Yago Romero Terzis 03/05/2019

Swisstransport

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc7780978)

[1.1 Zweck des Dokuments 2](#_Toc7780979)

[1.2 Was wurde umgesetzt? 2](#_Toc7780980)

[1.3 Anforderungen 2](#_Toc7780981)

[2 Funktionalität 3](#_Toc7780982)

[2.1 Use Case 3](#_Toc7780983)

[2.2 Design 3](#_Toc7780984)

[2.2.1 Mockups 3](#_Toc7780985)

[2.2.2 GUI Design 4](#_Toc7780986)

[2.3 Endgültiges Design 5](#_Toc7780987)

[2.4 Diagramme zur Umsetzung des Projekts 7](#_Toc7780988)

[2.4.1 UML Diagramm 7](#_Toc7780989)

[2.4 .2 Flussdiagramm 8](#_Toc7780990)

[3 Tests 8](#_Toc7780991)

[3.1 Bugs 9](#_Toc7780992)

[4 Installation 9](#_Toc7780993)

[5 Bewertung 10](#_Toc7780994)

# Einleitung

Während dieses ÜK, habe ich das Projekt SwissTransport vollendet, welches zur Vertiefung der OOP dienen soll.

## Zweck des Dokuments

Dieses Dokument soll als Gebrauchsanleitung und als Bewertung des Projekts dienen. Dafür werde ich die Funktionalitäten des Programms genau beschreiben.

## Was wurde umgesetzt?

Umgesetzt wurde die API von Transport, um eine Desktop Anwendung zu erzeugen, die den Nutzer die Suche nach ÖV Verbindungen ermöglicht. Dies wird anhand der eingegebenen Stationen Namen, die später mit den Ergebnissen einer Datenbankabfrage verglichen werden, ermöglicht.

## Anforderungen

Folgende Anforderungen werden bei der Herstellung dieser Anwendung berücksichtigt:

* Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. Priorität 1
* 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. Priorität 1
* Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann. Priorität 1
* Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such Resultate erscheinen, damit ich effizienter nach Stationen suchen kann. Priorität 2
* 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. Priorität 2
* Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. Priorität 3
* Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. Priorität 3
* Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann.

# Funktionalität

## Use Case

Mit diesem Programm soll der Benutzer einfach und bequem alle mögliche ÖV Verbindungen erhalten, die er für seinen Alltag braucht.

Ein Bild, das Text, Karte enthält.

Automatisch generierte Beschreibung

Abbildung 1 Use Case Diagramm

## Design

### Mockups

Als erstes design habe ich folgende Mockups entworfen:

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 2 Mockup für gesamt Funktionalität des Programms

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 3 Mockup bei Eintrag von einer einzelnen Station

### 2.2.2 GUI Design

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 4 Erstes GUI design

Ich habe die Mockups als Entwurfs mustern benutzt und aus ihnen ein fertiges GUI erstellt. Dieses GUI implementiert die Autocomplete Funktion, sowie die Funktion der Suche durch einzelne Buchstaben.

Ich habe auch ein DateTime picker implementiert, damit der Benutzer einen beliebigen Tag oder eine Uhrzeit wählen kann, an welche er gerne fahren mochte.

In der Mitte des Programms werden die Verbindungen angezeigt, die die Kriterien des Nutzers erfühlen. Sie werden in ein ListView angezeigt.

Mit dem Link «Mehr Information», kann der Benutzer auf weitere Funktionalitäten des Programms zugreifen.

Nachdem ich mit dem ersten GUI fertig war, habe ich ein eleganteres GUI erstellt, der den Gebrauch des Programms eleganter und einfacher machen soll.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 5 Zweites GUI Design

## 2.3 Endgültiges Design

Am Ende habe ich mich für eine Version des Zweites GUI Design entschieden:

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 6 GUI Fertiges design

In dieser Version wurde der Knopf «Tauchen» eingefügt, der die Von- und Nach Stationen tauschen lässt, falls der Benutzer wiesen möchtet, wenn die Rückfahrt fährt.

Wenn der Benutzer auf den Knopf «löschen» klickt, dann löschen sich die Inhalte der Suchfelder. Wenn er auf den Knopf «Ort anzeigen» drückt, wird er zu Google Maps weitergeleitet und wenn er auf «Email senden» drückt, wird er zu seinen Standard E-mail Provider weitergeleitet.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Ich habe auch eine Abfahrtstafel gemacht, die durch den Knopf «Abfahrtstafel» erscheint. Dort kann der Nutzer alle Verbindungen, die ab einer bestimmten Station fahren, nachschauen.

## 2.4 Diagramme zur Umsetzung des Projekts

### 2.4.1 UML Diagramm

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 7 Klassendiagramm Projekt

### 2.4 .2 Flussdiagramm

Ein Bild, das Text, Karte enthält.

Automatisch generierte Beschreibung

Abbildung 8 Flussdiagramm Eingabe von Stationen

# Tests

Anforderung

Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.

Wenn der Benutzer anfängt in einer der Beiden Eingabefeldern zu schreiben, dann soll das darunterliegende Feld dieses Wort suchen und alle möglichen Resultate anzeigen, die die Buchstaben dieses Worts enthalten. Am Schluss soll der Benutzer wählen können, welche Station er gemeint hat.

Anforderung

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.

Wenn der User auf den Suchen Knopf drückt, sollen mindestens 4 Verbringungen mit die eingegebenen Von- und Nachstationen erscheinen.

Anforderung

Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann.

Wenn der Nutzer auf Abfahrtstafel drückt, soll ein neues Fenster geöffnet werden, welches oben die bestimmte Station enthält und darunter alle Verbindungen ab diese Station.

Anforderung

Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können

Wenn der Benutzer auf dem Email Knopf drückt, soll er an der Webseite ihres Email Provider weitergeleitet werden. Dort kann er ein Email schreiben.

## Bugs

Es kann sein, dass die API überlastet werden wird und dadurch wird das Programm langsamer oder funktioniert nicht.

Beim klicken von «Email senden» Knopf, ohne eine Verbindung zu wählen, stürzt das Programm ab.

# Installation

Damit der Benutzer dieses Programm in seiner Maschine installieren kann, habe ich einen Installationsprojekt gemach, welcher InstallerSetup heisst. Der Nutzer muss nur die Anweisungen des Installerst folgen.

Bsp.:



Abbildung 9 Icon vom installer

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 10 Setup Wizzard

# Bewertung

Ich glaube, dass dieses Projekt gelungen ist, da alle Anforderungen erfüllt wurden, Fehlern wurden minimiert, der Code wurde sauber gehalten und das GUI ein attraktives design hat.