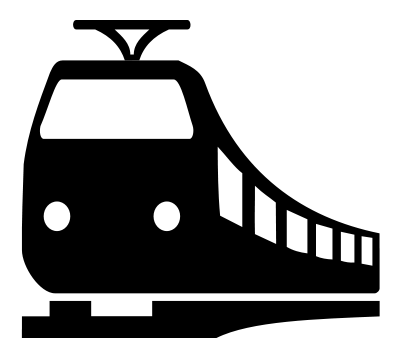
**Dokumentation**

**SwissTransport**

**Samuel Wirth**

**M318**

Inhalt

[Einleitung 3](#_Toc452541979)

[Ziel 3](#_Toc452541980)

[Vorgehen 3](#_Toc452541981)

[use-case diagramme erstellen 4](#_Toc452541982)

[Gui-skizze 5](#_Toc452541983)

[Anforderungen 5](#_Toc452541984)

[A001 5](#_Toc452541985)

[A002 6](#_Toc452541986)

[A003 6](#_Toc452541987)

[A004 6](#_Toc452541988)

[A005 7](#_Toc452541989)

[A006 7](#_Toc452541990)

[A007 8](#_Toc452541991)

[A008 9](#_Toc452541992)

[Testfälle 9](#_Toc452541993)

[Testfall 1 9](#_Toc452541994)

[Testfall 2 9](#_Toc452541995)

[Testfall 3 10](#_Toc452541996)

[Testfall 4 10](#_Toc452541997)

[Testfall 5 10](#_Toc452541998)

[Testfall 6 10](#_Toc452541999)

[Testfall 7 11](#_Toc452542000)

[Testfall 8 11](#_Toc452542001)

# Einleitung

## Ziel

In diesem Dokument wird erklärt, wie der Code vom SwissTransport Programm aufgebaut ist und es sind dazu einige Testfälle beschrieben. Dies sollte dazu dienen den Code zu verstehen und um das Programm festzuhalten.

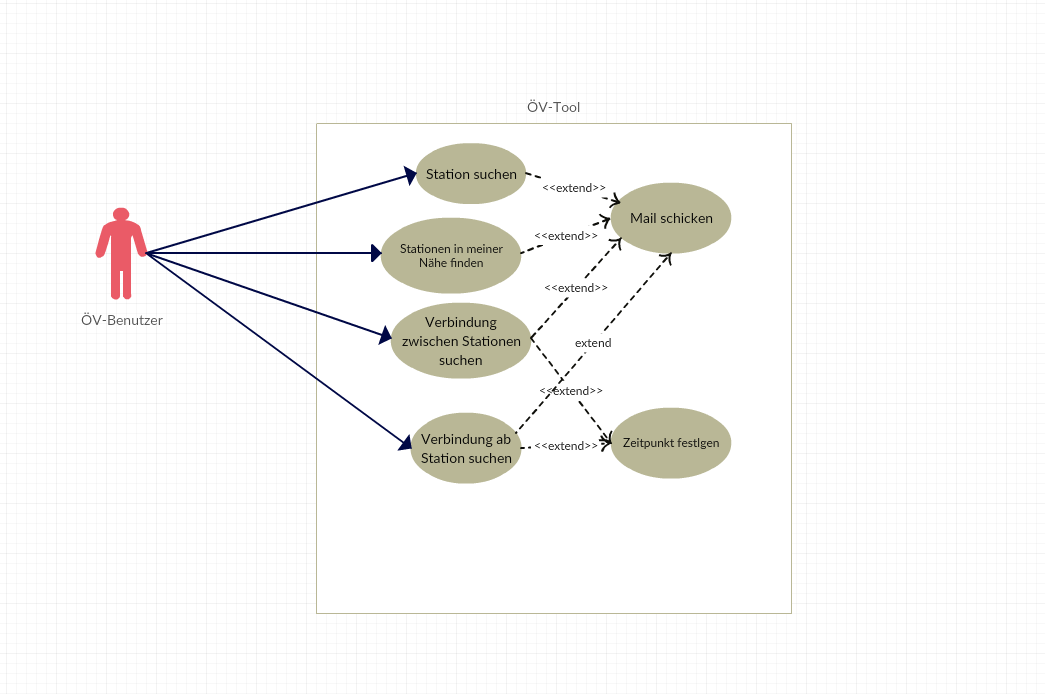
## Vorgehen

Zuerst habe ich alle einzelnen Anforderungen an das Programm beschrieben. Danach habe ich gezeigt, wie und ob ich die Aufgaben lösen konnte. Schlussendlich wurde das Programm mit den am Anfang beschriebenen Testfällen getestet und abgeschlossen.

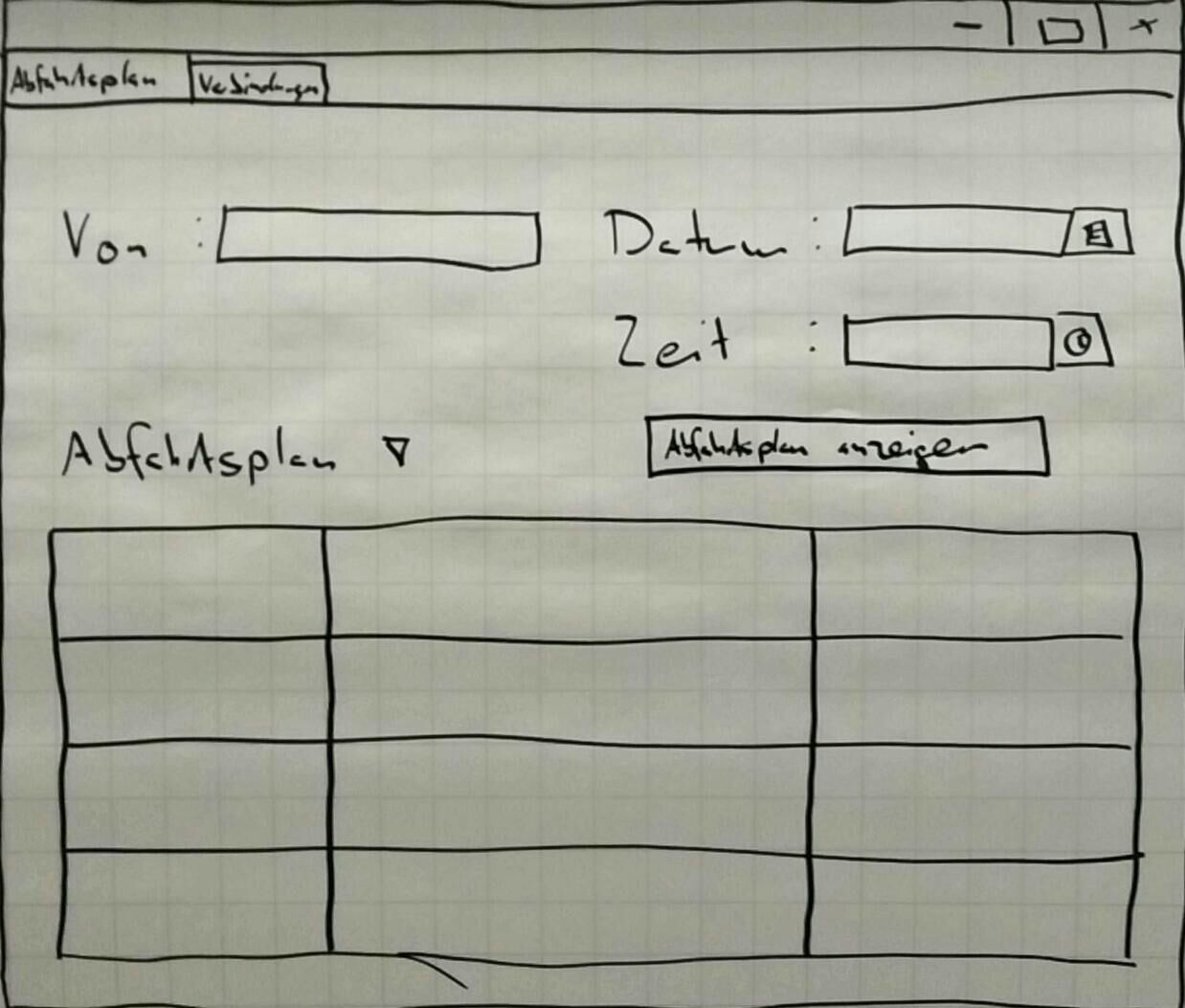
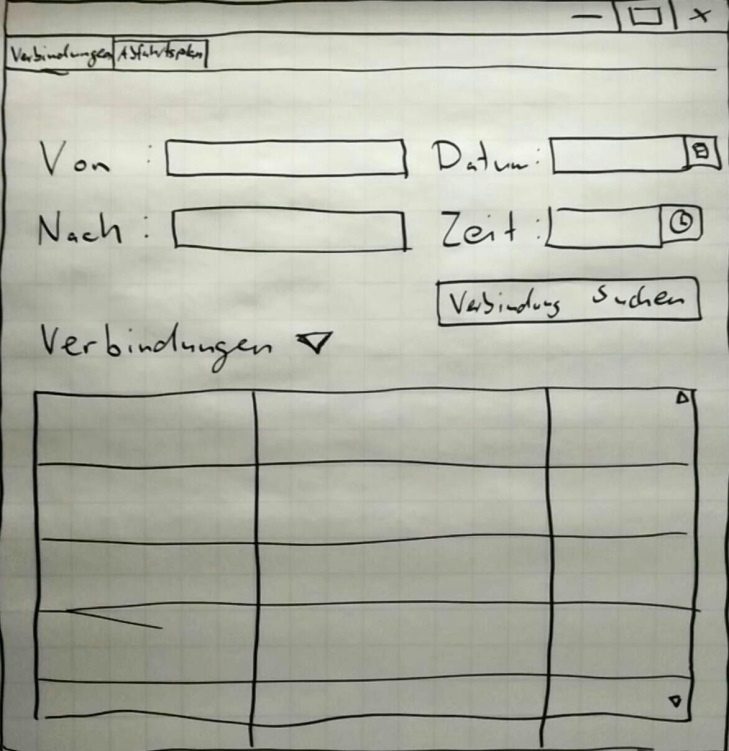
# use-case diagramme erstellen

Damit wir uns vorstellen können, was das Programm können muss und damit wir Abhängigkeiten sehen können.

Ich und Dario haben dazu ein erstes Diagramm erstellt:



# Gui-skizze

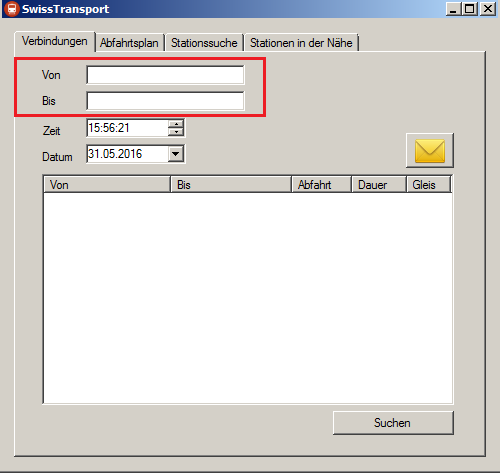
Um es leichter uns zu machen und damit wir einen Plan haben, wie wir das GUI gestalten, haben wir eine GUI-Skizze erstellt:

# Anforderungen

## A001

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

Als erstes habe ich mit dieser Anforderung angefangen. Dafür habe ich das Tab Verbindungen erstellt und die TextBoxen Von, Bis wie auch die Labels dazu eingefügt.



## A002

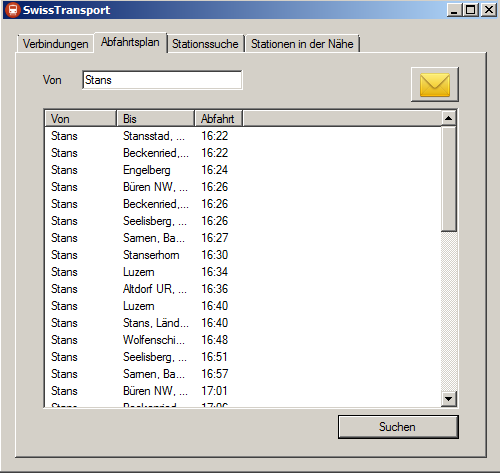
Ich als ÖV-Benutzer möchte die aktuellen Verbindungen zwischen den beiden gefundenen und ausgewählten Stationen sehen, damit ich weiß wann ich zur Station muss.

Um diesen diese Anforderung im Code zu erstellen, habe ich im GUI eine ListView erstellt, in welcher die Verbindungen nachher angezeigt werden. Danach musste ich im ClickEvent des Buttons suchen die Suchen-Logik erstellen. Dazu habe ich die Verbindungen von der Transport-Klasse geladen und diese in die ListView geschrieben.

## A003

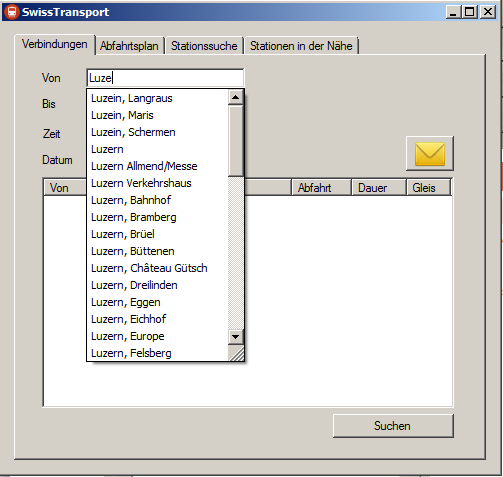
Ich als ÖV-Benutzer möchte sehen welche Verbindungen ab einer bestimmten Station vorhanden sind. Damit ich eine Art Abfahrtstafel Zuhause haben kann.

Für diesen Schritt habe ich ein neues Tab Abfahrtstafel erstellt. Darin habe ich auch eine Von Textbox erstellt und eine ListView, wie auch ein Button Suchen erstellt. Die Logik des Buttons Suchen holt auf der StationBoardRoot-Klasse den Abfahrtsplan und fügt in mit der gleichen Logik der ListView hinzu.



## A004

Ich als ÖV-Benutzer möchte, dass schon währen meiner Eingabe erste Suchresultate erscheinen, damit ich effizienter nach Stationen suchen kann.

Um erste Suchresultate erscheinen zu lassen, habe ich als erstes das Event TextChanged auf den Textboxen erstellt. In diesen Events erstelle ich eine neue AutoCompleteStringCollection, welche extra für diese Vorschläge gemacht ist. Die Liste befülle ich mit den Resultaten der Stationensuche, welche ich mit dem Query suche. Die Liste wird jetzt, falls man mehr als 2 Zeichen eingibt mit den Vorschlägen befüllt.

## A005

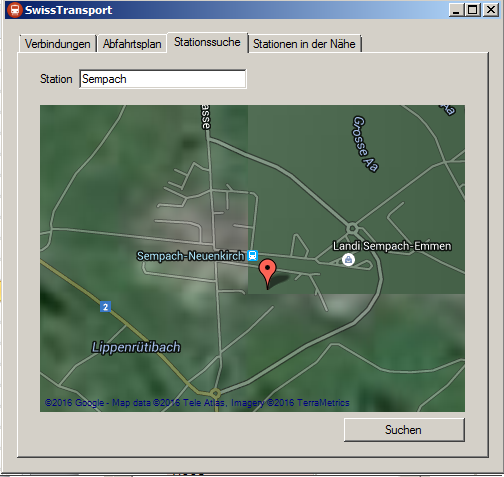
Ich als ÖV-Benutzer möchte Verbindungen zu einem anderen Zeitpunkt suchen können, damit ich zukünftige Reisen planen kann.

Um diese Anforderung zu erfüllen, musste die Verbindungensuche angepasst werden. In der Transport-Klasse habe ich eine neue Methode GetConnectionsByTime erstellt, welche die Verbindungen jetzt auch mit der Zeit und Datum sucht. Dazu musste ich die Grundmethode GetConnections anpassen. Ich musste den WebRequest mit der Zeit und dem Datum ergänzen. Danach konnte ich die Verbindungen auch an einem anderen Datum oder Zeit suchen.

## A006

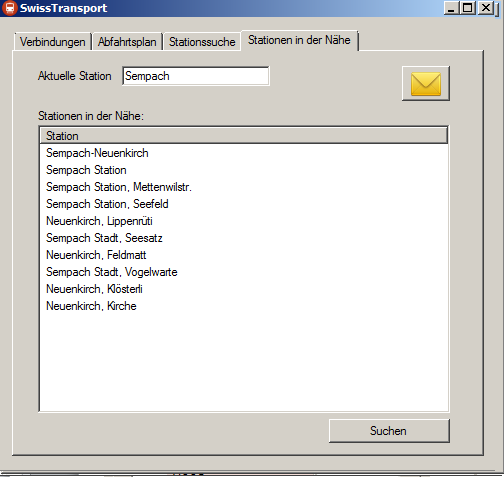
Ich als ÖV-Benutzer möchte sehen wo sich eine Station befindet, damit ich mir die vor Ort-Situation besser vorstellen kann.

Als erstes wurde wieder ein neues Tab Stationssuche erstellt. Ebenfalls wurde wieder Textbox, Label und Button eingefügt. Damit ich das GMapControl von Google verwenden kann, welches nur die Map anzeigt und auf welchem man Marker (Zeiger) setzen kann, musste ich ein Framework herunterladen. Nachdem ich es erfolgreich heruntergeladen habe, habe ich das Control im Tab eingefügt. Falls ich jetzt eine Station suche, werden die Koordinaten der zu suchenden Station gespeichert. Auf diese Station wird jetzt ein Marker gesetzt und herangezoomt.



## A007

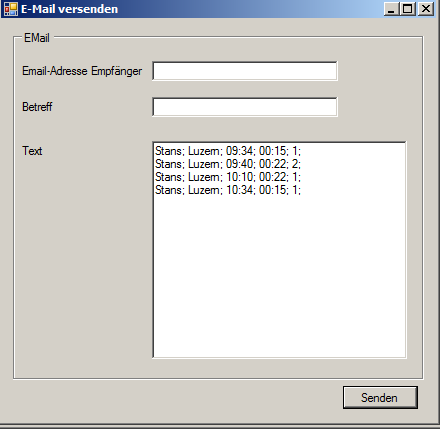
Ich als ÖV-Benutzer möchte Stationen finden die in der Nähe meiner aktuellen Position sind, damit ich schnell auf das ÖV Netzt der Schweiz kommen kann.

Diesen Schritt habe ich als letztes erledigt. Ich habe dafür auch wieder ein neues Tab erstellt „Stationen in der Nähe“. In diesem Tab habe ich wieder ein Textbox, Label, ListView und ein Suchen-Buttton eingefügt. Die Station, für welche man in der Nähe liegenden Stationen finden möchte, kann man in der TextBox „Aktuelle Station“ eingeben. Im Event der Suchen-Buttons habe ich jetzt die Koordinaten dieses Punktes herausgelesen und in Variablen gespeichert. In der Transport-Klasse habe ich eine neue Methode GetStationsByCoordinate erstellt, welche mit den Koordinaten nach in der Nähe liegenden Stationen sucht. Diese Methode habe ich jetzt im Buttonevent aufgerufen, die Variablen (Koordinaten der Station) mitgegeben und die Resultate in der ListView augegeben.

## A008

Ich möchte meine gefundenen Resultate via Mail weiter schicken können, damit ich oder andere von meinen Recherchen profitieren kann.

Als erstes habe ich auf allen Tabs ausser dem Stationen suchen ein Button mit dem Mail Icon hinzugefügt. Als nächstes habe ich ein neues GUI „SendMail“ erstellt. Danach habe ich eine Testemailadresse erstellt, welche zum absenden des Mails dient. Im SendMail Code musste ich als erstes die RichTextBox mit der ListView befüllen. Danach musste ich einige Einstellungen zum Mail-Server definieren, wie z.B. den Port oder die Sicherheitseinstellungen. Im Code habe ich jetzt das Passwort und die E-Mail des Absenders fest hinterlegt. Dies in nicht schön, reicht aber, da es nur eine Testemailadresse ist und da ich dies ja nur zum Testen, ob es funktioniert gebraucht habe.



# Programminstallation

Um das Programm zu installieren, kann man im Programmverzeichnis die .exe-Datei kopieren, wohin man will und sie kann dann von überall gestartet werden. Die .exe Datei findet man hier:

modul-318-student-master\modul-318-student-master\OVTool\bin\Debug\OvTool.exe

# Testfälle

## Testfall 1

Wenn ich eine Start und Endstation im Verbindungen Tab eingebe und dann suche, werden mir die richtigen Verbindungen dieser Stationen angezeigt.

Ergebnis:

Nachdem ich gesucht habe, wurden mir die richtigen Verbindungen angezeigt. Die Abfahrt, Dauer und das Gleis wurden von mir kontrolliert und stimmen.

## Testfall 2

Ich suche ebenfalls wieder nach den Verbindungen, ändere dieses Mal aber die Zeit und das Datum und schaue, ob die Verbindungen von dieser Zeit/Datum geladen werden.

Ergebnis:

Die Verbindungen wurden von der richtigen Zeit geladen und falls ich ein Datum wähle, an welchem Sonderabfahrtszeiten gelten, werden die Verbindungen von diesem Datum angezeigt.

## Testfall 3

Wenn ich nur das Von oder Bis eingebe oder beide leer lass und suche, bei den Verbindungen, stürzt das Programm nicht ab.

Ergebnis:

Das Programm stürzt nicht ab, sondern es wird einfach nichts geladen. Ich verbessere das Programm noch, indem ich bei der Suche prüfe, ob ich ein Von und ein Bis eingegeben habe.

## Testfall 4

Wenn ich im Tab Abfahrtsplan eine Station eingebe und suche, werden mir alle Verbindungen von dieser Station angezeigt.

Ergebnis:

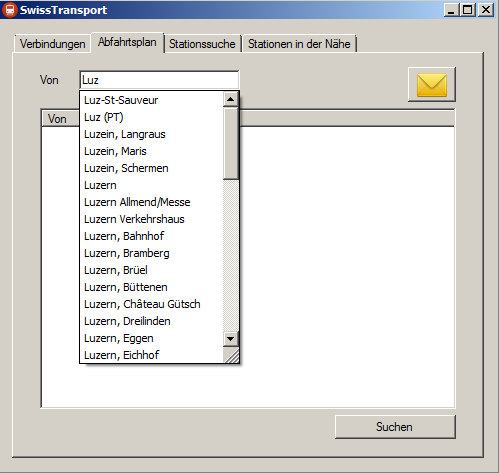
Ich suche nach der Station Luzern und es werden mir alle Verbindungen von Luzern angezeigt, also funkioniert dies richtig.

## Testfall 5

Falls ich in allen TextBoxen nach einer Station suche, wird mir ab dem 3. Zeichen die Stationen vorgeschlagen.

Ergebnis:

Ich gebe in allen TextBoxen eine Station ein und ab dem 3. Zeichen werden mir Stationen vorgeschlagen.



## Testfall 6

Im Tab Stationen Suche, suche ich nach einer Station und sie wird mir markiert und darauf gezoomt. Wenn ich eine Station eingebe, welche es nicht gibt, wird keine markiert.

Ergebnis:

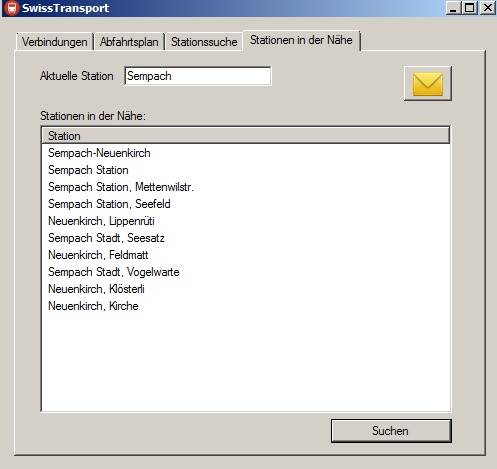
Es wird die richtige Station markiert und auch darauf gezoomt, aber falls ich eine Station eingebe, welche es nicht gibt, wird irgendetwas markiert. Dies verändere ich jetzt nicht, da dies nicht so schlimm ist, dass irgendeine markiert wird, weil das Programm ja immer noch läuft und man weitersuchen kann.

## Testfall 7

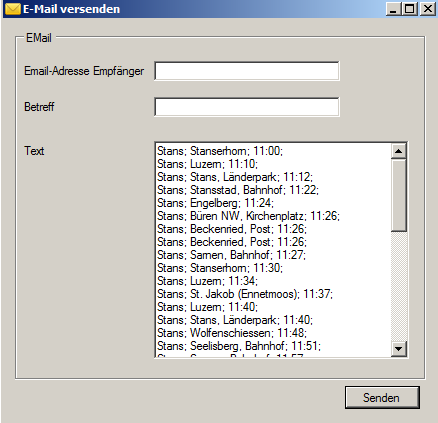
Im Tab Stationen in der Nähe gebe ich in der TextBox „Aktuelle Station“ eine Station ein und es werden mir Stationen, welche in der Nähe dieser Station liegen angezeigt.

Ergebnis:

Ich gebe Sempach ein und es werden mir Stationen anzeigt, welche in der Nähe davon liegen.



## Testfall 8

Ich kann in allen Tabs ausser dem Stationssuchetab die Verbindungen/Ergebnisse mit dem Mail Button versenden. Es öffnet sich zuerst ein Mail Dialog, in welchem die Ergebnisse schon richtig abgefüllt werden und ich den Empfänger und den Betreff definieren kann.

Ergebnis:

Der Dialog wird geöffnet und ich kann den Empfänger, wie auch den Betreff definieren. Das Mail wurde erfolgreich versendet und die Formatierung stimmt.