**Inhaltsverzeichnis**

[Einleitung 3](#_Toc533107473)

[Planung 3](#_Toc533107474)

[Mockups 3](#_Toc533107475)

[Class Diagramm 5](#_Toc533107476)

[Aktivitätsdiagramm 5](#_Toc533107477)

[Anforderungen 6](#_Toc533107478)

[Richtlinien 7](#_Toc533107479)

[Bugs 7](#_Toc533107480)

[Such Funktion 7](#_Toc533107481)

[Tabstopps 7](#_Toc533107482)

[Google Maps 7](#_Toc533107483)

[Testing 7](#_Toc533107484)

[Testfall 1: Start und Endstation 7](#_Toc533107485)

[Testfall 2: Abfahrts plan 8](#_Toc533107486)

**Einleitung**

Am 19.12.18 habe ich vom ÜK Instruktor den Auftrag bekommen einen SBB Fahrplan Applikation in C# zu erstellen, welchen ich am Freitag dem 20.12.2018 abgeben muss. In diesem Dokument beschreibe ich meine Vorgehensweis zu den einzelnen Vorgaben.

**Planung**

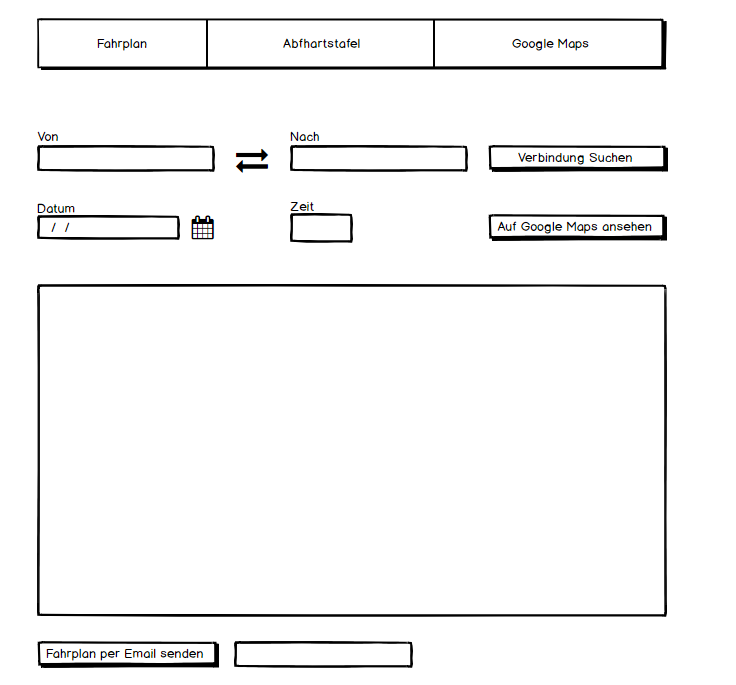
Bevor ich mit dem Projekt angefangen habe, musste ich zuerst ein paar Mockups erstellen, damit ich sehen kann wie die Applikation aussehen soll. Zudem habe ich zu den ersten drei Anforderungen ein Use Case und ein Aktivitätsdiagramm Diagramm erstellt.

Zweck der Dokumentation

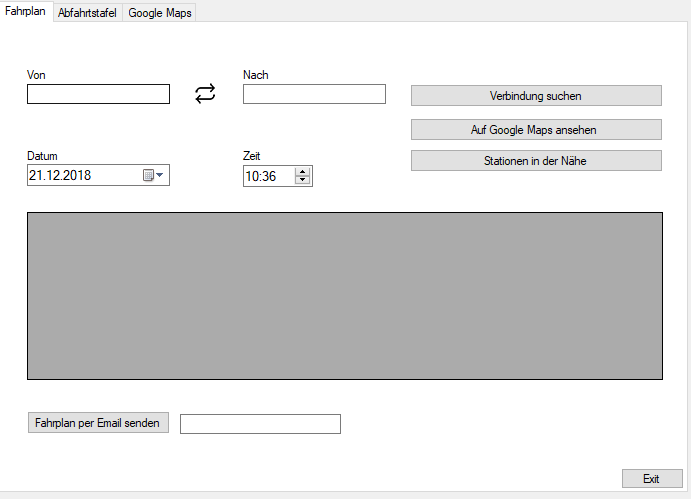
Dieses Dokument soll dazu dienen, dass man den Verlauf und die Entwicklung der Applikation nachvollziehen kann. Man sieht wie die Applikation strukturell aufgebaut ist. Der Hauptzweck dient dazu, Punkte für die Bewertung zu sammeln, da eine Dokumentation eine Anforderung ist.

**Mockups**

**Fahrplan Mockup**

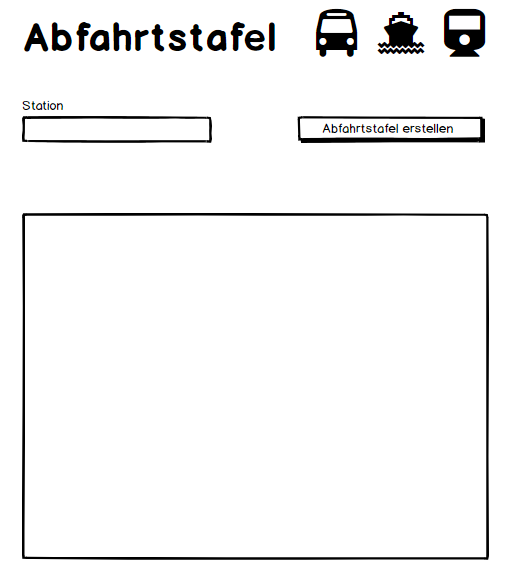


**Fahrplanansicht in der Applikation**



Dieses Layout hat sich nicht sehr stark verändert. Ich habe jedoch die Navigation mit einem Tab Controller umgesetzt und nicht mit Buttons. Ich habe auch noch zwei Buttons hinzugefügt, «Stationen in der Nähe» und «Exit».

**Abfahrtstafel Mockup**

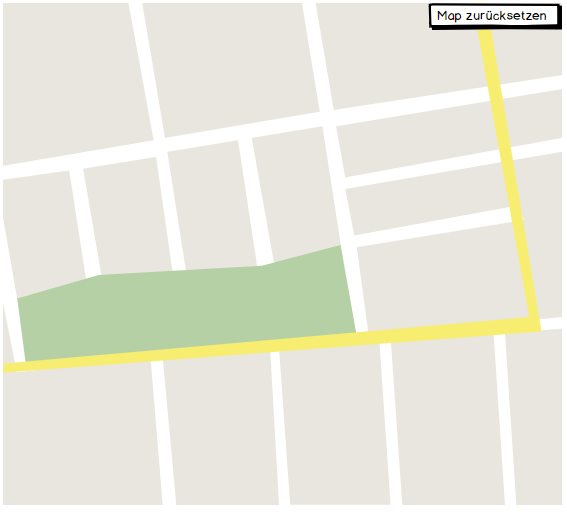


**Abfahrtstafelansicht in der Applikation**

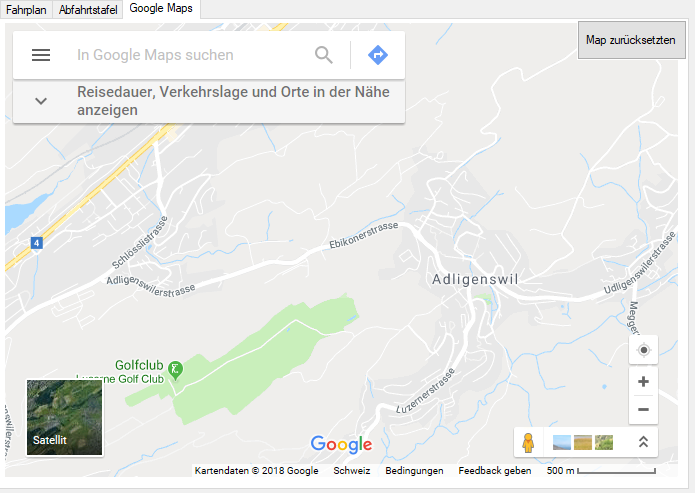


Bei der Abfahrtstafel musste ich das Schiff Symbol entfernen, da sonst das Bild zu gross wurde.

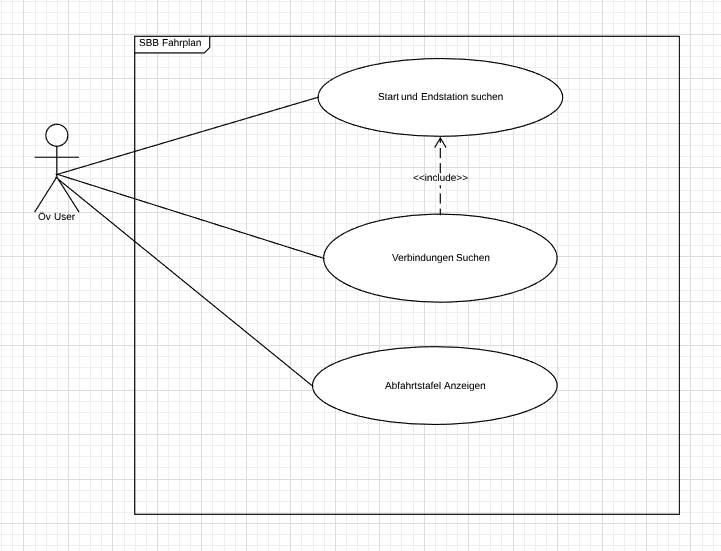
**Google Maps**

Bei der Google Maps anzeige habe ich noch zusätzlich eine Navigation hinzugefügt.

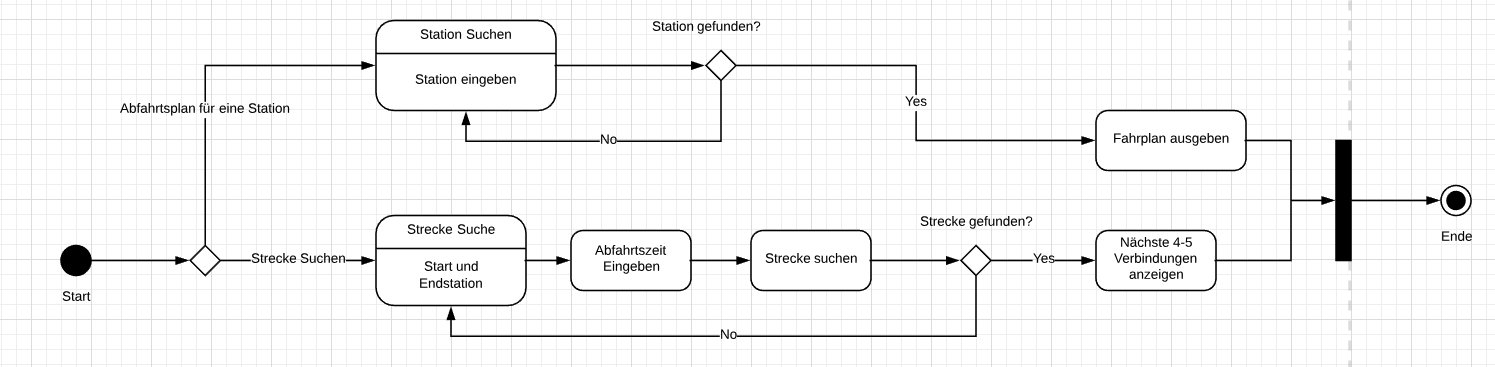
**Google Maps Anzeige in der Applikation**



**Use Case Diagram A001 – A003**



**Aktivitätsdiagramm A001 – A003**



**Richtlinien**

Diese definieren die Schreibweise von meinem Code bezogen auf Gross- / Kleinschreibung und Formatierung. Im Code möchten wir alles so benannt haben: präfixFunktion

Hierbei haben wir folgende Grundregeln:

1. Das Präfix besteht aus 3 Buchstaben welche das Element am besten beschreibt.

2. Das Präfix und die Funktion schreiben wir zusammen und der erste Buchstabe wird grossgeschrieben. z.B. txtVon

**Variablen**

Der Name der Variable hängt davon ab was sie macht z.B. **String ausgabe**

**Wo werden die Variablen definiert**

Dort wo sie gebraucht werden

**Wie werden Eigenschaften benennt**

Sie werden so benannt was sie machen sollten

**Methoden**

* Kurzes Präfix wo der Erste Buchstaben grossgeschrieben wird.
* Eine kurze Beschreibung erster Buchstaben wird Kleingeschrieben
* z.B. getGrid

**Klassen**

* Erster Buchstaben wird grossgeschrieben.

**Kommentare**

* Es werden nur die wichtigsten Sachen beschrieben.

**GUI-Controls**

* Kurzbeschreibung von Elementen z.B. txtBox
* Oder für den Button btnSuchen

**Anforderungen**

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

Bekannter Bug:

Der Webbrowser welcher in Visual Studio implementiert ist, ist der Internet Explorer, welcher sehr langsam ist. Wenn man die Station auf der Karte ansehen will gibt Google folgendes aus: «Anscheinend hast du die Kompatibilitätsansicht für Internet Explorer aktiviert. Google Maps funktioniert nur dann ordnungsgemäss, wenn die Ansicht deaktiviert ist. »

Workaround

Schritt 1

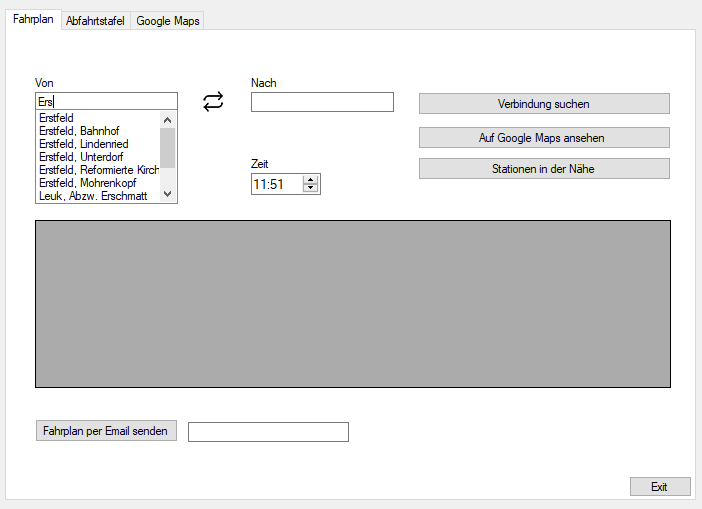
1. Starte den Internet Explorer welcher vorinstalliert ist.
2. Oben rechts auf folgendes Symbol drücken:
3. «Internetoptionen» öffnen
4. Unter «Allgemein» auf «Löschen» drücken
5. Alle Boxen auswählen
6. Anschliessend auf «Löschen» klicken

Schritt 2

1. Oben rechts auf folgendes Symbol drücken
2. «Einstellungen der Kompatibilitätsansicht» auswählen
3. «google.com» aus der List löschen
4. Beide Checkboxen müssen leer sein

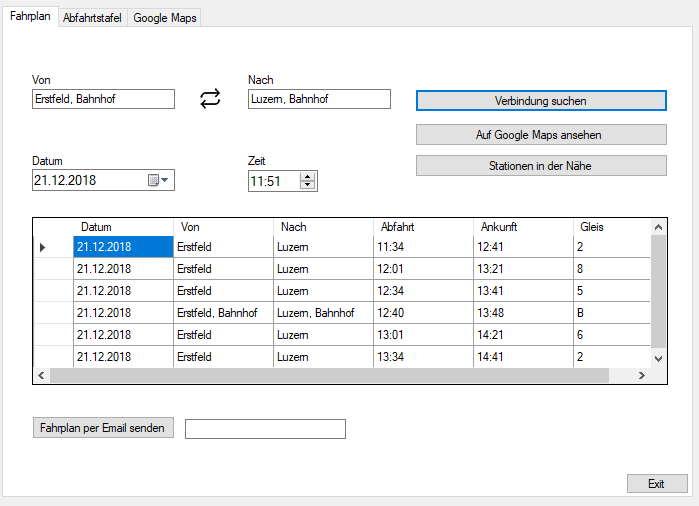
**Applikation im Betrieb**

**A001 + A004**



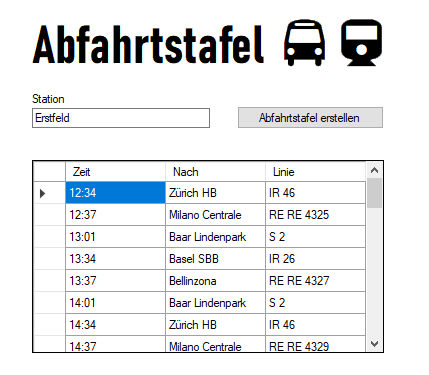
Station «Von» wird gesucht

**A002**

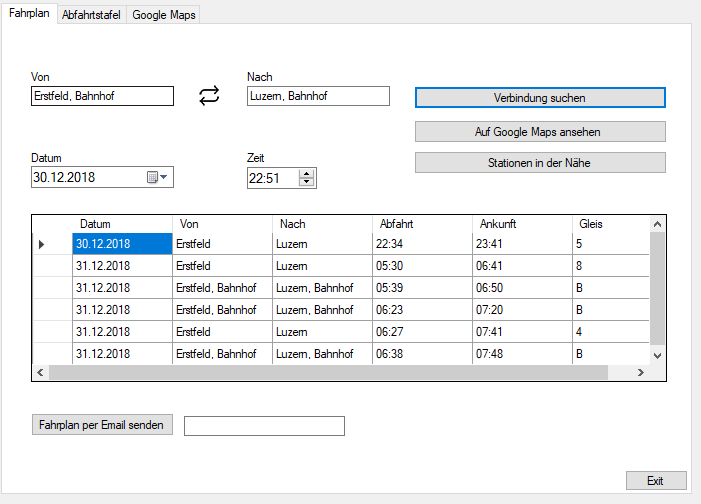


Fahrplan wurde ausgegeben

**A003**

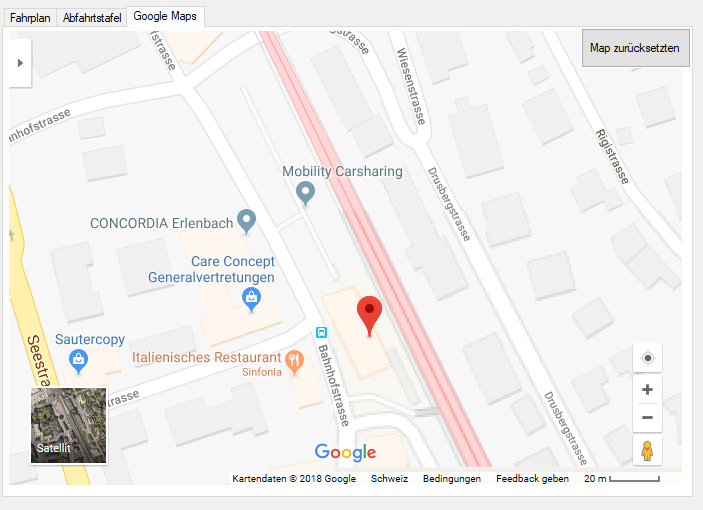


**A005**



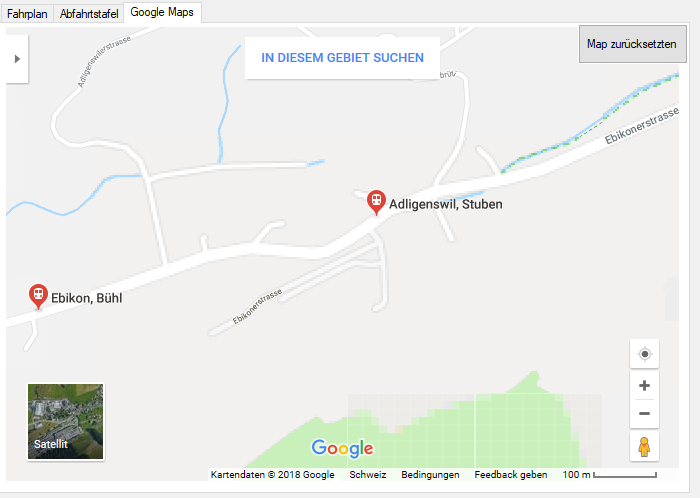
Datum wurde abgeändert

**A006**



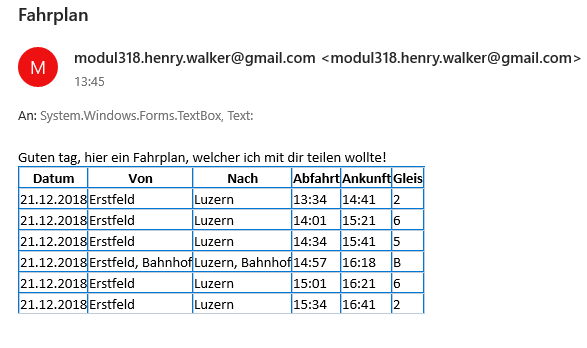
Station wird auf Google Maps gesucht

**A007**



Stationen in der Nähe werden auf Google Maps angezeigt

**A008**



So sieht der Fahrplan in der Email aus.

**Testing**

**Testfall 1: Start und Endstation**

A001-A002, A004-A005

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Interessent gibt in der TextBox «txtVon» «Ers» ein und drückt dann die Pfeiltasten nachunten, bis in der Listbox «lstVon» «Erstfeld» ausgewählt ist und drückt dann Enter. | Der Ort wird in die TextBox «txtVon» eingefügt und der Fokus geht automatisch zur TextBox Nach. |
| 2 | Interessent gibt in der TextBox «txtNach» «Luz» ein und drückt dann die Pfeiltasten nachunten, bis in der Listbox «lstNach» «Luzern» ausgewählt ist und drückt dann Enter. | Der Ort wird in die TextBox Nach eingefügt und der Fokus geht automatisch auf die Datumsauswahl |
| 3 | Interessent wählt das Datum aus, indem er auf das Kalender Icon mit der Maus klickt und wählt den 31.12.2018 | Das Datum «31.12.2018» sollte nun in der Datumsauswahl stehen. |
| 4 | Interessent klickt mit der Maus auf die Zeitauswahl. Er wählt die Zeit «Stunden = 10» «Minuten = 10» | In der Box sollte jetzt «10:10» stehen |
| 5 | Interessent Drückt auf den Knopf Verbindung suchen. | Im Grid «dgvAnzeigen» unten sollten jetzt 6 Verbindungen von Erstfeld nach Luzern stehen die am «31.12.2018 um 10:10 Uhr» abfahren. Die erste Verbindung die angezeigt wird, sollte auf Gleis 6 um «10:01 Uhr» Los Fahren. |

**Testfall 2: Abfahrtstafel**

A003

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Interessent drückt auf den Tab Page «Abfahrtstafel» Button | Man sollte nun das Layout von der Abfahrttafel sehen. |
| 2 | Interessent gibt bei der TextBox «txtStation» einen Ort ein. | Während der Interessent den Ort eingibt sollten schon Vorschläge für den Ort kommen. |
| 3 | Interessent drückt zwei Mal auf einen Ort in der Vorschlags liste. | Der Vorschlag wird in die TextBox Station eingefügt und der Fokus geht automatisch auf den Button Abfahrt. |
| 4 | Interessent drückt Enter. | Der Button Abfahrt wird gedrückt, im grid «dgvAbfahrtstafel» wird eine Liste von den nächsten Verbindungen vom angegebenen Ort angezeigt |

**Testfall 3: Station Google Maps**

A006

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Interessent gibt in der TextBox «txtVon» folgendes ein: Erstfeld  und drückt auf den Button «Auf Google Maps ansehen» | Die Koordinaten (46.820335,8.650443) von der Station «Erstfeld» wird in das Google Maps Suchfeld eingetragen.  Man kann nun den Bahnhof auf der Karte sehen. |

**Testfall 4: Stationen in der Nähe**

A007

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Interessent Drückt auf den Knopf Stationen in der Nähe | Es wird angezeigt wo er sich befindet und zeigt alle Stationen in der Nähe an. |

**Testfall 5: Email senden**

A008

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Der Interessent gibt folgende Daten im Fahrplan ein:   * Von: Erstfeld * Nach: Luzern * Datum: - * Zeit: - | Die Daten sollten jetzt in den verschiedenen TextBoxen stehen |
| 2 | Interessent drückt auf den Button Verbindung Suchen. | Es sollte unten im grid eine Tabelle öffnen wo der fahr plan der nächsten stunde von Erstfeld nach Luzern  Steht |
| 3 | Interessent gibt in der TextBox Email seine Email ein und Drückt auf den Button «Fahrplan per Email senden» | Es sollte eine E-Mail an die eingegebene Email gesendet werden mit den entsprechenden daten. |

**Using Test**

Dass der Test funktioniert hat musste ich das Datum und die Zeit noch hinzufügen.

public void Connections()

{

testee = new Transport();

var connections = testee.GetConnections("Sursee", "Luzern", "2018-07-08", "20:20");

Assert.IsNotNull(connections);

}

**Installationsanleitung**

Schritt 1: Laden Sie sich diese GIT-Repo hinunter und entpacken Sie es.