ÜK318 Projektarbeit:

Swiss Transport App

Yazdan Musa

KR2, ICT. Berufsbildung Zentralschweiz

12.12.2019 Adligenswil

# Inhaltsverzeichnis:

[Inhaltsverzeichnis: 2](#_Toc27568773)

[Einleitung 3](#_Toc27568774)

[Aufgabenstellung 3](#_Toc27568775)

[Anforderungen 3](#_Toc27568776)

[Erfüllt 3](#_Toc27568777)

[Teilweise Implementiert 3](#_Toc27568778)

[Known-Issues 4](#_Toc27568779)

[Programmierrichtlinien 4](#_Toc27568780)

[Naming Conventions 4](#_Toc27568781)

[Declaration 4](#_Toc27568782)

[Comments 4](#_Toc27568783)

[Statements 4](#_Toc27568784)

[Mockups 4](#_Toc27568785)

[Hauptseite 5](#_Toc27568786)

[Departure Board: 5](#_Toc27568787)

[Use Cases 5](#_Toc27568788)

[A001, Textergänzung bei der Eingabe von Stationsnamen 6](#_Toc27568789)

[A002, Verbindungsmöglichkeiten zwischen A und B anzeigen 6](#_Toc27568790)

[A003, Abfahrtstafel anzeigen 6](#_Toc27568791)

[Exceptions und Testing 7](#_Toc27568792)

[Testfall 1 (Was passiert wenn man keine Station ausgewählt hat?) 7](#_Toc27568793)

[Testfall 2 (Falsches Zeit Format) 8](#_Toc27568794)

[Testfall 3 (Was passiert wenn man keine Internetverbindung hat?) 9](#_Toc27568795)

[Installation 9](#_Toc27568796)

# Einleitung

Der Zweck dieses Dokuments ist es, die Aufgabe, bestimmte Schritte und die Installation des Projekts, also der TransportApp, zu beschreiben. Genauer gesagt, was ich tun musste, wie ich angefangen habe, meine Planung, die noch bestehenden Probleme und das Endprodukt am Ende der ÜK.

## Aufgabenstellung

Entwicklung einer Anwendung, die öffentliche Verkehrsverbindungen von Punkt A nach B anzeigt. Dazu gehören Funktionen wie eine eigenständige Abfahrtstabelle, die N Verbindungen von Punkt A aus anzeigt, eine Stationskarte, die die Position einer Station auf einer Karte mit GPS-Koordinaten anzeigt. Alle Daten werden vom SBB OpenData REST Service bereitgestellt und die Abhängigkeiten sind bereits im Projekt.

# Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Beschreibung** | **Priorität** | **Erfüllt** |
| 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 | Nicht ganz |
| 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.

## Teilweise Implementiert

Die Anforderung A007 ist teilweise implementiert. Das heisst die Fundament existiert. Die Location wird anhand der Public IP des Users geholt. Die Funktionen dafür sind GetPublicIP() und GetLocationFromIP().

GetPublicIP() braucht ein 3rd Party API (api.ipify.org)

GetLocationFromIP() auch. (ip-api.com)

## Known-Issues

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

# Programmierrichtlinien

## Naming Conventions

* Soll immer auf Englisch sein, keine Mischung.
* Generell PascalCase verwenden.
* Private Members in camelCase.
* Constants in UPPER\_CASE.
* Buttons mit «Btn\_» Präfix
* Text mit «Txt\_» Präfix, usw.

## Declaration

Constructor, Properties und Methods.

## Comments

Comments an einer separaten Zeile platzieren, nicht am Ende einer Zeile.

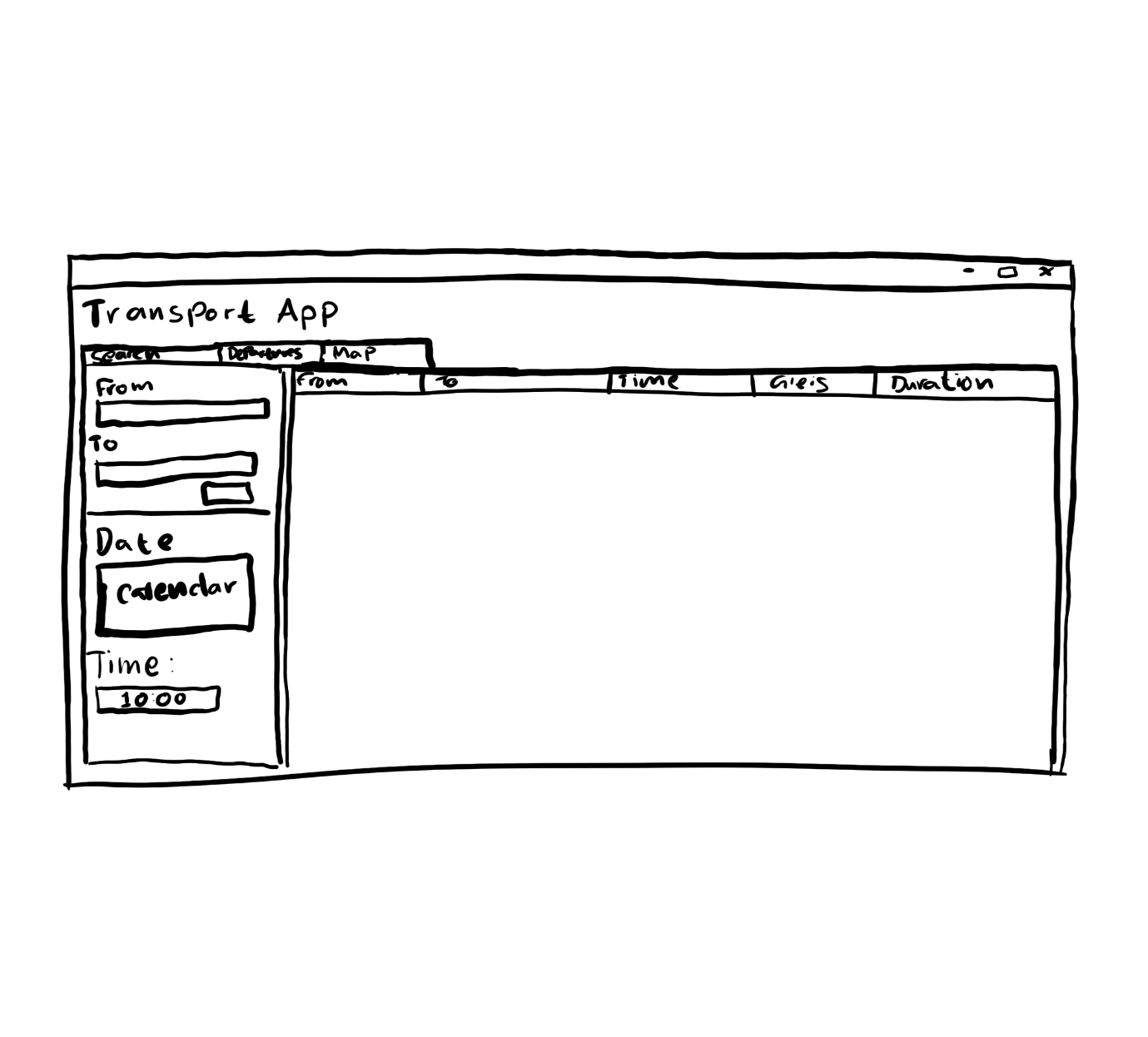
1 Space zwischen // und Anfang von Comment.

Comment mit « **.** » beenden.

## Statements

{ } bei Statements brauchen, falls es mehr als eine Zeile Code braucht.

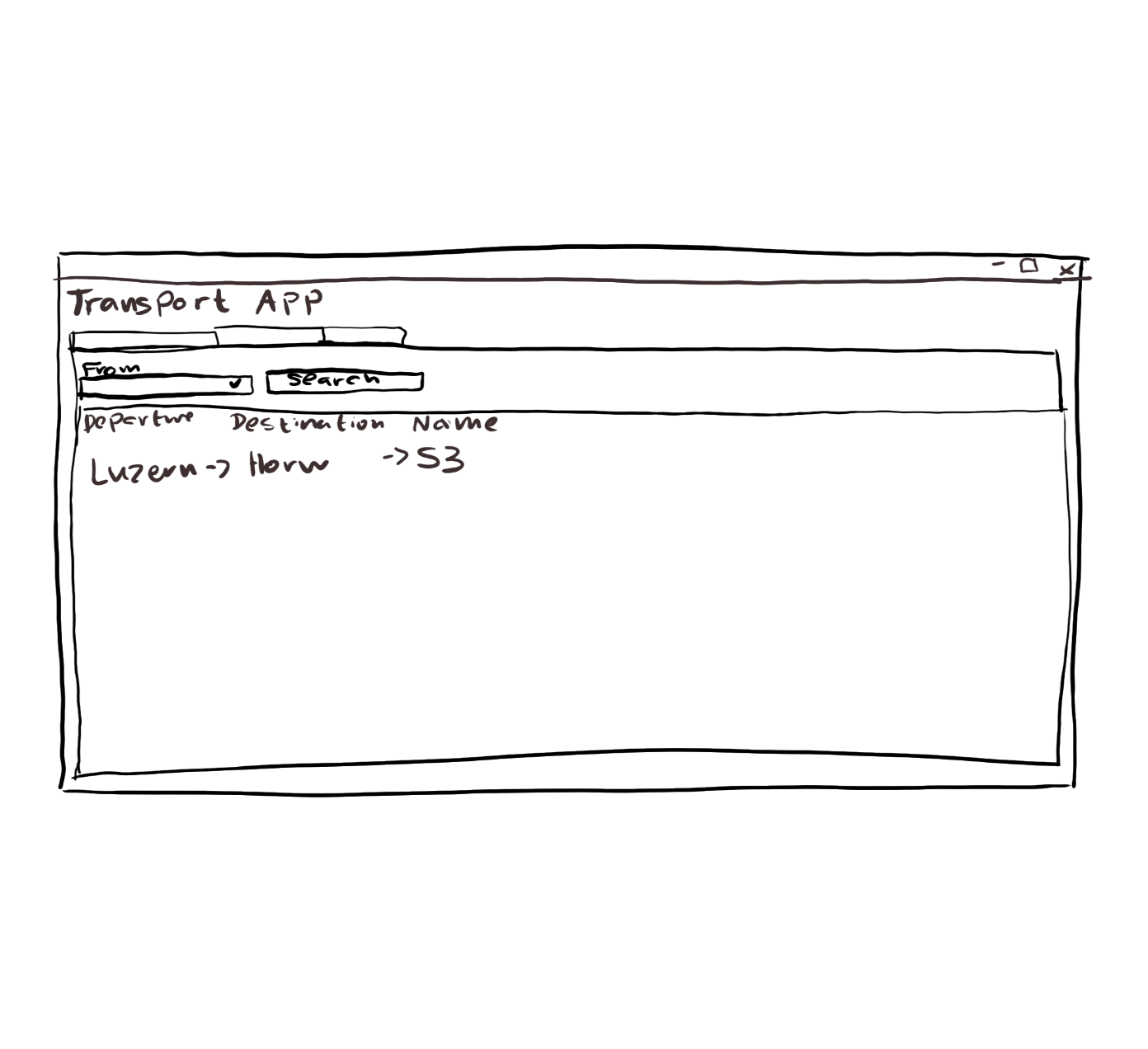
# Mockups

Die Mockups wurden in Adobe Photoshop von Hand gezeichnet.

## Hauptseite

Hier kann man Verbindungen Suchen. Man gibt ein Startstation, und Zielstation ein. Nach wünsch kann das Datum und Zeit angepasst werden. Benutzer sollte mit Tabulator und Enter arbeiten können. Folgende Details sollten angezeigt werden:

Abfahrtsstation, Zielstation, Abfahrt- und Ankunftszeiten, Gleis und Reisedauer.



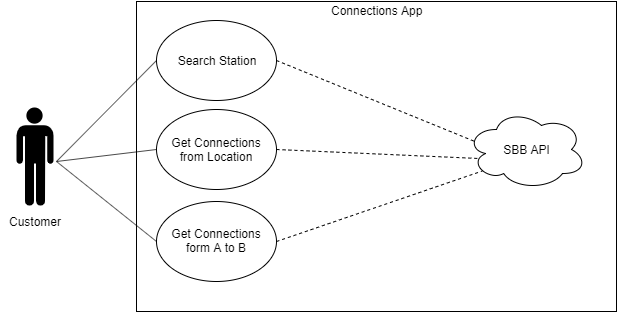
## Departure Board:

Hier sollte Man ein Station eingeben können und die nächste Verbindungen ab dem Station sehen.

Folgende Details sollten angezeigt werden:

Abfahrtsstation, Ziel und Name bzw. Linie.

# Use Cases



|  |  |
| --- | --- |
| **Use Case** | A001, Textergänzung bei der Eingabe von Stationsnamen |
| Beschreibung | Kunde möchte Textvorschläge bekommen, damit er die Stationsnamen nicht auswendig lernen muss. |
| Akteur | Kunde |
| Auslöser | Kunde fangt an zu tippen |
| Vorbedingungen | Der Kunde muss eine Internetverbindung haben und der REST-Dienst von SBB muss online sein. |
| Ergebnis | Ergänzungsvorschläge à la Google für den eingegebenen Text wird angezeigt |

|  |  |
| --- | --- |
| **Use Case** | A002, Verbindungsmöglichkeiten zwischen A und B anzeigen |
| Beschreibung | Kunde möchte 4 bis 5 Verbindungsmöglichkeiten sehen |
| Akteur | Kunde |
| Auslöser | Kunde gibt Ziel- und Ausgangsstationen ein und Sucht |
| Vorbedingungen | Der Kunde muss eine Internetverbindung haben und der REST-Dienst von SBB muss online sein. |
| Ergebnis | Fünf Verbindungen werden angezeigt |

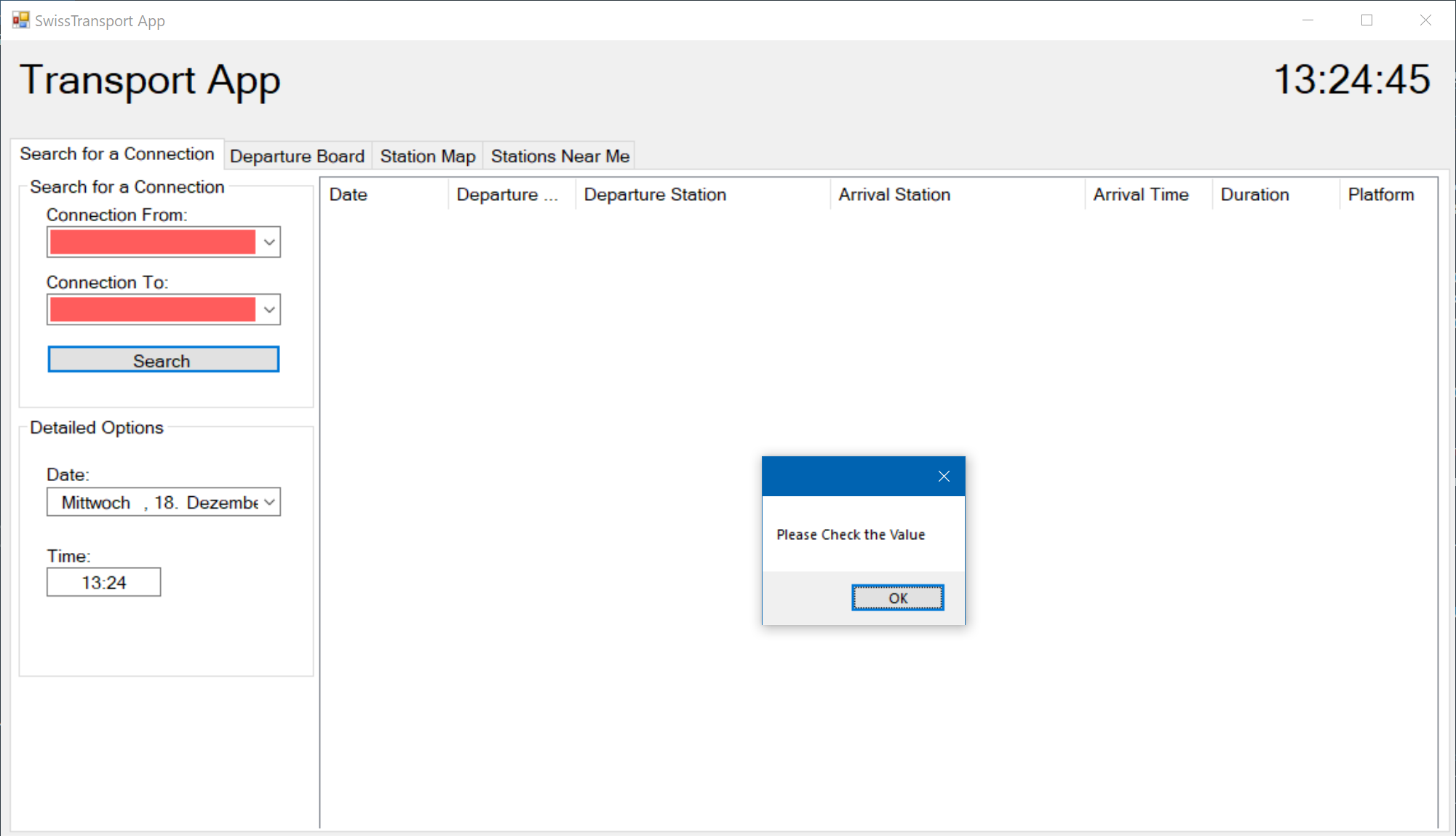
|  |  |
| --- | --- |
| **Use Case** | A003, Abfahrtstafel anzeigen |
| Beschreibung | Kunde möchte alle Verbindungen von ein “Home” Station sehen. |
| Akteur | Kunde |
| Auslöser | Kunde klickt auf Connections from Home |
| Vorbedingungen | Kunde muss eine Home Station definiert haben.  Der Kunde muss eine Internetverbindung haben und der REST-Dienst von SBB muss online sein. |
| Ergebnis | Abfahrtstafel mit alle Verbindungen wird angezeigt |

# Exceptions und Testing

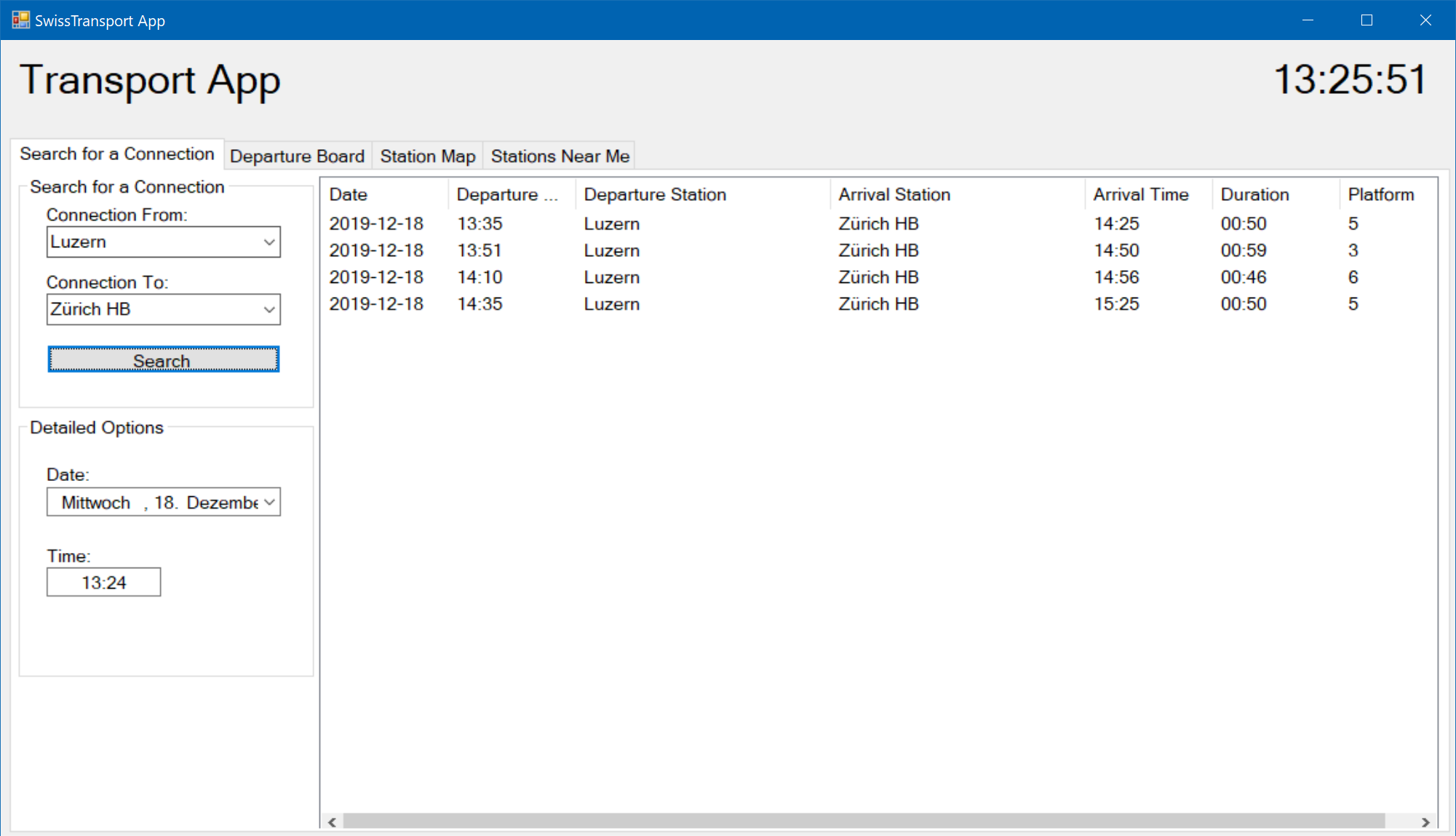
Ein Unittest wurde mit «Zeit» ergänzt. Restliche Testing wurde manuell von Hand gemacht.

## Testfall 1 (Was passiert wenn man keine Station ausgewählt hat?)

Wenn man keine Station auswählt, werden die falsche bzw. leere Felder Rot markiert und ein Pop-up Fenster erscheint.



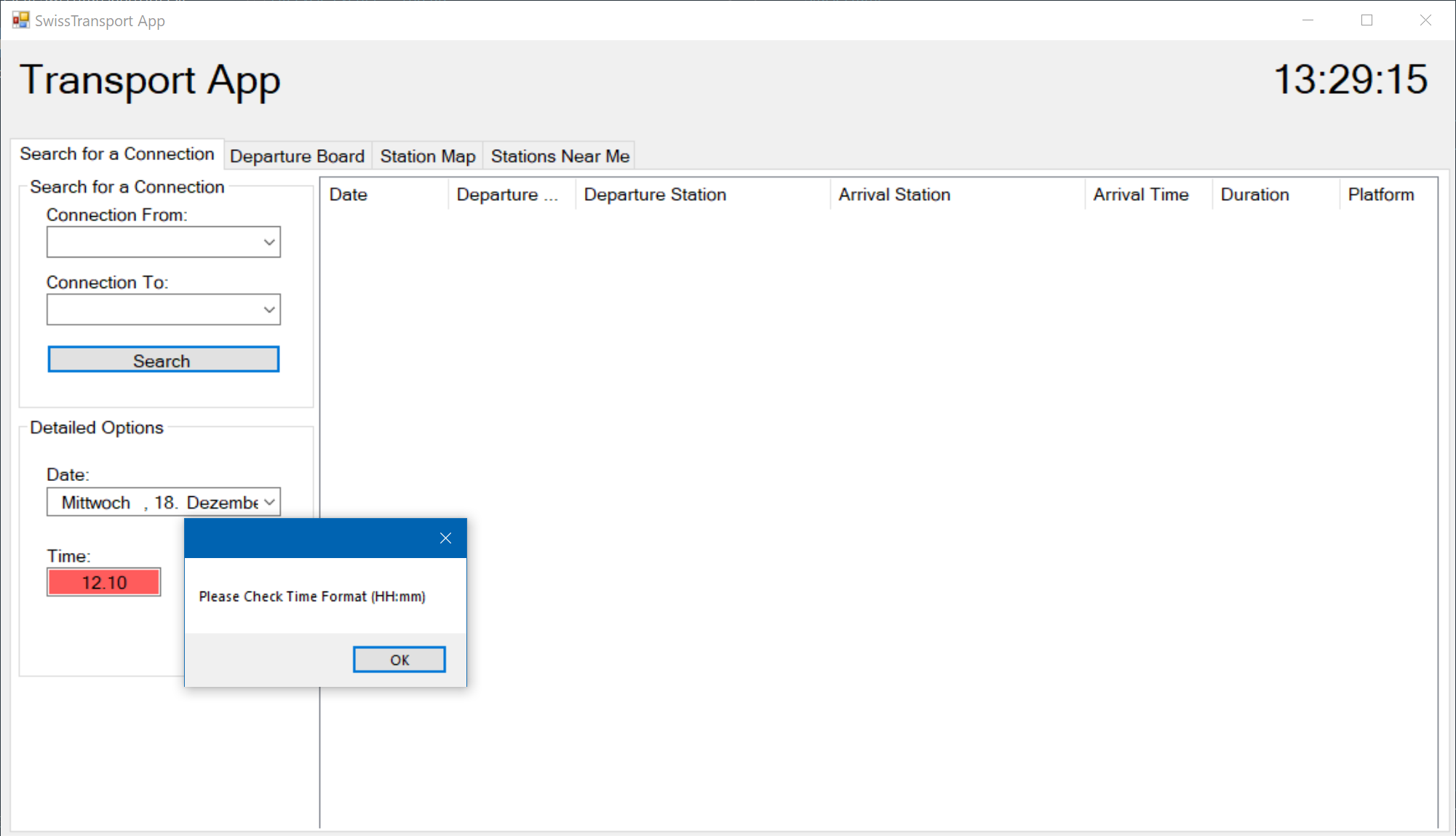
Im Gegensatz, wenn man alles richtig eingibt, erscheinen die nächste 4 Verbindungen wie gewünscht:



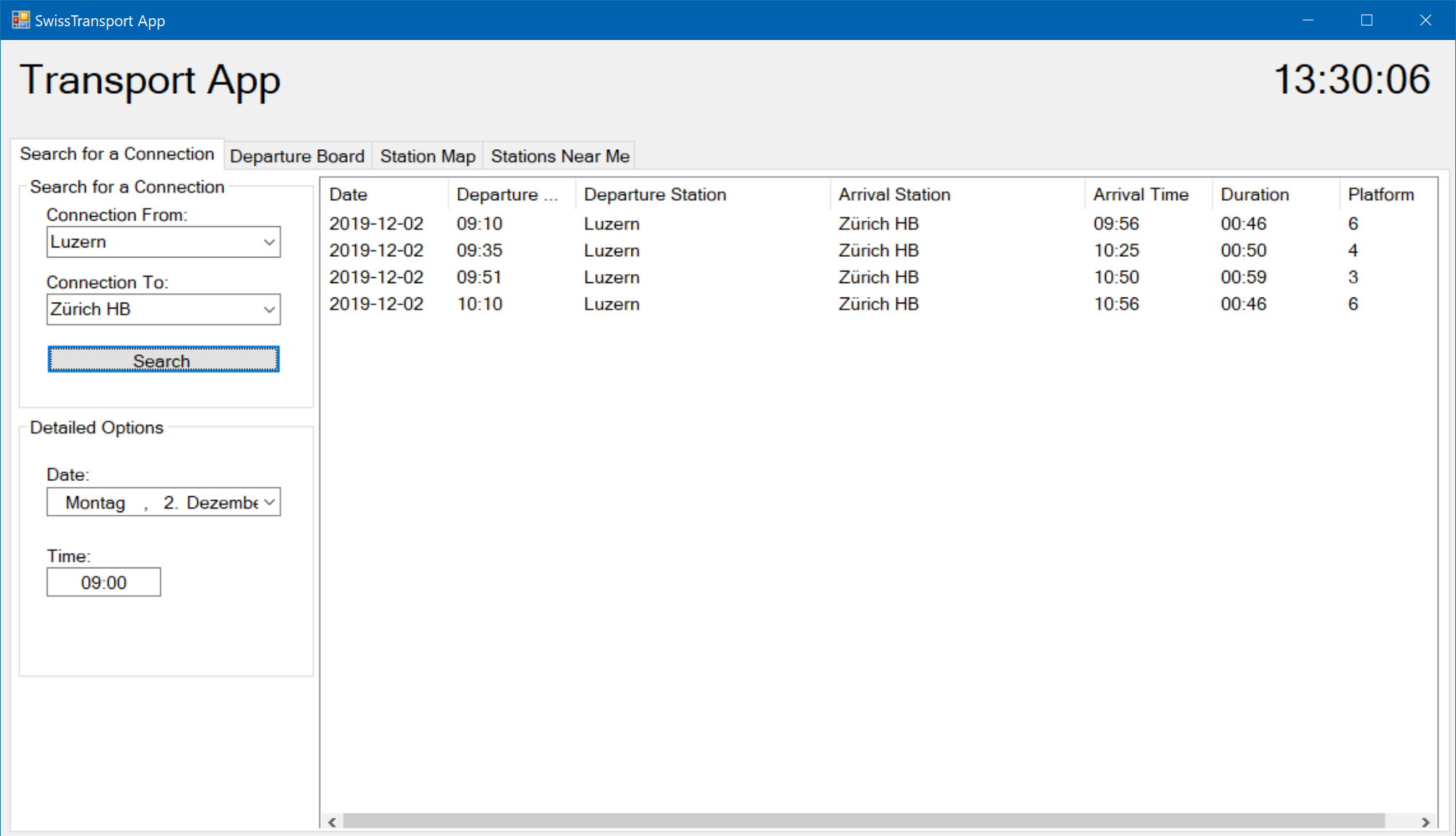
Dieser Test hat Yazdan Musa am 18.12.2019 um 13:25 erfolgreich ausgeführt.

## Testfall 2 (Falsches Zeit Format)

Die Zeittextbox akzeptiert nur das HH:mm Zeitenformat. Wenn man dies nicht einhält, bekommt man folgende Fehlermeldung:

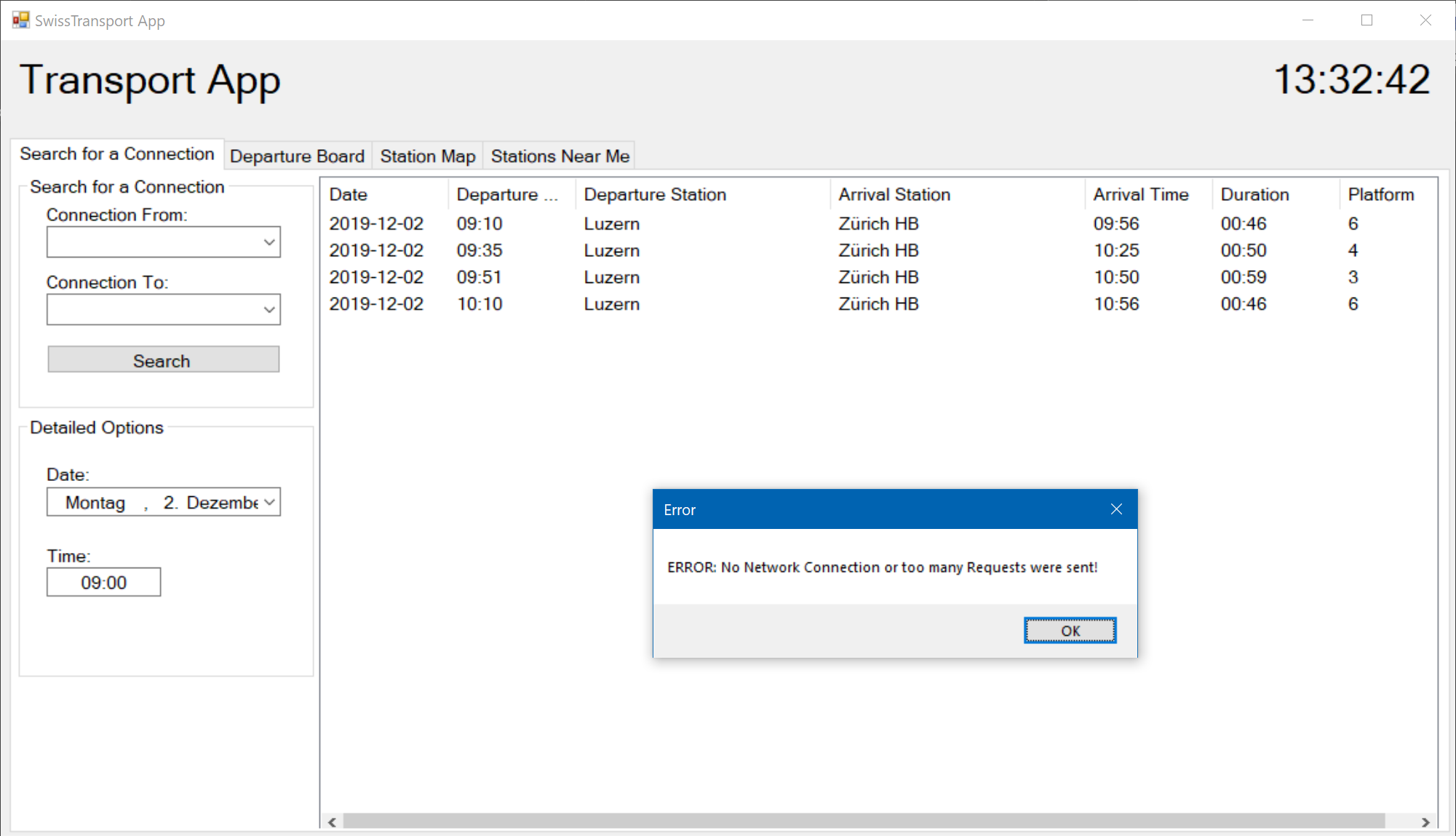


Wenn das Zeitformat stimmt, kann man Verbindungen aus einer beliebigen anderen Zeitpunkt sehen:



Dieser Test hat Florian Stettler am 18.12.2019 um 13:30 erfolgreich ausgeführt.

## Testfall 3 (Was passiert wenn man keine Internetverbindung hat?)

Wenn man keine Internetverbindung hat, oder wenn das API nichts zurückliefert, bekommt man folgende Fehlermeldung: 

Sonst funktioniert es wie gewöhnt.

Dieser Test hat Yazdan Musa am 18.12.2019 um 13:32 erfolgreich ausgeführt.

# Installation und Deinstallation

Ein Installer wird mitgeliefert.

Installationsanleitung:

Setup ausführen

Pfad auswählen

Installieren

Deinstallationsanleitung:

Setup ausführen

«Deinstallieren» wählen

Deinstallieren