

|  |
| --- |
| Projekt SBB ÜK Modul 318 |
|  |
| 20. Dezember  Autor  Nikolic Nikola  17.12.2019 |

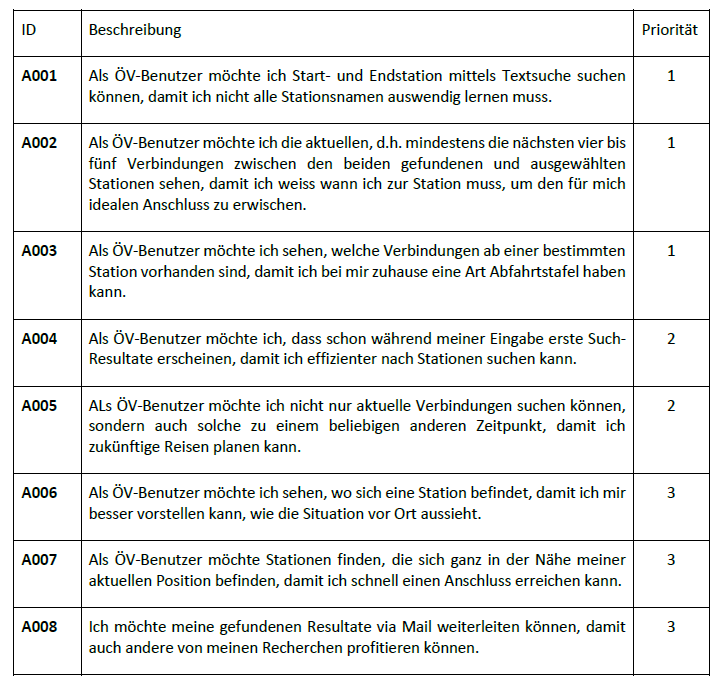
[](https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn.handelszeitung.ch%2Fsites%2Fdefault%2Ffiles%2Fstyles%2F16x9_280%2Fpublic%2Fjapan_transport_bullet_train.jpg&imgrefurl=https%3A%2F%2Fwww.handelszeitung.ch%2Funternehmen%2Fjapan-testet-den-schnellsten-zug-der-welt&docid=11VX9DJKiP87FM&tbnid=5729cmetGWQlBM%3A&vet=10ahUKEwjWnZaD77zmAhVT6qYKHR1iD2sQMwiEASgpMCk..i&w=420&h=236&bih=751&biw=1536&q=zug&ved=0ahUKEwjWnZaD77zmAhVT6qYKHR1iD2sQMwiEASgpMCk&iact=mrc&uact=8)

# Index

* Anforderungen
* Davon Ausgefüllte Anforderungen
* Use Case Diagramm
* Use Cases Beschreibung
* Aktivitätsdiagramm
* Einleitung
* Zweck
* Programmierrichtlinien
* Mockups
* Testfälle
* Testprotokoll
* Installationsanleitung
* EVT Bilder

Anforderungen

1 = Muss 2 = Soll 3 = nice to have



Davon Ausgefüllte Anforderungen

Die 3 wichtigsten Prioritäten konnte ich ins Projekt einbinden (die A001 -003) die weiteren Anforderungen habe ich versucht möglichst gut hinzubringen.

Ich konnte auch eigene Ideen einbringen.

Die Richtlinien habe ich so gut wie möglich beachtet.

Anforderung A001

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «Lu» ein. | Es sollen Stationen welche «Lu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| 2 | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Zu» ein. | Es sollen Stationen welche «Zu» enthalten vorgeschlagen werden.  Es muss so aussehen: |

Anforderung A002

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «Lu» ein. | Es sollen Stationen welche «Lu» enthalten vorgeschlagen werden.  Es muss so aussehen: |

|  |  |  |
| --- | --- | --- |
| 2 | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Zu» ein. | Es sollen Stationen welche «Zu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| 3 | Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | In der List Box über der «Suchen»-Taste werden die nächsten vier Verbindungen angezeigt (in diesem Beispiel die Verbindungen zwischen Luzern und Zürich) |

Anforderung A004

|  |  |  |
| --- | --- | --- |
| 1 | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «Lu» ein. | Es sollen Stationen welche «Lu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| 2 | Der Benutzer kann direkt auf «Suchen» klicken, ohne einen Ankunftsort eingegeben zu haben. | Es muss ein neues Fenster aufgehen, wo sich alle Verbindungen ab der Abfahrtstation anzeigen lassen.  Hiermit Beispiel als Abfahrtort Luzern: |

Anforderung A004

|  |  |  |
| --- | --- | --- |
| 1 | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «L» ein. Die Suche beginnt ab dem ersten Char. | Auto Fill wird direkt gestartet. Es werden alle möglichen Vorschläge, welche zur Suche passen angezeigt.  Es muss so aussehen: |
| 2 | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Z» ein. Die Suche beginnt ab dem ersten Char. | Es sollen Stationen welche «Z» enthalten vorgeschlagen werden.  Es muss so aussehen: |

|  |  |  |
| --- | --- | --- |
| 3 | Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | In der List Box über der «Suchen»-Taste werden die nächsten vier Verbindungen angezeigt (in diesem Beispiel die Verbindungen zwischen Luzern und Zürich) |

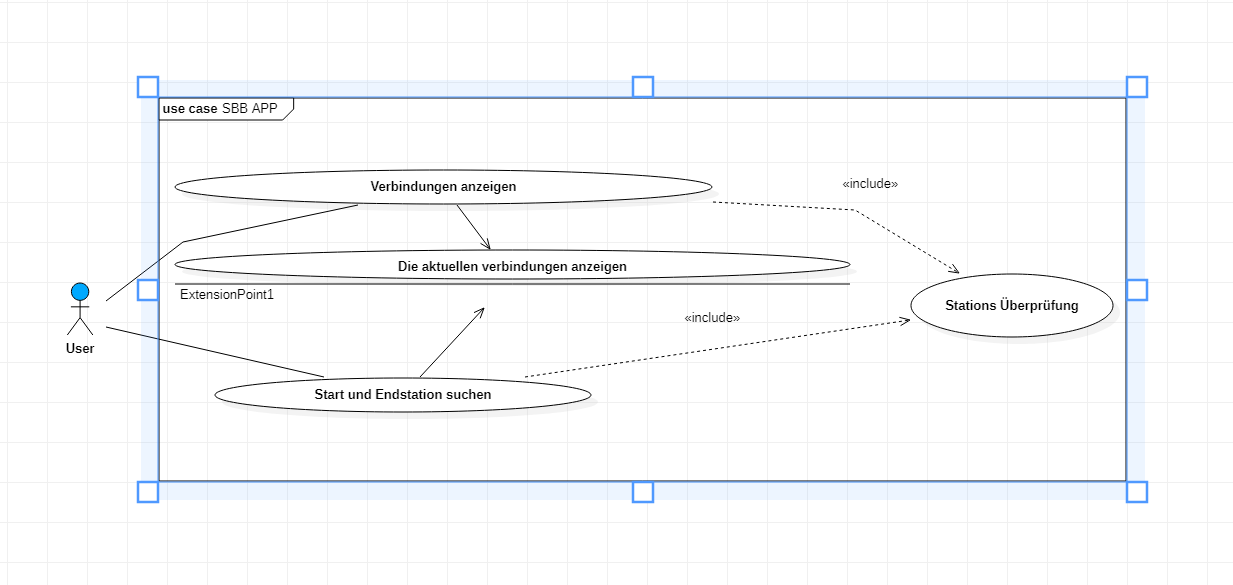
Anforderung A005

|  |  |  |
| --- | --- | --- |
| 1 | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «L» ein. Die Suche beginnt ab dem ersten Char. | Autofill wird direkt gestartet. Es werden alle möglichen Vorschläge, welche zur Suche passen angezeigt.  Es muss so aussehen: |
| 2 | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Z» ein. Die Suche beginnt ab dem ersten Char. | Es sollen Stationen welche «Z» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| 3 | Der Benutzer kann auch die Abfahrtszeit Beliebig ändern. Er kann also die ersten zwei stellen mit der Stunde befüllen und die nächsten zwei mit den Minuten befüllen.  Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | Es sollen nur Verbindungen angezeigt werden, welche innerhalb der Zeiteingabe des Benutzers liegen.  (Hier ein Beispiel falls der Benutzer als Abfahrtszeit «09.35» eingegeben hat) |

Anforderung A008

|  |  |  |
| --- | --- | --- |
| 1 | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «L» ein. Die Suche beginnt ab dem ersten Char. | Autofill wird direkt gestartet. Es werden alle möglichen Vorschläge, welche zur Suche passen angezeigt.  Es muss so aussehen: |
| 2 | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Z» ein. Die Suche beginnt ab dem ersten Char. | Es sollen Stationen welche «Z» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| 3 | Der Benutzer kann auch die Abfahrtszeit Beliebig ändern. Er kann also die ersten zwei stellen mit der Stunde befüllen und die nächsten zwei mit den Minuten befüllen.  Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | Es sollen nur Verbindungen angezeigt werden, welche innerhalb der Zeiteingabe des Benutzers liegen.  (Hier ein Beispiel falls der Benutzer als Abfahrtszeit «09.35» eingegeben hat) |
| 4 | Danach kann der Benutzer seine gefundene Resultate via Mail teilen. | Diese Funktion ist mir zuhause bereits einmal gelungen. Ich habe die Zieladresse und den Absender HardCoden müssen. |

Use Case Diagramm



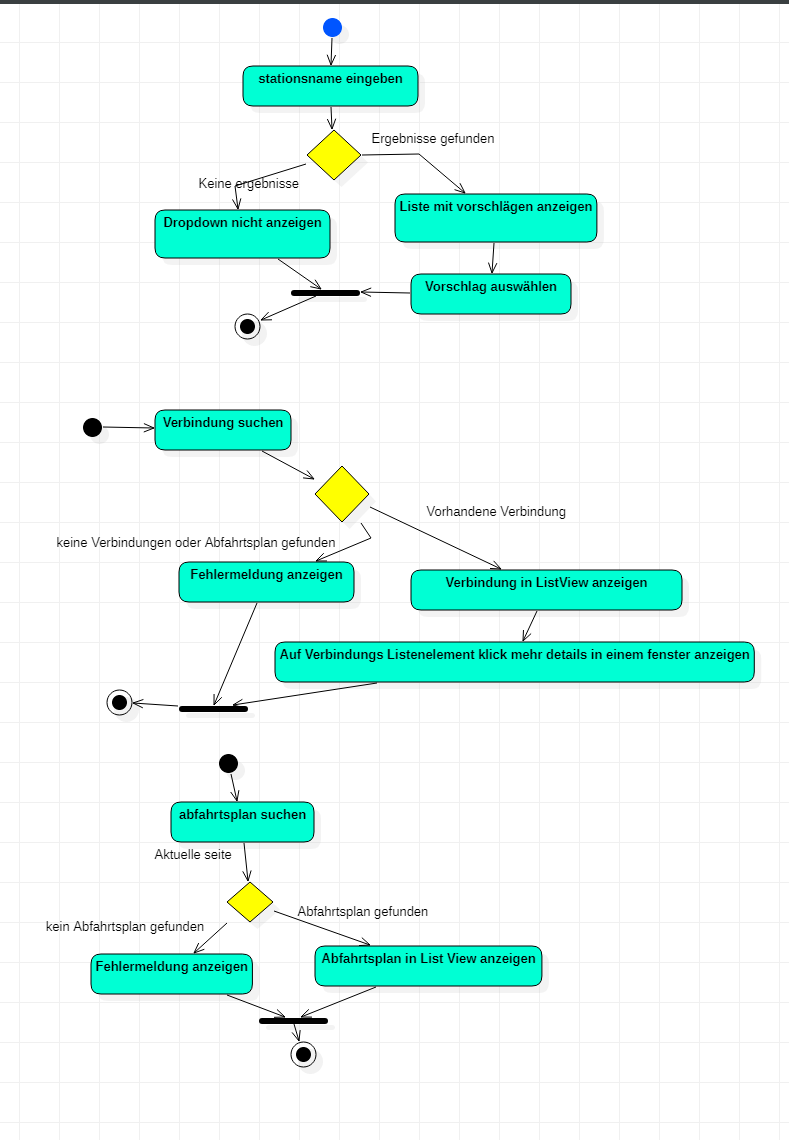
Use Case Beschreibung

|  |  |
| --- | --- |
| Use Case | 1. Verlängerung erfassen |

|  |  |
| --- | --- |
| Beschreibung | Ein Kunde / Mitarbeiter der SBB «Die Schweizerische Bundesbahnen AG» möchte wissen wo es in der Nähe Bus / Zugverbindungen gibt und welche Verbindungen wohin führen |
| Akteur | Kunde der SBB sowie jede öffentliche Person und Mitarbeiter der SBB |
| Auslöser | Interesse am abfahrtsplan und Verbindungen zwischen strecken und ab einer Station. |
| Vorbedingungen | Der Benutzer muss lesen und schreiben können |
| Ablauf | 1. Programm installieren 2. Programm öffnen 3. Station suchen 4. Verbindungen ansehen 5. Hingehen 6. In den Zug / Bus steigen und losfahren |
| Alternativer Ablauf | Kein |
| Ergebnis | Fahrkarte wurde gekauft und man kann sich in den Bus oder Zug setzen und losfahren |

Aktivitätsdiagramm

Mein Aktivität Diagramm ist recht einfach, da das was ich jetzt mache meiner Meinung nach genug ist. Sollte mir auffallen das mir was fehlt sollte ich am schliss noch genug zeit haben, so werde ich bestimmt noch zusätzliche Inhalte in das Programm einfügen können.



Einleitung

Über dieses Projekt

In diesem Projekt geht es um ein Programm das den Benutzern ermöglicht eine Verbindung zwischen Punkt A und punkt B, ab einer bestimmten Station oder in der Nähe des Benutzers eine Station und deren aus- und eingehenden Verbindungen zu suchen.

Eigenschaften

Es können zwei Stationen oder Orte (Startort und Ankunftsort) angegeben werden. Standardmäßig werden das aktuelle Datum und Zeit eingefüllt, jedoch kann das Abfahrtsdatum und die Abfahrtszeit beliebig angegeben werden. Wird nach Verbindungen gesucht werden von der API die nächsten Verbindungen anhand der Eingaben zurückgegeben, welche in die Liste gefüllt werden. Es werden pro Verbindung die Startort Station und Abfahrtszeit sowie der Ankunftsort und Ankunftszeit angezeigt. Wird ein Listenelement angewählt, erscheint ein neues Fenster mit mehr Details (Der Start Station die genaue Abfahrtszeit sowie das Gleis, die Reisezeit und der Ankunftsort mit genauer Ankunftszeit und Angaben des Ankunftsgleises). Bei Falschen Angaben erscheint eine Nachricht, welche die fehlerhaften Angaben zurückmeldet.

Zweck

Dieses Dokument soll dazu dienen, dass man den Verlauf und die Entwicklung der Applikation nachvollziehen kann. Man sieht wie die Applikation strukturell aufgebaut ist. Der Hauptzweck dient dazu, Punkte für die Bewertung zu sammeln, da eine Dokumentation eine Anforderung ist.

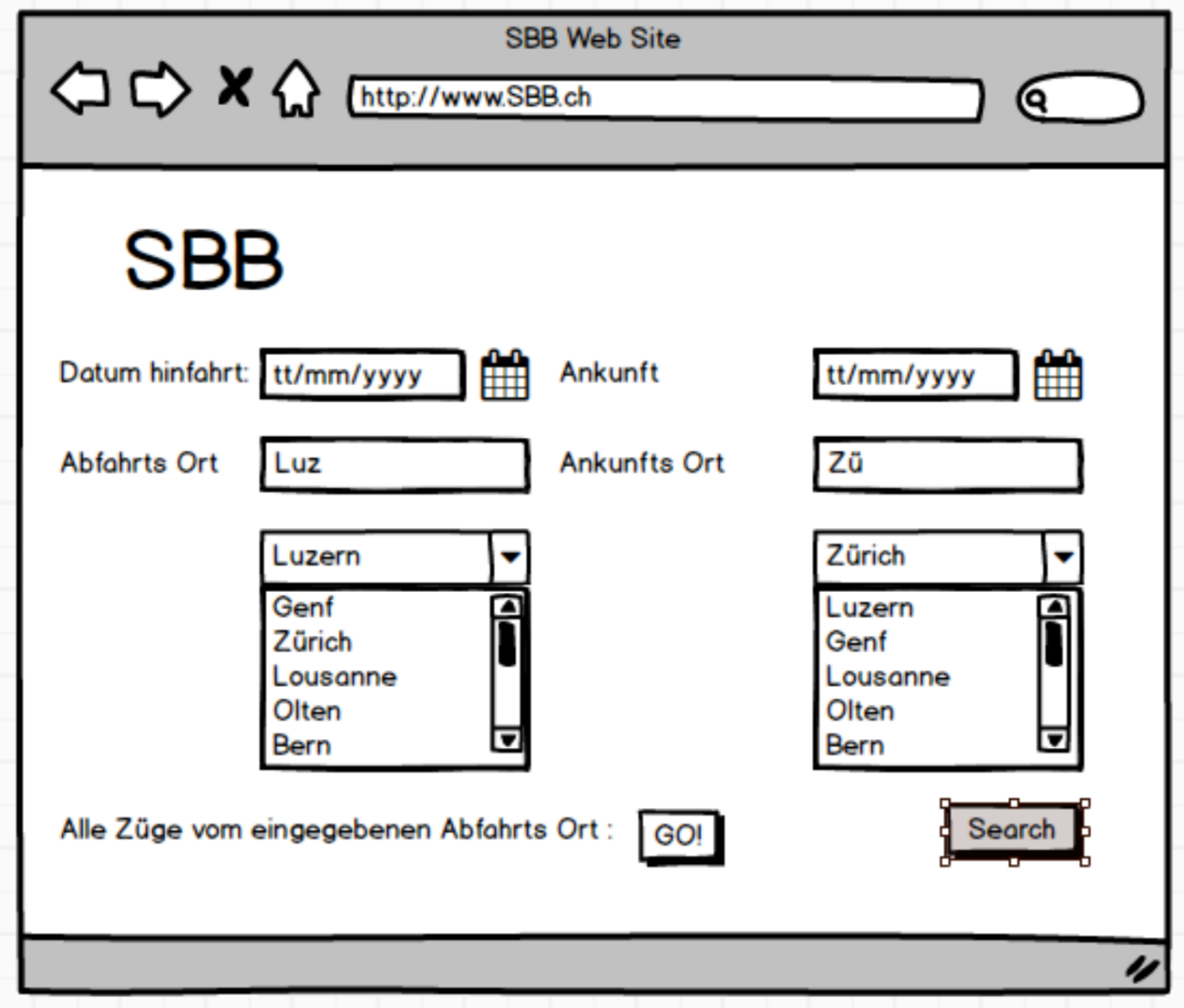
Programmierrichtlinien

Beim Programmieren versuche ich meinen Code so zu schreiben, dass er ohne Kommentare klar und verständlich ist. Die Namen der Objekte und Variablen sind alle einheitlich im Camel-Case Stil geschrieben. Camel Case ist ein Code, der mehrmals gebraucht wird, deshalb habe ich ihn jeweils in Methoden gepackt, um Redundanzen zu verhindern.

Mockups

Bevor ich mit dem Programmieren beginne, muss ich noch ein paar Diagramme erstellen, damit ich sehe was ich alles machen kann. Als erstes erstellte ich ein GUI-Design an welchem ich mich orientieren werde.

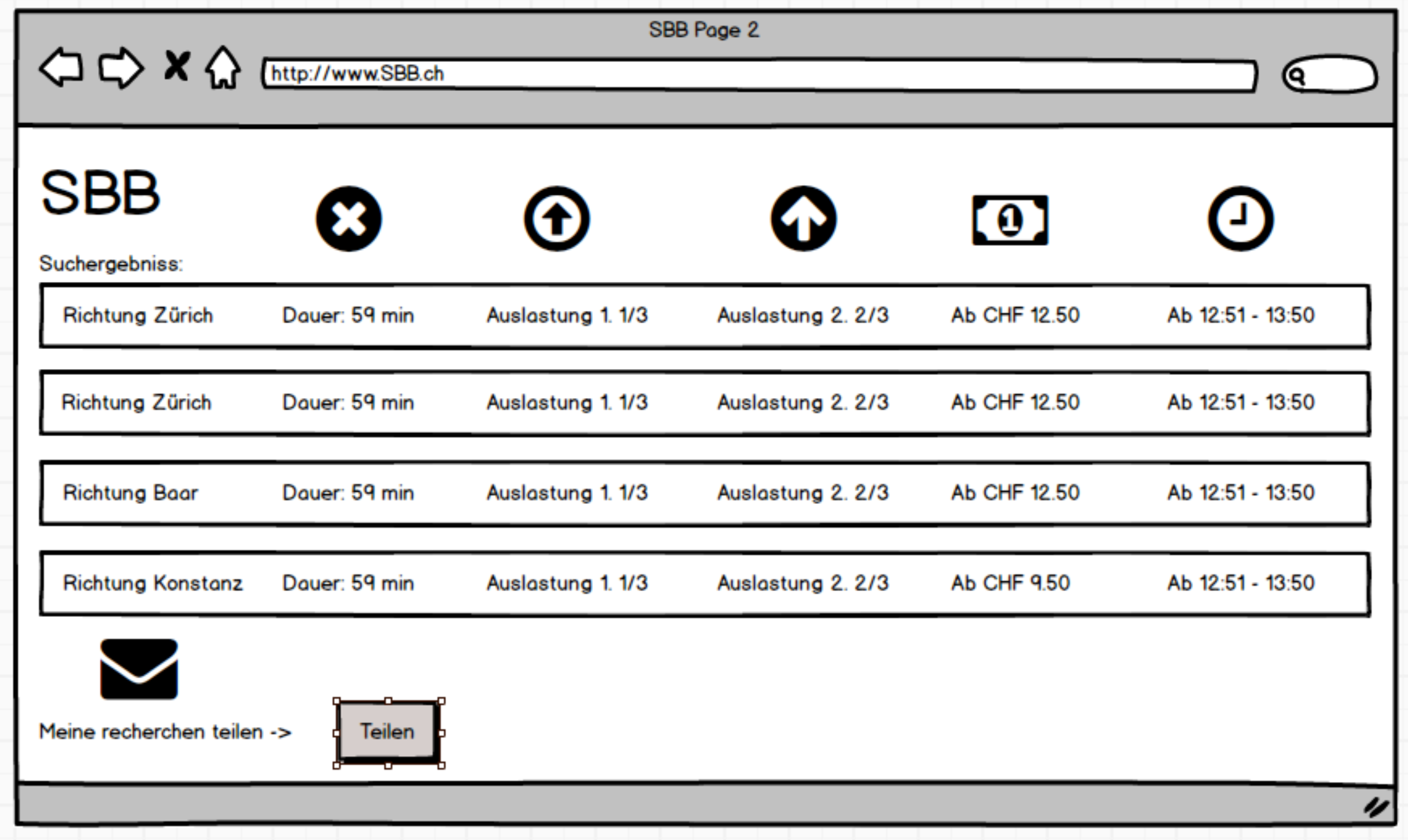
Main Site



Mockups

Ich habe versucht, das Design schlicht und einfach zu halten, sodass man ohne gross zu überlegen weiss, was man machen muss.

Second Site



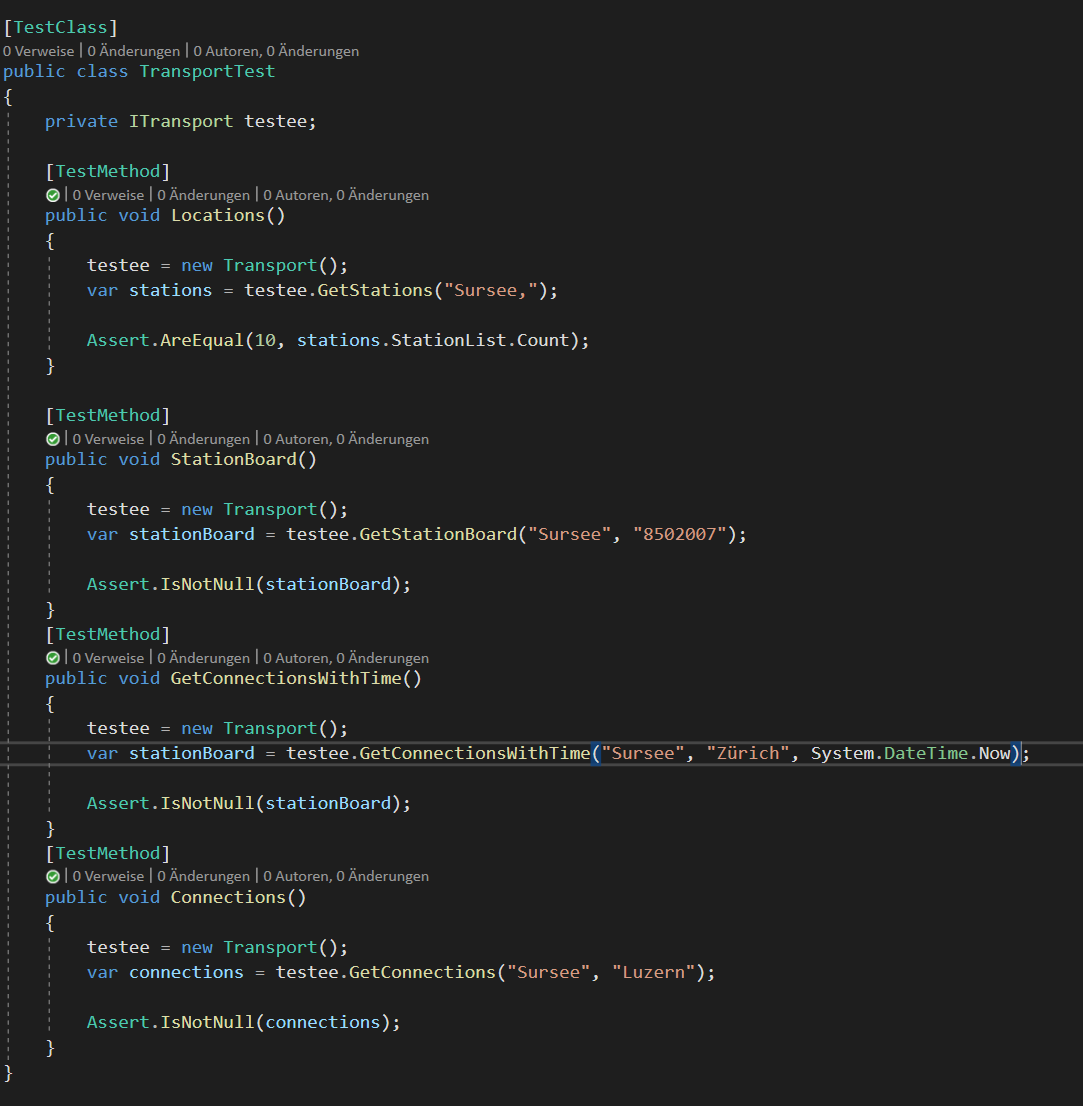
Mockups

Nearest Station



Testfälle

Alle testcodes die ich habe die später auch erfolgreich durch den test sind:



Testprotokoll

Alle Tests wurden erfolgreich abgeschlossen und sogar einer hinzugefügt 😊



Installationsanleitung

Hier eine «Schritt für Schritt» Anleitung, wie man sich meine «SBB»-App lokal herunterladen kann:

1. Man geht auf github.com und klickt auf «Nikola/modul-318-student
2. Danach muss man es einfach nur noch als Zip-Datei runterladen
3. Man öffnet den Speicherort, in dem von der Datei welche man als zip heruntergeladen hat
4. Dort öffnet man den Ordner «Download App»
5. Als nächstes doppelklickt man die Datei «Download App»
6. Klicken sie auf weiter
7. Entweder man kann seinen eigenen Speicher-Pfad Entweder man kann seinen eigenen Speicher-Pfad angeben oder den vom
8. Klicken sie auf weiter
9. weiter drücke und dann auf Schließen (wurde erfolgreich heruntergeladen).