Dokumentation M318

12.03.2020

Durrer Nick

**Inhaltsverzeichnis**

[2 Einleitung 2](#_Toc34479431)

[2.1 Aufgabenstellung 2](#_Toc34479432)

[2.2 Anforderungen 2](#_Toc34479433)

[2.3 Zweck des Dokumentes 2](#_Toc34479434)

[3 Programmrichtlinien 2](#_Toc34479435)

[3.1 Naming 2](#_Toc34479436)

[3.2 Declaration 2](#_Toc34479437)

[3.3 Comments 2](#_Toc34479438)

[4 Planung 2](#_Toc34479439)

[4.1 Use cases 2](#_Toc34479440)

[4.2 Aktiviäts Diagramm 2](#_Toc34479441)

[4.3 Mockups 2](#_Toc34479442)

[5 Funktionen des Programmes 2](#_Toc34479443)

[5.1 Vollständig umgesetzt 2](#_Toc34479444)

[6 Bugs und Fehler 2](#_Toc34479445)

[7 Testing 2](#_Toc34479446)

[7.1 Testfälle 2](#_Toc34479447)

[8 Installationsanleitung 2](#_Toc34479448)

[8.1 Installation 2](#_Toc34479449)

[8.2 Deinstallation 2](#_Toc34479450)

# Einleitung

## Aufgabenstellung

In dieser Arbeit wird eine Desktop Applikation erstellt, welche 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, gestellt und priorisiert. Die Anforderungen sind im Kapitel 2.2 Anforderungen aufgeführt. Es ist zu entscheiden, welche Funktionen man in seinem Programm einbinden möchte. Jedoch sollten die Anforderungen mit Priorität 1 auf jeden Fall im Programm vorhanden sein. Mein Ziel ist es bis Ende des ÜK ein funktionstüchtiges Programm zu besitzen, welches auch die Code Konventionen einhält.

## Anforderungen

A screenshot of a cell phone

Description automatically generated

## Zweck des Dokumentes

Zweck dieses Dokumentes ist es, die praktische Arbeit im Rahmen des ÜK 318 zu dokumentieren. Ausserdem enthält dieses Dokument Planung sowie Testing der praktischen Arbeit und andere wichtige Informationen.

# Programmrichtlinien

## Naming

|  |  |  |
| --- | --- | --- |
| Stelle | Regel | Beispiel |
| Allgemein | Aussagekräftiger Name auf Englisch |  |
| Variabeln | CamelCase, lokale und Membervariabeln werden nicht unterschieden | numberOfConnections |
| MEthoden | Anfang Gross | AreStationFieldsFilled() |
| Klassen | Gross | Station |
| gui | Vor Name Kürzel des GUI Elements | btnSearchConnections |

## Declaration

|  |  |
| --- | --- |
| Regel | Beispiel |
| Globale Variablen unterhalb des Klassenkopfes definieren und im Konstruktor deklarieren | int firstVar;  Konstruktor()  {   firstVar= 0  } |
| Lokale variablen unterhalb des Methodenkopfes definieren und deklarieren | Methode()  {  Int firstVar = 0;  } |

## Comments

|  |  |
| --- | --- |
| Regel | Beispiel |
| Der Code sollte eigentlich selbsterklärend sein. Bei schwierigen Methoden wird kommentiert. | somethingSimple()  //Comment  somethingDiffucult() |

## Statements

|  |  |
| --- | --- |
| Regel | Beispiel |
| Alle geschweiften Klammern werden auf eine neue Zeile platziert | Methode()  {  Int firstVar = 0;  } |
|  |  |

# Planung

## Use cases

A close up of a map

Description automatically generated

|  |  |
| --- | --- |
| Use Case | Start - & End-station suchen |
| Beschreibung | Um die Stationssuche dem User zu erleichtern, soll nicht exakt nach dem richtigen Stationsnamen gesucht werden müssen. Die gewünschte Station soll auch schon gefunden werden, wenn nur ein Teil davon steht. |
| Akteur | ÖV User |
| Auslöser | Wenn der User den Namen einer Station herausfinden möchte und er diesen nicht sich merken muss. |
| Vorbedingung | Internetverbindung |
| Ablauf | 1. User öffnet das Programm 2. User entscheidet ob er Abfahrstaffel oder Anschlussmöglichkeiten anzeigen möchte 3. Der User sucht gibt den Text des Stationsnamen ein. 4. User klickt auf «Suchen». 5. Stationen werden aufgelistet |
| Ergebnis | Die Station wird trotzdem gefunden, auch wenn diese nicht exakt so eingegeben wurde wie sie richtig heisst. Zum Beispiel wenn der Benutzer nach «Wolfenschiessen» sucht, wird «Wolfenschiessen, Bahnhof» angezeigt. |

|  |  |
| --- | --- |
| Use Case | Verbindungen anzeigen |
| Beschreibung | Dem User werden die nächsten fünf ÖV-Verbindungen zwischen eingegebenen Start- und Zielort angezeigt. |
| Akteur | ÖV User |
| Auslöser | Wenn der User die besten Verbindungen zwischen diesen zwei Stationen wissen möchte |
| Vorbedingung | Internetverbindung |
| Ablauf | 1. User öffnet das Programm 2. User klickt auf «Verbindungen anzeigen» 3. User gibt Start- und Zielstation ein 4. User klickt auf «Suchen» 5. User bekommt die nächsten 5 Verbindungen angezeigt |
| Ergebnis | Die nächsten fünf Verbindungen mit Uhrzeiten, welche vom Start- bis zum Zielort führen, werden dem User aufgelistet. |

|  |  |
| --- | --- |
| Use Case | Abfahrtstaffel anzeigen |
| Beschreibung | Dem Benutzer werden alle Anschlussmöglichkeiten einer Station angezeigt, wie man es sich auch bei Abfahrtstafeln an den Bahnhöfen sich gewöhnt ist. Es sollen Verbindungen, egal in welche Richtung, welche an dieser Station anhalten angezeigt werden. |
| Akteur | ÖV User |
| Auslöser | Wenn der User möchte die nächsten Verbindungen vom gewünschten Ort in alle Richtungen wissen |
| Vorbedingung | Internetverbindung |
| Ablauf | 1. User öffnet das Programm 2. User klickt auf «Abfahrtstaffel» anzeigen 3. User gibt Stationsnamen ein 4. User drückt auf «Suchen» 5. Dem User werden alle Verbindungen in alle Richtungen angezeigt |
| Ergebnis | Dem User werden alle Verbindungen vom gewünschten Ort in alle Richtungen angezeigt. |

## Aktiviäts Diagramm

A picture containing screenshot

Description automatically generated

## Mockups

### Start Fenster

A close up of text on a white background

Description automatically generated

### Verbindung suchen

A close up of text on a black background

Description automatically generated

### Station suchen

A screenshot of a cell phone

Description automatically generated

# Funktionen des Programmes

## Vollständig umgesetzt

# Bugs und Fehler

# Testing

## Testfälle

# Installationsanleitung

## Installation

## Deinstallation