

Todo list

„Webanwendungen werden immer beliebter“ -> „deshalb habe ich auch eine erstellt“ ist für mich kein gutes Argument. Besser: 1. Was ist das Problem? (Fahrwege, Mieten) 2. Wie soll das Problem gelöst werden? (Anwendung) 3. Details (Webanwendungen werden beliebter wegen folgenden Vorteilen (Quelle), diese vorteile treffen hier auch zu 2

Bell

**Thema: Webanwendung zum Finden eines optimalen Standortes in
Abhängigkeit vom Weg mithilfe von Online-Kartendiensten**

Gernot Zacharias

13. Februar 2019

1 Inhaltsverzeichnis

2 Einleitung

Durch den aufstrebenden Online-Versand-Handel, wie zum Beispiel durch Onlineversandapotheken, entstehen neue Probleme. Die Versandhändler benötigen Lagerhäuser für ihre Waren und die Fahrwege vom Lager zum Kunden sollten möglichst minimal gehalten werden, um Zeit und Geld zu sparen. Auch für Privatanwender kann es interessant sein die Fahrwege zu minimieren, um zum Beispiel den Weg auf Arbeit kurz zu halten, oder um einen Wohnort mit niedrigen Mieten zu wählen. Somit lässt sich auch der Treibstoff Verbrauch reduzieren, was dem Geldbeutel und der Umwelt zu Gute kommt. Dieses Problem lässt sich durch eine Anwendung lösen, die aus der Häufigkeit der angesteuerten Orte eine optimale Position bestimmt mit möglichst kurzen Fahrwegen. Diese Anwendung kann zum Beispiel auf Basis der Kartendienste von Google Maps und Open Street Map realisiert werden. Ich habe mich für eine Webanwendung entschieden, da Webanwendungen plattformunabhängig sind und keine Installation von zusätzlicher Software nötig ist, außer einem aktuellen Webbrowser.

„Webanwendungen werden immer beliebter“ -> „deshalb habe ich auch eine erstellt“ ist für mich kein gutes Argument. Besser: 1. Was ist das Problem? (Fahrwege, Mieten) 2. Wie soll das Problem gelöst werden? (Anwendung) 3. Details (Webanwendungen werden beliebter wegen folgenden Vorteilen (Quelle), diese vorteile treffen hier auch zu

2.1 Problemstellung

Der Online Versandhandel ist ein zurzeit stark wachsender Sektor der Wirtschaft. Deshalb verwundert es auch nicht, dass es heutzutage auch schon Online-Versand-Apotheken gibt, welche einem direkt die Medikamente an die Tür liefern. Dies ist vor allem für bewegungseingeschränkte Personen interessant, da diese sich nun nicht mehr zu nächsten Apotheke bewegen müssen. Auch ermöglicht es Personen die nicht in der Nähe einer Apotheke wohnen, sich die benötigten Medikamente zu besorgen. Für die Anbieter dieser Versandapotheken ist es nun interessant herauszufinden, wie man Zeit und Geld beim Versand einsparen kann. Dazu kann man zum Beispiel die Warenlager dort platzieren, wo die Nachfrage am größten ist.

2.2 Grundlagen

2.2.1 World Wide Web

(Geschichte WWW) In der heutigen Zeit gewinnt das World Wide Web (WWW) immer mehr an Bedeutung und ist fast schon nicht mehr wegzudenken. Das World Wide Web ist der Teil des Internets, in dem die Daten mithilfe des HTTP/HTTPS -Protokolls ausgetauscht werden, wobei ein Computer, der Server mit der entsprechenden Webseite, die Daten auf Anfrage des Clients zu diesem schickt. Das WWW umfasst also fast alle öffentlich zugänglichen Webseiten. Das World Wide Web (Berners-Lee u. a., 1992)

2.2.2 HTML

Die Hyper Text Markup Language (HTML) ist die benutzte Sprache im WWW und beschreibt eine Textdatei, in welcher mit einer bestimmten Syntax geschrieben werden muss. Diese Textdatei wird dann von einem Browser analysiert und zeigt dem Endnutzer den Text mit entsprechender Formatierung. Reine HTML Webseiten besitzen keine dynamischen Elemente und bieten dem Nutzer wenig Interaktionsmöglichkeiten.

2.2.3 JavaScript

JavaScript ist eine Scriptsprache welche direkt in eine HTML Datei implementiert werden kann und auch vom Browser entsprechend verarbeitet wird und dynamische Webseiten erlaubt. JavaScript wird über den `<script>...</script>` -tag in die HTML Datei eingebunden.

2.2.4 Google Maps

Google Maps ist der bekannteste Online Kartendienst und bietet viele Möglichkeiten zum erstellen von dynamischen Kartendarstellungen auf der eigenen Webseite. Allerdings benötigt man zur Nutzung der Google Maps API einen benutzerbezogenen Key, welchen man allerdings nur gegen Angabe von Kreditkarten Informationen erstellen kann.

2.2.5 Open Street Map

Open Street Map spielt im Vergleich mit Google Maps eine untergeordnete Rolle. Dafür bietet Open Street Map freies Kartenmaterial an und es steht eine große Menge an freien API's zu Verfügung. In meinem Prototypen setze ich auf Leaflet, welches die Kartendaten über eine externe Webseite bezieht, welche das Kartenmaterial von Open Street Map als .png zur Verfügung stellt.

2.2.6 Koordinatenberechnung

Koordinaten Längengrad und Breitengrad:

wgs84 ?

<https://www.linz.govt.nz/data/geodetic-system/datums-projections-and-heights/geodetic-datums/world-geodetic-system-1984-wgs84>

<https://www.kompf.de/gps/distcalc.html>

3 Existierende Lösungen

was gibt es für Quellen oder ähnliche Implementierungen

4 Zielstellung

Mein Ziel ist es eine Anwendung zu kreieren, welche aus der Häufigkeit des Ansteuerns von bestimmten Orten, einen Ort bestimmt, von welchem aus die Gesamtdistanz, in Abhängigkeit von der Häufigkeit, zu den gegebenen Orten möglichst klein ist.

Gesucht ist eine Anwendung, die diesen kürzesten Gesamtweg (oder diese Zeit) minimiert. Wenn JavaScript: Interface mit Google Maps, keine Installation nötig, Webanwendungen auf dem Vormarsch. . . Wenn andere Programmiersprache, auch da kurz schreiben warum. (zum Beispiel in Schule beigebracht)

5 Algorithmus

Die Anwendung berechnet den Mittelpunkt anhand der Häufigkeit, wie oft man den Ort pro Woche ansteuert. Der Benutzer setzt dazu Marker, welche am Anfang die Häufigkeit von 5x/Woche aufweisen, auf die Karte. Während des Setzens der Marker wird bereits ein Mittelpunkt berechnet. Durch einen Klick auf den Marker erscheint ein Popup in welchem man die Häufigkeit ändern kann. Danach wird die Position des Mittelpunktes erneut berechnet. Der Mittelpunkt wird durch drei Kreise dargestellt, einen inneren roten, einen mittleren gelben und durch einen grünen äußeren Kreis.

für jedes Ziel weg pro Woche (oder pro Tag...) ausrechnen; oder Zeit minimieren, es können ja unterschiedliche Fortbewegungsarten sein erstmal aber Weg minimieren, wenn Zeit ist dann Optionen wie Spritverbrauch, Zeit, eigene Priorität; erstmal Luftlinie baseline; optional: kürzester Weg; Punkte finden, an dem sich das Summe minimiert; ist das gleich dem Mittelpunkt, wenn man die Strecken entsprechend dem durchschnittlichen Weg verlängert oder kürzt; für Leipzig oder überall? bei Leipzig könnte man konstanten Umrechnungsfaktor zwischen Längen/Breitengradunterschieden und Metern/Kilometern wählen (kurz sagen wie groß der Fehler ist), ansonsten braucht man Formel; wenn global, problematisch, wenn Wohnort nicht zugänglich ist;

6 Implementierung

Für die Kartendarstellung nutze ich Leaflet (Crickard III, 2014), eine OpenSource Bibliothek für die Darstellung von Kartenmaterial aus verschiedenen Quellen. Eine Kartendatenquelle, die auch ich benutzt habe, ist OpenStreetMap¹. Für die Webanwendung ist nur ein halbwegs aktueller Webbrowser notwendig. Die Webanwendung besteht im wesentlichen bloß aus einem HTML Dokument und einer Javascript Datei sowie der Leaflet Bibliothek.

```
...
<link rel="stylesheet" href="leaflet/leaflet.css" crossorigin=""/>
//Einbetten der CSS Datei von Leaflet
<script src="leaflet/leaflet.js"></script>
//Einbettung der Leaflet Bibliothek
...
<div id="map"></div> //Kartencontainer
<script src="js/leafscript1.js" type = "text/javascript"></script>
//Einbettung des selbstgeschriebenen Scripts
...
```

Hier werden dann die technischen Details der Implementierung und gegebenenfalls der Algorithmen beschrieben. Verwendete Technologien, Lizenz, Systemvoraussetzungen und so weiter.; Aufpassen: Fehlerquelle sinus/cosinus Bogenmaß oder Gradmaß;

¹<https://www.openstreetmap.org>

7 Schlussfolgerung

7.1 Auswertung

wieviel bringt das wirklich

7.2 Ausblick

Momentan berechnet die Anwendung die Entfernungen nur per Luftlinie und lässt Straßen und natürliche Hindernisse außer Acht. Auch basiert die Berechnung bisher nur auf Linien in der Ebene, aber die Erde ist ein drei dimensionales Objekt und der Abstand zwischen zwei Punkten müsste durch einen Kreisbogen beschrieben werden. Somit besteht noch viel Spielraum für die Erweiterung der Anwendung, zum Beispiel Integration von einem Wegfindalgorithmus, einer veränderten Distanzrechnung, Zeit, statt Distanz; nicht Luftlinie, sondern kürzesten Weg (über Google Maps API); global, statt Leipzig; Maximallänge für Wege

7.3 Abkürzungsverzeichnis

API	Application programming Interface
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JS	JavaScript
PNG	Portable Network Graphic
WGS84	52
WWW	World Wide Web

Literaturverzeichnis

[Berners-Lee u. a. 1992] BERNERS-LEE, Tim ; CAILLIAU, Robert ; GROFF, Jean-François ; POLLERMANN, Bernd: World-Wide Web: The information universe. In: *Internet Research* 2 (1992), Nr. 1, S. 52–58

[Crickard III 2014] CRICKARD III, Paul: *Leaflet. js Essentials*. Packt Publishing Ltd, 2014