

M318

Analysieren und objektbasiert programmieren mit Komponenten



21. April 2020

Timon Schmid

CSS Versicherung

Inhaltsverzeichnis

[1. Projekt Informationen 2](#_Toc38626067)

[1.1 Einleitung 2](#_Toc38626068)

[1.2 Dokumentation 2](#_Toc38626069)

[1.3 Anforderungen 2](#_Toc38626070)

[1.3.1 Priorität 1 2](#_Toc38626071)

[1.3.2 Priorität 2 2](#_Toc38626072)

[1.3.3 Priorität 3 2](#_Toc38626073)

[2. Konzeptionierung 2](#_Toc38626074)

[2.1 Programmier-Richtlinien 2](#_Toc38626075)

[2.2 Wireframes 2](#_Toc38626076)

[2.2.1 Verbindungen 2](#_Toc38626077)

[2.2.2 Abfahrt Tafel 3](#_Toc38626078)

[2.3 Use Cases 3](#_Toc38626079)

[2.3.1 Use Case “Station suchen” 4](#_Toc38626080)

[2.3.2 Use Case “Verbindungen anzeigen” 4](#_Toc38626081)

[2.3.3 Use Case “Abfahrt Tafel anzeigen” 5](#_Toc38626082)

[2.4 Aktivität Diagramm 5](#_Toc38626083)

[2.4.1 A001 5](#_Toc38626084)

[2.4.2 A002 5](#_Toc38626085)

[2.4.3 A003 5](#_Toc38626086)

[2.5 Testfälle 5](#_Toc38626087)

[2.5.1 Testfall “Station suchen” 5](#_Toc38626088)

[2.5.2 Testfall “Verbindungen suchen” 6](#_Toc38626089)

[2.5.3 Testfall “Abfahrts Tafel anzeigen” 6](#_Toc38626090)

[3. Umsetzung 7](#_Toc38626091)

[3.1 Funktionalitäten 7](#_Toc38626092)

[3.2 Bugs und Fehler 7](#_Toc38626093)

[3.3 Testprotokoll 8](#_Toc38626094)

[4. Installation 8](#_Toc38626095)

# Projekt Informationen

**Autor:** Timon Schmid  
**Erstelldatum:** 21.04.2020  
**Projekt-Start:** 22.04.2020  
**Projekt-Ende:** 24.04.2020

## Einleitung

Dieses Projekt ist für das Modul 318. Es geht darum eine Desktop Applikation zu erstellen. Diese Applikation zeigt Verbindungen zwischen zwei Standorte oder mögliche Zielorte von einem ausgewählten Standort an. Die Daten werden anhand einer API von [Opendata](https://transport.opendata.ch/) zur Verfügung gestellt. Das GUI dazu wurde mit einer Windows Forms Applikation erstellt. Der Quellcode zum Programm ist auf GitHub zu finden.

## Dokumentation

Die Dokumentation dient zur Übersicht der Vorgehensweise und der aktuell vorhandenen Funktionalitäten der Applikation. Es ermöglicht eine einfache Erweiterbarkeit.

## Anforderungen

### Priorität 1

* Start- und Endstation mittels Textsuche suchen.
* Aktuelle Verbindungen zwischen zwei Stationen sehen.
* Vorhandene Verbindungen ab einer bestimmten Station sehen.

### Priorität 2

* Während der Eingabe einer Station bereits Suchresultate erhalten.
* Verbindungen zu einem beliebigen Zeitpunkt suchen

### Priorität 3

* Sehen wo sich eine Station befindet
* Stationen in meiner Nähe finden
* Resultate via Mail weiterleiten

# Konzeptionierung

## Programmier-Richtlinien

Als Programmier-Richtlinien werden die Standard-Richtlinien von Microsoft zu C# verwendet.  
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>

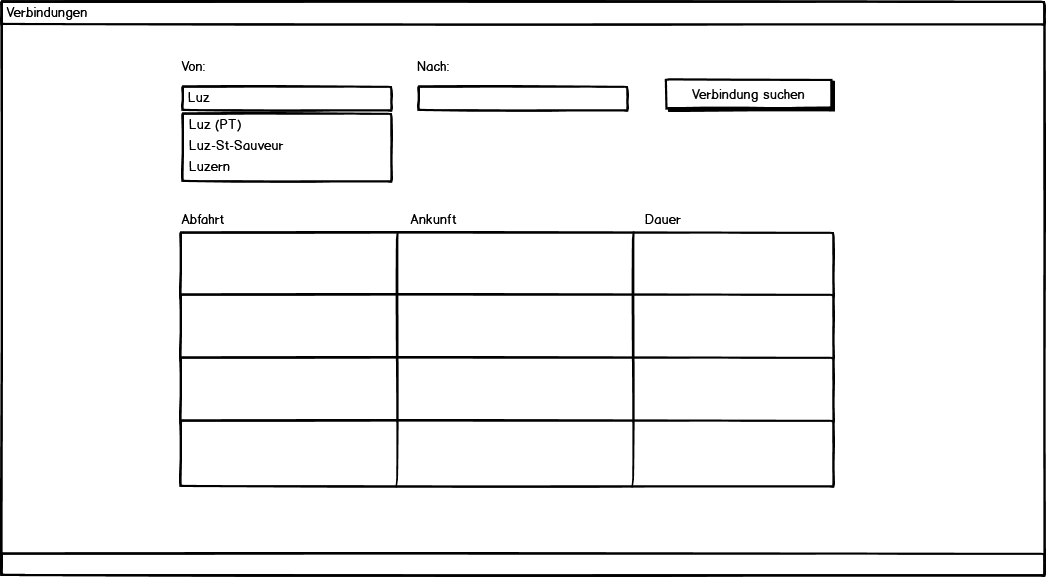
Visual Studio unterstützt diese Richtlinien. Zusätzlich verwende ich noch den ReSharper von JetBrains, welcher zusätzliche Unterstützung bietet.

## Wireframes

Um eine Grobansicht der Applikation zu erstellen, verwendete ich das Programm Balsamiq, welches das Erstellen von Wireframes sehr einfach ermöglicht.

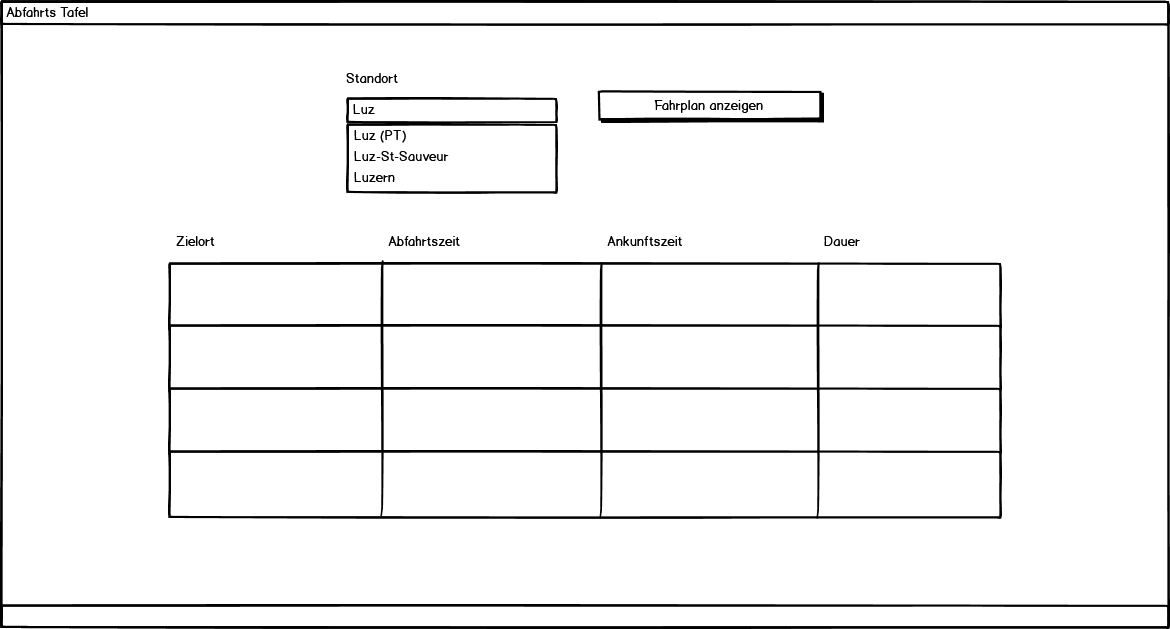
### Verbindungen

Auf der Seite ist es möglich, Verbindungen zwischen zwei Stationen zu suchen. Beim Eintippen einer Station werden mögliche Stationen aufgelistet, um die Suche zu vereinfachen. Beim Bestätigen der Suche werden die aktuellen Verbindungen zwischen den Stationen in der Tabelle angezeigt. Mit diesem Wireframe werden Anforderungen A001 & A002 abgedeckt.



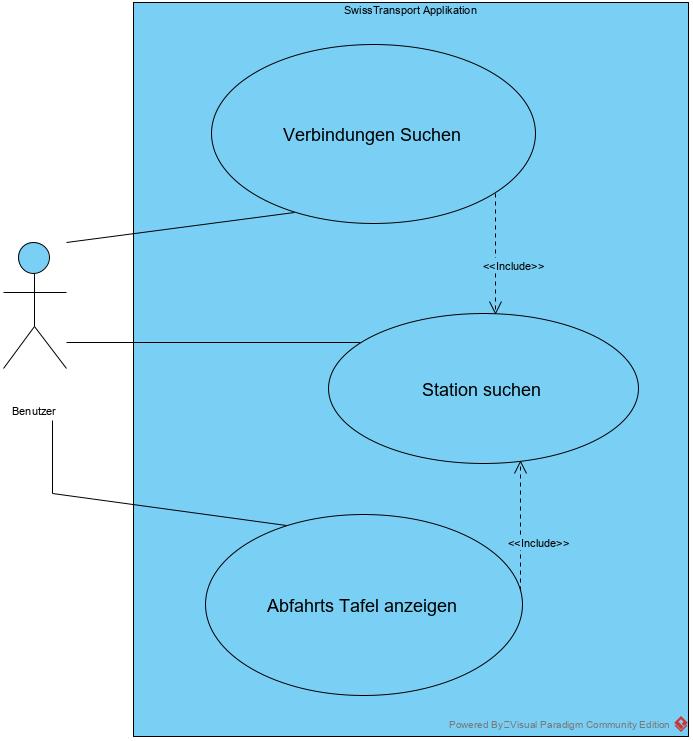
### Abfahrt Tafel

Bei der Abfahrt Tafel kann man eine gewünschte Station eingeben. Sobald der Knopf “Fahrplan anzeigen” geklickt wird, werden alle aktuellen Verbindungen, die ab dieser Station möglich sind, angezeigt. Mit diesem Wireframe wird die Anforderung A003 abgedeckt.



## Use Cases

Dieses Use Case Diagramm wurde für die ersten 3 Anforderungen A001 – A003 erstellt. Es zeigt die Haupt-Anwendungsfälle zwischen dem Benutzer und der Applikation.



### Use Case “Station suchen”

|  |  |
| --- | --- |
| Use Case | 1, Station suchen |
| Beschreibung | Der Benutzer möchte eine Station auswählen. Er tippt in das Textfeld und nach 2 eingegebenen Zeichen werden ihm Lösungsvorschläge angezeigt. |
| Akteur(e) | Benutzer |
| Auslöser | Benutzer tippt eine Station im Feld ein. |
| Vorbedingungen | Der Benutzer muss Internetverbindung haben damit einen Zugriff auf die API gewährleistet werden kann. |
| Ablauf | 1. Benutzer schreibt in das Textfeld 2. Ab 2 Zeichen werden Vorschläge angezeigt 3. Benutzer wählt eine vorgeschlagene Station aus |
| Alternativer Ablauf #1 | Keine |
| Ergebnis | Vorschläge wurden angezeigt und angewendet |

### Use Case “Verbindungen anzeigen”

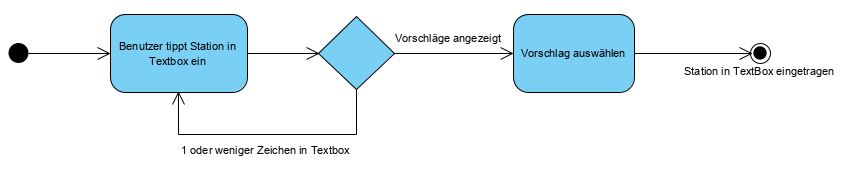
|  |  |
| --- | --- |
| Use Case | 2, Verbindungen anzeigen |
| Beschreibung | Der Benutzer möchte die aktuellen Verbindungen zwischen 2 ausgewählten Stationen anzeigen lassen. |
| Akteur(e) | Benutzer |
| Auslöser | Benutzer klickt auf den Knopf “Verbindung suchen” |
| Vorbedingungen | Der Benutzer muss Internetverbindung haben damit einen Zugriff auf die API gewährleistet werden kann. Ebenso müssen zwei Stationen ausgewählt sein, da sonst keine Suchergebnisse gefunden werden können. |
| Ablauf | 1. Benutzer klickt auf “Verbindung suchen” 2. Die nächsten 4 Verbindungen werden in der untenstehenden Tabelle angezeigt |
| Alternativer Ablauf #1 | Keine |
| Ergebnis | Die aktuellen Verbindungen zwischen den 2 Stationen werden angezeigt |

### Use Case “Abfahrt Tafel anzeigen”

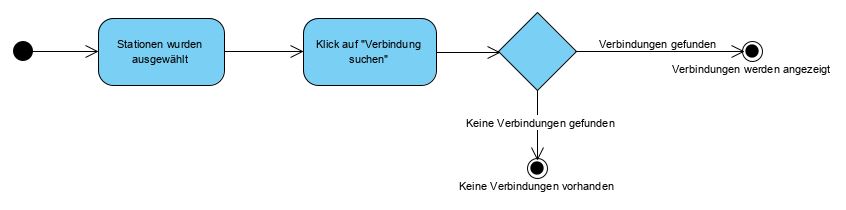
|  |  |
| --- | --- |
| Use Case | 3, Abfahrt Tafel anzeigen |
| Beschreibung | Der Benutzer kann möglichen Verbindungen ab einer bestimmten Station anzeigen lassen. |
| Akteur(e) | Benutzer |
| Auslöser | Benutzer klickt auf den Knopf “Fahrplan anzeigen” |
| Vorbedingungen | Der Benutzer muss Internetverbindung haben damit einen Zugriff auf die API gewährleistet werden kann. Ebenso muss eine Station ausgewählt sein, da sonst keine Suchergebnisse gefunden werden können. |
| Ablauf | 1. Benutzer klickt auf “ Fahrplan anzeigen ” 2. Die Nächsten möglichen Verbindungen werden in der Tabelle angezeigt |
| Alternativer Ablauf #1 | Keine |
| Ergebnis | Die aktuell möglichen Verbindungen, ab der Ausgewählten Station, werden angezeigt. |

## Aktivität Diagramm

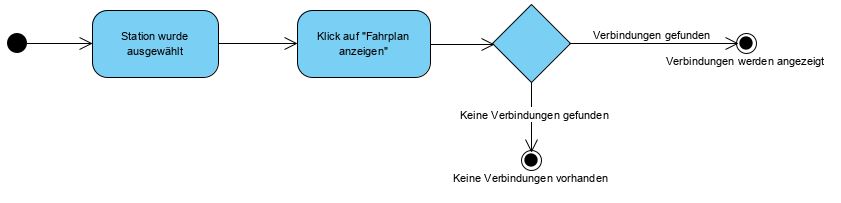
### A001



### A002



### A003



## Testfälle

### Testfall “Station suchen”

**Betroffene Anforderungen:**

A001 & A004

**Vorbedingungen**

Es muss eine Internetverbindung bestehen, damit das System auf die API zugreifen kann.

**Testszenario**

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Benutzer startet die Applikation | Applikation wird gestartet |
| 2 | Benutzer klickt auf den Tab “Verbindungen” | Tab “Verbindungen” wird angezeigt |
| 3 | Benutzer tippt erste Zeichen in die TextBox, um eine Station zu wählen:  Text = “Lu” | Mögliche Stationen zur Eingabe werden unterhalb aufgelistet:   * Luzern * Luzern, Bahnhof * Luzern, Kantonalbank * Lugano * ... |
| 4 | Benutzer wählt Vorschlag “Luzern” aus | Luzern wird in der TextBox eingetragen |

### Testfall “Verbindungen suchen”

**Betroffene Anforderungen:**

A002 & A005

**Vorbedingungen**

Es muss eine Internetverbindung bestehen, damit das System auf die API zugreifen kann.

**Testszenario**

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Benutzer startet die Applikation | Applikation wird gestartet |
| 2 | Benutzer klickt auf den Tab “Verbindungen” | Tab “Verbindungen” wird angezeigt |
| 3 | Benutzer wählte zwei Stationen:  Von = “Luzern”  Bis = “Sursee” | Stationen von “Luzern” nach “Sursee” wurden festgelegt. |
| 4 | Benutzer klickt auf den DateTimePicker für das Datum und wählt ein Datum aus:  Datum = 24.04.2020 | Datum “24.04.2020” wird in die DateTimePicker Box für Datum eingetragen |
| 5 | Benutzer klickt auf den DateTimePicker für die Zeit und wählt eine Zeit aus:  Datum = 12:00 | Zeit “12:00” wird in die DateTimePicker Box für Zeit eingetragen |
| 6 | Benutzer klickt auf “Verbindung suchen” | Verbindungen passend zu den eingegeben Werten werden angezeigt |

### Testfall “Abfahrts Tafel anzeigen”

**Betroffene Anforderungen:**

A005

**Vorbedingungen**

Es muss eine Internetverbindung bestehen, damit das System auf die API zugreifen kann.

**Testszenario**

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Benutzer startet die Applikation | Applikation wird gestartet |
| 2 | Benutzer klickt auf den Tab “Abfahrts Tafel” | Tab “Abfahrts Tafel” wird angezeigt |
| 3 | Benutzer wählte eine Station:  Station = “Kriens, Hammerschmiede” | Station “Kriens, Hammerschmiede” wird in die TextBox eingetragen |
| 4 | Benutzer klickt auf “Fahrplan anzeigen” | Verbindungen ab der Station werden angezeigt |

# Umsetzung

## Funktionalitäten

|  |  |  |  |
| --- | --- | --- | --- |
| Anforderung | Priorität | Status | Beschreibung |
| A001 | 1 | ✓ | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. |
| A002 | 1 | ✓ | 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. |
| A003 | 1 | ✓ | Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann. |
| A004 | 2 | ✓ | Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann. |
| A005 | 2 | ✓ | 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. |
| A006 | 3 | 🗶 | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. |
| A007 | 3 | 🗶 | Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann. |
| A008 | 3 | 🗶 | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. |

✓ = umgesetzt  
🗶 = Nicht umgesetzt

## Testprotokoll

### Testfall “Station suchen”

**Durchgeführt von:** Timon Schmid  
**Durchgeführt am:** 24.04.2020 – 11:30 Uhr

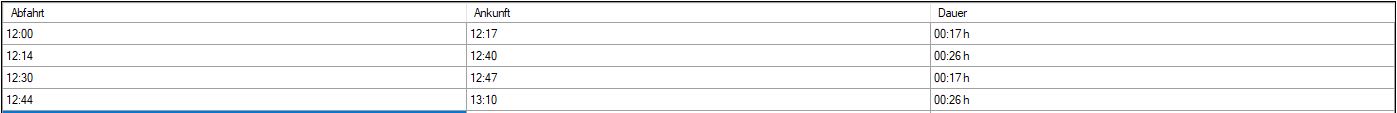
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Benutzer startet die Applikation | Applikation wird gestartet | Applikation wird gestartet | ✓ |
| 2 | Benutzer klickt auf den Tab “Verbindungen” | Tab “Verbindungen” wird angezeigt | Tab “Verbindungen” wird angezeigt | ✓ |
| 3 | Benutzer tippt erste Zeichen in die TextBox, um eine Station zu wählen:  Text = “Lu” | Mögliche Stationen zur Eingabe werden unterhalb aufgelistet:   * Luzern * Luzern, Bahnhof * Luzern, Kantonalbank * Lugano * ... | Mögliche Stationen zur Eingabe werden unterhalb aufgelistet:   * Luzern * Luzern, Bahnhof * Luzern, Kantonalbank * Lugano   ... | ✓ |
| 4 | Benutzer wählt Vorschlag “Luzern” aus | Luzern wird in der TextBox eingetragen | Luzern wird in der TextBox eingetragen | ✓ |

### Testfall “Verbindungen suchen”

**Durchgeführt von:** Timon Schmid  
**Durchgeführt am:** 24.04.2020 – 11:45 Uhr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Benutzer startet die Applikation | Applikation wird gestartet | Applikation wird gestartet | ✓ |
| 2 | Benutzer klickt auf den Tab “Verbindungen” | Tab “Verbindungen” wird angezeigt | Tab “Verbindungen” wird angezeigt | ✓ |
| 3 | Benutzer wählte zwei Stationen:  Von = “Luzern”  Bis = “Sursee” | Stationen von “Luzern” nach “Sursee” wurden festgelegt. | Stationen von “Luzern” nach “Sursee” wurden festgelegt. | ✓ |
| 4 | Benutzer klickt auf den DateTimePicker für das Datum und wählt ein Datum aus:  Datum = 24.04.2020 | Datum “24.04.2020” wird in die DateTimePicker Box für Datum eingetragen | Datum “24.04.2020” wird in die DateTimePicker Box für Datum eingetragen | ✓ |
| 5 | Benutzer klickt auf den DateTimePicker für die Zeit und wählt eine Zeit aus:  Datum = 12:00 | Zeit “12:00” wird in die DateTimePicker Box für Zeit eingetragen | Zeit “12:00” wird in die DateTimePicker Box für Zeit eingetragen | ✓ |
| 6 | Benutzer klickt auf “Verbindung suchen” | Verbindungen passend zu den eingegebenen Werten werden angezeigt | Verbindungen passend zu den eingegebenen Werten werden angezeigt:  *Siehe Abbildung* | ✓ |

Resultat:



### Testfall “Abfahrts Tafel anzeigen”

**Durchgeführt von:** Timon Schmid  
**Durchgeführt am:** 24.04.2020 – 13:00 Uhr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Tatsächliches Resultat | Status |
| 1 | Benutzer startet die Applikation | Applikation wird gestartet | Applikation wird gestartet | ✓ |
| 2 | Benutzer klickt auf den Tab “Abfahrts Tafel” | Tab “Abfahrts Tafel” wird angezeigt | Tab “Abfahrts Tafel” wird angezeigt | ✓ |
| 3 | Benutzer wählte eine Station:  Station = “Kriens, Hammerschmiede” | Station “Kriens, Hammerschmiede” wird in die TextBox eingetragen | Station “Kriens, Hammerschmiede” wird in die TextBox eingetragen | ✓ |
| 4 | Benutzer klickt auf “Fahrplan anzeigen” | Verbindungen ab der Station werden angezeigt | Verbindungen ab der Station werden angezeigt | ✓ |

## Bugs und Fehler

|  |  |
| --- | --- |
| **Nr** | **Beschreibung** |
| 1 | Wenn keine Stationen gefunden werden und aus der ComboBox geklickt wird stürzt die Applikation ab. |
| 2 | Das GUI ist vertikal nicht sehr anpassungsfähig |
| 3 |  |
| 4 |  |

# Installation

Eine Installationsmöglichkeit wurde nicht implementiert, da aufgrund der tiefen Priorität dieser Aufgabe, der Zeitaufwand zu gross gewesen wäre.