Projektarbeit TransportApp

Inhalt

[Einleitung 2](#_Toc25833673)

[Anforderungen 3](#_Toc25833674)

[Mockup 4](#_Toc25833675)

[Use Cases 6](#_Toc25833676)

[Testing 8](#_Toc25833677)

[Installation 8](#_Toc25833678)

[Programmierrichtlinien 8](#_Toc25833679)

[Naming Conventions 8](#_Toc25833680)

[Declaration 8](#_Toc25833681)

[Comments 8](#_Toc25833682)

[Statements 8](#_Toc25833683)

# Einleitung

Der zweck dieses Dokument ist die Aufgabenstellung, gewisse Schritte und die Installation des Projektes also die TransportApp festzuhalten. Genauer noch, was musste ich machen, wie ich angefangen haben(nicht zu genau festgehalten), meine Planung, Probleme, die noch vorhanden sind und zu guter Letzt das Endprodukt nach einer Woche.

## Aufgabenstellung

Die Aufgabe für dieses Projekt war eine Applikation zu machen, welche alle Anforderungen erfüllt die unter dem Punkt "Anforderungen" zu finden sind. Dafür haben wir eine Solution bekommen, welche schon alle nötigen Funktionen von der TransportApi implementiert hat.

# Anforderungen

Priorities: 1 = must / 2 = should / 3 = nice to have

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Beschreibung | Priorität | Gemacht |
| A001 | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | 1 | Ja |
| 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. | 1 | Ja |
| 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. | 1 | Ja |
| A004 | 04 Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Suchresultate erscheinen, damit ich effizienter nach Stationen suchen kann. | 2 | Ja |
| 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 | 2 | Ja |
| A006 | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. | 3 | Ja |
| A007 | Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann. | 3 | Nein |
| A008 | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. | 3 | Nein |

## Erfüllt

Erfüllt wurden die Anforderungen A001-A006. A007 und A008 konnte ich leider nicht erfüllen, da ich keine Zeit mehr hatte. Die A007 musste ich leider im Code Behind umsetzen, da das Custom Control leider mit MVVM nicht funktioniert hat.

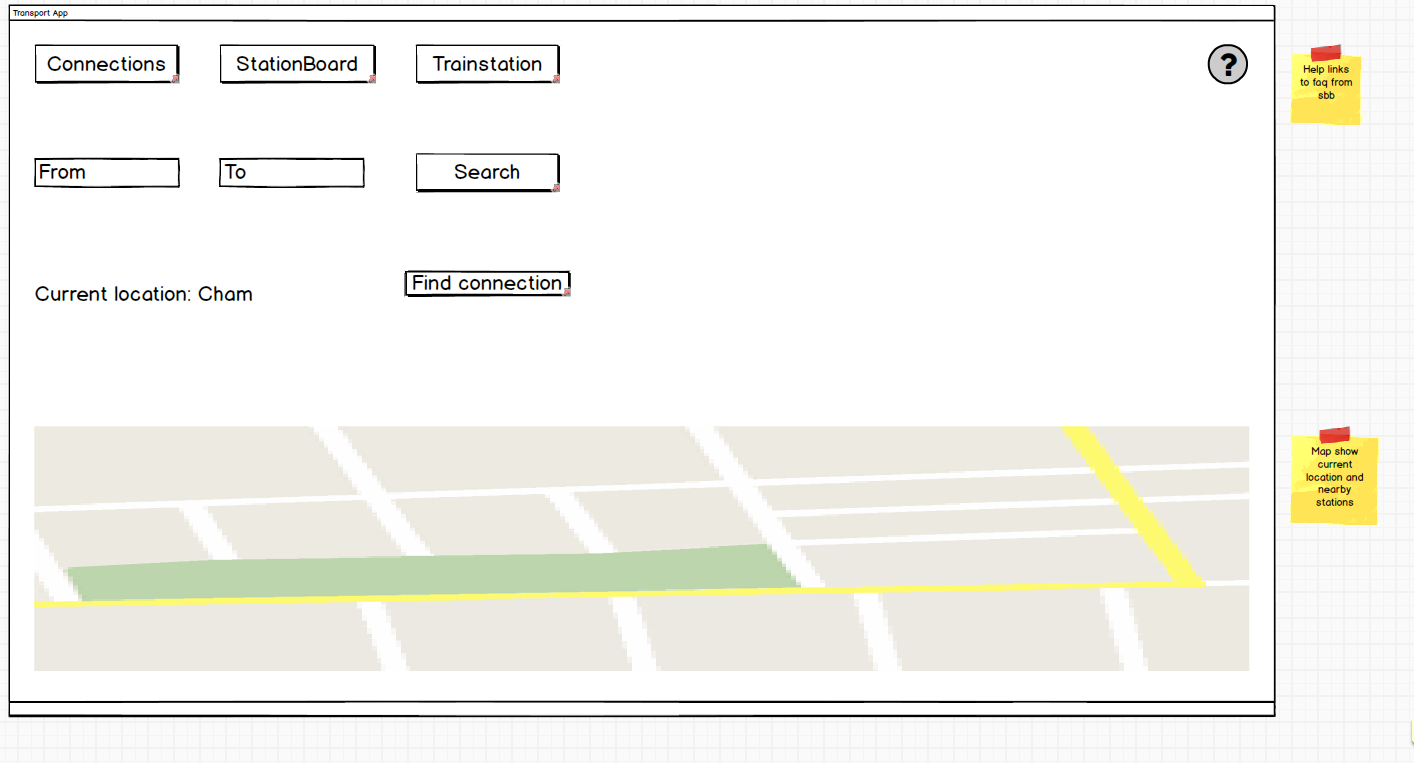
## Known-Issues

Falls man viele Zeichen in ein Stationsfeld, welche nicht gefunden werden können, gibt es Performance Probleme.

Falls ein Station vorhanden ist, gibt die Suche immer noch etwas zurück obwohl eine Station nicht existiert. Die Station, die nicht existiert wird, einfach von der Api mit einem Standard-Ort ersetzt.

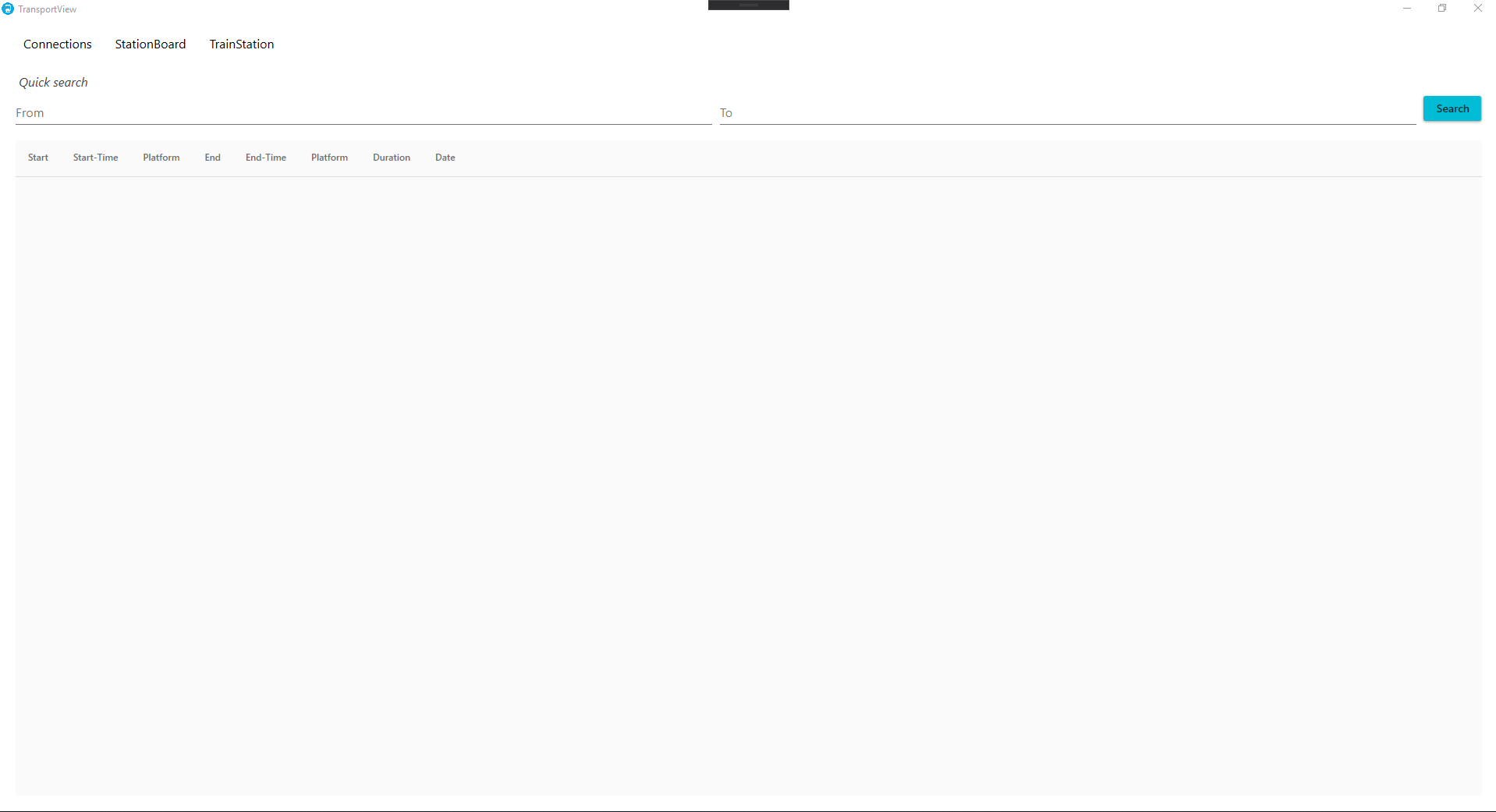
# Mockup

Dies ist die Startseite der Applikation.

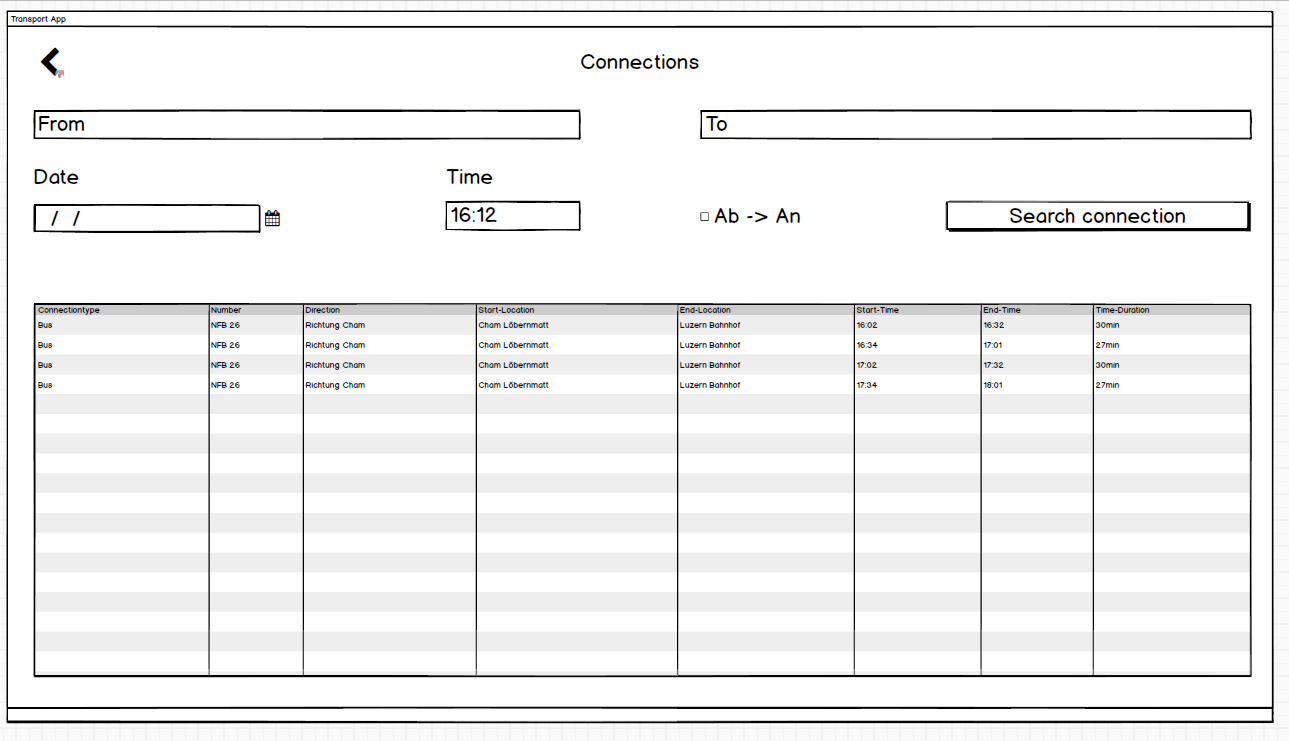


So sieht nun das Fenster wirklich aus. Die Map und die Current location habe ich nicht umgesetzt auf der Startseite, aber das Prinzip mit der schnell Suche habe ich gelassen. Ich habe einfach noch ein Grid hinzugefügt für die Verbindungen. Vorhher wollte ich diese im "Connections" Fenster abfüllen.

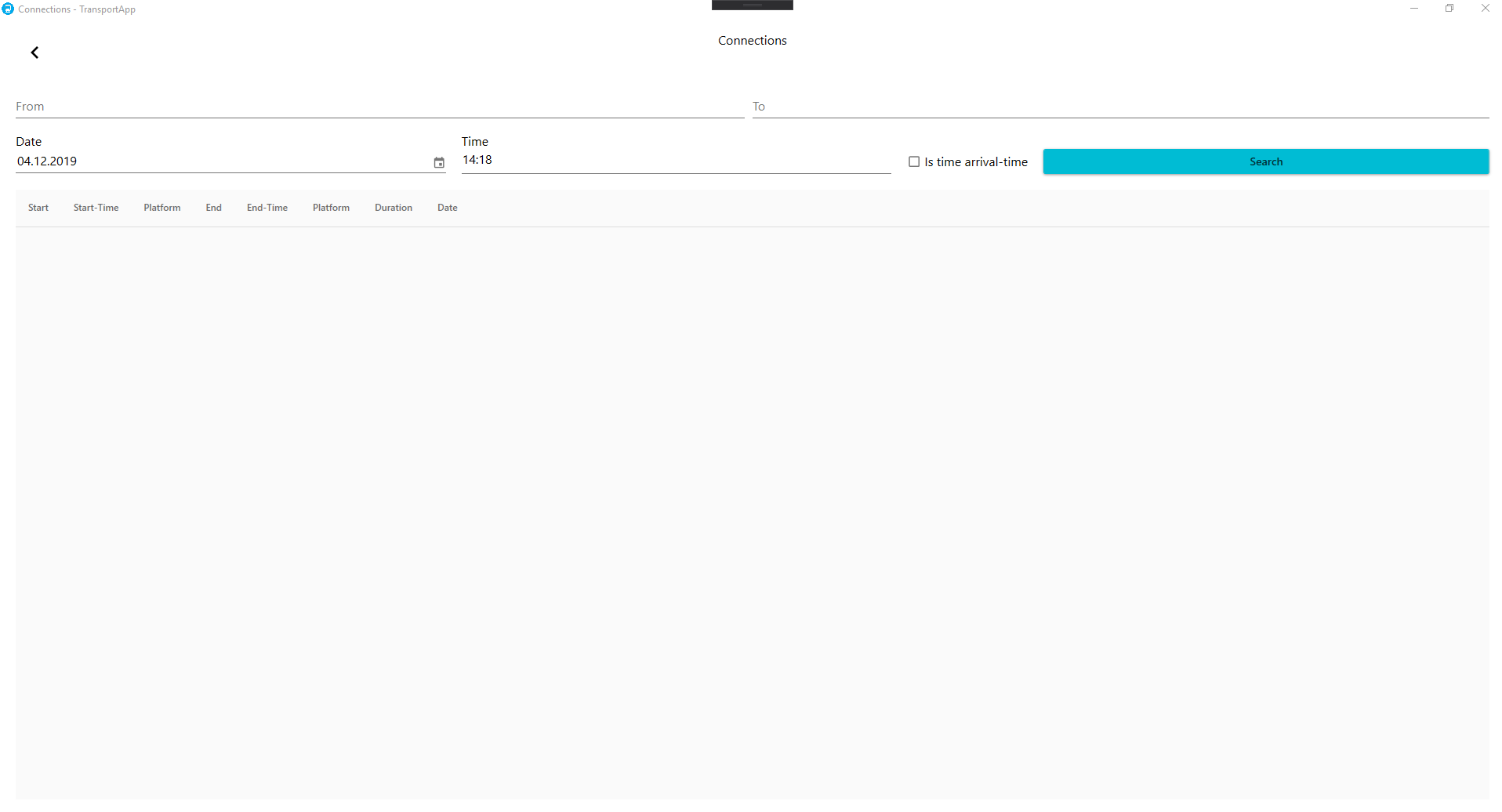
Die "Current Location" habe ich weggellasen da ich keine einfache Möglichkeit gefunden habe, den aktuellen Ort herauszufinden. Die Map habe ich deswegen auch weggelassen.



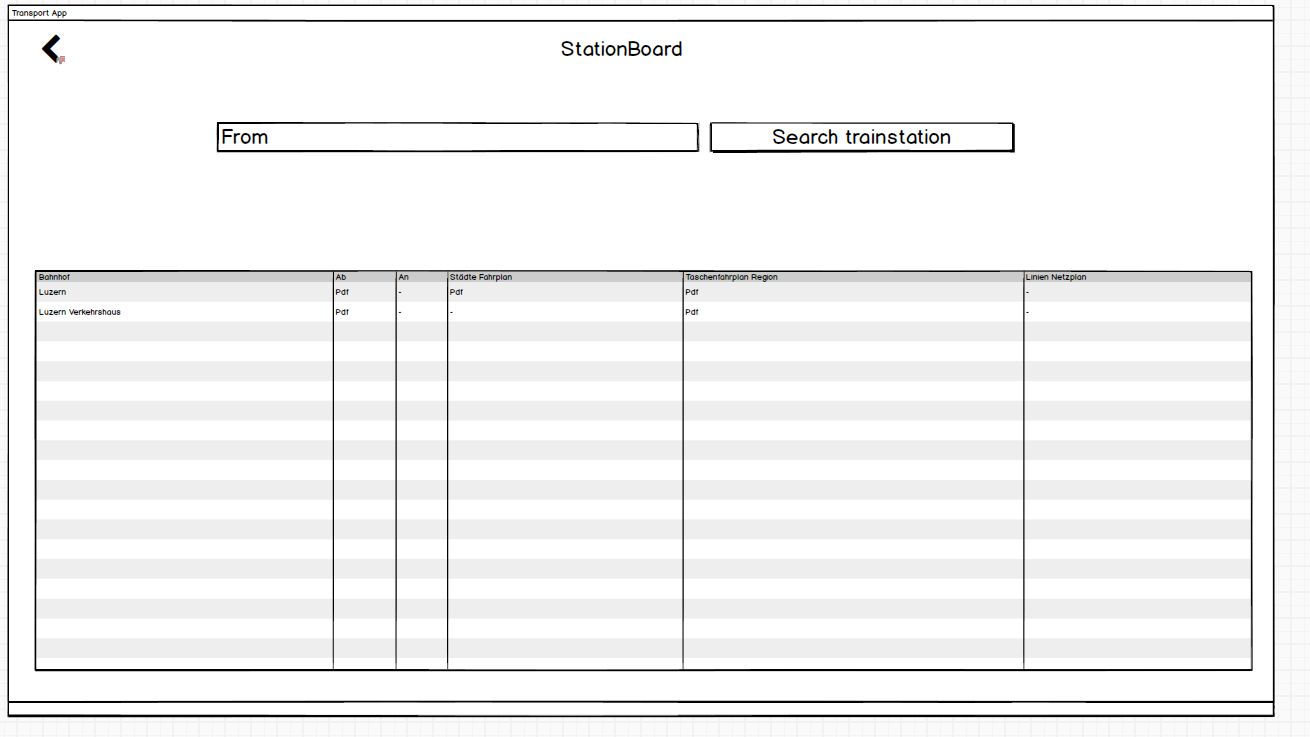
Auf diesem Fenster kömmem Verbindungen gesucht und angezeigt werden.



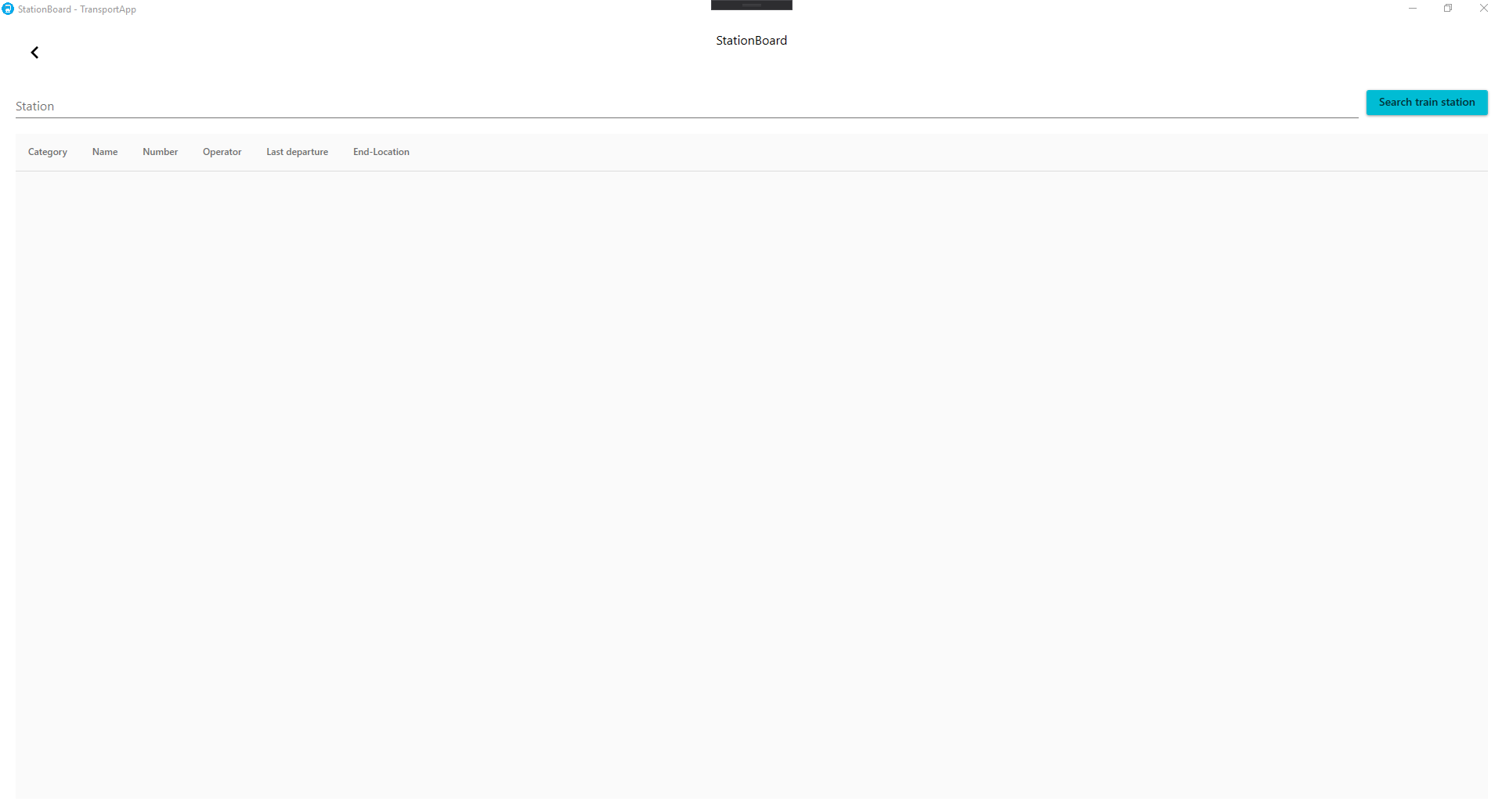
Das ist das Connection-Fenster umgesetzt. Da habe ich alles so gemacht wie geplant.



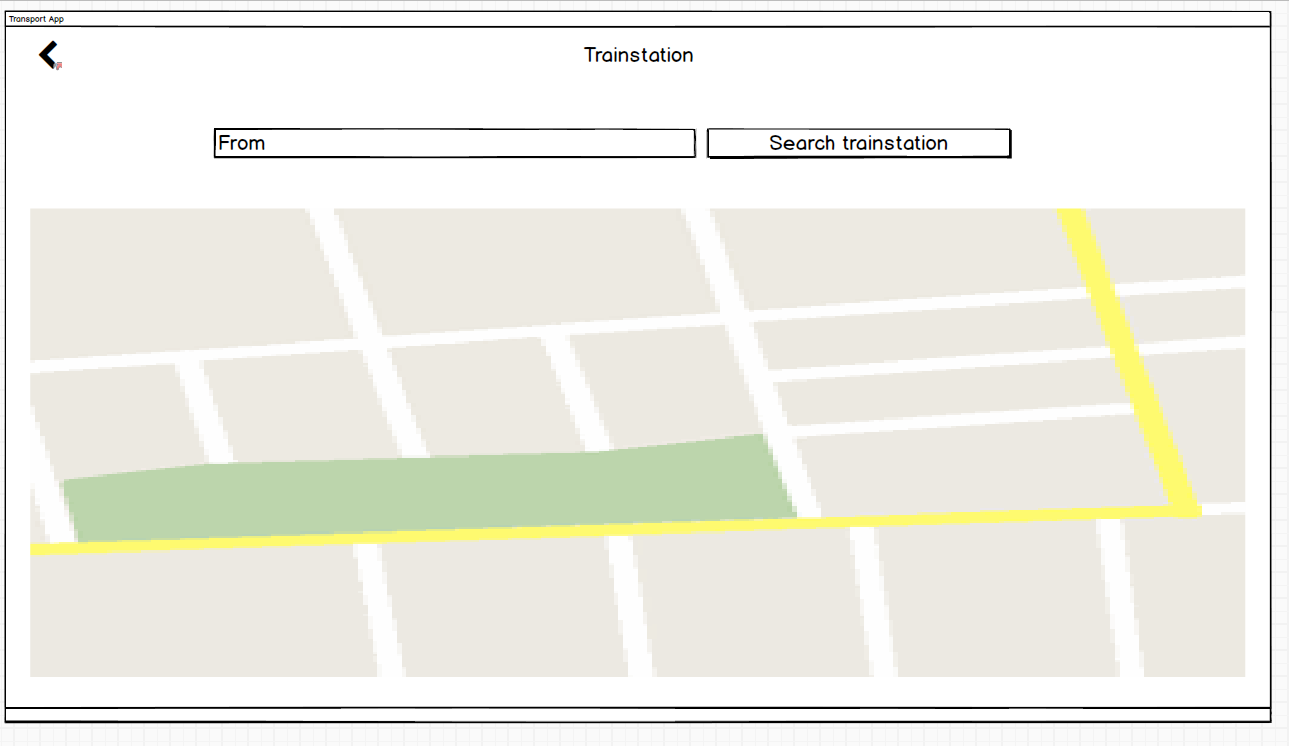
Dieses Fenster dient dazu eine Stationstabelle für eine bestimmte Station anzuzeigen.



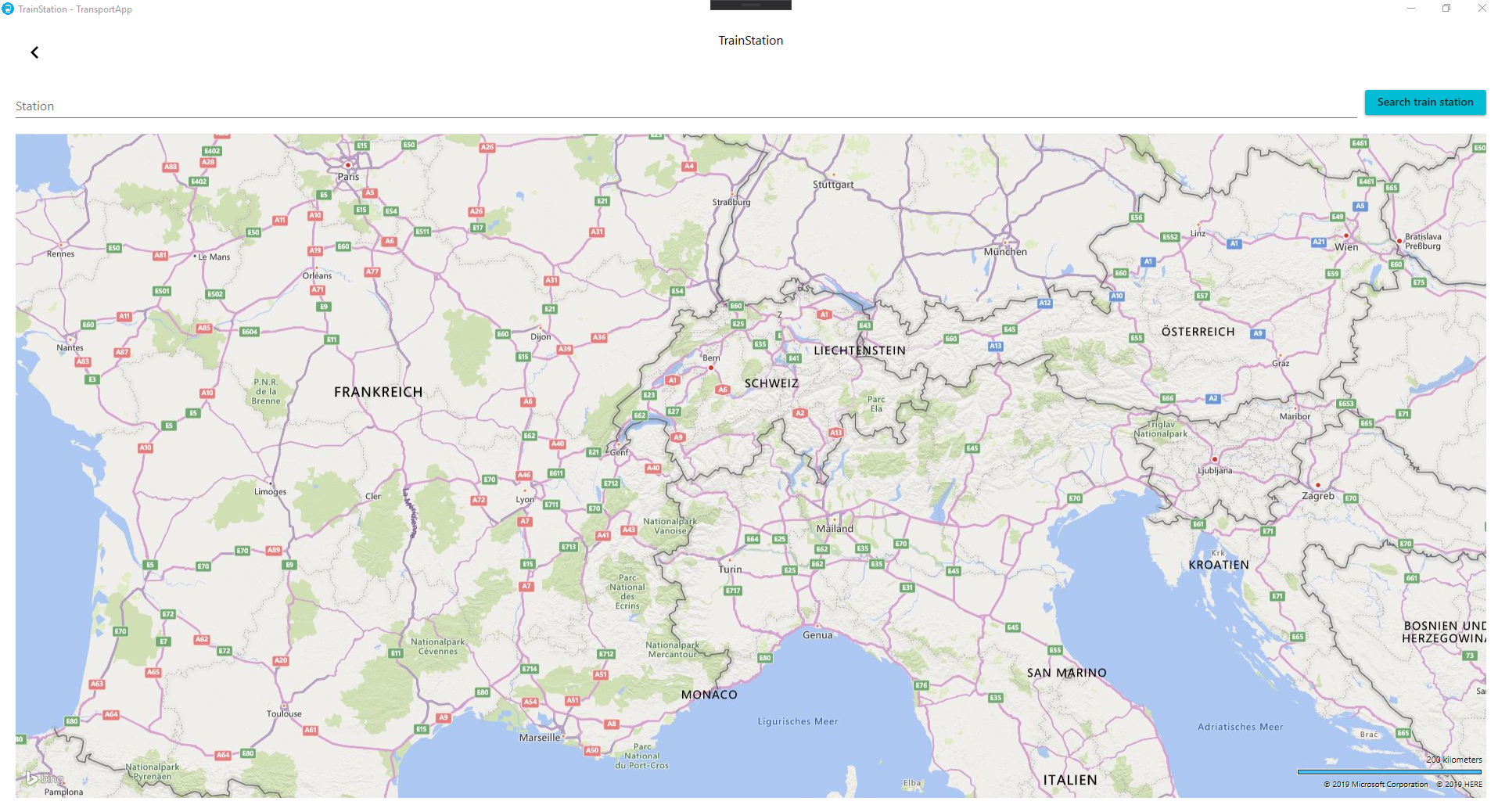
So sieht das Station-Board-Fenster aus, auch da nichts gross anders. Das einzige ist, es gibt kein Pdf, wie oben dargestellt, im Grid sondern es werden gleich Daten abgefüllt. Und die Station-TextBox und den Button habe ich einfach noch, schönheitshalber bis zum Rand gemacht.



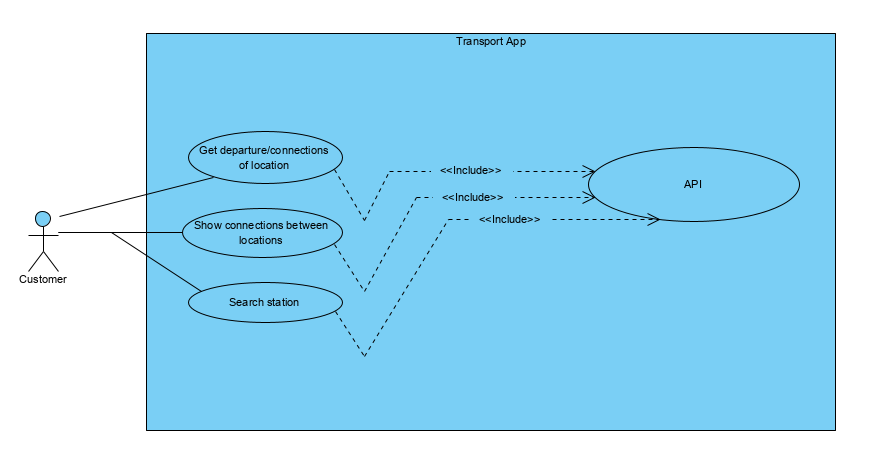
Mit diesem Fenster kann man eine Zugstation finden auf der Karte und die Umgebung anschauen.



So sieht die Umsetzung aus, auch hier habe ich einfach den Button und die TextBox bis zur Seite gemacht, sonst ist alles gleich.



# Use Cases

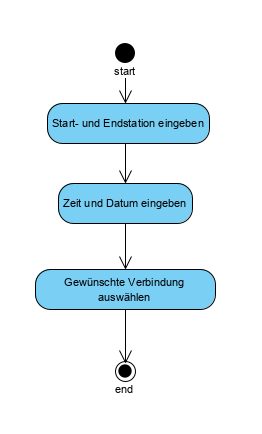


|  |  |
| --- | --- |
| Use Case | A001, Search station |
| Beschreibung | Kunde der Transport App, möchte nicht den vollständigen Name eingeben müssen. Sondern Vorschläge angezeigt bekommen. |
| Akteur | Eine zufällige Person, welche die Transport App heruntergeladen at. |
| Vorbedingungen | Der Kunde muss eine Internetverbindung haben. |
| Ablauf | 1. Auf Verbindung suchen klicken 2. Station Name anfangen einzugeben 3. Vorschläge werden angezeigt und sind auswählbar |
| Ergebnis | Ein oder mehrere Ergänzungsvorschläge für den bereits eingeben Text wird angezeigt. |

|  |  |
| --- | --- |
| Use Case | A002, Show connections between stations |
| Beschreibung | Kunde der Transport App gibt zwei Stationen ein und möchte mindestens 5 Verbindungen zur gewünschten Zeit angezeigt bekommen. |
| Akteur | Eine zufällige Person, welche die Transport App heruntergeladen at. |
| Vorbedingungen | Der Kunde muss eine Internetverbindung haben. |
| Ablauf | 1. Auf "Verbindung suchen" klicken 2. Start- und Endstation eingeben 3. Zeit und Datum eingeben 4. Auf "suchen" klicken 5. Fünf Verbindungen werden angezeigt |
| Ergebnis | Fünf Verbindungen werden anzeigt von der Startstation zur Endstation. |

|  |  |
| --- | --- |
| Use Case | A003, Get Departure table of location |
| Beschreibung | Kunde der Transport App kann eine Station eingeben und bekommt alle Verbindungen ab dieser Station. |
| Akteur | Eine zufällige Person, welche die Transport App heruntergeladen at. |
| Vorbedingungen | Der Kunde muss eine Internetverbindung haben. |
| Ablauf | 1. Kunde klickt auf "Abfahrtsplan" 2. Kunde gibt eine Station ein 3. Es zeigt alle Abfahrtspläne an |
| Ergebnis | Eine Art Abfahrtstafel wird angezeigt für alle Linien, die an dieser Haltstelle halten. |

Aktivität Diagramm zu A002

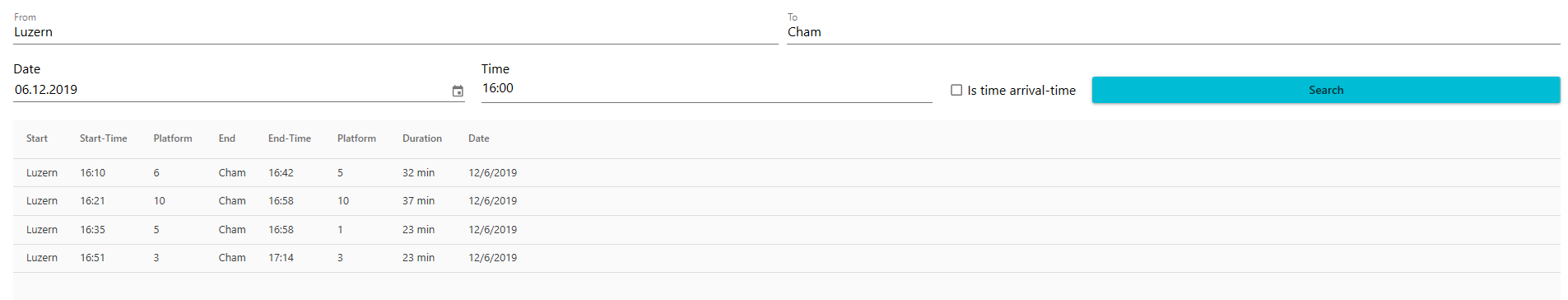


# Testing

Search Station, A001

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | In der TextBox für die "Start-Location" "Luz" eingeben. Und "Luzern" aus der List auswählen. | Luzern wird zuoberst in der Liste angezeigt und wenn es ausgewählt wird, wird es in die TextBox abgefüllt und die Liste mit den Vorschlägen verschwindet. |
| 2 | In der TextBox für die "End-Location" "Cha" eingeben. Und "Cham" aus der List auswählen. | Cham wird zuoberst in der Liste angezeigt und wenn es ausgewählt wird, wird es in die TextBox abgefüllt und die Liste mit den Vorschlägen verschwindet. |
| 3 | Dann klickt man auf den Date Picker geklickt und wählt das aktuelle Datum in 2 Tagen. | Ein Fenster mit dem aktuellen Datum selektiert erscheint. Das gewünschte Datum ist wählbar. |
| 4 | Darauf wird noch die Zeit auf 16:00 gesetzt. | Die Zeit ist nun auf 16:00. |
| 5 | Es wird auf Suchen geklickt. | Nun werden 4 Datensätze angezeigt, die von Luzern nach Cham fahren. Alle haben eine Abfahrtszeit nach "16:00". Das Datum entspricht dem gewählten Datum. |

Dieser Test wurde am 04.12.2019 von Raphael Härtel ausgeführt und gab folgendes Resultat zurück. Das Resultat stimmte mit dem Erwarteten Resultat überein.



Show connections between stations, A002

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | In der TextBox für die "Start-Location" "Luz" eingeben. Und "Luzern" aus der List auswählen. | Luzern wird zuoberst in der Liste angezeigt und wenn es ausgewählt wird, wird es in die TextBox abgefüllt und die Liste mit den Vorschlägen verschwindet. |
| 2 | In der TextBox für die "End-Location" "Cha" eingeben. Und "Cham" aus der List auswählen. | Cham wird zuoberst in der Liste angezeigt und wenn es ausgewählt wird, wird es in die TextBox abgefüllt und die Liste mit den Vorschlägen verschwindet. |
| 3 | Dann klickt man auf den Date Picker geklickt und wählt das aktuelle Datum in 2 Tagen. | Ein Fenster mit dem aktuellen Datum selektiert erscheint. Das gewünschte Datum ist wählbar. |
| 4 | Darauf wird noch die Zeit auf 16:00 gesetzt. | Die Zeit ist nun auf 16:00. |
| 5 | Es wird auf Suchen geklickt. | Nun werden 4 Datensätze angezeigt, die von Luzern nach Cham fahren. Alle haben eine Abfahrtszeit nach "16:00" |

Get Connection of location, A003

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | In der TextBox für die "Start-Location" "Luz" eingeben. Und "Luzern" aus der List auswählen. | Luzern wird zuoberst in der Liste angezeigt und wenn es ausgewählt wird, wird es in die TextBox abgefüllt und die Liste mit den Vorschlägen verschwindet. |
| 2 | In der TextBox für die "End-Location" "Cha" eingeben. Und "Cham" aus der List auswählen. | Cham wird zuoberst in der Liste angezeigt und wenn es ausgewählt wird, wird es in die TextBox abgefüllt und die Liste mit den Vorschlägen verschwindet. |
| 3 | Dann klickt man auf den Date Picker geklickt und wählt das aktuelle Datum in 2 Tagen. | Ein Fenster mit dem aktuellen Datum selektiert erscheint. Das gewünschte Datum ist wählbar. |
| 4 | Darauf wird noch die Zeit auf 16:00 gesetzt. | Die Zeit ist nun auf 16:00. |
| 5 | Es wird auf Suchen geklickt. | Nun werden 4 Datensätze angezeigt, die von Luzern nach Cham fahren. Alle haben eine Abfahrtszeit nach "16:00" |

- Testprotokoll mit Durchführungsdatum der Tests, Name des Testers und Protokollierung der tatsächlichen Testresultate.

# Installation

1. Zuerst muss man Git installieren.
2. Danach muss man zu dem Pfad gehen, an dem man das Programm möchte.  
   Zu diesem Pfad muss man im CMD wechseln und dann   
   "Git clone <https://github.com/LostLeqend/modul-318-student.git>" eingeben.
3. Dann das Projekt "modul-318-student" öffnen.
4. Dann in den "TransportApp" Ordner gehen.
5. Und zu guter Letzt in den "bin\Debug" Ordner und die "TransportApp.exe" starten.

Gesamter Pfad: "C:\**your own path**\modul-318-student\TransportApp\bin\Debug\TransportApp.exe"

Zum deinstallieren einfach den Ordner "modul-318-studen" löschen. An dem installierten Pfad.

# Programmierrichtlinien

## Naming Conventions

Alle Namen sind Englisch oder Deutsch, kein Mix.  
Keine Abkürzungen bei der Benennung.

Variablen sind klein geschrieben.  
Properties, Methoden, Klassen und Gui-Controls werden grossgeschrieben.  
Bei mehreren Wörtern sollen die nächsten Wörter gross sein.

Beispiel Variable: "\_numberOne" und nicht "NumberOne", bei lokalen variablen ohne bodenstrich. Also z.B. "numberOne".   
Bei einer Methode: "CalculateSinus()" und nicht "calculatesinus()".  
"Exception" und nicht "ex".

## Declaration

Zuerst sollte der Konstruktor sein, dann Properties und zuletzt Methoden.  
Deklaration von Properties sollen, wenn möglich im Konstruktor vorgenommen werden.

## Comments

Methoden sowie auch Properties sollen kommentiert sein. Innerhalb von Methoden sollten keine Kommentare vorhanden sein, da der Code selbsterklären sein sollte.

## Statements

Bei Statements die mehr als eine Zeile beinhalten sollten immer geschweifte Klammern verwendet werden.  
Falls aber z.B. ein If/Else nur eine Zeile Code beinhaltet dürfen die geschweifte Klammern weggelassen werden.

Keine Verschachtelung von Statements. Am besten invertiert man ein Statement, um dies zu vermeiden.