Dokumentation Modul 318

Elias Roth

[1 Einleitung 3](#_Toc38636194)

[1.1 Anforderungen 3](#_Toc38636195)

[1.2 Zweck der Dokumentation 3](#_Toc38636196)

[2 Use Case 4](#_Toc38636197)

[2.1 Use Case Beschreibungen 4](#_Toc38636198)

[2.1.1 End- Startstation suchen (A001) 4](#_Toc38636199)

[2.1.3 Verbindungen Suchen (A001) 5](#_Toc38636200)

[2.1.3 Vorgeschlagene Verbindungen (A002) 5](#_Toc38636201)

[2.1.4 Station Wählen (A003) 5](#_Toc38636202)

[3 Mockups 6](#_Toc38636203)

[3.1 Beschreibung der Mockups 6](#_Toc38636204)

[3.1.1 Startseite (A001 + A002) 6](#_Toc38636205)

[3.1.2 Stationen (A003) 7](#_Toc38636206)

[4 Aktivitätendiagramme 8](#_Toc38636207)

[4.1 Beschreibung der Aktivitätendiagramme 8](#_Toc38636208)

[4.1.1 Aktivitätendiagramm (A001 + A002) 8](#_Toc38636209)

[4.1.2 Aktivitätendiagramm (A003) 8](#_Toc38636210)

[5 Programmierrichtlinien 9](#_Toc38636211)

[5.1 Namensgebung 9](#_Toc38636212)

[5.2 Kommentare 9](#_Toc38636213)

[6 Testing 9](#_Toc38636214)

[6.1 Testszenarien 9](#_Toc38636215)

[6.1.1 Test 1(A001) 9](#_Toc38636216)

[6.1.2 Test 2(A002) 9](#_Toc38636217)

[6.1.3 Test 3(A003) 10](#_Toc38636218)

[6.1.4 Test 4(A004) 10](#_Toc38636219)

[6.1.5 Test 5(A005) 10](#_Toc38636220)

[6.1.6 Test 6(A006) 10](#_Toc38636221)

[6.1.7 Test 7(A008) 11](#_Toc38636222)

[6.1.8 Test 8(Umkehren der Start und Endstation) 11](#_Toc38636223)

[6.1.9 Test 9(Weitere Verbindungen Anzeigen lassen) 11](#_Toc38636224)

[7 Installationsanleitung 11](#_Toc38636225)

[7.1 Beschreibung 11](#_Toc38636226)

# 1 Einleitung

Im Ük für Modul 318 haben wir eine Projektaufgabe bekommen eine ÖV-Applikation (Sowie ungefähr im Stil der SBB Fahrplan App) zu erstellen. Dazu haben wir 8 Anforderungen erhalten, welche die Applikation erfüllen muss.

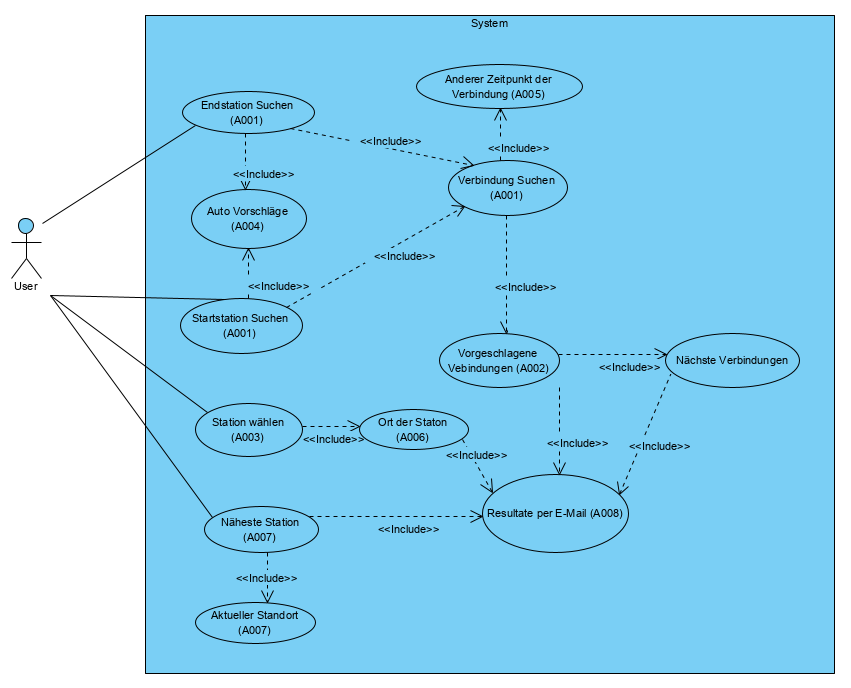
## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Beschreibung | Priorität | Erfüllt |
| **A001** | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | 1 |  |
| **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 |  |
| **A003** | 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 |  |
| **A004** | Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann. | 2 |  |
| **A005** | 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 |  |
| **A006** | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. | 3 |  |
| **A007** | Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann. | 3 |  |
| **A008** | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. | 3 |  |

## Zweck der Dokumentation

Diese Dokumentation dient der Festhaltung und Zusammenstellung des Beschriebes der Umsetzung der Anforderungen, der GUI-Mockups, der Use Cases und Aktivitäten Diagramme für A001 – A003, der Beschreibung der Installation bzw. Deinstallation, der durchgeführten Testfälle und einem Testprotokoll, in dem die Testergebnisse festgehalten werden.

# Use Case

Use Cases werden gebraucht, um die Funktionalität der Applikation aufzulisten.

## Use Case Beschreibungen

In den folgenden Zeilen werde ich die Use Cases der A001 – A003 beschreiben.

### End- Startstation suchen (A001)

|  |  |
| --- | --- |
| Use Case | Station Erfassen |
| Beschreibung | Ein Kunde will sich eine Bahnverbindung zwischen zwei Stationen suchen. |
| Akteur | Kunde der Bahngemeinschaft (Ist immer gleichdeshalb in den nächsten Beschreibungen nicht aufgelistet). |
| Auslöser | Jemand will eine Busverbindung suchen. |
| Vorbedingungen | Ungefähres wissen der Namen der gesuchten Stationen. |
| Ablauf | 1. Applikation Ausführen 2. Start und Endstationen eingeben. 3. Auf «Verbindung Suchen» klicken |
| Alternativer Ablauf | Keiner |
| Ergebnis | Stationsnamen werden mithilfe von «Verbindungen Suchen» an «Verbindungen Suchen geschickt. |

### Verbindungen Suchen (A001)

|  |  |
| --- | --- |
| Use Case | Station Erfassen |
| Beschreibung | Ein Kunde will sich eine Bahnverbindung zwischen zwei Stationen suchen. |
| Auslöser | Jemand will eine Busverbindung suchen. |
| Vorbedingungen | Kunde hat bereits den Start und Endstation angegeben |
| Ablauf | 1. Verbindungen zwischen angegebenen Stationen wird in der SBB API nach der nächsten Ankunftszeit gesucht. 2. Erhaltene Verbindungen werden an «Vorgeschlagene Verbindungen» weitergegeben. |
| Alternativer Ablauf | Keiner |
| Ergebnis | Verbindungen wurden gefunden und sind nun bereit ausgegeben zu werden. |

### 2.1.3 Vorgeschlagene Verbindungen (A002)

|  |  |
| --- | --- |
| Use Case | Station Erfassen |
| Beschreibung | Ein Kunde will sich eine Bahnverbindung zwischen zwei Stationen suchen. |
| Auslöser | Jemand will eine Busverbindung suchen. |
| Vorbedingungen | Die Verbindungen sind gefunden und müssen dem Kunden nur noch präsentiert werden. |
| Ablauf | 1. Erhaltene Verbindungen von «Verbindungen Suchen» werden in die Tabelle eingefügt und nach Ankunftszeit sortiert. |
| Alternativer Ablauf | Keiner |
| Ergebnis | Verbindungen sind für den Benutzer sichtbar. |

### Station Wählen (A003)

|  |  |
| --- | --- |
| Use Case | Station Erfassen |
| Beschreibung | Ein Kunde will sich eine Station und deren Verbindungen anschauen. |
| Auslöser | Jemand will die Verbindungen, die von einer Station ausgehen suchen. |
| Vorbedingungen | Kunde sollte den Namen der zu suchenden Station wissen. |
| Ablauf | 1. Applikation starten 2. Eingebe des Station namens 3. Suche der Station und deren Büsse/ Züge in der SBB API. 4. Ausgabe der Gefundenen büsse / Züge mit Abfahrtszeit. |
| Alternativer Ablauf | Keiner |
| Ergebnis | Der Kunde sieht nun alle Büsse /Züge, die von der gesuchten Station aus fahren. |

# 3 Mockups

Mockups dienen dazu das Design aufzuzeichnen und die Steuer Elemente, die für den Benutzer sichtbar sind zu platzieren.

## 3.1 Beschreibung der Mockups

In den folgenden Zeilen werden die Mockups zu den Anforderungen A001 – A003 und denen die gleich noch inbegriffen sind.

### 3.1.1 Startseite (A001 + A002)

Auf der Startseite meiner Applikation kann der Benutzer Start- und Endstation suchen (A001) und Stationsnamen werden vorgeschlagen damit man nicht alles auswendig wissen muss (A004). Wenn man beide Stationsnamen eingegeben hat und auch die Zeit und Datum der gewünschten Abfahrtszeit (A005) angegeben hat, findet man durch «Verbindung Suchen» die nächsten 5 Verbindungen mit Ab- und Ankunftszeiten (A002). Mit dem Button «Weitere Verbindungen» kann man die nächsten 4 Verbindungen Anzeigen lassen.

### 3.1.2 Stationen (A003)

Auf der Stationen Seite kann man durch das eingeben eines Station Namens eine Bestimmte Station mit den von ihr fahrenden Büssen oder Zügen etc. anzeigen lassen und deren Abfahrtszeiten (A003). Mit dem Button «Station auf der Karte Suchen» öffnet sich eine Karte, die die Station markiert hat (A006) und mit «Auswahl Mailen» sendet es die ausgewählten Daten an die links angegebene Adresse (A008). Es ist auch hier möglich Verbindungen zu einer anderen Zeit oder Datum zu suchen da zb sonntags andere Verbindungen sind lässt sich so die Info noch besser ablesen.

# 4 Aktivitätendiagramme

In Aktivitätendiagrammen wird Ablauf einer bestimmten Aufgabe im Programm dargestellt.

## 4.1 Beschreibung der Aktivitätendiagramme

In folgenden Zeilen werde ich das Aktivitätendiagramm zur Anforderung 1 +2 und jenes zur Anforderung 3 beschreiben und erklären was passiert.

### 4.1.1 Aktivitätendiagramm (A001 + A002)

In diesem Aktivitätendiagramm wird zuerst aufgezeigt, dass der Benutzer die Wahl hat ob er zuerst die Start- oder Endstation eingibt, da er danach auch noch die jeweilig andere eingeben kann. Nachdem die beiden Eingaben getätigt wurden wird gleich getestet ob dies auch gültige Eingaben sind, dies funktioniert im Programm dann indem es nur Vorgeschlagene Stationen, die dann angeklickt wurden als gültig ansieht, denn so werden unnötige Fehler verhindert. Sollte die Eingabe stimmen so werden die Daten zu den nächsten 4 Verbindungen geholt und schlussendlich in Form einer Tabelle ausgegeben. Sollte die Eingabe allerdings nicht stimmen so wird keine suche nach Verbindungen und dadurch auch keine Ausgabe genehmigt, aber es ist natürlich möglich die Eingabe noch zu ändern.

### 4.1.2 Aktivitätendiagramm (A003)

Hier gibt der Benutzer zuerst die gewollte Station ein, danach wird wie oben geprüft ob die Eingegebene Station als gültig angesehen wird oder nicht. Wenn nicht kann der Benutzer seine Angaben jederzeit ändern. Bei gültiger Eingabe wird das, zum Namen gehörige, Stationsboard aus der API geholt und in eine Tabelle Eingefüllt.

# 5 Programmierrichtlinien

Programmierrichtlinien dienen dazu einen einheitlichen Programmierstil beizubehalten und den Code verständlicher zu machen.

## 5.1 Namensgebung

1. Forms Elemente wie TextBoxen haben drei Buchstaben als Element-Bezeichner vor dem Namen. (txt\_textbox1)

2. Variablen sind klein zu schreiben. (var variabel)

3. Events haben On vor dem Namen und sind grossgeschrieben. (On\_Input)

## 5.2 Kommentare

1. Lehrzeichen und Lehrziele vor Kommentar. (// Text)

# 6 Testing

Mit dem Testing wird das geschriebene Programm getestet und auf anfällige Unstimmigkeiten überprüft.

## 6.1 Testszenarien

Es wird zu jedem Test ein Testprotokoll erstellt.

### 6.1.1 Test 1(A001)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Benutzer wählt auf der Startseite die TextBox Von an. | Die TextBox ist angewählt. | Die TextBox ist angewählt. |  |
| 2 | Benutzer gibt «Rüediswil» ein. | In der TextBox steht «Rüediswil». | In der TextBox steht «Rüediswil». |  |
| 3 | Benutzer wählt auf der Startseite die TextBox Nach an. | Die TextBox ist angewählt. | Die TextBox ist angewählt. |  |
| 4 | Benutzer gibt «Sursee» ein. | In der TextBox steht «Sursee». | In der TextBox steht «Sursee». |  |

### 6.1.2 Test 2(A002)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Testfall 1 ist beendet. | - | - |  |
| 2 | Der Benutzer klickt auf «Verbindung Suchen» | 4 Verbindungen werden ab der Startstation bis zur Zielstation angezeigt. | 4 Verbindungen werden ab der Startstation bis zur Zielstation angezeigt. |  |

## 6.1.3 Test 3(A003)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Der Benutzer klickt auf die TextBox Station im Fenster Stationen an | Die TextBox ist angewählt. | Die TextBox ist angewählt. |  |
| 2 | Der Benutzer gibt «Rüediswil» ein. | Die Verbindungen ab «Rüediswil» werden angezeigt. | Die Verbindungen ab «Rüediswil» werden angezeigt. |  |
| 3 | Der Benutzer klickt auf Station Suchen. |  |  |  |

### 6.1.4 Test 4(A004)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Der Benutzer klickt auf die TextBox Station im Fenster Stationen an. | Die TextBox ist angewählt. | Die TextBox ist angewählt. |  |
| 2 | Der Benutzer gibt «Rüedis» ein. | Vorschläge in Form einer List box mit dem ersten Eintrag «Rüediswil» | Vorschläge in Form einer List box mit dem ersten Eintrag «Rüediswil» |  |
| 3 | Der Benutzer klickt auf «Rüediswil» in der List box. | «Rüediswil» steht jetzt in der Textbox. | «Rüediswil» steht jetzt in der Textbox. |  |

### 6.1.5 Test 5(A005)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Testfall 1 ist beendet. | - | - |  |
| 2 | Der Benutzer gibt 20:04:2020 und die Uhrzeit 12:00 in den Dazugehörigen Boxen auf der Startseite. | Es steht im Datum und Uhrzeit die angegebenen daten. | Es steht im Datum und Uhrzeit die angegebenen daten. |  |
| 3 | Der Benutzer klickt auf «Verbindung Suchen» | Die ersten 4 Verbindungen ab der Angegebenen Zeit werden angegeben. | Die ersten 4 Verbindungen ab der Angegebenen Zeit werden angegeben. |  |

### 6.1.6 Test 6(A006)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Test 3 abgeschlossen. | - | - |  |
| 2 | Der Benutzer klickt «Station auf der Karte Ansehen» | Ein Popup ob der Browser geöffnet werden darf wird angezeigt. | Ein Popup ob der Browser geöffnet werden darf wird angezeigt. |  |
| 3 | Der Benutzer klickt auf «ok» | Der Browser mit der Position der Station geht auf. | Der Browser mit der Position der Station geht auf. |  |

### 6.1.7 Test 7(A008)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Test 3 beendet. | - | - |  |
| 2 | Der Benutzer setzt 3 kreuze in der Tabelle bei Mail(Oben, Mitte und Unten). | Angekreuzt. | Angekreuzt |  |
| 3 | Der Benutzer gibt «alias6017@gmail.com» in die Mailadresse TextBox ein. | Es steht «alias6017@gmail.com» in der Mailadresse TextBox. | Es steht «alias6017@gmail.  com» in der Mailadresse TextBox. |  |
| 4 | Der Benutzer klickt auf «Auswahl Mailen» | Eine Mail wird an «alias6017@gmail.com» gesendet. | Eine Mail wird an «alias6017@gmail.  com» gesendet. |  |
| 5 | Der Benutzer schaut das Mail an. | Im Mail stehen die Infos zu den angekreuzten Linien. | Im Mail stehen die Infos zu den angekreuzten Linien. |  |

### 6.1.8 Test 8(Umkehren der Start und Endstation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Testfall 1 ist beendet. | - | - |  |
| 2 | Benutzer wählt den Umkehrbutton zwischen den beiden TextBoxen. | «Sursee» steht in Von und «Rüediswil» steht in nach. | «Sursee» steht in Von und «Rüediswil» steht in nach. |  |

### 6.1.9 Test 9(Weitere Verbindungen Anzeigen lassen)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Test 1 +2 beendet | - | - |  |
| 2 | Der Benutzer klickt 1-mal auf «Weitere Verbindungen Anzeigen lassen». | 4 Weiter Verbindungen werden angezeigt. | 4 Weiter Verbindungen werden angezeigt. |  |

# 7 Installationsanleitung

Kurze Beschreibung zum Ausführen der Datei.

## 7.1 Beschreibung

Wenn der Ordner Elias\_Roth\_Modul\_318 heruntergeladen ist findet man dort einen Ordner mit den Binares und der exe Datei.