**Dokumentation**

Swiss Transport

|  |  |
| --- | --- |
| Inhalt | **Dokumentation Swiss Transport** |
| Erstelldatum | **Dienstag, 26. November 2019** |
| Änderungsdatum | **Dienstag, 3. Dezember 2019** |
| Autor | **Janis Kneubühler** |

**Inhaltsverzeichnis**

[1 Einleitung 3](#_Toc26356957)

[1.1.1 Worum geht es? 3](#_Toc26356958)

[1.1.2 Warum muss eine Entscheidung getroffen werden? 3](#_Toc26356959)

[1.1.3 Welche Ziele werden verfolgt? 3](#_Toc26356960)

[1.1.4 Welche Optionen gibt es? 4](#_Toc26356961)

[1.1.5 Welche Folgen haben die Optionen? 4](#_Toc26356962)

[1.1.6 Welche Option wird empfohlen? 4](#_Toc26356963)

[2 Zweck des Dokumentes 4](#_Toc26356964)

[3 Programmierrichtlinien 5](#_Toc26356965)

[3.1 Naming 5](#_Toc26356966)

[3.2 Declaration 5](#_Toc26356967)

[3.3 Comments 5](#_Toc26356968)

[3.4 GUI-Standards 5](#_Toc26356969)

[4 Funktionen des Programmes 5](#_Toc26356970)

[4.1 Vollständig umgesetzt 5](#_Toc26356971)

[4.2 Teilweise umgesetzt 5](#_Toc26356972)

[4.3 Nicht umgesetzt 5](#_Toc26356973)

[4.4 Bugs 5](#_Toc26356974)

[5 Mockups 5](#_Toc26356975)

[6 Use-Cases 5](#_Toc26356976)

[7 Systemtests 6](#_Toc26356977)

[8 Testprotokoll 6](#_Toc26356978)

[9 Installationsanleitung 6](#_Toc26356979)

Es wird keine Dokumentation im eigentlichen Sinne verlangt. Es braucht jedoch im Minimum ein Dokument,

das die folgenden Informationen enthält.

OK - Autor, Datum.

- Eine Einleitung (Management Summary).

- Zweck des Dokuments.

- Was (d.h. welche Funktionen) wurde umgesetzt?

- Falls bestimmte Funktionen nur teilweise umgesetzt wurden: Welche? Welcher Teil der

Funktionalität fehlt noch? Bekannte Fehler/Bugs?

- Mockups für alle Formulare/Dialoge (GUI) , die benötigt werden um die Anforderungen A001, A002

und A003 abzudecken.

- Use Cases, Use Case Beschreibungen und Aktivitätendiagramme mindestens für die mit Priorität 1

klassifizierten Anforderungen.

- Testfälle (Systemtests), verständlich und eindeutig nachvollziehbar geschrieben, so dass ein Tester

diese ohne weiteren Erklärungen durchführen kann.

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

tatsächlichen Testresultate.

- Installationsanleitung: Wie wird die Software installiert? Wie wird die Software deinstalliert?

- Andere spannende Informationen für die Bewertung.

Die Dokumentation soll sauber gestaltet, inhaltlich korrekt und übersichtlich strukturiert sein. Ein

Inhaltsverzeichnis soll vorhanden sein. Achte auf Typographie und Rechtschreibung!

# Einleitung

In dieser Arbeit wird ein Programm erstellt, welches ÖV-Verbindungen anzeigen soll. Diese Verbindungen werden mithilfe der vom ÜK zur Verfügung gestellten API gesucht. Der Kunde hat verschiedene Anforderungen, die das Programm haben soll, aufgeschrieben und priorisiert. Diese sind im Punkt 1.1 aufgeführt.  
Diese Arbeit mitsamt den Anforderungen wurde vom ÜK vorgegeben. Welche Anforderungen eingebaut werden, ist jedem selber überlassen. Jedoch sollten diese mit der Priorität 1 sicher vorhanden sein! Zudem können aber auch noch eigene Ideen eingebaut werden.  
Bis am Dienstag, 3. Dezember, möchte ich sicher mit den Anforderungen, mit der Priorität 1, fertig sein. Danach geht es an die Verschönerung des GUIS, an das Testen des Programms und ans Schreiben der Dokumentation. Erst dann werden zusätzliche Funktionen ins Programm eingebaut.  
Am Schluss möchte ich ein Programm ohne Bugs haben, falls nicht alle Funktionen eingebaut sind, ist dies kein Problem. Mir ist es wichtiger, dass die Qualität des Codes stimmt und dafür das Programm nicht ganz so umfangsreich ist.

## Anforderungen des Kunden

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

## Zweck des Dokumentes

Dieses Dokument zeigt, was im ÜK 318 alles gemacht wurde. Es soll einerseits zeigen, welche Funktionen und Fehler die Swiss Transport App. Aber auch Programmierrichtlinien, Test-Cases und die ganze Planung befinden sich in diesem Dokument.

# Programmierrichtlinien

## Naming

|  |  |  |
| --- | --- | --- |
| Stelle | REgel | Beispiel |
| Allgemein | Namen auf Englisch, aussagen welchen Zweck sie haben,Camel Case |  |
| Variablen | Anfang klein, lokale und Membervariablen werden nicht unterschieden | numberOfConnections |
| Methoden | Anfang gross | AreStationFieldsFilled() |
| Klassen | Gross | Station |
| GUI | Vor Name Kürzel des GUI-Elements (Angegeben in Modul 305 Dokument) | btnSearchConnections |

## Declaration

|  |  |
| --- | --- |
| REgel | Beispiel |
| Variablen: am Anfang deklarieren und im Konstruktor definieren | int numberOfConnections;  Konstruktor() {  numberOfConnections = 0  } |

## Comments

|  |  |
| --- | --- |
| REgel | Beispiel |
| Code soll durch richtiges naming selbsterklärend sein -> keine comments im code, ausser bei sehr komplizierten methoden | int numberOfConnections;  Konstruktor() {   numberOfConnections = 0  } |

# Use-Cases

# Mockups

# Funktionen des Programmes

## Vollständig umgesetzt

## Teilweise umgesetzt

## Nicht umgesetzt

## Bugs

# Systemtests

# Testprotokoll

# Installationsanleitung

Installation vom GIT auch?