SBB Station Finder

ÜK Projekt

Patrick Schneeberger

Inhaltsverzeichnis

[1.0 Vorwort 3](#_Toc34311489)

[1.1 3](#_Toc34311490)

[2.0 Anforderungen 4](#_Toc34311491)

[2.1 Kundenanforderungen 4](#_Toc34311492)

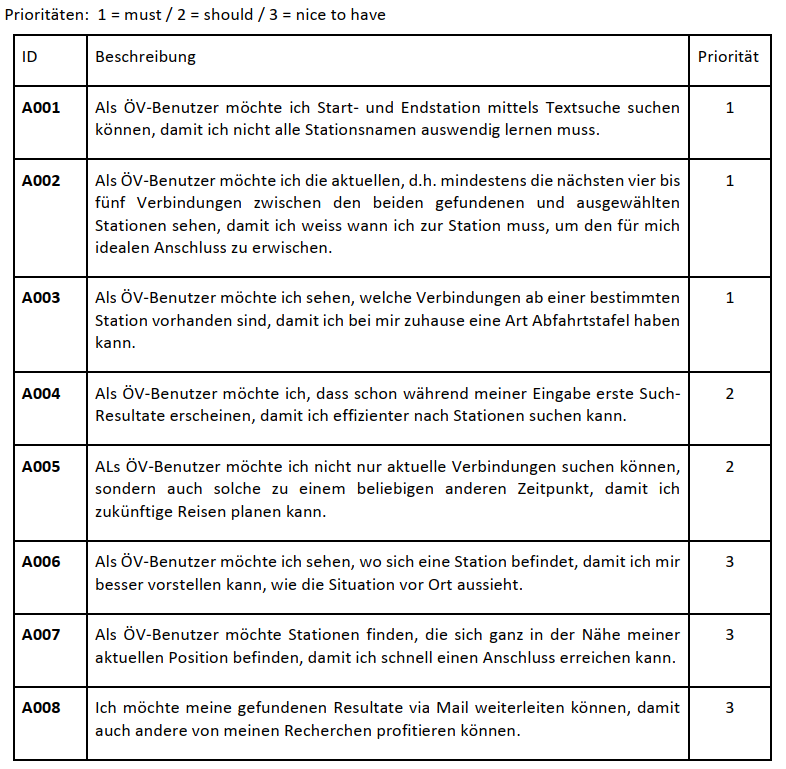
[3.0 Grafische Darstellung 5](#_Toc34311493)

[3.1 Mokup - frmMain 5](#_Toc34311494)

[3.2 Mokup - Hilfe 5](#_Toc34311495)

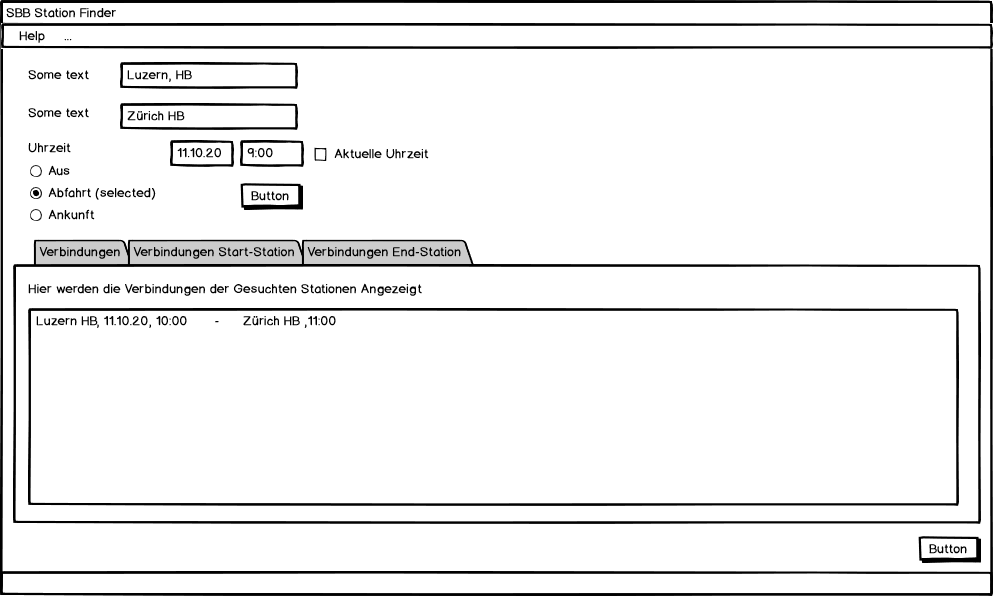
[3.3 Use Case Diagramm 6](#_Toc34311496)

1. Vorwort
2. Anforderungen
   1. Kundenanforderungen

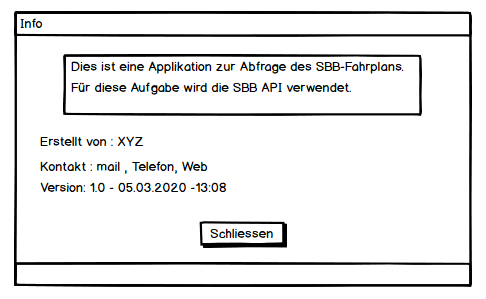


Quelle: ICT-Berufsbildung

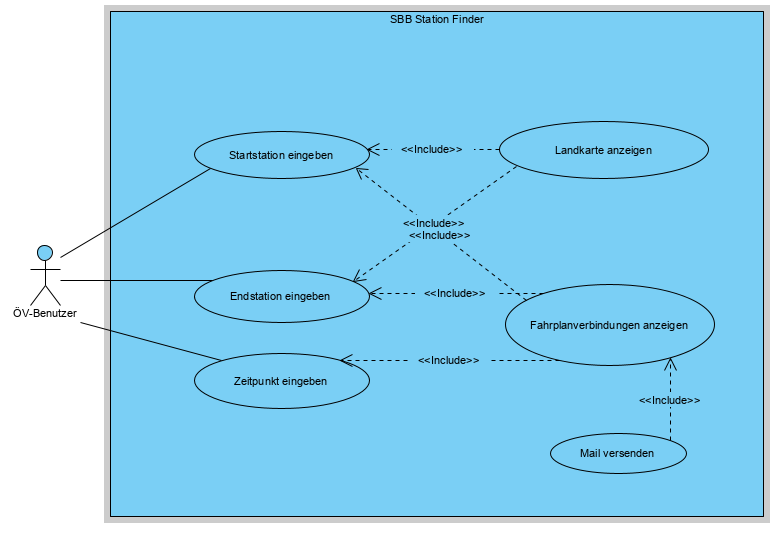
1. Grafische Darstellung
   1. Mokup - frmMain



* 1. Mokup - frmInfo



1. Use Case
   1. Diagramm



* 1. Beschreibung - Stationen eingeben

|  |  |
| --- | --- |
| **Use Case** | **Startstation, Endstation eingeben** |
| **Beschreibung** | Der Benutzer kann über die Eingabefelder Stationen definieren die bei der Suche nach Fahrplanverbindungen verwendet werden. |
| **Akteur(e)** | ÖV-Benutzer |
| **Auslöser** | Der Benutzer möchte Informationen zu Station- oder Fahrplan-Verbindungen abfragen. |
| **Vorbedingungen** | keine |
| **Ablauf** | Der Benutzer kann die Stationen wie folgt eingeben:   * Nur eine Startstation * Nur eine Endstation * Start- und End-Station   Wenn einer dieser Punkte erfüllt ist kann nach Verbindungsinformationen gesucht werden. |
| **Ergebnis** | Suche kann erfolgen |

* 1. Beschreibung - Zeitpunkt eingeben

|  |  |
| --- | --- |
| **Use Case** | **Zeitpunkt eingeben** |
| **Beschreibung** | Der Benuter kann über die Eingabefelder ein Datum und eine Uhrzeit definieren die bei der Verbindungssuche verwendet werden. |
| **Akteur(e)** | ÖV-Benutzer |
| **Auslöser** | Der Benutzer möchte Station oder Fahrplan-Verbindungen für einen apezifischen Zeitpunkt abfragen. |
| **Vorbedingungen** | Use Case : Startstation, Endstation eingeben |
| **Ablauf** | 1. Der Benutzer muss auswählen ob die Zeitangaben einer Abfahrt oder Ankunft zugeordnet werden sollen. Standardeinstellung ist die Abfahrtszeit 2. Die Zeitangaben könen Manuell angegeben werden oder es kann automatisch die aktuellen Zeitangaben verwendet werden. |
| **Ergebnis** | Suche kann erfolgen |

* 1. Beschreibung - Fahrplanverbindungen anzeigen

|  |  |
| --- | --- |
| **Use Case** | **Fahrplanverbindungen anzeigen** |
| **Beschreibung** | Der Benutzer kann über die Suche Fahrplaninformationen anfordern und anzeigen lassen. |
| **Akteur(e)** | ÖV-Benutzer |
| **Auslöser** | Der Benutzer möchte nach Stations- oder Verbindungs-Informationen suchen. |
| **Vorbedingungen** | Use Case : Startstation, Endstation eingeben  Use Case : Zeitpunkt eingeben |
| **Ablauf** | 1. Der Suche-Button wird betätigt 2. Die eingegebenen Informationen werden auf korrektheit und notwendigkeit überprüft. 3. Die Abfrage der API-Erfolgt |
| **Hinweis** | * Ist eine Abfahrtstation definiert können nur die Verbindungen der Abfahrtstation angezeigt werden. * Ist eine Ankunftstation definiert können nur die Verbindungen der Ankunftstationen angezeigt werden. * Ist eine Abfahrt- und Ankunft-Station definiert können die Verbindungen beider Stationen angezeigt werden. Dazu können die möglichen Verbindungen der einzelnen Stationen angezeigt werden. |
| **Ergebnis** | Station- oder Fahrplan-Informationen werden angezeigt |

1. Programmierrichtlinien
   1. Benennung der visuellen Elemente

Folgende Bennennungsrichtlinien gelten für visuelle Komponenten.

|  |  |
| --- | --- |
| **Komponente** | **Richtline , Beispiel** |
| Form | frmX , frmMain |
| Button | btnX , btnClose |
| TextEdit | edtX , edtVorname |
| DateTimePicker | dtpX , dtpJahrgang |
|  |  |
|  |  |

* 1. Codierrichtlinien

\*Da es manchmal schwierig ist ein geeignetes Wort für eine Variable zu finden verwende ich Wörter aus dem Deutschen oder Englischen. Dazu möchte ich umlaute im Code vermeiden:

Beispiel: Öffnen -> ~~fileÖffnen~~ -> fileOpen

|  |  |
| --- | --- |
| **Beschreibung** | **Richtline , Beispiel** |
| { } | Werden auf einer neuen Zeile geschreiben |
| Variablen | Beginnen immer mit einem Kleinbuchstaben, weitere Wörter weden mit einem Grossbuchstaben begonnen  Beispiel: personVorname |
| Variablen (Parameter) bei Funktionen oder Methoden deklaration. (Nicht beim Aufruf!) | Beginnen immer mit einem "**\_**"  Beispiel:  void Funktion2( string \_a)  {  Console.WriteLine("Ich habe {0} mitgegeben", \_a);  } |
| Variablennamen | \*Werden Englisch oder Deutsch geschreiben. |
| Umlaute | Umlaute sind nur in Kommentaren erlaubt |

1. Umgesetzte Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Priorität** | **ID** | **Beschreibung** | **Erledigt** |
| 1 | A001 | Start- und Endstation mittels Textsuche suchen | OK |
| 1 | A002 | Verbindungen zwischen Stationen anzeigen. | OK |
| 1 | A003 | Abfahrtstafeln anzeigen | OK |
| 2 | A004 | Autocomplete für Suchfelder | OK |
| 2 | A005 | Verbindungen mit Zeitpunkt suchen | OK |
| 3 | A006 | Station auf Karte anzeigen | NEIN |
| 3 | A007 | Stationen in der Nähe finden | NEIN |
| 3 | A008 | Resultate via Mail versenden | NEIN |

* 1. Bekannte Bugs
* A004 - Autocomplete für Suchfelder

Im Hintergrund wird nach jedem neuen Zeichen in der Start- oder Ziel-Suche eine abfrage für das Autocomplete gemacht. Bei vielen Resultaten ist das Control nicht sehr responsiv.

Es ist notwendig die Station im Dropdownmenü zu verwenden sonst werden keine Abfahrtstafeln angezeigt da keine ID vorhanden ist. Die ID wird von der API gefordert.

**Bei Eingabe von einem sehr langem Text entsteht manchmal eine "System.ArgumentOutOfRangeException". Diese führt zu einem ProgrammCrash.**

**Diese Exception wird von der Combobox verursacht.**

* A003 - Abfahrtstafeln anzeigeigen

Beim Suchen von "Zürich HB" funktioniert das anzeigen der Abfahrtstationen nicht korrekt. Es werden keine Abfahrtstationen zurückgegeben.

1. Kürzübersicht über eigene Methoden und Funktionen

* private void autoComplete(ComboBox \_sender)

Füllt die Liste des senders mit Daten für die Autocomplete

* private Stations getValidSations(string \_s)

Hier werden Stationen zurückgegeben die eine ID und Namen besitzen

* private void dateAndTimeGUIControl()

Hier werden DateTimeControls deaktiviert.

* private void getDateTimeConnection(string \_start, string \_end, string \_date, string \_time, short \_isArrival)

Hier werden Verbindungen abgefragt bei denen eine spezifische Zeit angegeben wird.

* private void getStationConnection(string \_start, string \_end)

Hier werden Verbindungen ohne Zeitangabe abgefragt.

* private void getStationBoard(ComboBox \_cbSender, ListBox \_lb, TabPage \_tb)

Hier werden die Stationen für die Anzeigetafel abgefragt

* private string convertToAPIDate(string \_string)

Hier wird das Datum für die API konvertiert.

1. API Erweiterung

Um Verbindungen an einem spezifischen Zeitpunkt zu suchen war es notwendig die API anzupassen.

Neu:

Transport.cs -> public Connection GetConnectionByDateTime(…)

TransprtTest.cs -> public void ConnectionsByDateTime()

Itransport.cs -> Connections GetConnectionByDateTime(…)