**Dokumentation ÜK**

M318

|  |  |
| --- | --- |
| Inhalt | **Dokumentation ÜK 318** |
| Erstelldatum | **Montag, 3. Dezember 2018** |
| Änderungsdatum | **Dienstag, 4. Dezember 2018** |
| Autor | **Ghezzi Lars** |

**Inhaltsverzeichnis**

[1 Einleitung 3](#_Toc531695153)

[1.1 Zweck des Dokuments 3](#_Toc531695154)

[1.2 Anforderungen 3](#_Toc531695155)

[2 Funktionen 4](#_Toc531695156)

[2.1 Station Textsuche 4](#_Toc531695157)

[2.2 Verbindungen suchen 4](#_Toc531695158)

[2.3 Abfahrtstafel anzeigen 5](#_Toc531695159)

[2.4 Station Textsuche während dem Schreiben 5](#_Toc531695160)

[2.5 Verbindungen zu bestimmter Zeit suchen 6](#_Toc531695161)

[3 Installationsanleitung 7](#_Toc531695162)

[4 Probleme/Bugs 8](#_Toc531695163)

[4.1 GUI Responsiv 8](#_Toc531695164)

[4.2 Performance 8](#_Toc531695165)

[4.2.1 Abfrage 8](#_Toc531695166)

[4.3 List Box Elemente zu TextBox 9](#_Toc531695167)

[4.4 Zugriff 9](#_Toc531695168)

[4.4.1 List Box befüllen ausserhalb des GUIs 9](#_Toc531695169)

[4.5 Arbeitsspeicher 10](#_Toc531695170)

[5 Planung 11](#_Toc531695171)

[5.1 Use Case 11](#_Toc531695172)

[5.2 Aktivitätsdiagramm 12](#_Toc531695173)

[5.3 Klassendiagramm 12](#_Toc531695174)

[5.4 Mock-Up 13](#_Toc531695175)

[6 Testfälle 14](#_Toc531695176)

[7 Coding Guidelines 17](#_Toc531695177)

# Einleitung

## Zweck des Dokuments

Dieses Dokument soll die Projektarbeit des 5. Überbetrieblichen Kurs erklären und aufzeigen, was erledigt wurde, wo es Fehler gab, wie es zu Installieren ist und wie es geplant wurde.

## Anforderungen

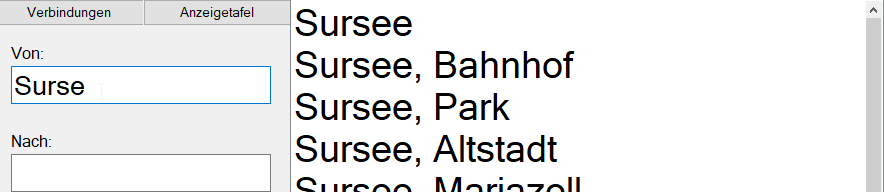
Diese Acht Anforderungen wurden zu Beginn definiert. Dabei müssen solche mit Priorität Eins unbedingt erledigt werden, solche mit Priorität Zwei sollten dabei sein und die mit Priorität drei wären schön wenn es die geben würde.

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

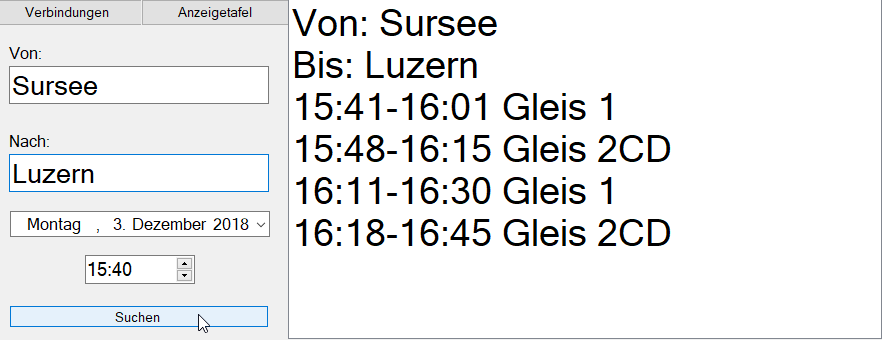
# Funktionen

## Station Textsuche

Mithilfe der Textsuche kann der Benutzer ein Teil des Station Namens eingeben. Dabei erhält er nachher Vorschläge für den gesamten Namen der Stationen. Um diese Funktion in der TextBox zu benutzen muss man mindestens Vier Zeichen eingegeben haben.  
Diese Funktion ist hilfreich, wenn man teile des Namens vergessen hatte.



## Verbindungen suchen

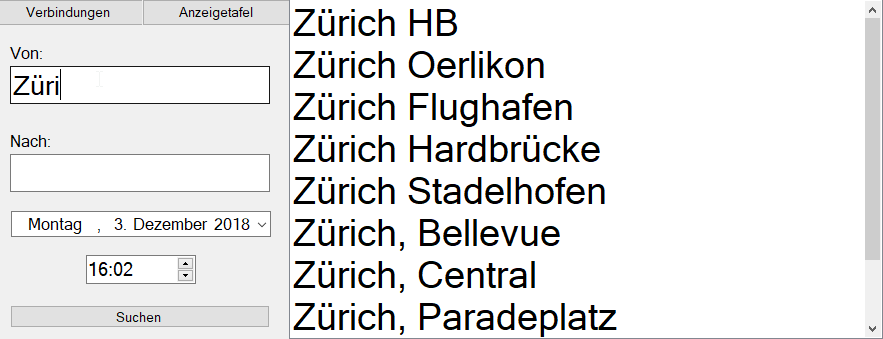
Der Benutzer kann Zwei Stationen in den TextBoxen angeben und erhält danach die nächsten Verbindungen. In den meisten Fällen werden die nächsten Vier Verbindungen angezeigt, dies kann aber variieren.  
In der Grafik unten sieht man die Verbindungen von Sursee nach Luzern.

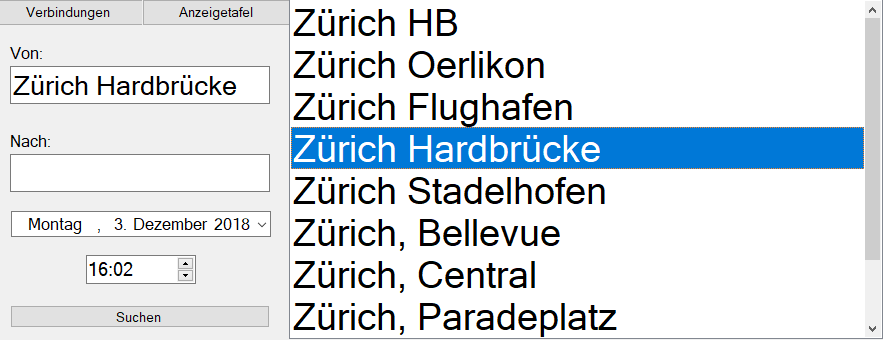
## Abfahrtstafel anzeigen

Mithilfe der Abfahrtstafel sieht der Benutzer die nächsten Verbindungen ab einer Bestimmten Station. Dies ist zu Vergleichen mit der Anzeige am Bahnhof. Um diese zu aktivieren, muss man Zuerst den Button "Anzeigetafel" anwählen, dann ein Text in die TextBox eingeben und zum Schluss den Button "Suchen" klicken.

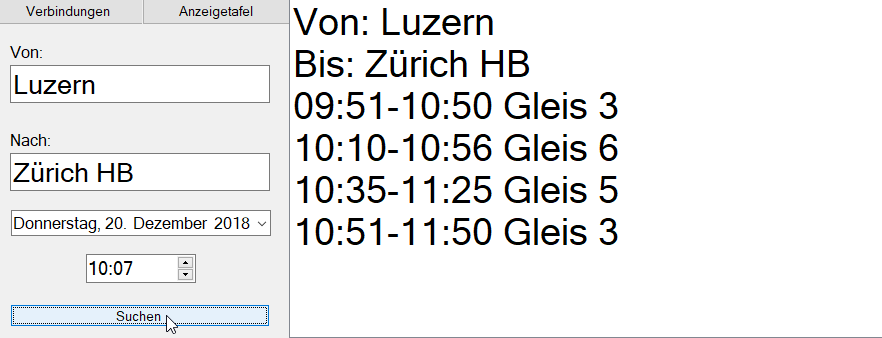


## Station Textsuche während dem Schreiben

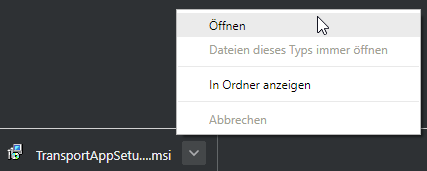
Nachdem man Vier Buchstaben eingegeben hat, wird rechts in der List Box Vorschläge angezeigt, für mögliche Station Namen (Kapitel 2.1).  
Sobald rechts in der List Box die Vorschläge angezeigt werden kann man die Taste "Enter" drücken und man springt in die List Box, wobei das erste Element Blau markiert wird. Nachher kann man mithilfe der Pfeiltaste zwischen den Vorschlägen hin und her wechseln. Sobald man den richtigen Eintrag selektiert hat, kann man die Taste "Enter" klicken und schon erscheint der Station Name Links in der zuvor ausgewählten TextBox.



## Verbindungen zu bestimmter Zeit suchen

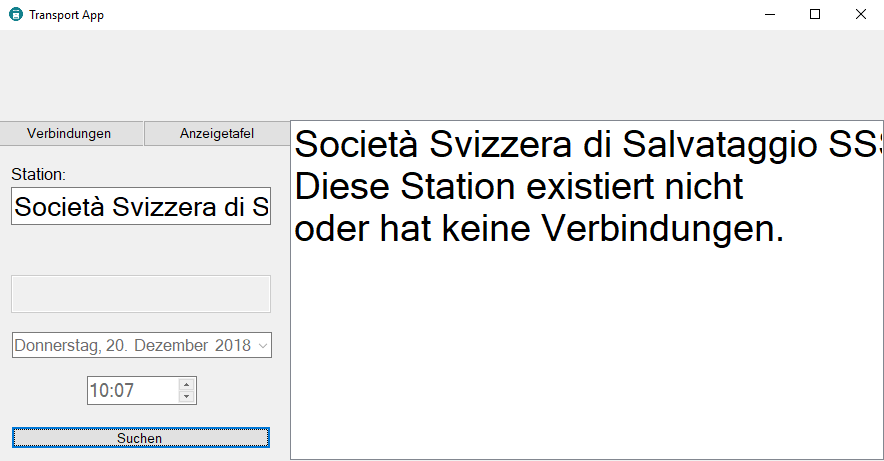
Beim Feature Verbindungen suchen (Kapitel 2.2) kann man jetzt noch ein bestimmtes Datum und eine bestimmte Zeit mitgeben.  
Dabei ist zu beachten, dass Ergebnisse gefunden werden, wenn das Datum in diesem, letztes oder nächstes Jahr ist. Sonst wird es meistens kein Ergebnis geliefert und der Benutzer erhält eine Information.   


# Installationsanleitung

Das Installieren des Programms kann man über folgenden Link machen:   
<https://github.com/02lars02/modul-318-student/raw/master/TransportAppSetup.msi>  
Danach öffnet sich der Browser und eine Installation startet.  
Nachdem die Installation abgeschlossen ist, kann man rechts auf den Pfeil klicken. Da öffnet sich ein Menü, wo man "Öffnen" auswählen muss.  
Danach startet sich der Installer. Es öffnet sich ein Fenster auf welchem alles beschrieben ist. Dieses Fenster wird den User durch die Installation leiten.  
Nachdem die Installation erfolgreich war, wird sich das Fenster schliessen und es sollte ein Shortcut auf Ihrem Desktop sehen.

# Probleme/Bugs

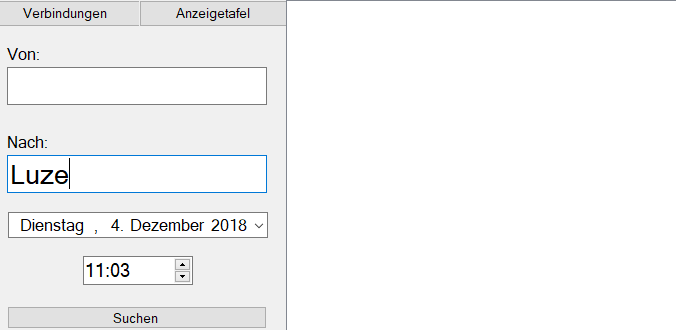
## GUI Responsiv

Das GUI verfügt über eine feste Grösse. Daher kann es vorkommen, dass Elemente, welche einen langen Namen haben, nicht vollumfänglich in der List Box ersichtlich sind.  
Dieser Fehler, wie unten im Screenshot zu sehen, kommt nur sehr selten vor und oft auch meistens nur, wenn man nach irgendeiner Firma sucht, wie man im Screenshot erkennen kann.  


## Performance

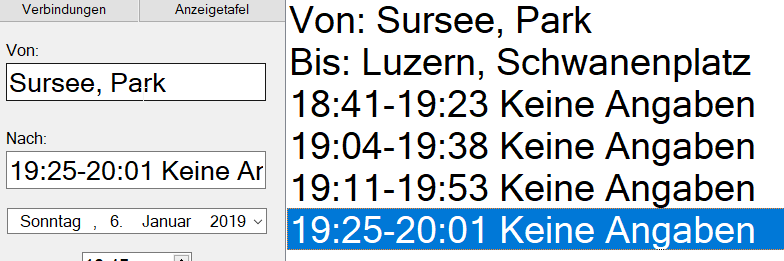
### Abfrage

Sobald man den vierten Buchstaben in ein Textfeld eingibt kann es dazu kommen, dass es zu einer kurzen Verzögerung kommen kann. Dies aufgrund der Abfrage der API zu diesem Zeitpunkt.



## List Box Elemente zu TextBox

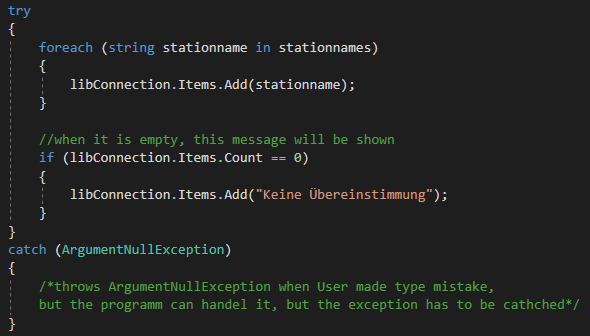
Ein weiterer Fehler im Programm ist, dass man ein Eintrag aus der List Box selektieren kann und dann mit der Taste "Enter" kann man den Text des Eintrages in einer TextBox anzeige lassen.



## Zugriff

### List Box befüllen ausserhalb des GUIs

Der Code-Block, unterhalb im Screenshot, kommt im Moment noch zweimal vor in der Klasse GUI. Nämlich innerhalb der Funktion "txtStartstation\_KeyDown" und "txtEndstation\_KeyDown".  
Den Code-Block kann man nicht in eine andere Klasse verschieben, da es den Control "libConnection" mit Items büffeln soll. Jedoch kann man nur innerhalb der Klasse GUI auf den Control zugreifen und da Logik innerhalb der Klasse des Guis nicht erwünscht ist, ist ein auslagern in eine Funktion nicht möglich.



## Arbeitsspeicher

Ein kleines Problem mit dem Arbeitsspeicher besteht, wenn man die Abfahrtstabelle ausführt.  
Sobald man die Abfahrtstablle aufruft, passiert es, dass die Arbeitsspeicher Nutzung um 0.4MB bis zu 1.2MB steigt. Wenn man dies einige Male ausführt, steigert sich der reservierte Arbeitsspeicher. Dies könnte zu Problemen führen bei Computer, welche über wenig Arbeitsspeicher verfügen.  
Unten sieht man ein Beispiel vor der Abfrage und nach der Abfrage.



# Planung

## Use Case

Ein use case ist ein Anwendungsbeispiel, wie das Programm später gebraucht werden könnte.

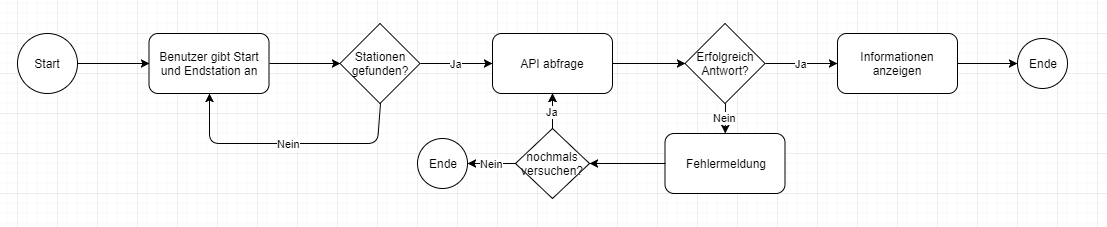
|  |  |
| --- | --- |
| Use Case | 1. Station suchen |
| Beschreibung | Der Benutzer kann nach Stationen Namen suchen und erhält dabei Vorschläge für ähnliche Station Namen. |
| Akteur | ÖV-Benutzer |
| Auslöser | Der Benutzer weiss den genauen Namen der Station nicht. |
| Vorbedingungen | Man benötigt eine Internetverbindung.  Programm muss installiert sein. |
| Ablauf | 1. Benutzer startet Programm 2. Benutzer gibt einige Anfangsbuchstaben einer Station an 3. Vorschläge werden angezeigt |
| Ergebnis | Der Benutzer erhält den genauen Namen der Station. |

|  |  |
| --- | --- |
| Use Case | 1. Verbindung zwischen Zwei Stationen suchen |
| Beschreibung | Der Benutzer möchte die nächsten Vier Verbindungen zwischen Zwei Stationen erhalten. |
| Akteur | ÖV-Benutzer |
| Auslöser | Der Benutzer möchte von einer Station zur nächsten fahren. |
| Vorbedingungen | Man benötigt eine Internetverbindung.  Programm muss installiert sein. |
| Ablauf | 1. Benutzer startet Programm 2. Benutzer gibt Startstation ein 3. Benutzer gibt Endstation ein 4. Benutzer betätigt einen Button 5. Informationen werden angezeigt |
| Ergebnis | Der Benutzer erhält die Information, wann der nächste Bus oder Zug, von einer bestimmten Station zu einer anderen, bestimmten Station fährt. |

|  |  |
| --- | --- |
| Use Case | 1. Anzeigetafel einer Station anzeigen |
| Beschreibung | Der Benutzer möchte alle Verbindungen ab einer gewissen Station angezeigt bekommen. |
| Akteur | ÖV-Benutzer |
| Auslöser | Der Benutzer möchte alle Verbindungen ab einer Station sehen. |
| Vorbedingungen | Man benötigt eine Internetverbindung.  Programm muss installiert sein. |
| Ablauf | 1. Benutzer startet Programm 2. Benutzer gibt eine Station ein 3. Benutzer betätigt einen Button 4. Informationen werden angezeigt |
| Ergebnis | Der Benutzer sieht alle Züge und/oder Busse, welche ab dieser Station fahren. |

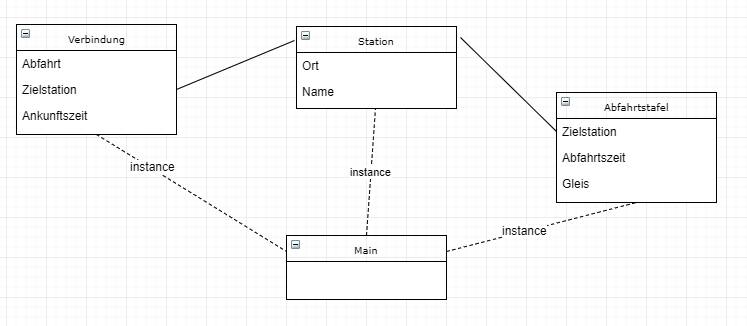
## Aktivitätsdiagramm

Unterhalb in der Grafik ist dargestellt, wie das Programm funktionieren soll, wenn der Benutzer nach einer Verbindung suchen will.



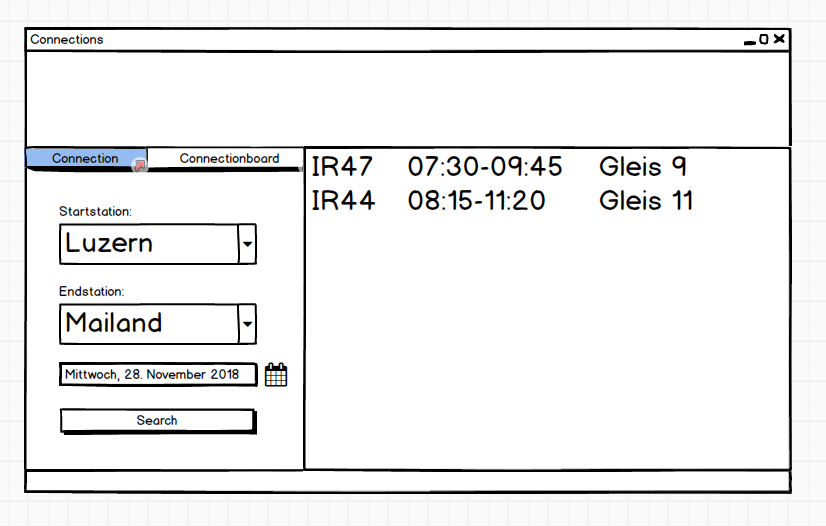
## Klassendiagramm

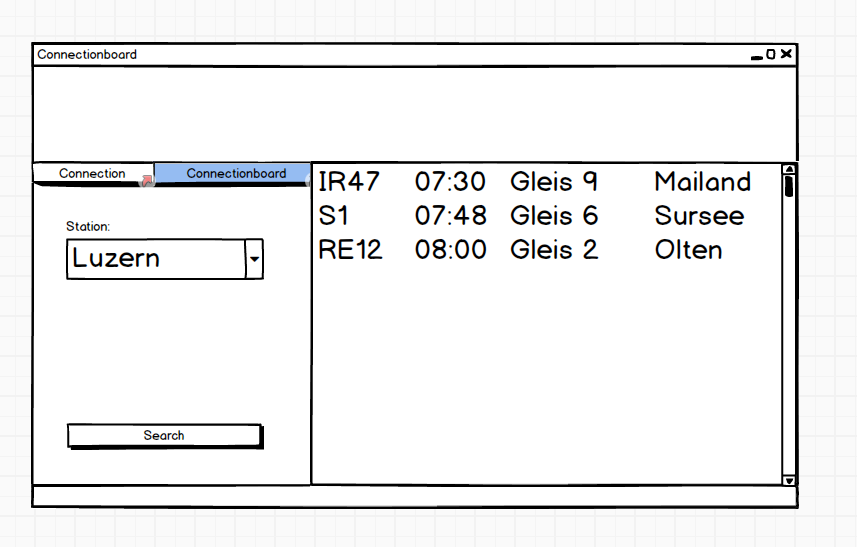
Das Klassen Diagramm soll darstellen, welche Klassen das Programm habe soll und wie diese miteinander verbunden sind.



## Mock-Up

Hier sieht man erste Entwürfe des GUIs. Diese haben mit dem Aussehen des Schlussendlichen Programm nichts zu tun. Diese dienen bloss als Anhaltspunkt.





# Testfälle

|  |  |
| --- | --- |
| Test | Ergebnis |
| Eine existente Station wird eingegeben.  Diese wird in den Vorschlägen angezeigt. |  |
| Einen Teil einer existenten Station wird eingegeben.  Es werden Vorschläge erscheinen. |  |
| Zwei existente Stationen werden eingegeben, welche Verbindungen haben. Die nächsten Verbindungen werden angezeigt. |  |
| Eine inexistente Startstation wird eingeben und eine existente Endstation werden eingeben. Eine Fehlermeldung soll erscheinen. |  |
| Eine existente Startstation wird ausgegeben und eine inexistente Endstation werden eingeben. Eine Fehlermeldung soll erscheinen. |  |
| Zwei inexistente Stationen werden eingegeben. Eine Fehlermeldung soll erscheinen. |  |
| Zwei Existente Stationen werden eingegeben. Danach wird nach einer Verbindung in der Vergangenheit gesucht. Die Verbindungen werden angezeigt oder es erscheint eine Fehlermeldung (siehe Screenshot), da das Datum zu weit entfernt liegt. |  |
| Zwei Existente Stationen werden eingegeben. Danach wird nach einer Verbindung in der Zukunft gesucht. Die Verbindungen werden angezeigt oder es erscheint eine Fehlermeldung (siehe Screenshot), da das Datum zu weit entfernt liegt. |  |
| Eine existierende Station wird angegeben. Die Abfahrtstafel erscheint. |  |
| Eine erfundene Station wird eingegeben. Fehlermeldung erscheint. |  |
| Der Button "Anzeigetafel" wird gedrückt. Nicht benötigte Controls werden disabled oder verschwinden. Nötige werden enabled. |  |
| Der Button "Verbindung" wird gedrückt. Nicht benötigte Controls werden disabled oder verschwinden. Nötige werden enabled. |  |

# Coding Guidelines

|  |  |
| --- | --- |
| Naming Conventions |  |
| Variablen sin klein zu schreiben und in camel case |  |
| Methoden werden grossgeschrieben und in camel case |  |
| Klassen werden grossgeschrieben und in camel case |  |
| Controls haben zuerst drei Buchstaben, welche der Control bestimmen, danach ein spezifischer Name, was der Control ist. |  |
| Declaration |  |
| Lokale Variablen, werden unterhalb des Methodenkopfes  Definiert. Deklariert werden diese erst sobald Sie gebraucht werden. |  |
| Globale Variablen, werden unterhalb des Klassenkopfes  Definiert. Deklariert werden diese erst sobald Sie gebraucht wird. |  |
| Comments |  |
| Kommentare werden jeweils oberhalb des Objekts geschrieben, welches es beschreibt. |  |
| Statements |  |
| Bei If-Else Statements werden die Kammern auf die nächste Zeile geschrieben. |  |
| Bei allen Schlaufen (z.B. For und Foreach) werden die Kammern auf die nächste Zeile geschrieben. |  |