Leander Duft  
ÜK318  
Dokumentation ÖV Info

Abgabe: 01.06.2016

Inhalt

[1. Einführung 2](#_Toc452454333)

[1.1. Definition des Projektes 2](#_Toc452454334)

[1.2. Was wurde Umgesetzt 2](#_Toc452454335)

[2. Testfälle Öv Info 2](#_Toc452454336)

[2.1. A001: 3](#_Toc452454337)

[2.2. A002: 5](#_Toc452454338)

[2.3. A003: 10](#_Toc452454339)

[3. Reflexion 12](#_Toc452454340)

[3.1. Zusammenarbeit 12](#_Toc452454341)

[3.2. Resultat 12](#_Toc452454342)

[3.3. Verbesserungen für folgende Male 12](#_Toc452454343)

[3.4. Fazit 12](#_Toc452454344)

[4. Quellen 12](#_Toc452454345)

[5. Abbildungsverzeichnis 12](#_Toc452454346)

# Einführung

## 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.

## 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 )

# 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.

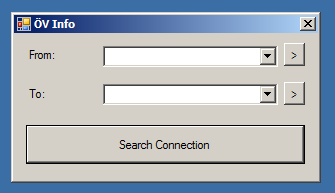
Nun sollte man das Hauptfenster der Applikation zu sehen bekommen:

Abbildung Hauptfenster ÖV-Info

## 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*

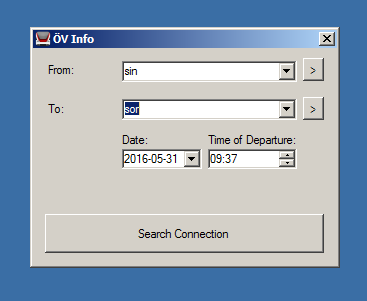


Abbildung Eingabe in die Textfelder

* Ergänzt Automatisch das geschriebene (Drop Down)

**WAS?** Nach der Eingabe des ersten Buchstabens 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*

Somit wurde die komplette Anforderung 1 erfüllt. Alle Test wurden mit Positivem Resultat abgeschlossen.

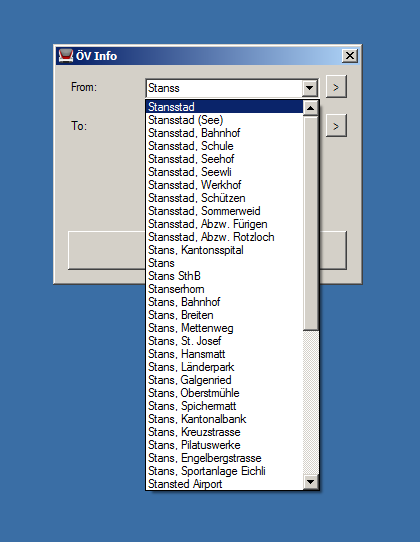
**

Abbildung Drop-Down Funktion

## 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*

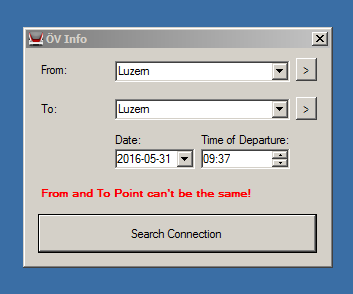


Abbildung Fehlermeldung in beiden Textboxen steht das gleiche

* 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*

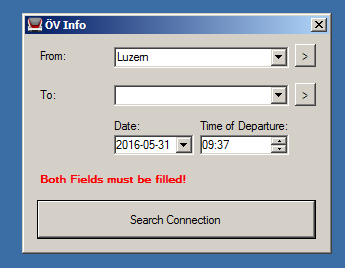
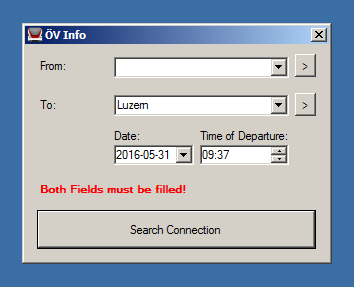


Abbildung Fehlermeldung - die andere Angabe fehlt

Abbildung Fehlermeldung - eine Angabe fehlt

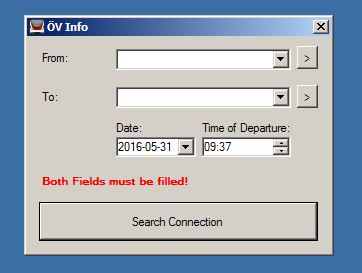


Abbildung Fehlermeldung - Beide Angaben fehlen

* 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.

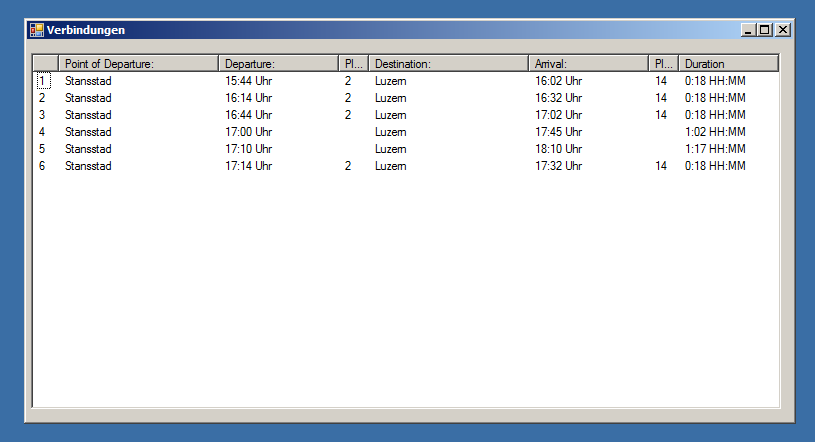


Abbildung SBB Pendant mit denselben Verbindungen

Abbildung 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*

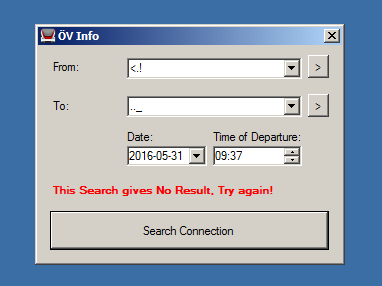
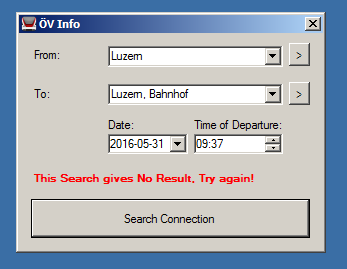


Abbildung Fehlermeldung - Keine Verbindung zwischen Start und Ziel Ort

Abbildung 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*

## 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.

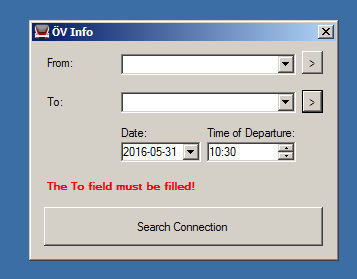
**ERGEBNIS?** Das Programm öffnet keine weiteren Fenster und es erscheint eine Meldung mit folgendem Text: „The To field must be filled!“

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

*Test erfüllt*

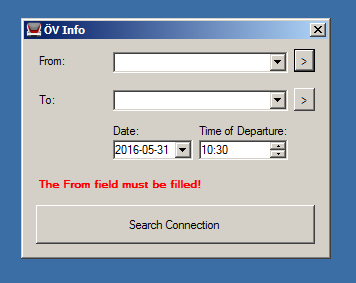


Abbildung Fehlermeldung - Das From Feld ist leer und es wurde auf den Button geklickt

* 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 momentan noch ab und gibt keine Fehlermeldung aus. Das Try Catch Konstrukt greift nicht.

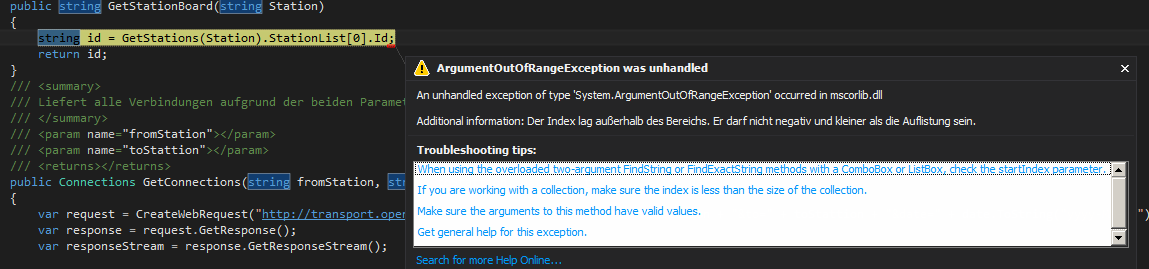
*Test NICHT erfüllt*

Abbildung Fehlermeldung im Visual Studio Try Catch fängt es nicht auf

# 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

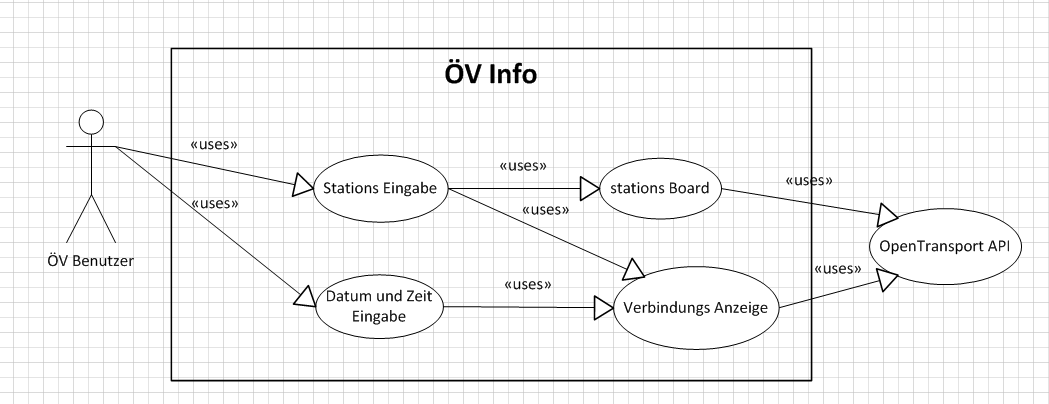


Abbildung Use Case Diagramm

## 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.

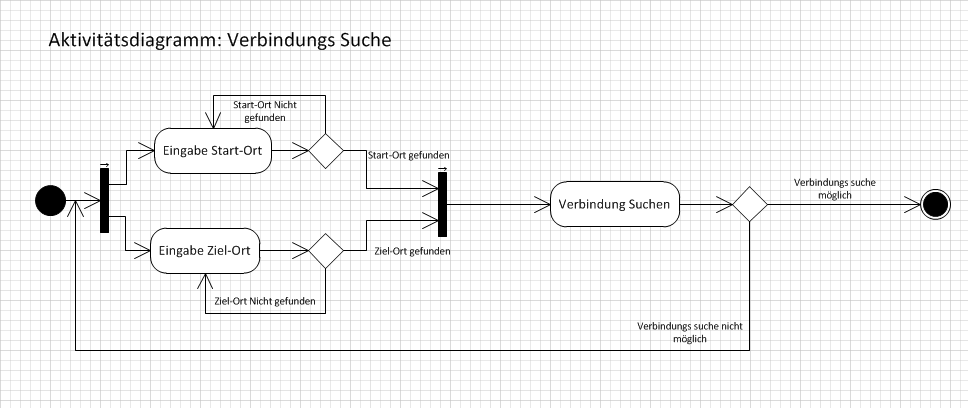


Abbildung Aktivitäts Diagramm Verbindungs Suche

Als zweites haben wir noch eines für das StationsBoard gezeichnet:

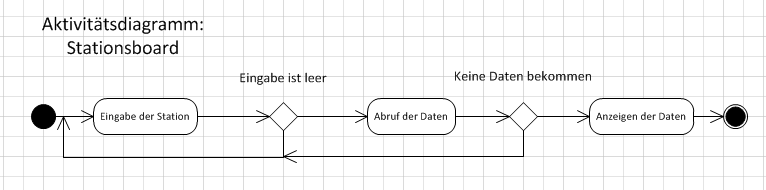


Abbildung Aktivitäts Diagramm Stationsboard

# 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

# Reflexion

## Zusammenarbeit

In der ersten Lektion konnte nur Lukas arbeiten, da Leander abwesend war. Deshalb haben wir via Dropbox unsere Dateien synchronisiert und konnten so von Zuhause aus speditiv arbeiten. Via Internettelefonie konnten wir uns unter der Woche absprechen.

## Resultat

Das Resultat hat unsere Erwartungen mehr als erfüllt, da wir nicht wirklich erwartet hatten, die heutige Organisation zu übertreffen. Auch der Theorieteil kam besser heraus als angenommen, da wir zuvor nichts über das Thema wussten und ein informatives Dokument im Internet gefunden haben.

## Verbesserungen für folgende Male

Wir sollten uns im Falle einer Absenz besser schon vor der Lektion absprechen. Da dies nicht geschah konnten wir erst Montags in der letzten Woche richtig loslegen.

## Fazit

Wir haben viele Organisationsformen kennen gelernt und durch die Arbeit mit ihnen intensiv genug beschäftigt, um sie auch verstanden zu haben. Wir haben auch wertvolle Erfahrungen mit Dateimanagement und Kommunikation gesammelt.

# 5 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 2](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464548)

[Abbildung 2 Eingabe in die Textfelder 3](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464549)

[Abbildung 3 Drop-Down Funktion 4](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464550)

[Abbildung 4 Fehlermeldung in beiden Textboxen steht das gleiche 5](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464551)

[Abbildung 5 Fehlermeldung - die andere Angabe fehlt 6](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464552)

[Abbildung 6 Fehlermeldung - eine Angabe fehlt 6](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464553)

[Abbildung 7 Fehlermeldung - Beide Angaben fehlen 6](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464554)

[Abbildung 8 SBB Pendant mit denselben Verbindungen 7](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464555)

[Abbildung 9 Verbindungsfenster mit den korrekten Verbindungen 7](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464556)

[Abbildung 10 Fehlermeldung - Keine Verbindung zwischen Start und Ziel Ort 8](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464557)

[Abbildung 11 Fehlermeldung - Kein Stationsname 8](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464558)

[Abbildung 12 Fehlermeldung - Das To Feld ist leer und es wurde auf den Button geklickt 10](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464559)

[Abbildung 13 Fehlermeldung - Das From Feld ist leer und es wurde auf den Button geklickt 10](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464560)

[Abbildung 14 Fehlermeldung im Visual Studio Try Catch fängt es nicht auf 11](file:///W:\Teilnehmer\Leander%20Duft\Repo\Dokumentation_ÖV-Info.docx#_Toc452464561)

[Abbildung 15 Use Case Diagramm 12](#_Toc452464562)

[Abbildung 16 Aktivitäts Diagramm Verbindungs Suche 13](#_Toc452464563)

[Abbildung 17 Aktivitäts Diagramm Stationsboard 13](#_Toc452464564)