

Dokumentation für das Softwaretechnik-Projekt AppCiMo (application for city movement)

Appcimo

Application for City Movement

Please enter your location and destination.

Dachauer Straße, Obersleißheim, München, Deutschl

Latitude: 48.2507798

Longitude: 11.528520200000003

Straße der DSF, Bergen auf Rügen, Deutschland

Latitude: 54.408382399999999

Longitude: 13.434692600000062

Timo Schwertfeger, Daniel Kaiser, Patrick Preuß

19. Juni 2017

Inhaltsverzeichnis

1	Einleitung	3
1.1	Ausgangssituation	3
1.2	Zielsetzung	3
2	Projektvorbereitung	4
2.1	Projektmanagementsoftware Taiga.io	4
2.1.1	Vorgehensmodell	4
2.1.2	Einrichtung Taiga.io	4
2.1.3	Definition von User-Stories	4
2.1.4	Definition von Tasks	4
2.1.5	Integration HipChat	4
2.2	Projektdurchführung	4
2.2.1	Sprintplanung	4
2.2.2	Retrospektive	4
3	Pflichtenheft	5
3.1	Produktanforderungen	5
3.1.1	funktionale Anforderungen	5
3.1.2	Nicht-funktionale Anforderungen	6
3.1.3	Produktanforderungen	6
3.1.4	Abnahmekriterien	6
3.2	Konfiguration und Einrichtung zur Softwareentwicklung	6
3.2.1	Entwicklungsumgebung	6
3.2.2	VueJS	6
3.2.3	Node.js und NPM	6
3.2.4	Github	6
3.2.5	TravisCI	6
4	Systementwurf und Umsetzung	7
4.1	Systemkomponenten	7
4.1.1	Komponentendiagramm	7
4.1.2	Komponentenbeschreibung	7
4.2	Google API	7
4.2.1	Google Services	7
4.2.2	Verarbeitung JSON-Objekte	7
4.2.3	Methoden	7
4.2.4	Refactoring und Tests	7
5	Zusammenfassung und Ausblick	8
5.1	Zusammenfassung	8
5.2	Ausblick	8
6	Anhnge	9
6.1	Glossar	9
6.2	verwendete Software	9

1 Einleitung

1.1 Ausgangssituation

1.2 Zielsetzung

2 Projektvorbereitung

2.1 Projektmanagementsoftware Taiga.io

2.1.1 Vorgehensmodell

2.1.2 Einrichtung Taiga.io

2.1.3 Definition von User-Stories

2.1.4 Definition von Tasks

2.1.5 Integration HipChat

2.2 Projektdurchführung

2.2.1 Sprintplanung

2.2.2 Retrospektive

3 Pflichtenheft

3.1 Produktanforderungen

3.1.1 funktionale Anforderungen

Tabelle 3.1: funktionale Anforderungen

Nr.	Beschreibung
FU01	Das System muss fähig sein JSON-Objekte aus den Anfragen an die Google API zu verarbeiten.
FU02	Sobald der Benutzer eine Verbindung sucht, muss das System dem Benutzer die Möglichkeit bieten einen Start- und Zielort einzugeben.
FU03	Sobald der Benutzer den Start- und Zielort eingibt, muss das System die Möglichkeit bieten, dem Benutzer eine Vorschlagsliste während der Eingabe anzuzeigen.
FU03.01	Die Vorschlagsliste muss bei der Eingabe des ersten Zeichens angezeigt werden.
FU03.02	Die vorgeschlagenen Tupel sollen mit folgender Reihenfolge angezeigt werden, 1. Strasse, 2. Ort, 3. Postleitzahl
FU04	Sobald der Benutzer Start und Zielort eingegeben hat, muss das System die Möglichkeit bieten, die gesuchte Verbindung mit unterschiedlichen Transportmitteln anzuzeigen.
FU04.01	Das Transportmittel zu Fuß muss auswählbar sein.
FU04.02	Das Transportmittel Auto muss auswählbar sein.
FU04.03	Das Transportmittel öffentliche Verkehrsmittel muss auswählbar sein.
FU05	Falls der Benutzer die Suchanfrage ändert, muss das System die Möglichkeit bieten, die gesuchte Verbindung und die Karte zu aktualisieren.
FU05.01	Die Vorschlagsliste muss bei Neueingabe des Start- und Zielorts angezeigt werden.
FU05.02	Die Verbindungskarte muss mit neuem Start- und Zielort die gesuchte Verbindung anzeigen.
FU05.03	Die Distanz der neuen Verbindung muss aktualisiert werden.
FU05.04	Die Dauer der neuen Verbindung muss aktualisiert werden.
FU05.05	Der Preis der neuen Verbindung muss angezeigt werden.
FU06	Falls der Benutzer eine Verbindung sucht, muss das System die Möglichkeit bieten, mehrere Transportmittel auszuwählen.
FU07	Sobald der Benutzer eine Verbindung sucht, muss das System die Möglichkeit bieten, eine Ergebnisliste der gesuchten Verbindung anzuzeigen.
FU07.01	In der Ergebnisliste muss der aktuelle Preis angezeigt werden.
FU07.02	In der Ergebnisliste muss die Dauer angezeigt werden.
FU07.03	In der Ergebnisliste muss die Distanz angezeigt werden.

3.1.2 Nicht-funktionale Anforderungen

Tabelle 3.2: Projektanforderungen

Nr.	Beschreibung
NFU01	Das System muss plattformunabhängig und webbasiert sein.
NFU02	Das System muss mit einem Entwicklungs-Framework umgesetzt werden.
NFU03	Das System soll als Single-Page Anwendung umgesetzt werden.

3.1.3 Projektanforderungen

Tabelle 3.3: Nicht-funktionale Anforderungen

Nr.	Beschreibung
PRJ01	Für die Umsetzung des Systems soll ein modernes Entwicklungs-Framework für die Softwareerstellung genutzt werden.
PRJ02	Es muss eine Projektmanagementsoftware, für die mit der Softwareerstellung einhergehende Projektarbeit, genutzt werden.
PRJ03	Der entwickelte Programmcode muss auf Github als Master-Branch hochgeladen werden.
PRJ04	Für das Software-Projekt muss eine Projektdokumentation erstellt werden.

3.1.4 Abnahmekriterien

3.2 Konfiguration und Einrichtung zur Softwareentwicklung

3.2.1 Entwicklungsumgebung

3.2.2 VueJS

3.2.3 Node.js und NPM

3.2.4 Github

3.2.5 TravisCI

4 Systementwurf und Umsetzung

4.1 Systemkomponenten

4.1.1 Komponentendiagramm

4.1.2 Komponentenbeschreibung

4.2 Google API

4.2.1 Google Services

4.2.2 Verarbeitung JSON-Objekte

4.2.3 Methoden

4.2.4 Refactoring und Tests

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

5.2 Ausblick

6 Anhnge

6.1 Glossar

6.2 verwendete Software