HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Stadt Zürich

Zugang für alle
Accès pour tous
Accesso per tutti
Access for all

Accessible Map App

Bachelorarbeit
Abteilung Informatik
Hochschule für Technik Rapperswil
Herbstsemester 2013/2014

Autorinnen:	Gwendoline Rothauser, Julia Schmucki
Betreuer:	Stefan Keller
Projektpartner:	Stadt Zürich, Stiftung „Zugang für alle“
Experte:	Claude Eisenhut
Gegenleser:	Eduard Glatz

Impressum

Studentin 1: Name: Gwendoline Rothauser
Studentin 2: Name: Julia Schmucki
Betreuer: Name: Stefan Keller
Experte: Name: Claude Eisenhut
Gegenleser: Name: Eduard Glatz

Revision

Datum	Person	Bearbeiteter Teil
30.09.2013	Julia Schmucki	Analyse bestehende Apps
15.10.2013	Gwendoline Rothauser	Analyse Fussgänger Routing erfasst
25.11.2013	Julia Schmucki	Abstract, Stand der Technik
28.11.2013	Julia Schmucki	Aufgabenstellung, Use Cases, Resultate, Objektdiagramm
09.12.2013	Gwendoline Rothauser	Impressum/Revision, Abstract, Gliederung
10.12.2013	Gwendoline Rothauser	Gliederung, Management Summary
10.12.2013	Julia Schmucki	Umsetzung (SW)
11.12.2013	Julia Schmucki	Konzeptideen
12.12.2013	Gwendoline Rothauser	Management Summary und Gliederung angepasst, Umsetzung
12.12.2013	Julia Schmucki	Objektdiagramm
13.12.2013	Gwendoline Rothauser	Implementation
15.12.2013	Julia Schmucki	Umsetzung
15.12.2013	Gwendoline Rothauser	Abbildungen in Management Summary, Resultate
17.12.2013	Julia Schmucki	Berechnung der Ausgabe für POIs, Grafiken zu Algorithmen
17.12.2013	Gwendoline Rothauser	Risikomanagement, Resultate
19.12.2013	Julia Schmucki, Gwendoline Rothauser	Abschlusskorrektur

Abstract

Um blinden und sehbehinderten Menschen eine Möglichkeit zu bieten, ihre Umgebung besser erkunden zu können und mehr Informationen über ihre nähere Umgebung zu erhalten, wurde in dieser Arbeit in Zusammenarbeit mit der Stiftung „Zugang für alle“ und der Stadt Zürich eine mobile Web Applikation erstellt. Sie kann auf einem Computer sowie auf einem Smartphone mit beliebigem Betriebssystem genutzt werden. Durch die im Smartphone

verfügbare oder auf dem Computer installierte Vorlesefunktion erhält der Blinde oder Sehbehinderte Zugriff auf die dargestellten Informationen und vorhandenen Bedienelemente. Bisher gab es nur Anwendungen, welche dem Benutzer sogenannte POIs (Points of Interest) wie zum Beispiel Restaurants, Parks und Einkaufsmöglichkeiten in seiner näheren Umgebung ausgeben.

Aus der Arbeit ging eine Anwendung hervor, die den Standort des Nutzers bestimmt und gewünschte Informationen aus OpenStreetMap, einer Plattform für öffentliche Kartendaten, bezieht. Im Gegensatz zu bereits vorhandenen Applikationen bietet die entwickelte Anwendung ein Fussgängerrouting sowie eine Standortausgabe, die mit für blinde und sehbehinderte Menschen wichtigen Orientierungspunkten (Points of Orientation, kurz POO) angereichert wird. Dazu gehört das Einteilen der Ausgabe in zwei Strassenseiten. Als POOs gelten Brunnen, Container, Sitzbänke, Hydranten, Fussgängerstreifen, Lichtsignale, Baustellen, Kreuzungen und Bäume sowie Abfalleimer. Durch die Angabe von Tunnels, Brücken und Treppen wird die Routenbeschreibung noch detaillierter. Zudem werden die Maximalgeschwindigkeit und der Strassenbelag zum jeweiligen Routenabschnitt angegeben, sofern diese Attribute in OpenStreetMap erfasst sind. Ist die Maximalgeschwindigkeit zum Beispiel 20 km/h, so weiss der Blinde, dass er keinen Fussgängerstreifen zu suchen braucht.

Die Daten von Bäumen und Abfalleimern im Gebiet der Stadt Zürich stammen aus Open Government Data, dies sind öffentliche Daten der Stadt Zürich. Sie werden jährlich einmal aktualisiert und sind im Gegensatz zu den OpenStreetMap Daten zwar nicht öffentlich editierbar, dafür aber vollständiger.

Die Web Applikation ist unter www.accessiblemap.ch erreichbar.

Management Summary und Web-Publikation

Ausgangslage

Die Stadt Zürich stellt seit Juni 2012 auf dem Open Government Data Portal viele Daten öffentlich zur Verfügung. Dies sind zum Beispiel der Stadtplan, alle öffentlichen Toiletten, alle Schulen, alle Bus- und Tramhaltestellen, alle Restaurants und vieles mehr. Die Aufgabenstellung war nun, diese Informationen für blinde oder sehbehinderte Menschen, welche ein Smartphone besitzen, zugänglich zu machen. Dies sollte in der Form einer Standortausgabe erfolgen, welche die Umgebung vorliest, zum Beispiel: „12 Uhr Bahnhofstrasse, 3 Uhr Waagstrasse, 4 Uhr Poststrasse, 8 Uhr Bleicherweg, 10 Uhr Talacker“. Diese Ausgabe soll den blinden und sehbehinderten Personen helfen, sich besser in der Stadt Zürich zu orientieren. Verwendet werden sollten die öffentlichen Daten der Stadt Zürich sowie weiteres öffentlich zugängliches Kartenmaterial.

Nach dem ersten Meeting mit der Stiftung „Zugang für alle“ und zwei von einer Sehbehinderung betroffenen Personen stellte sich heraus, dass es bereits genügend Anwendungen gibt, die sogenannte Points of Interest (Restaurants, Einkaufsmöglichkeiten etc.) in der Umgebung ausgeben. Eine weitere Anwendung, die genau dies macht, würde somit keinen wirklichen Mehrwert für die Betroffenen generieren. Es musste deshalb das grundsätzliche Konzept der Arbeit, so wie es in der Aufgabenstellung definiert war, komplett überarbeitet werden. Zusammen mit der Stiftung wurde das Konzept der sogenannten „Orientierungspunkte“ (Points of Orientation, kurz POO) entwickelt. Dies sind Objekte, welche den blinden oder sehbehinderten Personen im Alltag helfen, sich in ihrer Umgebung zu orientieren. Namentlich sind dies Brunnen, Abfalleimer, Fussgängerstreifen, Kreuzungen, Sitzbänke, Container, Bäume, Baustellen und Hydranten. Wenn eine blinde oder sehbehinderte Person eine Route läuft, die an einem Brunnen vorbeiführt und sie diesen beim Vorbeilaufen plätschern hört, dann weiss sie, dass sie auf dem richtigen Weg ist.

Anhand dieser Erkenntnis wurde auch ein mit diesen Informationen angereichertes Routing von den Betroffenen gewünscht. Die bereits vorhandenen Anwendungen bieten nur wenig Hilfe um zu den POIs (kurz für Points of Interest) zu gelangen. Es konnten nur zwei iPhone Applikationen gefunden werden, die eine Fussgängernavigation anbieten. Das zugrundeliegende Kartenmaterial (Google und Apple Maps) der Anwendungen verfügt jedoch nicht über so feinkörnige Daten wie die definierten Orientierungspunkte. Die Anwendungen wählten im Test zudem nicht die vorhandenen Fussgängerwege wie Unter- und Überführungen, sondern nahmen den Weg, den ein Auto nehmen würde. Dieser kann unter Umständen viel länger oder für Fussgänger gar nicht zugänglich sein. Auch hilfreich ist das Anreichern einer Route mit aktuellen Baustellen. Diese werden von der Webseite TROBDB bezogen, die schweizweit alle eingetragenen Baustellen und Verkehrshindernisse auf staatlichen Strassen führt.

Ergebnisse

Die Idee, ein Routing für Blinde und Sehbehinderte mit Orientierungspunkten anzureichern, ist völlig neu. Um die Route generieren zu lassen wurden einige Fussgänger Routing Dienste auf einer für sehbehinderte Fussgänger kritischen Route getestet. Diese beinhaltet zum Beispiel das Überqueren einer stark befahrenen Strasse oder eine Strecke, die eine Bahnhofunterführung als Abkürzung verwenden könnte. Am besten schnitten dabei OSRM (OpenStreetMap Routing Machine) und YOURS (Yet another OpenStreetMap Routing Service) ab. Aufgrund der häufigeren Aktualisierungen der zugrundliegenden Daten wurde OSRM gewählt. Dieser gibt die jeweiligen Wegkoordinaten einer angefragten Route zurück. In der Arbeit wurde ein Algorithmus entwickelt, der daraus eine Route mit einzelnen Wegabschnitten generiert.

Im ersten Schritt werden die Wegkoordinaten, anhand der Berechnung der kleinsten Distanz, auf sogenannte „Nodes“ (in Abbildung 1 als orange Kreise dargestellt), abgebildet. Diese „Nodes“ sind in Strassen immer dort, wo sie beginnen, enden oder eine andere Strassen kreuzen (siehe „Node“ mit der Nummer 2). „Nodes“ haben mindestens eine eindeutige Identifikationsnummer und eine geographische Koordinate. Sie werden in OSM für Knotenpunkte von Strassen, POIs, POOs oder andere Objekte verwendet. Daher werden für die Route in diesem ersten Schritt nur „Nodes“ von Strassen gesucht. Für die Suche in OpenStreetMap wird eine sogenannte „bounding box“ benötigt.

Abbildung 1 - Nodes auf Karte

Dies ist ein Rahmen mit höchster und niedrigster Koordinate der Route, dem noch 10m Abstand für Orientierungspunkte an den Ecken hinzugefügt werden. Ein Beispiel für eine solche „bounding box“ ist der in Abbildung 1 sichtbare Bereich.

Anhand der nun den Wegkoordinaten zugewiesenen „Nodes" werden dann die sogenannten „Ways“ auf der Route gesucht. Ein „Way“ in OpenStreetMap ist nicht mehr als eine Linie. Er kann damit von Strassen über Flüsse bis zu Kanten eines Gebäudes eigentlich alles darstellen, was nicht nur als Punkt („Node“) gekennzeichnet wird. Zusätzlich kann er verschiedene sogenannte „Tags“ (so werden Merkmale in OpenStreetMap bezeichnet) besitzen. Ein „Way“, der den „Tag“ „highway“ besitzt, kann nur eine Strasse, eine Brücke, ein Tunnel, ein Fussweg, ein Fahrradweg oder eine Kante eines Platzes sein.

Die Suche nach Strassen auf einer Route begrenzt sich deshalb auf die Suche nach „Ways“ mit diesem „Tag“.

Immer dann, wenn ein „Node“ der Route nur einen gemeinsamen „Way“ mit dem nächsten „Node“ der Route aufweist, gilt dieser „Way“ als ein Teil der Route. Für den letzten „Node“ der Route wird rückwärts gerechnet um den richtigen „Way“ zu finden. Anschliessend wird nach Orientierungspunkten in den jeweiligen Routenabschnitten gesucht, die entweder auf die linke oder rechte Strassenseite eingeteilt werden. Dazu werden berechnete „Buffer“ um jede Strassenseite eines Strassenabschnitts herum gespannt. Wenn sich die Koordinate eines Orientierungspunkts in einem „Buffer“ befindet, so wird er in jenem Routenabschnitt erwähnt. So wird der in Abbildung 1 blau umkreiste grüne Punkt neben dem „Node“ mit der Nummer 2 auf der linken Strassenseite als Baum erwähnt. Anhand der so angereicherten Route weiss der Blinde oder Sehbehinderte, ob er sich noch auf dem richtigen Weg befindet. Zudem wird die Route mit den Informationen zur Maximalgeschwindigkeit oder dem Bodenbelag ergänzt, sofern diese als „Tags“ für die „Ways“ in OpenStreetMap eingetragen sind. Ist die Maximalgeschwindigkeit zum Beispiel 20 km/h, so befindet sich der Anwender in einer Begegnungszone und weiss, dass er keinen Fussgängerstreifen zu suchen braucht.

In Abbildung 2 ist ein Kartenausschnitt aus OpenStreetMap sichtbar und in Abbildung 3 ist ersichtlich, wie die Daten anhand dieser Karte in einer Route in der entwickelten Anwendung dargestellt werden. Man sieht klar, dass die vielen Bäume (blaue Kreise mit grünen Punkten in der Mitte) am Seequai in der Routenbeschreibung erscheinen. Die in Abbildung 2 dargestellte Route wird mit anfangs nach Norden ausgerichtetem Kompass berechnet.

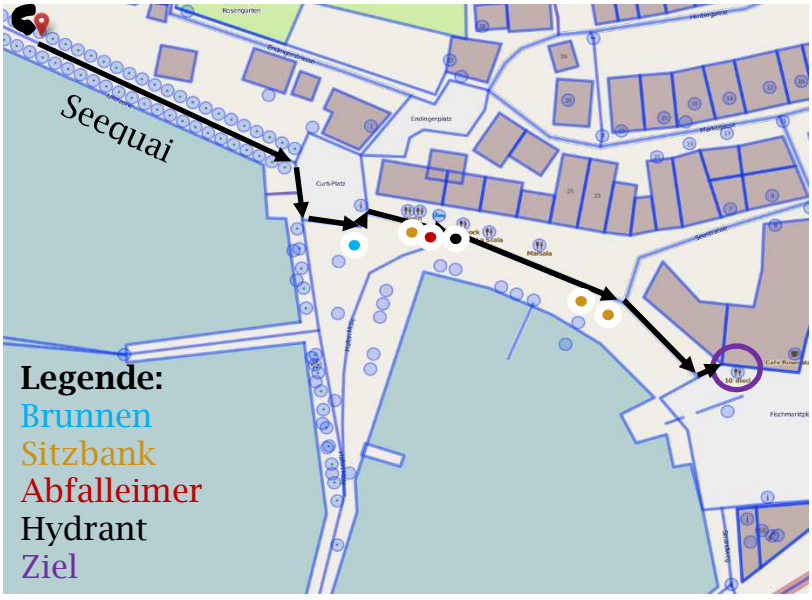


Abbildung 2 - Kartenausschnitt OSM Route Seequai nach 10‘ dieci

Deshalb lautet die erste Anweisung in Abbildung 3 links und rechts jeweils „scharf links abbiegen“. Sollte der Blinde die Strassenseite gewechselt haben, so kann er mittels „zeige rechte Seite“ zu den Orientierungspunkten auf der anderen Strassenseite wechseln. Wenn er das Gefühl hat, er habe sich von der Route wegbewegt, so kann er die Route aktualisieren. Sie wird dann von seinem aktuellen Standort aus neu berechnet.

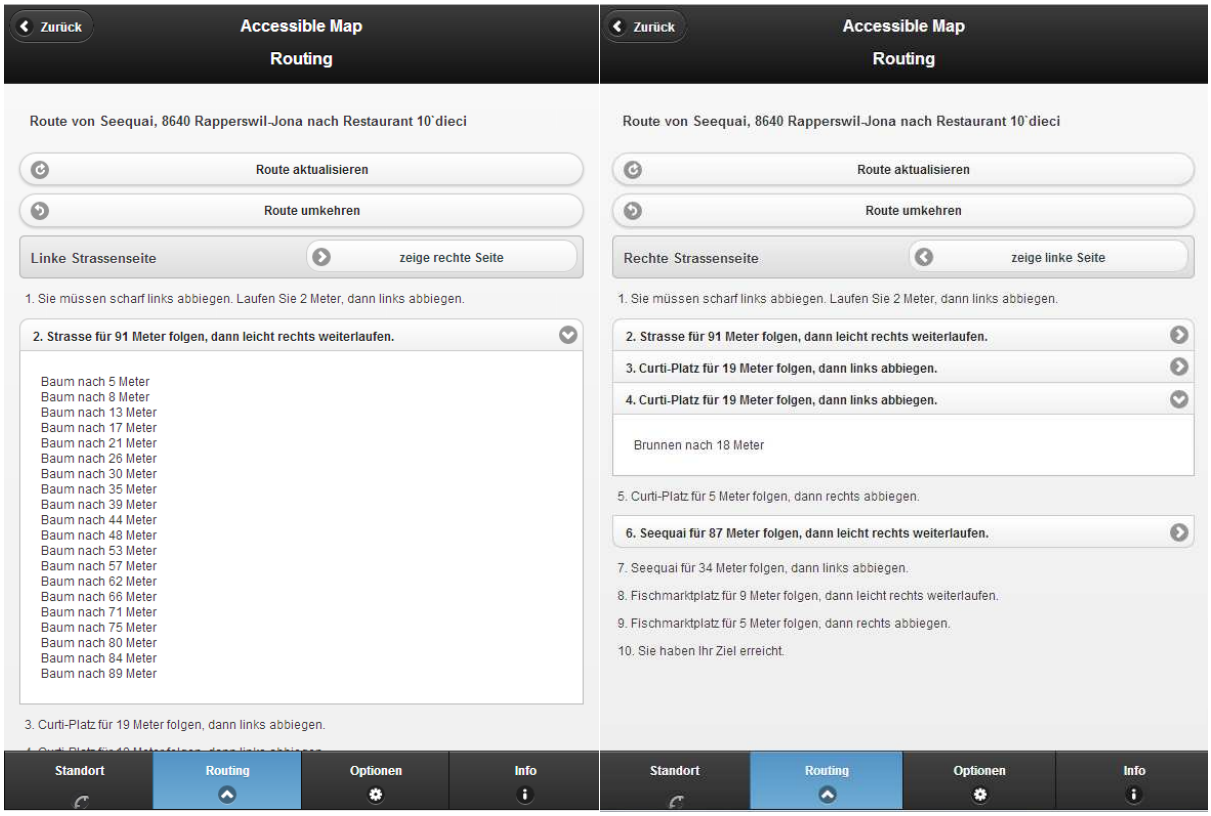


Abbildung 3 - Screenshots Routing

Die zugrundeliegenden Daten werden in der Stadt Zürich teilweise durch Open Government Data ersetzt. Dies trifft für Bäume und Abfalleimer zu. Die Baustellen wurden von der Stadt Zürich bezogen und in eine schweizweite Datenbank integriert. So muss bei Baustellen nicht unterschieden werden, ob sich der Benutzer in der Stadt Zürich befindet oder nicht. Bis zum Ende der Bachelorarbeit konnten keine weiteren Daten von der Stadt öffentlich zur Verfügung gestellt werden. Wünschenswert gewesen wären Daten zu Leitlinien, Brunnen, Fussgängerstreifen und der Hinweis ob es bei einem solchen eine Ampel mit integriertem Vibrationsmelder für Blinde hat.

Ausblick

In einer Weiterentwicklung soll die Einhaltung der Route permanent überwacht und im Falle einer Abweichung automatisch eine neue Route berechnet werden. Ebenfalls von Vorteil wäre das Implementieren einer manuellen Zieleingabe anhand einer Adresse, zusätzlich zur Navigation zu einem POI. Dieses Szenario wurde nicht berücksichtigt, da in der Arbeit die Standortausgabe erst später um das Routing erweitert wurde. Denkbar ist auch eine „Bring mich nach Hause“-Funktion, die den Benutzer vom aktuellen Standort aus an eine gespeicherte Adresse navigiert. Der Grundstein für ein barrierefreies Fussgängerrouting wurde mit dieser Arbeit jedoch gelegt.

Die Qualität der angereicherten Route hängt stark von den in OpenStreetMap vorhandenen Daten ab. Sind in einer Strasse zwar Bäume vorhanden, aber in OpenStreetMap nicht erfasst, so hilft dies dem Anwender nichts. An diesem Punkt ist es wichtig zu erwähnen, dass jede Person die Möglichkeit hat, Daten in OpenStreetMap zu erfassen. Diese sind kurz nach der Erstellung bereits verfügbar. Um die Anwendung schweizweit qualitativ hochwertiger zu machen, könnten öffentliche Daten von allen Städten in OpenStreetMap eingetragen oder auch manuell in die Anwendung eingebunden werden. Damit könnte die Qualität im städtischen Raum stark verbessert werden. Falls die Daten in OpenStreetMap eingetragen werden, muss in der Applikationslogik keine Fallunterscheidung nach Stadt angewendet werden.

Um die Daten für die betroffenen Personen zu verbessern kann zum Beispiel der Betreuer der blinden oder sehbehinderten Person die wichtigsten Orientierungspunkte auf einer bevorstehenden Route kontrollieren und gegebenenfalls fehlende Punkte eintragen.

In einer Weiterentwicklung ist es denkbar

- den Anwendern der Applikation eine Möglichkeit anzubieten, fehlende Daten selbst zu erfassen.
- eine Kommentarfunktion einzubauen, die es ermöglicht, Hinweise einzugeben, welche auch für andere Benutzer der Applikation sichtbar sind. Gibt es zum Beispiel ein Trottoir, das sehr hoch ist, so kann der Betroffene dies vermerken. Führt die Route das nächste Mal wieder über diesen Weg, wird dies erwähnt und auch allen anderen Benutzern zugänglich gemacht.

Inhaltsverzeichnis

1 Teil I : Technischer Bericht 11
1.1 Einleitung 11
1.2 Aufgabenstellung 11
1.3 Ziele 12
1.4 Rahmenbedingungen 12
1.5 Aufbau der Arbeit 13
1.6 Umsetzung 14
1.6.1 Stand der Technik 14
1.6.2 Vision 35
1.6.3 Begriffserklärungen OpenStreetMap 40
1.6.4 Algorithmen 40
1.6.5 Resultate 51
1.6.6 Schlussfolgerung 54
2 Teil II: SW-Projektdokumentation 56
2.1 Analyse 56
2.1.1 Anforderungsspezifikation 56
2.2 Design 58
2.2.1 Objektdiagramm 58
2.2.2 Interaktionsdiagramm 61
2.3 Implementation 63
2.3.1 Technologien 63
2.3.2 Testing 64
2.4 Resultate 66
2.4.1 Funktionsumfang 66
2.4.2 Weiterentwicklung 71
2.5 Projektmanagement und Projektmonitoring 73
2.5.1 Projektmanagement 73
2.5.2 Projektmonitoring 75
2.5.3 Risikomanagement 76
3 Teil III: Anhang 77
3.1 Erklärung 77
3.2 Glossar 78
3.3 Abbildungsverzeichnis 80
3.4 Literaturverzeichnis 82
3.5 Persönliche Berichte und Dank 83
3.5.1 Dank 83
3.5.2 Persönlicher Bericht Gwendoline Rothauser 83

3.5.3 Persönlicher Bericht Julia Schmucki 83
3.6 Inhaltsverzeichnis der CD 84

1 Teil I : Technischer Bericht

1.1 Einleitung

Die Stadt Zürich bietet Open Government Data an. Dabei handelt es sich um Datensätze, die für die Öffentlichkeit in digitaler Form erfasst wurden. Darin enthalten sind zum Beispiel der Stadtplan der Stadt Zürich und weitere interessante Daten wie zum Beispiel alle Bäume, alle Schulen, Restaurants und Gotteshäuser.

Viele blinde oder sehbehinderte Personen besitzen ein Smartphone. Die meisten haben ein iPhone, da es über eine gute Vorlesefunktion verfügt und es einer blinden oder sehbehinderten Person ermöglicht, das Telefon wie ein Sehender zu bedienen.

Nun ist die Frage, ob man diese öffentlichen Daten auch für blinde oder sehbehinderte Personen mittels einer Webapplikation für Smartphones zugänglich machen kann und zwar so, dass es für diese Personen einen neuen Nutzen bringt.

1.2 Aufgabenstellung

Seit Juni letzten Jahres stellt die Stadt Zürich auf dem OGD (Open Government Data) Portal öffentlich Daten frei zur Verfügung. Diese Daten können in Kombination mit weiteren Datenquellen genutzt werden um innovative Anwendungen zu erstellen. Aus diesen offenen Daten ist beispielsweise die Entsorgung Stadt Zürich App oder eine Budgetvisualisierung des Budgets der Stadt Zürich entstanden:

http://data.stadt-zuerich.ch/content/portal/de/index/ogd/anwendungen.html

Ziel der Arbeit **Accessible Map App** ist eine Anwendung für sehbehinderte und blinde Menschen, die ihnen bei der Orientierung in der Stadt Zürich hilft („sie liest ihnen die Umgebung vor“). Steht eine blinde Person beispielsweise am Paradeplatz, soll die Anwendung ihr vorlesen: „12 Uhr Bahnhofstrasse, 3 Uhr Waagstrasse, 4 Uhr Poststrasse, 8 Uhr Bleicherweg, 10 Uhr Talacker.“ Damit kann sich die sehbehinderte oder blinde Person in der Stadt Zürich besser orientieren.

Umgesetzt werden soll die Applikation mit Open Government Data der Stadt Zürich und weiterem, offen zugänglichem Kartenmaterial. Die Technologie an sich kann frei gewählt werden. Einzige Bedingung ist, dass die Anwendung auf mobilen Endgeräten genutzt werden kann. Präferiertes, mobiles Gerät von sehbehinderten und blinden Personen ist das iPhone, v.a. wegen der guten Vorlesefunktionalität.

1.3 Ziele

In der Aufgabenstellung wurden folgende Ziele definiert:

Anforderungsanalyse

Erhebung der aktuellen Probleme, die sich Blinden und Sehbehinderten bei der Orientierung und Fortbewegung im städtischen Raum stellen. Zusammenstellung der wichtigsten, funktionalen Benutzeranforderungen an eine solche Applikation (diese Arbeit erfolgt in enger Zusammenarbeit mit Technologieexperten und betroffenen Mitarbeitenden der Stiftung „Zugang für alle“).

Technische Evaluation

Abklärung der technischen Möglichkeiten, die Zielgruppe in diesem Problembereich auf der Basis von Geodaten und Sprachausgabe zu unterstützen.

Datenumfang

Festlegung des zur Verfügung stehenden Datenbestandes in der Stadt Zürich (Points of Interest, Haltestellen, öffentliche Plätze, Strassen etc.). Die Stadt unterstützt hier mit mehreren Experten das Projekt aktiv.

Implementation

Entwurf und Entwicklung der Applikation Accessible Map auf Basis der erhobenen Benutzerbedürfnisse. Die Benutzerfreundlichkeit aus Sicht der Betroffenen soll dabei besonders berücksichtigt werden. Kernpunkte sind, wie die Geodaten sinnvoll in Text umgewandelt werden können sowie die barrierefreie Darstellung derselben auf einer (mobilen) Website.

Demonstration

Demonstration eines lauffähigen Prototyps. Der Prototyp soll eine eventuelle, zukünftige Weiterentwicklung zulassen. Mögliche Szenarien wären beispielsweise eine schweizweite Anwendung, Ausbau des Funktonalitätsumfangs sowie Ausweitung des Datenbestandes.

1.4 Rahmenbedingungen

Die Rahmenbedingungen wurden wie folgt definiert:

- Kenntnisse der Programmierung von Webseiten, die auch auf mobilen Endgeräten genutzt werden können. Technologie und Framework sollen selbstständig so gewählt werden, dass sie die Anwendung auf mobilen Endgeräten optimal unterstützen.
- Gute Kenntnisse von JavaScript oder Interesse, sich dieses anzueignen.
- Code, Kommentare und Versionsverwaltung sind in Englisch. Alles andere ist in deutscher Sprache verfasst.

1.5 Aufbau der Arbeit

Die Arbeit ist in drei Teile gegliedert.

Im ersten Teil, dem technischen Bericht, werden die Aufgabenstellung und Ziele vorgetragen (Kapitel 1.1 – 1.4). Weiter werden die Umsetzungsentwürfe zu Beginn der Arbeit vorgestellt. Dieser umfasst eine Analyse der vorhandenen Applikationen im selben Themenbereich sowie der Routingalgorithmen (Kapitel 1.6.1). Danach werden die anfänglichen Konzept- und Designideen gezeigt und erklärt (Kapitel 1.6.2). Die OpenStreetMap-spezifischen Begriffe werden in Kapitel 1.6.3 erläutert. In Kapitel 1.6.4 werden die Algorithmen beschrieben, so wie sie in der Anwendung implementiert wurden. Zum Schluss werden die Resultate der Arbeit und ein Ausblick auf mögliche Weiterentwicklungen gegeben (Kapitel 1.6.5 und 1.6.6).

Der zweite Teil umfasst die Software-Dokumentationen. Darin enthalten sind die Anforderungsspezifikation (Kapitel 2.1.1) sowie ein Objekt- und Interaktionsdiagramm (Kapitel 2.2.1 und 2.2.2). Ausserdem wird in Kapitel 2.3 die Implementation der einzelnen Funktionalitäten näher erläutert. Anschliessend werden die Resultate als einzelne Punkte vorgetragen (Kapitel 2.4). Das Projektmanagement befindet sich in Kapitel 2.5. Dort wird die Zeitplanung, die Arbeitsaufteilung, das Projektmonitoring und das Risikomanagement vorgestellt.

Den letzten Teil der Arbeit bildet der Anhang. Darin befinden sich neben Glossar, Abbildungs- und Literaturverzeichnis auch die persönlichen Berichte der Studentinnen.

1.6 Umsetzung

1.6.1 Stand der Technik

In der Anforderungsanalyse wurden bestehende Anwendungen genauer analysiert. Darunter waren drei Web-Applikationen, sechs iPhone-Anwendungen, eine Android-Anwendungen und eine Anwendung, die sowohl für iOS als auch für Android verfügbar ist.

1.6.1.1 Web-Anwendungen

Neben der Anwendung AmauroMap [2] für Blinde wurden noch zwei weitere Web-Anwendungen analysiert. Diese sind zwar für Rollstuhlfahrer entworfen, zeigen jedoch, dass viel Engagement für Projekte, welche behinderte Personen durch Internettechnologien unterstützen wollen, vorhanden ist. Vielfach sind die Entwickler solcher Webseiten selbst von einer Behinderung betroffen und kennen die Bedürfnisse der Anwender.

Web-Applikation AmauroMap

Dieses Projekt der Internet-Privatstiftung Austria IPA war der Auslöser für die Ausschreibung dieser Bachelorarbeit. AmauroMap ist eine interaktive Onlinekarte, die Blinden hilft, sich ihr Umfeld räumlich vorstellen zu können. Um dies zu erreichen, werden digitale Stadtplandaten so aufbereitet, dass diese im Kopf der Blinden zu kognitiven Karten kombiniert werden (siehe Abbildung 4). Dazu haben sich die Entwickler darüber informiert, wie räumliche Beschreibungen aufgebaut werden müssen, dass diese für die Zielgruppe brauchbar und verständlich sind. Es wird dabei darauf geachtet, dass möglichst nur Open-Source-Software und freie Datenbestände verwendet werden.

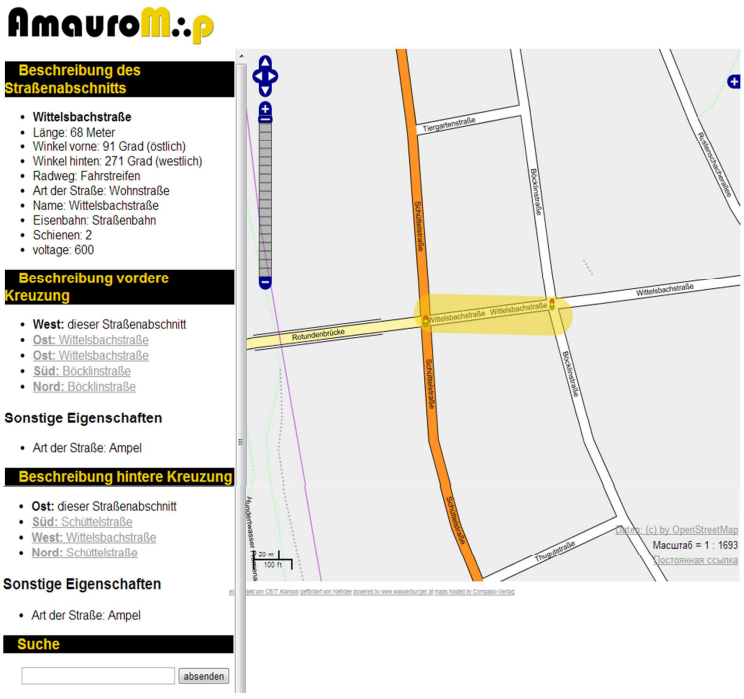


Abbildung 4 - Screenshot AmauroMap aus Google-Cache

Leider kann die Webseite aus unbekannten Gründen seit Dezember nicht mehr aufgerufen werden. Die Anwendung war über den Link amauro.map.at oder accessible.map.at erreichbar.

Accessibility Guide

Um Menschen im Rollstuhl das Leben zu erleichtern wurde die Internet-Plattform www.accessibility-guide.org geschaffen. Sie erlaubt es, Orte und Lokalitäten mit Informationen zur Rollstuhlzugänglichkeit anzureichern und diese einzusehen. Durch Anpassen der Einstellungen auf eigene Bedürfnisse werden die Angaben zu den Orten als Ampelfarben dargestellt. Grün weist dabei auf eine rollstuhlfreundliche Lokalität hin und rot auf eine nicht rollstuhlgängige.

Die Anwendung hat bisher nicht viele Einträge und ist auf die Mithilfe und Spenden von Freiwilligen angewiesen.

Auf Abbildung 5 ist ein Eintrag inklusive Kartenausschnitt der Anwendung zu sehen. Bisher wurden nicht viele Eigenschaften bewertet. Es handelt sich hier um einen nicht kategorisierten Ort mit Namen Frau Gerolds Garten.

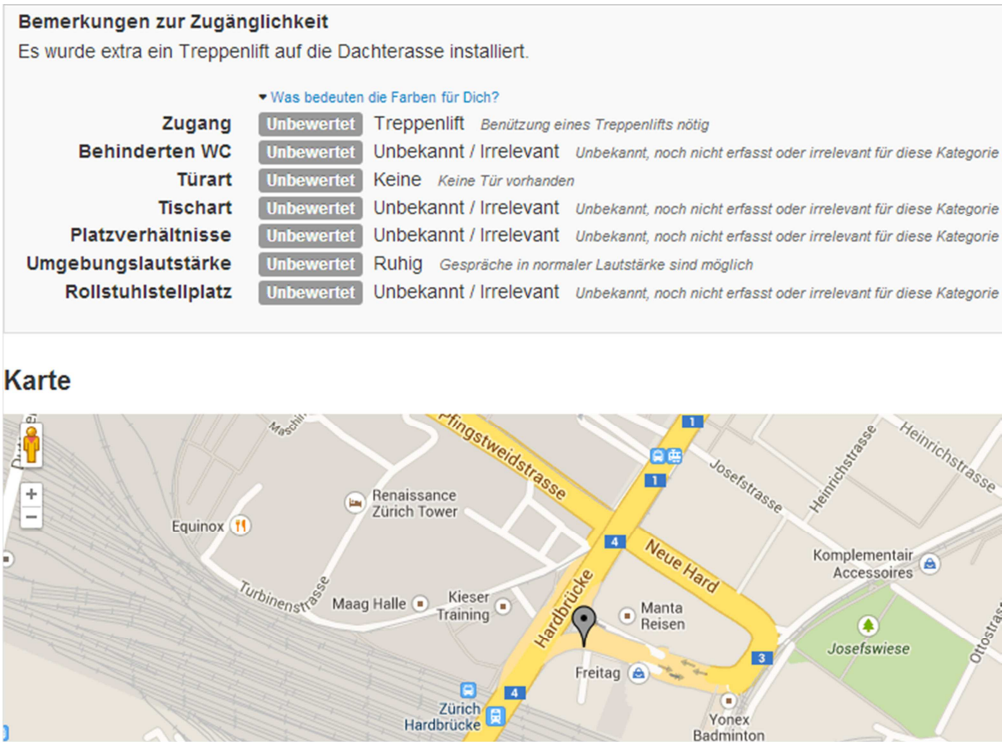


Abbildung 5 - Screenshot Accessibility Guide.

Wheelmap

Auch diese Anwendung ist für Rollstuhlfahrer gedacht, aber weit besser ausgebaut. Informationen können eingesehen und vervollständigt werden. Unterschieden werden dabei die Kategorien voll, teilweise und nicht rollstuhlgerecht, auch hier in den 3 Ampelfarben. Unkategorisierte Einträge werden in Grau angezeigt. Es kann nach verschiedenen Farben- und Themenkategorien gesucht werden. Das verwendete Kartenmaterial stammt von OpenStreetMap.

Auf Abbildung 6 ist ein Kartenausschnitt aus Rapperswil mit vielen Einträgen zu sehen. Viele von ihnen sind noch nicht bewertet worden.

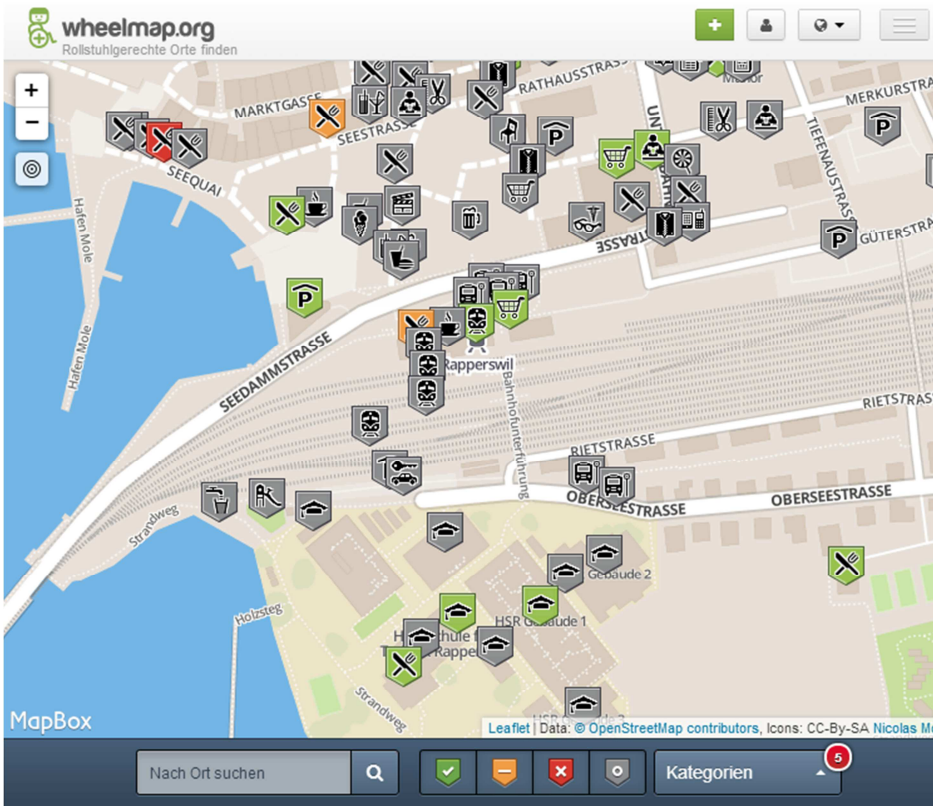


Abbildung 6 - Ausschnitt Wheelmap in Rapperswil

Die Applikation kann über www.wheelmap.org aufgerufen werden.

1.6.1.2 iOS Anwendungen

Mit der integrierten Vorlesefunktion „VoiceOver“ wird die Bedienung des iPhones für Blinde und Sehbehinderte zugänglich gemacht. Um beispielsweise eine Seite auf ihren Inhalt zu überprüfen muss über den Bildschirm gewischt werden. Dabei wird jedes Element fokussiert und vorgelesen. Um einen Knopf zu betätigen muss doppelt statt einfach getippt werden.

My Position **Preis: 1.00 CHF**

Mit dieser App erfährt man die eigene Position in Längen- und Breitengrad mit Höhenangabe und Bewegungsgeschwindigkeit, jedoch ist keine Adressangabe vorhanden (siehe Abbildung 7 links). Mit der Taste „Map“ wird man auf GoogleMaps in einem neuen Browserfenster umgeleitet.

Nach der Eingabe der Zielstadt im Suchbildschirm (siehe Abbildung 7 rechts) wird die Distanz und Richtung berechnet, die Angaben sind jedoch nur in Yards und Feet angegeben. Ein Routing wird nicht angeboten. Die Anwendung ist nicht zum Positionstracking ausgelegt und arbeitet komplett offline.

Es werden keine POIs angezeigt und bei der Zielsuche auch nicht gefunden. Diese App ist zudem nur in Englisch verfügbar.

Abbildung 7 - Screenshots My Position.

Sendero GPS LookAround **Preis: Kostenlos**

Diese Anwendung gibt dem Benutzer die nächste Kreuzung, die nächsten fünf Points of Interest und einen ungefähren Standort als Adresse direkt auf dem Startbildschirm aus (siehe Abbildung 8 links). Ausserdem wird die Ausrichtung des Smartphones als Himmelsrichtung angegeben. Diese kann mit der mitgelieferten Kompassfunktion aktualisiert werden.

Es können nur POIs von einer Kategorie ausgewählt werden (siehe Abbildung 8 rechts). Diese werden dann mit Distanz- und Richtungsangabe angezeigt, wobei eine Maximaldistanz von 2 Kilometern festgelegt ist. Es wird keine Routingfunktion angeboten. Zur Aktualisierung kann das Gerät geschüttelt werden. Auch diese App ist nur in Englisch verfügbar. Entwickelt wurde sie von Sendero Group[1].

Abbildung 8 - Screenshot Sendero GPS LookAround

[1] Sendero Group - http://senderogroup.com/index.htm

Around Me **Preis: kostenlos**

Aus einer Liste von POI-Kategorien (siehe Abbildung 9 links) kann eine bestimmte gewählt werden, die dann bis zu 20 POIs dieser Art in 10 Kilometer Entfernung zum aktuellen oder manuell eingegebenen Standort anzeigt. Dabei ist sowohl die Entfernung in Meter, sowie die Richtung als Kompass visualisiert angegeben (siehe Abbildung 9 Mitte). Letzterer kann leider nicht von „VoiceOver“ vorgelesen werden. Neben den Kategorien kann ein POI auch über die Suchleiste gefunden werden. Dabei können Adressen oder Attribute wie zum Beispiel „italienisch“ für die Suche nach einem Restaurant mit italienischer Küche angegeben werden.

Wird ein POI ausgewählt, kann er als Favorit gespeichert und/oder als Navigationsziel eingetragen werden (siehe Abbildung 9 rechts). Für das Routing wird die Karten-App des iPhones genutzt.

Ein spezielles Feature der App ist die Integrierung von FourSquare-Tipps und Kontaktangaben zu den POIs. Mit der Funktion „Umkreissuche“ werden Informationen zur Umgebung angegeben. Weiter gibt es die Realitätsansicht, in der mit der Kamera gearbeitet wird. Das Gerät wird in eine Richtung gehalten und die POIs, die in jener Richtung liegen, auf einem Radar angezeigt. Ausserdem erscheinen sie im Kamerabild. Dieses Feature kann von Blinden leider nicht verwendet werden, da es nur bildlich dargestellt wird. Die Anwendung funktioniert auch offline.

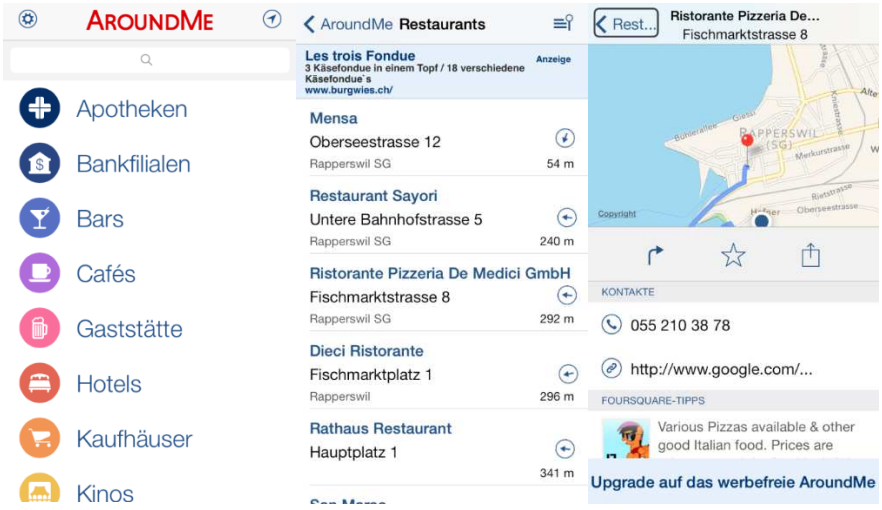


Abbildung 9 - Screenshots AroundMe

Ariadne GPS **Preis: 6.00 CHF**

Die Adressangabe dieser Anwendung[2] fällt sehr detailliert aus, kann aber in den Einstellungen verkürzt werden (siehe Abbildung 10 rechts). Sie erscheint in der Mitte des Bildschirms (siehe Abbildung 10 links). Der Standort kann manuell oder automatisch in regelmässigen Abständen aktualisiert werden. Neben der Bewegungsrichtung wird auch die Ausrichtung des Geräts in Grad angegeben. Beide Werte sind stets aktuell. In der unteren Hälfte des Bildschirms werden verschiedene Icons angezeigt, die für „VoiceOver“ einen Beschreibungstext und den dazugehörenden Wert anbieten. So wird beispielsweise angegeben, wie gut die Wifi-Verbindung oder wie stark das GPS-Signal ist.

Es können nur Adressen als Favoriten gespeichert werden. Diese können als aktiv eingestellt werden. Befindet sich ein Favorit in der Nähe (die maximale Entfernung ist einstellbar), so wird die Richtung und Entfernung angegeben und beim Weiterlaufen stetig aktualisiert. Bei bestehender Internetverbindung können die Strassennamen und Hausnummern aus OpenStreetMap fortlaufend ausgegeben werden.

Ein interessantes Feature dieser Anwendung ist die sprechende Karte. Durch Berührung einer angezeigten Karte werden die Strassennamen und Hausnummern vorgelesen, wobei sich der aktuelle Standort anfangs in der Mitte der Karte befindet. Durch das zur Seite Wischen mit zwei Fingern kann der Ausschnitt verschoben werden. Durch Hoch- und Runterschieben kann gezoomt werden. Auch entfernte Gebiete können erkundet werden, indem Strassen- bzw. Ortsnamen angegeben werden. Ausserdem werden Gewässer und Strassen mit entsprechenden Geräuschen hinterlegt.

Leider werden für einige Zusatzfunktionen Kosten erhoben. Darunter fallen personalisierte Töne für Favoriten oder das Wiederholen von Meldungen.

Abbildung 10 - Screenshots von Ariadne GPS

[2] Ariadne GPS - http://www.ariadnegps.eu/

BlindSquare **Preis: 24.00 CHF**

Diese Anwendung von Blinden für Blinde[3] bietet eine eigene Sprachausgabe an, die sich leider teilweise mit der Stimme von „VoiceOver“ überlagert. Für die Anwendung werden Daten von FourSquare und OpenStreetMap in Echtzeit abgerufen, weshalb sie nicht offline genutzt werden kann. Durch Schütteln des Geräts kann eine Aktualisierung erfolgen.

Im Aktionsmenü (Schraubenschlüssel in Abbildung 11 links) können verschiedene Funktionen ausgewählt werden. Wird „Umsehen“ gewählt, werden POIs und Strassen in der Umgebung angegeben. Mit „Ihr Standort“ erhält man die Adressangabe zur aktuellen Position, die mit der Funktion „nächste Kreuzungen“ vervollständigt wird. Die Genauigkeit der GPS-Info kann abgefragt werden.

Die POIs sind in Kategorien unterteilt, die einzeln ein- oder ausgeschaltet werden können. Es kann allerdings auch über die Suchleiste nach bestimmten POIs gesucht werden. Der Suchradius kann zwischen 25 Meter und 2 Kilometer eingestellt werden. Es werden Entfernung und Richtung zu den POIs angegeben (siehe Abbildung 11 Mitte). Die Richtungsangabe kann wahlweise als Uhrzeit, Grad oder proportional (links und rechts) ausgegeben werden. Es können eigene Orte als Favoriten gespeichert werden (siehe Abbildung 11 rechts). Befindet sich ein solcher Ort in eingestellter Entfernung, erfolgt ein Benachrichtigungston. Möchte an einen bestimmten Ort navigiert werden, wird die KartenApp des iPhones geöffnet, wobei BlindSquare im Hintergrund weiterläuft und Kreuzungen ansagen kann. Neben der Navigation besteht die Möglichkeit, einen POI nur zu verfolgen. Dadurch werden Entfernung und Richtung zum Punkt in regelmässigen Abständen angesagt.

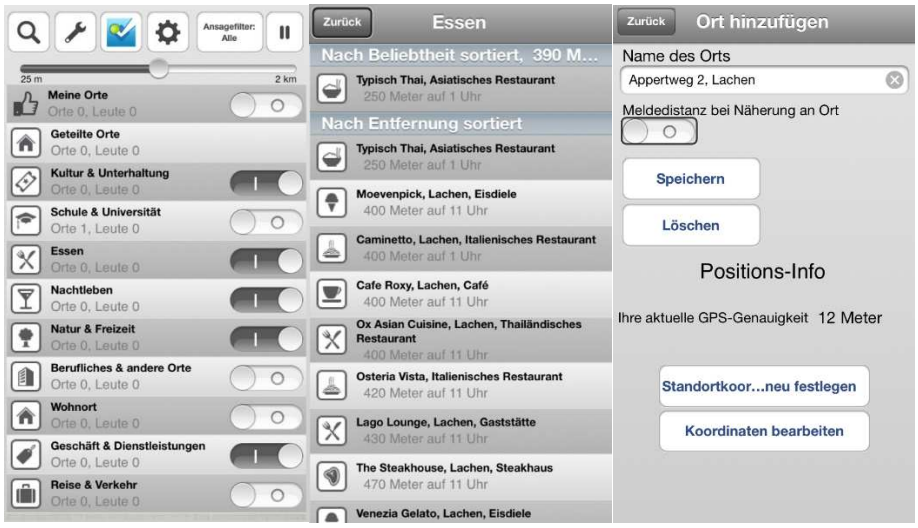


Abbildung 11 - Screenshots von BlindSquare

[3] BlindSquare - http://blindsquare.com/

Karten

Die in iOS eingebaute Karten-App[4] verfügt bereits über ein Fussgängerrouting. Die Standortausgabe erfolgt nur als Punkt auf der Karte (siehe Abbildung 12 links).

Es besteht die Möglichkeit, nach POIs oder Adressen zu suchen und sich dorthin navigieren zu lassen. Wird dabei die Route verlassen, wird eine Warnung ausgegeben. Die ganze Route kann auch als Text angezeigt werden (siehe Abbildung 12 rechts), um sie virtuell zu erkunden.

Leider wird der Beginn der Route nur als Himmelsrichtung angegeben, so dass man eine Kompassfunktion zur Orientierung benötigt. Ausserdem werden Querstrassen nicht angesagt. In der Karte sind auch keine oder nur wenige Fusswege eingezeichnet, wodurch man einen längeren Weg in Kauf nehmen muss (siehe Abbildung 12 Mitte).

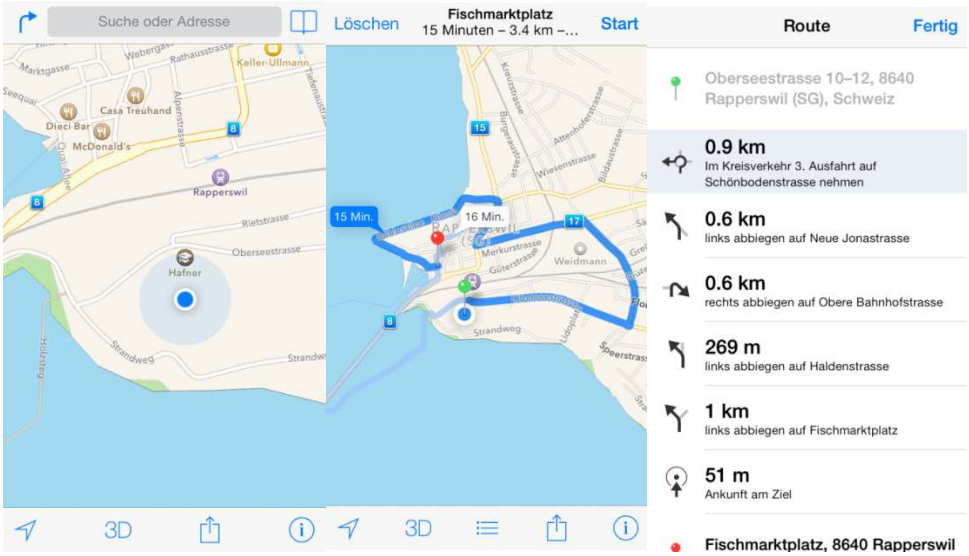


Abbildung 12 - Screenshots von Karten-Applikation

[4] Karten - http://www.apple.com/de/ios/maps/

1.6.1.3 Android Anwendungen

In Android wird der Bildschirm mit Hilfe des Screen Readers „TalkBack“ vorgelesen. Die Bedienung ist praktisch gleich wie bei „VoiceOver“ in iOS. Im Gegensatz zu iOS gibt es für Android-Geräte nur sehr wenige ausgereifte Anwendungen für sehbehinderte und blinde Personen.

Intersection Explorer **Preis: Kostenlos**

Die nächste Kreuzung zum aktuellen Standort wird ermittelt und auf einer Karte angezeigt (siehe Abbildung 13). Mit dem Finger kann anschliessend die Gegend erforscht werden. Durch Berührung einer Strasse wird deren Name zusammen mit einer Himmelsrichtung relativ zur Kreuzung ausgegeben. Will man die nächste Kreuzung sehen, wischt man mit dem Finger in die entsprechende Richtung.

Es ist weder eine POI-Suche noch eine Routingfunktion vorhanden. Negativ aufgefallen ist, dass die Strassennamen nicht immer richtig angegeben werden und Fusswege gar nicht eingezeichnet sind. Die App ist zudem im Test mehrmals abgestürzt. Die Karte zeigt immer nach Norden und kann nicht gedreht werden. Die Applikation ist nur auf Englisch verfügbar.

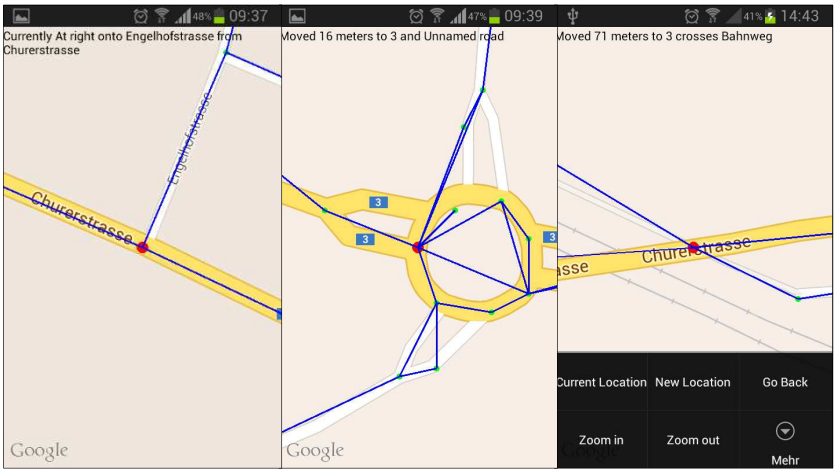


Abbildung 13 - Screenshots Intersection Explorer.

1.6.1.4 Anwendungen für iOS und Android

Guide4Blind **Preis: Kostenlos**

Diese App[5] ist für blinde wie auch sehende Touristen in Soest geschrieben. Sie funktioniert deshalb auch nur in Soest. Eine Standortausgabe ist nicht vorhanden. Vordefinierte Routen wurden abgefilmt und führen dadurch über Fussgängerstreifen und Gehwege bis vor die Haustür. Gefahrenstellen wie Treppen, aber auch vordefinierte POIs, an denen man vorbeiläuft werden von der Anwendung angesagt. Die Navigation kann per Sprache, Vibration oder Geiger (Klickgeräusch) erfolgen.

Durch ein Zusatzgerät (GPS-Verstärker), welches mit dem Smartphone verbunden wird, kann die Genauigkeit der Navigation auf wenige Zentimeter erhöht werden.

[5] Guide4Blind - http://www.guide4blind.de/

1.6.1.5 Bewertung der analysierten Anwendungen

In Tabelle 1 werden die Funktionen verglichen, die eine Anwendung wie sie in der Aufgabenstellung verlangt wird, bieten sollte.

Einerseits muss die Applikation eine Standortausgabe in Form einer Adresse bieten können (siehe Spalte 1). Andererseits sollte sie in der Lage sein, POIs in der näheren Umgebung auszugeben (siehe Spalte 2). Dabei ist es nützlich, POIs aus mehreren Kategorien angezeigt zu bekommen, damit man sich ein Bild der Umgebung machen kann (siehe Spalte 3). Von Vorteil wäre eine Favoritenspeicherung, mit der man einen oft besuchten Ort schnell wiederfinden kann (siehe Spalte 4).

In der Spalte 5 wird verglichen, ob ein Routing angeboten wird. Dies muss nicht zwingend von der gleichen Anwendung ausgeführt werden. Es reicht, wenn eine Navigation überhaupt angeboten wird.

Dass die Anwendung auf Deutsch verfügbar sein soll (Spalte 6), ist vor allem für die Vorlesefunktion wichtig. Ist diese auf Deutsch eingestellt, so werden fremdsprachige Sätze mit deutscher Betonung vorgelesen. Diese sind dann akustisch unverständlich.

App	Standort-ausgabe als Adresse	POIs	Mehrere POI-Kategorien	Favoriten speichern	Routing	In Deutsch verfügbar
My Position	x	x	x	x	x	x
Sendero GPS Lookaround	✓	✓	x	x	x	x
Around me	x	✓	✓	✓	✓	✓
Ariadne GPS	✓	x	x	✓	✓	✓
BlindSquare	✓	✓	✓	✓	✓	✓
Karten	x	x	x	x	✓	✓
Intersection Explorer	x	x	x	x	x	x

Tabelle 1 - Vergleich der untersuchten iPhone Applikationen

Fazit

Die einzige Anwendung, die alle geforderten Funktionen bieten kann, ist „BlindSquare“. Sie bietet sowohl eine detaillierte Auskunft über den momentanen Standort und POIs in der näheren Umgebung, als auch die Möglichkeit, sich zu diesen hin navigieren zu lassen. Darüber hinaus können Favoriten gespeichert und später wiedergegeben werden. Der Preis für die Applikation ist damit auch gerechtfertigt.

Den gestellten Anforderungen gar nicht gerecht werden „My Position“ und „Intersection Explorer“. Sie besitzen weder Routing, noch geben Sie den Standort als Adresse an. Zudem fehlen auch Funktionen zur Ausgabe von POIs.

1.6.1.6 Fussgänger-Routing

Rahmenbedingungen

Folgende Dienste wurden zum Zeitpunkt der Entscheidung der Entwicklung eines Fussgängerrouting-Dienstes analysiert:

- Google Directions API
- OSMR
- OSRM (Open Source Routing Machine)
- Bing
- MapQuest
- YOURS

Zur Analyse des aktuellen Standes wurde eine Route in Rapperswil SG genommen, welche die Vollständigkeit der Fussgängernavigation bei einem Bahnhof mit Unterführung prüft.

Startkoordinate: 47.223842,8.817279 – Hochschule Für Technik Rapperswil

Zielkoordinate: 47.225474,8.815717 – Fischmarktplatz Rapperswil

Richtig wäre folgende Route:

1. Nach Nordosten Richtung Rietstrasse
2. Durch die Fussgängerunterführung zum Bahnhofplatz gelangen
3. Auf dem Bahnhofplatz links zur Fussgängerunterführung gelangen
4. Die Fussgängerunterführung durchqueren und nach links verlassen
5. Rechts Abbiegen
6. Ziel erreicht

Im nächsten Abschnitt werden nun die einzelnen Dienste und deren Fussgängernavigation für diese Route dokumentiert.

Google Directions API

Dies ist der Routing-Dienst von Google, welcher auch für Routingabfragen in GoogleMaps verwendet wird.

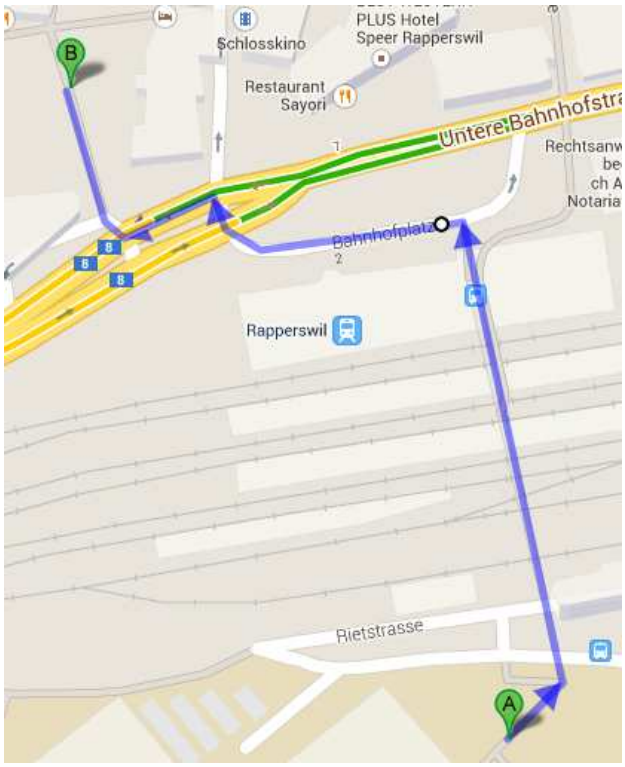
Routenbeschreibung	Abbildung
1. Nach Nordosten Richtung Rietstrasse 22 m 2. Bei Rietstrasse links abbiegen 130 m 3. Links abbiegen auf Bahnhofplatz 78 m 4. Links abbiegen auf Untere Bahnhofstrasse/Route 8 27 m 5. Rechts abbiegen 44 m 6. Ziel erreicht.	

Abbildung 14 - Routenansicht Google Maps

Fazit

Die Fussgängerunterführungen zwischen Schritt 1 und 2 sowie zwischen 3 und 4 werden mit keinem Wort erwähnt. Obwohl diese in der Datenbank so hinterlegt sind. Für eine blinde oder sehbehinderte Person ist diese Navigation nicht zu gebrauchen.

OSMR

Dies ist der Dienst Open Route Service [9] der Universität Heidelberg. Er basiert auf Open Street Map.

Routenbeschreibung	Abbildung
1. Start (Northeast) auf no name 0.0 km (0.0 km) 2. Gehe links 0.1 km (0.0 km) 3. Gehe links 0.1 km (0.1 km) 4. Gehe rechts 0.0 km (0.2 km) 5. Gehe links 0.0 km (0.2 km) 6. Gehe rechts auf Fischmarktplatz 0.1 km (0.2 km) 7. Gehe links auf Fischmarktplatz 0.0 km (0.3 km) 8. Gehe links - Ziel erreicht! 0.0 km (0.3 km)	 **Abbildung 15 - Routenansicht OSMR**

Fazit

Die Route geht nicht immer den rot gestrichelt eingezeichneten Fussgängerpfaden entlang. Die Fussgängerunterführungen sind zwar eingezeichnet, werden jedoch nicht erwähnt. Man könnte meinen, man müsse über die Gleise und über die stark befahrene Strasse laufen, die an diesem Ort keine Fussgängerstreifen hat. Auch auffällig ist der letzte Abschnitt. Der Fischmarktplatz ist ein autofreier Platz. Es ist nicht nötig, immer den Häusern entlang einen Bogen bis zum Ziel zu machen.

OSRM

Dies ist der Dienst, welcher unter routing.osm.ch zur Verfügung steht. Der Dienst basiert auf Open Street Map Daten. Er steht nur in der Schweiz zur Verfügung.

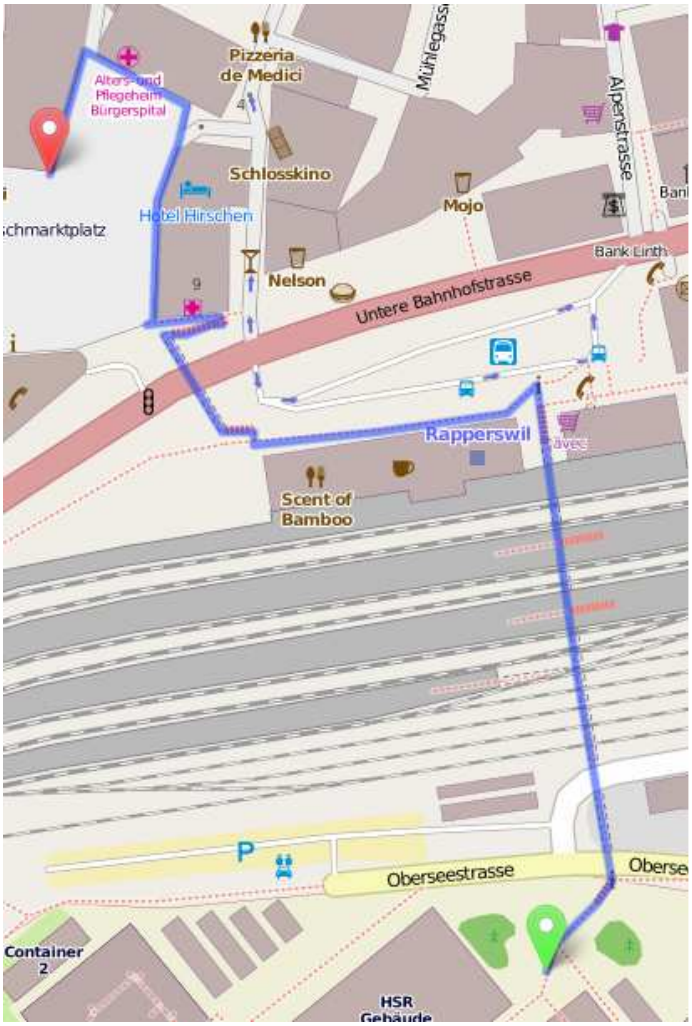
Routenbeschreibung	Abbildung
1. Fahren Sie Richtung Norden 7 m 2. Leicht rechts abbiegen 20 m 3. Leicht links abbiegen 50 m 4. Geradeaus weiterfahren 14 m 5. Geradeaus weiterfahren 16 m 6. Geradeaus weiterfahren 24 m 7. Geradeaus weiterfahren 8 m 8. Geradeaus weiterfahren 5m 9. Scharf links abbiegen 71 m 10. Rechts abbiegen 55 m 11. Links abbiegen 17 m 12. Scharf rechts abbiegen auf Fischmarktplatz 0.12 km 13. Sie haben Ihr Ziel erreicht	

Abbildung 16 - Routenansicht von OSRM

Fazit

Die Fussgängerunterführungen werden sichtbar verwendet, jedoch textuell nicht erwähnt. Auch hier wird wie bei OSMR ein Bogen den Häusern entlang dem autofreien Fischmarktplatz gemacht.

Bing

Dies ist der Routing-Dienst von Microsoft. Das Kartenmaterial beinhaltet keine Fussgängerunterführungen [10].

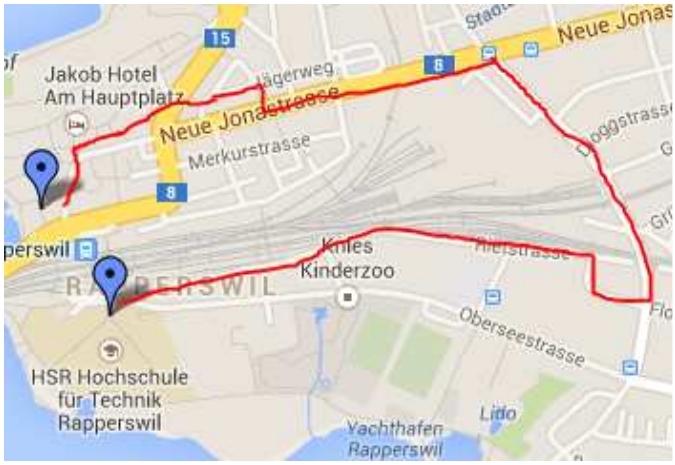
Routenbeschreibung	Abbildung
1. Strasse Richtung Rietstraße verlassen 21 m 2. Rechts abbiegen in die Rietstraße 0.8 km 3. Rechts abbiegen um auf der Rietstraße zu bleiben 0.2 km 4. Links abbiegen in die Schönbodenstrasse 0.5 km 5. Links abbiegen auf 8 / Neue Jonastrasse 0.6 km 6. Geradeaus laufen bis Cityplatz / Rathausstraße 0.1 km 7. Links abbiegen bei Fischmarktstraße / Hauptplatz 67 m 8. Rechts auf den Fischmarktplatz laufen 41 m 9. Links abbiegen um auf dem Fischmarktplatz zu bleiben 15 m 10. Ziel erreicht	 **Abbildung 17 - Routenansicht von Bing**

Fazit

Diese Route ist wohl eher für einen Autofahrer gedacht, da sie keine einzige Fussgängerunterführung nutzt. Sie dauert daher viel länger als bei vergleichbaren Diensten. Eine blinde oder sehbehinderte Person würde aber eher ans Ziel kommen als bei den Diensten welche Strassen an irgendwelchen Orten überqueren, denn die verwendeten Strassen haben alle Trottoirs.

MapQuest API

Dies ist der API-Zugriff, welcher hinter Mapquest.com [11] liegt. Für die textuelle Route kann eine JSON-Abfrage verwendet werden.

Routenbeschreibung	Abbildung
1. Fahren Sie zunächst auf die Oberseestrasse nach Osten (Teilweise nicht asphaltiert). (0.01 km) 2. Abbiegen nach links. (0.02 km) 3. Abbiegen nach rechts auf die Rietstrasse. (0.83 km) 4. Abbiegen nach rechts und bleiben Sie auf die Rietstrasse. (0.15 km) 5. Abbiegen nach links auf die Schönbodenstrasse. (0.53 km) 6. Abbiegen nach links auf die Neue Jonastrasse/8. (0.39 km) 7. Abbiegen nach rechts auf die Falkenstrasse. (0.04 km) 8. Abbiegen nach links auf die Klaus Gebert Strasse. (0.09 km) 9. Bleiben Sie geradeaus und gehen Sie auf den Fussweg. Gehen Sie weiter nach Westen (0.09 km) 10. Abbiegen nach leicht links auf die Webergasse. (0.14 km) 11. Abbiegen nach links auf den Hauptplatz. (0.03 km) 12. Bleiben Sie geradeaus und fahren Sie auf die Fischmarktstrasse. (0.11 km) 13. Ziel erreicht	 **Abbildung 18 – Von Hand erstellte Routenansicht für Mapquest API**

Fazit

Es wird zwar auch keine Fussgängerunterführung verwendet, jedoch führt ein Teil der Route durch die Fussgängerzone in der Altstadt. Auch diese Route ist viel zu lang.

YOURS

Der Routing Dienst YOURS (Yet another OpenStreetMap Route Service) [12] ist unter der BSD Lizenz verfügbar. Es gibt eine laufende Version, welche für Anwendungen verwendet werden kann und von OpenStreetMap sowie von Spendern finanziert wird.

Routenbeschreibung	Abbildung
1. Gehen Sie gerade aus. Folgen Sie der Strasse für 0.0 Meilen. 2. Bleiben Sie leicht links im Tunnel. Folgen Sie der Strasse für 0.1 Meilen. 3. Biegen Sie scharf links ab. Folgen Sie der Strasse für 0.0 Meilen. 4. Bleiben Sie leicht rechts. Folgen Sie der Strasse für 0.1 Meilen. 5. Biegen Sie rechts ab. Folgen Sie der Strasse für 0.0 Meilen. 6. Gehen Sie weiter zum Strandweg. Folgen Sie der Strasse für 0.0 Meilen. 7. Gehen Sie weiter zum Fischmarktplatz. Folgen Sie der Strasse für 0.1 Meilen. 8. Ziel erreicht.	 **Abbildung 19 - Routenansicht von YOURS**

Fazit

Wenn man die schnellste Route wählt, wird ein gültiger Fussgängerübergang der Unteren Bahnhofstrasse gewählt. Wählt man jedoch die kürzeste Route, so wird eine Strassenüberquerung bei der mit X markierten Stelle gewählt. Die Strassenüberquerungen werden auch nicht angesagt. Auf der Webseite werden zudem nur Angaben in Meilen ausgegeben. Klarer Vorteil gegenüber allen anderen getesteten Diensten ist jedoch, dass hier die Fussgängerunterführung beim Bahnhof als Tunnel erkannt wird.

Weitere Dienste

Nicht genauer erläutert wird CloudMade [13], da das Routing-Ergebnis ungefähr das gleiche war wie bei OSRM. Die Unterführung wurde ebenfalls nicht textuell erwähnt.

Bewertung der Routingresultate

Für blinde und sehbehinderte Personen ist es wichtig, dass ganz klar beschrieben ist, was für ein Typ von Strasse oder Weg der nächste Routing-Schritt beschreibt. Diese Information holt sich eine sehende Person aus den Kartenbildern. Folgt wie in der Beispielroute z.B. zuerst eine Treppe in eine Unterführung, so muss dies angegeben werden. Eine Ansage wie: „geradeaus bis Bahnhofplatz“ würde auslösen, dass blinde oder sehbehinderte Personen bis zum Zaun vor den Gleisen laufen würden, dann aber nicht mehr weiter kämen.

Man kann die Resultate aus zwei Perspektiven bewerten, einerseits anhand der Dauer/Länge der Route und andererseits anhand der Machbarkeit der Route für blinde Fussgänger.

In der Spalte „Unterführung verwendet/angegeben“ wird evaluiert, ob eine vorhandene Unterführung in der Route verwendet wird, also ob Fusswege falls vorhanden auch wirklich genutzt werden. Ein Resultat „Ja/Nein“ bedeutet, dass die Unterführung zwar verwendet, jedoch in der textuellen Ausgabe nicht erwähnt wurde. Mit „Strassenüberquerungen vorhanden/angegeben“ ist gemeint ob eine Strassenüberquerung auf der Route liegt und falls ja, ob diese textuell erwähnt wird. Bei allen Diensten wurden diese nicht textuell erwähnt sondern einfach als „rechts abbiegen“ angegeben. Dies wäre für blinde Personen sehr wichtig, denn sie sehen nicht, ob es an diesem Punkt einen Fussgängerstreifen hat oder nicht.

Dienst	Länge in km	Unterführungen verwendet/ angegeben	Strassenüberquerungen vorhanden/ angegeben	Abdeckung
Google Directions	0.3 km	Ja/Nein	Nein/-	Global
OSMR	0.3 km	Ja/Nein	Nein/-	Europa
OSRM	0.41 km	Ja/Nein	Nein/-	Schweiz
Bing	2.5 km	Nein/-	Nein/-	Global
MapQuest API	2.43 km	Nein/-	Ja/Nein	Global
YOURS	0.3 Meilen (0.48 km)	Ja/Ja	Ja/Nein bei Wahl von schnellster Route Nein/Nein bei Wahl kürzester Route	Global
Cloudmade	0.2 km	Ja/Nein	Nein/-	Global

Tabelle 2 - Bewertung Routingdienste Route

Fazit

Obwohl YOURS im Test besser abschneidet, wurde OSRM für die entwickelte Anwendung ausgewählt. Dies liegt an der Aktualisierungsrate des für den Routingalgorithmus verwendeten Kartenmaterials. OSRM wird im Gegensatz zu YOURS täglich aktualisiert. Würde also ein neuer Fussweg eingezeichnet, so könnte er am nächsten Tag bereits in einer Route verwendet werden. Die Anwendung soll jedoch so programmiert werden, dass ein Austauschen des Routingdienstes kein Problem darstellt. Es werden zur Generierung einer Route lediglich die Koordinaten der einzelnen Wegpunkte benötigt.

1.6.2 Vision

1.6.2.1 Konzeptideen

Zu Beginn der Elaborationsphase wurden drei verschiedene Ansätze zur Aufgabe der Standortausgabe erstellt. Da zu diesem Zeitpunkt noch nicht klar war, dass auch Routing implementiert werden soll, beinhalten die Entwürfe auch keine solchen Funktionen. Es ging dabei allein um die Ausgabe von POIs und Kreuzungen in der näheren Umgebung.

Idee 1

Wenn der Benutzer den Knopf „Standort ausgeben“ betätigt (siehe Abbildung 20 links), erhält er eine Beschreibung seines Standorts: „Sie laufen auf dem Limmatquai in Richtung Norden, die nächste Abzweigung rechts führt in die Schmidgasse, die nächste Abzweigung links führt in den Mühlesteg.“

Der Benutzer kann dann im Menu gewisse POIs (z.B. Restaurants) anwählen (siehe Abbildung 20 rechts). Diese werden ihm dann auch vorgelesen. Zum Beispiel: „Auf der rechten Strassenseite befindet sich das Restaurant zum Rotkreuz.“

Die Anwendung soll wie die Anwendung AmauroMap aus Österreich sein, ohne jedoch das Kartenmaterial herunterzuladen. Die Ausgabe des aktuellen Standpunkts erfolgt per manuellen Aufruf, also nicht ständig während sich der Benutzer fortbewegt.

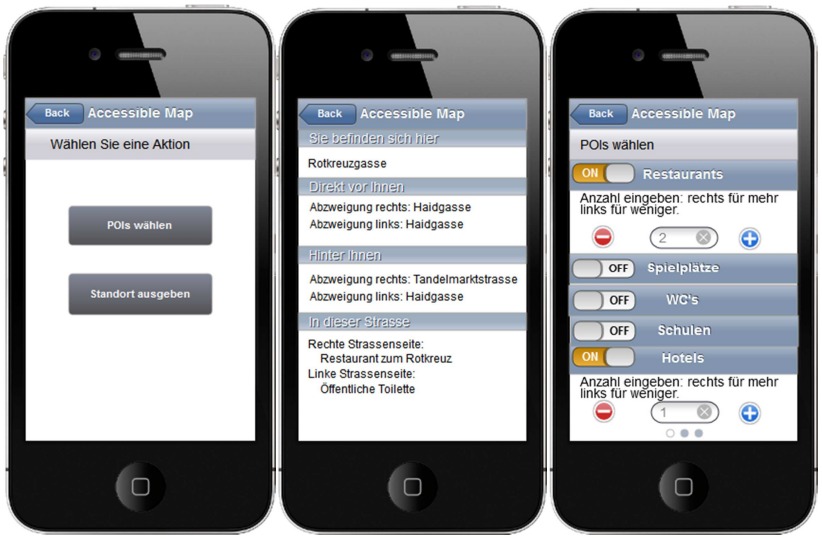


Abbildung 20 - Mockups zur Idee 1

Umsetzung

Man bräuchte zu jeder Strasse die jeweiligen Informationen, welche Strassen vor- und hinter ihr liegen, sowie die jeweiligen POIs. Diese Information braucht man bei jedem Aufruf von „Standort ausgeben“.

Idee 2

Mittels optischem und haptischem Feedback soll der Benutzer auf der App ertasten, wo er sich auf der Karte befindet und sich dann orientieren, indem er mit dem Finger über die Karte fährt (siehe Abbildung 21 links).

So könnte er zum Beispiel als erstes durch ein immer lauter werdendes Geräusch herausfinden, wo er sich befindet. Anschliessend kann er mit dem Finger auf der Karte herumfahren. Dabei wird für jedes Objekt ein anderes Geräusch zurückgegeben. Für ein Gebäude ertönt zum Beispiel ein dumpfer Ton, für eine Kreuzung ein höhere. Will er dann mehr Information zu einem Gebäude oder einer Strasse, so kann er auf die gewünschte Stelle tippen und es erfolgt eine Sprachausgabe. Der Standpunkt des Benutzers würde sich immer aktualisieren, während er sich fortbewegt. Je nach gewählten POIs (siehe Abbildung 21 rechts) gibt es dann mehr abrufbare Informationen.

Abbildung 21 - Mockups zur Idee 2

Umsetzung

Man müsste sich anhand des mittels GPS (und allenfalls WLAN) ermittelten Standpunkts alle Daten dieser Strasse sowie die nächstgelegenen POIs liefern lassen um Sie auf der Karte richtig zu platzieren. Der Standpunkt muss dann alle paar Sekunden aktualisiert und Informationen nachgeladen werden.

Idee 3

Der aktuelle Standort soll mit Strassennamen und Hausnummer angegeben werden. Eine regelmässige Aktualisierung wäre möglich (z.B. alle 10 Sekunden). Kreuzende Strassen sollen mit Richtungs- und Entfernungsangabe angezeigt werden, damit der Nutzer weiss, wo er wann abbiegen kann.

Mit einer Standortsuche soll die Entfernung und Richtung zu einem Ort angegeben werden (siehe Abbildung 22 links). Auch hier ist eine regelmässige Aktualisierung einstellbar. Falls gewünscht, können POIs in der Nähe des aktuellen Standorts angegeben werden. Welche Arten von POIs wichtig sind (zum Beispiel Restaurants oder Sehenswürdigkeiten), wird in den Einstellungen bestimmt.

Durch Speichern von Favoriten können bestimmte Orte schneller wieder gefunden werden und Vergleiche zwischen deren Entfernung und Richtung gemacht werden. Zusätzlich zu einer textuellen Angabe kann auf eine Kartenansicht gewechselt werden (siehe Abbildung 22 rechts). Darauf kann der Nutzer mit dem Finger streichen und so die Strassennamen und weitere Angaben grafisch erkunden. Somit soll er ein Gefühl für die Entfernungen und Positionen erhalten.

Abbildung 22 - Mockups zu Idee 3

Umsetzung

Es muss bekannt sein, wo sich welche Strassen schneiden. Die POIs müssen in Kategorien eingeteilt sein, um eine sinnvolle Auswahl zu ermöglichen.

1.6.2.2 Mockups

Konzeptidee 1 wurde übernommen und das Design detaillierter ausgearbeitet. Die Funktionalitäten sollen möglichst benutzerfreundlich angeboten werden.

Hauptbildschirm

Die Anwendung soll nach dem Starten auf den Hauptbildschirm leiten. Wie in nebenstehender Abbildung zu sehen ist, kann dabei zwischen 2 Punkten gewählt werden. Auch lässt sich über die untere Toolbar am unteren Bildschirmrand navigieren. Die aktuelle Ansicht kann über das Haussymbol wieder erreicht werden.

Abbildung 23 – Mockup Hauptbildschirm

Standort ausgeben

Bei Tippen auf den Knopf „Standort ausgeben“ oder auf den Pin in der Toolbar wird der aktuelle Standort ausgegeben.

Die Ausgabe beinhaltet die Strasse, in der sich der Nutzer befindet, welche Kreuzungen vor und welche hinter ihm liegen und welche POIs sich in der Nähe befinden. Die POIs werden nach Entfernung geordnet aufgelistet und haben eine Richtungsangabe, die im Uhrzeitformat angegeben wird. So bedeutet 3 Uhr, dass sich der Ort rechts vom Nutzer befindet. Um die Ausgabe zu aktualisieren, wird auf den Knopf „Update“ gedrückt.

Abbildung 24 – Mockup Standortausgabe

POIs auswählen

Abbildung 25 - Mockup POIs wählen

Durch Betätigen des Knopfs „POIs wählen“ oder Tippen auf das Zahnrad in der Toolbar wird eine Liste mit Kategorien von POIs angezeigt. Hier kann gewählt werden, welche POIs ausgegeben werden sollen, indem der Schieber rechts neben dem Kategoriennamen auf „An“ oder „Aus“ gestellt wird. Die Liste der Kategorien ist alphabetisch geordnet. Der Knopf „Vorherige“ leitet auf die vorangehende Seite, der Knopf „Weitere“ auf die nächste Seite.

Um den Umkreis für die anzuzeigenden POIs zu verändern, wird auf die Schaltfläche „Umkreiseinstellungen“ gedrückt.

Umkreiseinstellungen

Abbildung 26 – Mockup Umkreiseinstellungen

In den Umkreiseinstellungen wird der Umkreis festgelegt, der für die POI-Suche verwendet werden soll. Die Zahl kann manuell durch Tippen auf das Zahlenfeld eingegeben werden oder in 100 m Schritten auf- oder abgezählt werden.

Informationsbildschirm

Wird in der Toolbar auf das „i“ getippt, wechselt man zum Informationsbildschirm. Hier bekommt man Hilfe zur Bedienung der Applikation. Ausserdem werden die Entwickler und alle Beteiligten genannt.

1.6.3 Begriffserklärungen OpenStreetMap

Im folgenden Teil der Arbeit werden die Begriffe „Way“, „“Node“ und „Tag“ verwendet. Der sogenannte „Node“ kann einen POI, Orientierungspunkt (Point of Orientation, kurz POO) oder andere Objekte repräsentieren. Ein sogenannter „Way“ steht in OpenStreetMap für eine Linie und besteht aus mehreren Nodes. Er kann zum Beispiel eine Strasse oder nur ein Strassensegment darstellen. Beide Elemente können sogenannte „Tags“ aufweisen. Dies sind Merkmale oder Zusatzinformationen wie zum Beispiel die Maximalgeschwindigkeit einer Strasse, die Gattung eines Baumes und vieles mehr.

1.6.4 Algorithmen

Die im Folgenden erkannten Problemstellungen wurden mit eigens entwickelten oder recherchierten Algorithmen gelöst.

1.6.4.1 Berechnung einer „bounding box“

Eine sogenannte „bounding box“ bezeichnet in OpenStreetMap einen Rahmen um eine Koordinate (Quadrat in Abbildung 27). Sie wird benötigt um Bereichsabfragen in OpenStreetMap abzusetzen. Definiert wird sie durch die Koordinaten des linken unteren sowie des rechten oberen Eckpunktes. Um sie zu berechnen wird die Koordinate sowie ein gewünschter Radius als Abstand zur Koordinate benötigt. Weil mit Koordinaten gerechnet wird, muss der Radius im ersten Schritt in Grad konvertiert werden.

$$Radius\ in\ Grad = \frac{radius}{111{,}111}$$

Die Zahl 111,111 steht für ein Grad im kartesischen Koordinatensystem. Sie berechnet sich folgendermassen:

$$\frac{10^7}{90} = 111{,}111$$

Da in der Anwendung nur ein Radius von maximal 2000 Meter möglich ist, reicht diese Formel zur Berechnung aus. Die Erdkrümmung kann vernachlässigt werden. Für die Berechnung des Abstandes in Längengrad wird folgende Formel verwendet:

$$Abstand\ Längengrad = Radius\ in\ Grad * \cos\left(\frac{Längengrad\ der\ Koordinate * \pi}{180}\right)$$

Dieser Abstand muss dann vom Längengrad der Koordinate für den unteren Eckpunkt subtrahiert und für den oberen Eckpunkt (roter Punkt in Abbildung 27) addiert werden.

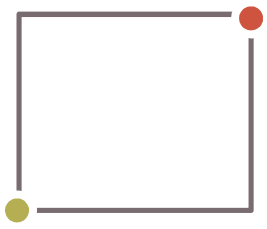


Abbildung 27 - "bounding box"

Für den Breitengrad des unteren Eckpunktes (grüner Punkt in Abbildung 27) wird der Radius in Grad von der ursprünglichen Breitengradkoordinate subtrahiert und für den des oberen addiert.

Die Formel dazu stammt aus einem Forum für OSM-Benutzer [14].

1.6.4.2 Distanzberechnung mit Erdkoordinaten

Beim Rechnen mit Koordinaten wird die „Koversinus“ Funktion verwendet. Sie ermöglicht es, die Neigung der Erdkugel in die Distanzrechnung miteinzubinden. Implementiert wurde dazu eine Formel der Webseite „Movable-Type Scripts“ [14], die sich mit geographischen Berechnungen beschäftigt.

1.6.4.3 Nächstgelegenen „Node“ finden

Um für die vom Routing-Dienst erhaltenen Koordinaten die passenden „Nodes“ in OpenStreetMap zu finden wird zuerst eine „bounding box“ berechnet. Diese beinhaltet den höchsten und niedrigsten Längen- sowie Breitengrad der Route. In dieser „bounding box“ werden via Overpass API alle „Nodes“ und „Ways“ in einer einzigen Abfrage gewonnen und zwischengespeichert. Es wird dann die Distanz zu allen „Nodes“ berechnet und derjenige mit der kleinsten Distanz als passender „Node“ für die Koordinate übernommen.

Damit ein Fehler in den Daten keine falsche Route generiert, muss die kleinste Distanz kleiner als 15m sein. Ein solcher Fehler kann zum Beispiel durch einen gerade erst gelöschten „Node“, welcher in den Daten des Routingdienstes noch vorhanden ist, entstehen.

1.6.4.4 Nächstgelegenes Strassensegment finden

Bei der Standortlokalisierung muss beachtet werden, dass der nächstgelegene „Node“ nicht unbedingt zu der nächstgelegenen Strasse gehört. In der folgenden Abbildung ist die Problematik aufgezeigt. Man nehme an, man steht an der Stelle, wo der rote Pfeil hinzeigt. Die rote Linie ist also die Strasse, welche am nächsten liegt. Durch den reinen Vergleich von Abständen zu den naheliegenden „Nodes“ der Strassen würde man jedoch hier der grauen Linie, genauer, dem „Node“ 2 zugeteilt werden.

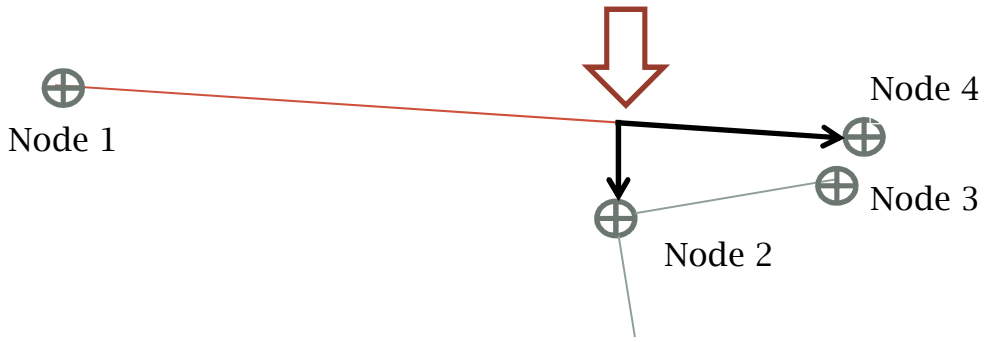


Abbildung 28 - Darstellung nächstes Strassensegment finden

Deshalb wurde eine Berechnung zum nächsten Segment angestellt. Hierbei werden alle Segmente der umliegenden Strassen mit dem aktuellen Standort folgendermassen verglichen:

Gegeben: Punkt p mit x und y Koordinate, Start- und Endpunkt (s und e) des Segments mit x und y Koordinaten.

Gesucht: Abstand von p zur Linie zwischen s und e.

Als erstes wird die Distanz im Quadrat zwischen v und w folgendermassen berechnet:

$$Distanz\ im\ Quadrat = (s.x - e.x)^2 + (s.y - e.y)^2$$

Falls die Distanz im Quadrat 0 ist, dann ist die Distanz zum Strassensegment gleich dem Abstand zum Startpunkt. Ist sie grösser als 0, so wird das Integral anhand folgender Formel berechnet.

$$Integral = (p.x - s.x) * (e.x - s.x) + (p.y - s.y) * (e.y - s.y)$$

Dieses Integral wird nun durch die Distanz im Quadrat geteilt. Falls das Resultat nun kleiner als 0 ist, ist das Endresultat gleich der Distanz im Quadrat zwischen p und s.

Falls das Resultat grösser als 1 ist, ist das Endresultat gleich der Distanz im Quadrat zwischen p und e.

Für alle Werte des Resultats zwischen 0 und 1 ist das Endresultat die Distanz im Quadrat von p, zu einem neuem Punkt mit folgenden Koordinaten, wobei t für den Wert des Resultats steht.

$$x: (s.x + t * (e.x - s.x))\ \ y: (s.y + t * (e.y - s.y))$$

Um nun den Abstand zum Strassensegment zu erhalten wird die Wurzel des anhand der obigen Formel berechneten Resultats genommen.

1.6.4.5 Berechnung der Ausgabe für POIs in einem Radius

Für das Auffinden der POIs wird ein Umkreisradius verwendet. Diesen kann der Anwender aus den Werten 200, 500, 1000, 1500 und 2000 Meter selbst wählen. Mittels Pythagoras wird der Abstand für eine „bounding box“ berechnet.

$$Abstand = \sqrt{r^2 + r^2}$$

In diesem Bereich werden nun über Overpass-Abfragen die verschiedenen POIs gesucht. Die Resultate werden dann auf die Entfernung geprüft. Damit fallen die Resultate, die sich ausserhalb des eingestellten Umkreises befinden (gelber Kreis in Abbildung 29) weg.

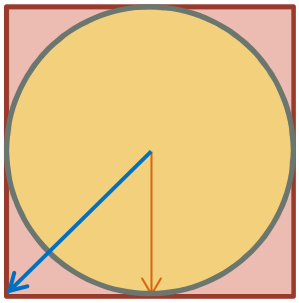


Abbildung 29 - Bounding Box für POI-Suche

1.6.4.6 Berechnung der Orientierungspunkte in der Strasse

Wie die POIs werden auch die meisten Orientierungspunkte über das Overpass API abgefragt. Die „bounding box“ wird dafür auf die Grösse des ganzen „Ways“ beziehungsweise der Route angepasst.

Bei der Standortausgabe werden alle „Nodes“ berücksichtigt, die sich in der aktuellen Strasse befinden. Es werden jene Punkte beachtet, die sich im 8 Meter breiten „Buffer“ der Strasse und in einer Reichweite von 300 Meter zum aktuellen Standort befinden. Beim Routing werden nur die Segmente mit den „Nodes“ der Route genommen, die auch passiert werden müssen.

Für eine optimale Orientierung wird eine Unterscheidung zwischen rechter und linker Strassenseite gemacht. Somit werden dem Benutzer nur Orientierungspunkte angesagt, die er auch antreffen kann. Die Berechnung der Strassenseite wird im Unterkapitel 1.6.4.11 erklärt. Die vorhandenen Orientierungspunkte werden in verschiedenen „Buffer“ für die rechte und linke Strassenseite gespeichert. Um diese beiden „Buffer“ zu erhalten, werden als erstes die Eckpunkte der Linien berechnet, um die dann ein 4 Meter breites Polygon erzeugt wird. Die Koordinaten werden mit folgender recherchierten Formel [14] berechnet:

Gegeben: Zum Winkel der Strasse wird für die rechte Seite 90° addiert, für die linke Seite 90° subtrahiert. Der Abstand von der Strasse zu den Bufferkoordinaten beträgt 4 Meter. Dieser wird in der Formel als „dif“ bezeichnet. Die Koordinaten der Start- und Endnodes (schwarze Punkte in Abbildung 30) werden als „lat“ (Längengrad) und „lon“ (Breitengrad) bezeichnet. Der Winkel, der Breiten- und der Längengrad sind im Bogenmass angegeben.

Gesucht: Koordinate mit Längen- und Breitengrad, die sich parallel verschoben zum „Node“ der Strasse befindet (blaue und rote Punkte in Abbildung 30).

$$\text{Längengrad} = \sin^{-1}(\sin(\text{lat}) * \cos(\text{dif}) + \cos(\text{lat}) * \sin(\text{dif}) * \cos(\text{Winkel}))$$

Falls der Kosinus des Längengrades 0 ergibt, wird derselbe Breitengrad zurückgeliefert, der übergeben wurde. Ansonsten wird er berechnet.

$$\text{Breitengrad} = ((\text{lon} - \sin^{-1}(\sin(-\text{Winkel}) * \frac{\sin(\text{dif})}{\cos(\text{Längengrad})} + \pi)\%(2 * \pi)) - \pi$$

Die resultierenden Koordinaten werden dann noch in Grad umgerechnet.

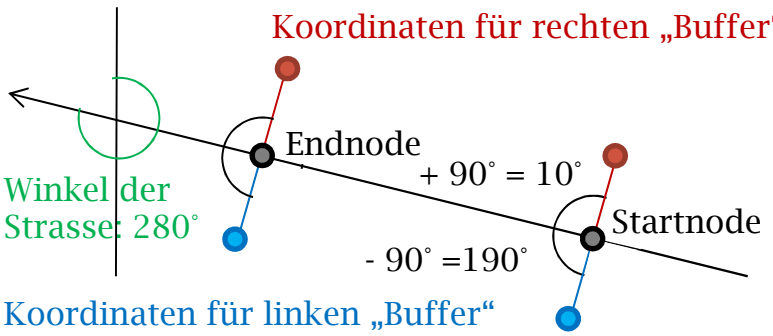


Abbildung 30 - Berechnung der Seitenbufferkoordinaten

Nun wird der Point-In-Polygon Algorithmus angewendet um zu ermitteln, welche der abgefragten Orientierungspunkte sich in den „Buffers“ (Polygons) befinden. Der Algorithmus wurde im Internet recherchiert [15] und läuft folgendermassen ab:

Es wird eine Liste der x und y Koordinaten des Buffers erstellt (in der Formel als „xlist“ und „ylist“ definiert). Die Koordinaten eines zu prüfenden Orientierungspunktes werden in die Variablen „lat“ für Längengrad und „lon“ für Breitengrad gespeichert. Um auf die Werte des jeweiligen Eintrags an einer Stelle der Liste zuzugreifen, werden die Indexe „i“ und „j“ genutzt, wobei „j“ immer um eins kleiner ist als „i“ und damit den vorangehenden „Node“ betrifft.

Nun müssen zwei Bedingungen erfüllt sein, damit der Orientierungspunkt als im „Buffer“ erkannt wird.

1. $ylist[i] > lon \neq ylist[j] > lon$

2. lat < (xlist[j] – xlist[i]) ∗ (lon − ylist[i]) / (ylist[j] − ylist[i]) + xlist[i]

Falls beide Tests bestanden werden, wird eine Variable, die anfangs auf falsch eingestellt ist, auf den jeweils anderen Wahrheitswert gesetzt.

Nachdem durch das ganze Polygon iteriert wurde, wird die Variable mit dem Wahrheitswert geprüft. Falls er wahr ist und die Distanz zum Orientierungspunkt nicht grösser als die Strecke, wird er zur Liste der Orientierungspunkte im „Buffer“ eingefügt.

Abbildung 31 veranschaulicht den Point-In-Polygon Algorithmus und bietet drei Beispiele mit Ergebnissen. Der linke gelbe Punkt fällt beim ersten Test durch, weil seine y-Koordinate kleiner ist als die jeden Punktes des Polygons. Beim oberen gelben Punkt wird der Wahrheitswert im Laufe des Algorithmus mehrmals gewechselt, da er sich über dem Polygon befindet. Dies wird durch den zweiten Test bewirkt, der den x-Wert des gelben Punktes mit den Koordinaten der Polygonecken vergleicht. Nach Ablauf der ganzen Schleife steht der Wahrheitswert jedoch auf falsch. Beim grünen Punkt sind alle Bedingungen erfüllt.

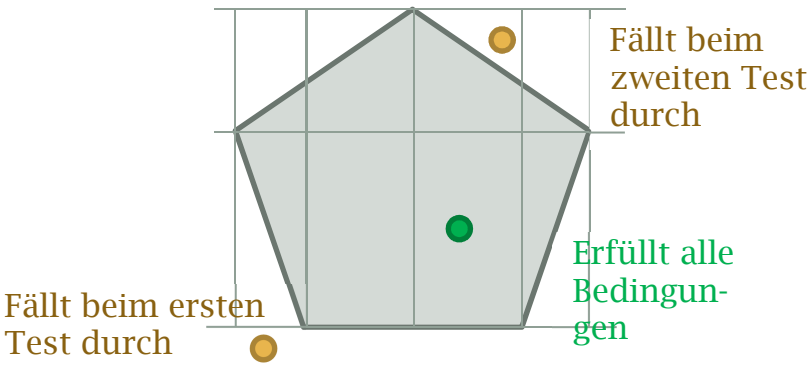


Abbildung 31 - Darstellung des Point-In-Polygon Algorithmus

In Abbildung 32 ist ein Kartenausschnitt der Zeughausstrasse in Winterthur mit der dazugehörigen Standortausgabe zu sehen. Der Kompass des Geräts ist in diesem Beispiel nach Norden ausgerichtet. In der Karte wurden die beiden Seitenbuffer als blaue und rote Bereiche um das jeweilige Strassensegment eingezeichnet. Die roten Punkte sind dabei die „Nodes“ der Strasse, die grünen sind in OpenStreetMap erfasste Bäume. Der textuelle Ausschnitt (Abbildung 32 rechts) zeigt, dass die Orientierungspunkte nach Entfernung geordnet sind. Darum sind zwischen den Bäumen auch Nennungen von Kreuzungen vorhanden.

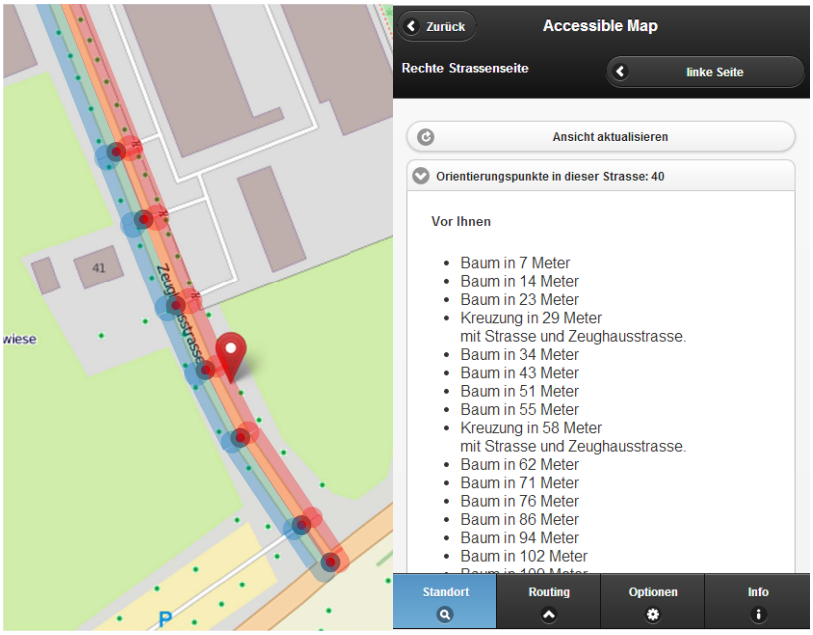


Abbildung 32 – Vergleich Kartenausschnitt mit Seitenbuffer und textuelle Beschreibung

Das Auffinden von Baustellen und Kreuzungen stellen Ausnahmen dar, weil sie auf beiden Seiten der Strasse aufgelistet werden. Für sie muss der Point-In-Polygon-Algorithmus nicht angewendet werden. Wie Kreuzungen gefunden werden, wird in Kapitel 1.6.4.8 ausführlicher beschrieben.

Die Baustellen werden von der Baustellen-Datenbank "Traffic Obstruction Database" (TROBDB) [1] abgefragt. Um die Daten zu erhalten werden die Way-Ids der Route als Parameter an die Datenbank übergeben. Als Ergebnis wird ein Objekt zurückgeliefert. Wichtig für die entwickelte Anwendung ist das Attribut der Dauer sowie der Fläche (dies kann ein Punkt, eine Linie oder ein Polygon sein). Aus dieser geometrischen Form wird nun, sofern die Baustelle in Betrieb ist (das aktuelle Datum wird mit der Dauer verglichen), der am Nächsten gelegene Punkt zum aktuellen Standort oder beim Routing zum nächsten Routenpunkt berechnet. Somit wird der Anwender informiert, falls sich in der Strasse eine Baustelle befindet.

1.6.4.7 Open Government Data der Stadt Zürich

Die Stadt Zürich bietet viele öffentlich zugängliche Daten auf ihrer Webseite an. Für die Arbeit von Bedeutung waren nur Bäume und Abfalleimer, da sie als Orientierungspunkte für Blinde und Sehbehinderte genutzt werden können. Diese Daten wurden aus einem heruntergeladenen JSON-File gelesen. Sie sind detaillierter erfasst als jene in OpenStreetMap, wobei sie nur einmal im Jahr aktualisiert werden.

Werden Bäume oder Abfalleimer als Orientierungspunkte ausgewählt, wird vor der Suche eine Abfrage gestartet, ob es Koordinaten in der Route gibt, die sich in Zürcher Stadtgebiet befinden. Dazu wird wie bei den Seitenbuffer für die Strasse der Point-In-Polygon-Algorithmus verwendet. Das Polygon ist dabei die Grenze des Stadtgebietes. Wird eine Koordinate im Stadtgebiet gefunden, werden für die Route die Zürcher-, an Stelle der OSM-Daten zu Bäumen und Abfalleimern verwendet. Dafür wird eine „bounding box“ der Grösse der gesamten Route berechnet und für die darin gefundenen Bäume oder Abfalleimer aus Zürich später geprüft, ob sie im „Buffer“ einer Strasse liegen. Wie die Orientierungspunkte in einer Strasse berechnet werden, wurde bereits im vorangehenden Abschnitt erläutert.

In Abbildung 33 werden zwei Ausschnitte gezeigt. Der rechte Ausschnitt stammt vom „Züriplan“ [17], der linke aus OpenStreetMap. Die Bäume sind im linken Bild als grüne Kreise, im rechten als grüne Punkte erkennbar. Es ist klar ersichtlich, dass die Karte der Stadt Zürich viel mehr Einträge besitzt.

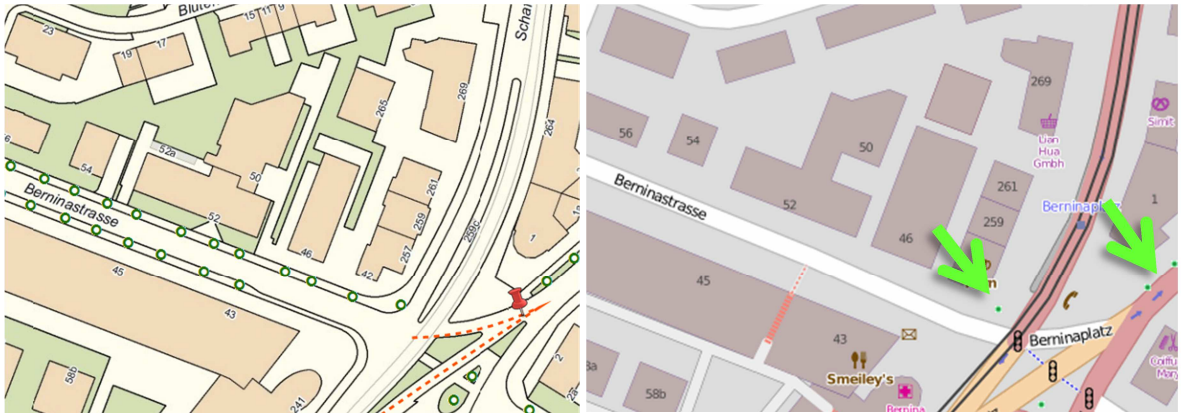


Abbildung 33 - Vergleich Zürcher Bäume (oben) mit OpenStreetMap Bäumen

1.6.4.8 Finden von Kreuzungen mit aktueller Strasse

Bei der Standortausgabe werden Kreuzungen der Strasse, in der man steht, mit anderen Strassen ausgegeben. Dies beinhaltet auch Kreuzungen mit Fusswegen und Treppen. Es werden sowohl die Entfernung zum aktuellen Standort als auch die Namen der kreuzenden Strassen angesagt, sofern sie in OSM erfasst sind. Ist dies nicht der Fall, werden die „Tags“ des „Ways“ analysiert. Dies kann dazu führen, dass die kreuzenden Wege als „Fussweg und Strasse“ angegeben werden.

Um die Kreuzungen zu finden werden alle „Ways“ in einer „bounding box“ mit 100 Meter Abstand zum aktuellen Standort über das Overpass API mit dem Keyword „highway“ abgefragt. Danach werden ihre Nodekoordinaten mit denen der aktuellen Strasse verglichen. Stimmen diese überein, ist eine Kreuzung vorhanden. Es werden alle „Ways“ hinzugefügt, die sich an diesem Punkt schneiden.

1.6.4.9 Finden von unvermerkten Kreuzungen

Es kann vorkommen, dass sich in OpenStreetMap Strassen ohne gemeinsamen Knotenpunkt kreuzen (siehe roter Pfeil Abbildung 34). Dies sind Fehler in den Daten und sollten verbessert werden. Zum einen kann es sich um Kreuzungen mit Brücken oder Tunnel handeln, wobei die Strassen nicht über die benötigten „Tags“ verfügen. Zum anderen ist es möglich, dass vergessen wurde, einen Schnittpunkt in Form eines „Nodes“ zu setzen. Da der tatsächliche Aufbau aus den gegebenen Daten nicht erkenntlich ist, werden diese Punkte in der Anwendung als „Unvermerkte Kreuzungen“ angegeben. Sie sind als Orientierungspunkte in der Standortansicht auswählbar und werden in der Standortausgabe und dem Routing angegeben.

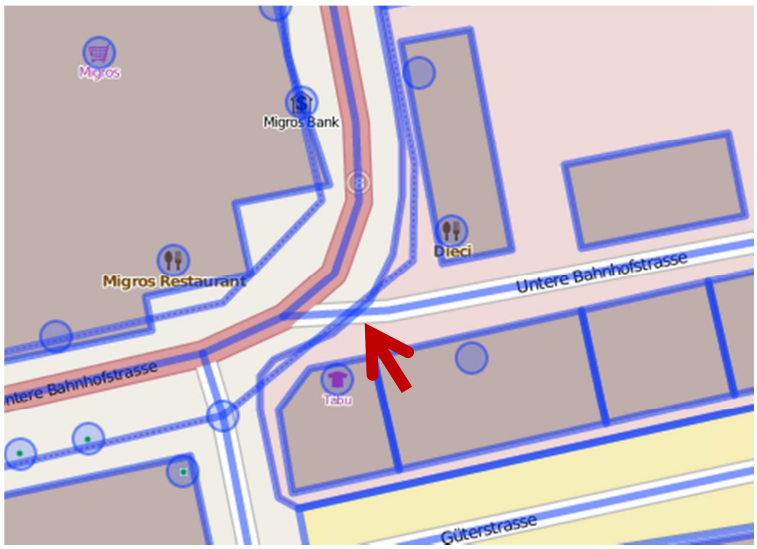


Abbildung 34 - Kartenausschnitt mit Daten von OpenStreetMap überlagert.

Wie bei den normalen Kreuzungen werden alle „Ways“ einer „bounding box“ genutzt und einzeln miteinander verglichen, wobei Brücken und Tunnel nicht beachtet werden. Von jedem „Way“ werden je zwei „Nodes“ genommen, die nacheinander auftreten und eine Linie bilden und mit der Linie zweier „Nodes“ eines anderen „Ways“ verglichen. Nun prüft man mit einer mathematischen Formel, ob sie einen gemeinsamen Schnittpunkt haben.

Schnittpunkte in der Verlängerung der Linien werden nicht beachtet (siehe Abbildung 35 rechts). Der mathematische Algorithmus wurde im Internet recherchiert [12] und sieht folgendermassen aus:

Für den Startnode der ersten Linie werden die Variablen „x1“/„y1“ verwendet, „x2“/„y2“ für den Endnode. Gleichermassen steht „x3“/„y3“ für den Startnode der zweiten Linie und „x4“/„y4“ für Endnode. Der Längengrad wird jeweils in der „x“ Variable gespeichert, der Breitengrad in der „y“ Variable. Es wird die Determinante aus den x und y Abständen der Linien ermittelt.

$$Determinante\ d = (x2 - x1) * (y4 - y3) - (y2 - y1) * (x4 - x3)$$

Falls die Determinante ungleich 0 ist, sind die Linien keine Parallelen. Es gibt einen Schnittpunkt, der mit den Determinanten für x und y ermittelt wird.

$$Determinante\ x = \frac{(x4 - x3) * (y1 - y3) - (y4 - y3) * (x1 - x3)}{(x2 - x1) * (y4 - y3) - (y2 - y1) * (x4 - x3)}$$

$$Determinante\ y = \frac{(x2 - x1) * (y1 - y3) - (y2 - y1) * (x1 - x3)}{(x2 - x1) * (y4 - y3) - (y2 - y1) * (x4 - x3)}$$

Liegen die Werte beider Determinanten zwischen 0 und 1, kann ein Schnittpunkt berechnet werden.

$$x = x1 + Determinante\ x/d * (x2 - x1)$$

$$y = y1 + Determinante\ x/d * (y2 - y1)$$

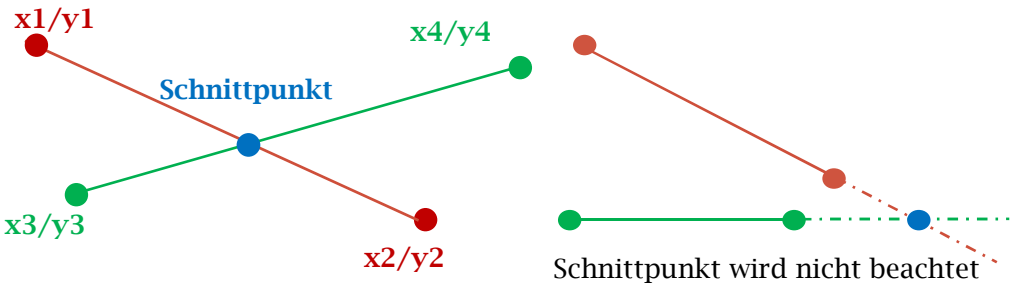


Abbildung 35 - Berechnung des Schnittpunktes

1.6.4.10 Berechnung der Anweisungen für das Fussgänger-Routing

Anhand der Koordinaten, welche der Routingdienst zurückgibt, werden die „Nodes“ und „Ways“ der Route bestimmt. Die „Nodes“ werden mit dem in Kapitel 1.6.4.3 beschriebenen Verfahren zum Finden der nächstgelegenen „Nodes“ gesucht. Anschliessend werden den gefundenen „Nodes“ die richtigen „Ways“ zugewiesen. Dabei wird zuerst geprüft ob der erste „Node“ einer Route derselbe ist wie der Zweite. Dies kann geschehen, falls man auf einer Strasse startet und der Routing-Dienst als erste Routing-Koordinate den nächsten „Node“ auf dieser Strasse liefert. In diesem Fall würden beide anhand der Distanzberechnung auf denselben „Node“ abgebildet werden. Um dies zu verhindern wird in diesem Fall für den ersten Routenabschnitt der in der Standortausgabe lokalisierte „Way“ anhand der Bestimmung des nächstgelegenen Strassensegments genommen. Für den Fall dass die „Nodes“ verschieden sind, wird verglichen, welchen „Way“ der nächste und der aktuelle

„Node“ gemeinsam haben. Dies ist dann der „Way“ für diesen Routenabschnitt.

Sollte eine Koordinate keinen „Node“ haben, der nahe genug ist (Distanz grösser als 15m), so wird diesem Routenabschnitt kein „Way“ zugewiesen.

Für alle Fälle, in denen ein passender „Node“ gefunden werden konnte, wird Anhand der so gewonnen „Ways“ für die einzelnen Abschnitte die Route mit Zusatzinformationen angereichert. Dies sind die maximale Geschwindigkeit, der Bodenbelag und, falls gegeben, das Attribut Treppe, Brücke, Fussweg oder Tunnel.

Für die Routenanweisungen an sich werden nicht zwingend „Ways“ benötigt. Die Orientierungspunkte werden im Bereich zwischen jeweils zwei Koordinaten in die linke und rechte Strassenseite eingeordnet. Die allererste Routinganweisung wird anhand des Kompasswerts des Smartphones generiert. Die Anweisungen zur Richtungsänderung werden anhand des Winkels zum nächsten und übernächsten „Node“ berechnet. In Abbildung 36 ist ersichtlich, dass der Winkel zwischen der Kompassrichtung (gelber Pfeil) beim Startpunkt und dem ersten „Node“ etwas grösser als 90° ist. Die erste Anweisung lautet daher: „rechts abbiegen“. Da der Winkel vom ersten zum zweiten „Node“ praktisch gleich ist, lautet die nächste Anweisung: „dann geradeaus weiterlaufen“. Als Distanzangabe wird der Abstand zwischen den Koordinaten berechnet. Für die Angabe der Orientierungspunkte in den Routenabschnitten wird das Verfahren aus Kapitel 1.6.4.6 verwendet.

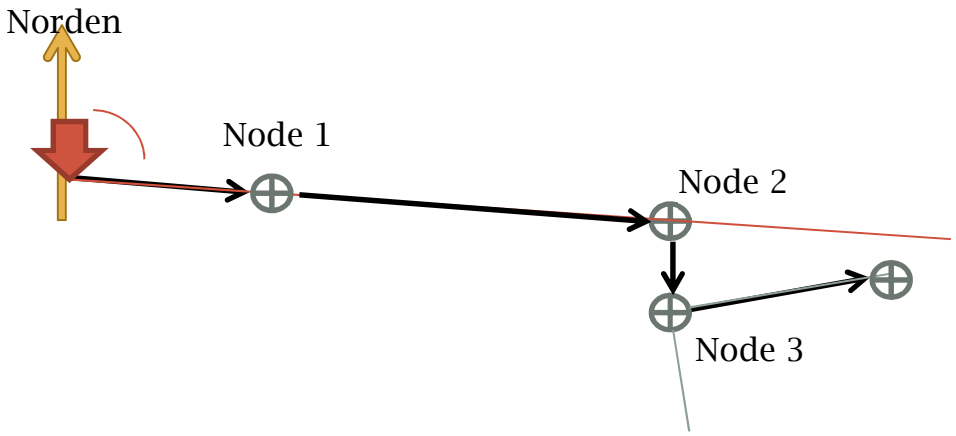


Abbildung 36 - Darstellung Routinganweisungen

Würde die Route bei „Node“ 2 weiter geradeaus gehen, so würden diese zwei Anweisungen zusammengefasst werden. Die Orientierungspunkte werden in so einem Fall ebenfalls zusammengefügt und die Distanzen neu berechnet. Dies geschieht jedoch nur unter der Voraussetzung, dass es sich immer noch um denselben „Way“ handelt. Ansonsten könnten die Zusatzinformationen verloren gehen.

1.6.4.11 Berechnung der Strassenseite

Weil Orientierungspunkte auf beiden Seiten der Strasse vorkommen, wird zwischen Auftreten auf linker und rechter Strassenseite unterschieden. Beim Wechseln auf die Standortausgabe oder auf eine Route wird die aktuelle Strassenseite ermittelt.

Um die Strassenseite zu berechnen, werden Angaben zur aktuellen Position, Blickrichtung und Richtung des Strassensegments relativ zur Nordachse benötigt. Während die ersten beiden Komponenten leicht mit GPS und eingebautem Kompass ermittelt werden können, muss der Winkel des Segments zuerst berechnet werden. Dazu wird als erstes geprüft, ob der Punkt, der als Ende des Segments eingetragen wurde, in Blickrichtung liegt. Ist dies nicht der Fall, werden die Punkte umgekehrt. Die Formel zur Berechnung des Segmentwinkels wird hier erklärt:

Gegeben: „lat1“/„lon1“ stehen für die Koordinaten des Startnodes und „lat2“/„lon2“ für die Koordinaten des Endnodes. Die Werte der Längen- und Breitengrade sind im Bogenmass. Der Abstand der Breitengrade wird als „dLon“ bezeichnet.
Gesucht: Segmentwinkel θ

$$\theta = \operatorname{atan2}(\sin(dLon) * \cos(lat2), \cos(lat1) * \sin(lat2) - \sin(lat1) * \cos(lat2) * \cos(dLon))$$

Als nächstes wird eine Formel benutzt, die bestimmt, ob ein Punkt auf der linken Seite einer Linie liegt oder nicht.

Gegeben: a Startknoten des Segments, b Endknotens des Segments, c aktueller Standort. Der Längengrad eines Punktes wird als „lat“ bezeichnet, der Breitengrad als „lon“.
Gesucht: Position relativ zum Strassensegment

Ist der resultierende Wert aus folgender Formel grösser als 0, so steht der Benutzer auf der linken Seite.

$$(blon - alon) * (clat - alat) - (blat - alat) * (clon - alon)$$

Das bedeutet auch, dass der Nutzer auf die rechte Strassenseite positioniert wird, falls er direkt auf der Linie steht. Sollte wegen schlechtem GPS Signals die Seite nicht stimmen, kann der Benutzer sie selbstständig wechseln. Dazu steht ihm ein Bedienelement auf dem Bildschirm zur Verfügung.

Abbildung 37 zeigt die Bedeutung der Blickrichtung zur Ermittlung der Strassenseite. Weil in der linken Grafik die Blickrichtung nach rechts zeigt, wird ermittelt, dass man sich auf der linken Strassenseite befindet. In der rechten Grafik ist man hingegen dem anderen Ende der Strasse zugewendet. Dies bewirkt einen Wechsel von Start- und Endknoten zur Bestimmung der Strassenseite.

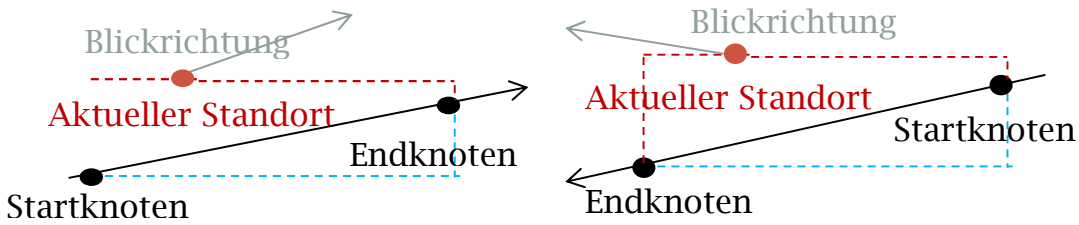


Abbildung 37 - Strassenseitenermittlung

1.6.5 Resultate

1.6.5.1 Funktionalitäten der entwickelten Anwendung

Als Resultat der Arbeit ging ein Prototyp einer barrierefreien Web-Anwendung für blinde und sehbehinderte Personen hervor. Er generiert einen grossen Mehrwert, da er eine völlig neue Art des Routings, nämlich eine mit Orientierungspunkten angereicherte Fussgängernavigation und Rundumsicht, bietet. Durch die Unterscheidung von linker und rechter Strassenseite wird die Orientierung zusätzlich verbessert. Auch dies ist bisher noch einzigartig.

Da die Bäume, Baustellen und Abfalleimer für Routen im Gebiet der Stadt Zürich direkt vom Open Government Portal der Stadt Zürich kommen, sind die Daten dort vollständiger als die aus OpenStreetMap. In den restlichen Gebieten der Schweiz werden die Daten komplett von OpenStreetMap bezogen. Fehlende oder unkorrekte Daten können so schnell korrigiert werden.

Das Beispiel nimmt an, dass ein Smartphone nach Norden zeigt und an dem mit einem roten Pin markierten Ort steht (siehe Abbildung 38). Der Benutzer wird beim Start der Anwendung von diesem Ort aus via GPS-Signal geortet.

Abbildung 38 - Kartenansicht Seequai Rapperswil

Auf der Startseite erscheint eine Liste mit Auswahlmöglichkeiten von Orientierungspunkten (standardmässig sind alle selektiert) und POIs (standardmässig sind Restaurants, Tram- und Bushaltestellen, Imbisslokale, Cafés, Supermarkte, Postboxen und öffentliche Toiletten selektiert). Dort kann der Benutzer einstellen, was er gerne angezeigt haben möchte. Hat der Benutzer seine Auswahl getroffen, so erhält er mittels eines Klicks auf „Ausgeben“ (siehe Abbildung 39 links) die Rundumsicht (Abbildung 39 Mitte und rechts) angezeigt.

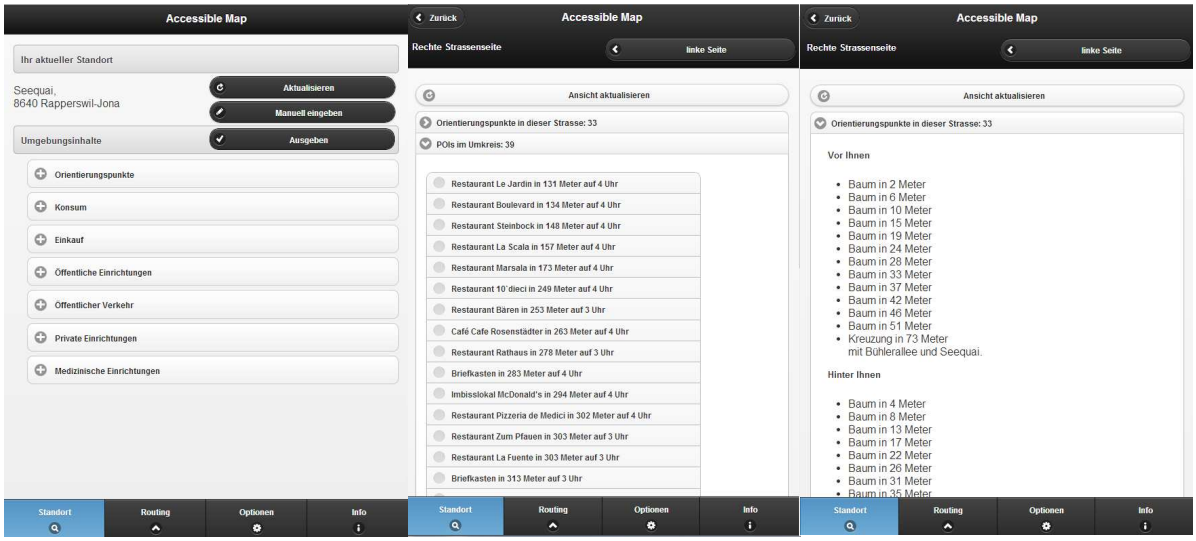


Abbildung 39 - Screenshots Standort und Rundumsicht Seequai Rapperswil

Wählt der Benutzer einen POI aus der nun erschienen Liste von POIs in der Umgebung an, so kann er diesen als Navigationsziel eingeben. Anschliessend wird eine Route generiert. In Abbildung 40 sieht man die textuelle Ausgabe einer mit Orientierungspunkten angereicherten Route. Die Routenabschnitte, welche Orientierungspunkte oder zusätzliche Angaben wie Strassenbelag und/oder Maximalgeschwindigkeit besitzen, sind als aufklappbares Bedienelement realisiert. Sollte der Anwender die Strassenseite wechseln und möchte nun die Orientierungspunkte auf der anderen Strassenseite sehen, so kann er „zeige rechte Seite“ anwählen und erhält somit die Orientierungspunkte der anderen Strassenseite (siehe Abbildung 40).

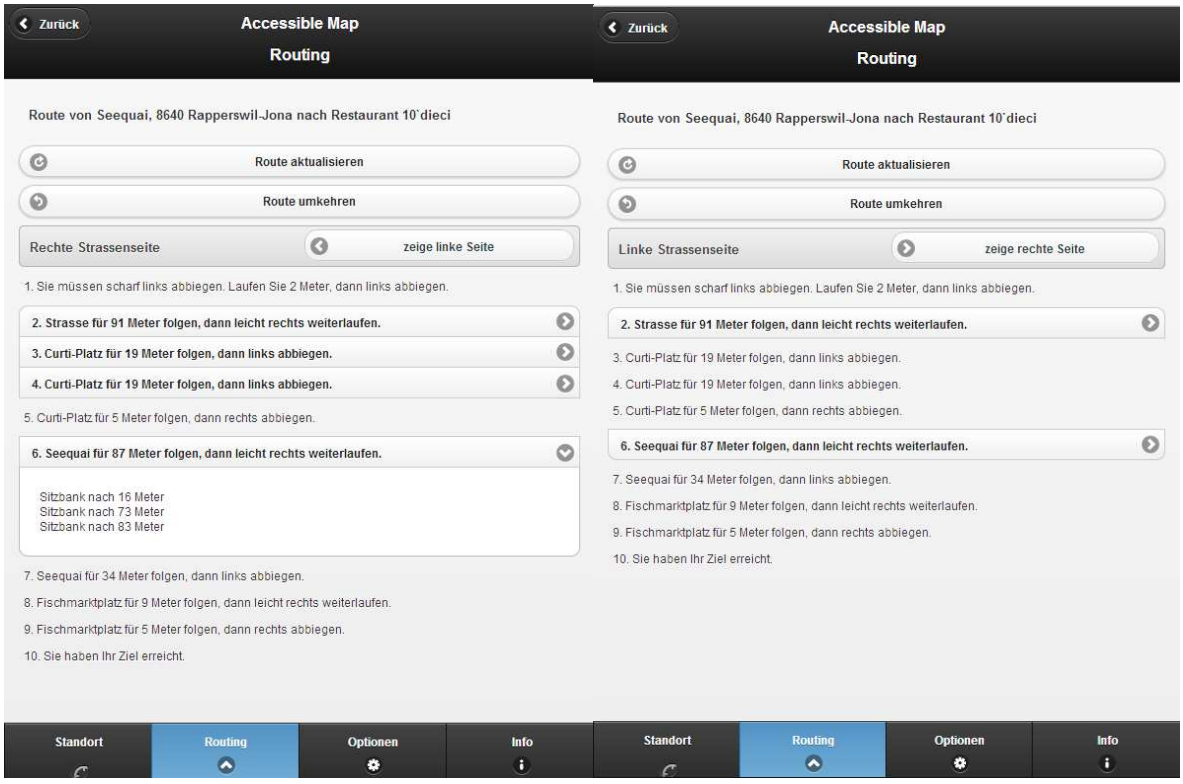


Abbildung 40 - Screenshots Routing

Die Bestimmung der Strassenseite erfolgt mittels mathematischer Formel (siehe Kapitel 1.6.4.11) und wird beim Generieren einer Route sowie bei der Ausgabe der Rundumsicht einmal bestimmt. Dazu wird der geräteinterne Kompass verwendet, sofern die Webseite auf einem Smartphone besucht wird. Beim Zugriff der Webseite von einem Computer wird als Kompasswert Norden genommen. Der Kompass wird nur einmal bei Beginn einer Routenabfrage oder Rundumsicht abgefragt. Anhand dieses Wertes wird in der Rundumsicht berechnet, welche Orientierungspunkte vor und hinter dem Benutzer liegen. Auch die Richtung POIs wird so berechnet. Beim Routing ist der Kompasswert nur für die erste Anweisung relevant. Die erste Anweisung führt den Benutzer zum Beginn der Route, steht er zum Beispiel mit dem Rücken zur Strasse, so lautet die erste Anweisung „Sie müssen sich umdrehen, dann…“.

1.6.5.2 Genauigkeit der generierten Daten

Die Genauigkeit von GPS sowie die des Kompasses ist nicht immer sichergestellt. Im schlimmsten Fall würde man an einen falschen Punkt lokalisiert werden, der bis zu 30m von der aktuellen Position entfernt ist. Dies ist meist der Fall, wenn zu viele hohe Gebäude dem GPS-Sensor die Sicht auf genügend GPS-Satelliten versperren. Im städtischen Gebiet könnte dies bedeuten, dass man aufgrund der Dichte von Strassen sogar auf eine falsche Strasse lokalisiert wird. Für diesen Fall wurde ein Ansatz zur manuellen Lokalisierung implementiert. Sie bietet dem Benutzer die Möglichkeit einen Strassennamen und einen Ort manuell einzugeben. Sofern es in OpenStreetMap nur einen „Way“ mit diesem Namen gibt, ist die neue Position nun der erste „Node“ dieser Strasse. In OpenStreetMap ist es jedoch besonders bei sehr langen Strassen (die lange gelbe Strasse in Abbildung 43) der Fall, dass es mehrere „Ways“ mit demselben Namen gibt. Die Anwendung liefert dann anhand des Suchresultates von Nominatim eine Auswahl aller „Ways“ mit diesem Namen. Um diese Problematik optimal zu lösen reichte die Zeit in der Entwicklungsphase nicht mehr.

Abbildung 41 - Suchresultate aus OpenStreetMap für die Oberseestrasse in Rapperswil

Eine praktikable Lösung wäre das Angeben von Kreuzungen an den beiden Enden der gefundenen „Ways“. Zusätzlich könnten Orientierungspunkte in den einzelnen Segmenten der Strasse angegeben werden um die Ortung für den Anwender zu erleichtern. Dies ist bei der Webseite von OpenStreetMap auch nicht optimal gelöst (siehe Abbildung 43). Man muss alle Resultate durchklicken und auf der Karte den Abschnitt verifizieren, den man möchte. Eingegeben wurde die Oberseestrasse in Rapperswil SG.

1.6.6 Schlussfolgerung

In der Analysephase fiel auf, dass zwar viele Applikationen einen Routing-Dienst verknüpft mit ihrer eigenen Anwendung anbieten, dieser jedoch nicht sehr gut auf blinde oder sehbehinderte Personen zugeschnitten ist. Für die Zielpersonen ist es wichtig, eine möglichst detaillierte Routenbeschreibung zu bekommen. Das visuelle Wiedererkennen einer schon einmal gelaufenen Strecke fällt für diese Personen je nach restlicher Sehstärke komplett weg. Sie orientieren sich mit ihrem Tastsinn und Gehör. Wichtig sind deshalb zum Beispiel Brunnen, die sie plätschern hören können und dann wissen, dass sie auf dem richtigen Weg sind. In der Arbeit konnte deshalb ein grosser Fortschritt in Bezug auf barrierefreies Fussgängerrouting erreicht werden.

Die Thematik erfordert für einen Sehenden sehr viel Einfühlungsvermögen. Deshalb war der Kontakt mit den Betroffenen und den Mitgliedern der Stiftung „Zugang für alle“ sehr wichtig. Es konnten wichtige Erkenntnisse über die barrierefreie Bedienbarkeit von Web-Applikationen sowie echte Bedürfnisse abgeholt werden. Anhand der vorhandenen Erkenntnisse ist eine Weiterentwicklung der Anwendung wünschenswert. Die Anwendung wurde gezielt so entwickelt, dass der Routingdienst, welcher die Koordinaten liefert, ausgetauscht werden kann. Auch das Erweitern um eine andere Sprache oder das Einfügen von mehr Daten der Stadt Zürich ist möglich.

1.6.6.1 Problematik Fussgängerstreifen

Was in der Arbeit nicht realisiert werden konnte, aber sehr wichtig für Blinde oder Sehbehinderte wäre, sind die Abschnitte der Route, die Fussgängerstreifen sind, von normalen Fusswegen zu unterscheiden. Die Problematik ist hier die Datenqualität von OpenStreetMap. Einige Fussgängerstreifen sind als „Nodes“ mit dem „Tag“ „crossing“ eingetragen, was für einen Fussgängerstreifen steht. In anderen Fällen wird der Fussgängerstreifen als „Way“ mit dem „Tag“ „highway = pedestrian“ (dieser „Tag“ steht eigentlich für eine reine Fussgängerzone) oder „highway = footway“ gekennzeichnet. In den meisten Fällen führt jedoch ein Fussweg über eine befahrene Strasse und es hat lediglich den „Node“ mit dem „Tag“ „crossing“. Man müsste bei einer Implementierung also sehr genau darauf achten, welcher Fall bei einer Strassenüberquerung der Fall ist. Weiter gibt es das Problem, dass einige Strassenüberquerungen gar nicht als solche eingetragen sind, weil die Fusswege oder Strassen gar keine gemeinsamen „Nodes“ haben. Dies konnten wir jedoch Abfangen und warnen den Benutzer auf der Route an so einer Stelle.

1.6.6.2 Stetige Verbesserung der Datenqualität

Das entwickelte Fussgängerrouting fiel bei den Mitarbeitern der Stiftung auf viel Anklang. Auch die Möglichkeit, dass sie zusammen mit einer blinden oder sehbehinderten Person fehlende Daten eintragen oder fehlerhafte Daten korrigieren können, gefällt ihnen sehr. Dies ist natürlich bei den Daten, die von der Stadt Zürich bezogen werden, nicht so einfach möglich. Diese haben dafür eine höhere Genauigkeit. Hätte man alle Daten, die OpenStreetMap bietet, auch von der Stadt Zürich beziehen können, so wäre im Gebiet der Stadt Zürich ein noch viel besseres Resultat möglich. Hier besteht noch sehr viel Potential, das noch ausgeschöpft werden kann.

1.6.6.3 Funktionsumfang vergrössern

Weiteres Potential steckt in den möglichen Funktionalitäten, welche von den Betroffenen Personen gewünscht worden sind, für die jedoch in dieser Arbeit keine Zeit vorhanden war.

Dies wären folgende Funktionalitäten:

- Angabe einer beliebigen Adresse mit Hausnummer als Ziel und/oder Startpunkt für Routinganfragen
- Suchen von POIs anhand von Suchbegriffen
- Kommentarmöglichkeit um anderen Benutzern an gewissen Stellen Hinweise zu hinterlassen (zum Beispiel: „Achtung: Sehr hoher Gehsteig“).
- Das Speichern von Favoriten (Adressen oder POIs)
- Das Tracking einer gelaufenen Strecke und die Möglichkeit, diese wiederzugeben.
- Das automatisierte Prüfen ob der Benutzer sich noch auf der Route befindet und Ausgeben einer Warnung inkl. Neuberechnung der Route.
- Das Ausgeben von POIs als erweiterte Orientierungspunkte während einer Route gefolgt wird.

2 Teil II: SW-Projektdokumentation

2.1 Analyse

2.1.1 Anforderungsspezifikation

2.1.1.1 Funktionale Anforderungen

Die Funktionalen Anforderungen bestehend aus Use Cases wurden anhand der Meetings mit der Stiftung erstellt und spiegeln die Bedürfnisse der blinden und sehbehinderten Menschen wieder. In diesem Projekt gelten folgende Use Cases:

Use Cases

UC1: Standort ausgeben: Der Nutzer erhält eine Auskunft darüber, an welcher Stelle er sich im Moment befindet. Die Ausgabe beinhaltet die Adresse mit Strassen- und Ortsangabe.

UC2: Umgebung ausgeben: Der Nutzer will Informationen zu seiner Umgebung erhalten. Dies beinhaltet Orientierungspunkte, die in derselben Strasse liegen, und POIs, die sich in einem bestimmten Umkreis befinden.

UC2a: POIs und Orientierungspunkte auswählen: Der Nutzer kann auswählen, welche Informationen zur Umgebung er erhalten will.

UC3: Manuell einen Standort wählen: Der Nutzer möchte die Umgebung eines entfernten Ortes erkunden oder möchte seine aktuelle Position korrigieren. Dazu fügt er manuell eine Adresse als Standort ein.

UC4: Routing zu einem POI: Der Nutzer will zu einem bestimmten POI geleitet werden.

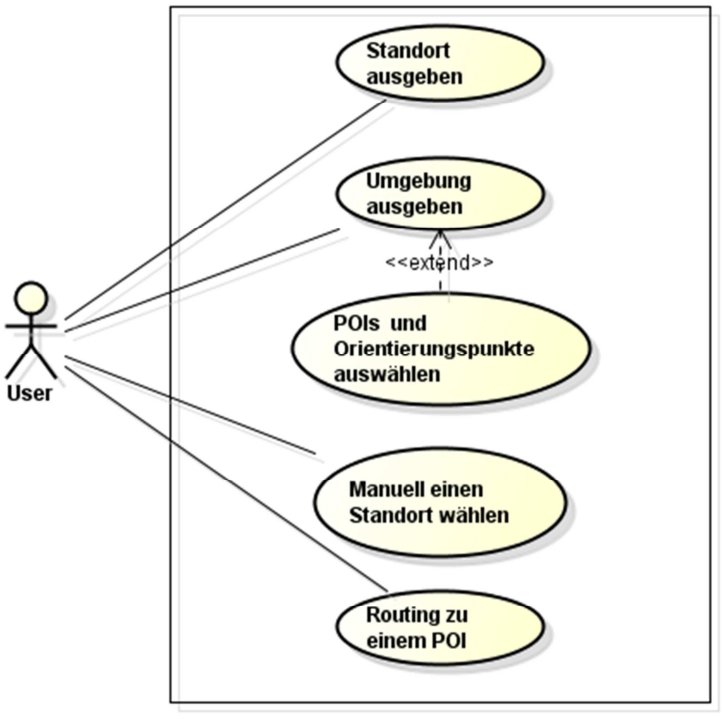


Abbildung 42 - Use Case Diagramm

2.1.1.2 Nicht-Funktionale Anforderungen

Verständlichkeit

Die Bedienung soll selbsterklärend sein. Wenn der Benutzer im Umgang mit Smartphones vertraut ist, sollte er keine zusätzliche Hilfe benötigen. Die einzelnen Ansichten sollen sich am Design von ähnlichen Applikationen, die im Abschnitt „Stand der Technik“ erwähnt werden, orientieren.

Zeitverhalten

Das Berechnen der Routen soll nicht länger als 10s dauern. Das Abfragen des aktuellen Standorts und die Ausgabe der POIs in der Umgebung soll nicht länger als 5s dauern.

Wartbarkeit, Änderbarkeit

Die Web-Applikation wird so geschrieben, dass sie um weitere statische Daten und andere Routing-Dienste oder Sprachen erweitert werden kann.

Portierbarkeit und Installierbarkeit

Die App ist eine reine Web-Applikation. Sie kann somit auf jedem Browser, der JavaScript unterstützt, verwendet werden. Man könnte das github Repository auch mit Hilfe von Phonegap in eine lokale Smartphone Anwendung umwandeln. Für den Prototyp reicht jedoch der Zugriff als Webseite.

2.2 Design

2.2.1 Objektdiagramm

Da JavaScript keine Klassen kennt, werden hier die verschiedenen Objekte erklärt, die in der Anwendung verwendet werden. Sie sind in der Datei objects.js definiert.

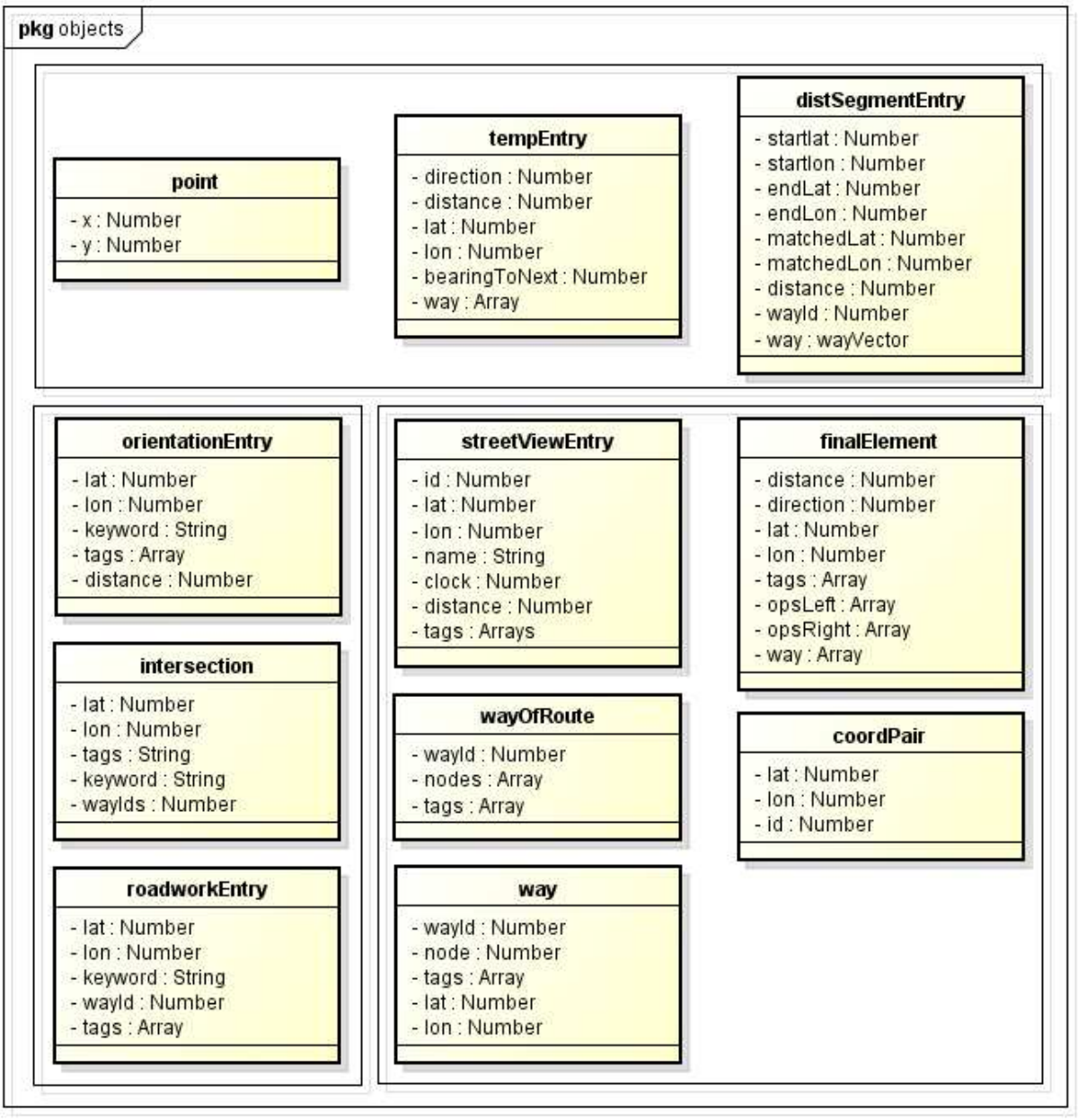


Abbildung 43 - Objektdiagramm

point

Das Point-Objekt wird für die Speicherung von Koordinaten gebraucht. X ist dabei der Längengrad, y der Breitengrad.

tempEntry

Dieses Objekt wird in vielen Scripts verwendet. Es wird gebraucht, um die Routen- und Strassenverlaufsinformation anzureichern. Zu den Attributen gehören Richtungsangabe für das Routing, Distanz und Winkel zum nächsten Knoten, Längen- und Breitengrad des Knotens und der „Way“.

distSegmentEntry

Mit dem distSegmentEntry-Objekt wird der berechnete am Nächsten liegende Strassenabschnitt zur aktuellen Position gespeichert. Somit stellt es die Basis für die Standortausgabe und die Routingberechnung dar. Das Objekt beinhaltet die Koordinaten des aktuellen Standorts (matchedLat und matchedLon), des Start- und Endknotens des Wegsegments sowie die Distanz zwischen Strassensegment und aktuellem Standort. Weiter kommen die Way-Id des Segments und der „Way“ welcher das Segment beinhaltet hinzu.

orientationEntry

Wie der Name andeutet wird in diesem Objekt ein Orientierungspunkt gespeichert. Es werden Längen- und Breitengrad, ein passendes Schlüsselwort, die „Tags“ und die Distanz zur aktuellen Position oder dem nächsten Routenpunkt gespeichert.

intersection

Kreuzungen werden getrennt von den Orientierungspunkten gespeichert, da andere Attribute zur Analyse benötigt werden. Zum einen sind unter den „Tags“ nur die Namen der kreuzenden Wege gespeichert. Zum anderen werden deren „Way-Id“s in einem Array gespeichert.

roadworkEntry

Auch Baustellen benötigen einen anderen Eintrag als Orientierungspunkte. Im Objekt wird ein Attribut wayId gespeichert, das anders als bei den Kreuzungen nicht aus einem Array, sondern aus einem einzelnen Eintrag besteht. Zudem sind die „Tags“ mit Zusatzinformationen zur Baustelle gefüllt, z.B. Öffnungszeiten, Strassenverengungen usw.

streetViewEntry

Um die POIs in der näheren Umgebung zu speichern, wird ein streetViewEntry-Objekt angelegt. Damit werden alle wichtigen Informationen zu den POIs gespeichert, wie id (eindeutige osm_id), Koordinaten, Bezeichnung, Uhrzeitrichtung, Distanz und „Tags“.

wayOfRoute

In diesem Objekt werden die Koordinaten für die in der „bounding box“ enthaltenen Wege eingefüllt. Ein Weg besteht meist aus mehreren Segmenten und damit aus mehreren Knoten. Auch hier werden noch die „Way-Id“ und die “Tags“ gespeichert.

way

Das Objekt way wird für das Routing gebraucht. Es speichert alle Informationen des Strassensegments, das sich auf der Route befindet. Dazu gehören die Way-ID und -“Tags“, die ID des gemappten „Nodes“ und die Koordinaten des Routenknotens. Daraus können später die Strasseninformationen wie Belag oder Höchstgeschwindigkeit gelesen werden.

finalElement

Die fertig berechnete Standortausgabe oder ein Abschnitt einer Route wird in diesem Element gespeichert. Neben den Orientierungspunkten auf der rechten und linken Strassenseite beinhaltet ein finalElement-Objekt die „Tags“ zum „Way“, Breiten- und Längengrad des Strassensegmentanfangs und die Richtungsangabe für das Routing.

coordPair

Ähnlich wie das point-Objekt speichert coordPair die Koordinaten eines Punktes. Allerdings ist auch die Node-Id des Knotens darin enthalten, damit beim Routing sichergestellt ist, dass der richtige Knoten verwendet wird.

2.2.2 Interaktionsdiagramm

Um die verschiedenen Informationen zu Adresse, Strassen, POIs, Orientierungspunkten und Routen zu erhalten, müssen verschiedene Webdienste abgefragt werden. Die in Abbildung 44 verwendeten Dienste werden auf den folgenden Seiten erläutert.

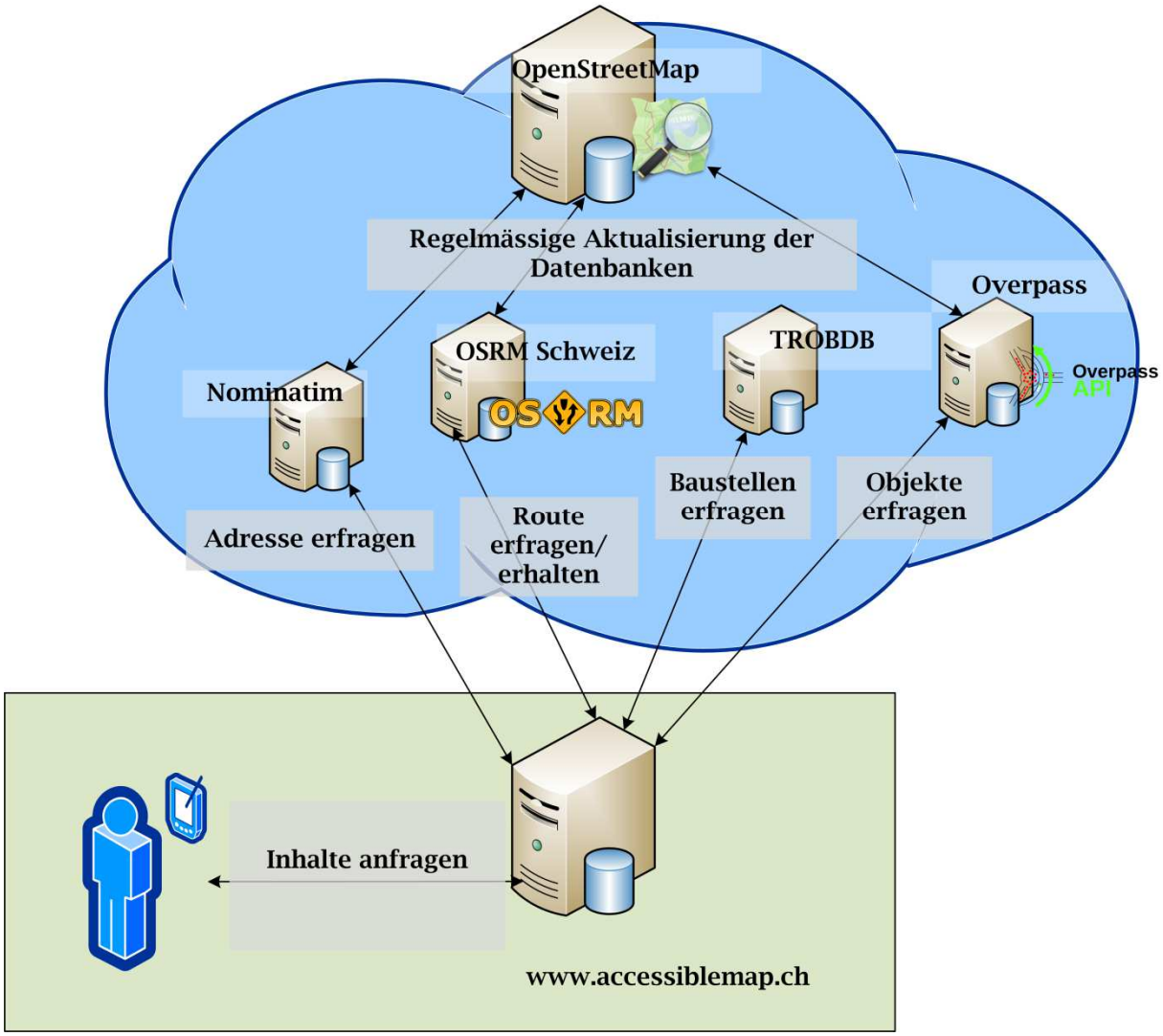


Abbildung 44 - Interaktionsdiagramm

2.2.2.1 Nominatim

Die Adresse des aktuellen Standortes wird mit Hilfe von Nominatim ermittelt. Dazu werden der Breiten- und Längengrad, die vom GPS des benutzten Geräts stammen, dem Server übergeben. Zurück kommt ein Objekt, welches die gesamten vorhandenen Daten zur Position beinhaltet.

Eine typische URL zur Abfrage kann wie folgt aussehen:

http://nominatim.openstreetmap.org/search?q=[47.223741,8.81696]

In diesem Beispiel wird als Ergebnis zurückgeliefert „Bahnhofunterführung, Rapperswil-Jona, Wahlkreis See-Gaster, St. Gallen, 8640, Schweiz“. Daraus werden nun Strassen- und Stadtname sowie Postleitzahl entnommen und in der Anwendung für den Nutzer angezeigt.

2.2.2.2 Overpass

Overpass ist eine API, mit dem selektiv Daten direkt aus OpenStreetMap abgefragt werden können. In der entwickelten Anwendung wird es dazu benötigt, die Strasseninformationen wie Belag und Maximalgeschwindigkeiten, aber auch POIs und die meisten Orientierungspunkte zu erfragen.

Für Strasseninformationen wird dem Server die ID des „Ways“ mitgeteilt, von dem man mehr wissen will. Als Antwort wird ein Objekt zurückgegeben, welches alle „Tags“ des „Ways“ beinhaltet. Ein Beispiel, wie eine solche Abfrage aussehen kann, wird hier gegeben. Die verwendete Way-Id gehört zur Oberseestrasse in Rapperswil.

http://overpass.osm.rambler.ru/cgi/interpreter?data=[out:json];way(35177096);out;

Für die POIs wird zunächst eine „bounding box“ definiert. Dies ist ein Ausschnitt der Karte, der betrachtet werden soll. Die Koordinaten für diese „bounding box“ werden von der entwickelten Anwendung berechnet. Darin werden dann alle POIs mit dem entsprechenden Schlüsselwort gesucht. Eine passende Abfrage dazu wäre:

http://overpass.osm.rambler.ru/cgi/interpreter?data=[out:json];node[amenity=restaurants](47.218678,8.810819,47.227847,8.824319);out;

Die Abfrage retourniert alle Restaurants, welche sich in dieser „bounding box“ befinden.

2.2.2.3 OSRM

Die Angaben zur Route werden aus der Routing API von OSRM extrahiert. Dieser Dienst berechnet aus einer Start- und einer Zielkoordinate eine Route. Das zugrundeliegende Kartenmaterial stammt von OpenStreetMap. Als Resultat erhält man die Route mit Koordinate der Route.

2.2.2.4 TROBDB

Diese API stellt Daten von Baustellen aus der ganzen Schweiz zur Verfügung. Darin eingebunden sind auch die Baustellendaten vom Open Government Data Portal der Stadt Zürich. Die Gültigkeit der Daten wird durch eine stündliche Aktualisierung auf der Datenbank sichergestellt.

Für die Abfrage wird eine Way-Id übergeben.

Beispiel-Link: http://trobdb.hsr.ch/getTrafficObstruction?osmid=177866164

Zurückgeliefert wird ein Objekt, das verschiedene Informationen zur Baustelle beinhaltet. Für die Verwendung in der entwickelten Applikation sind jedoch nur das Datum und das Polygon mit den Koordinaten der Baustelle von Bedeutung. [1]

2.3 Implementation

2.3.1 Technologien

Die Web-Anwendung „Accessible Map“ wurde in der Programmiersprache JavaScript geschrieben.

Für die Darstellung der Mockups in der Elaborationsphase wurde die opensource Animationssoftware Pencil verwendet.

Zu Beginn wurde die Verwendung von „Sencha Touch“ für das User Interface angedacht. Da dieses Framework jedoch eine lange Einarbeitungszeit benötigt, wurde entschieden, stattdessen jQuery mobile zu verwenden. Weiter wurde die Programm-Bibliothek JSTS verwendet. Sie ermöglicht es, „Buffer“ mit beliebigem Umkreis um Punkte, Linien und Polygone zu erstellen. Diese Funktion kam bei Angabe der POOs in einer Strassenseite zum Einsatz, da dort „Buffer“ um ein Strassensegment genutzt wurden.

Als Entwicklungsumgebung war Eclipse im Einsatz. Der Code wurde regelmässig auf das öffentlich zugängliche github-Repository unter https://github.com/grothauser/accessiblemap kopiert. Um die Webseite zu testen wurde xampp lokal installiert und ein Testserver vom Gratisanbieter Square7.ch verwendet.

2.3.2 Testing

Automatisiertes Testing

Der Code wurde mit QUnit getestet. Die Tests wurden in JavaScript geschrieben. Getestet wurden alle mathematischen Berechnungen und das Laden von Daten aus JSON oder GPX-Files (Routing).

In Abbildung 45 sind die gelaufenen Tests zu sehen, die alle erfolgreich bestanden wurden.

Abbildung 45 - Ergebnisse der geschriebenen Tests mit Test Runner

Manuelles Testing

Um bestimmte Testfälle auszuführen mussten manuelle Tests durchgeführt werden. Diese sind in der untenstehenden Tabelle dokumentiert.

Test	Erwartetes Resultat	Eingetretenes Resultat	Massnahme
Routenabfrage auf Autobahn ausgelöst	Es wird keine Route berechnet	Wie erwartet.	Ausgabe für den Benutzer, dass er sich nicht auf Fussgängerterrain befindet.
Route vom selben Punkt aus umgekehrt	Die Route wird umgekehrt angezeigt, Standort wird nicht neu abgerufen	Wie erwartet.	-
Aufruf zur Aktualisierung der Route während man sich am Ziel befindet.	Die Anwendung benachrichtigt den Anwender, dass das Ziel erreicht wurde	Wie erwartet	-
GPS-Position wird falsch ermittelt, so dass die falsche Strassenseite angezeigt wird	Es werden andere Orientierungspunkte angezeigt	Wie erwartet	Der Anwender muss darauf aufmerksam gemacht werden, dass er die angezeigte Seite manuell in der Anwendung wechseln kann.
Der Umkreis wird vom Benutzer geändert	Der Umkreis wird bei der Suche nach POIs entsprechend angepasst	Wie erwartet	-

Tabelle 3 - Testszenarien

2.4 Resultate

Das Resultat der Arbeit stellt den funktionsfähigen Prototyp einer Anwendung für barrierefreies Fussgängerrouting und Standortausgabe dar.

2.4.1 Funktionsumfang

Der entwickelte Prototyp verfügt über die im Folgenden erklärten Funktionalitäten.

2.4.1.1 Standortausgabe

Ausgabe des via GPS lokalisierten Standorts

Dazu wurde der Sensor verwendet, der in jedem Gerät zur Verfügung steht und der mit dem folgenden Befehl aus dem Browser angesteuert werden kann:

```
navigator.geolocation.getCurrentPosition(success, error, options);
```

Manuelle Angabe des Standorts

Die Suchbegriffe werden dem Dienst Nominatim übergeben und die Suchresultate als Auswahlliste (siehe Abbildung 46) zur Verfügung gestellt.

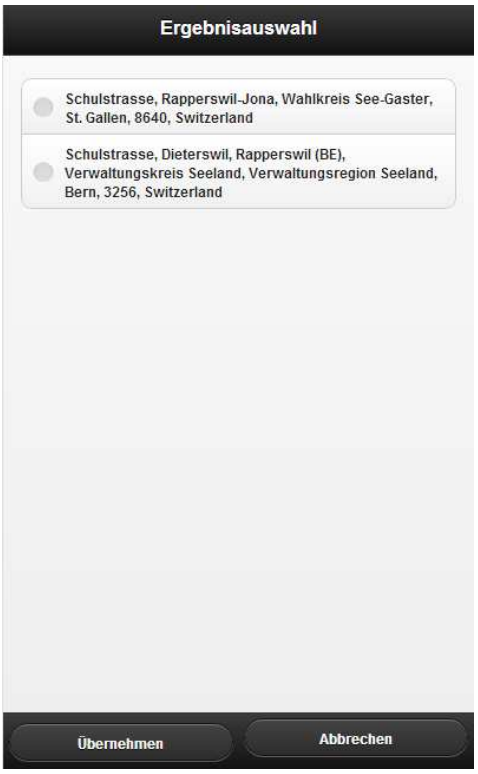


Abbildung 46 - Manuelle Selektierung

Selektierung von gewünschten Orientierungspunkten und POIs

Die Angewählten Checkboxen werden im Localstorage des Browsers vermerkt, so dass sie beim nächsten Besuch der Seite ebenfalls wieder selektiert sind. (siehe Abbildung 46 links)

Ausgabe der selektierten Orientierungspunkte in der Strasse

In der angegebenen Strasse wird nach allen Orientierungspunkten gesucht. Diese werden dann anhand der geografischen Koordinaten und dem gemessenen Kompasswert des Smartphones in „Vor Ihnen“ und „Hinter Ihnen“ unterteilt. Der Abstand berechnet sich ebenfalls aus den Koordinaten der Orientierungspunkte und des aktuellen Standorts. Wurde der Standort manuell eingegeben, so geht die Anwendung vom ersten „Node“ der Strasse aus. (Siehe Abbildung 47 Mitte)

Ausgabe der selektierten POIs in der Umgebung

Im unter Optionen eingestellten Umkreis wird nach Objekten in OpenStreetMap gesucht, welche die passenden „Tags“ aufweisen. Der Abstand und die Richtung werden ebenfalls mittels Kompass und geografischen Koordinaten berechnet. (Siehe Abbildung 47 rechts)

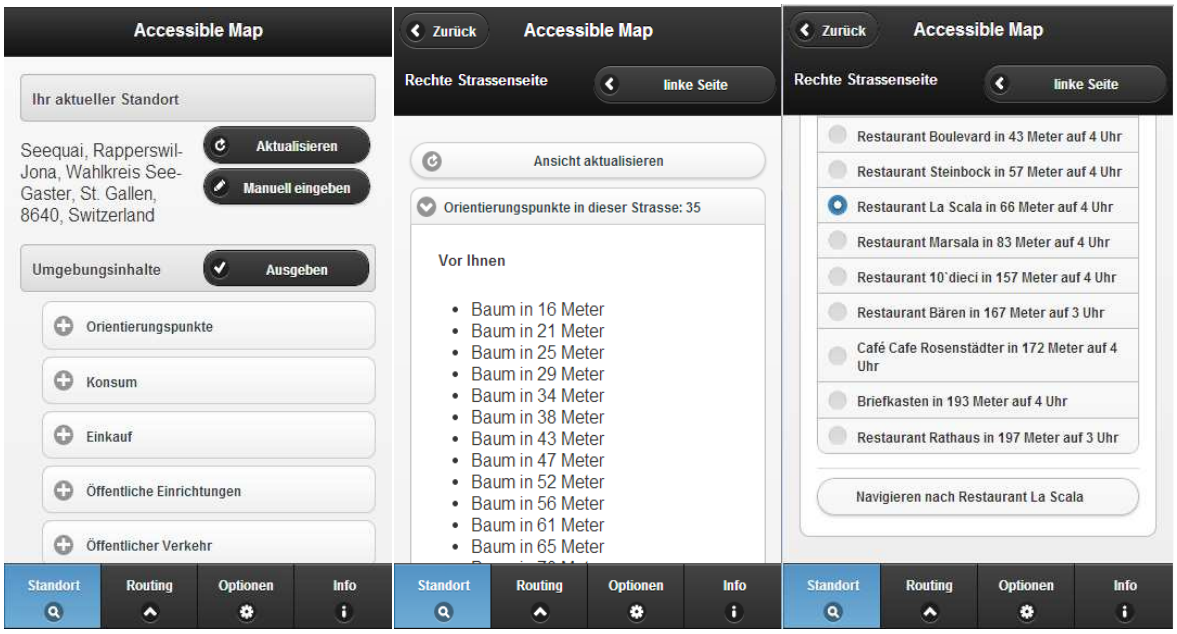


Abbildung 47 - Screenshots Standortausgabe GUI

2.4.1.2 Fussgängerrouting

Navigation zu einem POI

Dem in der Standortansicht angegebenen POI ist eine Koordinate hinterlegt. Wird eine Route generiert so wird der aktuelle Standort als Startpunkt und die Koordinate des POIs als Ziel verwendet. Die Route wird via OSRM mit folgendem Link erfragt:

http://routing.osm.ch/routed-foot/viaroute?loc="+lat1+","+lon1+"&loc="+lat2+","+lon2+"&output=gpx

Als Resultat kommt ein GPX-File zurück. Dies ist ein File im XML-Format, welches die einzelnen Koordinaten der Route beinhaltet. Wie sich die Route daraus berechnet ist in Teil 1 in Kapitel „Berechnung der Anweisungen für das Fussgängerrouting“ erklärt.

Orientierungspunkte in der Route

Als Orientierungspunkte werden in der Route diejenigen gewählt, welche in der Standortausgabe selektiert wurden. Sie werden unterteilt in linke und rechte Strassenseite. Dies geschieht mittels eines „Buffers“, welcher für jede Strassenseite errechnet wird (siehe Kapitel 1.6.4.6). Ausgegeben werden in der Route nur die Orientierungspunkte, welche noch vor einem liegen (siehe Abbildung 48 links). Dies wird ebenfalls anhand eines ausgelesenen Kompasswerts bestimmt.

Kreuzungen werden in der Route als Orientierungspunkte bewusst weggelassen. Der Gedanke war hier, dass die Angabe von Kreuzungen den Anwender verwirren würde, weil er dann nicht weiss, ob er dort abbiegen muss oder nicht. Für das Gebiet der Stadt Zürich werden für die Kategorie Bäume und Abfalleimer die Daten aus dem JSON-File durchsucht, welches vom OGD-Portal bezogen werden kann. Dieses ist sehr gross und bestimmt keine optimale Lösung für eine Web-Anwendung. Besser wäre hier ein API, welches für Koordinaten als Parameter die gewünschten Objekte zurückliefert.

Route aktualisieren

Zurzeit wird beim Anwählen der Navigation zu einem POI eine statische Route generiert. Um herauszufinden ob der Benutzer sich noch auf der Route befindet, kann er die Route aktualisieren lassen. Es wird dann die aktuelle Position via GPS bestimmt und eine neue Route von diesem Punkt aus berechnet. Ist die Distanz zum Ziel kleiner als 3 Meter so wird angegeben, dass man sich bereits beim Ziel befindet. Es wird keine neue Route berechnet.

Route umkehren

Es besteht die Möglichkeit die Route umkehren zu lassen. Dabei wird die Ziel- zur Startkoordinate und umgekehrt. Da eine neue Route beim Routingdienst erfragt wird kann diese Route von der Route beim Hinweg abweichen.

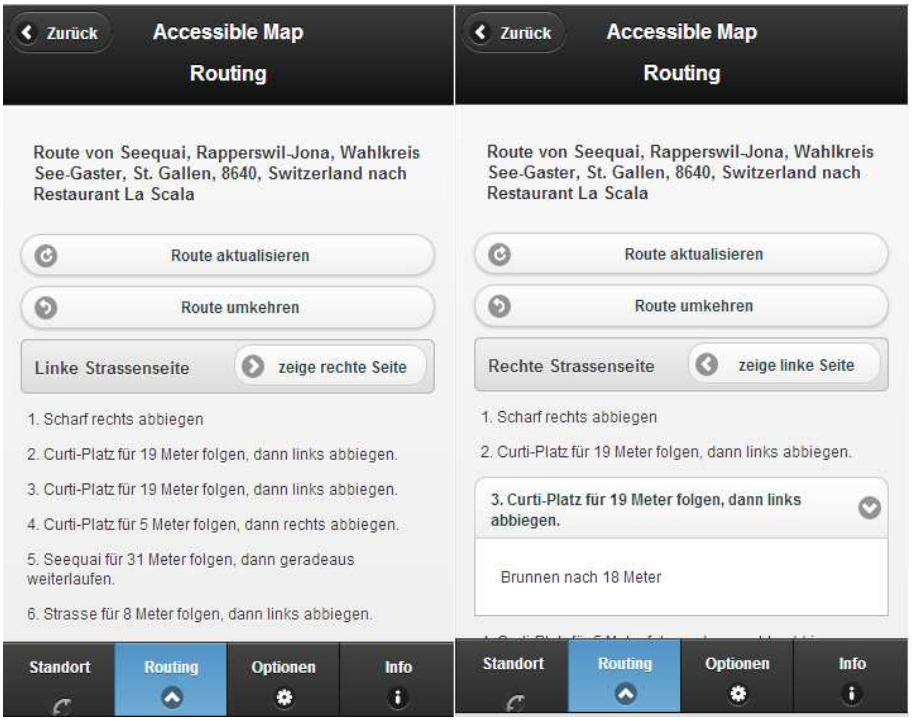


Abbildung 48 - Screenshots Routing GUI

2.4.1.3 Optionen

Umkreisradius einstellen

Es besteht die Möglichkeit den Radius, in dem nach POIs gesucht wird, einzustellen. Dieser kann auf 200, 500, 1000, 1500 oder 2000 Meter eingestellt werden. Standartmässig eingestellt sind 500 Meter. Es wurde dieses Bedienelement (siehe Abbildung 49) ausgewählt, da sich weder ein Slider noch ein inkrementierbares Feld mit „VoiceOver“ gut bedienen liessen.

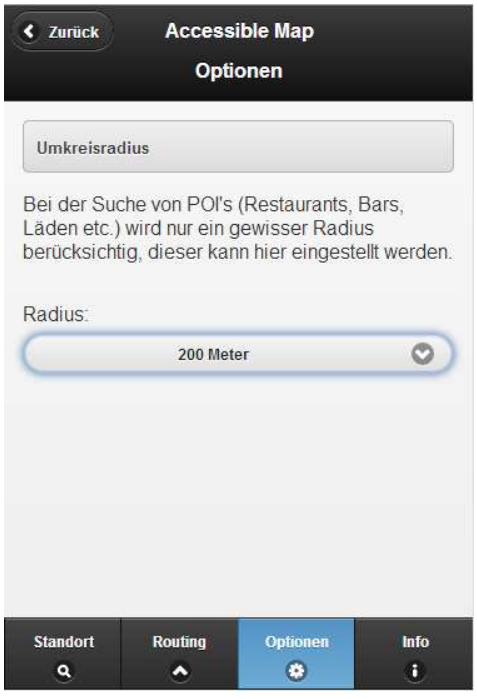


Abbildung 49 - Screenshot Optionen GUI

2.4.1.4 Zielerreichung

Die Bedienung wurde möglichst selbsterklärend realisiert. Die Anforderungen an das Zeitverhalten sowie an die Wartbarkeit und Änderbarkeit wurden eingehalten. Zudem wurden alle in der Anforderungsspezifikation definierten Use Cases in der entwickelten Anwendung realisiert. Es wurden sogar noch zusätzliche Funktionen wie das Umkehren und Aktualisieren einer Route eingebaut. Dennoch bietet die entwickelte Anwendung noch viel Potential für weitere Funktionalitäten. Dies wird im folgenden Kapitel behandelt.

2.4.2 Weiterentwicklung

In einer Weiterentwicklung wichtig werden Verbesserungen der Anwendung im Bereich der Fussgängerstreifen und der Sicherstellung des Einhaltens der Route. Ebenfalls wünschenswert wäre das Einbauen der Funktionen welche die blinden und sehbehinderten Personen der Stiftung „Zugang für alle“ für eine nächste Version der Anwendung gewünscht haben.

2.4.2.1 Automatisches Überprüfen auf Einhaltung der Route

Da in der momentanen Version der Anwendung nur einmal eine statische Route generiert wird, wäre es von grossem Vorteil wenn in einer Weiterentwicklung eingebaut wird, dass die Route sich aktualisiert und dass die Anwendung merkt, wenn der Benutzer sich zu weit von der Route entfernt.

Dies könnte man mittels regelmässigen Abfragen des GPS-Standortes realisieren. Kombinieren könnte man dies indem man die GPS-Geschwindigkeit verwendet und daraus die gelaufenen Meter berechnet. Dadurch könnte eine eventuelle GPS-Ungenauigkeit kompensiert werden. Dies ist gerade im städtischen Gebiet wichtig, da dort die Abweichung zum realen Standort je nach Anzahl sichtbarer GPS-Satelliten sehr gross sein kann.

2.4.2.2 Finden der Fussgängerstreifen auf der Route

In der von OSRM generierten Route werden nur Strassen überquert, welche an diesen Stellen mit Fusswegen überquert werden können. OSRM berücksichtigt keine Strassenübergänge, welche nur Nodes mit dem „Tag“ „crossing“ haben, welches in OpenStreetMap für einen Fussgängerübergang steht. Aus diesem Grund wird im Routingalgorithmus, welcher in der entwickelten Anwendung die textuelle Ausgabe formuliert, dieser Übergang nur als Fussweg erkannt und ausgegeben. Dies ist suboptimal aus dem Grund, dass der blinde oder sehbehinderte selber erkennen muss, dass er sich nun vor einer befahrenen Strasse befindet und diese überqueren soll. Optimal wäre das Erweitern dieser Routingabschnitte um die Information, dass es sich um eine Strassenüberquerung handelt. Dies kann geschehen indem die Umgebung nach Strassen abgesucht wird, welche den aktuellen Routenabschnitt kreuzen oder indem geprüft wird ob sich ein „crossing“-“Node“ darin befindet.

2.4.2.3 Kommentarfunktion

In einem Gespräch bei der Stiftung „Zugang für alle“ hatte eine blinde Person die Idee einer Kommentarfunktion. Diese wäre für feinkörnige Informationen gedacht wie sie auch OpenStreetMap nicht erfasst. Dazu gehören zum Beispiel die Gehsteighöhe oder hervorstehende Briefkästen. Der Anwender könnte dann an einem Punkt in der Route einen Marker setzen, welcher die aktuelle GPS-Position erhält. Dazu könnte er einen Text hinterlassen welcher für alle anderen Anwender genau an dieser Stelle auch angezeigt wird. Eventuell müsste dann aber jemand manuell die Marker nachbearbeiten und an die richtige Stelle setzen da GPS nicht immer punktgenau ist.

2.4.2.4 Manuelle Zieleingabe

Während der Arbeit wurde die Standortausgabe um die Navigation zu den ausgegebenen POIs erweitert. In einer Weiterentwicklung wäre die Möglichkeit einer manuellen Zieleingabe als Adresse wichtig. Hier spielt die Problematik der Suchresultate aus Nominatim eine Rolle. Nicht immer sind Hausnummern in OpenStreetMap eingetragen. Es wäre daher ein anderer Dienst notwendig, welcher die genauen Koordinaten einer gegebenen.

Adresse mit Hausnummer ausgeben kann. Dies wäre auch für die manuelle Standortangabe relevant.

2.4.2.5 Suchen von POIs anhand von Suchbegriffen

Zurzeit werden die POIs anhand von „Tags“ gesucht, so hat ein Restaurant zum Beispiel den „Tag“ „amenity=restaurant“. Wenn der Anwender nun genau weiss wie ein Lokal heisst, aber nicht weiss, ob es als ein Café, Pub, Restaurant oder sogar als Imbiss in OpenStreetMap erfasst ist, so muss er unter Umständen eine lange Liste von Resultaten in der Standortausgabe durchsuchen um das passende Lokal zu finden. Unter Umständen könnte das Lokal auch gar nicht in OSM eingetragen sein. In diesem Fall müsste auf eine andere Datenquelle zurückgegriffen werden. Sofern das Lokal in OSM erfasst ist könnte eine Suchfunktion via Nominatim eingebaut werden und wie bei der manuellen Standortangabe eine Auswahl an Suchergebnissen präsentiert werden.

2.4.2.6 Aufzeichnen einer Route

Dies ist eine Funktion welche sich einer der Betroffenen gewünscht hat. Er hat damit gemeint, dass er vielleicht eine bessere Route zu einem Ziel kennt, als ihm das Navigationssystem oder die Routinganwendung angibt. In diesem Fall könnte er die Route aufnehmen und bei Bedarf abspielen lassen. So könnte er sich unter Umständen auch weniger leicht verlaufen, da er einfach die aufgezeichnete Route zurücklaufen kann. Dies macht auf jeden Fall Sinn, da die Daten in OSM nicht immer vollständig sind und der eine oder andere Fussgängerstreifen oder Fussweg noch fehlt und die von OSRM erhaltene Route einen Umweg nehmen würde.

2.4.2.7 Orientierungspunkte an der Zürcher Stadtgrenze

Um eine bessere Angabe der Orientierungspunkte an der Grenze des Stadtgebietes zu erhalten, müssten alle „Nodes“ der Route, die sich innerhalb Zürichs befinden, einzeln abgespeichert werden. Danach müsste einzeln nach OpenStreetMap-Daten und nach Zürcher-Daten für die jeweiligen Abschnitte gesucht werden.

2.5 Projektmanagement und Projektmonitoring

2.5.1 Projektmanagement

Das Projekt wurde agil durchgeführt. Es wurden jede Woche vor dem Meeting mit dem Betreuer die Ziele für die nächste Woche definiert.

2.5.1.1 Zeitplanung

Zu Beginn der Arbeit wurden 5 Meilensteine als Zeitplan für die Arbeit gesetzt. Der Meilenstein 1 schliesst die Inception-Phase ab. Der Meilenstein 2 schliesst die Elaborationsphase ab und der Meilenstein 3 (bzw. 4) steht für das Ende der Construction-Phase. Nach der Construction-Phase der Arbeit beginnt dann die Transition Phase, in der die Arbeit an den Kunden übergeben wird.

M1 – Analysephase abgeschlossen (06.10.13)

In der Analysephase wurde das Umfeld des Projekts angeschaut. Es wurde geklärt, welche Geräte und Anwendungen blinde und sehbehinderte Menschen nutzen und wie sie sich damit zurechtfinden.

Technologien wie OpenStreetMap und Phonegap, die zum Entwickeln einer mobilen Web Map gebraucht werden, wurden betrachtet und getestet.

M2 – Evaluationsphase abgeschlossen (13.10.13)

Der aktuelle Stand der Technik wurde betrachtet. Vorhandene Produkte, die ähnliche Funktionalitäten aufweisen, wurden mit der zu entwickelnden Applikation verglichen um eine Anwendung mit allen Vorzügen zu entwickeln.

M3 – Implementationsphase abgeschlossen (24.11.13)

Bei der Implementation wurden das Aussehen und die Funktionalitäten der Applikation entwickelt und ausgebaut. Dazu wurden die Aufgaben in Arbeitspakete eingeteilt und unabhängig voneinander abgearbeitet.

M4 – Testphase abgeschlossen (08.12.13)

Um sicherzustellen, dass die Anwendung richtig läuft, wurden verschiedene Tests durchgeführt. Da die Applikation auf verschiedenen Endgeräten laufen soll, müssen die Tests auch auf diesen korrekt durchlaufen.

M5 – Dokumentationsphase abgeschlossen (20.12.13)

Bis zur Abgabe wurde an der Dokumentation des Projekts gearbeitet. Der entstandene Bericht enthält alle wichtigen Errungenschaften, Erkenntnisse und Anstösse für Weiterentwicklungen der Anwendung.

2.5.1.2 Arbeitsaufteilung

Die Arbeitsteilung wurde nach Können und Arbeitsgeschwindigkeit gerichtet. In der Arbeit gab es die zwei Hauptfunktionalitäten Routing und Standortausgabe. Daraus resultierte folgende Aufstellung:

Name	Arbeitspaket
Gwendoline Rothauser	Analyse Fussgänger-Routing Dienste
	Interaktionsdiagramm
	Distanzberechnung mit Koordinaten
	Nächstgelegenes Strassensortiment bestimmen
	Nächstgelegenen Node bestimmen
	Berechnung der Anweisungen für das Fussgänger-Routing
	Ausgabe selektierte POIs in der Umgebung
	Ausgabe des via GPS lokalisierten Standorts
	Speicherung von Einstellungen im Localstorage
	Ausgabe angereicherte Route
	Auslesen des Geräteinternen Kompass
	Manuelle Standorteingabe
	GUI-Gestaltung und Programmierung mit jQuery Mobile
Julia Schmucki	Analyse bestehende Anwendungen
	Use Cases
	Objektdiagramm
	Einpflegen von Open Government Daten
	Einbauen der Baustellendaten via TROBDB
	Zuweisen der Orientierungspunkte in die zwei Strassenseiten
	Bestimmung der Strassenseite
	Ausgabe Orientierungspunkte in Strasse
	Finden von vermerkten und unvermerkten Kreuzungen mit aktueller Strasse
	Gestaltung Logo und Favoriten-Icon für die Webseite

Tabelle 4 - Arbeitsaufteilung

2.5.1.3 Zielerreichung

Als Aufgabenstellung gegeben war eine App zur reinen Standortausgabe. In der Evaluationsphase kristallisierte sich dann nach mehreren Meetings mit Mitgliedern (darunter auch blinde und sehbehinderte Personen) der Stiftung „Zugang für alle“ heraus, dass die Grundidee der Aufgabenstellung nicht wirklich neu und innovativ war. Vielmehr wurde ein brauchbares Werkzeug zur Fussgängernavigation gewünscht. Deshalb verschob sich die Evaluationsphase um mehrere Tage. Am Ende konnte dann zusätzlich zur Aufgabenstellung der Standortausgabe auch noch ein funktionsfähiges Fussgängerrouting erstellt werden. Das Ziel der Arbeit wurde somit erreicht, sogar etwas übertroffen.

2.5.2 Projektmonitoring

Der HTML-Code wurde mit einem HTML-Validator (http://validator.w3.org/) online getestet. Der JavaScript Code wurde mit dem Eclipse-Plugin JSHint validiert. Es wurden stetige Code-Reviews der Änderungen vorgenommen, zeitlich immer dann, wenn eine Funktionalität in das github Repository geschrieben wurde.

2.5.3 Risikomanagement

Zu Beginn des Projektes wurden folgende (siehe Tabelle 5) Risiken erkannt und Massnahmen definiert. Der Schaden wird in Arbeitsstunden gerechnet, da das Projekt keinen Kostenaufwand für den Kunden produziert hat. Der gewichtete Schaden berechnet sich aus der Eintrittswahrscheinlichkeit und dem maximalen Schaden.

Risiko	**Auswirkung**	**Massnahme**	**Kosten der Massnahmen**	**Max. Schaden in Arbeitsstunden**	**Eintrittswahrscheinlichkeit**	**Gewichteter Schaden in Arbeitsstunden**	**Priorität**
Die App wird so entwickelt, dass der Nutzer sie nicht gebrauchen kann.	Die Bewertung der Arbeit ist schlecht, Bachelor-Arbeit muss wiederholt werden.	Testen im Blindenmodus mit (halb-) geschlossenen Augen.	50 h	720 h	5%	36 h	Hoch
Einstellung der verwendeten Dienste von Drittanbietern.	Die Anwendung kann nicht mehr verwendet werden.	Redundanz der Dienste sicherstellen.	150 h	720 h	5%	36 h	Hoch
Teammitglied verliert Daten einer Woche durch Hardwaredefekt	Arbeit ist verloren, muss nachgearbeitet werden.	Eigene Sicherungen des Arbeitsstatus	2 h	128 h	20%	52 h	Gering
Total möglicher Schaden in Arbeitsstunden						**123.2 h**	

Tabelle 5 - Risikomatrix

3 Teil III: Anhang

3.1 Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:

Studentin 1: Name, Unterschrift Studentin 2: Name, Unterschrift

3.2 Glossar

Begriff	Erklärung
POI	Point of Interest Orte, die für den Nutzer einer Karte oder eines Navigationssystems von Bedeutung sein können.
POO	Point of Orientation Orientierungspunkt wie er in der Arbeit verwendet wird. Darunter fallen Bäume, Container, Baustellen, Brunnen, Sitzbänke, Abfalleimer, Kreuzungen, Lichtsignale und Fussgängerstreifen.
YOURS	Yet another OpenStreetMap Routing System Navigationsdienst, der auf OpenStreetMap Daten beruht.
Screen Reader	Bildschirmvorlese-Programm Ein Programm, das die Informationen auf dem Bildschirm eines elektronischen Geräts vorliest, so dass sie auch Blinden und Sehbehinderten zugänglich werden.
OSM	OpenStreetMap Ein freies Projekt, das für jeden frei nutzbare Geodaten sammelt.
OSRM	OpenStreetMap Routing Machine Routing-Dienst basierend auf OpenStreetMap. Nur für die Schweiz.
GPS	Global Positioning System System um den Standort eines GPS-Empfängers anhand von Satellitensignalen zu orten.
Node	Geometrischer Punkt in OpenStreetMap. Er hat geographische Koordinaten und kann entweder nur zu einer Strasse gehören oder ein Punkt von Interesse sein. Diese können ein Baum, ein Restaurant, eine Kreuzung usw. sein.
Way-Id	Eindeutige Identifizierungsnummer für einen „Way“ in OpenStreetMap.

Tag	Kennzeichen, Merkmal Zusatzinformationen zu Wegen, Orientierungspunkten oder POIs. Bsp: Strassenbelag, Höchstgeschwindigkeit, Farbe einer Sitzbank, Name eines Restaurants, …
API	Application Programming Interface (Programmierschnittstelle) Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird
iOS	Betriebssystem für mobile Endgeräte der Firma Apple Inc.

3.3 Abbildungsverzeichnis

Abbildung 1 - Nodes auf Karte 5
Abbildung 2 - Kartenausschnitt OSM Route Seequai nach 10‘ dieci 6
Abbildung 3 - Screenshots Routing 7
Abbildung 4 - Screenshot AmauroMap aus Google-Cache 14
Abbildung 5 - Screenshot Accessibility Guide. 15
Abbildung 6 - Ausschnitt Wheelmap in Rapperswil 16
Abbildung 7 - Screenshots My Position. 17
Abbildung 8 - Screenshot Sendero GPS LookAround 18
Abbildung 9 - Screenshots AroundMe 19
Abbildung 10 - Screenshots von Ariadne GPS 20
Abbildung 11 - Screenshots von BlindSquare 21
Abbildung 12 - Screenshots von Karten-Applikation 22
Abbildung 13 - Screenshots Intersection Explorer. 23
Abbildung 14 - Routenansicht Google Maps 27
Abbildung 15 - Routenansicht OSMR 28
Abbildung 16 - Routenansicht von OSRM 29
Abbildung 17 - Routenansicht von Bing 30
Abbildung 18 – Von Hand erstellte Routenansicht für Mapquest API 31
Abbildung 19 - Routenansicht von YOURS 32
Abbildung 20 - Mockups zur Idee 1 35
Abbildung 21 - Mockups zur Idee 2 36
Abbildung 22 - Mockups zu Idee 3 37
Abbildung 23 – Mockup Hauptbildschirm 38
Abbildung 24 – Mockup Standortausgabe 38
Abbildung 25 - Mockup POIs wählen 39
Abbildung 26 – Mockup Umkreiseinstellungen 39
Abbildung 27 - "bounding box" 40
Abbildung 28 - Darstellung nächstes Strassensegment finden 41
Abbildung 29 - Bounding Box für POI-Suche 42
Abbildung 30 - Berechnung der Seitenbufferkoordinaten 43
Abbildung 31 - Darstellung des Point-In-Polygon Algorithmus 44
Abbildung 32 – Vergleich Kartenausschnitt mit Seitenbuffer und textuelle Beschreibung 45
Abbildung 33 - Vergleich Zürcher Bäume (oben) mit OpenStreetMap Bäumen 46

Abbildung 34 - Kartenausschnitt mit Daten von OpenStreetMap überlagert. 47
Abbildung 35 - Berechnung des Schnittpunktes 48
Abbildung 36 - Darstellung Routinganweisungen 49
Abbildung 37 - Strassenseitenermittlung 50
Abbildung 38 - Kartenansicht Seequai Rapperswil 51
Abbildung 39 - Screenshots Standort und Rundumsicht Seequai Rapperswil 52
Abbildung 40 - Screenshots Routing 52
Abbildung 41 - Suchresultate aus OpenStreetMap für die Oberseestrasse in Rapperswil 53
Abbildung 42 - Use Case Diagramm 56
Abbildung 43 - Objektdiagramm 58
Abbildung 44 - Interaktionsdiagramm 61
Abbildung 45 - Ergebnisse der geschriebenen Tests mit Test Runner 64
Abbildung 46 - Manuelle Selektierung 66
Abbildung 47 - Screenshots Standortausgabe GUI 67
Abbildung 48 - Screenshots Routing GUI 69
Abbildung 49 - Screenshot Optionen GUI 69

Tabelle 1 - Vergleich der untersuchten iPhone Applikationen 24
Tabelle 2 - Bewertung Routingdienste Route 33
Tabelle 3 - Testszenarien 65
Tabelle 4 - Arbeitsaufteilung 74
Tabelle 5 - Risikomatrix 76

3.4 Literaturverzeichnis

[1] „AmauroMap,“ [Online]. Available: http://deutsch.ceit.at/ceit-alanova/referenzprojekte-alanova/projekte/amauromap. [Zugriff am 26 11 2013].

[2] U. Heidelberg, „Routing der Universität Heidelberg,“ Universität Heidelberg, [Online]. Available: http://openrouteservice.org/. [Zugriff am 12 15 2013].

[3] Microsoft, „Bing Maps,“ Microsoft, [Online]. Available: http://www.bing.com/maps/ . [Zugriff am 15 12 2013].

[4] „MapQuest,“ MapQuest, Inc., [Online]. Available: http://open.mapquestapi.com . [Zugriff am 18 12 2013].

[5] „Yours Navigation,“ [Online]. Available: http://www.yournavigation.org/ . [Zugriff am 18 12 2013].

[6] C. D. GmbH, „CloudMade,“ CloudMade Deutschland GmbH, [Online]. Available: http://cloudmade.com/. [Zugriff am 18 12 2013].

[7] „Lists OpenStreetMap,“ OpenStreetMap, [Online]. Available: https://lists.openstreetmap.org/pipermail/newbies/2008-May/001404.html. [Zugriff am 12 19 2013].

[8] „Movable Type Scripts,“ [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html. [Zugriff am 13 12 2013].

[9] „PNPOLY - Point Inclusion in Polygon Test,“ [Online]. Available: http://www.ecse.rpi.edu/Homepages/wrf/Research/Short_Notes/pnpoly.html.

[10] „TROBDB,“ [Online]. Available: http://trobdb.hsr.ch/.

[11] „Züriplan,“ [Online]. Available: http://www.stadtplan.stadt-zuerich.ch/zueriplan/stadtplan.aspx.

[12] „Wolfram MathWorld,“ [Online]. Available: http://mathworld.wolfram.com/Line-LineIntersection.html.

3.5 Persönliche Berichte und Dank

3.5.1 Dank

Der Dank geht an die Stiftung „Zugang für alle“ aus Zürich. Ohne die aus den Gesprächen gewonnen Erkenntnisse wäre die Web-Anwendung nicht so innovativ geworden. Wir hoffen, dass die Anwendung Anklang findet und eine Weiterentwicklung, sei es durch eine andere Instanz oder durch die Studentinnen selber, eintreffen wird. Ein Dank geht auch an die Mitarbeiter der Stadt Zürich, die für Fragen immer zur Verfügung standen. Auch den Korrekturlesern Martin Rothauser und Josua Schmid sei gedankt.

3.5.2 Persönlicher Bericht Gwendoline Rothauser

Die Zusammenarbeit funktionierte gut. Das unterschiedliche Vorgehen zur Problemlösung bereitete mir manchmal etwas Kopfzerbrechen. Durch genaue Absprachen konnte dies jedoch gelindert werden. Die Thematik fand ich wirklich sehr interessant und auch die Zusammenarbeit mit der Stiftung bedeutete mir viel. Das Projekt war in der Hinsicht eine grosse Herausforderung, dass es besonders viel Einfühlungs- und Vorstellungsvermögen verlangte.

Ich habe das Gefühl, dass ich etwas Gutes tun konnte und dass diese Arbeit nach Abschluss der Bachelorarbeit weiterverwendet werden kann. Gerne hätte ich noch mehr Funktionalitäten eingebaut, es war jedoch im Vorhinein klar, dass wir nicht alles Mögliche in der gegebenen Zeit realisieren können. Ich würde mich auch freuen, wenn ich dieses Projekt weiterhin im Auge behalten, wenn nicht sogar weiterentwickeln darf. Auch nach der Implementierungsphase kamen uns noch viele Ideen, die man für eine nächste Version verwenden könnte.

3.5.3 Persönlicher Bericht Julia Schmucki

Als JavaScript-Neuling hätte ich nie gedacht, dass in dieser Arbeit ein so umfangreicher Prototyp entstehen wird. Ich konnte mir sehr viel Wissen über Web-Applikationen und Webdienste aneignen. OpenStreetMap habe ich als mächtiges Werkzeug kennengelernt.

Die Arbeit hat mir sehr gefallen, einerseits wegen der Verwendung von Kartenmaterial, mit dem man ein wenig experimentieren konnte, andererseits wegen den Menschen, denen wir mit der entwickelten Webapplikation helfen können. Es freut mich, einen gemeinnützigen Beitrag an die Behindertengemeinde leisten zu können. Den direkten Kontakt mit der Stiftung „Zugang für alle“, die als unser Kunde fungierte, kann ich nur positiv werten. Es konnten echte Informationen gesammelt und in der Applikation umgesetzt werden. Diese Erfahrung konnte ich bisher nicht machen und hat mich deshalb weiter gebracht.

Ich hoffe, dass sich die Nachhaltigkeit dieses Projekts noch zeigen wird und ich die Anwendung noch weiter verbessern und anpassen kann.

3.6 Inhaltsverzeichnis der CD

- Bericht als Word- und PDF-Datei
- Bedienungsanleitung
- Web-Anwendung als Zip-Ordner
- Logbuch (Wochenjournal)
- Poster