



# ÜK MODUL 318

Joël Theiler, 24.4.2018

## Inhalt

1	Einleitung.....	1
2	Zweck des Dokuments.....	1
3	Funktionen.....	1
3.1	Umgesetzte Funktionen .....	1
3.2	Nicht umgesetzt/teilweise umgesetzte Funktionen .....	2
3.2.1	Fehler/Bugs.....	2
4	UML Anwendungsfalldiagramm .....	2
5	Mockups .....	4
6	Programmrichtlinien .....	4
7	Testfälle .....	5
8	Installationsanleitung .....	6
8.1	Deinstallieren der App.....	6
9	Schlusswort .....	6

## 1 Einleitung

In diesem ÜK Modul haben wir den Auftrag erhalten, eine Windows Forms App zu programmieren, die auf eine API von Opendata zurückgreift. Diese muss Verbindungen anzeigen, die der Benutzer selbst eingibt. Die Eingabe wird mithilfe von Type Ahead erleichtert. Zusätzlich muss die App auch einen Abfahrtsplan beinhalten. Am Schluss wird diese App bewertet.

## 2 Zweck des Dokuments

Dieses Dokument ist dafür gedacht, die Programmierinformationen festzuhalten. Zusätzlich dient sie auch als Installationsanleitung, sodass ein fremder Programmierer das Programm besser verstehen kann.

## 3 Funktionen

### 3.1 Umgesetzte Funktionen

Ich habe alle Funktionen umgesetzt, welche als obligatorisch gekennzeichnet waren. Zudem habe ich die meisten Zusatzaufgaben auch programmieren können.

### 3.2 Nicht umgesetzt/teilweise umgesetzte Funktionen

A004 → Die Funktion mit dem Auto Complete funktioniert nicht, da ich mich nicht so gut mit dem Benutzersteuerelement auskenne.

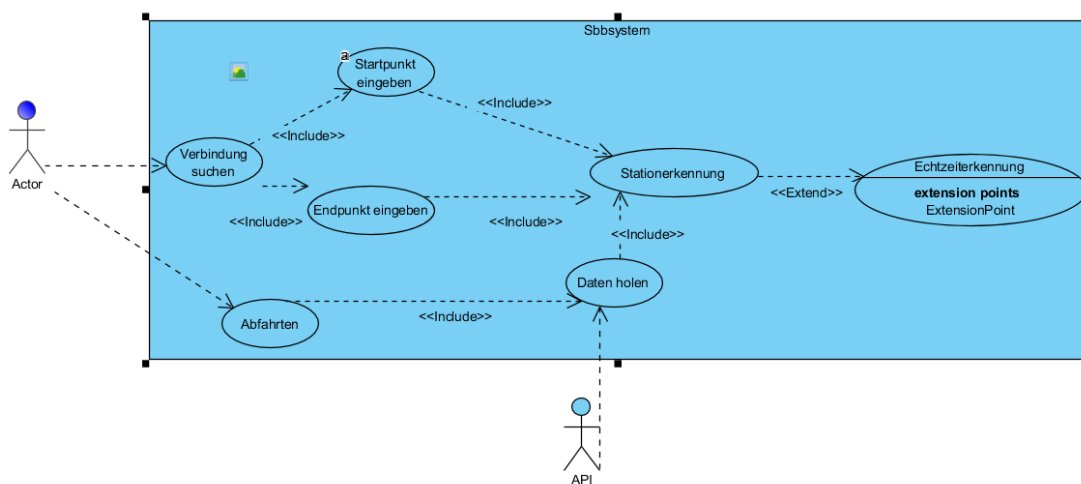
A007 → Diese Funktion funktioniert nicht, da ich kein Parameter gefunden habe, der die Entfernung misst.

#### 3.2.1 Fehler/Bugs

Wenn das Programm keine Internetverbindung herstellen kann, stürzt es ab.

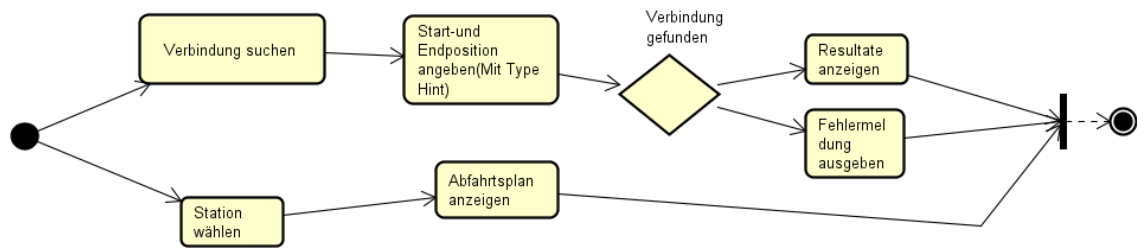
## 4 UML

### 4.1 Anwendungsfalldiagramm



## 4.2 Aktivitätsdiagramm

actSbbapp



## 5 Mockups

Fahrplan

Station(Mit Text ahead)

Station	Uhrzeit	Gleis
Meggen Zentrum	11:15	22
Meggen Zentrum	11:25	2
Meggen Zentrum	11:35	22
Meggen Zentrum	11:45	2
Meggen Zentrum	11:55	22
Meggen Zentrum	12:15	2

Transport App

Verbindungen werden angezeigt

## 6 Programmrichtlinien

- Jede definierte Variable erhält einen sinnvollen Namen
- Ich verwende möglichst wenig redundanten Code
- Button beginnen mit btn\_
- Comboboxen beginnen mit cb\_

## 7 Testfälle

Schritt	Aktivität	Erwartetes Resultat
<b>A001/A002</b>		
<b>1</b>	Interessent schreibt in die Combobox Startposition und Endposition. Dann drückt er auf die beiden «Station suchen» Buttons	Die exakten ÖV Haltestellen werden eingefüllt.
<b>2</b>	Der Interessent klickt auf den Button «Verbindung suchen» Button	Die Verbindung wird von der API berechnet und mithilfe eines JSON Files beantwortet.
<b>3</b>	Interessent gibt folgende Daten ein: <ul style="list-style-type: none"> <li>Combobox Startposition: Huob</li> <li>Combobox Endposition: Luzern Bahnhof</li> </ul> Dann klickt er auf «Verbindung suchen»	Dem Wert Huob wird Meggen, hinzugefügt. Der Wert Luzern Bahnhof
<b>A003</b>		
<b>1</b>	Der Interessent wählt nach dem Ausfüllen der Start— und Endposition den Button «Verbindung suchen»	Die Verbindungen werden im DataGridView angezeigt.
<b>2</b>	Der Interessent wählt den Button «zum Fahrplan»	Die Form2 wird gestartet.
<b>3</b>	Der Interessent gibt folgende Daten ein: <ul style="list-style-type: none"> <li>Combobox: Huob</li> </ul> Anschliessend drückt er auf den Button «Verbindungen»	Die API berechnet die Verbindungen. Sie werden in dem DataGridView ausgegeben.
<b>A005</b>		
<b>1</b>	Der Interessent drückt auf die DateTimePicker	Der Kalender des DateTimePickers öffnet sich.
<b>2</b>	Der Interessent gibt folgende Daten ein: <ul style="list-style-type: none"> <li>Datum DateTimePicker: 24/04/2018</li> <li>Time DateTimePicker: 12:00:00</li> </ul>	Die API rechnet die Verbindungen aus, die an diesen zwei Bedingungen beginnt und gibt sie anschliessend im DataGridView aus.
<b>A008</b>		
<b>9</b>	Der Interessent wählt den Button «zum Fahrplan». Anschliessend gibt er die gewünschte Station und validiert sie mit dem «Station suchen» Button. Danach drückt er auf den «Verbindungen Button». Als Letztes gibt er die E-Mail Adresse ein.	Die API berechnet die Verbindungen. Anschliessend sendet das Programm mittels smtp Server das Email via eine Gmail.
<b>A006</b>		
<b>10</b>	Der Interessent wählt den Button «Zum Station Fahrplan»	Die Form2 wird gestartet
<b>11</b>	Der Interessent gibt folgende Daten ein: <ul style="list-style-type: none"> <li>Combobox: Huob</li> </ul> Anschliessend klickt er auf «Verbindungen». Danach klickt er auf «Station auf Google Maps anzeigen»	Die API holt nun die Koordinaten der Station. Im Anschluss werden die Koordinaten umformatiert.

## 8 Installationsanleitung

1. Auf GitHub diese Repository Klonen <https://github.com/Jothei/modul-318-student>
2. Im Ordner «Fertiges Programm» die Setup.exe ausführen oder alternativ die Setup.msi
3. Die Schritte im Installer befolgen (Installationspfad merken!)
4. Wenn das Programm erscheint. Schliessen, damit das Setup erfolgreich Abgeschlossen werden kann
5. Anschliessen das Programm starten (Exe vom Installationspfad)

### 8.1 Deinstallieren der App

- Windows Taste drücken
- «Programme hinzufügen oder entfernen» eintippen
- Die SetupSBBAPP auswählen und auf «Deinstallieren» drücken
- Den Wizzard befolgen
- Fertig

## 9 Schlusswort

Dieses ÜK Modul war eines der anspruchsvolleren ÜKs in meiner bisherigen Ausbildung. Es mussten einige Hürden überwunden werden, um ans Ziel zu kommen. Doch genau das war mein Challenge, möglichst viele Zusatzaufgaben zu lösen.

Besonders das Suchen einer Verbindung zwischen zwei ÖV Stationen erwies sich als anspruchsvoll. Es gab immer Syntaxfehler. Die API wurde immer falsch angesteuert. Nach langem Prübeln und austesten kam ich dann auf die Idee, eine foreach-Schleife zu verwenden, welche dann perfekt für diese Situation passte. Danach funktionierte dieser Programmierteil einwandfrei. Solche Problemsituationen zeigen mir immer wieder auf, dass man keine komplizierten Lösungsansätze beim Programmieren suchen muss, sondern in der Kürze, respektive in der Einfachheit liegt die Würze!