

Projektarbeit M318

Florian Ryser, 12.12.2017

Leiter: Hanspeter Stalder

Inhalt

1. Einleitung.....	3
2. Zweck des Dokuments.....	3
3. Idee/Mockup	3
3. Zielgruppe.....	4
4. Umgesetzte Funktionen	4
4.1 Aufgabe 001 + 004.....	4
4.2 Aufgabe 002	4
4.3 Aufgabe 003	5
4.4 Aufgabe 006	5
4.5 Eigene Funktion	5
4.6 Fehlende Funktionen / Bugs / Fehler	5
5. USE-CASE	5
6. Activity-Diagramm.....	5
7. Testfälle	6
7.1 Aufgabe 001 + 004 / Ausgabe von Stationen	6
7.2 Aufgabe 002 / Ausgabe von Verbindungen.....	7
7.3 Aufgabe 003 / Ausgabe vom Stationsplan	7
7.4 Aufgabe 004 / Google Maps.....	7
8. Installation / Deinstallation	7

1. Einleitung

In diesem ÜK geht es darum eine Applikation zu programmieren, welche die Fahrplandaten des Schweizerischen öffentlichen Verkehrs benutzt. Mit der Applikation soll es möglich sein, Verkehrsverbindungen zwischen zwei Stationen zu suchen. Die Applikation und diese Dokumentation werden zu Abschluss von Kursleiter bewertet.

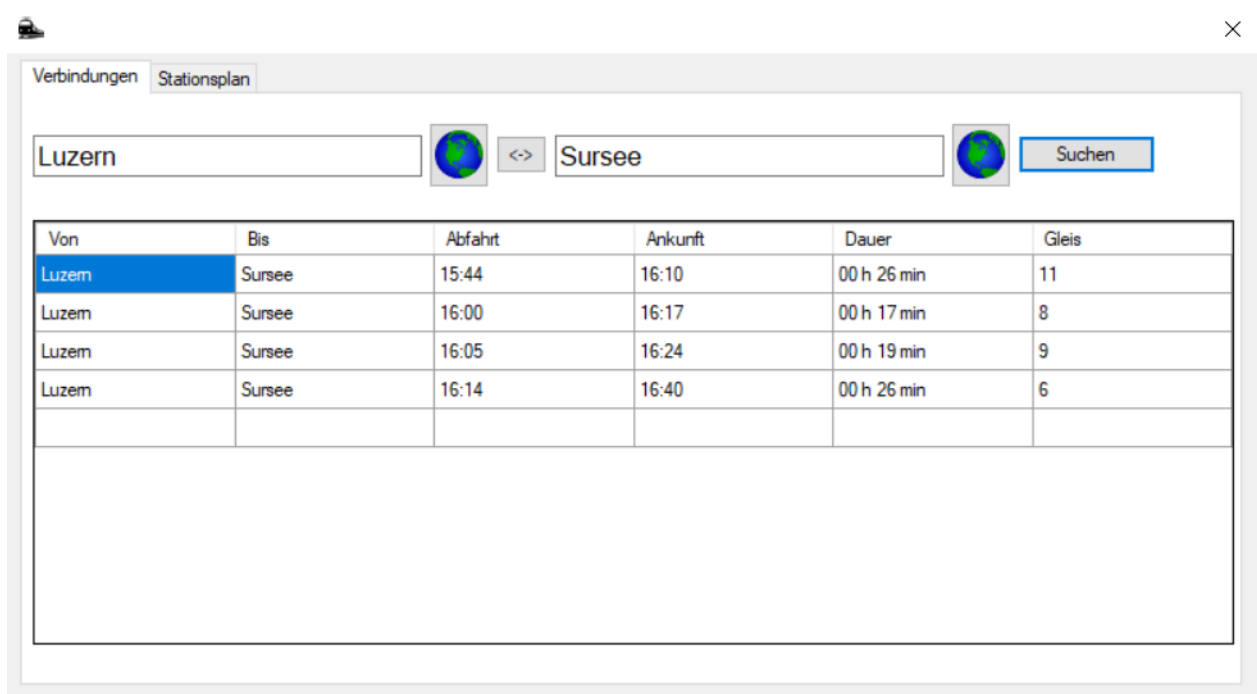
2. Zweck des Dokuments

Zweck dieser Dokumentation ist, dass verschiedene Punkte wie umgesetzte Funktionen, Use-Case, Aktivitätendiagramm, Testfälle und weiteres klar für den Anwender dokumentiert ist. Die Dokumentation zählt auch zur Bewertung. Folgend werden die wichtigsten Punkte und eigene Infos zum Projekt erläutert.

3. Idee/Mockup

Die Idee ist eine schlichte Windows-Applikation im Stile der SBB-Website. Es soll der Abfahrts- und Zielort, Datum und Zeit eingeben. Bei Eingabe sollten mindestens fünf Verbindungen zu den eingegebenen Angaben angezeigt werden. Ausserdem sollten Features wie ein Button zum Tauschen des Abfahrt- und Zielorts implementiert, Standort-Anzeige usw. angezeigt werden.

Das Mockup der fertigen Applikation sieht wie folgt aus:



The mockup shows a window titled 'Verbindungen' with a 'Stationsplan' tab. It features two input fields for station names, 'Luzern' and 'Sursee', separated by a swap button '< >'. A search button 'Suchen' is to the right. Below the inputs is a table with the following data:

Von	Bis	Abfahrt	Ankunft	Dauer	Gleis
Luzern	Sursee	15:44	16:10	00 h 26 min	11
Luzern	Sursee	16:00	16:17	00 h 17 min	8
Luzern	Sursee	16:05	16:24	00 h 19 min	9
Luzern	Sursee	16:14	16:40	00 h 26 min	6

Bis	Nummer	Abfahrt	Gleis
Basel SBB	21	15:54	7
Bern	RE 4478	15:57	5
Willisau	RE 4478	15:57	5
Stans	4	15:57	14
Genève-Aéroport	15	16:00	8
Lenzburg	9	16:02	10
Olten	RE 4728	16:05	9
Interlaken Ost	IR 2932	16:06	12
Brunnen	3	16:06	11
Zürich Flughafen	70	16:10	4
Engelberg	IR 2978	16:10	13

3. Zielgruppe

Zielgruppe bei einer Applikation dieser Art sind Pendler die tagtäglich mit dem ÖV unterwegs sind. Sie vertrauen auf eine schnelle und präzise Angabe über die benötigten Zugverbindungen.

4. Umgesetzte Funktionen

4.1 Aufgabe 001 + 004

Beschreibung: Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. Ausserdem möchte ich, dass schon während meiner Eingabe erste Suchresultate erscheinen, damit ich effizienter nach Stationen suchen kann.

Umsetzung: Diese beiden Aufgaben habe ich zusammengefasst. Da das Autocomplete bei der Textbox eigentlich das Gleiche bewirkt wie eine Textsuche für die Stationen. Da aber die Autocomplete-Funktion einer Textbox nicht dynamisch erweitert werden kann musste ich diese Aufgabe mit Hilfe einer direkt anliegenden ListBox lösen. Diese wird bei Texteingabe mit Stationen gefüllt welche dem eingegebenen Text ähneln. Die Einträge in der ListBox können doppelt angeklickt werden und somit direkt in die Textbox eingefügt werden.

4.2 Aufgabe 002

Beschreibung: Als ÖV-Benutzer möchte ich die aktuellen, d.h. mindestens die nächsten vier bis fünf Verbindungen zwischen den beiden gefundenen und ausgewählten Stationen sehen, damit ich weiss wann ich zur Station muss, um den für mich idealen Anschluss zu erwischen.

Umsetzung: Bei dieser Aufgabe musste eine Verbindung zwischen dem eigenen Code und der zur Verfügung gestellten API hergestellt werden. Die eingegebenen Stationen werden bei Klick auf den Suchen-Button an eine Funktion in der API weitergegeben. Von dieser Funktion wird eine Sammlung der gefundenen Verbindungen an ein DataGridView weitergegeben und somit auf der Form in einer Tabelle angezeigt.

4.3 Aufgabe 003

Beschreibung: Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann.

Umsetzung: Diese Aufgabe wurde gleich umgesetzt wie die Aufgabe 002. Der Unterschied ist aber, dass nur eine Station eingegeben werden muss. Bei Klick auf den Suchen-Button wird die eingegebene Station an eine Funktion in der API weitergegeben und die Sammlung in ein DataGridView weitergegeben.

4.4 Aufgabe 006

Beschreibung: Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht.

Umsetzung: Hierzu habe ich neben allen TextBoxen einen Button mit einer Erdkugel als Icon erstellt. Bei Klick auf diesen Button wird die eingetragene Station auf Google Maps im Standard-Browser geöffnet.

4.5 Eigene Funktion

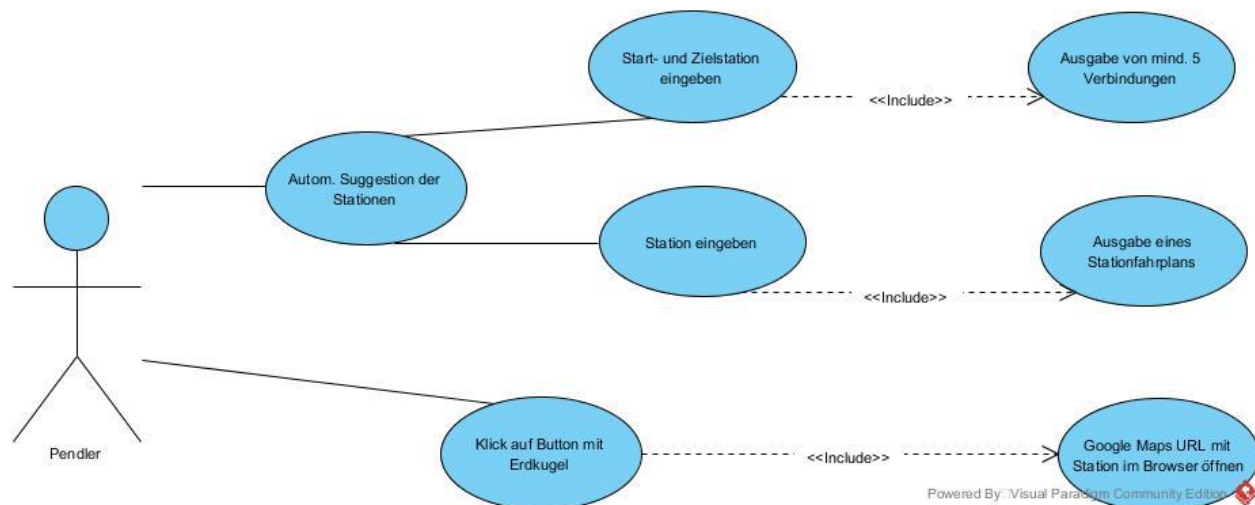
Eigens von mir eingefügt wurde ein Button auf dem Tab „Verbindungen“ zwischen den beiden TextBoxen. Bei Klick auf diesen Button werden die Einträge in den TextBoxen vertauscht. Dies kann hilfreich sein, wenn man die Rückreise auch direkt mitplanen will bei einer Reise.

4.6 Fehlende Funktionen / Bugs / Fehler

Eine wichtige fehlende Funktion ist sicherlich, dass es zurzeit nicht möglich ist die Zeit und das Datum zu ändern. Eine Feature wäre noch, dass man Einträge in der Listbox unterhalb der TextBoxen mit Hilfe der Pfeiltasten und der Enter-Taste auswählen kann.

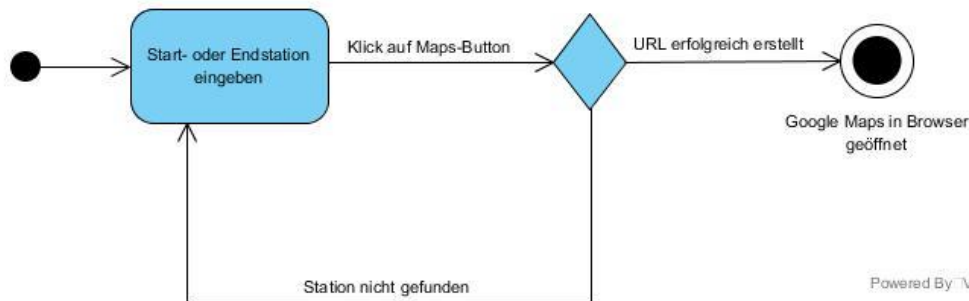
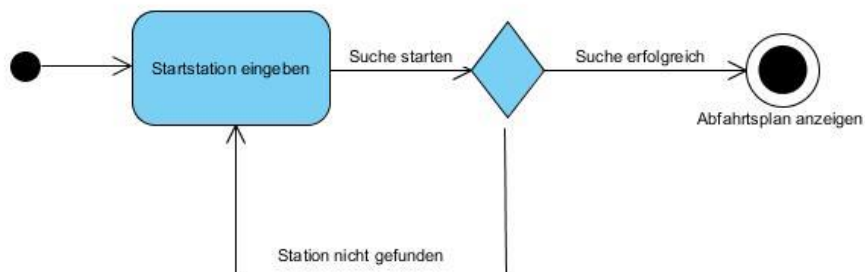
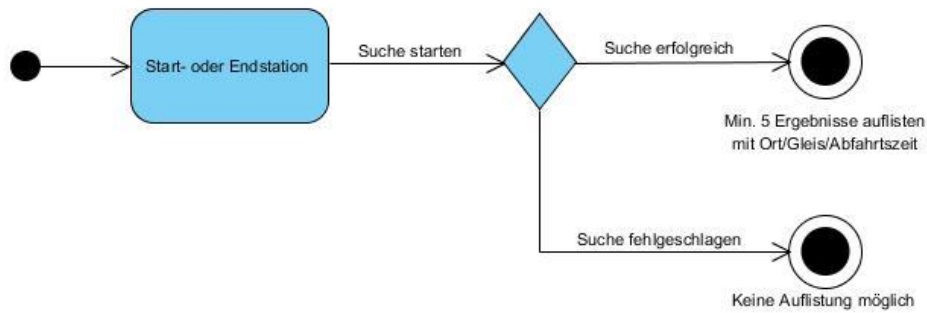
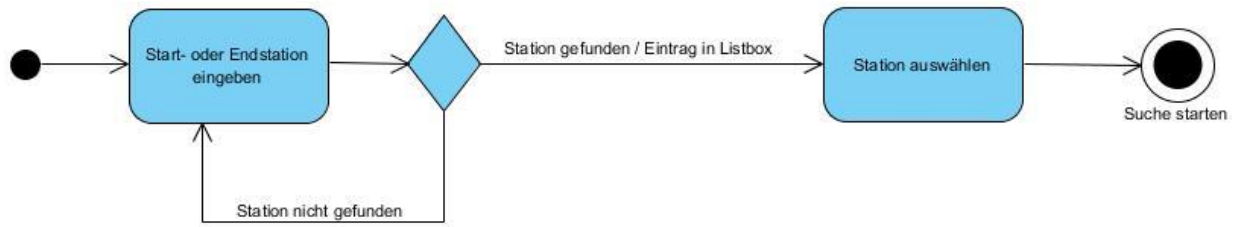
5. USE-CASE

Hier sieht man das USE-CASE-Diagramm zu den Aufgaben 1, 2, 3, 4 und 6. Aufgabe 1 und 4 wurden wie bereits oben beschrieben zusammengefügt da bei eigentlich das Gleiche bewirken.



6. Activity-Diagramm

Hier sieht man das Activity-Diagramm zu den Aufgaben 1, 2, 3, 4 und 6. Aufgabe 1 und 4 wurden wie bereits oben beschrieben zusammengefügt da bei eigentlich das Gleiche bewirken.



Powered By Visual Paradigm Community Edition

7. Testfälle

7.1 Aufgabe 001 + 004 / Ausgabe von Stationen

Eingabe: Bei den Textboxen muss nur der Anfang eine Station eingegeben werden. Zum Beispiel „Lu“.

Ausgabe: In Listboxen unterhalb der Textboxen werden dann alle Stationen aufgelistet die mit „LU“ beginnen. Die Listbox wird dynamisch erweitert wenn der Text in der Textbox geändert wird.

7.2 Aufgabe 002 / Ausgabe von Verbindungen

Eingabe: Auf dem Tab Verbindungen bei den Textboxen jeweils eine gewünschte Station eingeben. Zum Beispiel „Luzern“ und „Sursee“. Dann Klick auf den Button Suchen.

Ausgabe: Bei Klick auf den Button dauert es einen kurzen Moment und dann erscheint eine Tabelle mit den Verbindungen von Luzern nach Sursee. Die Ausgabe beinhaltet Von, Bis, Abfahrtszeit, Ankunftszeit, Dauer und das Gleis.

7.3 Aufgabe 003 / Ausgabe vom Stationsplan

Eingabe: Auf dem Tab Stationsplan bei der Textbox eine gewünschte Station eingeben. Zum Beispiel „Luzern“. Dann Klick auf den Button Suchen.

Ausgabe: Bei Klick auf den Button dauert es einen kurzen Moment und dann erscheint eine Tabelle mit den Fahrten von Luzern aus. Die Ausgabe beinhaltet Bis, Zugnummer, Abfahrtszeit und das Gleis.

7.4 Aufgabe 004 / Google Maps

Eingabe: Auf jedem Tab befindet sich jeweils neben einer Textbox ein Button mit einer Erdkugel drauf. Nach Eingabe, z.B. „Luzern, Bahnhof“, kann auf diesen Button geklickt werden.

Ausgabe: Bei Klick auf den Button wird die eingetragene Station mit Hilfe von Google Maps in einem neuen Browserfenster angezeigt.

8. Installation / Deinstallation

Um das Programm zu installieren muss man das Projekt zuerst aus meinem GitHub herunterladen. Das ist auf: <https://github.com/OneAndOnlyGOD/modul-318-student.git>

Man muss dafür einfach auf den Download Button klicken. Wenn man dann die VisualStudio Solution einmal gestartet hat, kann man in seinem Projektordner auf \ProjektM318\bin\Debug dort findet man eine .exe Datei. Die kann man dann wegkopieren und überall benutzen als eigenständiges Programm benutzen.

Um das Projekt dann zu deinstallieren kann man diese Datei dann auch wieder von dort weglöschen, wo man es hin kopiert hat.