Autor: Luca Aeberhard

Abgabedatum: 01.04.2021

Entwicklungsort: Adligenswil

M318-Dokumentation

Luca Aeberhard

Inhalt

[Einleitung 2](#_Toc68180005)

[Zweck des Dokuments 2](#_Toc68180006)

[Mockups 2](#_Toc68180007)

[User Stories 4](#_Toc68180008)

[Aktivitätsdiagramm 6](#_Toc68180009)

[Testfälle / Testprotokoll 7](#_Toc68180010)

[Testprotokoll 7](#_Toc68180011)

[Testfall "Nach Stationen suchen" 7](#_Toc68180012)

[Testfall "Nach Verbindungen suchen" 8](#_Toc68180013)

[Testfall "Abfahrtstafel anzeigen" 8](#_Toc68180014)

[Testfall "Kartenansicht" 9](#_Toc68180015)

[Testfall "Verbindung per Mail senden" 9](#_Toc68180016)

[Testfall "Stationen switchen" 10](#_Toc68180017)

[Installationsanleitung 10](#_Toc68180018)

# Einleitung

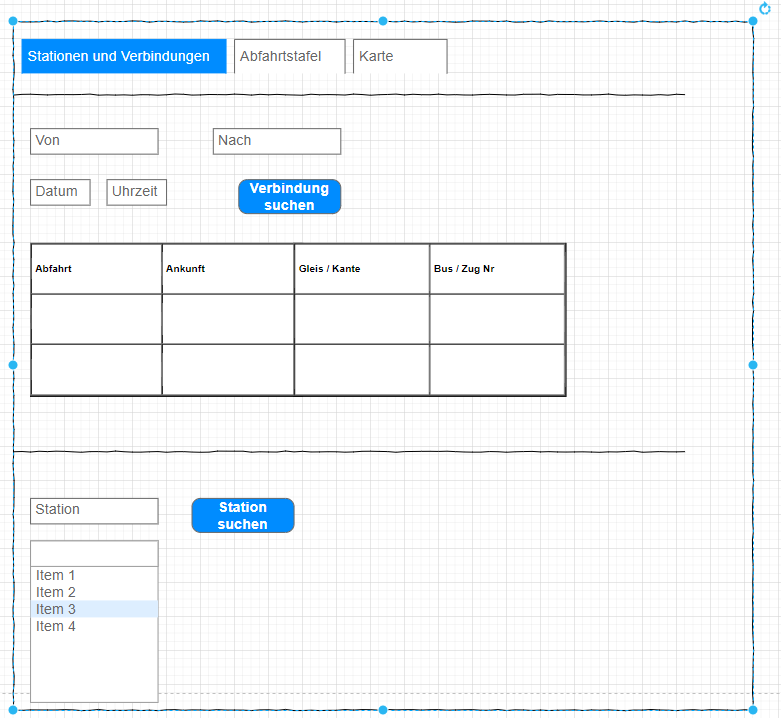
Dieses Projekt wurde im ÜK Modul 318 erarbeitet. Das Ziel war es, eine SBB ähnliche Applikation zu erstellen, mit welcher man z.B. nach Verbindungen von öffentlichen Verkehrsmitteln, Verbindungen ab bestimmten Stationen und Stationen in der Nähe des Benutzers suchen kann. Wir hatten für die Programmierung des Programms 3 Tage Zeit.

# Zweck des Dokuments

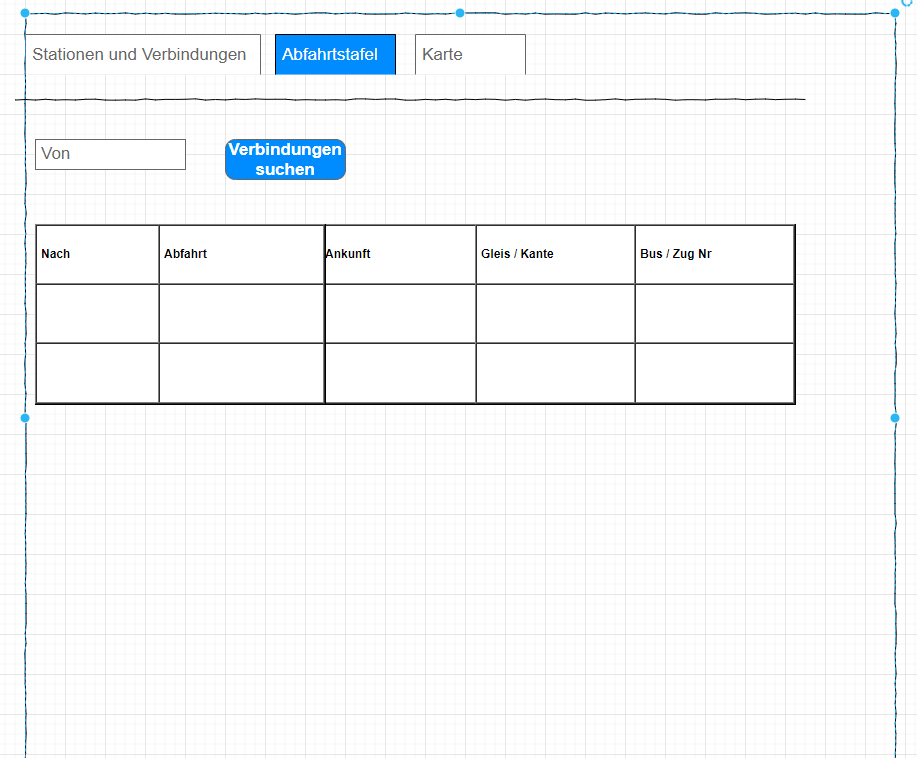
Dieses Dokument soll dazu dienen, die Dokumentation für den ÜK 318 darzustellen. Dabei soll die Planung mit Mockups und User-Storys vorhanden sein. Zudem sollen Testfälle und Testprotokolle im Dokument enthalten sein. Zusätzlich beinhaltet dieses Dokument eine kurze Installationsanleitung für das Programm.

# Mockups

Die Mockups dienen dazu, einen ersten Entwurf des Graphical User Interface darzustellen. Dabei macht man sich schon erste Gedanken, welche Komponenten (Textboxen, DataGridView usw.) man verwenden möchte und wie diese ungefähr angeordnet sein werden. Da dies nur Entwürfe sind, kann das richtige Produkt am Ende Abweichungen enthalten, jedoch sollte es grob den Mockups entsprechen. Die Aufgabe war es, mindestens alle Komponenten der Prio 1 darzustellen.  
  
Dies ist die erste Seite der Applikation, auf welcher man landet, wenn man diese startet. Oben hat man dann eine Tab-Navigation, mit welcher man durch die verschiedenen Seiten navigieren kann. Darunter hat man eine Verbindungssuche, bei welcher man eine Start- und Endstation eingeben kann. Die Ergebnisse werden dann in einer DataGridView dargestellt. Ganz unten auf der Seite hat man noch eine einfache Stationssuche.



Auf der zweiten Seite der Applikation befindet sich die Abfahrtstafel. Bei dieser kann man in einer Textbox die Startstation eingeben und in einer DataGridView werden danach alle Verbindungen, welche von dieser Station ausgehen aufgelistet.



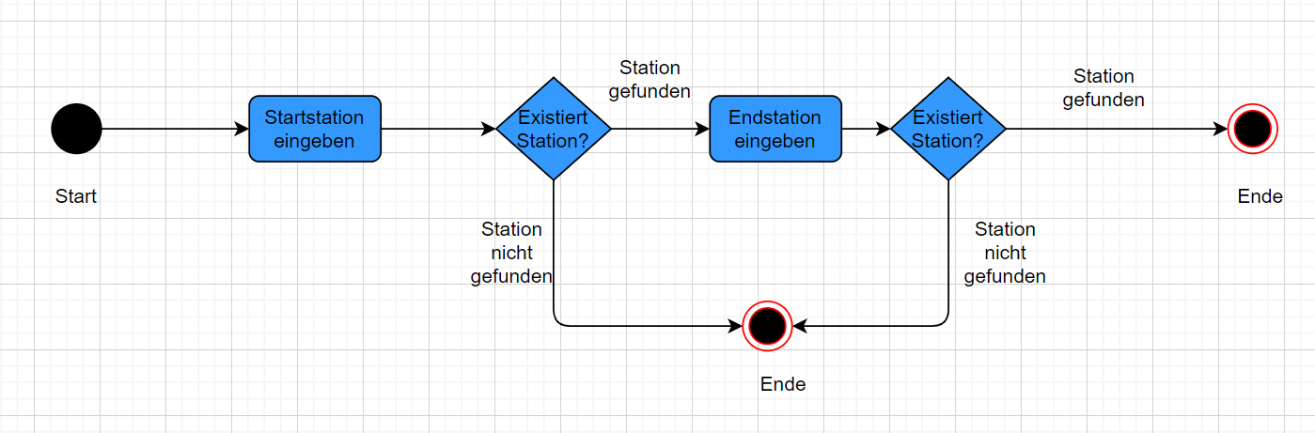
# User Stories

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Titel | User-Storys | Abnahmekriterien | Priorität | Umgesetzt Ja / Nein |
| A01 | Stationen suchen | Als Nutzer möchte ich nach einer Startstation und einer Endstation suchen, um eine Suche der Verbindungen vorzubereiten. | Es sollen zwei Stationen in je ein Textfeld eingegeben (eine Start- und eine Endstation) werden können. | 1 | Ja |
| A02 | Verbindungen anzeigen | Als Nutzer möchte ich beim Suchen nach Verbindungen sicher die nächsten 4 angezeigt bekommen, um mir eine passende Fahrt auszusuchen | Nach der Eingabe der Start- und Endstation sollen mindestens 4 Verbindungen in einer DataGridView angezeigt werden. | 1 | Ja |
| A03 | Vorschläge | Als Nutzer möchte ich, dass beim Eintippen der Station bereits Vorschläge auftauchen, um nicht den ganzen Namen der Station wissen zu müssen. | Beim Eintippen einer Station sollen für die bereits eingegebenen Wörter Vorschläge vorgeschlagen werden. | 2 | Ja |
| A04 | Abfahrtstafel | Als Nutzer möchte ich eine Abfahrtstafel angezeigt bekommen, um von dort aus, die nächsten Verbindungen zu sehen. | Wenn eine Station eingegeben wird, erscheinen die nächsten Verbindungen, welche von dieser Station ausgehen. | 1 | Ja |
| A05 | Datum und Uhrzeit | Als Nutzer möchte ich ein Datum und die Uhrzeit angeben können, um Verbindungen in der Zukunft zu suchen. | Man kann ein Datum und eine Uhrzeit eingeben und es werden die Verbindungen welche dann fahren angezeigt. | 2 | Nein |
| A06 | Kartenansicht | Als Nutzer möchte ich auf einer Karte die gewünschte Station angezeigt bekommen, um zu sehen wo sich diese befindet. | Wenn eine Station eingegeben wird hat man die Option diese auf der Karte anzuzeigen. | 3 | Ja |
| A07 | Nächste Stationen ab Standort | Als Nutzer möchte ich vom aktuellen Standort die nächsten Stationen meiner Fahrlinie angezeigt bekommen, um zu wissen an welchen Stationen ich vorbei muss. | Es werden die nächsten Stationen vom aktuellen Standort angezeigt. | 3 | Nein |
| A08 | Verbindung per Mail | Als Nutzer möchte ich eine Verbindung jemand anderem per Mail zuschicken, um die Verbindung zu teilen. | Wenn man zwei Stationen eingegeben hat kann man die oberste Station auf der Liste jemanden per Mail schicken. | 3 | Ja |
| A09 | Stationen switchen | Als Nutzer möchte ich einen Button mit welchem ich die Start- und die Endstation miteinander vertauschen kann, um diese nicht löschen und neu eingeben zu müssen. | Wenn man den vorgesehenen Button anwählt wird die Station im Startstation-Feld ins Endstation-Feld eingetragen und umgekehrt. | 3 | Ja |

# Aktivitätsdiagramm

Das Aktivitätsdiagramm dient dazu, den Ablauf einer User-Story graphisch nach UML darzustellen. Dies hilf später beim Umsetzen. Bei einem Aktivitätsdiagramm wird der Start und das Ende mit einem Punkt dargestellt. Abgerundete Rechtecke symbolisieren die Aktionen und Rauten stellen Verzweigungen dar. Das Ziel dieser Aufgabe war es ein Aktivitätsdiagramm für eine User Story mit der höchsten Priorität darzustellen.

Das unten abgebildete Aktivitätsdiagramm stellt die User Story A01 dar, bei welcher man nach einer Start- und Endstation suchen soll.



# Testfälle / Testprotokoll

## Testprotokoll

Dokument Version: 1.6

Durchführungsdatum: 01.04.2021

Tester/in: Vangelisti Alessio

App Version & Umgebung: 1.5 (DEV)

## Testfall "Nach Stationen suchen"

In diesem Testfall wird die Suche nach einer Station geprüft. Zudem wird auch getestet ob die Autovervollständigung funktioniert.

Getestete Anforderungen: A01, A03

Vorbedingung: Das Programm läuft und der Benutzer befindet sich auf der ersten Seite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt Id | Aktivität | Erwartetes Resultat | Abw. Resultat | Erfüllt |
| 1 | Der User tippt "L" in das "von-Textfeld" ein. | Es werden Stationen, welche mit L beginnen oder ein L im Namen enthalten angezeigt. |  | x |
| 2 | Der User klickt mit der Maus eine Station an. | Die Station steht nun im Startstation Textfeld. |  | x |

## Testfall "Nach Verbindungen suchen"

In diesem Testfall wird die Suche nach Verbindungen getestet.

Getestete Anforderungen: A02, A03

Vorbedingungen: Das Programm läuft und der Benutzer befindet sich auf der ersten Seite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt Id | Aktivität | Erwartetes Resultat | Abw. Resultat | Erfüllt |
| 1 | Der User wählt als "From Station" Ebikon aus und als "To Station" Stuben aus. | Es passiert noch nichts. |  | x |
| 2 | Der User drückt auf den "Verbindungen suchen" Button. | In der Tabelle unten werden nun die Verbindungen mit Abfahrt, Ankunft, Dauer, und Kante/Gleis zwischen den 2 Stationen angezeigt. |  | x |

## Testfall "Abfahrtstafel anzeigen"

In diesem Testfall wird die Anzeige der Abfahrtstafel getestet.

Getestete Anforderungen: A04, A03

Vorbedingungen: Das Programm läuft und der Benutzer befindet sich auf der zweiten Seite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt Id | Aktivität | Erwartetes Resultat | Abw. Resultat | Erfüllt |
| 1 | Der User wählt als Station Ebikon. | Es passiert noch nichts. |  | x |
| 2 | Der User drückt auf den "Verbindungen anzeigen" Button. | In der Tabelle werden nun alle Verbindungen, welche von der Station Ebikon ausgehen angezeigt. Dabei sieht man den Ankunftsort, die Abfahrt und den Bus / Zug. |  | x |

## Testfall "Kartenansicht"

In diesem Testfall wird die Kartenansicht einer Station getestet.

Getestete Anforderungen: A06

Vorbedingungen: Das Programm läuft und der Benutzer befindet sich auf der ersten Seite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt Id | Aktivität | Erwartetes Resultat | Abw. Resultat | Erfüllt |
| 1 | Der User gibt bei der Stationen-Suche Ebikon ein. | Es passiert noch nichts. |  | x |
| 2 | Der User drückt nun den Button "Auf Karte anzeigen". | Die Webseite "map.search.ch" wird aufgerufen und auf der Karte wird die Station Ebikon angezeigt. |  | x |

## Testfall "Verbindung per Mail senden"

In diesem Testfall wird die Versendung einer Verbindung per Mail getestet.

Getestete Anforderungen: A08

Vorbedingungen: Das Programm läuft und der Benutzer befindet sich auf der ersten Seite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt Id | Aktivität | Erwartetes Resultat | Abw. Resultat | Erfüllt |
| 1 | Der User gibt als "Start-Station" Ebikon und als "Endstation" Stuben ein und drückt auf den Verbindungen anzeigen Button. | Es werden aller Verbindungen von Ebikon nach Stuben angezeigt. |  | x |
| 2 | Der User gibt nun beim "An" Textfeld die Email [luca.aeberhard@gmx.net](mailto:luca.aeberhard@gmx.net) ein und drückt auf den "Mail senden" Button. | Outlook öffnet sich und ein Entwurf mit dem Titel "ÖV-Verbindung", dem Empfänger [luca.aeberhard@gmx.net](mailto:luca.aeberhard@gmx.net) und dem Inhalt mit der ersten Verbindung zwischen Ebikon und Stuben öffnet sich. |  | x |

## Testfall "Stationen switchen"

In diesem Testfall wird das Switchen der Start- und Endstation in der Verbindungen-Suche getestet.

Getestete Anforderungen: A09

Vorbedingungen: Das Programm läuft und der Benutzer befindet sich auf der ersten Seite.

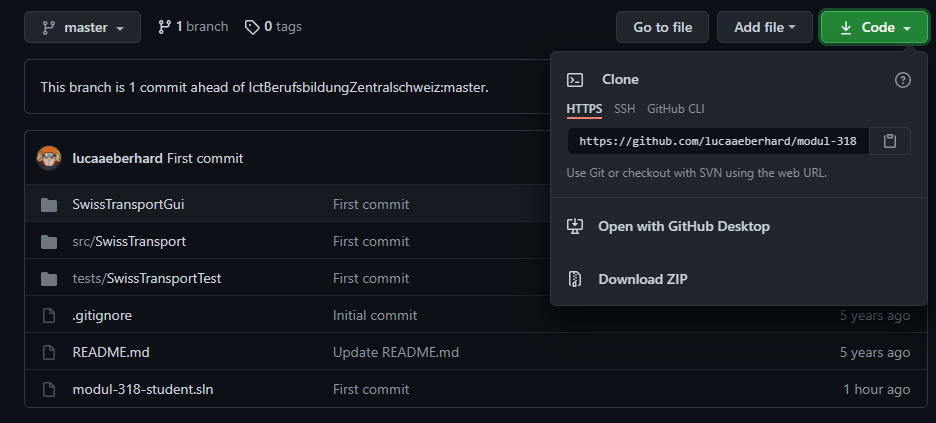
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt Id | Aktivität | Erwartetes Resultat | Abw. Resultat | Erfüllt |
| 1 | Der User gibt als "Start-Station" Ebikon und als "Endstation" Stuben ein. | Es passiert noch nichts. |  | x |
| 2 | Der User klickt nun auf den "<->" Button. | Im Startstation Textfeld steht nun Stuben und im Endstation Textfeld steht nun Ebikon. |  | x |

# Installationsanleitung

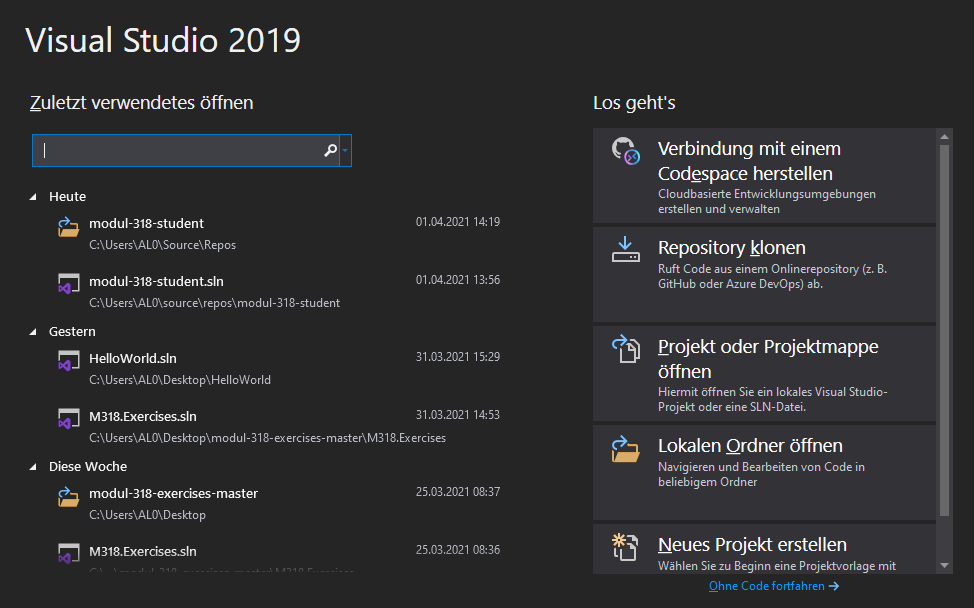
Schritt 1:

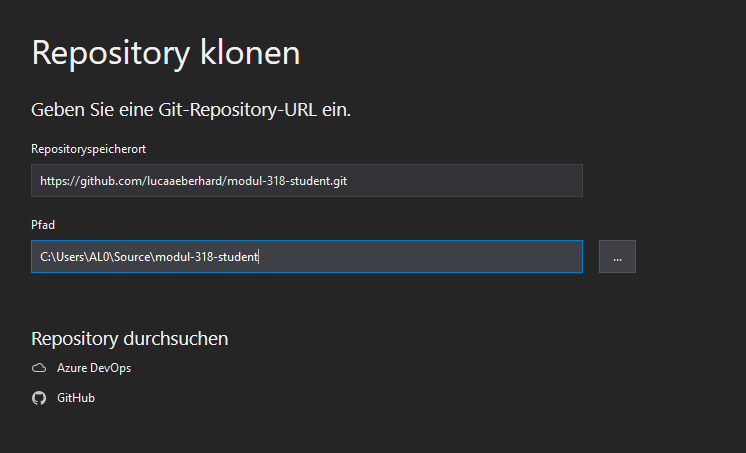
Um das Projekt herunterzuladen geht man als erstes auf das „modul-318 Projekt“ in meinem Git-Repo (<https://github.com/lucaaeberhard/modul-318-student>.)

Schritt 2:

Dort klickt man auf den grünen „Code“ Button. Bei dem Pop-up kopiert man nun diesen Link.  


Schritt 3:

Als nächstes öffnet man Visual Studio. Danach klickt man auf „Repository klonen“.  


Als nächstes erscheint dieses Fenster bei welchem man den kopierten Link einfügen muss.  


Schritt 4:

Als letztes wählt man „SwissTransportGui“ als Startprojekt.  
