1.4.2021

Vangelisti, Alessio

Komax ag

SBB Applikation

ÜK-M318

**Inhaltsverzeichnis**

[Einleitung 2](#_Toc68180204)

[Zweck des Dokuments 2](#_Toc68180205)

[Planung 3](#_Toc68180206)

[User Stories und Abnahmekriterien 3](#_Toc68180207)

[Mockup 5](#_Toc68180208)

[Aktivitätsdiagramm 5](#_Toc68180209)

[Tests 6](#_Toc68180210)

[Testprotokoll 6](#_Toc68180211)

[Testfälle 6](#_Toc68180212)

[Testfall "Verbindungen suchen" 6](#_Toc68180213)

[Testfall "Nach Stationen suchen" 7](#_Toc68180214)

[Testfall "Abfahrtstafel anzeigen" 7](#_Toc68180215)

[Testfall "Station auf Karte" 7](#_Toc68180216)

[Testfall "Verbindung per Mail senden" 8](#_Toc68180217)

[Testfall "Stationen switchen" 8](#_Toc68180218)

[Installation 9](#_Toc68180219)

# Einleitung

In diesem Projekt geht es um das Erledigen einer Applikation, welche ähnlich ist, wie die SBB-App oder die ÖV Live App. Jeder Lernende hat rund drei Tage Zeit eine Applikation mit enthaltener Dokumentation zu liefern. Dies wird anschliessend nach einem bekannten Kriterienraster bewertet und gibt somit eine Note, welche ein Teil zur Abschlussnote zählt.

# Zweck des Dokuments

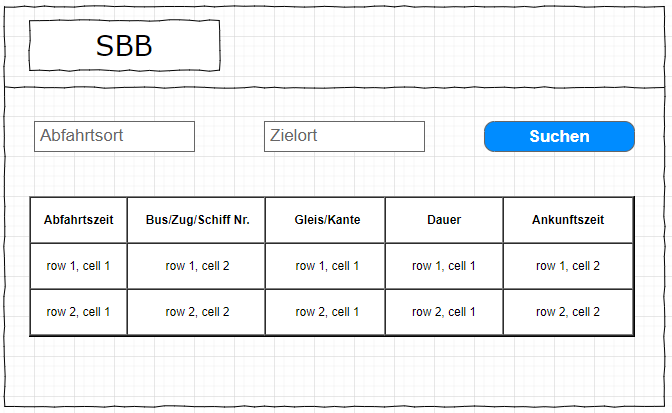
Dieses Dokument dient als Dokumentation für die SBB Applikation im ÜK Modul 318. Dabei enthalten ist zum einen die Planung mit enthaltenen User Stories und den dazugehörigen Abnahmekriterien. Ebenso enthält die Planung auch Mockups in Form von Wireframes welche als Grobplanung für die Benutzeroberfläche gilt. Anschliessend werden alle Testfälle, die durchgeführt wurden, gründlich protokolliert.

# Planung

## User Stories und Abnahmekriterien

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Titel** | **Abnahmekriterium** | **User Story** | **Prio** | **X** |
| 1 | Verbindungen suchen | Es sollen zwei Stationen in je ein Textfeld eingegeben werden (Startstation und Endstation) können. | Als Benutzer möchte ich Start und Endstationen angeben, um eine Suche der Fahrpläne vorzubereiten. | 1 | X |
| 2 | Nächste Verbindungen anzeigen | Beim Bedienen des Suchknopfes sollen die zwei Eingabewerte der Stationen ausgewertet und die Fahrpläne in einer DataGridView angezeigt werden. | Als Benutzer möchte ich beim Suchen einer Verbindung mindestens die nächsten vier Verbindungen angezeigt bekommen, um mir eine passende Abfahrtszeit auszusuchen. | 1 | X |
| 3 | Abfahrtstafel | Sobald nur eine Startstation eingegeben wird, sollen die darauffolgenden Fahrpläne von dieser Station als Abfahrtstafel ausgegeben werden. | Als Benutzer möchte ich eine Abfahrtstafel angezeigt bekommen, um von dort aus, die nächsten Verbindungen zu sehen. | 1 | X |
| 4 | Stationen Suchen | Es besteht die Möglichkeit eine Station zu suchen, welche mithilfe von Autofill dem Benutzer hilft. | Als Benutzer möchte ich eine Station suchen können, damit ich weiss, von wo aus ich abfahren muss. |  | X |
| 5 | Vorschläge | Beim Eingeben einer Station sollen Vorschläge für die bereits eingegeben Buchstaben ausgewertet und vorgeschlagen werden. | Als Benutzer möchte ich, während eine Station eingegeben wird, bereits Vorschläge angezeigt bekommen, um die Station nicht ausschreiben zu müssen oder ich Vorschläge bekomme, falls ich den Namen der Station nicht ganz kenne. | 2 | X |
| 6 | Datum und Uhrzeit | Die Suche der Fahrpläne kann von dem Benutzer anhand eines Datums und einer Uhrzeit angepasst werden. Es werden dann die Verbindungen ab diesem Zeitpunkt angezeigt werden. | Als Benutzer möchte ich Datum und Uhrzeit angeben können, um Verbindungen in einer späteren Zeit zu suchen. | 2 |  |
| 7 | Karte | Sobald eine Station gesucht wird, besteht die Möglichkeit die Station auf einer Karte zu sehen. | Als Benutzer möchte ich eine Karte angezeigt bekommen, um zu sehen, wo sich die Station befindet. | 3 | X |
| 8 | Nächste Stationen | Es werden die nächsten Stationen vom aktuellen Standort angezeigt. | Als Benutzer möchte ich vom aktuellen Standort aus, die nächsten Stationen anzeigen, damit ich weiss welche Stationen in der Nähe sind. | 3 |  |
| 9 | Verbindungen Mailen | Sobald eine passende Fahrlinie gefunden wurde, kann man sie sich per Mail zukommen lassen. | Als Benutzer möchte ich eine ausgewählte Verbindung per Mail weiterleiten, um jemand darüber zu informieren. | 3 | X |
| 10 | Verbindungen wechseln | Wenn zwei Stationen eingegeben wurden, können diese anhand eines Knopfes vertauscht werden. | Als Benutzer möchte ich über einen Pfeil den Abfahrtsort und Ankunftsort wechseln können, um schnell die Verbindungen in die gegenliegende Richtung zu bekommen. | 3 | X |

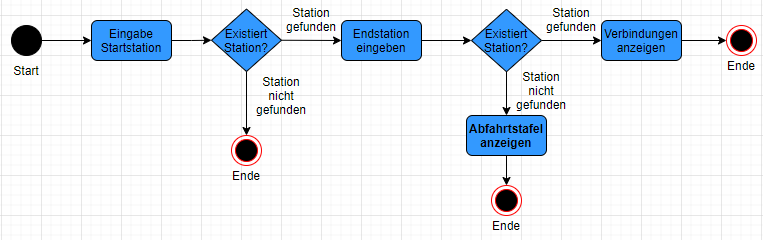
## Mockup



Dieses Mockup erfüllt alle Anforderungen der Priorität 1. Hier kann der User einen gewünschten Abfahrtsort wie auch einen gewünschten Zielort eingeben. Beim Betätigen des «Suchen»-Knopfes wird die Eingabe ausgewertet und dann als Tabelle im DataGridView ausgegeben. Falls der User jedoch nur einen Abfahrtsort eingibt, dann wird nur dieser ausgewertet und als Abfahrtstafel angezeigt.

## Aktivitätsdiagramm

Aktivitätsdiagramm zur User Story Abfahrtstafel (ID: 3)



Dieses Ablaufdiagramm ist so aufgebaut, dass anhand der Userbedienung etwas anderes ausgeführt werden kann. Wenn der User beim Suchen einen Start und Zielort eingibt, dann werden ganz normal die Verbindungen gesucht. Sollte der User jedoch nur einen Startort eingeben, dann wird er von diesem Ort aus, eine Abfahrtstafel angezeigt bekommen.

# Tests

## Testprotokoll

Dokument Version: 3.4

Durchführungsdatum: 01.04.2021

Tester/in: Luca Aeberhard (Suva)

App Version & Umgebung: 3.8 (DEV)

## Testfälle

Hier werden alle eingebauten Features und Funktionen, welche von Luca Aeberhard getestet wurden, dokumentiert. Alle Tests haben am 01.04.2021 stattgefunden

### Testfall "Verbindungen suchen"

Getestete Abnahmekriterien:

1. Es sollen zwei Stationen in je ein Textfeld eingegeben werden (Startstation und Endstation) können.

2. Beim Bedienen des Suchknopfes sollen die zwei Eingabewerte der Stationen ausgewertet und die Fahrpläne in einer DataGridView angezeigt werden.

Getestete Elemente:

textBoxStartStation, textBoxEndStation, buttonSearchSonnections, dataGridViewConnections

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Aktivität** | **Gewünschtes Resultat** | **Erfüllt** |
| 1 | Benutzer gibt Start und Endstation ein | Nichts | X |
| 2 | Benutzer betätigt Suchknopf | Verbindungen der zwei Stationen werden im dataGridViewConncections angezeigt. | X |

### Testfall "Nach Stationen suchen"

Getestete Abnahmekriterien:

1. Es besteht die Möglichkeit eine Station zu suchen, welche mithilfe von Autofill dem Benutzer hilft.

Getestete Elemente:

textBoxStation, listBoxStations

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Aktivität** | **Gewünschtes Resultat** | **Erfüllt** |
| 3 | Benutzer gibt Buchstabe ein | In der listBoxStations werden alle Stationen angezeigt, welche mit dem Buchstaben anfangen oder ihn enthalten. | X |

### Testfall "Abfahrtstafel anzeigen"

Getestete Abnahmekriterien:

1. Sobald nur eine Startstation eingegeben wird, sollen die darauffolgenden Fahrpläne von dieser Station als Abfahrtstafel ausgegeben werden.

Getestete Elemente:

textBoxStartStation, textBoxEndStation, buttonSearchSonnections, dataGridViewConnections

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Aktivität** | **Erwartetes Resultat** | **Erfüllt** |
| 4 | Der Benutzer gibt nur einen Abfahrtsort ein ohne Ankunftsort. | Im DataGridView werden die nächsten Verbindungen von dieser Station aus angezeigt. | X |

\*Die Abfahrtstafel ist auch erreichbar, wenn der Benutzer eine Station sucht und diese anschliessend doppelklickt.

### Testfall "Station auf Karte"

Getestete Abnahmekriterien:

1. Sobald eine Station gesucht wird, besteht die Möglichkeit die Station auf einer Karte zu sehen.

Getestete Elemente:

pictureBoxSearchOnMap, textBoxStation

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Aktivität** | **Erwartetes Resultat** | **Erfüllt** |
| 5 | Der Benutzer sucht eine Station und klickt auf das Kartenlogo rechts daneben. | Der Browser öffnet sich und zeigt automatisch den Standort der Station an. | X |

### Testfall "Verbindung per Mail senden"

Getestete Abnahmekriterien:

1. Sobald eine passende Fahrlinie gefunden wurde, kann man sie sich per Mail zukommen lassen.

Getestete Elemente:

pictureBoxSendMail, textBoxReceiverAddress, buttonSendMail

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Aktivität** | **Erwartetes Resultat** | **Erfüllt** |
| 6 | Der Benutzer gibt nur sucht eine Verbindung und klickt anschliessend auf das Mail Icon. | Es öffnet sich das Standart Mailprogramm und Subject, Receiver und Content sind enthalten. | X |

### Testfall "Stationen switchen"

Getestete Abnahmekriterien:

1. Wenn zwei Stationen eingegeben wurden, können diese anhand eines Knopfes vertauscht werden.

Getestete Elemente:

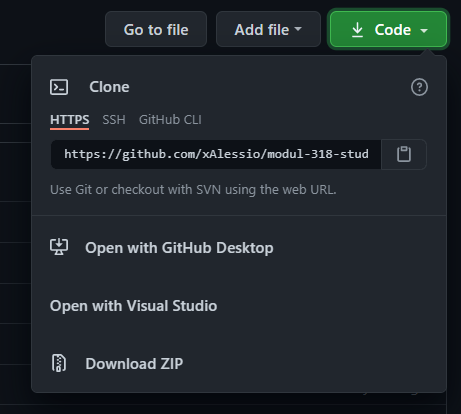
textBoxStartStation, textBoxEndStation, buttonSwitchStations

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Aktivität** | **Erwartetes Resultat** | **Erfüllt** |
| 6 | Der Benutzer gibt nur einen Abfahrtsort und Ankunftsort ein. Anschliessend klickt er auf den «<=>» Knopf. | Die Werte von Start- und Endstation werden getauscht. | X |

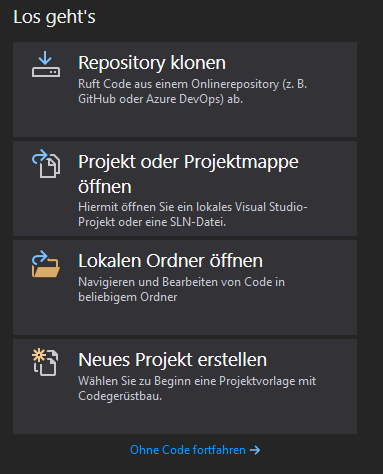
# Installation

Hier werden alle Schritte beschrieben, um das Projekt in Visual Studio zu öffnen.

1. Den Link öffnen: <https://github.com/xAlessio/modul-318-student>

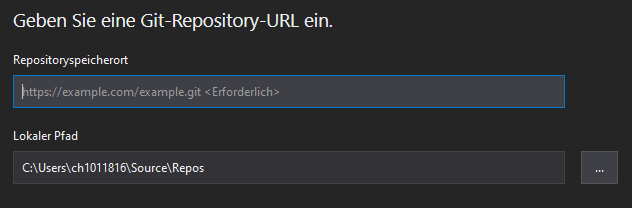
2.

Auf den grünen Knopf drücken und den https-Link kopieren.

3.

Visual Studio starten und auf Repository klonen klicken.

4.



Zuletzt den Link einfügen und unten einen Pfad angeben, wo das Repository lokal gespeichert werden soll.