Leander Duft  
ÜK318  
Dokumentation ÖV Info

Abgabe: 01.06.2016

Inhalt

[1. Einführung 2](#_Toc452533649)

[Definition des Projektes 2](#_Toc452533650)

[Was wurde Umgesetzt 2](#_Toc452533651)

[2. Testfälle Öv Info 2](#_Toc452533652)

[A001: 3](#_Toc452533653)

[A002: 5](#_Toc452533654)

[A003: 10](#_Toc452533655)

[A004: 12](#_Toc452533656)

[3. Vorbereitung 13](#_Toc452533657)

[Use Case Diagramm 13](#_Toc452533658)

[Fluss Diagramm (Aktivitätsdiagramm) 14](#_Toc452533659)

[Installation 16](#_Toc452533660)

[4. Reflexion 17](#_Toc452533661)

[Zusammenarbeit 17](#_Toc452533662)

[Resultat 17](#_Toc452533663)

[Verbesserungen für folgende Male 17](#_Toc452533664)

[Fazit 17](#_Toc452533665)

[5. Quellen 17](#_Toc452533666)

[6. Abbildungsverzeichnis 18](#_Toc452533667)

# Einführung

## 1.1 Definition des Projektes

Bei diesem Projekt handelt es sich um eine WinForms Programm zu erstellen in C#. Viel war schon Vorgegeben, zum einen wurden die Anforderungen schon vor dem ÜK definiert, wir mussten die Anforderungen also nur noch umsetzen. Auch war schon die Anbindung an die API der Datenquelle (<http://transport.opendata.ch> ) komplett vorprogrammiert.

## 1.2 Was wurde Umgesetzt

Im Momentanen Zustand des Projekts wurden folgende Dinge implementiert:

* Der ÖV-Benutzer kann einen Start (From) und einen Ziel (To) Punkt auswählen, die Stationsvorschlag Funktion hilft ihm dabei seine gewünschte Station zu finden.
* Der ÖV-Benutzer kann mit dem Button rechts neben der Start bzw. Ziel Eingabe eine Liste abrufen, auf welcher er alle weiteren Verbindungen zu sehen bekommt, welche von der gewünschten Station ausgehen. (Angezeigt wir Abfahrtszeit, Destination und die Linie Bsp. S 44)
* Der ÖV-Benutzer kann mit einem Klick auf den „Verbindung Suchen“ (Search Connection) Button eine Liste Abrufen, welche die nächsten Verbindungen zwischen den zuvor eingegebenen Start- und Zielorten anzeigt. (Angezeigt wird der Startbahnhof, die Abfahrtszeit, Das Perron am Abfahrtsbahnhof, der Zielbahnhof, die Ankunftszeit, das Perron am Zielbahnhof und die Dauer der Fahrt )
* Durch das Ändern des Datums und der Zeit werden auch die entsprechenden Verbindung Angepasst, Die Zeit bezieht sich auf die Abfahrtszeit.

# Testfälle Öv Info

Jedes Programm muss auch getestet werden, bekommt man nach den Tests das erwartete Ergebnis ist alles im grünen Bereich. Wird jedoch an einem Ort etwas anderes Erwartet als schlussendlich zurückkommt, sollte man sich noch einmal in den Code stürzen um zu sehen an was es liegt und wie dieser Fehler behoben werden kann.

Hier werden die einzelnen Testfälle zu den jeweiligen Anforderungen beschrieben.

Bei allen Testfällen muss als allererstes das Visual Studio gestartet werden und die Solution Datei geöffnet werden. (in unserem Fall die modul-318-student.sln)   
Nachdem die „.sln“ Datei geöffnet wurde kann man das Programm mit „F5“ starten oder man klickt ganz einfach auf klicken.

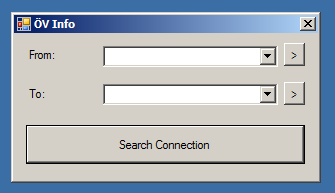
Nun sollte man das Hauptfenster der Applikation zu sehen bekommen:

Abbildung 1 Hauptfenster ÖV-Info

## 2.1 A001:

**Anforderung:**

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

**Testfall:**

* Sucheingabe ist möglich (From , To)

**WAS?** Der erste ganz einfache Test: Ist es möglich in den 2 Textfeldern etwas hineinzuschreiben?

**WIE?** In die Textfelder klicken (oder Taben) und eine Tastatureingabe tätigen

**ERWARTUNG?** In den Textboxen wird der getippte Text angezeigt

**ERGEBNIS?** Der eingegebene Text wird komplett angezeigt.

*Test erfüllt*

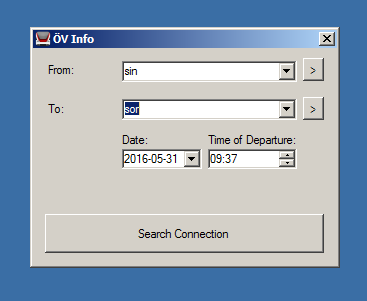


Abbildung 2 Eingabe in die Textfelder

## 2.2 A002:

**Anforderung:**

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

**Testfall:**

* Der Abfahrt und Zielort dürfen nicht derselbe sein

**WAS?** Falls In beiden TextBoxen das gleiche steht, sollen kein Verbindungsfenster geöffnet werden **WIE?** In Beide Textfelder denselben Wert eingeben und der „Search Connection“ Button wird gedrückt  
**ERWARTUNG?** Das Verbindungsfenster wird nie geöffnet, stattdessen wir ein Roter Text angezeigt, welcher dem ÖV-Benutzer erklärt, das seine Eingabe invalid ist.

**ERGEBNIS?** Das Programm öffnet kein weiteres Fenster und es erscheint eine Meldung mit dem Text: „From and To Point can't be the same!“

*Test erfüllt*

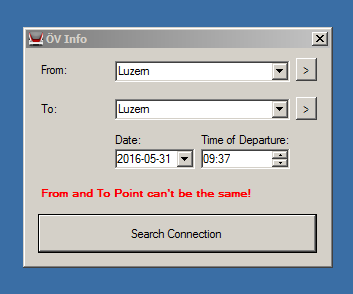


Abbildung 3 Fehlermeldung in beiden Textboxen steht das gleiche

* Der Abfahrt und/oder Zielort darf nicht leer sein.

**WAS?** Falls eine der beiden TextBoxen oder beide leer sein sollten, darf kein Verbindungsfenster geöffnet werden. **WIE?** In einer oder beiden TextBoxen steht kein Wert drin und man klickt den „Search Connection“ Button  
**ERWARTUNG?** Das Verbindungsfenster wird nie geöffnet, stattdessen weist ein roter Test den Öv-Benutzer daraufhin das ein oder mehrere Felder ausgefüllt werden müssen.

**ERGEBNIS?** Das Programm öffnet keine weiteren Fenster und es erscheint eine Meldung mit folgendem Text: „Both Fields must be filled!“

*Test erfüllt*

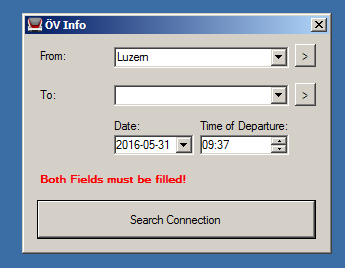
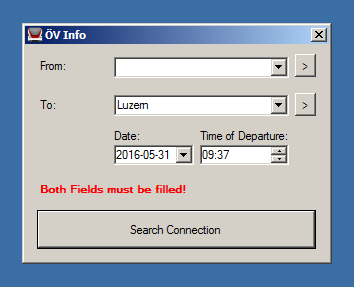


Abbildung 4 Fehlermeldung - die andere Angabe fehlt

Abbildung 5 Fehlermeldung - eine Angabe fehlt

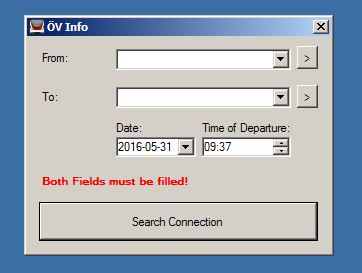


Abbildung 6 Fehlermeldung - Beide Angaben fehlen

* Die Eingabe From „Stansstad“ und To „Luzern“, gibt Verbindungen zwischen Stansstad und Luzern aus. um XX.14 und XX.44 in der Stunde, fahren die Züge. Sie brauchen 18 Minuten von Stansstad nach Luzern.

**WAS?** Ein neues Verbindungsfenster wird angezeigt mit den richtigen Verbindungen (Stansstad, Luzern), Die Orte stimmen, Die Zeiten Stimmen, die Dauer und die Perrons stimmen. **WIE?** In die From TextBox wird der Wert „Stansstad“ eingegeben , in die To TextBox wird der Wert „Luzern“ eingegeben. Der „Search Connection “ Button wird gedrückt.  
**ERWARTUNG?** Das Verbindungsfenster wird geöffnet und die dazugehörige Verbindung (Stansstad, Luzern) wird angezeigt. 2 Einträge müssen mindestens vorhanden sein.

1. Stansstad, XX.14 Uhr ,Gleis 2, Luzern, XX.32 Uhr, Gleis 14, 00:18 HH:MM
2. Stansstad, XX.44 Uhr, Gleis 2, Luzern, XX.02 Uhr, Gleis 14, 00:18 HH:MM

Dies sind die Zug Verbindungen von Stansstad nach Luzern mit der S4 welche halbstündlich fahren.

**ERGEBNIS?**  Das Ergebnis liefert ein geöffnetes Verbindungsfenster mit den dazugehörigen Verbindung, angezeigt in der ListView. Die minimal benötigten Einträge sind zu sehen.

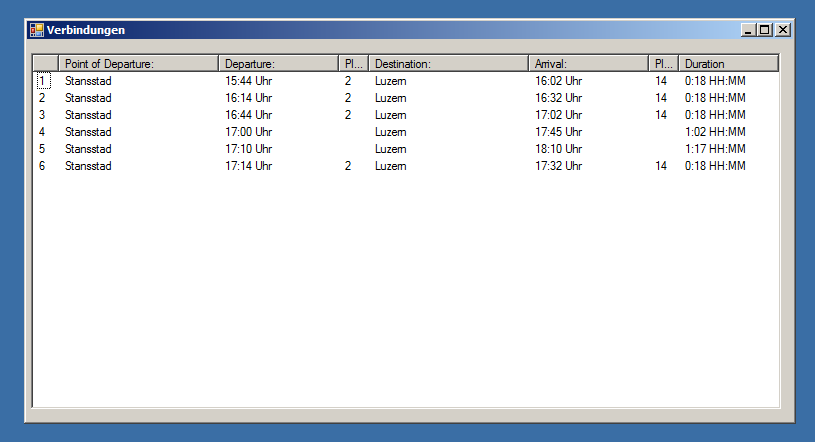


Abbildung 7 SBB Pendant mit denselben Verbindungen

Abbildung 8 Verbindungsfenster mit den korrekten Verbindungen

Die „Normalen“ halbstündlichen Zugverbindungen werden hier noch mit 2 Schiffsverbindungen ergänzt. Darum auch keine Perron Angabe bei zwei Verbindungen. Das eine Schiff fährt um 17.00 los das andere um 17.10.

*Test erfüllt*

* Falls ein Ort nicht existiert oder beide nicht existiert (Keine Verbindung zwischen den Orten besteht), soll eine Fehlermeldung ausgegeben werden.

**WAS?** Ein komischer Wert wird in die Textboxen eingegeben, bzw, openTransport gibt keine Daten zu dieser Verbindung zurück, da keine existiert. **WIE?** In beide TextBoxen gibt man Text mit Sonderzeichen ein. Um das ganze zu Bestätigen frückt man den „Search Connection“ Burtton  
**ERWARTUNG?** Das Verbindungsfenster wird nie geöffnet, stattdessen weist ein roter Test den Öv-Benutzer daraufhin das die Eingegebenen Werte kein Resultat erbracht haben.

**ERGEBNIS?** Das Programm öffnet keine weiteren Fenster und es erscheint eine Meldung mit folgendem Text: „This Search gives No Result, Try again!“

*Test erfüllt*

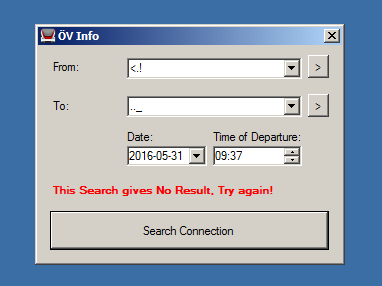
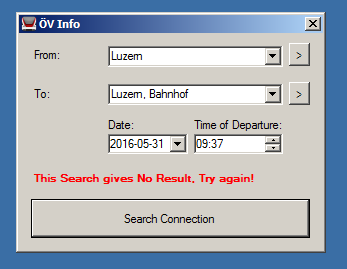


Abbildung 9 Fehlermeldung - Keine Verbindung zwischen Start und Ziel Ort

Abbildung 10 Fehlermeldung - Kein Stationsname

* Bei einem Server Timeout (Error 500), erscheint eine Fehlermeldung

**WAS?** Wenn der ÖV-Benutzer zu schnell oder zu viel in die TextBox eingibt, kann es ein das der Server von Open Transport nicht mehr hinterher kommt. Das Programm würde sich selbst Schließen. **WIE?** Schwierig zu Testen da man den Server überlasten muss, mit vielem schnellem Tippen in das Textfeld sollte es gelingen.  
**ERWARTUNG?** Das Programm sollte nicht abstürzten, sondern ein Meldung an den User in roter Schrift aufleuchten.

**ERGEBNIS?** Ich konnte diesen Testfall nicht nach produzieren.

*Test NICHT erfüllt*

## 2.3 A003:

**Anforderung:**

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

**Testfall:**

* Der Stationsname darf nicht leer sein

**WAS?** Falls die jeweilige TextBox leer ist, wird das StationBoard nicht angezeigt.  
**WIE?** In der jeweiligen TextBoxen steht kein Wert drin und man klickt den Button um das StationBoard anzuzeigen  
**ERWARTUNG?** Das StationBoard wird nie geöffnet, stattdessen weist ein roter Test den Öv-Benutzer daraufhin dass, das Feld ausgefüllt werden muss.

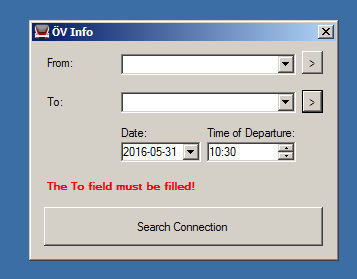
**ERGEBNIS?** Das Programm öffnet keine weiteren Fenster und es erscheint eine Meldung mit folgendem Text: „The To field must be filled!“

Abbildung 11 Fehlermeldung - Das To Feld ist leer und es wurde auf den Button geklickt

*Test erfüllt*

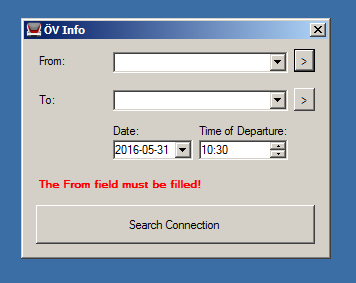


Abbildung 12 Fehlermeldung - Das From Feld ist leer und es wurde auf den Button geklickt

* Falls der Stationsname nicht existiert(Keine Verbindungen von der Station weggehen), soll eine Fehlermeldung angezeigt werden.

**WAS?** Falls die jeweilige TextBox leer ist, wird das StationBoard nicht angezeigt.  
**WIE?** In der jeweiligen TextBoxen steht kein Wert drin und man klickt den Button um das StationBoard anzuzeigen  
**ERWARTUNG?** Das StationBoard wird nie geöffnet, stattdessen weist ein roter Test den Öv-Benutzer daraufhin dass, das Feld ausgefüllt werden muss.

**ERGEBNIS?** Das Programm stürzt nicht ab und es wird eine korrekte Fehlermeldung angezeigt.

Abbildung 13 Fehlermeldung im Visual Studio Try Catch fängt es nicht auf

*Test erfüllt*

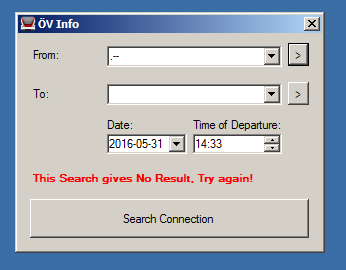


Abbildung 14 Fehlermeldung - Station existiert nicht

## 2.4 A004:

**Anforderung:**

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

**Testfälle:**

* Ergänzt Automatisch das geschriebene (Drop Down)

**WAS?** Nach der Eingabe des Suchbegriffs erscheint das Dropdown Menu mit Vorschlägen zur Station **WIE?** In eines der Textfelder klicken und mit der Stations Eingabe beginnen. **ERWARTUNG?** Nach dem ersten Zeichen erscheint eine leere Dropdownliste, Nach drei weiteren Zeichen erscheinen die ersten Autocomplete Ergebnisse im Drop Down.

**ERGEBNIS?** Nach dem Eingeben des ersten Buchstabens erscheint wie erwartet die Dropdown box, nach den weiteren 3 Zeichen werden die ersten Ergebnisse angezeigt.

*Test erfüllt*

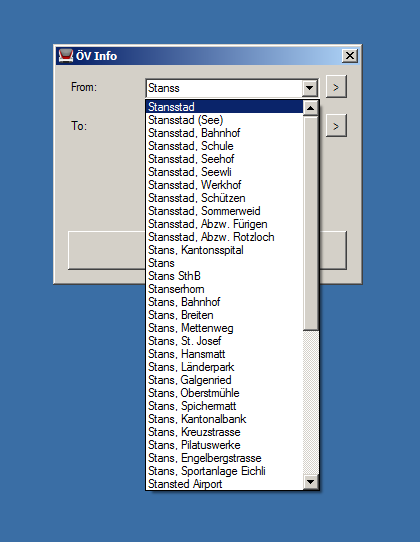
**

Abbildung 15 Drop-Down Funktion

* Die Dropdown Liste erscheint beim ersten geschriebenen Buchstaben.

**WAS?** Nach der Eingabe des ersten Buchstabens erscheint das Dropdown   
**WIE?** In eines der Textfelder klicken und einen Buchstaben in Textfeld eingeben. **ERWARTUNG?** Nach dem ersten Zeichen erscheint eine leere Dropdownliste

**ERGEBNIS?** Nach dem Eingeben des ersten Buchstabens erscheint wie erwartet die Dropdown box

*Test erfüllt*

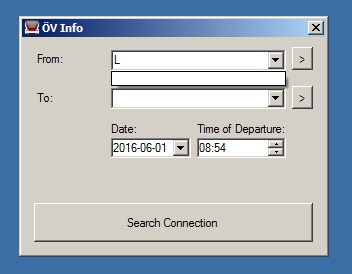


Abbildung 16 Beim ersten Buchstaben öffnet sich das Drop Down

* Es ist möglich mit den Pfeiltasten durch die Dropdownliste zu navigieren um eine Auswahl zu treffen

**WAS?** Mit einem einfachen drücken der Pfeil Tasten Hoch und Runter, selektiert man den Vorherigen bzw. den Nachfolgenden Eintrag im Drop Down. Der Eingegebene Stationsname im Textfeld ändert sich gleich mit, er nimmt den selektierten Stationsnamen an.  
**WIE?** Eingabe einer Station (Bsp. Luzern, ), Pfeiltaste Hoch und Runter Betätigen.   
**ERWARTUNG?** Der selektierte Eintrag sollte sich jeweils ändern(Hoch bzw. Runter rutschen), der Selektierte Eintrag wird verändert. Der geschriebene Text ändert sich ebenfalls bei jeder weiteren selektierung in den selektierten Text.

**ERGEBNIS?** Nach dem drücken der Hoch bzw. der runter Taste, wird wie erwartet der Selektierte Eintrag geändert und der eingegebene Text passt sich auch dem selektierten an.

*Test erfüllt*

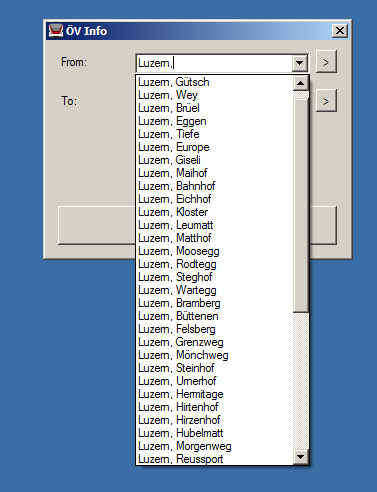
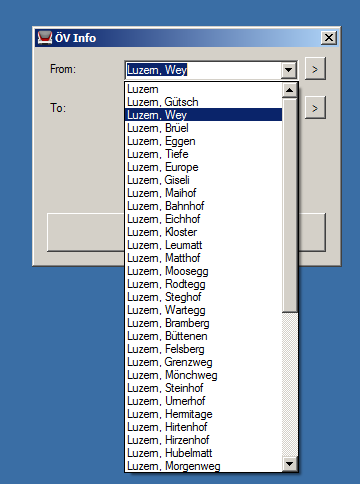
C:\Users\dok409\Downloads\Down Squared-32 (1).png

Abbildung 17 Eingabe der Station - Drop Down wird angezeigt

Abbildung 18 Aussuchen der Station mit Pfeiltasten

## 2.5 A005:

**Anforderung:**

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

**Testfälle:**

* Eingabe von Datum und Zeit ohne Problem möglich

**WAS?** Eingeben in das DatetimePicker Form möglich, beliebige Daten können gewählt werden. Datum und auch Zeit  
**WIE?** Das Formelement anwählen und Datum im richtigen Format eingeben, bzw. auf Pfeil Klicken und das Datum aussuchen. Bei der Zeit ist ähnlich: Das Form Element anwählen und die Uhrzeit im richtigen Format anwählen, bzw. Mit dem Regler die Stunden und Minuten verändern.  
**ERWARTUNG?** Die Uhrzeit und das Datum lassen sich ohne weiteres verändern und werden richtig angezeigt.

**ERGEBNIS?** Nach dem drücken der Hoch bzw. der runter Taste, wird wie erwartet der Selektierte Eintrag geändert und der eingegebene Text passt sich auch dem selektierten an.

*Test erfüllt*

# Vorbereitung

Bevor man zu Programmieren beginnt, sollte man sich Gedanken machen zu den Anforderungen. Wie soll ich das ganze Umsetzen? Welche Elemente brauche ich dazu? Wann passiert Was? Und viele weitere Fragen müssen vor dem Start geklärt werden. Um einen guten Überblick zu behalten kann man folgende Diagramme aufzeichnen

## Use Case Diagramm

Die wichtigsten Punkte um ein Use Case Diagramm zu zeichnen:

* ***Ziel****ist es, möglichst einfach zu zeigen, was man mit dem zu bauenden Softwaresystem machen will, welche Fälle der Anwendung es also gibt.*
* ***Akteure****werden als „*[*Strichmännchen*](https://de.wikipedia.org/wiki/Strichm%C3%A4nnchen)*“ dargestellt, welche sowohl Personen wie Kunden oder Administratoren als auch ein System darstellen können (bei Systemen wird manchmal auch ein Bandsymbol verwendet).*
* ***Anwendungsfälle****werden in Ellipsen dargestellt. Sie müssen beschrieben werden (z. B. in einem Kommentar oder einer eigenen Datei).*
* ***Assoziationen****zwischen Akteuren und Anwendungsfällen müssen durch Linien gekennzeichnet werden.*
* ***Systemgrenzen****werden durch Rechtecke gekennzeichnet.*
* ***include-Beziehungen****werden mittels (mit <<include>> gekennzeichneter) gestrichelter Linie und einem Pfeil zum inkludierten Anwendungsfall gekennzeichnet, wobei dieser für den aufrufenden Anwendungsfall notwendig ist.*

*Quelle: Wikipedia.org*

Auch bei unserer ÖV Applikation durften wir ein Use Case Diagramm aufzeichnen

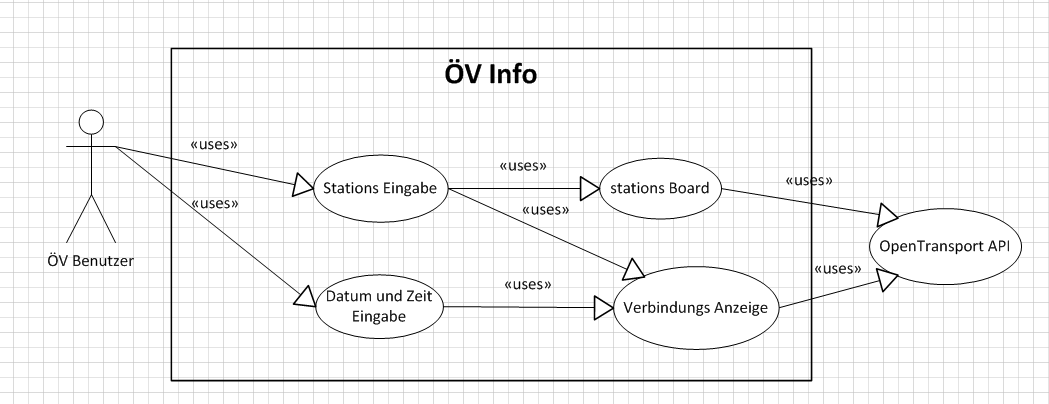


Abbildung 19 Use Case Diagramm

## Fluss Diagramm (Aktivitätsdiagramm)

Auch ein Fluss Diagramm mit den einzelnen Programm Verzweigungen darf nicht fehlen. Diese Verzweigungen zeigen an, wo im Code überall Schlaufen und IF Abfragen sind.

Bei unserem Projekt haben wir zum einen für die Verbindungssuche ein Aktivitätsdiagramm gezeichnet.

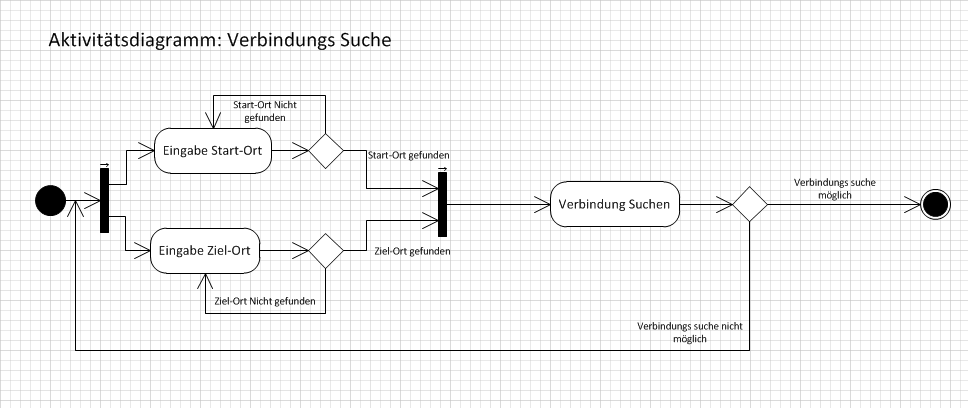


Abbildung 20 Aktivitäts Diagramm Verbindungs Suche

Als zweites haben wir noch eines für das StationsBoard gezeichnet:

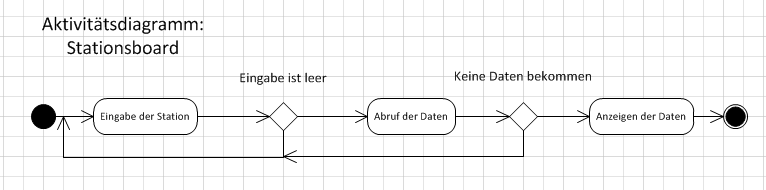


Abbildung 21 Aktivitäts Diagramm Stationsboard

Die beiden Aktivitäts Diagramme liegen auch dem Projekt bei und können mit einem Visio geöffnet werden.

# Installation

Bei meinem Projekt ist die Installation sehr einfach gehalten, man lädt sich von <https://github.com/SirRednael/modul-318-student>

Das ganze Projekt herunter und entzippt den Ordner „[ÖV-Application.zip](https://github.com/SirRednael/modul-318-student/blob/master/%C3%96V-Application.zip)“.   
Darin befindet sich die .Exe Datei und weitere Dateien die zum Funktionieren der Applikation beitragen.

Mit einem Doppelklick auf die .Exe Datei öffnet sich das Programm, bereit zum Arbeiten.

Um das Programm im Visual Studio 2015 zu öffnen und zu starten öffnet man die .sln Datei, welche beim Heruntergeladenen Projekt beiliegt. Danach nur noch F5 oder Starten drücken.

# Reflexion

## 5.1 Zusammenarbeit

Obwohl es nicht als Gruppenarbeit angedacht ist, haben mein Pultnachbar und Ich uns sehr gut unterstützt. Viele Probleme, welche bei uns beiden Auftauchten konnten wir so zu zweit schneller lösen. Auch mit der restlichen Gruppe war das Zusammenarbeiten im Klassenraum sehr angenehm.

## 5.2 Resultat

Mit dem Resultat kann ich mehr oder minder zufrieden sein. Da ich in der Firma nur sehr wenig mit dem Visual Studio und WinForms arbeite, kam ich bei vielen Problemen nur sehr schleichend voran. Aber alles in allem gesehen habe ich eine solide Software programmiert, welche mit den Grundfunktionen A001 – A005 zurechtkommt. Der ÜK half mir meine Code Qualität zu verbessern und hat mir vieles über die Diagramm arten erklären können.

## 5.3 Verbesserungen für folgende Male

Bei einem nächsten ÜK, würde ich darauf achten das ich wenn ich Fragen oder ein Problem habe, früher zum Instruktor gehe und da mein Problem versuche zu lösen.

## 5.4 Fazit

Viele neue Aspekte der Code Qualität kennengelernt, wie auch neue spannende Informationen zu Diagrammen erhalten. Gelernt welche Dinge essentiell für das erschaffen einer Applikation sind. Viele Dinge über den Kundenkontakt haben wir durch gute, realitätsnahe Beispiele kennengelernt. Ein ÜK welcher mich sehr viel neues gelehrt hat, meine Bilanz fällt Positiv aus.

# Quellen

<http://transport.opendata.ch>

<https://github.com/SirRednael/modul-318-student>

<https://www.wikipedia.org> (<https://de.wikipedia.org/wiki/Anwendungsfalldiagramm> )

# Abbildungsverzeichnis

[Abbildung 1 Hauptfenster ÖV-Info 3](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468749)

[Abbildung 2 Eingabe in die Textfelder 3](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468750)

[Abbildung 3 Drop-Down Funktion 4](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468751)

[Abbildung 4 Fehlermeldung in beiden Textboxen steht das gleiche 5](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468752)

[Abbildung 5 Fehlermeldung - die andere Angabe fehlt 6](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468753)

[Abbildung 6 Fehlermeldung - eine Angabe fehlt 6](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468754)

[Abbildung 7 Fehlermeldung - Beide Angaben fehlen 6](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468755)

[Abbildung 8 SBB Pendant mit denselben Verbindungen 7](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468756)

[Abbildung 9 Verbindungsfenster mit den korrekten Verbindungen 7](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468757)

[Abbildung 10 Fehlermeldung - Keine Verbindung zwischen Start und Ziel Ort 8](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468758)

[Abbildung 11 Fehlermeldung - Kein Stationsname 8](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468759)

[Abbildung 12 Fehlermeldung - Das To Feld ist leer und es wurde auf den Button geklickt 10](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468760)

[Abbildung 13 Fehlermeldung - Das From Feld ist leer und es wurde auf den Button geklickt 10](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468761)

[Abbildung 14 Fehlermeldung im Visual Studio Try Catch fängt es nicht auf 11](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468762)

[Abbildung 15 Fehlermeldung - Station existiert nicht 11](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation\Dokumentation_ÖV-Info.docx#_Toc452468763)

[Abbildung 16 Use Case Diagramm 12](#_Toc452468764)

[Abbildung 17 Aktivitäts Diagramm Verbindungs Suche 13](#_Toc452468765)

[Abbildung 18 Aktivitäts Diagramm Stationsboard 13](#_Toc452468766)