Dokumentation ÖV-Applikation



Inhalt

[Einleitung 3](#_Toc59111871)

[Zweck des Dokuments 3](#_Toc59111872)

[Fehler Bugs 3](#_Toc59111873)

[Mockups 3](#_Toc59111874)

[User Stories & Aktivitätsdiagramm 5](#_Toc59111875)

[Testfälle 7](#_Toc59111876)

[Testprotokoll 8](#_Toc59111877)

[Installationsanleitung 9](#_Toc59111878)

[Schlusswort 9](#_Toc59111879)

# Einleitung

Als Auftrag von den letzten 3 Tagen im ÜK, wurde uns erst eine E-Mail gegeben von einen «Kunden», welcher eine App für den ÖV-Verkehr braucht. Um dies genauer zu analysieren habe ich von einer Textdatei mit dem beschriebenen Problemen User-Stories geschrieben, bezüglich der gegebenen Anforderungen. Danach habe ich angefangen ein Mockup von Projekt zu erstellen, welche die groben Funktionen und Design der Applikation veranschaulichten. Mit dem Mockup find ich an ein Repository im GitHub zu eröffnen und an mein Projekt zu arbeiten. Dieses Dokument habe ich dann am letzten Tag im ÜK geschrieben.

Das Programm schrieb ich in C# mit WPF. Ich habe mich für WPF entschieden, weil ich die Fenster mit wenig Aufwand responsiver gestalten kann. Zudem ist man meines Erachtens nach mehr Freiheit, beim gestalten des Fensters.

# Zweck des Dokuments

Dieses Dokument wurde als Anforderung neben dem entwickeln des Programmes geschrieben. Es beinhaltet den Aufbau und Planung von Programm, die dazugehörigen Testfälle, gelöste Tätigkeiten, offene Probleme und gesammelte Erfahrungen.

# Fehler Bugs

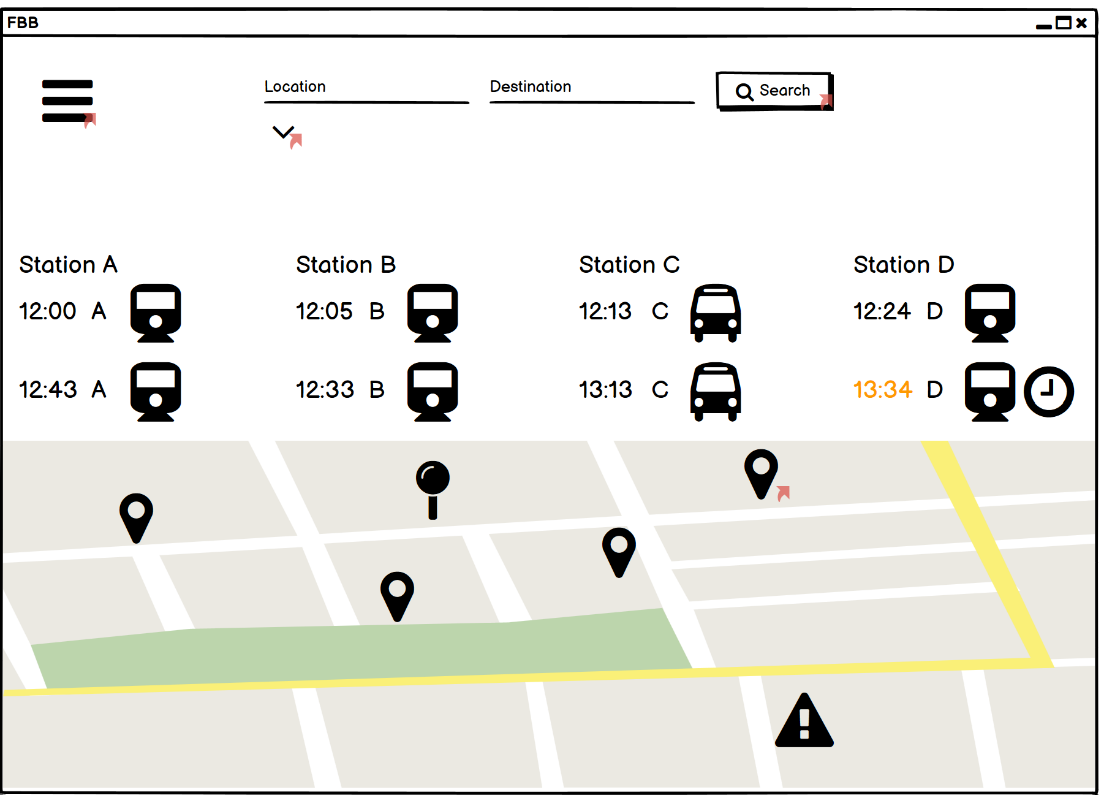
Viele Fehler die ich im Programm gefunden habe befinden sich meistens im SwissTransport-Projekt. Das Problem lag meistens in der Funktion, welches die Elemente aus einem JSON-Text holt. Es gibt zudem so gut wie keine Fehlerbehandlung was erklärt, wieso viele Fehler sich im SwissTransport auffindbar sind. Diese Probleme habe ich grundsätzlich lösen können, indem ich die Benutzereingaben so angepasst habe, dass sinnfreie Eingaben von Programm aus nicht gelesen werden.

# Mockups

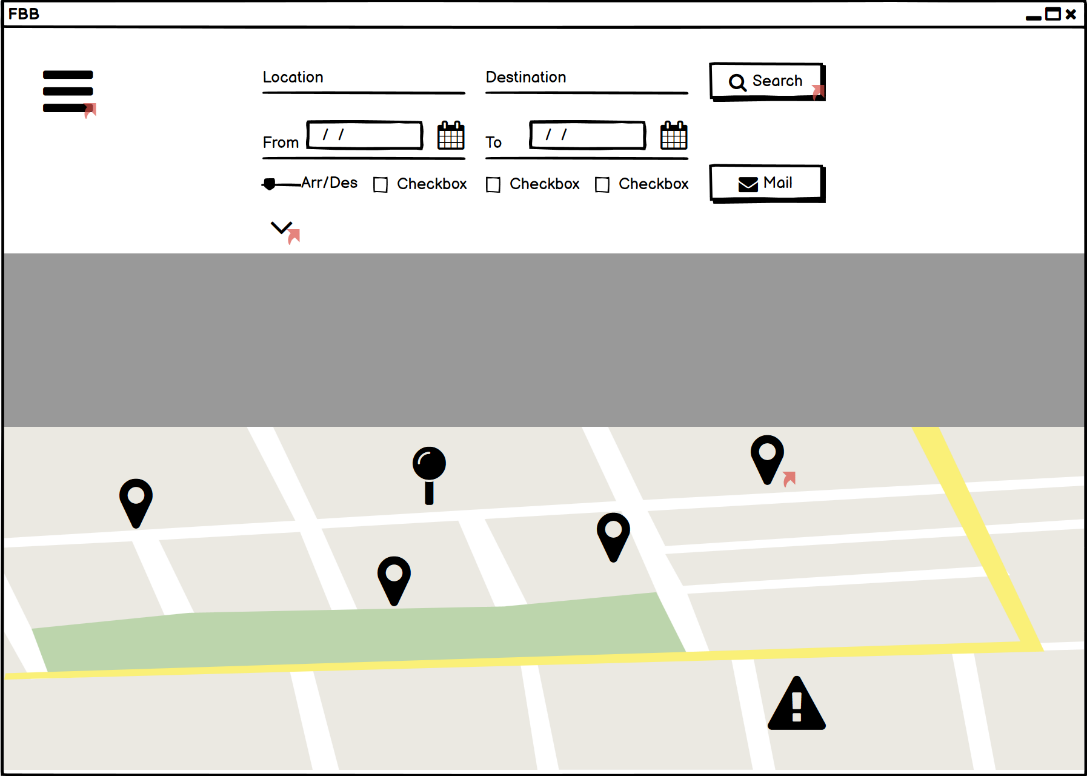
Von Design her ist geplant worden, das Suchfenster an einer Zentralen stelle zu haben. Da habe ich mich entschieden, ähnlich wie bei Google Maps, mittig oben zu positionieren.

Unterhalb der Suche sollten die Informationen und Verbindungen der gesuchten und nächstgelegenen Stationen angezeigt. Allenfalls werden diese auf der Karte markiert.

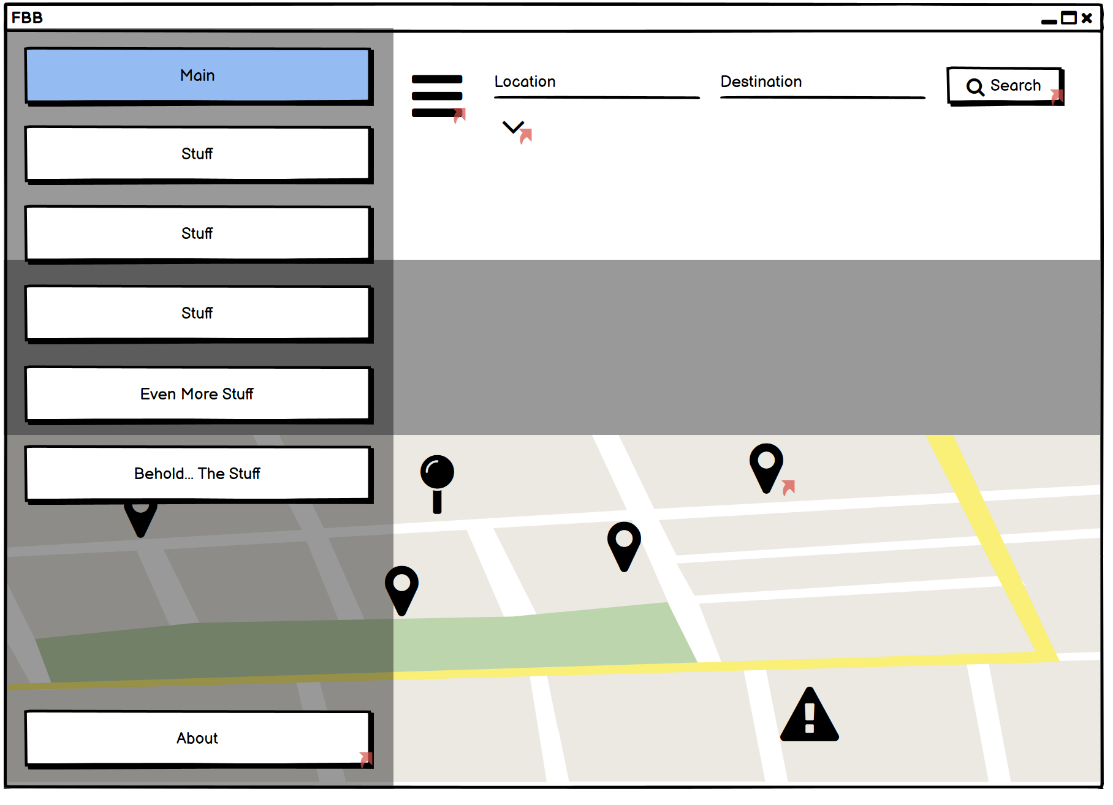
Weil ich nicht gerade alles auf einer Programmseite genug Platz hat, habe ich fürs Mockup ein Menü erstellt, welches weitere Seiten öffnet.



Klappt man die Suche auf, werden weitere Möglichkeiten fürs Suchen der Verbindungen dargestellt. Man kann nach diversen Kriterien, wie die Ankunftszeit suchen.



Das Menü ist schlicht gehalten und im Mockup mit Platzhalterseiten verlegt. Grundsätzlich sind mindestens zwei Seiten geplant worden für die Stationssuche und Verbindungssuche.



# User Stories & Aktivitätsdiagramm

Die Testfälle beschreiben Fälle aus der Sicht des Konsumenten, welches Lösungen auf diverse Probleme beschreibt. Zudem wurde ein Aktivitätsdiagramm geschrieben. Dies beschreibt den grundsätzlichen Aufbau der Suchfunktion von der Eingabe.

*Rot = Erste Priorität*

*Gelb = Zweite Priorität*

*Grün = Dritte Priorität*

1. **Eingabe Start- und Endstation**  
    Ich als **Aussendienstmitarbeiter** möchte **Start**- und **Endstation** eingeben können um Verbindungen heraus zu suchen.

* Start und Endstation kann eingeben werden
* Verbindungen werden gesucht

1. **Verbindungsvorschläge**  
   Ich als Aussendienstmitarbeiter möchte die **nächsten 4 Verbindungen** angezeigt bekommen, damit ich potenzielle Verbindungen nicht verpasse.  
   - Nächsten 4 Verbindungen werden angezeigt
2. **Abfahrtstafel**  
   Ich als **Aussendienstmitarbeiter** möchte eine **Abfahrtstafel von der gesuchten Station** sehen um die nächsten Abfahrten von einer Station zu sehen.   
   - Abfahrtstafel von der gesuchten Station wird angezeigt
3. **Datum- und Uhrzeiteingabe**  
   ich als **Aussendienstmitarbeiter** möchte eine **Datum- und Uhrzeiteingabe**, um im Voraus die Verbindungen zu betrachten.  
   - Datum und Uhrzeit kann man eingeben

* Verbindungen werden gesucht

1. **Suchvorschläge**  
   Ich als **Aussendienstmitarbeiter** möchte während der Eingabe einer Station, **Vorschläge** erhalten, damit die Eingabe einfach und schnell geht.  
   - Vorschläge werden angezeigt
2. **Karte (nicht umgesetzt)**  
   Ich als **Aussendienstmitarbeiter** möchte **eine Karte** sehen, auf welcher alle Stationen der Schweiz zu sehen sind, **damit** ich die Station finden kann.

* Karte wird angezeigt
* Stationen werden angezeigt und sind auswählbar

1. **Nahegelegene Stationen (nicht umgesetzt)**  
   Ich als **Aussendienstmitarbeiter** möchte alle **Stationen in der Nähe** angezeigt haben, **damit** ich keine Umwege machen muss.

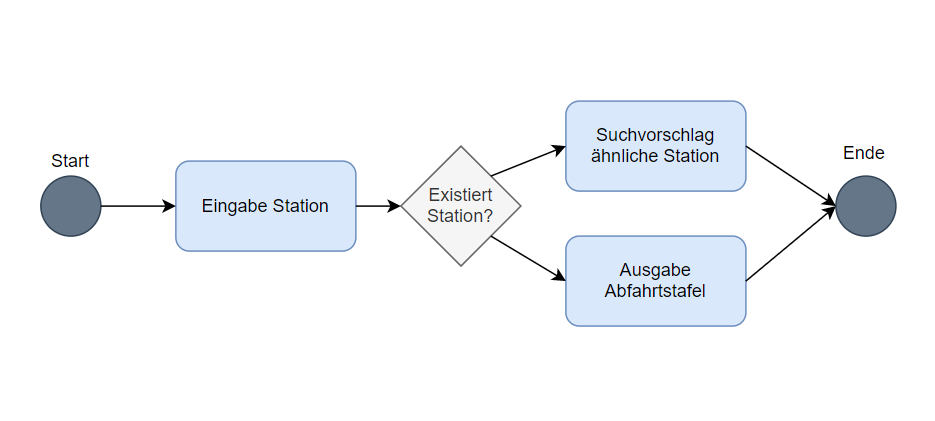
* Stationen in der Nähe werden angezeigt

1. **Mailversand (nicht umgesetzt)**  
   Ich als **Aussendienstmitarbeiter** möchte eine **Verbindung per Mail** versenden können, **damit** ich diese einem anderen Mitarbeiter oder Kunden teilen kann.

* Mail wird verschickt
* Verbindung wird angezeigt

1. **Verkehrszustand (nicht umgesetzt)**  
   Ich als **Aussendienstmitarbeiter** möchte den momentanen **Stand von Verkehr** sehen, damit ich nicht im Stau stecke.

* Zustand von Verkehr wird angezeigt



# Testfälle

|  |  |  |
| --- | --- | --- |
| **Testnummer** | **Funktion** | **Erwartetes Resultat** |
| 1 | Suchen von Verbindungen mit der Eingabe von zwei Stationen | Verbindungen werden im DataGrid angezeigt, keine Anzeige-/Binding-Fehler |
| 2 | Suchvorschläge beim Eingeben von Stationen werden angezeigt | Suchvorschläge in der Combobox werden angezeigt, Suchvorschläge auswählbar |
| 3 | Sortierung im DataGrid, ScrollBar und GridSplitter funktionieren einwandfrei | Keine Anzeigefehler, Elemente im DataGrid sortierbar (wenn man auf dem Header drückt), DataGrid darf nicht anpassbar sein (neue Columns erstellen, Alte löschen und bearbeiten) |
| 4 | Stationsinformationen werden beim Auswählen einer Verbindung angezeigt | Zielstation (Informationen) wird im DataGrid angezeigt, keine Anzeige-/Binding-Fehler |
| 5 | Suchen von Verbindungen mit der Uhrzeiteingabe | Alle angezeigten Verbindungen müssen sich in der ausgewählten Zeitspanne befinden |
| 6 | Stationen im «StationBoard» suchbar | Suchvorschläge werden angezeigt, Stationen können gesucht werden, keine Anzeigefehler |
| 7 | Abfahrtstafel in der Station nachschaubar | Abfahrtstafeln anschaubar, keine Anzeigefehler |
| 8 | Sinnvolle Tabstopps | Tabstopps von der Reihenfolge der Elemente nach, Benutzerfreundlichkeit |

# Testprotokoll

***Hinweis****: Datum und Tester müssen nicht spezifiziert sein, weil ich als Einzelner das Programm geschrieben und somit auch getestet habe. Die Tests wurden am Freitag den 17.12.2020 durchgeführt.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testnummer** | **Funktion** | **Erwartetes Resultat** | **Anmerkungen** | **I.O** |
| 1 | Suchen von Verbindungen mit der Eingabe von zwei Stationen | Verbindungen werden im DataGrid angezeigt, keine Anzeige-/Binding-Fehler | Dauer der reise beinhaltet die Anzahl Tage. | √ |
| 2 | Suchvorschläge beim Eingeben von Stationen werden angezeigt | Suchvorschläge in der Combobox werden angezeigt, Suchvorschläge auswählbar |  | √ |
| 3 | Sortierung im DataGrid, ScrollBar und GridSplitter funktionieren einwandfrei | Keine Anzeigefehler, Elemente im DataGrid sortierbar (wenn man auf dem Header drückt), DataGrid darf nicht anpassbar sein (neue Columns erstellen, Alte löschen und bearbeiten) | Sortierung und ScrollBar bereits im UserControl vorhanden. | √ |
| 4 | Stationsinformationen werden beim Auswählen einer Verbindung angezeigt | Zielstation (Informationen) wird im DataGrid angezeigt, keine Anzeige-/Binding-Fehler |  | √ |
| 5 | Suchen von Verbindungen mit der Uhrzeiteingabe | Alle angezeigten Verbindungen müssen sich in der ausgewählten Zeitspanne befinden |  | √ |
| 6 | Stationen im «StationBoard» suchbar | Suchvorschläge werden angezeigt, Stationen können gesucht werden, keine Anzeigefehler |  | √ |
| 7 | Abfahrtstafel in der Station nachschaubar | Abfahrtstafeln anschaubar, keine Anzeigefehler |  | √ |
| 8 | Sinnvolle Tabstopps | Tabstopps von der Reihenfolge der Elemente nach, Benutzerfreundlichkeit | Das DataGrid selbst und die Gridsplitters werden von Tabstopp absichtlich ignoriert. | √ |

# Installationsanleitung

Installiert kann die Software mithilfe des Installers. Mit dem Ausführen von FFB-Setup im doc-Ordner (auf GitHub) unter Setup wird die Software heruntergeladen.

# Schlusswort

Trotz der unüblichen Wahl das Programm mit WPF zu gestalten, behaupte ich mal in den vergangen Paar Tagen ein Endprodukt erstellt zu haben, welches sich sehen lassen könnte. Ich bin Zeitlich ganz gut durchgekommen, dabei habe ich die weniger wichtigen Funktionen ausgelassen oder anders als erwartet im Programm verbaut. In den vergangen Paar Tagen konnte ich vieles neues Lernen. Dabei habe ich neue Erfahrungen mit der Programmiersprache C# gesammelt.