

Arbeitsplan

Erstellt am:

11.01.2017

Erstellt von:

Bjarne Dasenbrook
Mohammed Kheir Alawa Kaviar
Robert Mildenberger
Jakob Rieke

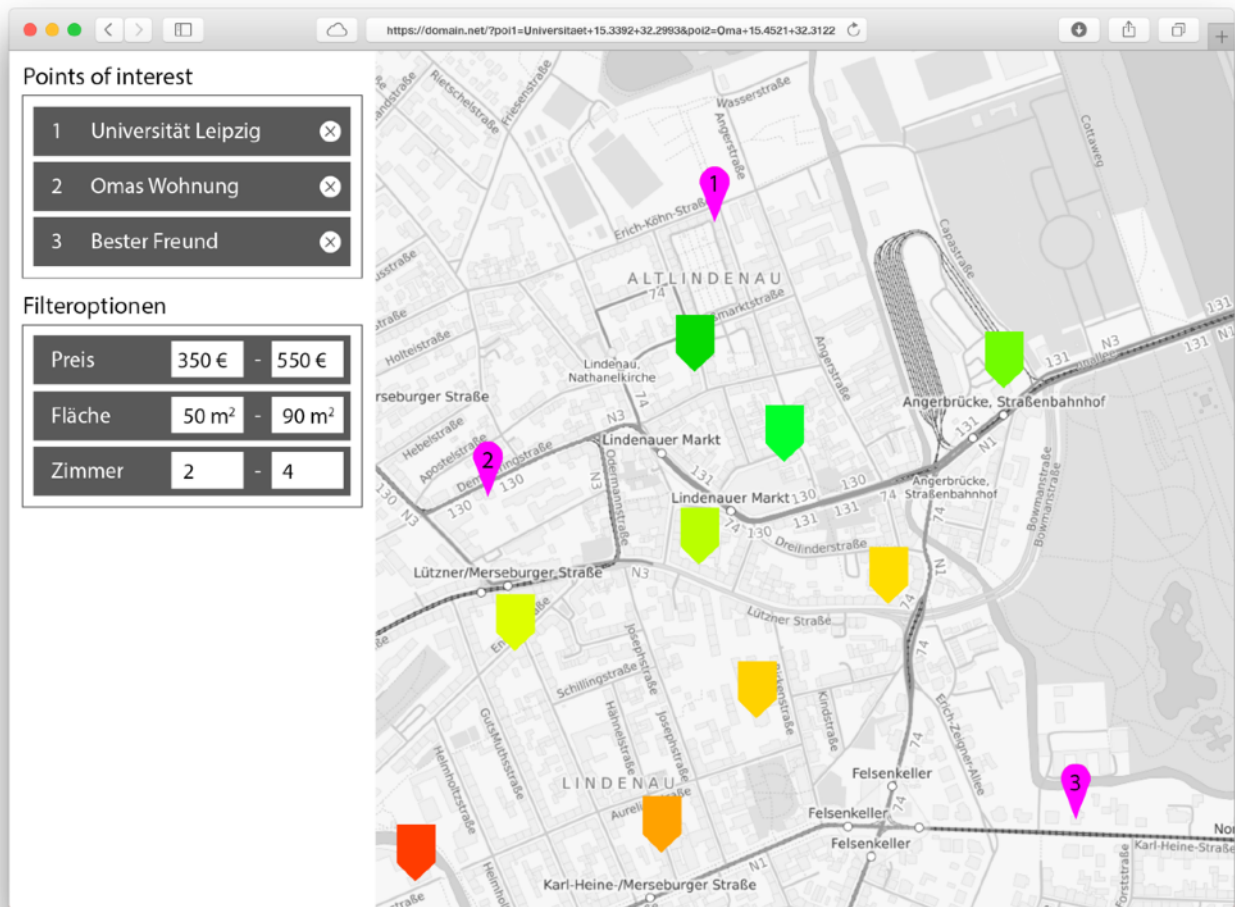
Erstellt im Rahmen von:

SwT-Praktikum 2016/2017

Inhaltsverzeichnis

1. Projektvision
2. Voraussetzungen
3. Designübersicht und Funktionalitäten
4. Arbeitspakete
5. Vorprojekt
6. Glossar

1. Projektvision



Es soll eine Plattform zur Wohnungssuche, unter Nutzung von NodeJs und Angular2, erstellt werden. Auf dieser Plattform können sich Wohnungssuchende einen Überblick über verfügbare Immobilien verschaffen, wobei der Fokus auf der individuellen Analyse der Lage einzelner Wohnungen liegt. Besucher der Seite können Orte angeben, die für Ihren Alltag relevant sind und die sie später in möglichst kurzer Zeit erreichen möchten.

Weiterhin soll es möglich sein nach wichtigen Kriterien wie z.B. Preis, Größe, Zimmeranzahl zu filtern. Da eine große Auswahl von aktuellen Immobilien wichtig ist, werden diese von etablierten t zur Wohnungssuche bezogen (z.B. Immobilienscout24, ImmoWelt, Immonet o.Ä.).

Um den Prozess der Wohnungssuche weitgehend zu vereinfachen, soll es möglich sein die zur Auswahl stehenden Immobilien als Favoriten zu speichern, damit diese auch zu einem späteren Zeitpunkt schnell wieder zu finden sind. Da häufig mehr als eine Person an der Wohnungssuche beteiligt sind, sollte dieser Prozess möglichst einfach gestaltet werden (persistente URI).

2. Voraussetzungen

2.1. Projekt Ressourcen

Zur Umsetzung des Services wird ein Webserver benötigt, auf dem NodeJs ausgeführt werden kann. Dieser wird verwendet um dem Nutzer die Website zur Verfügung zu stellen und um die Zugänge zu verwendeten APIs zu verwalten. Damit dies möglich ist, sind gültige Zugangsschlüssel für sämtliche APIs notwendig. So wird die Route360 API genutzt um die Erreichbarkeitsanalyse von Orten durchzuführen. Weiterhin benötigt werden die API eines Kartenservices, um Koordinaten von Adressen und Orten zu erhalten und auf einer Karte darzustellen und die APIs beliebiger Immobilienservices, um Daten über derzeit verfügbare Immobilien zu erhalten.

2.2. Management Ressourcen

Um die Team interne Kommunikation zu ermöglichen, wird die Organisationsplattform Slack verwendet.

Desweiteren steht für die Repräsentation nach außen eine Website zur Verfügung, welche alle Informationen enthält.

Zuletzt findet die Verwaltung und gemeinsamen Nutzung von Projektressourcen über die Software GitLab statt.

3. Designübersicht und Funktionalität

3.1 Suche nach POIs

Beschreibung:

Benutzer können verschieden POIs (Points of Interests) angeben und nach diesen werden ihm

Wohnungen angezeigt.

Funktionale Voraussetzungen:

Dies kann auf zwei Wegen passieren. Einmal können alle (z.B. täglich) Wohnungsdaten aus Immobilienservices gecrawled und in einer eigenen Datenbank abgespeichert werden. Dann werden diese Daten mittels der Route360 API ausgewertet und auf einer Karte dem Nutzer angezeigt. Oder die Wohnungsdaten werden direkt für jede Nutzeranfrage von den Immobilienservices angefragt und mittels der Route360 API ausgewertet und auf einer Karte dem Nutzer angezeigt.

Ziel:

Benutzer können Wohnungen finden ohne über die gegebenen Ortsbedingungen Hintergrundwissen haben zu müssen.

Beteiligte Rollen: Nutzer

3.2 Wohnungs Rating

Beschreibung:

Angezeigte Wohnungen werden farblich unterschiedlich gewertet, abhängig von ihrer Verkehrsanbindung zu den POIs. Zudem kann der Benutzer angeben mit was für einer Verkehrsanbindung (Fuß, Fahrrad, Auto, Bahn) er reisen möchte.

Funktionale Vorraussetzungen:

Es muss ein Algorithmus entworfen werden welcher die Wohnungen nach bestimmten Parametern bewertet z.B. nach den benötigten Zeiten zu den POIs. Zudem muss das Rating für eine Wohnung grafisch auf der Seite visualisiert werden.

Ziel:

Das Rating soll es dem Benutzer vereinfachen schnell eine passende Wohnung aus den gefundenen herauszusuchen.

Beteiligte Rollen: Nutzer

3.3 Herkömmliche Suchparameter

Beschreibung:

Benutzer können zu den POIs noch weitere Suchparameter wie die Fläche, den Preis oder die Zimmerzahl einer Wohnung eingeben um Wohnungen zu filtern.

Funktionale Vorraussetzungen:

Die Suchparameter müssen auf der Seite auswähl- oder eintragbar sein, desweiteren muss der Suchalgorithmus zur Wohnungssuche diese Parameter mit einbeziehen.

Ziel:

Dem Benutzer werden grundlegende Funktionalitäten zur Wohnungssuche zur Verfügung gestellt.

Beteiligte Rollen: Nutzer

3.4 Persistente URIs

Beschreibung:

Benutzer können die URL einer Suchanfrage an andere Nutzer schicken oder speichern und nach erneutem aufrufen der URL wird die gleiche Suchanfrage ausgeführt und angezeigt.

Funktionale Vorraussetzungen:

Der Inhalt und Aufbau der Seite muss so gewählt werden das er einzigartig mittels URI referenzierbar ist.

Ziel: Nutzer können Links zu ihren Anfragen verschicken und speichern.

Beteiligte Rollen: Nutzer

3.5 Simultane Wohnungssuche

Beschreibung:

Mehrere Benutzer können gleichzeitig zusammen nach Wohnungen suchen.

Funktionale Vorraussetzungen:

Es muss ein System eingerichtet werden, welches es ermöglicht die Sitzungen der Nutzer synchron zu halten, welche Sitzungsdaten (Favoriten, Suchkriterien, etc.) dabei synchronisiert werden ist von der Implementierung abhängig.

Ziel:

Wenn mehrere Leute zusammenziehen wollen, soll es diesen vereinfacht werden gemeinsam eine Wohnung zu finden.

Beteiligte Rollen: Nutzer

3.6 Anzeige der Erreichbarkeit um eine Wohnung

Beschreibung:

Der Benutzer klickt auf eine Wohnung und es wird ihm grafisch dargestellt wohin er mit dem gewählten Transportmittel (Fuß, Fahrrad, Bahn, Auto) in welcher Zeit kommt.

Funktionale Voraussetzungen:

Die Website muss die Auswahl des Benutzer verarbeiten können und abhängig von der Auswahl direkt über die Route360 Api oder über das BackEnd Erreichbarkeits Informationen erhalten und diese anzeigen.

Ziel:

Der Benutzer kann die Lage der Wohnung noch einmal nach eigenem Ermessen im Bezug zur Verkehrsanbindung abschätzen.

Beteiligte Rollen: Nutzer

3.7 Detailanzeige einer Wohnung

Beschreibung:

Benutzer können eine Wohnung auf der Karte anklicken und es werden ihm Details zu dieser Wohnung angezeigt.

Funktionale Voraussetzungen:

Die Website muss die Auswahl des Benutzer verarbeiten können und abhängig von der Auswahl Informationen zu der Wohnung heranziehen (direkt von Drittanbietern oder aus einem Backend) und diese präsentieren können.

Ziel:

Der Benutzer kann direkt Kerninformationen (Preis, Fläche, Zimmerzahl, ...) über die Wohnung ablesen.

Beteiligte Rollen: Nutzer

3.8 Benutzerkonten

Beschreibung:

Ein Benutzer kann sich anmelden und er bekommt zusätzliche Funktionen zur Verfügung gestellt.

Funktionale Voraussetzungen:

Es müssen im Backend eine Datenbank, sowie ein NodeJs Server eingerichtet werden, welche es erlaubt Benutzer zu registrieren, und einen geschützten Anmeldebereich zur Verfügung zu stellen.

Ziel:

Der Benutzer kann unabhängig von seinem Gerät seine Wohnungssuche durchführen.

Beteiligte Rollen: Nutzer

4. Arbeitspakete

4.1. Kern Pakete

4.1.1. Oberflächendesign (30%)

Es ist die das Website Design mit folgenden Anforderungen umzusetzen:

- Responsives Design
- Browserkompatibilität für die am häufigsten genutzten Browser

4.1.2. Arbeitsumgebung einrichten (5%)

Es müssen alle notwendigen Softwarekomponenten (Node.js, Datenbanksystem) nutzbar sein und ein Ng-Fullstack-Generator Projekt muss erstellt werden.

4.1.3 Immobilien Fetcher (15%)

Es ist ein Modul zu erstellen welches sich aus einem Immobilienportal Immobilien Angebote heranzieht und maschinenlesbar zur Verfügung stellt.

4.1.4. Benutzerkonto Funktionalitäten (5%)

Es müssen sowohl im Back- als auch im FrontEnd Funktionalitäten zur Verfügung gestellt werden:

- Benutzer können sich registrieren
- Benutzer können anmelden
- Benutzer können von Immobilien abspeichern
- Die letzte Suche der Nutzer wird automatisch gespeichert und beim einloggen aufgerufen

4.1.5. Persistente URIs (10%)

Die persistenten URIs sind eine Möglichkeit den Inhalt und die momentanen Einstellungen exakt in der URL zu referenzieren, sodass man sich derzeitige such anfragen oder Wohnungen per URL schicken kann.

4.1.6. Dokumentation (5%)

Beschreibung des verwendeten Codes und Anleitungen für erstelle Pakete, um eine später Einsicht zu erleichtern.

4.1.7 FrontEnd (20%)

Es ist eine Angular2 Web Applikation zu schreiben welche die Logik der Benutzerinteraktion, der Verarbeitung und Präsentation übernimmt dazu zählt:

- Die generelle Wohnungssuche
- Alle Benutzerkontenfunktionalitäten

4.1.8 BackEnd (10%)

Es ist eine Datenbank sowie ein Webserver eingerichtet werden außerdem muss ein NodeJs Server programmiert werden, diese Komponenten sollen folgende Dinge zur Verfügung stellen:

- Die Datenbank speichert Benutzerdaten ab
- Der NodeJs Server bietet eine Api die alle vom FrontEnd benötigten Daten bereitstellt

4.2. Optionale Pakete

4.2.1. Multiuser Synchronisierung (20%)

Mehrere Benutzer können gemeinsam nach Wohnungen suchen.

4.2.2. Erweiterung Immobilien Fetcher (50%)

Statt nur die API eines einzigen Immobilienservices zu verwenden, werden Wohnungen von mehreren verschiedenen Anbietern herangezogen und auf Duplikate überprüft.

4.3. Zukünftige Ideen

4.3.1. Monetarisierung der Anwendung

Spezielle Funktionalitäten werden nur noch zahlenden Kunden zur Verfügung gestellt

4.3.2. Einstellen und Verwalten von Immobilien (Eigene Datenbank)

Die Website wird zu einem voll funktionsfähigen Immobilien Portal erweitert.

5. Vorprojekt

Es ist ein Programm zu schreiben welches die Umsetzbarkeit des Projektes verdeutlicht, dieses muss folgendes können:

- Dem Benutzer eine Karte anzeigen auf welchem er Punkte markieren kann
- Für die markierten Punkte werden ihm gewertete Wohnungen angezeigt

6. Glossar

6.1. Webserver

Ein Webserver ist ein Server der Dokumente an Clients bzw. Webbrowser überträgt.

6.2. NodeJs

NodeJs ist eine Plattform für Netzwerkanwendungen basierend auf Javascript. Node.js ist dafür

bekannt, dass es besonders viele gleichzeitige Verbindungen mit relativ geringen Kosten aufrechterhalten kann und schnell zunehmenden Anforderungen gewachsen ist.

6.3. API

Application-Programming-Interface (API) ist ein Programmteil, der von einem Softwaresystem zur Verfügung gestellt wird, um anderen Programmen die Anbindung an das System zu ermöglichen. Die API definiert nur die Programmanbindung auf Ebene des Quelltextes.

6.4. Git

Git ist ein Versions-kontroll-system zur Verwaltung und Synchronisation von Daten.

6.5. Point of interest (POI)

Ein Point of interest (Punkt von Interesse) ist ein punkthafes Geoobjekt, das dem Nutzer einer Karte ein Objekt der realen Welt vereinfacht darstellt, da es für ihn von Bedeutung sein könnte

6.6. Crawler

Ein Crawler ist ein Programm, welches Bereiche des Internets systematisch durchsucht. Dazu werden nach einem bestimmten Muster Ressourcen geladen und diese nach relevanten Informationen durchsucht, welche dann gespeichert werden.

6.7. Route360 Api

Stellt eine API für geographische Netzwerkanalysen, Routenplanung und Visualisierung zur Unterstützung von Komplexen Geo-application bereit. Projektbezogen lassen sich mit Hilfe dieser API Analysen zur Erreichbarkeit verschiedener Orte bzw. zwischen POIs und Immobilien herstellen.

6.8. URI

Ein Uniform Resource Identifier beschreibt eine (meistens) im Web zugängliche Ressource (z.B. HTML-Dokumente).

6.9 persistente URI

Eine persistente URI erlaubt Nutzern den Zugriff auf angepasste Ressourcen unter einem ständig gleichbleibenden URI. Dies erleichtert das Teilen des Inhalte mit anderen.

6.10. Sitzung (Session)

Bezeichnet eine stehende Verbindung von einem Client mit einem Server.

6.11 Geocoder

Ein Geocoder ist eine Anwendung die einen Ortsnamen z.B. "Universität Leipzig" in geographische Koordinaten umwandelt.

6.12 OAuth

OAuth ist ein offenes Protokoll welches eine sichere Authentifizierung über das Web ermöglicht.

6.13 FrontEnd / BackEnd

Die Begriffe Front-End und Back-End werden in der Informationstechnik an verschiedenen Stellen

in Verbindung mit einer Schichten-Architektur verwendet. Dabei ist typischerweise das Front-End

näher am Benutzer, das Back-End näher am System.

Projektbezogen wird der Nutzer im Front-End Eingaben tätigen und die für ihn relevantesten

Immobilien sehen. Im Back-End werden die Datenbank der Immobilien und Nutzer liegen und

notwendige Berechnungen durchgeführt.