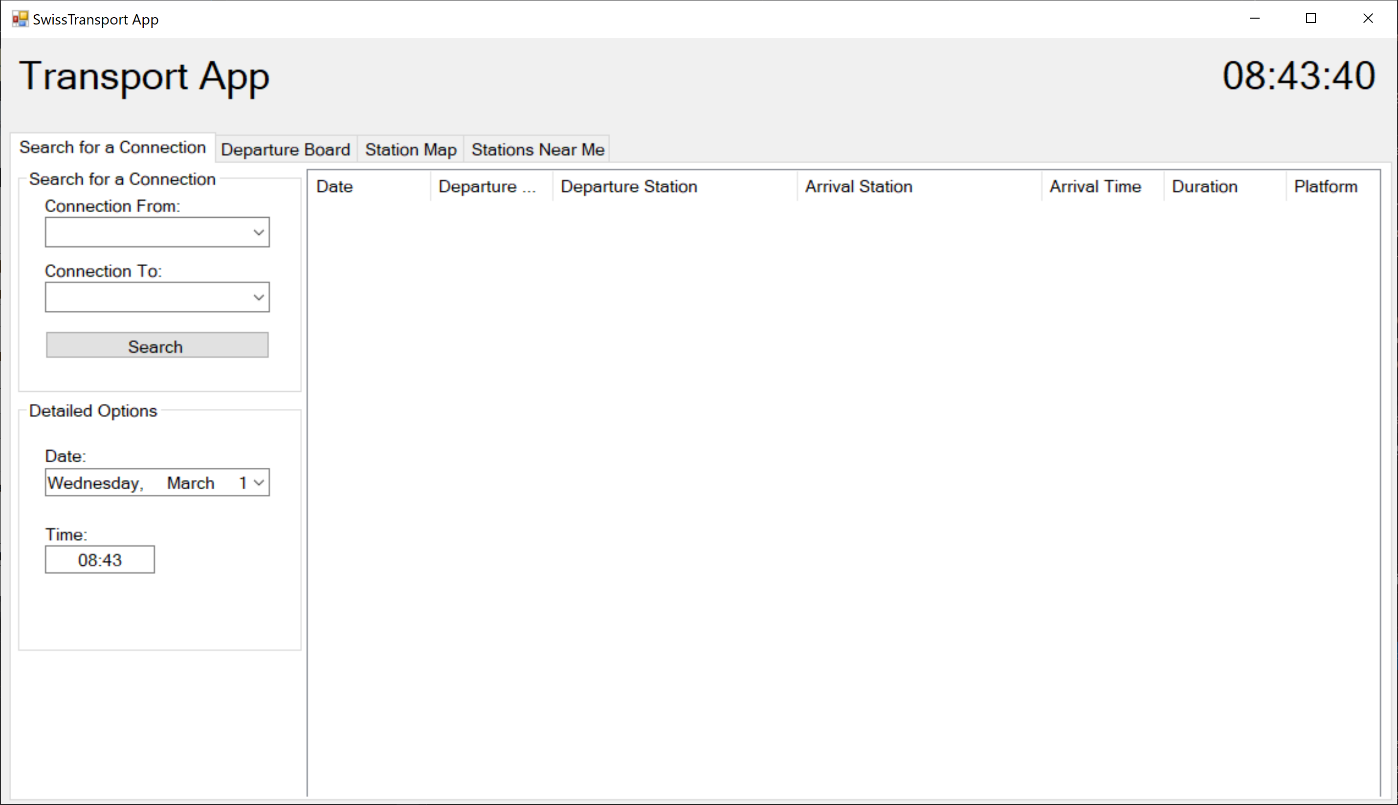
Dokumentation ÖV-Fahrplan



**Rouven Wicki**

ÜK M318

10.03.2020

Contents

[Einleitung 3](#_Toc34910863)

[Zweck 3](#_Toc34910864)

[Aufgabenstellung 3](#_Toc34910865)

[Anforderungen 3](#_Toc34910866)

[Erfüllt 4](#_Toc34910867)

[Teilweise implementiert 4](#_Toc34910868)

[Probleme 4](#_Toc34910869)

[Programmierrichtlinien 4](#_Toc34910870)

[Namensgebung 4](#_Toc34910871)

[Deklarationen 4](#_Toc34910872)

[Kommentare 4](#_Toc34910873)

[Statements 4](#_Toc34910874)

[Management Summary 5](#_Toc34910875)

[Mockups 5](#_Toc34910876)

[Use Cases 6](#_Toc34910877)

[Anforderung 001 7](#_Toc34910878)

[Anforderung 002 7](#_Toc34910879)

[Anforderung 003 7](#_Toc34910880)

[Aktivitätendiagramme 7](#_Toc34910881)

[Anforderung 001 7](#_Toc34910882)

[Anforderung 002 8](#_Toc34910883)

[Anforderung 003 8](#_Toc34910884)

[Testing 8](#_Toc34910885)

[Testfall 1: Stationen suchen 8](#_Toc34910886)

[Testfall 2: Gefundene Verbindungen anzeigen 9](#_Toc34910887)

[Testfall 3: Verbindungen ab Station anzeigen 9](#_Toc34910888)

[Testfall 4: Stationen auf Karte anzeigen 9](#_Toc34910889)

[Testfall 5: Mail verschicken 10](#_Toc34910890)

[Ergebnis der Tests 10](#_Toc34910891)

[Schlusswort 10](#_Toc34910892)

# Einleitung

Im Rahmen des ÜK 318 mussten wir eine Fahrplan Applikation auf Basis der zur Verfügung gestellten Bibliothek erstellen. Die Bibliothek greift auf die Daten der «Swiss Public Transport API» von OpenData zu. Das Ziel des Projektes war es, die Kenntnisse in der OOP zu verbessern und einen Einblick in die Entwicklung einer Applikation (mit Planung, etc.) zu erhalten. Im Fokus stand auch die Verwendung der oben erwähnten Web-API und wie man damit in einem Windows Forms Programm umgeht.

# Zweck

Der Zweck dieses Dokuments ist es, das von mir entwickelte Programm «Swiss Transport App» zu Erklären und die Entstehung zu dokumentieren. Auf den nachfolgenden Seiten werden Sie mein Programm näher kennenlernen und die nötigen Informationen für die Bedienung des Programms erhalten.

# Aufgabenstellung

Man soll eine Anwendung entwickeln, die die Verkehrsverbindungen von einer Station zu einer anderen Station anzeigt. Dazu gehören verschiedene 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 | Teilweise |
| A008 | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. | 3 | Ja |

## Erfüllt

Erfüllt wurden die Anforderungen A001-A006 und A008. A007 konnte ich leider nicht ganz lösen.

## 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() und braucht ein 3rd Party API (api.ipify.org) und GetLocationFromIP() braucht ip-api.com.

## Probleme

Es gibt insgesamt zwei Probleme.

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

Wenn man ein Mail schreiben möchte kann es sein, dass das nicht funktioniert, da der Google account das Programm als unsicher

# Programmierrichtlinien

## Namensgebung

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.

## Deklarationen

Constructor, Properties und Methods.

## Kommentare

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

1 Space zwischen // und Anfang von Comment.

Comment mit « . » beenden.

## Statements

Falls man ein bei einem Statement mehr als eine Zeile Code braucht { } brauchen.

# Management Summary

Im Rahmen des ÜK 318 mussten wir eine Fahrplan Applikation auf Basis der zur Verfügung gestellten Bibliothek erstellen. Die Bibliothek greift auf die Daten der «Swiss Public Transport API» von OpenData zu. Das Ziel des Projektes war es, die Kenntnisse in der OOP zu verbessern und einen Einblick in die Entwicklung einer Applikation (mit Planung, etc.) zu erhalten. Im Fokus stand auch die Verwendung der oben erwähnten Web-API und wie man damit in einem Windows Forms Programm umgeht.

# Mockups

Bevor ich mit dem Erstellen des Programms gestartet habe, habe ich Mockups für die Oberfläche (UI) des Programmes erstellt. Damit sollte mir die Arbeit vereinfacht werden.Die Mockups wurden jeweils mit Balsamiq Cloud erstellt.

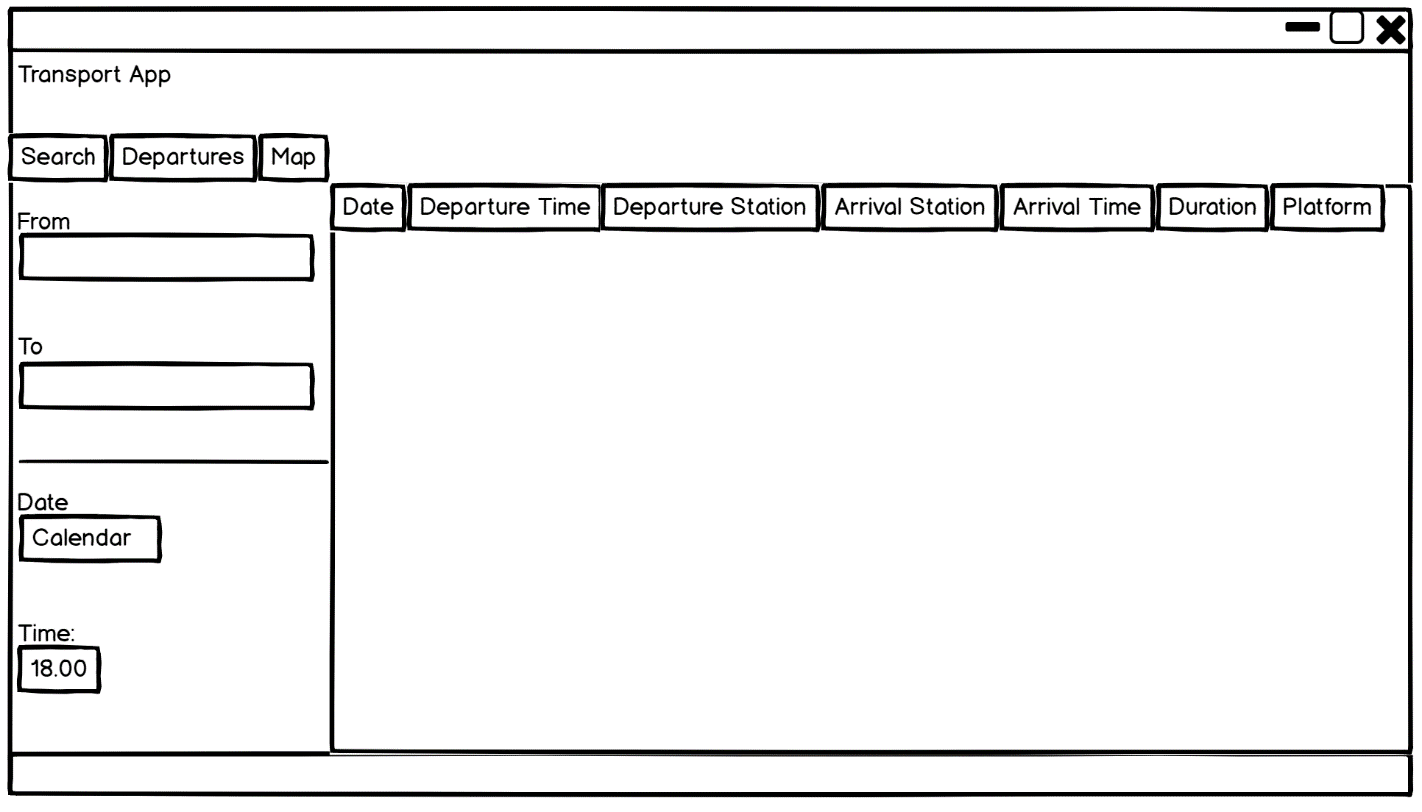


Abb. 1: Mockup1

**Hauptseite**

Auf der Hauptseite kann man Verbindungen Suchen. Durch eine Eingabe einer Startstation, Zielstation soll einem die verschiedenen Verbindungen gezeigt werden. Nach Wunsch kann das Datum und die Zeit angepasst werden. Der Benutzer sollte mit der Tabulator- und der Entertaste arbeiten können.

Es soll die Abfahrtsstation, Zielstation, Abfahrt- und Ankunftszeiten, Gleis und Reisedauer angezeigt werden.

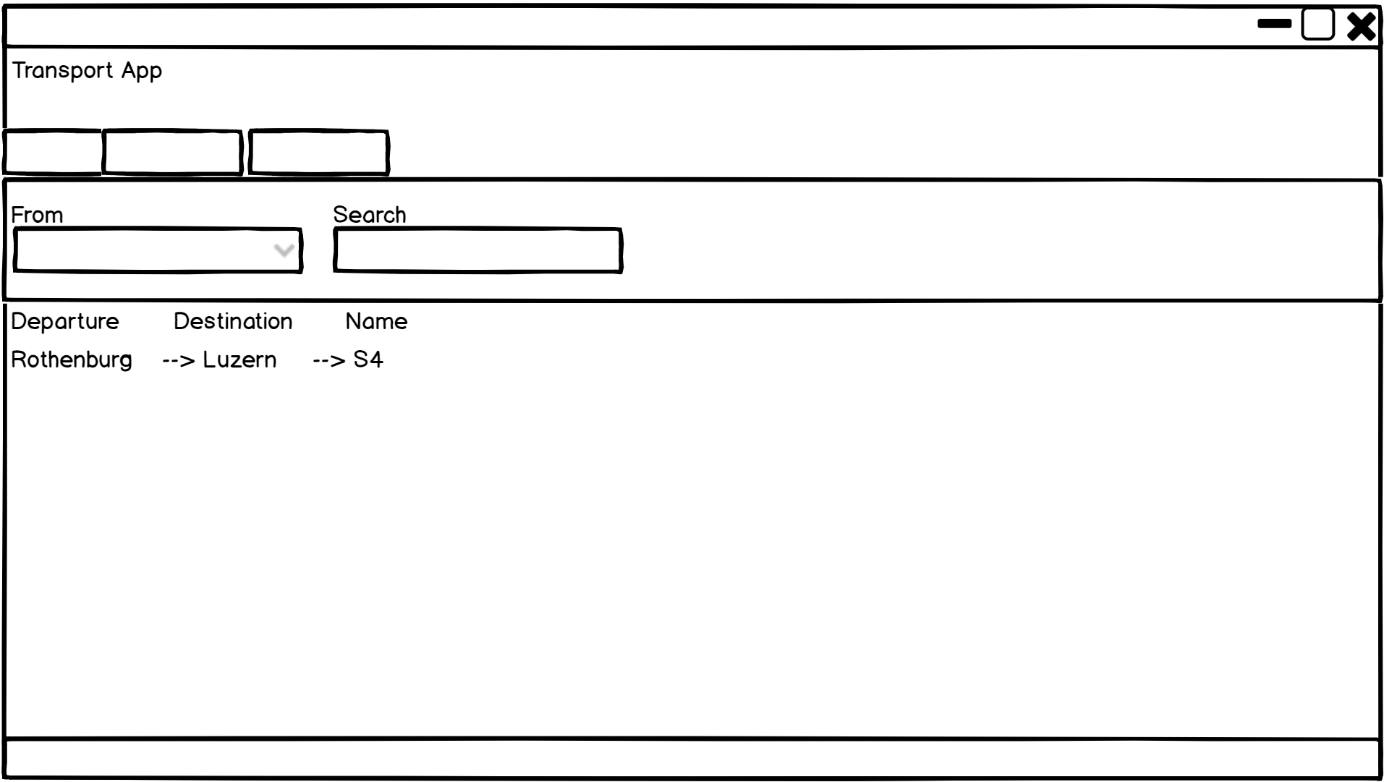


Abb. 2: Mockup2

**Departure Board**

Hier sollte Man eine Station eingeben können und die nächsten Verbindungen ab der Station sehen.

Es soll die Abfahrtsstation, Ziel und Name bzw. Zug-/ Buslinie angegeben werden.

# Use Cases

Um das Programmieren zu vereinfachen, habe ich ein Use Case Diagramm erstellt. Zum Erstellen habe ich Draw.io gebraucht. In meinem Use Case Diagramm habe ich die Aufgaben A001 bis A006 aufgezeichnet.

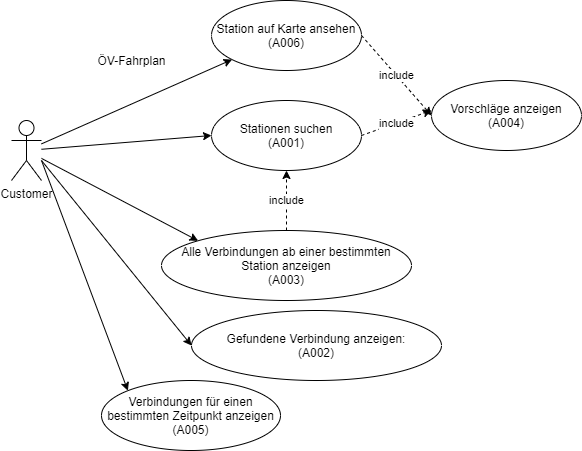


Abb. 3: Use Case Diagram

# Anforderung 001

|  |  |
| --- | --- |
| Use Case A001 | Textergänzung bei der Eingabe von Stationsnamen |
| Beschreibung | Kunde möchte Textvorschläge bekommen, damit er die Stationsnamen nicht auswendig lernen muss. |
| Akteur(e) | Der Benutzer der ÖV-Fahrplan Applikation. |
| Auslöser | Kunde fängt an zu tippen |
| Vorbedingung | 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 |

# Anforderung 002

|  |  |
| --- | --- |
| Use Case A002 | Verbindungsmöglichkeiten zwischen A und B anzeigen |
| Beschreibung | Der Benutzer möchte 4-5 Verbindungen von der Start- zur Endstation angezeigt bekommen |
| Akteur(e) | Der Benutzer der ÖV-Fahrplan Applikation. |
| Auslöser | Kunde gibt Ziel- und Ausgangsstationen ein und Sucht |
| Vorbedingung | Der Kunde muss eine Internetverbindung haben und der REST-Dienst von SBB muss online sein. |
| Ergebnis | In der ListView werden die nächsten 5 Verbindungen ab dieser Station angezeigt. |

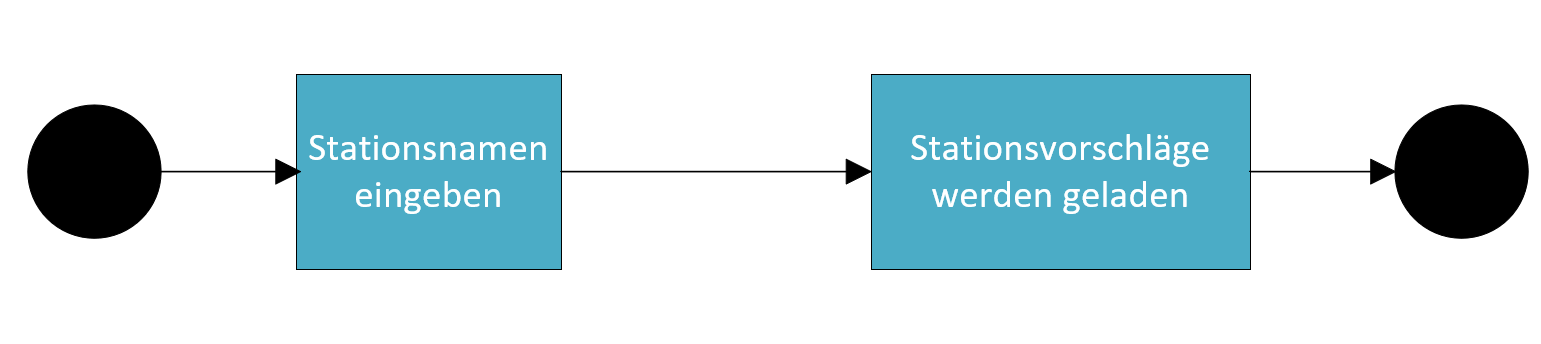
# Anforderung 003

|  |  |
| --- | --- |
| Use Case A003 | Abfahrtstafel anzeigen |
| Beschreibung | Der Benutzer sucht nach einer beliebigen Station und alle Verbindungen ab dieser Station werden angezeigt. |
| Akteur(e) | Der Benutzer der ÖV-Fahrplan Applikation. |
| Auslöser | Kunde klickt auf Connections from Home |
| Vorbedingung | 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 allen Verbindungen wird angezeigt |

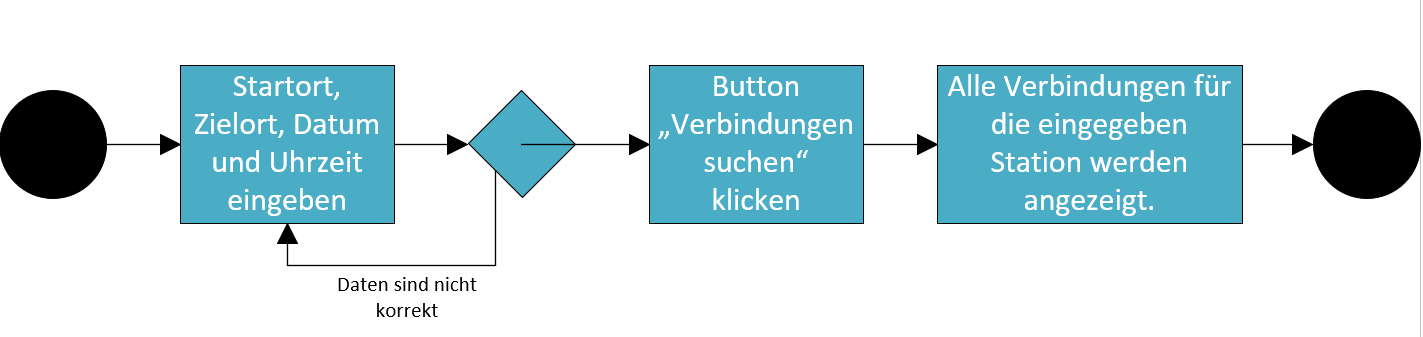
# Aktivitätendiagramme

## Anforderung 001

Der Benutzer möchte nach Stationen suchen und es sollen Ergebnisse dafür angezeigt werden.

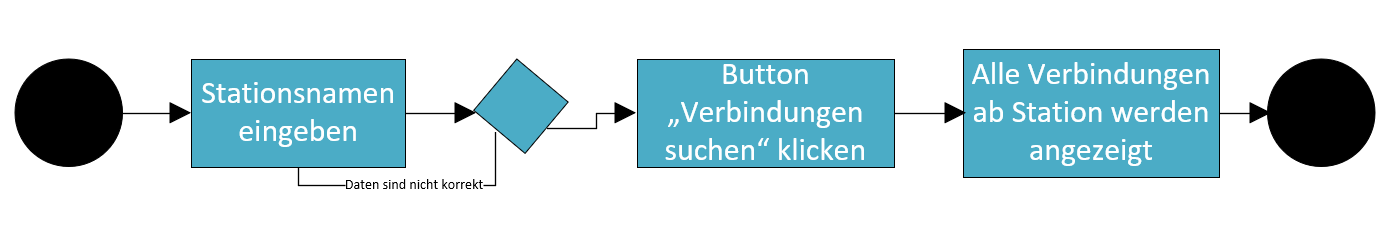


## Anforderung 002

Der Benutzer möchte 4-5 Verbindungen vom Start 

## Anforderung 003

Der Benutzer sucht nach einer beliebigen Station und alle Verbindungen ab dieser Station werden angezeigt.



# Testing

Damit das Programm möglichst keine Fehler hat, sollte es getestet werden. Dafür werden verschiedene Testfälle definiert und anschliessend ausgeführt. Es ist wichtig, dass für jeden Testfall das erwartete Resultat dasselbe ist, wie das tatsächliche Resultat. Das Testing sollte nicht erst am Ende gemacht werden, sondern immer wieder während dem Schreiben des Codes.

## Testfall 1: Stationen suchen

Durchgeführt von: Daniel Polgar

Durchgeführt am: 12.03.2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Tatsächliches Resultat** | **Status** |
| 1 | Benutzer beginnt den Namen der Station einzugeben. | Es werden Stationsvorschläge angezeigt, die zu den eingegebenen Buchstaben passen. | Passende Stationsvorschläge werden angezeigt. | Checkmark |
| 2 | Bei jedem weiteren Eingegebenen Buchstaben wird die Auswahl an Vorschlägen automatisch aktualisiert. | Automatische Aktualisierung der Auswahl | Die Auswahl wird automatisch aktualisiert beim Eingeben von weiteren Buchstaben. | Checkmark |
| 3 | Gewünschte Station wird aus dem DropDown Menu ausgewählt und erscheint in der TextBox. | Station erscheint in der TextBox | TextBox wird mit dem Stationsnamen befüllt. | Checkmark |
| 4 | DropDown verschwindet, nachdem die Station ausgewählt wurde. | DropDown verschwindet. | DropDown verschwindet | Checkmark |

## Testfall 2: Gefundene Verbindungen anzeigen

Durchgeführt von: Daniel Polgar

Durchgeführt am: 12.03.2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Tatsächliches Resultat** | **Status** |
| 1 | Benutzer gibt Abfahrtort und Zielort ein | Auswahl von Stationen wird angezeigt. | Stationsauswahl wird angezeigt. | Checkmark |
| 2 | Benutzer gibt Datum und Uhrzeit ein | Datum und Uhrzeit wird in TextBox angezeigt. | Datum und Uhrzeit wird in TextBox angezeigt. | Checkmark |
| 3 | Benutzer klickt auf Verbindungen suchen | 4 Verbindungen mit den eingegebenen Angaben werden angezeigt. | 4 Verbindungen werden angezeigt. | Checkmark |

## Testfall 3: Verbindungen ab Station anzeigen

Durchgeführt von: Rouven Wicki

Durchgeführt am: 12.03.2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Tatsächliches Resultat** | **Status** |
| 1 | Benutzer gibt Namen der Station ein | Auswahl von Stationen wird angezeigt. | Stationsauswahl wird angezeigt. | Checkmark |
| 2 | Benutzer klickt auf Button «Search» | Verbindungen ab gewünschter Station werden angezeigt. | Verbindungen werden angezeigt. | Checkmark |

## Testfall 4: Stationen auf Karte anzeigen

Durchgeführt von: Daniel Polgar

Durchgeführt am: 12.03.2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Tatsächliches Resultat** | **Status** |
| 1 | In der Combobox gibt der Benutzer die gewünschte Station ein | Auswahl an Station, die ähnlich wie die Eingabe sind, werden angezeigt. | Auswahl an Station, die ähnlich wie die Eingabe sind, werden angezeigt. | Checkmark |
| 2 | Der Benutzer wählt die Station aus | Die gewünschte Station wird auf der Karte angezeigt. | Die gewünschte Station wird angezeigt. | Checkmark |

## Testfall 5: Mail verschicken

Durchgeführt von: Rouven Wicki

Durchgeführt am: 12.03.2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Tatsächliches Resultat** | **Status** |
| 1 | Benutzer klickt auf den Button «Create E-Mail» | Ein neues Fenster öffnet sich. | Ein neues Fenster hat sich geöffnet. | Checkmark |
| 2 | Benutzer gibt seine Daten ein (Bei Email ID und Passwort seine Gmail Angaben). | Der Benutzer kann seine Daten eingeben. | Der Benutzer konnte seine Daten eingeben. | Checkmark |
| 3 | Der Benutzer drückt auf den Button «send Email» | Das Email soll abgeschickt werden und eine Nachricht «Message sent» soll aufgezeigt werden. | Das Email wurde abgeschickt und eine Nachricht «Message sent» wurde aufgezeigt. | Checkmark |

## Ergebnis der Tests

Alle Tests waren erfolgreich. Alles hat so funktioniert wie ich es erwartet habe und es gab keine Fehler. Ein Grund dafür war auch, dass ich während dem Schreiben des Codes immer wieder die Funktionen getestet habe. Dadurch konnte ich die Fehler immer sofort beheben oder ein Workaround finden.

## Schlusswort

Das Projekt hat mir sehr viel Spass gemacht.