Swiss Transport



Von Xavier Scherer

Inhaltsverzeichnis

[Einleitung 3](#_Toc102134632)

[Zweck des Dokuments 3](#_Toc102134633)

[Funktionen 3](#_Toc102134634)

# Einleitung

Wir haben die Aufgabe bekommen, eine Fahrplanapp für die Unternehmung von Herr Kunde zu Programmieren. Herr Kunde hat uns über LinkedIn kontaktiert und hat in der Nachricht grob erklärt was sein Problem ist. Danach hat an sich getroffen und hat die Bedürfnisse genauer angeschaut und die Anforderungen abgeklärt. Dann hat man sich 9 User Stories überlegt welche aus den besprochenen Punkten abgeleitet wurden. Diese wurden auch mit der dazugehörigen Priorität versehen, damit man beim Programmieren weiß auf was man sich fokussieren muss. Dann blieb nicht mehr viel zu machen und wir fingen an zu programmieren.

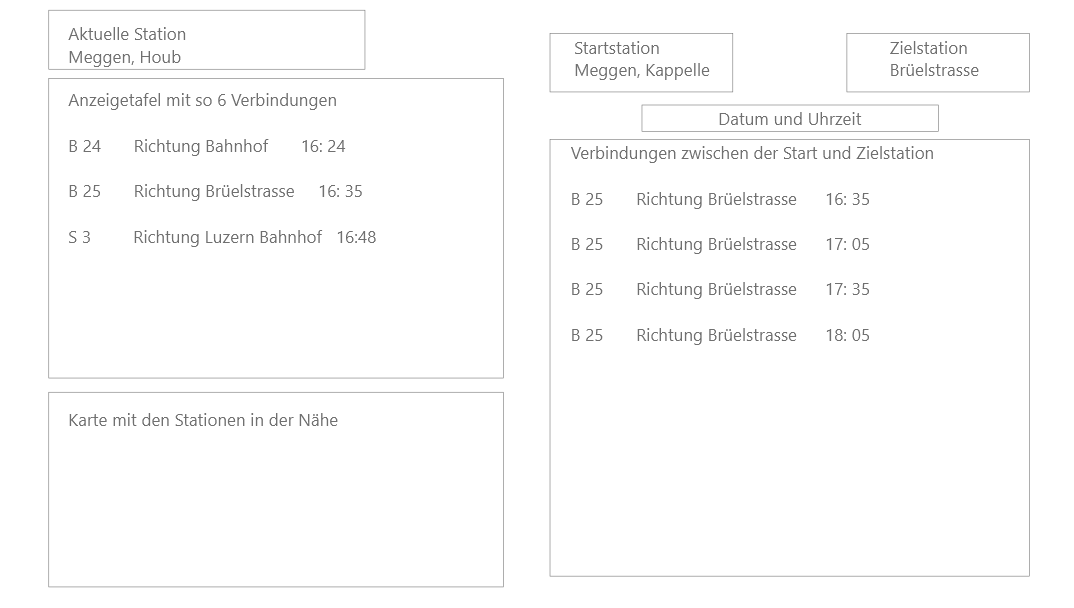
# Zweck des Dokuments

Dieses Dokument dient dazu dem Leser Einblick in die Entwicklung eines Programmes zu geben. Vom Anfang, wo man sich die User Stories überlegt, und mit dem Kunden abspricht bis dann das Produkt den Usern zur Verfügung gestellt wird. Mithilfe dieses Dokuments kann man auch die Gedankengänge des Programmierers besser Nachvollziehen und man sieht, wieso man ein Problem auf diese Art gelöst hat. Zudem hilft es auch zu verstehen, dass die Entwicklung einer App nicht einfach nur aus Programm-Code schreiben besteht, sondern aus weitaus mehr, weshalb manchmal auch nicht alles umgesetzt werden kann.

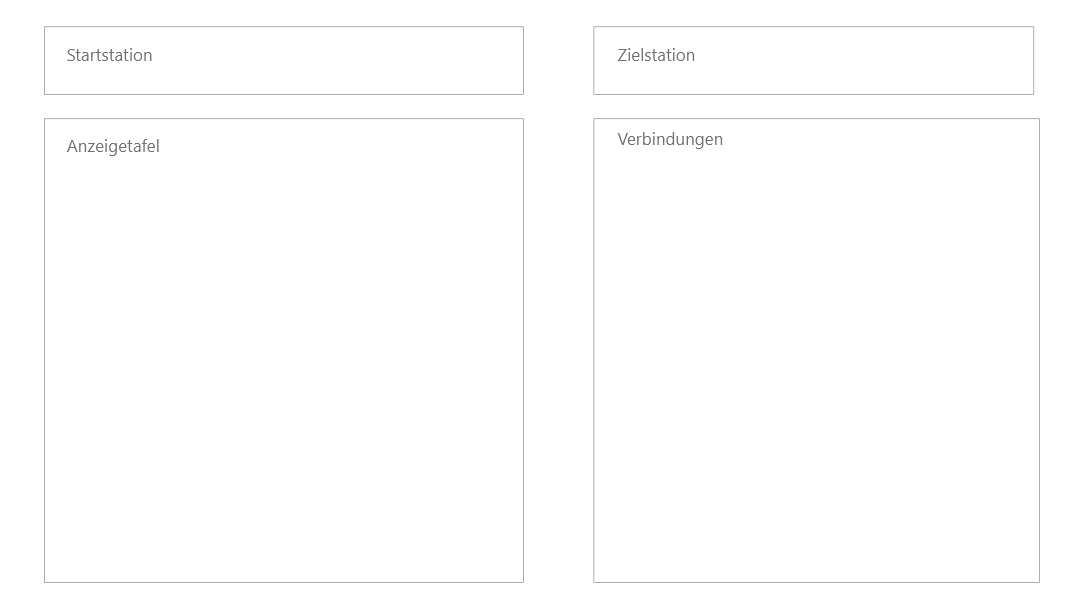
# Fehlende Funktionen

Ich habe mir für diese Applikation vorgenommen möglichst nahe an die SBB App heranzukommen, da dies meiner Meinung nach die beste ÖV-App ist. Dazu gehört eigentlich auch das die am naheliegendste Haltestelle angezeigt wird und auch die Anzeigetafel dieser Haltestelle. Das mit der Anzeigetafel habe ich hinbekommen, aber ich habe es nicht geschafft die Haltestelle herauszufinden. Dies war aufgrund von zu wenig Zeit und weil bestimmte Funktionen nicht funktioniert haben, welche das ganze um einiges einfacher gemacht hätten.

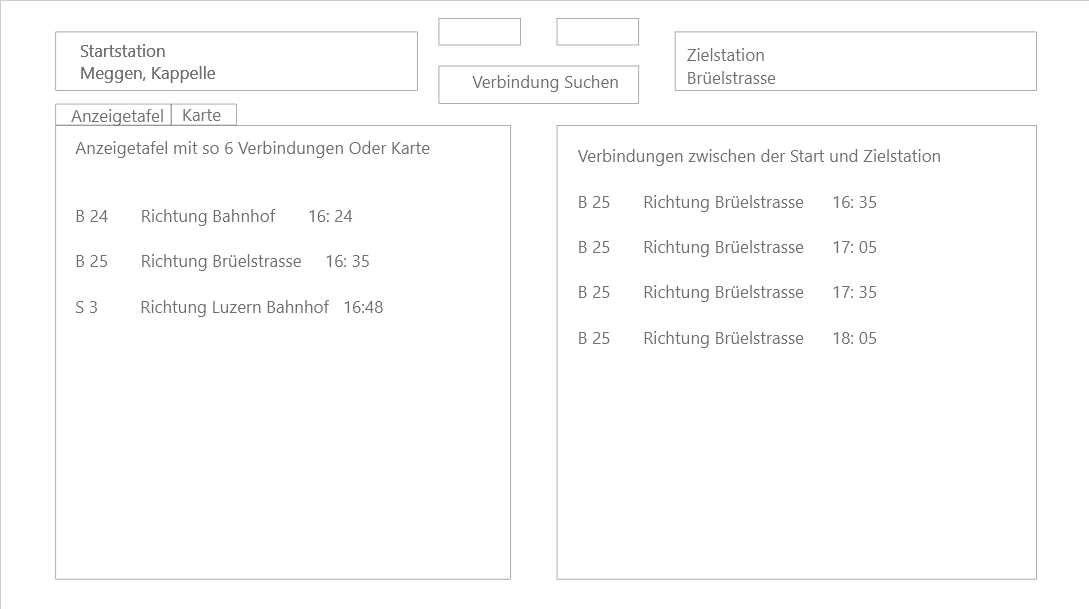
# Mockups

Ich habe zuerst dieses erste Mockup erstellt, wo ich einfach alle Funktionen welche ich gern hätte hinein getan habe ohne mir zu viele Gedanken zu machen ob all dies möglich wäre oder nicht. 

Bevor ich anfing zu programmieren habe ich dann noch dieses 2. Mockup erstellt, wo alles so ist, das es sie User Stories der Priorität 1 erfüllt also das absolute Minimum.



Während dem Programmieren habe ich dann realisiert was alles möglich ist und was eher nicht und ich habe dann schlussendlich noch dieses Mockup erstellt, welches auch sehr ähnlich wie mein Endprodukt aussieht.



# User Stories

1. Start Ende Suche: Als Aussendienstarbeiter möchte ich Verbindungen zwischen Start und Endstation suchen, um Zeit zu sparen.

Abnahmekriterien: Startstation muss eingegeben werden / Endstation muss eingegeben werden. Suchen Button muss gedrückt werden.

Priorität: 1 / Status: Done

1. Vier Verbindungen: Als Aussendienstarbeiter möchte ich, dass die nächsten vier Verbindungen angezeigt werden, damit ich mich gut zurechtfinde.

Abnahmekriterien: Start und Endstation müssen eingegeben werden.

Priorität: 1 / Status: Done

1. Anzeigetafel: Als Mitarbeiter möchte ich, dass meine Verbindungen auf einer Anzeigetafel angezeigt werden, damit alles übersichtlich ist.

Abnahmekriterien: Gerät mit Display.

Priorität: 1 / Status: Done

1. Station suche: Als Mitarbeiter möchte ich, nicht den vollen Namen der Ortschaft eingeben und trotzdem eine Station angezeigt bekommen, da ich nicht alles auswendig kann.

Abnahmekriterien: Die ersten Buchstaben des Orts eingeben und Enter drücken, damit Vorschläge erscheinen.

Priorität: 2 / Status: Done

1. Autocomplete: Als Mitarbeiter möchte ich, dass die Station angezeigt wird, auch wenn ich den Namen nicht komplett weiss, damit ich nicht lange suchen muss.

Abnahmekriterien: Erste Buchstaben der Station muss eingegeben werden und Enter drücken.

Priorität: 2 / Status: Done

1. Zukünftige Abfahrten: Als Mitarbeiter möchte ich ein Datum und eine Uhrzeit eingeben könne damit ich auch für Zukünftige Verbindungen nachschauen kann.

Abnahmekriterien: Start und Endstation + Datum und Uhrzeit in der Zukunft aussuchen + Suchen Button drücken.

Priorität: 2 / Status: Done

1. Weiterleitung: Als Mitarbeiter möchte ich, dass ich eine ausgewählte Verbindung an meinen Mitarbeiter weiterleiten kann, damit er nicht lange suchen muss.

Abnahmekriterien: Aussuchen einer Verbindung und weiterleiten.

Priorität: 3 / Status: ToDo

1. Aktueller Standort: Als Mitarbeiter möchte ich, von meinem aktuellen Standort, die nächsten Verbindungen anzeigen lassen, damit ich schnell eine Verbindung finde.

Abnahmekriterien: GPS-Fähiges Gerät

Priorität: 3 / Status: ToDo

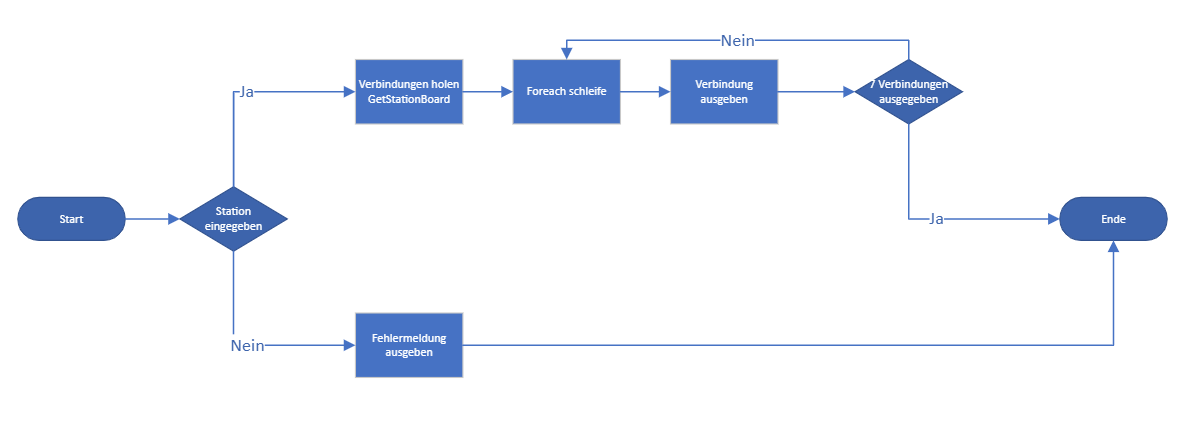
1. Karte: Als Mitarbeiter möchte ich, eine Karte anzeigen lassen, damit ich genau sehe, wo ich mich befinde.

Abnahmekriterien: GPS-Fähiges Gerät.

Priorität: 3 / Status: Done

# Aktivitätsdiagram

Hier ist mein Aktivitätsdiagramm zur User Story Anzeigetafel



# Testfälle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Erhaltenes Resultat | Erfüllt |
| 1 | Start und Endstation eingeben und Button drücken | Nächste 6 Verbindungen werden angezeigt | Nächste 6 Verbindungen werden angezeigt | Ja |
| 2 | Startstation eingeben und auf Anzeigetafel drücken | Nächste 7 Abfahrten werden angezeigt | Nächste 7 Abfahrten werden angezeigt | Ja |
| 3 | Auf Karte drücken | Google Maps wird in anstelle der Anzeigetafel geöffnet | Google Maps wird in anstelle der Anzeigetafel geöffnet | Ja |
| 4 | Zeit und Datum per DateTimePicker verändern und Verbindungen suchen | Verbindungen zum ausgewählten Datum/Zeitpunkt anzeigen | Verbindungen zum ausgewählten Datum/Zeitpunkt anzeigen | Ja |
| 5 | Die Hälfte eines Ortsnamens eingeben und Enter drücken | Es werden in der Combobox Vorschläge für Ortsnamen angezeigt | Es werden in der Combobox Vorschläge für Ortsnamen angezeigt | Ja |

# Installationsanleitung

Um die Anwendung auszuführen, muss man das Git Repository Klonen und danach im Visual Studio öffnen. Nun wählt man SwissTransportGUI aus und debuggt das.