Dokumentation

Swiss Transport Applikation

Autor

Name: Dehghan

Vorname: Zanyar

Datum: 20.11.17

Inhaltsverzeichnis

* Einleitung -------------------------------------------------- 1
* Zweck des Dokumentes -------------------------------- 2
* Programmierrichtlinien -------------------------------- 3
* Funktionalität--------------------------------------------- 4
* Fehlende Funktionalität ------------------------------- 5
* Use Case Diagramm & Aktivitätsdiagram
  + A001 & A002 --------------------------------------6
  + A003 & A004 ------------------------------------- 7
  + A005 & A006 ------------------------------------- 8
  + A007 & A008 ------------------------------------ 9
* Testfälle ---------------------------------------------------- 10
* Installation ------------------------------------------------ 12

**Einleitung**

Die Applikation hat das Ziel, eine einfach und schnelle Zugang zu Swiss Transport Datenbank ermöglichen. Es wurde immer wieder versucht, dass die Benutzeroberfläche simple und freundlich sein, sodass mit heutige Applikationen vergleichbar wäre.

Es wurde mithilfe des C#, .Net Bibliotheken, HTML und Java Skript entwickelt. Die Applikation ist Windows basierend und dazu wird auf sogenannte Windows Forms ausgeführt .

**Zweck des Dokuments**

Das Ziel der folgenden Dokumentation ist, dass eine kurz, ausführlich und zusammenfassende Inhalt über Struktur und Funktionalität der Applikation bietet. Sowie auf eine bestimmte Art die wichtigsten Dinge präsentiert, welche eine dritte Entwickler wissen sollte.

Es wird versucht den Benutzer sowie Entwickler eine schnell und ausführliche Ausmalung der Applikation darzustellen.

**Programmierrichtlinien**

**Naming Conventions**

**- Variablen :** Camelsäule

**- Eigenschaften:** Es sind keine spezielle Eigenschaften vorhanden.

**- Mehtoden:**

Da die Name der Methoden selbst ihre Funktion beschreibt, wird hier nicht zusätzliche Wörter darüber geschrieben.

|  |
| --- |
| private void showConnections(string from, string to) |
| private void showStBoard(string from, string to) |
| public string getParams() |
| public void getStListe(string clName, double lat, double longt, string clText) |
| getMap() |
| getLatlnUseGPS() |
| getInfo(string ipAddress) |
| getLocUseIP(string which) |

* **Declaration** : Camelsäule
* **Comments:** Mit // wo es notwendig gewesen, wurde es geschrieben.
* **GUI-Controls**

|  |  |
| --- | --- |
| **Name** | **Type** |
| listResult | **Listbox** |
| txtFrom | Textbox |
| txtTo | Textbox |
| txtStBoard | Textbox |
| btnNearst | Button |
| btnMap | Button |
| btnRrefresh | Button |
| btnReset | Button |
| btnMail | Button |
| btnDtDec | Button |
| btnDtInc | Button |
| tabClTxt | TabPage |
| dtmDate | Datetimepicker |
| txtHour | NumericUpDown |
| wbBrowserMap | Webbrowser |
| wbBrowserDataTable | Webbrowser |
| rdBtnNoLimit | Radio Button |
| rdBtnLimit | Radio Button |
| lblFrom | Label |
| lblTo | Label |
| LblStation | Label |
| lblTitel | Label |
| Transport Namen mit Checkbox[mehrere] | Checkbox |

* **Statements**

If-else

For

foreach

**Funktionalität**

Die Funktionen der Applikationen sind teilweise Hardware-abhängig. Als Beispiel will man seine genaue Position auf eine Karte zeigen, benötigt man eine GPS USB-Stick oder ähnliche Gerät, welche erst installiert sein soll. Solche Anforderungen spielen eine wichtige Rolle für Erfüllung der Funktionalität.

Folgende Funktionen sind in Software Implementiert und funktionieren einwandfrei:

* Abfrage eine bestimmte Verbindung
* Abfrage eine Abfahrtstafel
* Positionen auf Karte anzeigen
* Nähere Stationen auflisten (Latitude, Längengrad)
* Die Ergebnisse via Email senden.

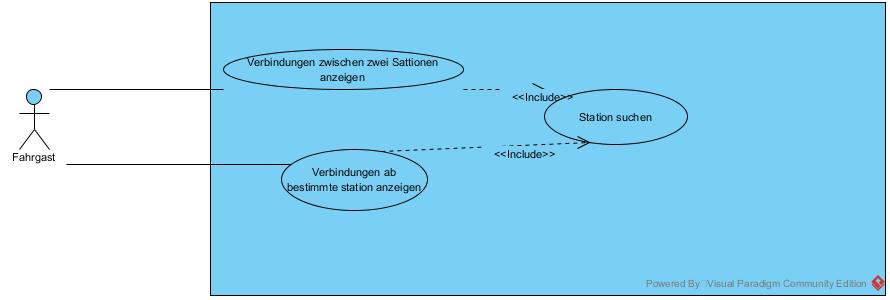
**Fehlende Funktionalität**

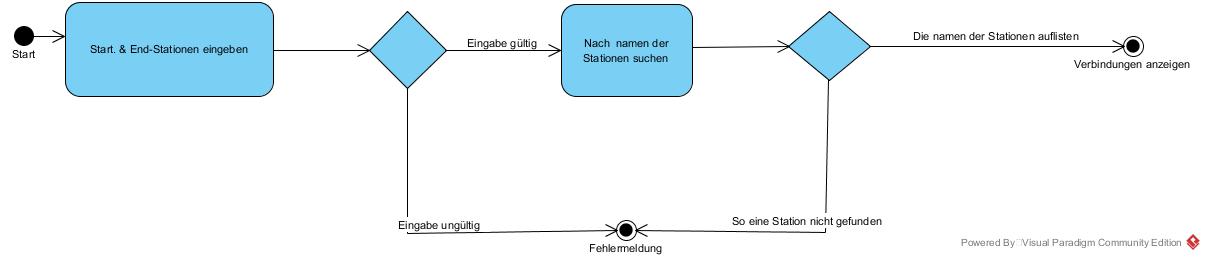
Alle gegebenen Anforderungen von A001 bis A008 sind erledigt und funktionieren problemlos. Eigentlich wenn der Benutzer ein GPS USB-Stick hat, wird das Programm am besten funktionieren.

**Use Case Diagramm**

**A001 bis A003**

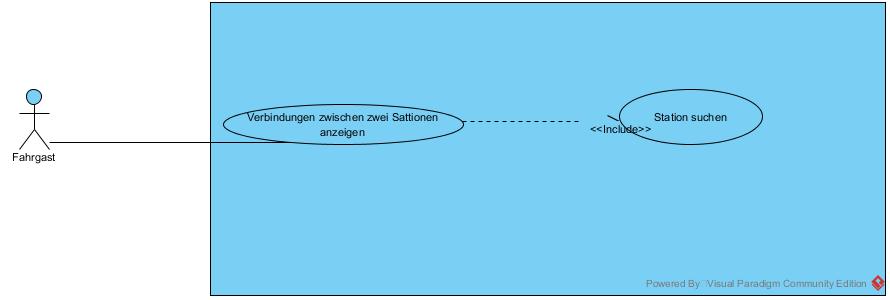
Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.

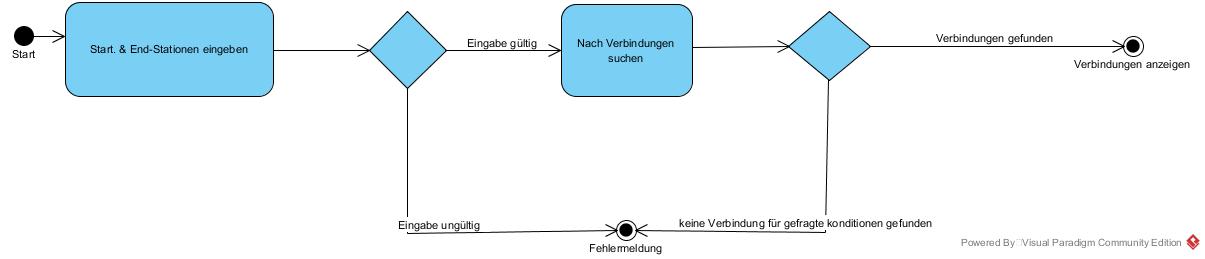




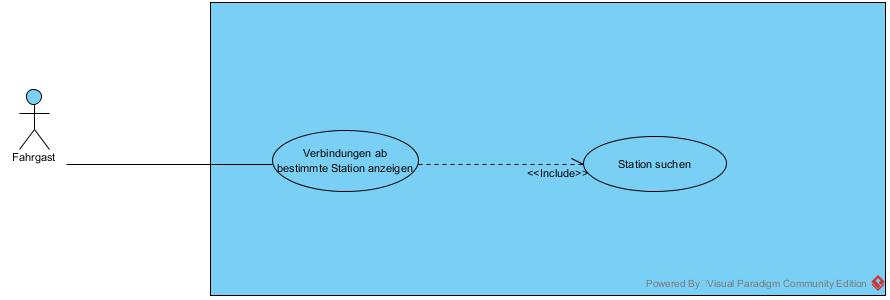
**A002**

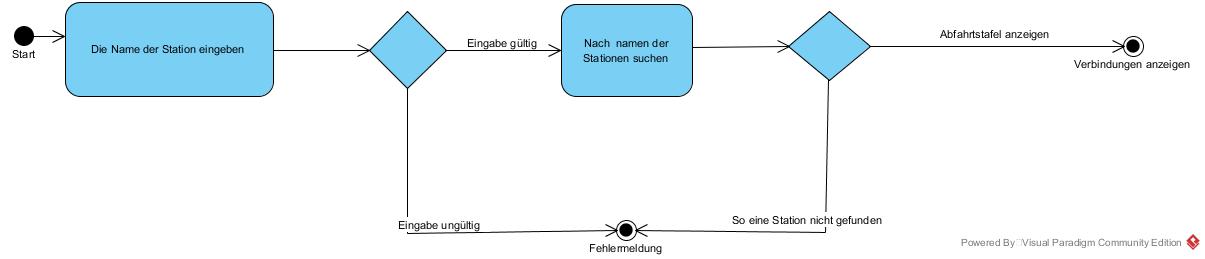
Als ÖV-Benutzer möchte ich die aktuellen, d.h. mindestens die nächsten vier bis fünf Verbindungen zwischen den beiden gefundenen und ausgewählten Stationen sehen, damit ich weiss wann ich zur Station muss, um den für mich idealen Anschluss zu erwischen.



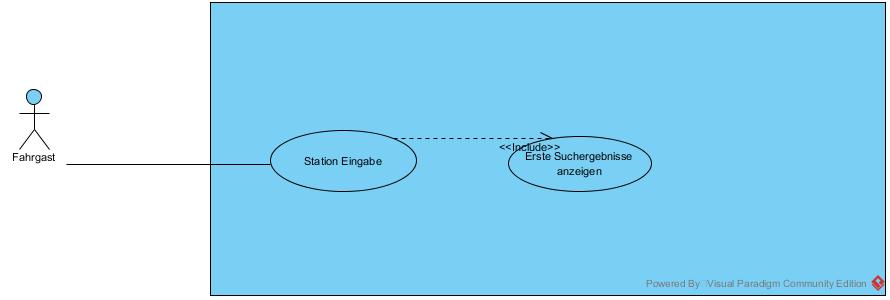
**A003**

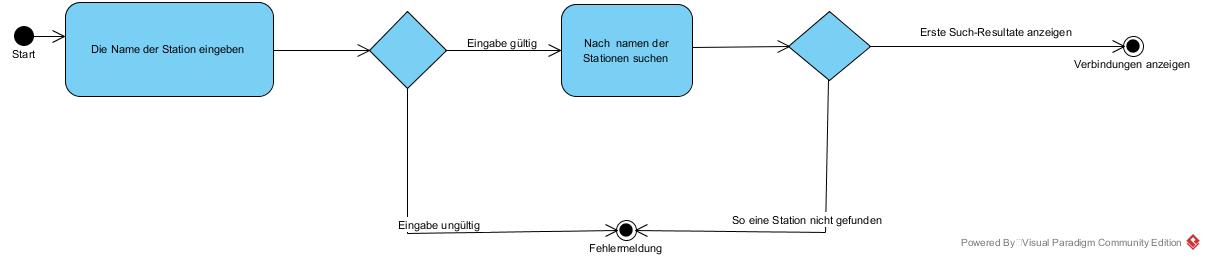
Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann.



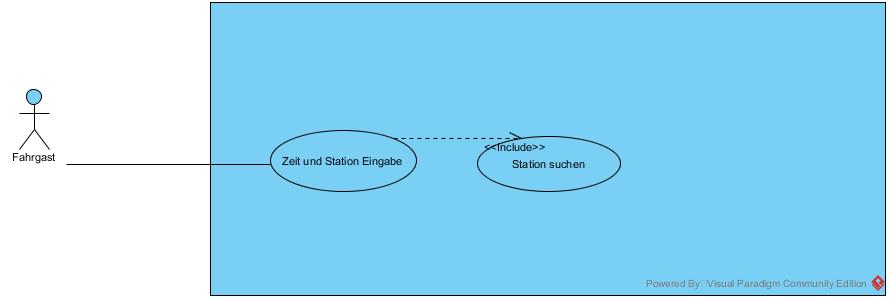
**A004**

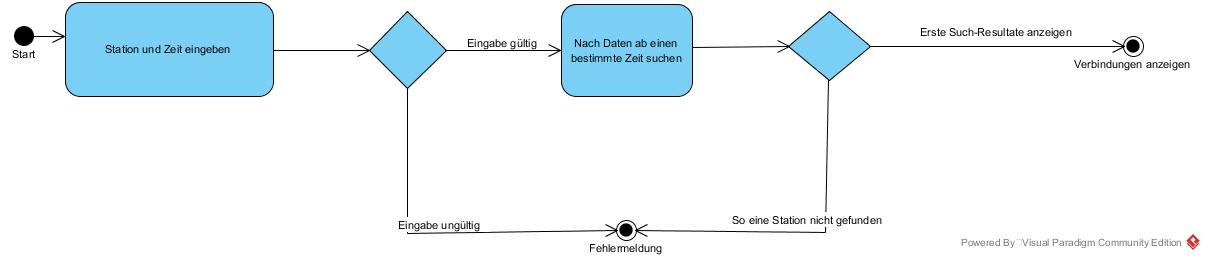
Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste SuchResultate erscheinen, damit ich effizienter nach Stationen suchen kann.



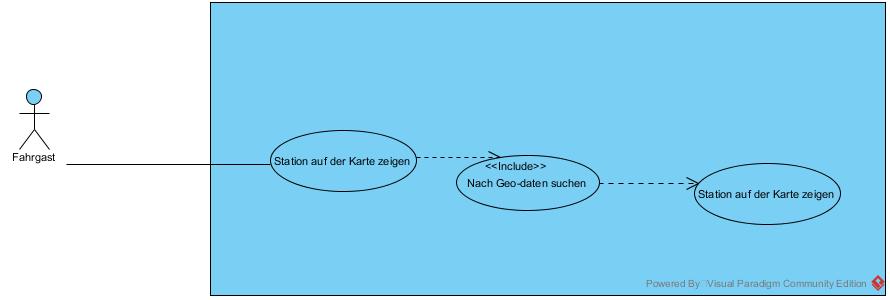
**A005**

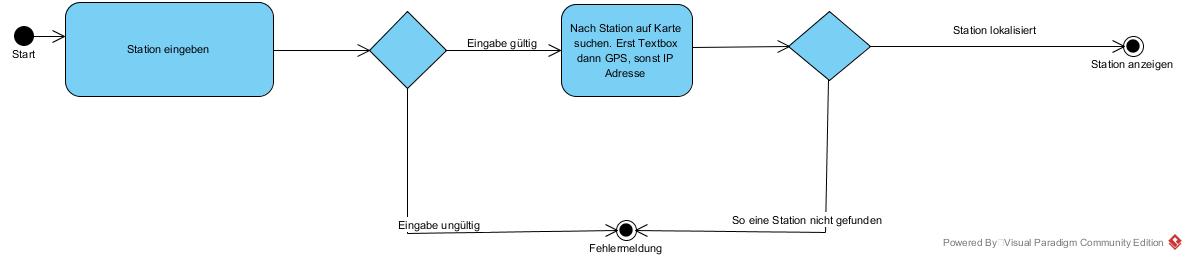
ALs ÖV-Benutzer möchte ich nicht nur aktuelle Verbindungen suchen können, sondern auch solche zu einem beliebigen anderen Zeitpunkt, damit ich zukünftige Reisen planen kann.



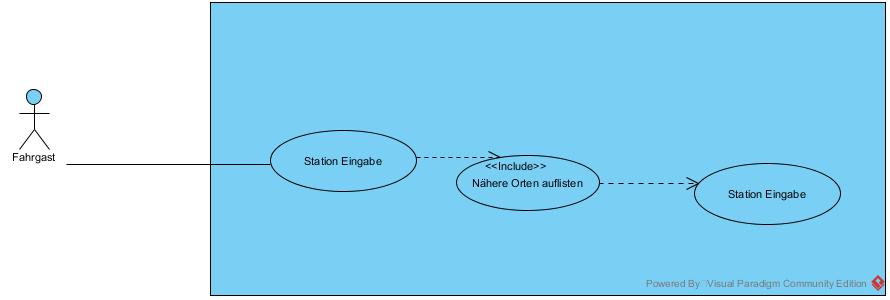
**A006**

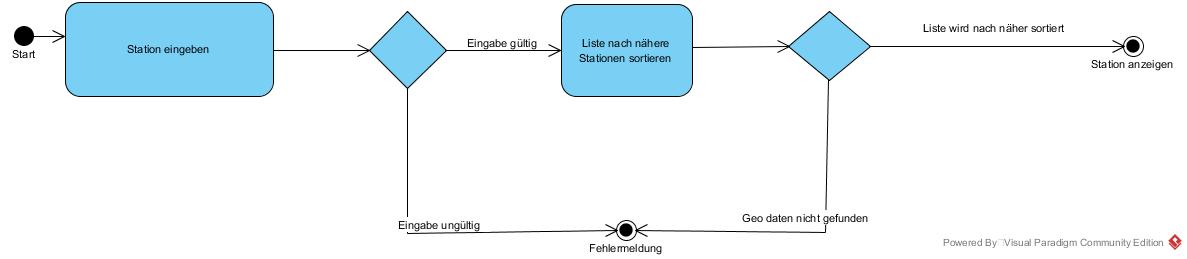
Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht.



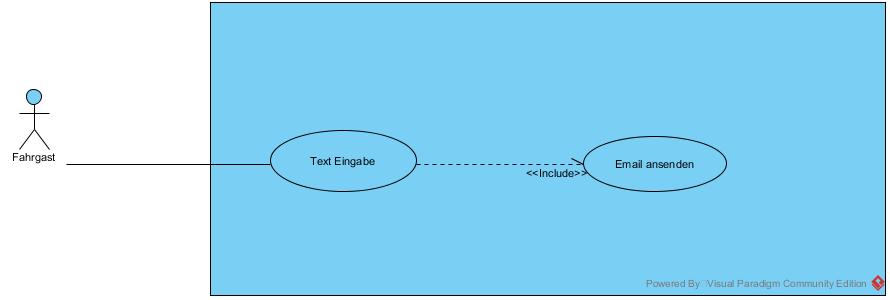
**A007**

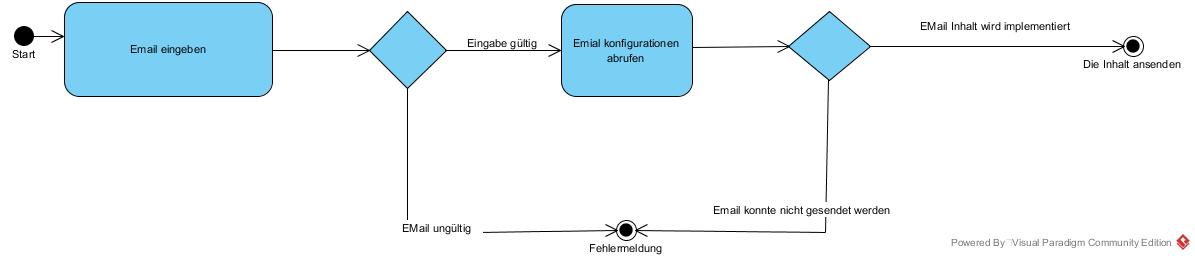
Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann.



**A008**

Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können.



**Testfall 1 "Start, End-Eingabe" A001**

**Testszenario**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Start-Station ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Benutzer gibt die End-Station ein. | Auflistung ähnliche und nähere Stationen |

**Testfall 2 "Verbindungen anzeigen" A002**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Start-Station ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Benutzer wählt Start-Station aus. | Es wird in Textbox ausgefüllt. |
| 3 | Benutzer gibt End-Station ein. | Auflistung ähnlich und nähere Stationen |
| 4 | Benutzer wählt End-Station aus. | Textbox wird ausgefüllt und mögliche Verbindungen werden angezeigt |

**Testfall 3 "Abfahrtstafel anzeigen" A003**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Station ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Benutzer wählt Station aus. | Es wird in Textbox ausgefüllt und Abfahrtstafel wird angezeigt. |

**Testfall 4 "Erste Suchergebnisse während Eingabe listen" A004**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Start-Station ein. | Auflistung Suchergebnisse sortiert nach nähere Stationen |
| 2 | Benutzer gibt End-Station ein. | Auflistung Suchergebnisse sortiert nach nähere Stationen |
| 3 | Auf der Abfahrtstafel-Tab die Station eingeben. | Auflistung ähnlich und nähere Stationen |

**Testfall 5 "Abfrage mit Datum und Zeit" A005**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Start-Station ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Benutzer wählt Start-Station aus. | Es wird in Textbox ausgefüllt. |
| 3 | Benutzer gibt End-Station ein. | Auflistung ähnlich und nähere Stationen |
| 4 | Benutzer wählt End-Station aus. | Textbox wird ausgefüllt. |
| 5 | Benutzer gibt das Datum und die Zeit ein. | mögliche Verbindungen werden angezeigt |

**Testfall 6 "Station auf der Karte anzeigen" A006**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Auf der Fahrplan gibt der Benutzer die Startstation ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Der Benutzer klickt auf Knopf "Karte" | Die Station wird gezeigt. |
| 3 | Auf der Abfahrtstafel-Tab gibt die Station-Name ein. | Auflistung ähnlich und nähere Stationen |
| 4 | Der Benutzer klickt auf Knopf "karte" | Die Station wird gezeigt. |

**Testfall 7 "Nähere Stationen finden" A007**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Start-Station ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Benutzer gibt End-Station ein. | Auflistung ähnlich und nähere Stationen |
| 3 | Auf Abfahrtstafel gibt der Benutzer die Station ein. | Auflistung ähnlich und nähere Stationen |

**Testfall 8 "Resultate via Mail weiterleiten" A008**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Benutzer gibt die Start-Station ein. | Auflistung ähnliche und nähere Stationen |
| 2 | Benutzer wählt Start-Station aus. | Es wird in Textbox ausgefüllt. |
| 3 | Benutzer gibt End-Station ein. | Auflistung ähnlich und nähere Stationen |
| 4 | Benutzer wählt End-Station aus. | Textbox wird ausgefüllt. |
| 5 | Benutzer gibt die Email Adresse in Textbox ein | Email wird gesendet und die Meldung "Es wurde gesendet erschient. |

**Installationsanleitung**

**Installieren**

Da Software ist Protabel ausführbar, das heisst falls man die Ordner auf eine Rechner kopiert und die .EXE Datei ausführt, wird es einwandrei ausgeführt. Dafür ist es entweder Setup Datei noch spezielle Einstellungen notwendig.

Es kann in Ordner EXE gefunden werden.

**Deinstallieren**

Für Deinstallation soll man nur die Inhalt der Portable Version löschen. Danach ist die Software gelöscht.