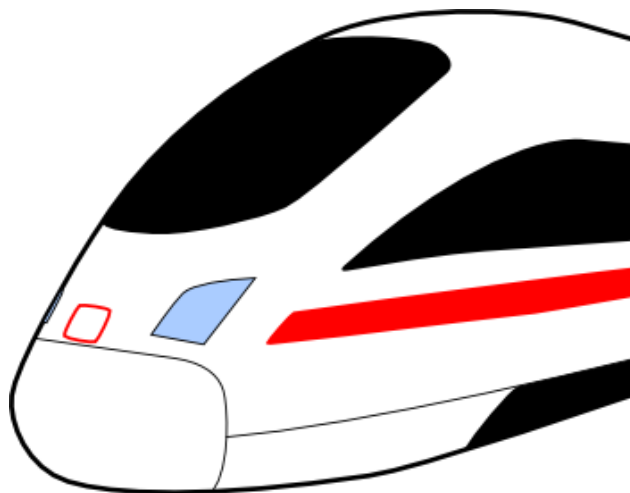


PROJEKTARBEIT ÜK318

Fahrplan



Manuel Troxler

12/5/2017



Inhalt

1	Einleitung.....	2
1.1	Zweck des Dokuments.....	2
2	Ablauf.....	2
2.1	Erstellung des Mockups.....	2
2.2	Erstellung des Use Case Diagramms.....	3
2.3	Erstellung eines Aktivitätsdiagramm.....	3
3	Eingebaute Funktionen.....	4
3.1	Aufgabe 001	4
3.2	Aufgabe 002	4
3.3	Aufgabe 003	4
3.4	Aufgabe 004	4
3.5	Aufgabe 006	4
4	Systemtests.....	5
4.1	Testfall 1 - Suchvorschläge.....	5
4.2	Testfall 2 – Verbindungen anzeigen	5
4.3	Testfall 3 – Haltestellen Abfahrtstafel anzeigen.....	5
4.4	Testfall 4 - Von: & Nach: Einträge wechseln	6
4.5	Testfall 5 – Haltestellen auf der Map anzeigen lassen.....	6
5	Beschreibung des GUI.....	7

1 EINLEITUNG

Zum Abschluss dieses Überbetrieblichen Kurses müssen wir ein Projekt machen. In dem Projekt müssen wir eine Windows Forms Anwendung machen, bei der wir den Fahrplan von SBB übernehmen müssen und so über unser Programm alle Fahrpläne abrufen können. Die Klassen für den Zugriff auf die Daten von SBB sind schon vorgegeben, denn das wäre zu aufwendig und zu anspruchsvoll.

1.1 Zweck des Dokuments

Dieses Dokument dient der genauen Beschreibung des Programmes(Funktionalität), dem Ablauf wie ich vorgegangen bin und auch eine genaue Erklärung wie man Testfälle ausführen muss. Ebenfalls in diesem Dokument findet man die Installations- / Deinstallationsanleitung.

2 ABLAUF

2.1 Erstellung des Mockups

Das Mockup dient dazu, dass man das gewünschte finale Aussehen des GUI ungefähr aufzeichnet. Dass kann einem später helfen, wenn man das GUI gestaltet. Das sieht bei mir so aus:

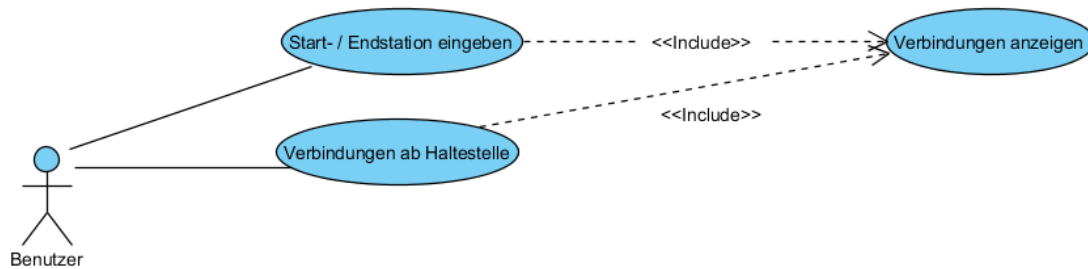
The mockup shows a window titled "Fahrplan" with the following elements:

- Input fields for "Von" (Origin) and "Nach" (Destination), each with a dropdown arrow and a globe icon.
- A date input field showing "01/01/2017" with a calendar icon.
- A button labeled "Inhalte tauschen" (Exchange content).
- A button labeled "Search".
- A dropdown for "Ab Haltestell" (From station) with a globe icon.
- Buttons labeled "Früher" (Earlier) and "Später" (Later).
- A large rectangular area containing a list of connections:
 - Verbindung 1
 - Verbindung 2
 - Verbindung 3

Das ist nicht genau wie es dann aussehen wird, weil das nur als ungefähre visuelle Unterstützung gedacht ist. Aber das GUI wird sicher an das Mockup anlehnen.

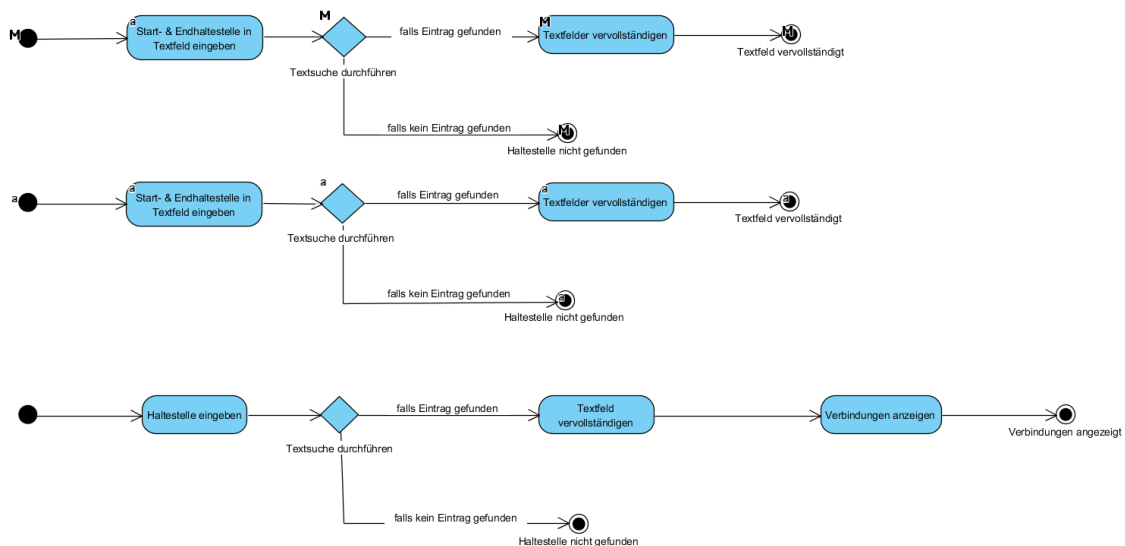
2.2 Erstellung des Use Case Diagramms

Das Use Case Diagramm dient dazu, dass man genau sieht was für Sachen der Benutzer später in der Software machen kann und was sie auslösen.



2.3 Erstellung eines Aktivitätsdiagramm

Wir haben auch ein Aktivitätsdiagramm gemacht in dem man sieht was für Aktivitäten was für Reaktionen auslösen und was das erfüllt sein muss, dass man zu dem gewünschten Endpunkt kommt.



3 EINGEBAUTE FUNKTIONEN

3.1 Aufgabe 001

Beschreibung: Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.

Ich habe diese Aufgabe mit der Aufgabe 004 verknüpft und es so gelöst dass man während der Eingabe direkt sieht, was es für Stationen gibt.

3.2 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.

Ich habe das gelöst in dem ich eine DataGridView verwendet habe und da dann die Verbindungen hereingeladen habe. Das habe ich dann auf dem Such-Button Ereignis gemacht.

3.3 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.

Das habe ich mit einem eigenen Textfeld und somit auch mit einem eigenen Button gelöst. Man kann dort auch mit den Bekannten Stationsvorschlägen nach einer Station suchen und zu der dann die nächsten Verbindungen von dieser Haltestelle ansehen.

3.4 Aufgabe 004

Beschreibung: Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Suchresultate erscheinen, damit ich effizienter nach Stationen suchen kann.

Wie schon bei Aufgabe 001 beschrieben, wird das durch eine Listbox gelöst, die auf jedes Textbox_TextChanged Ereignis reagieren und die passenden Vorschläge angibt.

3.5 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.

Ich habe neben allen Textboxen, bei denen man eine Haltestelle eingeben kann, ein Erde Symbol daneben gemacht. Wenn man dieses anklickt wechselt es automatisch zu Google Maps auf die gewünschte Haltestelle.

4 SYSTEMTESTS

In diesem Kapitel werden einige Tests beschrieben, so dass jede Person diese nachmachen könnte und das Programm ohne Vorwissen Testen. Ich werde somit jeden Testfall genau beschreiben.

4.1 Testfall 1 - Suchvorschläge

Bedingung ist, dass das Programm gestartet, ohne eine Änderung vor sich zu haben.

Schritt Nr.	Aktivität	Erwartetes Ereignis
1	Benutzer gibt in dem «Von:» Feld Buchstaben ein.	Es kommen bei jedem eingegebenen Buchstaben wieder neu geladen die besten Vorschläge.
2	Benutzer gibt in dem «Nach:» Feld Buchstaben ein.	Es kommen bei jedem eingegebenen Buchstaben wieder neu geladen die besten Vorschläge.
3	Benutzer gibt in dem «Haltestelle:» Feld Buchstaben ein.	Es kommen bei jedem eingegebenen Buchstaben wieder neu geladen die besten Vorschläge.
4	Benutzer macht Doppelklick auf einen Eintrag in der Liste.	Der Doppelgeklickte Eintrag soll in die TextBox übernommen werden.

4.2 Testfall 2 – Verbindungen anzeigen

Bedingung ist, dass das Programm gestartet ist und die Vorherigen Tests funktionieren.

Schritt Nr.	Aktivität	Erwartetes Ereignis
1	Benutzer gibt in die beiden Textboxen «Von:» & «Nach:» eine Haltestelle ein, entweder korrekt oder er wählt sie in der Liste an.	Die Haltestellen sind korrekt in der TextBox enthalten.
2	«Verbindung Suchen» Button wird gedrückt.	In der DataGridView sollten die nächsten vier Verbindungen angezeigt werden.

4.3 Testfall 3 – Haltestellen Abfahrtstafel anzeigen

Bedingung ist, dass das Programm gestartet ist und die Vorherigen Tests funktionieren.

Schritt Nr.	Aktivität	Erwartetes Ereignis
1	Benutzer gibt in der Textbox «Haltestelle:» eine Haltestelle ein, entweder korrekt oder er wählt sie in der Liste an.	Die Haltestelle ist korrekt in der TextBox enthalten.
2	«Haltestellen Fahrplan Suchen» Button wird gedrückt.	Nächste Abfahrten werden angezeigt.
3	«Haltestellen Fahrplan Suchen» Button wird gedrückt.	DataGridView soll die Kolonnenheader auf die beschränken, die es braucht bei der Haltestellenabfrage. (Ort, Abfahrtszeit und Zone)

4.4 Testfall 4 - Von: & Nach: Einträge wechseln

Bedingung ist, dass das Programm gestartet ist und die Vorherigen Tests funktionieren.

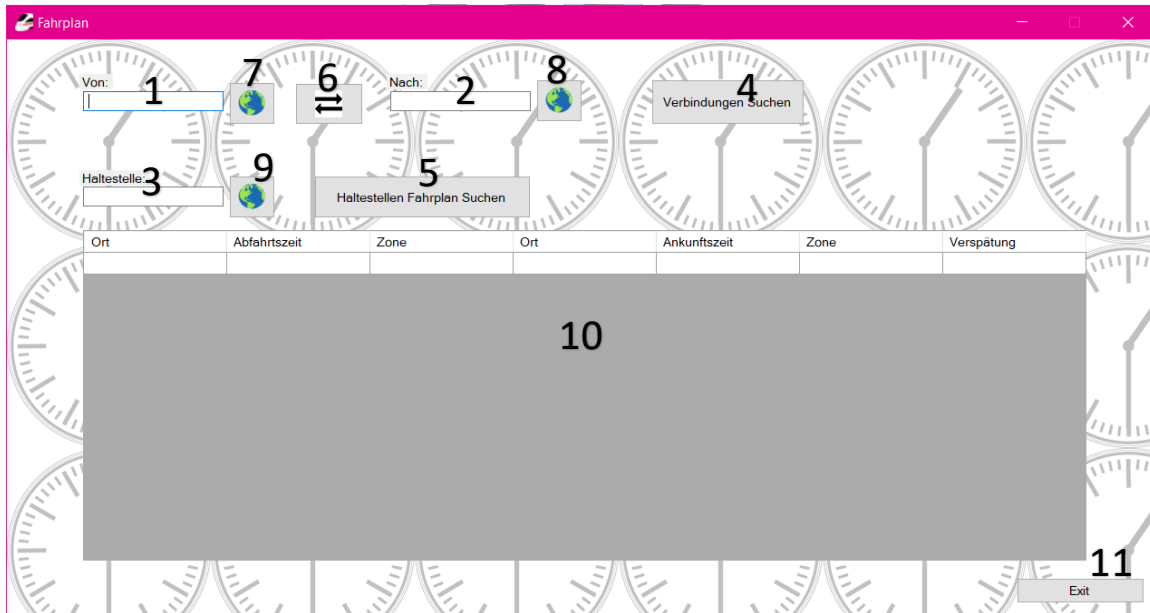
Schritt Nr.	Aktivität	Erwartetes Ereignis
1	Benutzer gibt in die beiden Textboxen «Von:» & «Nach:» eine Haltestelle ein, entweder korrekt oder er wählt sie in der Liste an.	Die Haltestellen sind korrekt in der TextBox enthalten.
2	Benutzer klickt auf den Button mit den beiden Pfeilen zwischen den beiden Feldern.	Die beiden Einträge werden gewechselt.

4.5 Testfall 5 – Haltestellen auf der Map anzeigen lassen.

Bedingung ist, dass das Programm gestartet ist und die Vorherigen Tests funktionieren.

Schritt Nr.	Aktivität	Erwartetes Ereignis
1	Benutzer gibt in eine beliebige der drei TextBoxen eine Haltestelle ein.	Die Haltestelle ist korrekt in der TextBox enthalten.
2	Benutzer klickt den Button, der neben der ausgefüllten TextBox ist an.	Es sollte sich ein neuer Browser Tab oder falls noch kein Fenster offen ist ein neues Fenster öffnen in dem man auf Google Maps genau bei der Haltestelle ist.
3	Gleiches Vorgehen bei den anderen Feldern auch Probieren.	Müsste überall genau gleich funktionieren.

5 BESCHREIBUNG DES GUI



Nr.	Beschreibung
1	Text um die AbfahrtsHaltestelle einzugeben
2	Text um die AnkunftsHaltestelle einzugeben
3	Text um die Haltestelle einzugeben, die für die Abfahrtstafel benötigt wird
4	Verbindung zwischen Abfahrts- und AnkunftsHaltestelle finden
5	Haltestellen abfahrtsplan anzeigen
6	Abfahrts- und AnkunftsHaltestelle wechseln
7	AbfahrtsHaltestelle auf Google Maps anzeigen
8	AnkunftsHaltestelle auf Google Maps anzeigen
9	AbfahrtsplanHaltestelle auf Google Maps anzeigen
10	DataGridView wo die Verbindungen und der Abfahrtsplan angezeigt werden
11	Exit-Button, mit dem das Programm geschlossen werden kann.

6. INSTALLATION / DEINSTALLATION

6.1 Installation

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

Man muss dafür einfach auf den Download Button klicken. Wenn man dann die VisualStudio Solution einmal gestartet hat, kann man in seinem Projektordner (Standard Pfad: C:\Users\vmadmin\Documents\Visual Studio 2017\Projects\ProjektM318\modul-318-student) dort muss man weiter auf \ProjektM318\bin\Debug dort findet man eine .exe Datei. Die kann man dann kopieren und überall benutzen als eigenständiges Programm benutzen.

6.2 Deinstallation

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