Dokumentation

Simon Ruckli

Modul 318 - Analysieren und objektbasiert programmieren

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc34830110)

[2 Zweck des Dokuments 2](#_Toc34830111)

[3 Umgesetzte Funktionen 2](#_Toc34830112)

[4 Mockups 2](#_Toc34830113)

[4.1 Abfahrtstafel 2](#_Toc34830114)

[4.2 Abfahrtsplan 2](#_Toc34830115)

[5 Use Case 2](#_Toc34830116)

[5.1 Abfahrtstafel anzeigen 2](#_Toc34830117)

[5.2 Startstation suchen 2](#_Toc34830118)

[5.3 Endstation suchen 2](#_Toc34830119)

[5.4 Verbindungen suchen 2](#_Toc34830120)

[6 UML-Diagramme 2](#_Toc34830121)

[6.1 Hauptfunktion Abfahrtsplan 2](#_Toc34830122)

[6.2 Hauptfunktion Abfahrtsplan 2](#_Toc34830123)

[6.3 Ortsvorschläge Abfahrtsort 2](#_Toc34830124)

[6.4 Ortsvorschläge Ankunftsort 2](#_Toc34830125)

[7 Programmierrichtlinien 2](#_Toc34830126)

[7.1 Namensgebung von Variablen, Eigenschaften, Methoden: 2](#_Toc34830127)

[7.2 Kommentare: 2](#_Toc34830128)

[7.3 GUI-Controls 2](#_Toc34830129)

# Einleitung

Im ÜK 318 habe ich ein C# Windows Forms Programm erstellt welches mir möglich macht, einen Busfahrplan automatisch zu erzeugen indem ich einen Abfahrtsort und einen Ankunftsort angebe. Zusätzlich habe ich noch eine Abfahrtstafel gemacht auf der alle in einem gewissen Zeitraum enthaltenen folgenden Fahrten von einer Busstation aus.  
Bei diesem Projekt konnte ich einiges neues lernen, jedoch war es nicht so als ob ich überfordert war. Ich erhielt auch immer eine gute Hilfe, falls mir mal eine Frage auflag.

# Zweck des Dokuments

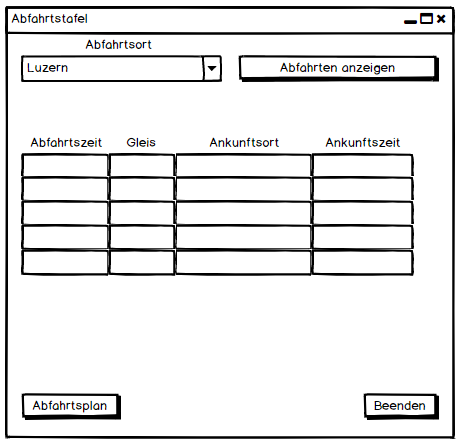
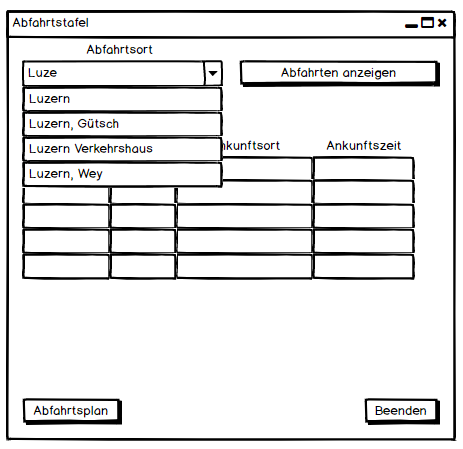
Der Zweck dieser Dokumentation ist, das Programm welches ich im Rahmen dieses Kurses erstellt habe zu dokumentieren. Dieses Dokument ist unter anderem auch ein Notenkriterium.

# Umgesetzte Funktionen

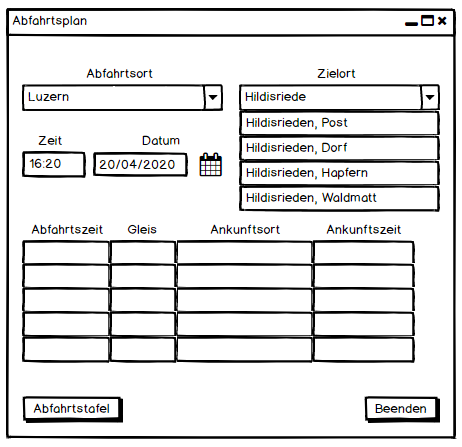
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Beschreibung | Priorität | Status |
| A001 | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | 1 | Abgeschlossen |
| A002 | Als ÖV-Benutzer möchte ich die aktuellen, 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 | Abgeschlossen |
| A003 | Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zu Hause eine Art Abfahrtstafel haben kann. | 1 | Abgeschlossen |

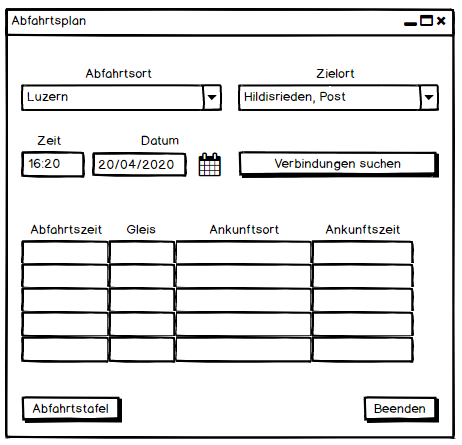
# Mockups

## Abfahrtstafel



## Abfahrtsplan





# Use Case

## Abfahrtstafel anzeigen

|  |  |
| --- | --- |
| Use Case | Abfahrtstafel anzeigen |
| Beschreibung | ÖV Benutzer möchte die Abfahrtstafel sehen, damit er sehen kann, wann welcher Bus / Zug Fahrt auf welchem Gleis. |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss Startstation eingeben |
| Ablauf | 1. Programm öffnen 2. Auf Button Abfahrtsplan klicken 3. Startstation eingeben 4. Auf Button Verbindung suchen klicken |
| Ergebnis | Verbindungen in der nächsten Zeit von dieser Station werden aufgelistet |

## Startstation suchen

|  |  |
| --- | --- |
| Use Case | Startstation suchen |
| Beschreibung | ÖV Benutzer möchte nicht den ganzen Namen der Startstation eingeben, weshalb er auf einen Button klicken kann dem Ihm alle Stationen vorschlägt, die mit seiner Eingabe anfängt |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss ein paar Buchstaben eingegeben haben, um das Suchergebnis einzuschränken |
| Ablauf | 1. Programm öffnen 2. Bei Startstation Teil des Ortes eintragen 3. Auf Button klicken |
| Ergebnis | Liste mit Stationen werden aufgelistet, welche mit dem User seiner Eingabe anfängt. |

## Endstation suchen

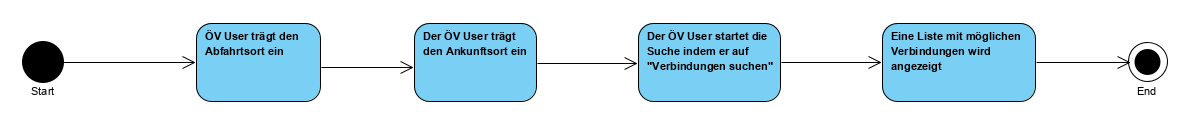
|  |  |
| --- | --- |
| Use Case | Endstation suchen |
| Beschreibung | ÖV Benutzer möchte nicht den ganzen Namen der Endstation eingeben, weshalb er auf einen Button klicken kann dem Ihm alle Stationen vorschlägt, die mit seiner Eingabe anfängt |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss ein paar Buchstaben eingegeben haben, um das Suchergebnis einzuschränken |
| Ablauf | 1. Programm öffnen 2. Bei Endstation Teil des Ortes eintragen 3. Auf Button klicken |
| Ergebnis | Liste mit Stationen werden aufgelistet, welche mit dem User seiner Eingabe anfängt. |

## Verbindungen suchen

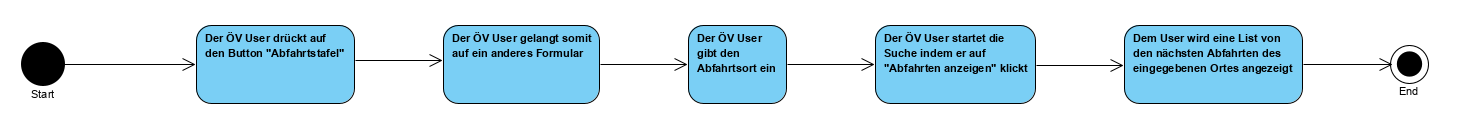
|  |  |
| --- | --- |
| Use Case | Verbindungen suchen |
| Beschreibung | ÖV Benutzer möchte alle Verbindungen zwischen 2 Stationen angezeigt bekommen |
| Akteur(e) | ÖV Benutzer |
| Auslöser | ÖV Benutzer |
| Vorbedingungen | User muss Startstation eingeben und Endstation |
| Ablauf | 1. Programm öffnen 2. Startstation eingeben 3. Endstation eingeben 4. Button klicken |
| Ergebnis | Liste mit Verbindungen zwischen den beiden Stationen werden aufgelistet |

# UML-Diagramme

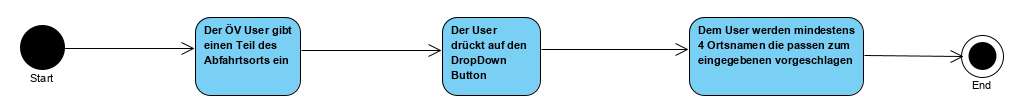
## Hauptfunktion Abfahrtsplan



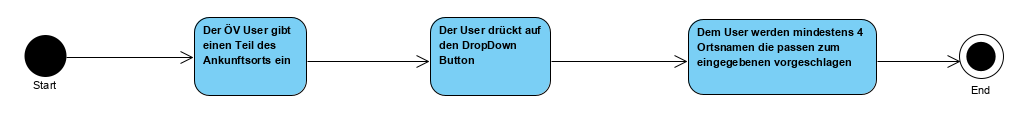
## Hauptfunktion Abfahrtsplan



## Ortsvorschläge Abfahrtsort



## Ortsvorschläge Ankunftsort



# Programmierrichtlinien

## Namensgebung von Variablen, Eigenschaften, Methoden:

* Globale Variabel mit Grossbuchstaben beginnen
* Variabel in Methoden mit Kleinbuchstaben beginnen & neues Wort mit Grossbuchstaben beginnen
* Methodennamen immer mit Grossbuchstaben beginnen

+

Man erkennt direkt um was es sich handelt

* Geschweifte Klammern immer auf neuer Zeile

+

Schöne Formatierung

Man erkennt direkt wo z.B. eine Methode endet

## Kommentare:

* Wenig Kommentare
* Kommentare vor Funktion was sie kann / immer mit drei Schrägstrichen beginnen

+

Man hat eine kleine Einführung, was die Funktion kann

Einheitlich

## GUI-Controls

* Control-Namen immer mit drei Buchstaben beginnen, welche aussagen was sie sind z.B.

1. Button = btn
2. Label = lbl
3. Checkbox = cbx
4. Radio Button = rbt
5. Datagridview = dgv
6. Textbox = txt
7. Listbox = lbx
8. Combobox = ddl