API, die mithilfe eines Protypes getestet wurde ist *DBpedia* [**DBpedia.02.02.2020**]. Auch diese API folgt dem crowd souring Prinzip und ist somit frei zugänglich. Zusätzlich können dort alle Informationen über einen zentralen Zugriff abgerufen werden, indem die benötigten Daten über URL-Parameter spezifiziert werden können. Außerdem bietet diese API die Daten in aufbereiteter Form an: Links können direkt aufgelöst werden, es wird das selbe Thumbnail ausgeliefert welches im original Artikel ausgewählt ist und es kann auf alle Daten der kompakten Infobox in der oberen rechten Ecke strukturiert zugegriffen werden.

Durch die einfachere Handhabung der *DBpedia* API wurde für den weiteren Verlauf entschieden auf diese API zu setzen und alle Informationen zu POIs und Städten von diesem Zugang abzufragen. Damit können erweiterte Infos geladen werden, die dem Nutzer während seines Trips kontinuierlich angezeigt werden können, um die Erfahrung weiter zu verbessern.

5 Konzept

5.1 Server Architektur

5.2 Client Architektur

5.3 Kommunikationsschema

6 Implementierung

6.1 Coding Conventions

6.2 Probleme

7 Evaluation

7.1 User Zufriedenheit

7.2 Aufbau

7.3 Ablauf

7.4 Ergebnis

8 Fazit

8.1 Ausblick