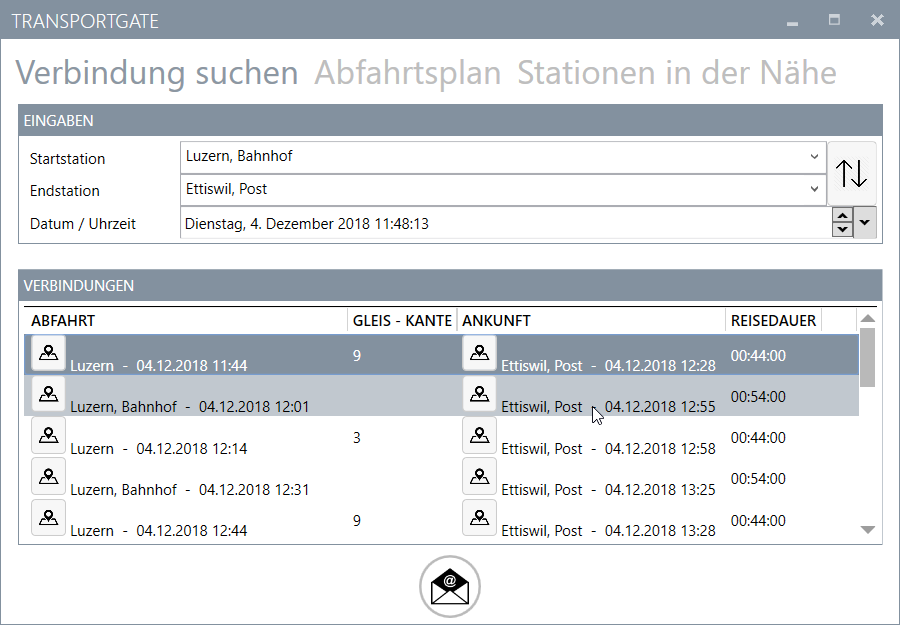
d

Projektarbeit Modul 318 – TransportGate



# Inhalt

[1 Inhalt 2](#_Toc531700938)

[2 Management Summary 3](#_Toc531700939)

[3 Zweck des Dokuments 4](#_Toc531700940)

[4 Umgesetzte Funktionen 5](#_Toc531700941)

[5 UML-Diagramme / Mockup 6](#_Toc531700942)

[5.1 Use Case-Diagramm 6](#_Toc531700943)

[5.2 Use Case-Beschreibungen 6](#_Toc531700944)

[5.3 Aktivitätsdiagramm 7](#_Toc531700945)

[5.4 Analyse-Klassendiagramm 7](#_Toc531700946)

[5.5 Mockups 8](#_Toc531700947)

[6 Programmierrichtlinien 9](#_Toc531700948)

[6.1 Naming Conventions 9](#_Toc531700949)

[6.2 Deklaration von Variablen 9](#_Toc531700950)

[6.3 Comments 9](#_Toc531700951)

[6.4 Geschweifte Klammern 10](#_Toc531700952)

[6.5 Verwendete Technologien 10](#_Toc531700953)

[6.6 Ordnerstruktur im Projekt 10](#_Toc531700954)

[7 Testfälle 11](#_Toc531700955)

[7.1 Testfall 01 – Verbindungen suchen 11](#_Toc531700956)

[7.2 Testfall 02 – Abfahrtsplan anzeigen 12](#_Toc531700957)

[7.3 Testfall 03 – Stationen in der Nähe anzeigen 13](#_Toc531700958)

[7.4 Testfall 04 – Mails versenden 14](#_Toc531700959)

[7.5 Testfall 05 – Stationen tauschen 15](#_Toc531700960)

[8 Installationsanleitung 16](#_Toc531700961)

[8.1 Installieren 16](#_Toc531700962)

[8.2 Deinstallieren 16](#_Toc531700963)

# Management Summary

Im Rahmen des überbetrieblichen Kurses «Modul 318 - Analysieren und objektbasiert programmieren» ist das Programm «TransportGate» entstanden. Der Zweck dieser Applikation ist es, aktuelle und zukünftige Verbindungen mit den öffentlichen Verkehrsmitteln der Schweiz abrufbar zu machen. Zusätzlich ist es möglich, in der Nähe gelegene anzuzeigen, damit eine möglichst bequeme Reise ohne grosse Recherche geplant und in Angriff genommen werden kann.

# Zweck des Dokuments

Dieses Dokument beinhaltet sämtliche Informationen, welche zur Analyse, Planung, Durchführung und Überprüfung des Programmes zusammengetragen wurden. Es soll dem Instruktor die Möglichkeit geben, eine genaue Bewertung abzugeben. Zusätzlich dient es als kleines Benutzerhandbuch zur Installation und zur effizienten Anwendung des Programmes.

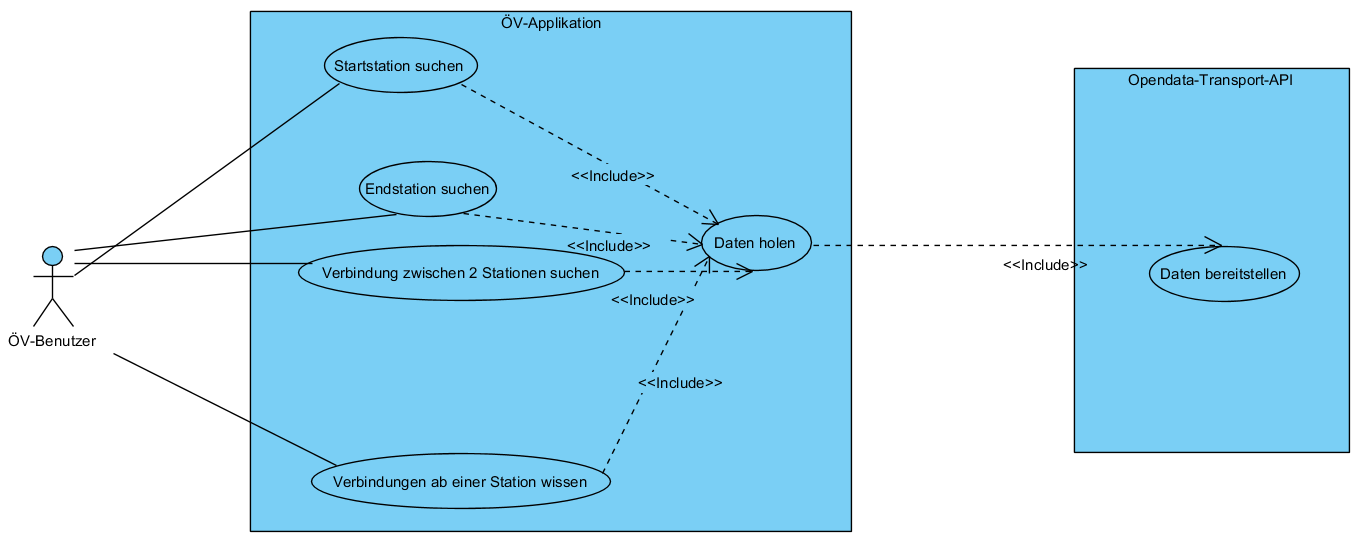
# Umgesetzte Funktionen

Der aktuelle Release 1.0.0 deckt die folgenden Anforderungen ab. Damit wären alle angeforderten Anforderungen implementiert:

|  |  |
| --- | --- |
| Anforderung | Beschreibung |
| 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. |
| Zusatzfeature: Start- und Endstation tauschen | Als ÖV-Benutzer möchte ich, nachdem ich die Start- und Endstation für die Suche nach Verbindungen eingegeben habe, die Stationen tauschen, damit ich sie nicht neu eingeben muss, wenn ich den Rückweg wissen will. |
| Zusatzfeature: Abfahrts- oder Ankunftszeit | Als ÖV-Benutzer möchte ich wählen, ob der eingegebene Zeitpunkt die Abfahrts- oder Ankunftszeit einer Buses / Zuges ist, wenn ich eine Verbindung suche. |
| Zusatzfeature: API-Calls asynchron refactorn | Als ÖV-Benutzer möchte ich nicht, dass das ganze Progrann einfriert und ich nichts mehr bewegen kann, wenn das Programm nach Stationen oder Verbindungen sucht. Ich möchte, dass die Abfragen im Hintergrund ablaufen. |

# UML-Diagramme / Mockup

## Use Case-Diagramm



## Use Case-Beschreibungen

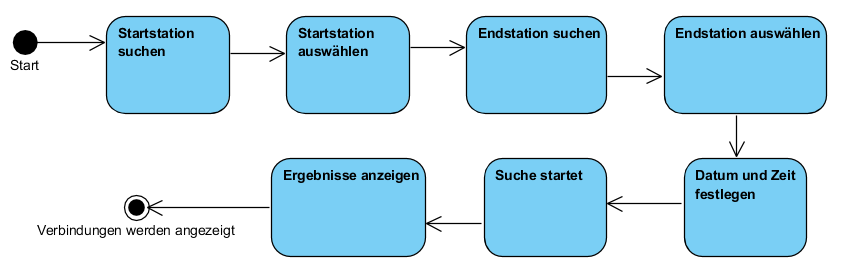
|  |  |
| --- | --- |
| Name | Station suchen |
| Beschreibung | Der Nutzer sucht mit dem ungefähren Namen einer Station nach einer Start- oder Endstation |
| Vorbedingungen | ÖV-App installiert, aktive Internetverbindung |
| Auslöser | Der Benutzer will eine Station auswählen. |
| Akteur | Nutzer der ÖV-App |
| Ablauf | 1. ÖV-App starten 2. Suchtext in Combobox eingeben 3. Gewünschte Station auswählen |
| Varianten | #1: Nach Startstation suchen  #2: Nach Endstation suchen |
| Ergebnis | Gewünschte Station wurde bei der Suche gefunden und ist nach Auswahl ausgewählt |

|  |  |
| --- | --- |
| Name | Verbindung zwischen 2 Stationen suchen |
| Beschreibung | Der Nutzer sucht mit bekannten Start- und Endstationen Verbindungen zwischen den Stationen |
| Vorbedingungen | ÖV-App installiert, aktive Internetverbindung |
| Auslöser | Der Benutzer will Verbindungen zwischen Stationen ab einem bestimmten Zeitpunkt wissen. |
| Akteur | Nutzer der ÖV-App |
| Ablauf | 1. ÖV-App starten 2. Start- und Endstation auswählen (siehe UseCase «Station suchen») 3. Zeitpunkt, ab dem die Verbindungen angezeigt werden sollen, festlegen |
| Varianten | - |
| Ergebnis | Die nächsten 16 Verbindungen von der Start- zur Endstation werden automatisch angezeigt. |

|  |  |
| --- | --- |
| Name | Verbindungen ab einer Station wissen |
| Beschreibung | Der Nutzer will alle Abfahrten von einer bestimmten Station wissen (=Abfahrtstafel). |
| Vorbedingungen | ÖV-App installiert, aktive Internetverbindung |
| Auslöser | Der Benutzer will alle Abfahrten von einer Station ab einem bestimmten Zeitpunkt wissen |
| Akteur | Nutzer der ÖV-App |
| Ablauf | 1. ÖV-App starten 2. Auf den Reiter «Abfahrtsplan» gehen 3. Startstation auswählen und Zeitpunkt (siehe UseCase «Station suchen») |
| Varianten | - |
| Ergebnis | Alle ab der ausgewählten Station werden angezeigt. |

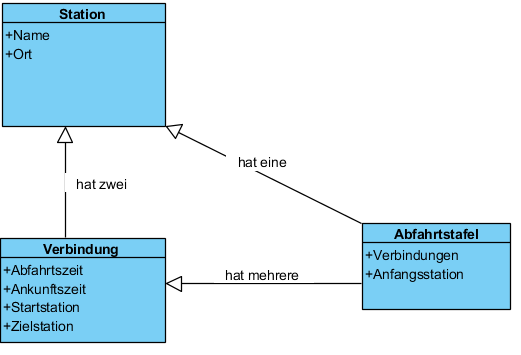
## Aktivitätsdiagramm

Das folgende Aktivitätsdiagramm ist für die Anforderung *A002*.

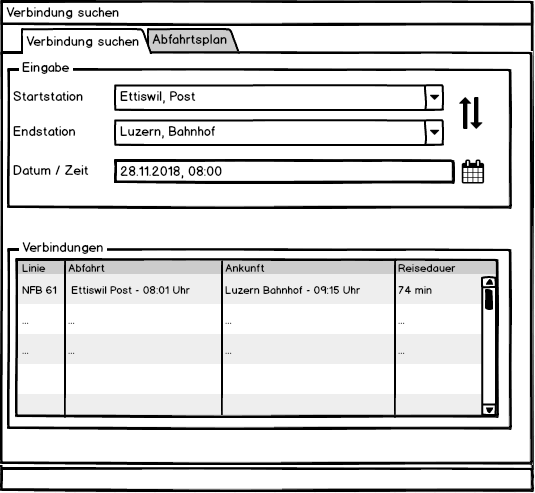
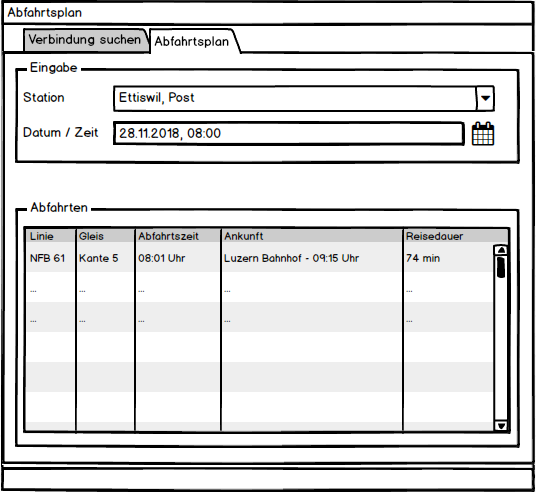


## Analyse-Klassendiagramm

Bevor mit der Implementation begonnen wurde, wurde ein Klassendiagramm auf Basis der von uns getätigten Analyse der benötigten Daten angefertigt. Das folgende Klassendiagramm stellt daher nicht die Situation in der fertigen Applikation dar und sollte nur zu Überlegungszwecken dienen.



## Mockups

Die folgenden Mockups wurden vor dem Beginn der Implementation erstellt und stellen nicht zwingenderweise das herausgekommene Resultat dar.

# Programmierrichtlinien

## Naming Conventions

Grundsätzlich werden alle Variablen, Klassen und Methoden in Englisch benannt.

### Variablen

Lokalen Variablen werden in Camel-Case geschrieben (thisVariable).

Globale Variablen (Membervariablen) werden zusätzlich mit dem Präfix «m\_» (m\_thisVariable) geschrieben.

### Properties

Properties werden in Pascal-Case geschrieben (ThisProperty).

### Methoden

Methoden werden in Pascal-Case geschrieben (ThisMethod).

Methoden sollen, wenn möglich nach dem Schema «Verben + Nomen» benannt werden (z.B. GetStation, SetConnenction).

### Klassen

Klassen werden in Pascal-Case geschrieben (ThisClass).

Klassen sollen, wenn möglich als Name ein Nomen haben (z.B. Station, Connection).

### GUI-Controls

Der Name der GUI-Controls besteht jeweils aus einem dreistelligen Präfix, welcher den Typ des Control abkürzt, und dem eigentlichen Namen. Er wird Camel-Case geschrieben (z.B. btnSearch, cmbStartStation).

## Deklaration von Variablen

### Lokale Variablen

Lokale Variablen werden dann definiert, wenn diese benötigt werden. Sie müssen nicht direkt initialisiert werden.

### Globale Variablen

Globale Variablen werden immer zu Beginn der Klasse (zu oberst) definiert. Sie müssen nicht direkt initialisiert werden.

## Comments

Alle Methoden, die public sind, müssen mit einem XML-Kommentar beschrieben werden. Bei Privaten Methoden ist es nicht zwingend und liegt im Ermessen des Entwicklers.

Innerhalb von Methoden werden Kommentare bei Notwendigkeit eine Zeile über dem zu kommentierenden Block geschrieben.

Der Inhalt der Kommentare beschreibt, was nachfolgend passiert. Es soll jedoch nicht «übersetzt» werden, sondern Zusatzinfos für die Verständlichkeit geliefert werden.

## Geschweifte Klammern

Nach If-, For-, Foreach-, While- und Switch-Statements wird die geschweiften Klammer auf eine neue Linie geschrieben.

Folgt nach dem Statement nur eine Linie Code, können die geschweiften Klammern gänzlich weggelassen werden.

## Verwendete Technologien

Es wird in der Sprache C# Version 7.2 und dem DotNet-Framework 4.7.2 entwickelt. Das GUI wird mithilfe dem Framework Windows Presentation Foundation (WPF) entwickelt. Es wird dabei das MVVM (Model-View-ViewModel) Pattern angewandt.

## Ordnerstruktur im Projekt

Im Projekt existieren folgende Ordner mit den beschriebenen Inhalten:

* Model: Enthält die Models (Data)
* ViewModel: Enthält die ViewModel
* View: Enthält die Views (XAML-Dateien mit den dazugehörigen Code-Behind Dateien)
* Helper: Klassen und Methoden, die generelle Aufgaben erledigen auf Hilfeservice bereitstellen.
* Ressources: Sämtliche Dateien, die kein Programmcode enthalten (z.B. Icons, externe Libraries).

# Testfälle

## Testfall 01 – Verbindungen suchen

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Verbindung suchen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste das Eingabefeld «Startstation» auswählen | Das Eingabefeld «Startstation» ist fokussiert.  Der Cursor blinkt darin |
| 2 | Den folgenden Text eintippen: «Ettiswil» | Im Eingabefeld «Startstation» steht der Text «Ettiswil» und das DropDown wird automatisch aufgeklappt.  Die ersten drei Resultate der Suche sind:   * Ettiswil, Post * Ettiswil, Schloss Wyher * Ettiswil, Oberdorf |
| 3 | Mit der Maus den ersten Eintrag «Ettiswil, Post» anwählen | Das DropDown wird geschlossen und in der Textbox steht der Text «Ettiswil, Post» |
| 4 | Analog zu Schritt 1&2 beim Eingabefeld «Endstation» durchführen | Im Eingabefeld «Startstation» steht der Text «Ettiswil» und das DropDown wird automatisch aufgeklappt.  Die ersten drei Resultate der Suche sind:   * Ettiswil, Schloss Wyher * Ettiswil, Oberdorf * Ettiswil, Surseestrasse |
| 5 | Mit der Maus den ersten Eintrag «Ettiswil, Schloss Wyher» anwählen | Das DropDown wird geschlossen und in der Textbox steht der Text «Ettiswil, Schloss Wyher».  Unten werden bei «Verbindungen» werden die nächsten Verbindungen angezeigt. |

## Testfall 02 – Abfahrtsplan anzeigen

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Verbindung suchen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste den Tab «Abfahrtsplan» selektieren | Der Tab «Abfahrtsplan» wird angezeigt.  Das Eingabefeld «Station» ist ausgewählt, der Cursor blinkt darin und das DropDown öffnet sich darin |
| 2 | Den folgenden Text eintippen: «Ettiswil» | Im Eingabefeld «Station» steht der Text «Ettiswil».  Die ersten drei Resultate der Suche sind:   * Ettiswil, Post * Ettiswil, Schloss Wyher * Ettiswil, Oberdorf |
| 3 | Mit der Maus den ersten Eintrag «Ettiswil, Post» anwählen | Das DropDown wird geschlossen und in der Textbox steht der Text «Ettiswil, Post».  Unten werden bei «Abfahrten» werden die nächsten Abfahrten zu verschiedenen Reisezielen angezeigt. |

## Testfall 03 – Stationen in der Nähe anzeigen

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Verbindung suchen» ist aktiv. Es besteht eine Internetverbindung. Der Positionsdienst ist in den Windows-Einstellungen deaktiviert.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste den Tab «Stationen in der Nähe» selektieren | Der Tab «Stationen in der Nähe»» wird angezeigt. |
| 2 | Auf den Knopf «Nach Stationen in der Nähe suchen» klicken | Es erscheint eine Warnung, dass die Positionierung fehlgeschlagen hat und dass man die Positionsdienste einschalten soll |
| 3 | In den Windows Einstellungen den Positionsdienst aktivieren.  Auf den Knopf «Nach Stationen in der Nähe suchen» klicken | Es erscheint keine Warnung (falls trotzdem eine Warnung erscheint konnte der Positionsdienst die aktuelle Position noch nicht bestimmen. Daher ein paar Sekunden warten und dann wiederholt versuchen.).  Im Abschnitt «Koordinaten» werden die gefundenen Koordinaten dargestellt (Korrektheit der Koordinaten bei Google-Maps überprüfen).  Im Abschnitt «Stationen» werden die nahegelegenen Stationen angezeigt |
| 4 | Im Eingabefeld «Manuell Station in der Nähe auswählen» nach «Ettiswil» suchen und das erste Resultat auswählen | Im Eingabefeld wird «Ettiswil, Post» angezeigt.  Im Abschnitt «Koordinaten» werden folgende Werte angezeigt:   * X-Koordinate: 47.147854 * Y-Koordinate: 8.01816   Die ersten drei Stationen im Abschnitt «Stationen» lauten:   * Ettiswil, Hinterdorf * Ettiswil, Oberdorf * Ettiswil, Surseestrasse |

## Testfall 04 – Mails versenden

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Verbindung suchen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Analog zum Testfall 01 die Verbindungen anzeigen lassen. | Es werden Verbindungen angezeigt. |
| 2 | Auf den Knopf unten in der Mitte (mit dem Mail-Symbol) klicken.  Gegebenenfalls das gewünschte Standard-Mail Programm selektieren. | Ihr Standard-Mail Programm öffnet sich und eine neue E-Mail wurde erstelle. Dabei sind folgende Werte gesetzt:   * Von-Adresse: [resultate@transportgate.ch](mailto:resultate@transportgate.ch) * Betreff: Resultate von TransportGate * Body-Text: Tabelle mit den im Programm ersichtlichen Verbindungen |
| 3 | In TransportGate analog zum Testfall 02 den Abfahrtsplan anzeigen lassen | Es werden Abfahrten von der gewählten Station angezeigt. |
| 4 | Auf den Knopf unten in der Mitte (mit dem Mail-Symbol) klicken. | Ihr Standard-Mail Programm öffnet sich und eine neue E-Mail wurde erstelle. Dabei sind folgende Werte gesetzt:   * Von-Adresse: [resultate@transportgate.ch](mailto:resultate@transportgate.ch) * Betreff: Resultate von TransportGate * Body-Text: Tabelle mit den im Programm ersichtlichen Abfahrten |
| 5 | In TransportGate analog zum Testfall 03 den Stationen in der Nähe anzeigen lassen | Es werden die Stationen in der Nähe angezeigt. |
| 6 | Auf den Knopf unten in der Mitte (mit dem Mail-Symbol) klicken. | Ihr Standard-Mail Programm öffnet sich und eine neue E-Mail wurde erstelle. Dabei sind folgende Werte gesetzt:   * Von-Adresse: [resultate@transportgate.ch](mailto:resultate@transportgate.ch) * Betreff: Resultate von TransportGate * Body-Text: Tabelle mit den im Programm ersichtlichen Abfahrten |

## Testfall 05 – Stationen tauschen

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Verbindung suchen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Analog zum Testfall 01 die Verbindungen anzeigen lassen. | Es werden Verbindungen angezeigt. |
| 2 | Auf den Knopf mit den zwei Pfeilen rechts der Eingabeboxen klicken. | Die Start- und Stoppstation werden miteinander getauscht.  Die Verbindungen werden neu berechnet und die Abfahrt und Ankunft werden neu gesetzt. |
| 3 | Mit der Maus den ersten Eintrag «Ettiswil, Post» anwählen | Das DropDown wird geschlossen und in der Textbox steht der Text «Ettiswil, Post».  Unten werden bei «Abfahrten» werden die nächsten Abfahrten zu verschiedenen Reisezielen angezeigt. |

# Installationsanleitung

## Installieren

|  |  |
| --- | --- |
| Schritt | Beschreibung |
| 1 | Das folgende Git-Repository auf den lokalen Rechner clonen: <https://github.com/Davee02/modul-318-daho> |
| 2 | Relativ zum Root-Verzeichnis des geclonten Repos in den folgenden Ordner navigieren: «src/SwissTransport.Installer/SwissTransport.Installer-SetupFiles/» navigieren |
| 3 | Die MSI-Datei doppelklicken und alle vom Wizard abgefragten Information eingeben oder validieren |
| 4 | Das Programm «TransportGate» kann, wenn die dazugehörige Option im Wizard angewählt wurde, über das Windows-Startmenü oder über das Shortcut auf dem Desktop geöffnet werden. |

## Deinstallieren

|  |  |
| --- | --- |
| Schritt | Beschreibung |
| 1 | Im Windows-Startmenü nach «Apps & Features» suchen |
| 2 | Nach «TransportGate» suchen und die Applikation anwählen |
| 3 | Den Button «Deinstallieren» betätigen und die Nachfrage bestätigen |