

Dokumentation ÖV-Fahrplan

The screenshot shows the 'Transport App' window. At the top right, the time is 08:43:40. Below the title bar, there are four tabs: 'Search for a Connection' (selected), 'Departure Board', 'Station Map', and 'Stations Near Me'. On the left side, under 'Search for a Connection', there are two dropdown menus for 'Connection From:' and 'Connection To:', followed by a 'Search' button. Below this is a 'Detailed Options' section with a 'Date:' dropdown set to 'Wednesday, March 1' and a 'Time:' dropdown set to '08:43'. The main area on the right contains a table with the following headers: 'Date', 'Departure ...', 'Departure Station', 'Arrival Station', 'Arrival Time', 'Duration', and 'Platform'.

Date	Departure ...	Departure Station	Arrival Station	Arrival Time	Duration	Platform
------	---------------	-------------------	-----------------	--------------	----------	----------

Rouven Wicki

ÜK M318

12.03.2020

Contents

Einleitung.....	4
Zweck.....	4
Aufgabenstellung.....	4
Anforderungen	4
Erfüllt	5
Teilweise implementiert.....	5
Probleme	5
Programmierrichtlinien	5
Namensgebung.....	5
Deklarationen	5
Kommentare.....	5
Statements	5
Management Summary.....	6
E-Mail.....	6
Mockups	6
Use Cases.....	7
Anforderung 001	8
Anforderung 002	8
Anforderung 003	8
Aktivitätendiagramme.....	8
Anforderung 001	8
Anforderung 002	9
Anforderung 003	9
Testing	9
Testfall 1: Stationen suchen	9
Testfall 2: Gefundene Verbindungen anzeigen	10
Testfall 3: Verbindungen ab Station anzeigen.....	10
Testfall 4: Stationen auf Karte anzeigen.....	10
Testfall 5: Mail verschicken	11
Ergebnis der Tests	11
Installationsanleitung	12
Deinstallationsanleitung.....	12
Schlusswort	12

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 markiert. Das kann man aber einfach in den Einstellungen des Google Accounts ausstellen.

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.

E-Mail

Um ein E-Mail schreiben zu können wird ein Gmail Konto gebraucht. Man muss seine gmail Adresse eingeben und das Passwort des Google Kontos.

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.

The mockup shows a window titled "Transport App" with standard Windows window controls (minimize, maximize, close). Below the title bar, there are three buttons: "Search", "Departures", and "Map". The main area is divided into two sections. On the left, there are input fields for "From" and "To", a "Date" field with a "Calendar" button, and a "Time:" field with a value of "18.00". On the right, there is a table with the following headers: "Date", "Departure Time", "Departure Station", "Arrival Station", "Arrival Time", "Duration", and "Platform". The table body is currently empty.

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.

The mockup shows a window titled "Transport App" with standard window controls (minimize, maximize, close). Below the title bar are three empty rectangular input fields. The main content area is divided into two sections. The top section has a "From" label followed by a dropdown menu showing a downward arrow, and a "Search" label followed by a text input field. The bottom section contains a table with three columns: "Departure", "Destination", and "Name". The first row of the table has the values "Rothenburg", "--> Luzern", and "--> S4".

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.

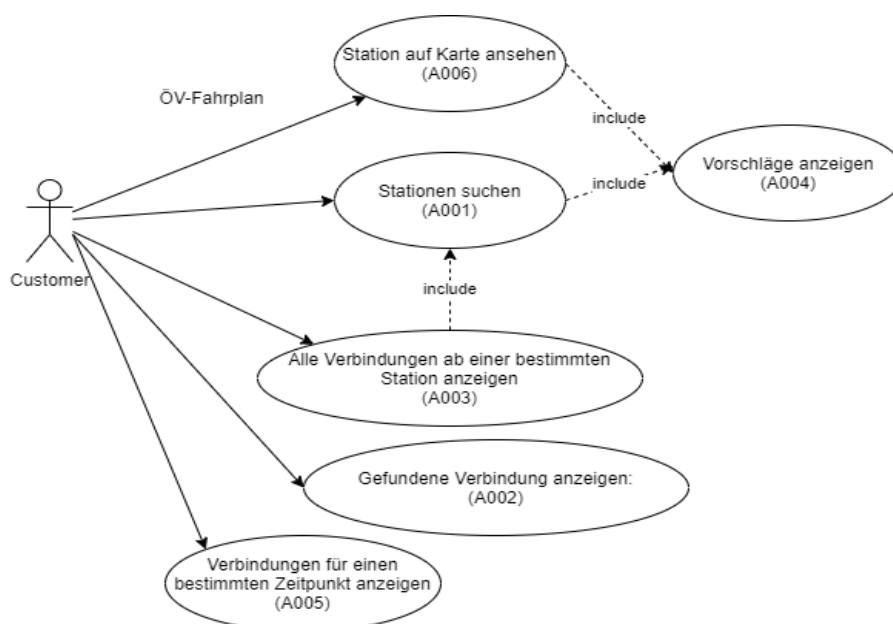


Abb. 3: Use Case Diagram

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.

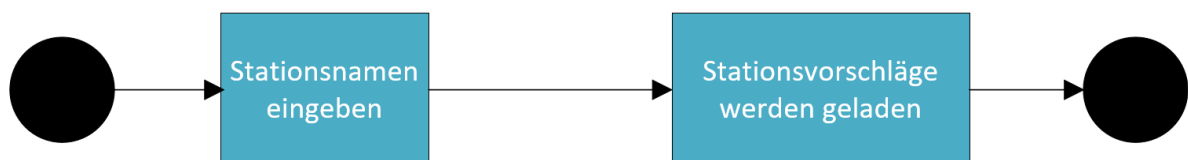
Anforderung 003

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

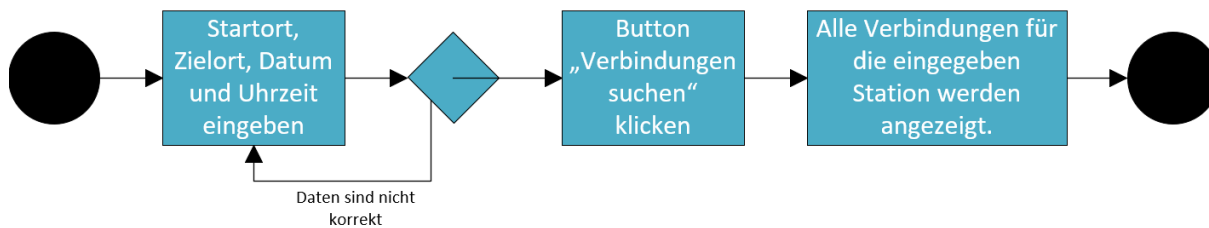
Anforderung 001

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



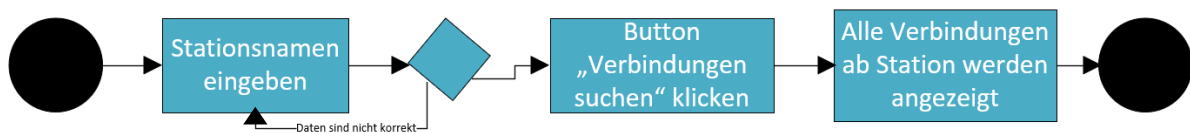
Anforderung 002

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.	✓
2	Bei jedem weiteren Eingeggebenen Buchstaben wird die Auswahl an Vorschlägen automatisch aktualisiert.	Automatische Aktualisierung der Auswahl	Die Auswahl wird automatisch aktualisiert beim Eingeben von weiteren Buchstaben.	✓
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.	✓

4	DropDown verschwindet, nachdem die Station ausgewählt wurde.	DropDown verschwindet.	DropDown verschwindet	✓
---	--	------------------------	-----------------------	---

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.	✓
2	Benutzer gibt Datum und Uhrzeit ein	Datum und Uhrzeit wird in TextBox angezeigt.	Datum und Uhrzeit wird in TextBox angezeigt.	✓
3	Benutzer klickt auf Verbindungen suchen	4 Verbindungen mit den eingegebenen Angaben werden angezeigt.	4 Verbindungen werden angezeigt.	✓

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.	✓
2	Benutzer klickt auf Button «Search»	Verbindungen ab gewünschter Station werden angezeigt.	Verbindungen werden angezeigt.	✓

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.	✓
2	Der Benutzer wählt die Station aus	Die gewünschte Station wird auf der Karte angezeigt.	Die gewünschte Station wird angezeigt.	✓

Testfall 5: Mail verschicken

Durchgeführt von: Rouven Wicki

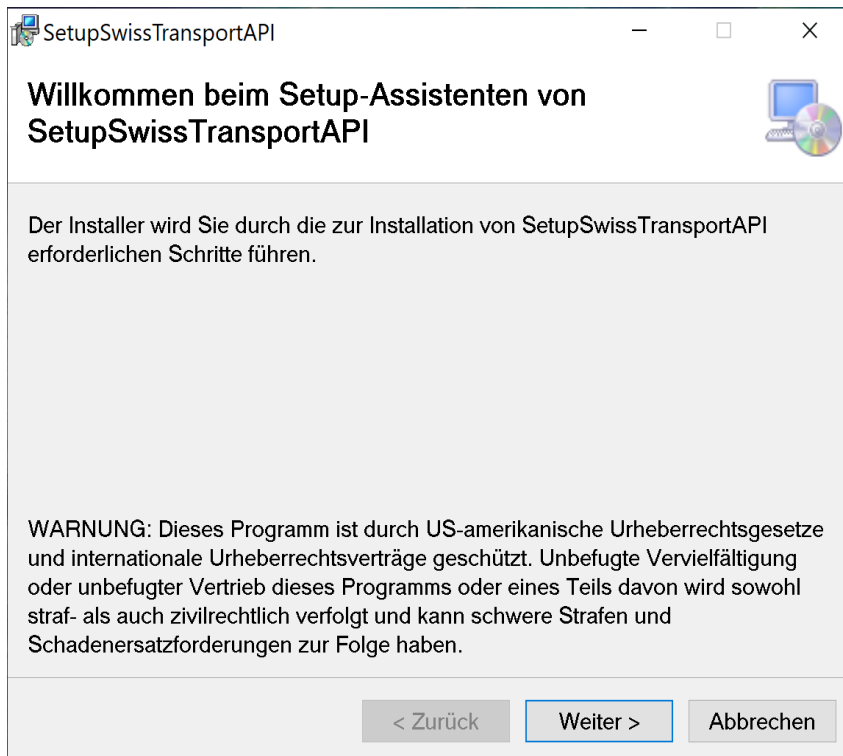
Durchgeführt am: 12.03.2020

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.	✓
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.	✓
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.	✓

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.

Installationsanleitung



Um das Programm «SwissTransport App» zu installieren, öffnen Sie die Datei «SetupSwissTransportAPI.msi».

Dann öffnet ein Fenster. Hier klicken sie auf weiter.

Im nächsten Schritt können Sie den Installationspfad wählen und dann auf weiter klicken.

Wenn Sie im nächsten Fenster auf weiter klicken, wird das Programm installiert und es kann benutzt werden.

Deinstallationsanleitung

Zum Deinstallieren startet man das Setup. Dann wählt man Deinstallieren. Danach wird das Programm deinstalliert.

Schlusswort

Das Projekt hat mir sehr viel Spass gemacht. Ich konnte meine Fähigkeiten in C# verbessern und viele neue Dinge lernen. Leider konnte ich A008 nicht vollständig lösen. Es war das erste Mal, dass ich mit einer API gearbeitet habe was ich sehr spannend fand. Mir hat sehr gefallen, dass wir mit Github gearbeitet haben.