**Dokumentation Projektarbeit** Fahrplan-Applikation

**ÜK 318-1**

Analysieren und objektbasiert programmieren mit Komponenten

**Lukas Bucher**

**Adligenswil, 04.12.2019**

Eingereicht bei Markus Estermann

Inhaltsverzeichnis

[1. Einleitung & Zweck 3](#_Toc26304155)

[2. Funktionsumfang 3](#_Toc26304156)

[3. Mockups 3](#_Toc26304157)

[3.1 Verbindung 3](#_Toc26304158)

[3.2 Abfahrtsplan 4](#_Toc26304159)

[3.3 Informationen 4](#_Toc26304160)

[4. Use Cases 5](#_Toc26304161)

[4.1 Use Case Diagramm 5](#_Toc26304162)

[4.2 Use Case Beschreibungen 5](#_Toc26304163)

[4.3 Aktivitätsdiagramm 7](#_Toc26304164)

[5. Programmierrichtlinien 8](#_Toc26304165)

[5.1 Naming Conventions 8](#_Toc26304166)

[5.2 Declaration 8](#_Toc26304167)

[5.3 Comments 8](#_Toc26304168)

[6. Testing 8](#_Toc26304169)

[7. Installationsanleitung 8](#_Toc26304170)

# Einleitung & Zweck

…

…

…

…

…

# Funktionsumfang

## Implementierte Funktionen

…

…

…

…

## Fehlende Funktionen

…

…

…

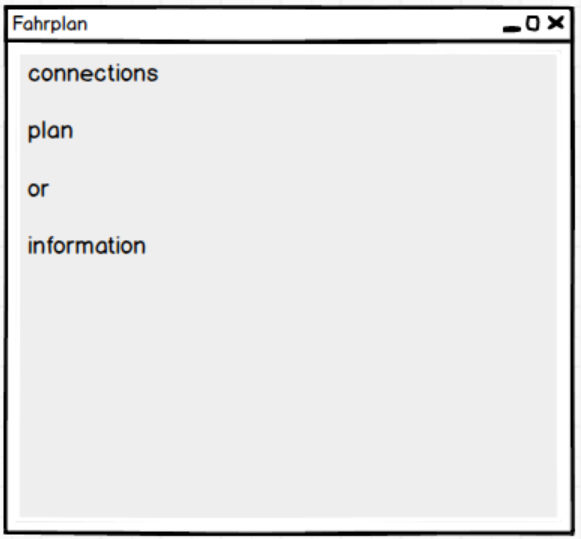
# Mockups

## Verbindung

## Abfahrtsplan

## Informationen

## Ausgabefenster



# Use Cases

## Use Case Diagramm

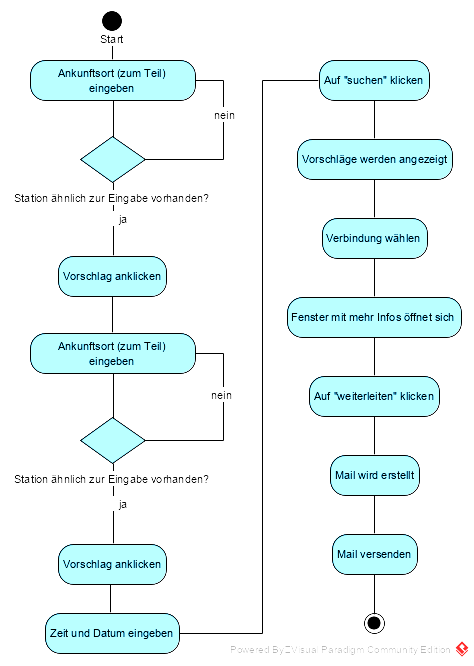
## Use Case Beschreibungen

### dfgdfsgdfgsdfg

|  |  |
| --- | --- |
| Use Case #01 | Mind. 5 Verbindungen anzeigen |
| Beschreibung | Der Benutzer sucht eine Verbindung zwischen selbst gewähltem Abfahrts- und Ankunftsort. Es kann die aktuelle oder eine selbst gewählte Zeit bestimmt werden. Es sollen dann die nächsten fünf Verbindungen sichtbar sein. |
| Ablauf | 1. Abfahrtsort, Ankunftsort, Zeit eingeben 2. Verbindung suchen 3. Eine der angezeigten Verbindungen für mehr Infos anwählen |
| Ergebnis | Der Benutzer findet schnell eine passende Verbindung für seine Reise. |
| Use Case #02 | Abfahrtstafel anzeigen |
| Beschreibung | Der Benutzer alle Verbindungen sehen, die zur gewählten Zeit vom gewählten Abfahrtsort fahren. |
| Ablauf | 1. Auf Abfahrtstafel-Maske wechseln 2. Abfahrtsort, Zeit eingeben 3. Verbindungen suchen 4. Eine der angezeigten Verbindungen für mehr Infos anwählen |
| Ergebnis | Der Benutzer sieht die Abfahrtstafel für den gewählten Abfahrtsort. |

|  |  |
| --- | --- |
| Use Case #03 | Eingabe vervollständigen |
| Beschreibung | Der Benutzer möchte nach Stationen suchen können, damit er sich nicht alle Namen auswendig merken muss. Allenfalls kann die Vervollständigung bereits während dem Tippen geschehen. |
| Ablauf | 1. Abfahrtsort / Ankunftsort eingeben 2. Eingabevervollständigung wird vorgeschlagen 3. Gesuchte Station anwählen |
| Alternativer Ablauf | 1. Auf Abfahrtstafel-Maske wechseln 2. Abfahrtsort eingeben 3. Eingabevervollständigung wird vorgeschlagen 4. Gesuchte Station anwählen |
| Ergebnis | Der Benutzer muss nicht den gesamten Namen der Station wissen und ihn abtippen. Er kann gemütlich einen Vorgeschlagenen Ort anklicken. |

## Aktivitätsdiagramm

Dieses Aktivitätsdiagramm stellt den Ablauf des Benutzers dar, wenn er sich Verbindungen mit gewähltem Abfahrts- und Ankunftsort anzeigen lässt und eine Verbindung anschliessend per Mail weiterleitet.

# Programmierrichtlinien

## Naming Conventions

* Variablen kurz, prägnant, logisch benennen
* camelCase verwenden
* Globale Variablen nicht redundant
* Methoden/Funktionen im Idealfall Verb und Nomen

## Declaration

* Variablen immer direkt initialisieren
* Globale Variablen nach load Form deklarieren

## Comments

* Code sollte selbsterklärend sein
* Kommentare nur wenn unbedingt nötig

# Testing

## Testfall «Hin- und Rückfahrt anzeigen lassen»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

# Installationsanleitung