

Dokumentation TransportApplication

|  |  |
| --- | --- |
| **Ersteller:** | Sacha Vogel |
| **Zweck:** | Dokumentation des TransportApplication Projekts |
| **Erstellungsdatum:** | 05.03.2020 |
| **Speicherdatum:** | 10.03.2020 |

Inhalt

[Einleitung 3](#_Toc34902556)

[Anforderungen 3](#_Toc34902557)

[GUI MockUps 4](#_Toc34902558)

[Use Case 5](#_Toc34902559)

[Use Case UML 5](#_Toc34902560)

[Use Case Beschreibung 5](#_Toc34902561)

[Aktivitäten-Diagramme 7](#_Toc34902562)

[Porgrammierrichtlinien 10](#_Toc34902563)

[Naming Conventions 10](#_Toc34902564)

[Declaration 10](#_Toc34902565)

[Comments 10](#_Toc34902566)

[Statements 10](#_Toc34902567)

# Einleitung

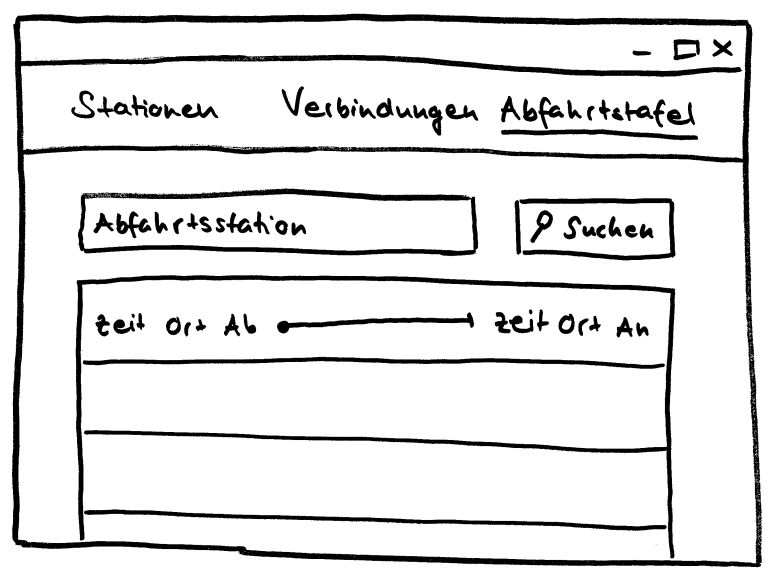
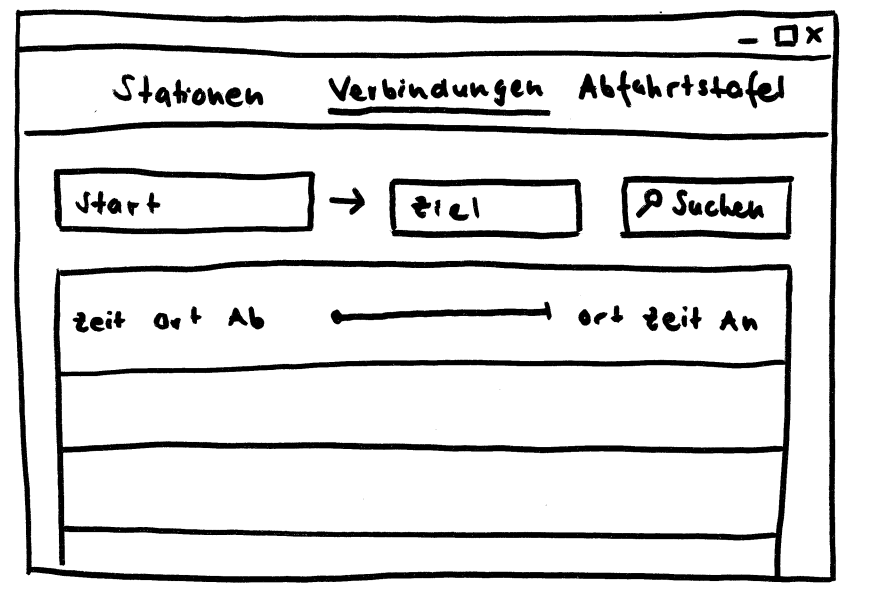
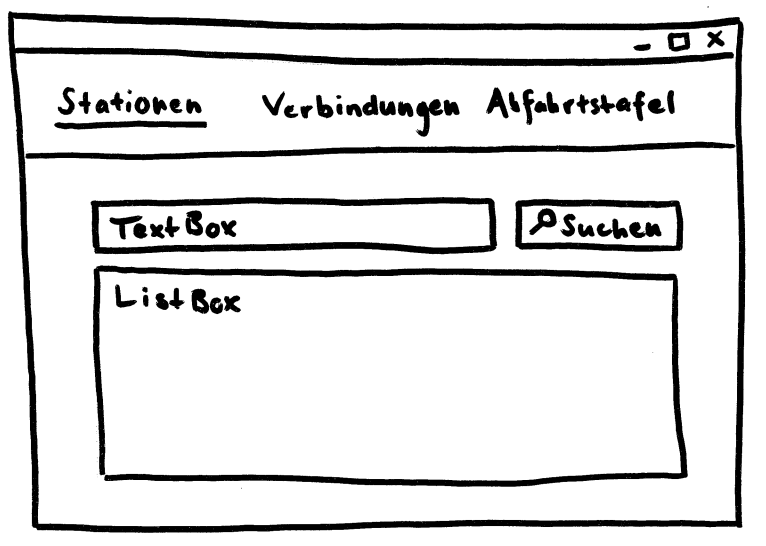
Dieses Dokument dient dem Zweck mein Projekt im ÜK-318 zu dokumentieren. Beim Projekt handelt es sich um eine Applikation die nach Verkehrs-Verbindungen suchen kann (TransportApplication). Die Planung sowie die ersten Entwürfe des GUI sind in diesem Dokument beigelegt. Die genauen Anforderungstests und auch die dazugehörigen Ergebnisse sind hier zu finden.

# Anforderungen

In meinem Programm wurden die Anforderungen A001 bis A006 umgesetzt und implementiert. Die restlichen Anforderungen wurden vernachlässigt, damit die Qualität der Software bei den vorhandenen Anforderungen höher ist.

Mein Programm kann nach allen Stationen suchen, auch Verbindungen zwischen zwei beliebigen Stationen kann man anzeigen lassen. Diese Verbindungen kann man auch mit einem Filter nach Datum und Uhrzeit suchen. Bei der Suche nach Verbindungen werden einem automatisch Stationen vorgeschlagen, die man aussuchen kann. Wenn man eine bestimmte Station sucht und diese auswählt, kann man sich diese auch in Google-Maps anzeigen lassen um den Standort genauer zu lokalisieren.

# GUI MockUps

Die ersten MockUps für mein GUI sind hier beschrieben, diese wurden von Hand gezeichnet. In meinem Programm kann man sehen, dass ich diesem Style treu geblieben bin und die Software am Schluss fast genau gleich ausschaut. In der Menü-Liste kann man jeweils sehen in welchem Tab man sich befindet.

# Use Case

Use Case sind eine Art Anforderungsbeschreibung, in der, gewisse Punkte genau beschrieben und dokumentiert werden, damit man sich über die Funktionen des Programms im Klaren ist.

## Use Case UML

## Use Case Beschreibung

|  |  |
| --- | --- |
| Use Case | 1, Stationen suchen |
| Beschreibung | Der ÖV-Benutzer kann alle Stationen im Programm suchen, diese werden ihm dann vorgeschlagen und angezeigt. |
| Akteur | Der ÖV-Benutzer |
| Auslöser | Der ÖV-Benutzer möchte alle Stationen die im System erfasst sind einfach suchen können, damit er diese nicht auswendig lernen muss. |
| Vorbedingungen | Der Benutzer hat einen Teil des Namens der Station eingegeben, ins vorgesehene Feld eingegeben und den Such-Button betätigt. |
| Ablauf | 1. Die TransportApplication öffnen. 2. Über das Menü zum Tab Stationen wechseln. 3. Den Anfang oder ganzen Namen der zu suchenden Station im dafür gesehenen Feld eingeben. 4. Auf den Button Suchen klicken. |
| Ergebnis | Alle Stationen die mit dem eingegebenen Text beginnen werden angezeigt. |

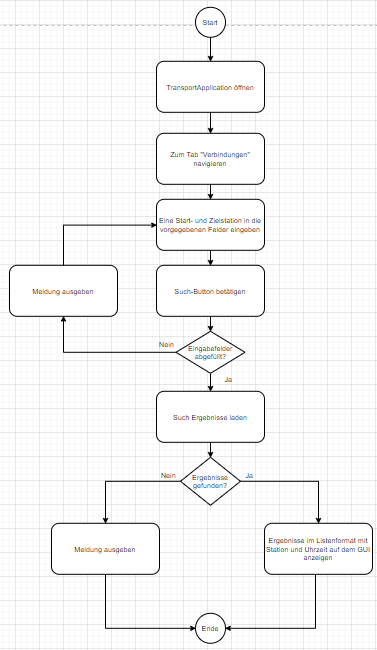
|  |  |
| --- | --- |
| Use Case | 2, Verbindungen sehen |
| Beschreibung | Der ÖV-Benutzer möchte Verbindungen zwischen verschiedenen Orten sehen, damit dieser weiss, wann er zur Station muss. |
| Akteur | Der ÖV-Benutzer |
| Auslöser | Der ÖV-Benutzer möchte Verbindungen zu verschiedenen Zeiten sehen. |
| Vorbedingungen | Der Benutzer hat eine Start- und Zielstation in die entsprechenden Felder eingegeben und den Such-Button betätigt. |
| Ablauf | 1. Die TransportApplication öffnen. 2. Über das Menü zum Tab Verbindungen wechseln. 3. Eine Start- und Zielstation in die vorgegebenen Felder eingeben. 4. Such-Button betätigen. |
| Ergebnis | Die nächsten Verbindungen zwischen den eingegebenen Stationen werden angezeigt. |

|  |  |
| --- | --- |
| Use Case | 3, Abfahrtstafel anzeigen |
| Beschreibung | Der ÖV-Benutzer kann nach einer beliebigen Station suchen, von dieser aus werden dann die nächsten Verbindungen überallhin angezeigt. |
| Akteur | Der ÖV-Benutzer |
| Auslöser | Der ÖV-Benutzer alle Verbindungen die von der eingegebenen Station ausgehen sehen. |
| Vorbedingungen | Der Benutzer hat eine Station ins vorgesehene Feld eingegeben und den Such-Button angeklickt. |
| Ablauf | 1. Die TransportApplication öffnen. 2. Über das Menü zum Tab Abfahrtstafel wechseln. 3. Eine Station in das entsprechende Feld eingeben. 4. Auf den Such-Button drücken. |
| Ergebnis | Die nächsten Verbindungen die von dieser Station ausgehen werden angezeigt. |

# Aktivitäten-Diagramme

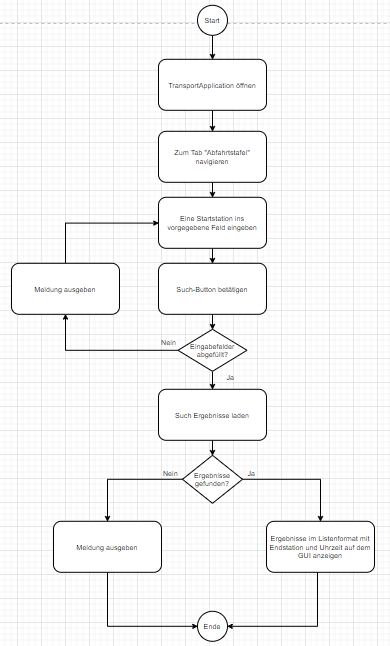
**Stationen suchen**

Das Aktivitäten-Diagramm zeigt welche Schritte abgearbeitet werden, wenn man nach einer Station sucht. Erst muss man die TransportApplication öffnen und zum Tab Stationen navigieren. Danach einen Teil oder die ganze Station ins Textfeld eingeben. Nach dem Betätigen des Such-Buttons wird geprüft, ob eine Station eingegeben wurde, falls das nicht der Fall ist wird eine Meldung ausgegeben. Wenn es OK ist werden die Suchergebnisse geladen. Wenn diese nicht vorhanden sind wird eine entsprechende Meldung ausgegeben. Ansonsten kann man nun die Stationen sehen.



**Verbindungen sehen**

Das Aktivitäten-Diagramm zeigt welche Schritte abgearbeitet werden, wenn man nach einer Verbindung sucht. Erst muss man die TransportApplication öffnen und zum Tab Verbindungen navigieren (Diese sollte schon von Anfang an geöffnet sein). Danach einen Teil oder die ganze Station in die Such-ComboBoxen eingeben. Nach dem Betätigen des Such-Buttons wird geprüft, ob beide Stationen eingegeben wurden, falls das nicht der Fall ist wird eine Meldung ausgegeben. Man kann auch nach einem gewünschten Datum und oder einer Uhrzeit suchen. Wenn alle Eingaben OK sind, werden die Suchergebnisse geladen. Wenn diese nicht vorhanden sind wird eine entsprechende Meldung ausgegeben. Ansonsten werden die gewünschten Verbindungen angezeigt.



**Abfahrtstafel anzeigen**

Das Aktivitäten-Diagramm zeigt welche Schritte abgearbeitet werden, wenn man nach einer Station für eine Abfahrtstafel sucht. Erst muss man die TransportApplication öffnen und zum Tab Abfahrtstafel navigieren. Danach die ganze Station in die TextBox eingeben. Nach dem Betätigen des Such-Buttons wird geprüft, ob eine Station eingegeben wurde, falls das nicht der Fall ist wird eine Meldung ausgegeben. Wenn die Eingabe OK ist, werden die Suchergebnisse geladen. Wenn diese nicht vorhanden ist wird eine entsprechende Meldung ausgegeben. Ansonsten werden die gewünschten Verbindungen angezeigt.

# Porgrammierrichtlinien

## Naming Conventions

* Variablen/Methoden Camel Case
* Klassen Pascal Case
* Statische Variablen und Konstanten alles gross, Wörter durch Bodenstrich trennen

## Declaration

* Membervariablen als private deklarieren
* Membervariablen zu oberst in der Klasse
* Konstruktor gleich nach Membervariablen deklarieren
* Getter und Setter mit C# Eigenschaften erstellen
* Methoden standartmässig private, ausser es ist nicht möglich dann public

## Comments

* Kommentare nur, wenn nötig
* Auf der Zeile oben der Anweisung oder des Blockes
* Kommentare nur auf einer Zeile
* Kommentare mit «//» erstellen
* Keine Kommentare mit «/\* \*/»

## Statements

* Geschweifte Klammern zum Öffnen eines Code-Blocks nicht auf neue Zeile
* Geschweifte Klammern zum Schliessen auf eigene Linie

# Testen

Das Testing ist ein wichtiger Punkt in der Entwicklung, man möchte ja sicherstellen, dass das Programm richtig funktioniert und möglichst viele Fehler vor dem Kunden finden. Es gibt verschiedene Arten wie man testen kann, ich habe die TransportApplication manuell getestet, die Ergebnisse und Testschritte zu den einzelnen Anforderungen sind in den folgenden Abschnitten dokumentiert.

## A001, Stationen suchen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Given | When | Then | Ergebnis |
| 1. | Die Applikation ist gestartet und man befindet sich im Tab «Stationen». | Benutzer gibt «Luz» in den Suchbalken ein. | - |  |
| 2. | Der vorherige Schritt wurde ausgeführt. | Such-Button klicken | Die Suchergebnisse werden geladen und alle Ergebnisse die «Luz» im Namen haben werden angezeigt. |  |

## A002, Verbindungen sehen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Given | When | Then | Ergebnis |
| 1. | Die Applikation ist gestartet und man befindet sich im Tab «Verbindungen». | Benutzer gibt «Sursee» in die Startstation ein. | - |  |
| 2. | Der vorherige Schritt wurde ausgeführt. | Benutzer gibt «Luzern» in die Endstation ein. | - |  |
| 3. | Der vorherige Schritt wurde ausgeführt. | Such-Button anklicken | Die Suchergebnisse werden geladen und alle Verbindungen von «Sursee» nach «Luzern» werden angezeigt. |  |

## A003, Abfahrtstafel anzeigen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Given | When | Then | Ergebnis |
| 1. | Die Applikation ist gestartet und man befindet sich im Tab «Abfahrtstafel». | Benutzer gibt «Sursee» in den Suchbalken ein. | - |  |
| 2. | Der vorherige Schritt wurde ausgeführt. | Such-Button klicken | Die Suchergebnisse werden geladen und alle Verbindungen die von Luzern weggehen werden angezeigt |  |

## A004, Suchvorschläge angeben

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Given | When | Then | Ergebnis |
| 1. | Die Applikation ist gestartet und man befindet sich im Tab «Verbindungen». | Benutzer gibt «Luz» in die Startstation ein. | Es werden sofort alle Ergebnisse die «Luz» im Namen haben vorgeschlagen. |  |
| 2. | Der vorherige Schritt wurde ausgeführt. | Benutzer wählt Ergebnis «Luzern» aus | «Luzern» wird in die Startstation abgefüllt. |  |
| 3. | Der vorherige Schritt wurde ausgeführt. | Benutzer gibt «Sur» in die Endstation ein. | Es werden sofort alle Ergebnisse die «Sur» im Namen haben vorgeschlagen. |  |
| 4. | Der vorherige Schritt wurde ausgeführt. | Benutzer wählt Ergebnis «Sursee» aus | «Sursee» wird in die Startstation abgefüllt. |  |
| 5. | Der vorherige Schritt wurde ausgeführt. | Such-Button klicken | Alle Verbindungen zwischen von Luzern nach Sursee werden angezeigt. |  |

## A005, Verbindungszeitpunkt anzeigen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Given | When | Then | Ergebnis |
| 1. | Die Applikation ist gestartet und man befindet sich im Tab «Verbindungen». | Benutzer gibt «Luzern» in die Startstation ein. | - |  |
| 2. | Der vorherige Schritt wurde ausgeführt. | Benutzer gibt «Sursee» in die Endstation ein. | - |  |
| 3. | Der vorherige Schritt wurde ausgeführt. | Benutzer wählt das heutige Datum ein. | - |  |
| 4. | Der vorherige Schritt wurde ausgeführt. | Benutzer gibt wählt eine beliebige Zeit aus. | - |  |
| 5. | Der vorherige Schritt wurde ausgeführt. | Such-Button klicken | Verbindungen ab dem eingegebenen Zeitpunkt werden angezeigt |  |

## A006, Standort anzeigen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Given | When | Then | Ergebnis |
| 1. | Die Applikation ist gestartet und man befindet sich im Tab «Stationen». | Benutzer gibt «Sursee» in das Suchfeld ein. | - |  |
| 2. | Der vorherige Schritt wurde ausgeführt. | «Google Maps»-Button klicken | Es öffnet sich der Browser und der Standort «Sursee» wird auf Google-Maps angezeigt |  |

# Installationsanleitung

Das Programm liegt auf Git in einem öffentlichen Repository. Wenn man die Applikation starten möchte, kann man dieses einfach klonen und ausführen. Wie man das genau macht wird hier Schritt für Schritt beschrieben.