

Arbeitsplan

– spe-16 –

Inhaltsverzeichnis:

1. Projektvision.....	S. 2
2. Voraussetzungen.....	S. 2
3. Designübersicht und Funktionalität.....	S. 2
3.1. Designvorstellungen.....	S. 2
3.2. Funktionale Anforderungen.....	S. 3
3.3. Nichtfunktionale Anforderungen.....	S. 4
4. Arbeitspakete.....	S. 5
5. Vorprojekt.....	S. 6
6. Glossar.....	S. 7

1. Projektvision

Mit dem Projekt SPE-16 wird dem BVL eine barrierefreie Webanwendung zur Verfügung gestellt, die es Nutzern ermöglicht, schnell und einfach diejenigen Orte innerhalb der Stadt Leipzig zu finden, deren Erreichbarkeit und Einrichtung mit ihren Bedürfnissen übereinstimmen.

Als Quelle dient eine RDF-Datei, welche die vom BVL zusammengetragenen Informationen von rund 1800 Objekten im Stadtgebiet Leipzig enthält.

Nachdem der Nutzer die Kriterien über ein Menü ausgewählt hat, werden die Objekte entsprechend gefiltert und passende Ergebnisse auf einer eingebundenen Stadtkarte markiert.

Zugunsten der Übersichtlichkeit wird ein bestimmter Umkreis ausgehend vom Standort des Nutzers festgelegt. Ändert der Nutzer diesen Ansichtsbereich, werden die Ergebnisse entsprechend angepasst.

Durch die Auswahl einer Markierung können weitere Informationen über das Objekt eingeholt werden.

2. Voraussetzungen

Da die Anwendung in JavaScript geschrieben wird und auch alle verwendeten Frameworks in JavaScript geschrieben sind, kann die Anwendung nur in aktuellen Browsern ausgeführt werden. Das führt mit sich, dass ein Anwender ein entsprechendes Gerät zur Verfügung haben muss um den Browser und damit die Anwendung ausführen zu können. Da noch nicht abzuschätzen ist wie leistungsaufwendig die Anwendung wird, können auch noch keine Systemanforderungen für ein solches Gerät genannt werden, wobei das Ziel darin liegt die Anwendung, zumindest auf den aktuellsten Smartphones und durchschnittlichen Home-PCs, lauffähig zu machen. Darüber hinaus benötigt das Gerät einen Internetzugang, mit genügend Bandbreite um in annehmbarer Zeit die Anwendung, sowie Kartenmaterial herunterzuladen. Ein Webserver muss die Anwendung zur Verfügung stellen.

3. Designübersicht und Funktionalität

3.1. Designvorstellungen

Bei Programmaufruf werden zunächst die Informationen aus der auf dem Server hinterlegten RDF-Datei sortiert und bereitgestellt. Für den Nutzer wird eine Oberfläche eingeblendet, über die er Objekte nach verschiedenen Kategorien – bezeichnet durch verständliche Piktogramme – filtern

kann. Sobald dies geschehen ist, werden die Ergebnisse in einem bestimmten Umkreis auf einer Karte markiert. Über einen Click auf die Markierung werden zusätzliche Informationen zu dem Objekt eingeblendet. Die Oberfläche wird bewusst einfach und übersichtlich gehalten. Da die Barrierefreiheit ein zentrales Konzept des Projekts ist, wird bei der Darstellung auf entsprechend starke Kontraste und eine angepasste Schriftgrösse geachtet.

3.2. Funktionale Anforderungen

Muss-Ziele

/FA 10/

Die hinterlegte RDF-Datei wird extrahiert und in eine Liste passender Objekte überführt

/FA 20/

Über einen Filter kann der Nutzer nach Objekten mit bestimmten Eigenschaften suchen

/FA 30/

Die Objekte werden je nach Ansichtsbereich des Nutzers gefiltert

/FA 40/

Die Filterergebnisse passen sich interaktiv zu Veränderungen des Ansichtsbereichs an

/FA 50/

Die Ergebnisse einer Anfrage werden auf einer eingebundenen Karte markiert

/FA 60/

Die Geo-Location des Nutzers wird automatisch ermittelt

/FA 70/

Der Nutzer kann sich zusätzliche Informationen zu einem einzelnen Ergebnis anzeigen lassen

Kann-Ziele

/FA 10 op/

Filtermöglichkeiten können auf Grundlage des Datensatzes dynamisch angepasst werden

/FA 20 op/

Der Nutzer kann entscheiden, ob der Ausgangspunkt die aktuelle Position oder ein beliebiger Ort sein soll

/FA 30 op/

Der Nutzer kann zwischen Karte und Liste als Anzeige wählen

/FA 40 op/

Es wird zwischen zugänglichen und eingeschränkt zugänglichen Objekten unterschieden (Nutzer kann zwischen optionalen und obligatorischen Merkmalen wählen)

/FA 50 op/

Das Produkt kann eine geeignete Route zu einem Objekt ermitteln

/FA 60 op/

Für bestimmte Kategorien ist eine Schnellsuche verfügbar

3.3. Nichtfunktionale Anforderungen

/NF 10/

Das Produkt ist auf jedem modernen Browser lauffähig

/NF 20/

Das Produkt ist auch auf kleinen Displays übersichtlich und gut bedienbar

/NF 30/

Das Produkt kommt mit Ausnahme vom Programmstart ohne längere Ladezeiten aus

/NF 40/

Das Produkt kann ohne Verzögerungen von wenigen Tausend Menschen simultan genutzt werden

/NF 50/

Das Produkt erfüllt die Kriterien zur barrierefreien Nutzung

4. Arbeitspakete

Vorprojekt:

In dem Vorprojekt soll in erster Linie ein Einstieg im Umgang mit der RDF-Datei vollzogen werden. Da eine ordentliche Bereitstellung und Sortierung der Informationen für die weiteren Funktionen und deren Performanz grundlegend ist, wird diesem Abschnitt besonders viel Aufmerksamkeit gewidmet. Desweiteren soll ein Prototyp zur Filterung von Objekten und eine entsprechende Ausgabe implementiert werden.

geschätzter Aufwand: 40%

3. Paket:

Im Zuge des dritten Paketes beschäftigt sich das Team mit der Fertigstellung und Erweiterung der Punkte aus dem Vorprojekt. Die Anfragemöglichkeiten sollen ausgebaut und verbessert werden. Zudem soll eine automatische Ermittlung des Nutzer-Standorts erfolgen.

geschätzter Aufwand: 10%

4. Paket:

Das vierte Paket dient der Optimierung der Datenverarbeitung. Dazu wird der Objektfilter auf einen bestimmten Umkreis des Nutzers angepasst. Das Produkt soll dynamischer und übersichtlicher werden.

geschätzter Aufwand: 15%

5. Paket:

Es wird an einer interaktiven Anpassung der Filterergebnisse gearbeitet. Zudem soll die Anzeige von zusätzlichen Informationen zu einem einzelnen Ergebnis implementiert werden. Das Design wird optimiert.

geschätzter Aufwand: 15%

6. Paket:

Dieses Paket widmet sich den Nichtfunktionalen Anforderungen, sofern diese noch nicht erreicht wurden. Auf Wunsch können optionale Funktionalitäten implementiert werden.

geschätzter Aufwand: 10%

7. Paket

Verfeinerung des Produkts, Abschluss und finale Tests

geschätzter Aufwand: 10%

5. Vorprojekt

Im Vorprojekt liegt das Hauptaugenmerk auf der Erstellung eines Datenmodells [FA-10]. Darin ist das Einlesen der RDF-Datenquelle, das Übertragen in eine Datenhaltung und die Bereitstellung für den Rest der Anwendung enthalten. Es wird von Anfang an eine gewisse Darstellung der Daten erzeugt, welche aber nur bei genügend verfügbarer Arbeitszeit so ausgebaut wird, dass ein Filtern [FA-20] und anzeigen der Daten auf einer Karte [FA-50] möglich ist.

Da der Umgang mit einer RDF-Datenquelle für alle Teammitglieder ein unbekanntes Gebiet ist, birgt sich darin eine große potentielle Fehlerquelle. Um dem entgegen zu wirken, konzentriert sich das Team im Vorprojekt auf den Verarbeitungsprozess der Daten, sodass nach dem Vorprojekt ein zuverlässiges Datenmodell zur Verfügung steht, auf welchem der Rest aller Funktionalitäten aufgebaut wird.

Dadurch, dass die Hauptfunktion der Anwendung im Bereitstellen von Daten liegt, kann das Abschließen des Vorprojekts mit einem funktionierenden Datenmodell, auch ohne Kartenansicht und Filterfunktion, als Demonstration der Hauptfunktion angesehen werden. Ergibt sich ein zügiges Voranschreiten mit der Funktionalität des Datenmodells, so werden Prototypen der Filterfunktion und Kartenansicht bereits im Vorprojekt geliefert. Diese können später je nach Entwicklung erweitert werden, aber das Datenmodell wird weitestgehend unverändert bleiben.

6. Glossar

6.1. RDF

RDF (Resource Description Framework) ist ein Standardmodell zur Formulierung von logischen Aussagen über beliebige Ressourcen. Das RDF Modell beschreibt den Zusammenhang von Ressourcen in einem 3-Tupel dem Subjekt Prädikat und Objekt. RDF wird im Projekt verwendet, da die Quelldaten in dieser Form beschrieben werden.

6.2. N-Triples-Language

N-Triples Language ist ein Format zur schriftlichen Kodifizierung von RDF Graphen in zeilenbasierte und schriftliche Texte. Das Format wird im Projekt verwendet um die RDF Quelldaten zu beschreiben.

6.3. W3C

World Wide Web Consortium(W3C) ist eine internationale Community, die öffentliche Webstandards entwickelt und dadurch für ein langzeitliches Wachstum des Internets sorgt.

6.4. Mash-Up

Mash-Up bedeutet das Erstellen und Vermischen von bereits bestehenden Webanwendungen. Mash-Up verwenden offene Programmierschnittstellen um Inhalte, Bilder, Töne oder Videos auf Websites neu zu kombinieren. Man verwendet im Projekt Mash-Up zur Kombination von Kartendaten mit Objektdaten.

6.5. OpenStreetMap

OpenStreetMap ist ein internationales Projekt um eine freie Weltkarten für jedermann zu erstellen. OpenStreetMap stellt dabei die gesammelten Daten in Rohform oder in vorberechneten Karten zur Verfügung. Die im Projekt SPE16 verfügbaren Daten werden auf einer OpenStreetMap dargestellt werden.

6.6. BVL

Der Behindertenverband Leipzig e.V. ist ein Interessenvertreter für behinderte Menschen in der Region Leipzig. Durch die angebotenen Leistungen werden Menschen mit Behinderungen im täglichen Leben unterstützt.

6.7. Barrierefreiheit

Eine Website ist barrierefrei, wenn sie neben einer einfachen und universellen technischen Bedienbarkeit auch klar und einfach strukturiert ist und bestimmte optische Kriterien erfüllt. Im Vordergrund steht die Gleichberechtigung aller Nutzer unabhängig von deren Einschränkungen oder Bildungsstand.

6.8. N3.js

Ist eine Bibliothek, die das Handling mit RDF (Resource Description Framework) in JavaScript erleichtert. Sowohl im Browser als auch in Node.js (browserlos).

6.9. Leaflet

Leaflet ist eine OpenSource JavaScript Bibliothek. Mit ihr ist es möglich interaktive OpenStreetMap Karten in Internetseiten einzubetten und auf diesen Karten Marker zu setzen. Die Marker können mit HTML Code versehen werden, sodass ein individuelles Pop-up-Fenster zu jedem Marker möglich ist.

6.10. Single-Page-Application

Bei einer Single-Page-Application (engl. Einzelseiten-Anwendung) handelt es sich um eine Webanwendung, die aus einem einzigen HTML-Dokument besteht und deren Inhalte dynamisch nachgeladen werden. Das Nachladen solcher Inhalte kann über die Verbindung zu einem Server, oder durch lokale Generierung der Inhalte geschehen. Dazu wird ein Programm vom Browser ausgeführt, welches mit dem HTML Dokument heruntergeladen wird. Meist ist dies ein JavaScript Programm. Durch dynamische Veränderung des HTML/CSS Codes dient dann dieses eine HTML Dokument als Benutzeroberfläche.

Bei dieser Art der Web-Architektur wird die Serverlast reduziert und der Client übernimmt einen Teil, oder alle der Berechnungen. Dadurch können Webanwendungen ohne optische Flussunterbrechung geschaffen werden, welche den Eindruck einer vollwertigen Client Anwendung (bzw. „App“) vermitteln.