Dokumentation

SBB Desktop Applikation

Joël Tschopp

Urs Nussbaumer

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc34898859)

[2 Zweck 4](#_Toc34898860)

[3 Funktionen 4](#_Toc34898861)

[4 Mockup 4](#_Toc34898862)

[5 Use Cases 6](#_Toc34898863)

[6 Flussdiagramme 8](#_Toc34898864)

[7 Programmrichtlinien 9](#_Toc34898865)

[8 Testfälle 10](#_Toc34898866)

[9 Testprotokoll 12](#_Toc34898867)

[10 Installationsanleitung 14](#_Toc34898868)

# Einleitung

Diese Dokumentation habe ich erfasst, als ich den ÜK 318 im Rahmen meiner Ausbildung besucht habe. Diese Dokumentation handelt über ein Programm, welches ich in diesem ÜK programmiert habe. Dieses Programm ist eine kleine SBB App. In der Applikation, kann der User 2 Stationen eingeben und dann werden ihm die Verbindungen zwischen den beiden Stationen aufgelistet.  
Dann hat der User auch noch die Möglichkeit auf eine Abfahrtstafel zuzugreifen. In dieser kann der User eine Station eingeben. Wenn er dann nach denn Verbindungen sucht, werden ihm alle Verbindungen ab dieser Haltestelle aufgelistet.

# Zweck

Der Zweck dieses Dokuments ist, dass man sieht, was umgesetzt wurde, was nicht funktioniert und die Planung.

Das Dokument, dient auch dazu, dass der User sich ein bisschen besser zurechtfindet im Programm.

# Funktionen

***Funktionen die umgesetzt wurden:***

A001: Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.

A002: 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.

A003: Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann.

A005: ALs ÖV-Benutzer möchte ich nicht nur aktuelle Verbindungen suchen können, sondern auch solche zu einem beliebigen anderen Zeitpunkt, damit ich zukünftige Reisen planen kann.

A006: Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht.

A007: Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann (hat nur mässig funktioniert).

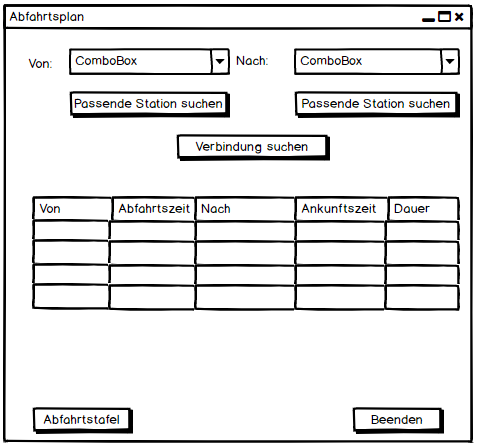
***Funktionen die nicht umgesetzt wurden:***

A004: Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such Resultate erscheinen, damit ich effizienter nach Stationen suchen kann.

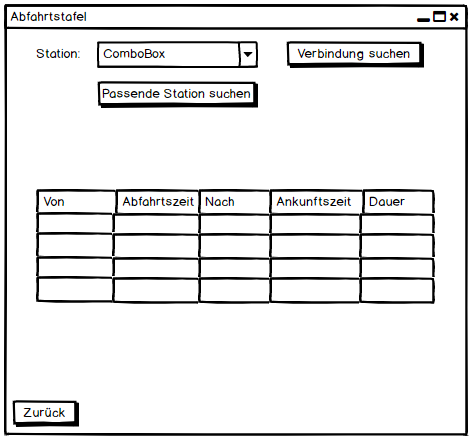
A008: Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können.

# Mockup

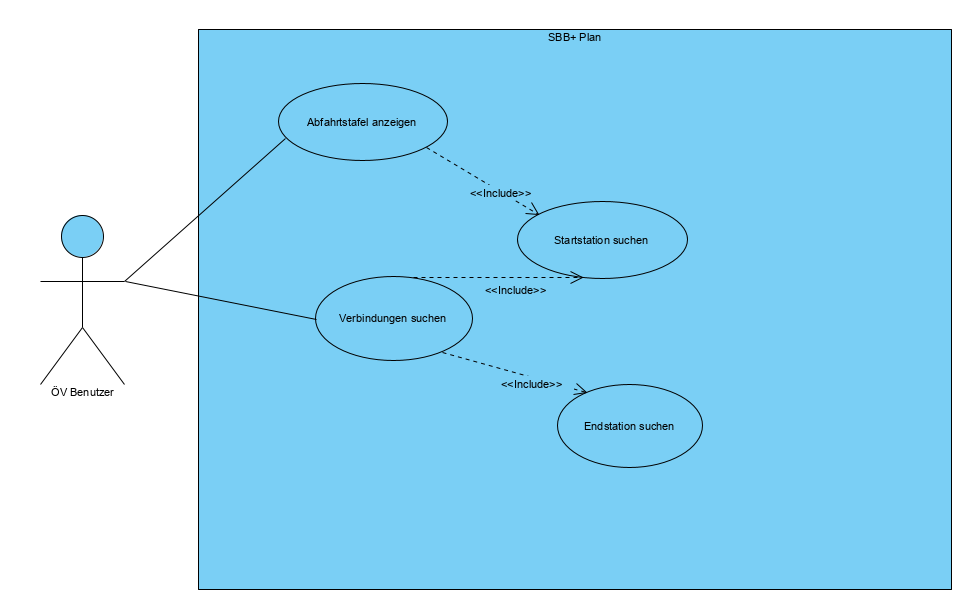
Main Form / Abfahrtsplan:



Abfahrtstafel:



# Use Cases



|  |  |
| --- | --- |
| Use Case | 1, Abfahrtstafel anzeigen |
| Beschreibung | ÖV Benutzer möchte die Abfahrtstafel sehen, damit er sehen kann, wann welcher Bus / Zug Fahrt auf welchem Gleis. |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss Startstation eingeben |
| Ablauf | 1. Programm öffnen 2. Auf Button Abfahrtsplan klicken 3. Startstation eingeben 4. Auf Button Verbindung suchen klicken |
| Ergebnis | Verbindungen in der nächsten Zeit von dieser Station werden aufgelistet |

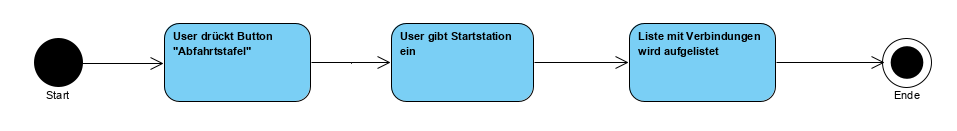
|  |  |
| --- | --- |
| Use Case | 2, Startstation suchen |
| Beschreibung | ÖV Benutzer möchte nicht den ganzen Namen der Startstation eingeben, weshalb er auf einen Button klicken kann dem Ihm alle Stationen vorschlägt, die mit seiner Eingabe anfängt |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss ein paar Buchstaben eingegeben haben, um das Suchergebnis einzuschränken |
| Ablauf | 1. Programm öffnen 2. Bei Startstation Teil des Ortes eintragen 3. Auf Button klicken |
| Ergebnis | Liste mit Stationen werden aufgelistet, welche mit dem User seiner Eingabe anfängt. |

|  |  |
| --- | --- |
| Use Case | 3, Endstation suchen |
| Beschreibung | ÖV Benutzer möchte nicht den ganzen Namen der Endstation eingeben, weshalb er auf einen Button klicken kann dem Ihm alle Stationen vorschlägt, die mit seiner Eingabe anfängt |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss ein paar Buchstaben eingegeben haben, um das Suchergebnis einzuschränken |
| Ablauf | 1. Programm öffnen 2. Bei Endstation Teil des Ortes eintragen 3. Auf Button klicken |
| Ergebnis | Liste mit Stationen werden aufgelistet, welche mit dem User seiner Eingabe anfängt. |

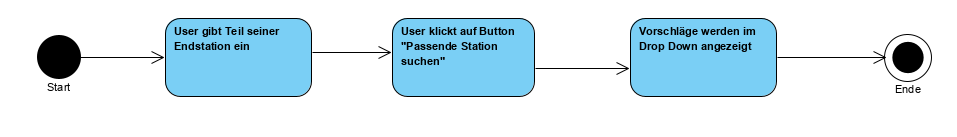
|  |  |
| --- | --- |
| Use Case | 4, Verbindungen suchen |
| Beschreibung | ÖV Benutzer möchte alle Verbindungen zwischen 2 Stationen angezeigt bekommen |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss Startstation eingeben und Endstation |
| Ablauf | 1. Programm öffnen 2. Startstation eingeben 3. Endstation eingeben 4. Button klicken |
| Ergebnis | Liste mit Verbindungen zwischen den beiden Stationen werden aufgelistet |

# Flussdiagramme

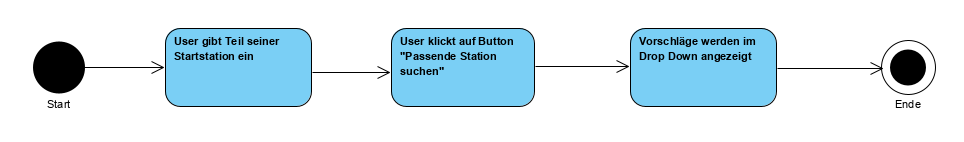
Abfahrtstafel Anzeigen:



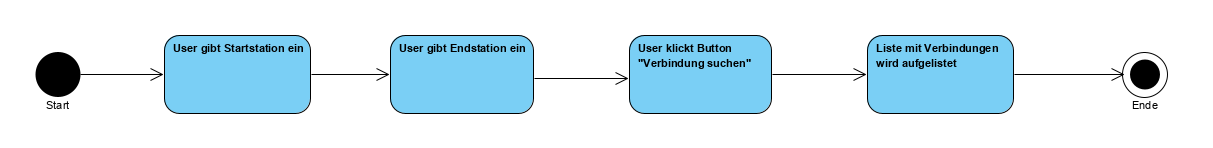
Endstation finden:



Startstation finden:



Verbindung suchen:



# Programmrichtlinien

**Namensgebung von Variablen, Eigenschaften, Methoden:**

* Globale Variabel mit Grossbuchstaben beginnen
* Variabel in Methoden mit Kleinbuchstaben beginnen & neues Wort mit Grossbuchstaben beginnen
* Methodennamen immer mit Grossbuchstaben beginnen

+

Man erkennt direkt um was es sich handelt

* Geschweifte Klammern immer auf neuer Zeile

+

Schöne Formatierung

Man erkennt direkt wo z.B. eine Methode endet

**Kommentare:**

* Wenig Kommentare
* Kommentare vor Funktion was sie kann / immer mit drei Schrägstrichen beginnen

+

Man hat eine kleine Einführung, was die Funktion kann

Einheitlich

**GUI-Controls**

* Control-Namen immer mit drei Buchstaben beginnen, welche aussagen was sie sind z.B.
  1. Button = btn
  2. Label = lbl
  3. Checkbox = cbx
  4. Radio Button = rbt
  5. Datagridview = dgv
  6. Textbox = txt
  7. Listbox = lbx
  8. Combobox = ddl

# Testfälle

Unterstützung beim Stationen suchen:

|  |  |  |
| --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** |
| ***1*** | User selektiert Combobox | Fokus auf Combobox |
| ***2*** | User gibt Teil seiner Daten ein   * Surse | Text wird in der Combobox angezeigt |
| ***3*** | User klickt auf Button: Passende Stationen anzeigen | Drop Down geht auf, mit Vorschlägen was der User für eine Station möchte |
| ***4*** | User wählt eine der Stationen aus | Station wird in der Combobox angezeigt |

Verbindung suchen:

|  |  |  |
| --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** |
| ***1*** | User selektiert Combobox für Startstation | Fokus auf linke Combobox |
| ***2*** | User gibt seine Daten ein   * Sursee, Bahnhof | Text wird in der linken Combobox angezeigt |
| ***3*** | User selektiert Combobox für Endstation | Fokus auf rechte Combobox |
| ***4*** | User gibt seine Daten ein   * Luzern | Text wird in der rechten Combobox angezeigt |
| ***5*** | User klickt den Button Verbindung suchen | Verbindungen zwischen den beiden Stationen werden im Datagridview angezeigt |

Abfahrtstafel Anzeigen:

|  |  |  |
| --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** |
| ***1*** | User klickt auf Button unten links «Abfahrtstafel» | Neues Form geht auf |
| ***2*** | User selektiert Combobox für die Startstation | Fokus auf Combobox |
| ***3*** | User gibt Daten ein   * Sursee, Bahnhof | Text wird in der Combobox angezeigt |
| ***4*** | User klickt den Button Verbindung suchen | Verbindungen ab der Startstation werden im Datagridview aufgelistet |

Ort auf Karte Anzeigen:

|  |  |  |
| --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** |
| ***1*** | User klickt auf Button unten links «Ort von Station» | Neues Form geht auf |
| ***2*** | User selektiert Combobox für die Station | Fokus auf Combobox |
| ***3*** | User gibt Daten ein   * Sursee, Bahnhof | Text wird in der Combobox angezeigt |
| ***4*** | User klickt den Button Station suchen | User wird auf Google Maps geschickt und sieht der Ort der Station vor sich |

Datum eingeben:

|  |  |  |
| --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** |
| ***1*** | User selektiert Combobox für Startstation | Fokus auf linke Combobox |
| ***2*** | User gibt seine Daten ein   * Sursee, Bahnhof | Text wird in der linken Combobox angezeigt |
| ***3*** | User selektiert Combobox für Endstation | Fokus auf rechte Combobox |
| ***4*** | User gibt seine Daten ein   * Luzern | Text wird in der rechten Combobox angezeigt |
| ***5*** | User gibt Datum und Zeit ein   * 15.03.2020 * 16:45 | Zeit wird im Datetimepicker angezeigt |
| ***6*** | User klickt den Button Verbindung suchen | Verbindungen zwischen den beiden Stationen ab dem 15.03 nach 16:45 Uhr werden im Datagridview angezeigt |

Nächste Haltestelle anzeigen

|  |  |  |
| --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** |
| ***1*** | User klickt auf Button unten links «Ort von Station» | Neues Form geht auf |
| ***2*** | User klickt auf Button «Station in der Nähe suchen» | Stationen in der Nähe werden im Drop Down angezeigt. |

# Testprotokoll

Unterstützung beim Stationen suchen:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** | ***Status*** |
| ***1*** | User selektiert Combobox | Fokus auf Combobox | √× |
| ***2*** | User gibt Teil seiner Daten ein   * Surse | Text wird in der Combobox angezeigt | √ |
| ***3*** | User klickt auf Button: Passende Stationen anzeigen | Drop Down geht auf, mit Vorschlägen was der User für eine Station möchte | √ |
| ***4*** | User wählt eine der Stationen aus | Station wird in der Combobox angezeigt | √ |

Verbindung suchen

|  |  |  |  |
| --- | --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** | ***Status*** |
| ***1*** | User selektiert Combobox für Startstation | Fokus auf linke Combobox | √ |
| ***2*** | User gibt seine Daten ein   * Sursee, Bahnhof | Text wird in der linken Combobox angezeigt | √ |
| ***3*** | User selektiert Combobox für Endstation | Fokus auf rechte Combobox | √ |
| ***4*** | User gibt seine Daten ein   * Luzern | Text wird in der rechten Combobox angezeigt | √ |
| ***5*** | User klickt den Button Verbindung suchen | Verbindungen zwischen den beiden Stationen werden im Datagridview angezeigt | √ |

Abfahrtstafel anzeigen

|  |  |  |  |
| --- | --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** | ***Status*** |
| ***1*** | User klickt auf Button unten links «Abfahrtstafel» | Neues Form öffnet sich | √ |
| ***2*** | User selektiert Combobox für die Startstation | Fokus auf Combobox | √ |
| ***3*** | User gibt Daten ein   * Sursee, Bahnhof | Text wird in der Combobox angezeigt | √ |
| ***4*** | User klickt den Button Verbindung suchen | Verbindungen ab der Startstation werden im Datagridview aufgelistet | √ |

Ort auf Karte Anzeigen:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** | ***Status*** |
| ***1*** | User klickt auf Button unten links «Ort von Station» | Neues Form öffnet sich | √ |
| ***2*** | User selektiert Combobox für die Station | Fokus auf Combobox | √ |
| ***3*** | User gibt Daten ein   * Sursee, Bahnhof | Text wird in der Combobox angezeigt | √ |
| ***4*** | User klickt den Button «Station suchen» | User wird auf Google Maps geschickt und sieht den Ort der Station vor sich | Marker nicht immer genau. |

Datum eingeben:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** | ***Status*** |
| ***1*** | User selektiert Combobox für Startstation | Fokus auf linke Combobox | √ |
| ***2*** | User gibt seine Daten ein   * Sursee, Bahnhof | Text wird in der linken Combobox angezeigt | √ |
| ***3*** | User selektiert Combobox für Endstation | Fokus auf rechte Combobox | √ |
| ***4*** | User gibt seine Daten ein   * Luzern | Text wird in der rechten Combobox angezeigt | √ |
| ***5*** | User gibt Datum und Zeit ein   * 15.03.2020 * 16:45 | Zeit wird im Datetimepicker angezeigt | √ |
| ***6*** | User klickt den Button Verbindung suchen | Verbindungen zwischen den beiden Stationen ab dem 15.03 nach 16:45 Uhr werden im Datagridview angezeigt | √ |

Nächste Haltestelle anzeigen

|  |  |  |  |
| --- | --- | --- | --- |
| ***Schritt*** | ***Aktivität*** | ***Erwartetes Resultat*** | ***Status*** |
| ***1*** | User klickt auf Button unten links «Ort von Station» | Neues Form geht auf | √ |
| ***2*** | User klickt auf Button «Station in der Nähe suchen» | Stationen in der Nähe werden im Drop Down angezeigt. | Meldung kommt das man nicht auf den Standort zugreifen konnte |

Getestet von: Simon Ruckli

# Installationsanleitung

1. Als erstes können Sie über den untenstehenden Link eine Datei downloaden

<https://github.com/joegijoegi/modul-318-student/blob/master/%C3%9CK%20SBB%2B%20App.zip>

1. Wenn der Download abgeschlossen ist, können Sie über die Tastenkombination «Windows + R» ein Feld aufrufen in das Sie nun «Downloads» eintippen. Dann bestätigen Sie dies mit der Enter Taste.
2. Nun sollten Sie die Datei «ÜK SBB+ App» vor sich haben. Im nächsten Schritt müssen Sie rechtsklick auf die Zip Datei drücken und wählen Sie «alle Extrahieren»
3. Nun geben Sie einen beliebigen Pfad ein und bestätigen das Ganze mit «Extrahieren»
4. Jetzt öffnet sich ein Ordner mit 3 Dateien darin. Führen Sie nun die Applikation aus indem Sie die Datei «TransportGUI.exe» öffnen.  
   Falls eine Meldung erscheint welche besagt, dass die Applikation blockiert ist, wählen Sie «Trotzdem ausführen».

***Wichtig:***

Sie müssen die 3 Dateien immer alle im gleichen Ordner abgelegt haben, wenn Sie das Programm starten möchten.