2016

Modul 318

Silvan Bucher

Überbetrieblicher Kurs

# Einleitung

Der ÜK 318 hat das Thema Objektorientiertes Programmieren. Der grösste Teil des Kurses besteht aus einem Projekt, welches auf einem Framework aufgebaut wird. Das Framework stellt eine Verbindung zu opendata.ch her, welche Daten zu den ÖV Verbindungen in der Schweiz zur Verfügung stellen. Unsere Aufgabe ist es, eine Applikation zu entwickeln, welche verschieden Funktionen, wie zum Beispiel Verbindungssuche oder Abfahrtspläne anbietet.

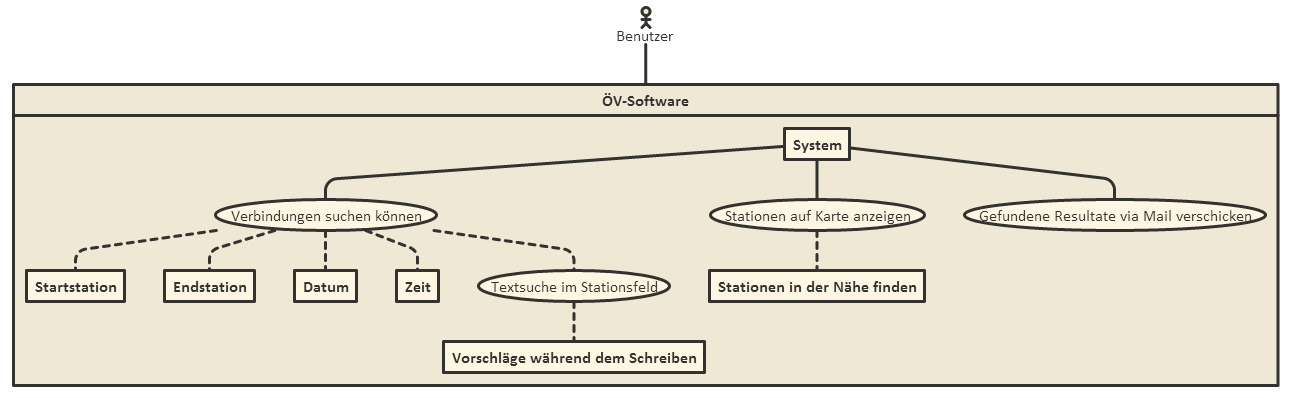
# Anforderungen

Hier eine komplette Liste der Anforderungen. Die Anforderungen 1-3 sind obligatorisch, 4-8 gelten als Zusatz.

|  |  |
| --- | --- |
| ID | Beschreibung |
| A001 | Ich als ÖV­Benutzer möchte die Start und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. |
| A002 | Ich als ÖV­Benutzer möchte die aktuellen Verbindungen zwischen den beiden gefundenen und ausgewählten Station sehen, damit ich weiss wann ich zur Station muss. |
| A003 | Ich als ÖV­Benutzer möchte sehen welche Verbindungen ab einer bestimmten Station vorhanden sind. Damit ich eine Art Station Abfahrtstafel Zuhause haben kann. |
| A004 | Ich als ÖV­Benutzer möchte, dass schon während meiner Eingabe erste Such­Resultate erscheinen, damit ich effizienter nach Stationen suchen kann. |
| A005 | Ich als ÖV­Benutzer möchte Verbindungen zu einem anderen Zeitpunkt suchen können, damit ich zukünftige Reisen planen kann. |
| A006 | Ich als ÖV­Benutzer möchte sehen wo sich eine Station befindet, damit ich mir die Vor Ort­Situation besser vorstellen kann. |
| A007 | Ich als ÖV­Benutzer möchte Stationen finden die in der Nähe meiner aktuellen Position sind, damit ich schnell auf das ÖV Netz der Schweiz kommen kann. |
| A008 | Ich möchte meine gefundenen Resultate via Mail weiter schicken können, damit ich oder andere von meinen Recherchen profitieren können. |

# Usecases

Diese Anforderungen habe ich auch in ein Usecase Diagram übertragen, welches das ganze Bildlich Darstellt.



# Testing

## Testmethode

Ich werde diese Testfälle mit der AAA Methode schreiben. Diese Methode besteht aus folgenden Schritten:

1. **Arrange**  
   Die Bedingungen für den Test werden Vorbereitet
2. **Act**  
   Es werden bestimmte Aktionen durchgeführt
3. **Assert**  
   Es wird geprüft, ob die vorher ausgeführten Aktionen das erwünschte Ergebnis zurückliefern

## Testfälle

### Anforderung 1

|  |
| --- |
| Die Applikation wird gestartet. Der Tab „Verbindungen“ wird geöffnet |
| Im Feld „Von“ wird der Text „abc“ eingegeben |
| Es wird die Ortschaft „Abcoude“ vorgeschlagen |

### Anforderung 2

|  |
| --- |
| Die Applikation wird gestartet. Der Tab „Verbindungen“ wird geöffnet |
| Im Feld „Von“ wird der Text „Sursee“ eingegeben  Im Feld „Nach“ wird der Text „Luzern“ eingegeben  Im Feld „Datum“ wird der Text „01.06.2016“ eingebeben  Im Feld „Zeit“ wird der Text „14:40:00“ |
| Der erste Eintrag in der Verbindungstabelle enthält die Abfahrtszeit „14:41“ |

### Anforderung 3

|  |
| --- |
| Die Applikation wird gestartet. Der Tab „Abfahrtsplan“ wird geöffnet |
| Im Feld „Station“ wird der Text „Sursee“ eingegeben |
| Es wird ein kompletter Abfahrtsplan von Sursee angezeigt |

# Installation

Die Applikation läuft grundsätzlich ohne grosse Abhängigkeiten. Um die Applikation zu starten sind aber trotzdem einige .dll Dateien nötig. Das .exe File der Applikation und alle nötigen .dll Files sind im *rel* Ordner auf dem Root meines Repositorys zu finden.

Eine separate Installation ist nicht nötig, das .exe File muss einfach im Ordner mit den .dll Dateien gestartet werden.