

# ÖV-Applikation

Projektarbeit ÜK 318

Inhalt

Erstelldatum

Änderungsdatum

Autor

**Projektarbeit ÜK 318**

**Montag, 3. Dezember 2018**

**Dienstag, 4. Dezember 2018**

**Bucher Luca [Bison Schweiz AG]**

## Inhaltsverzeichnis

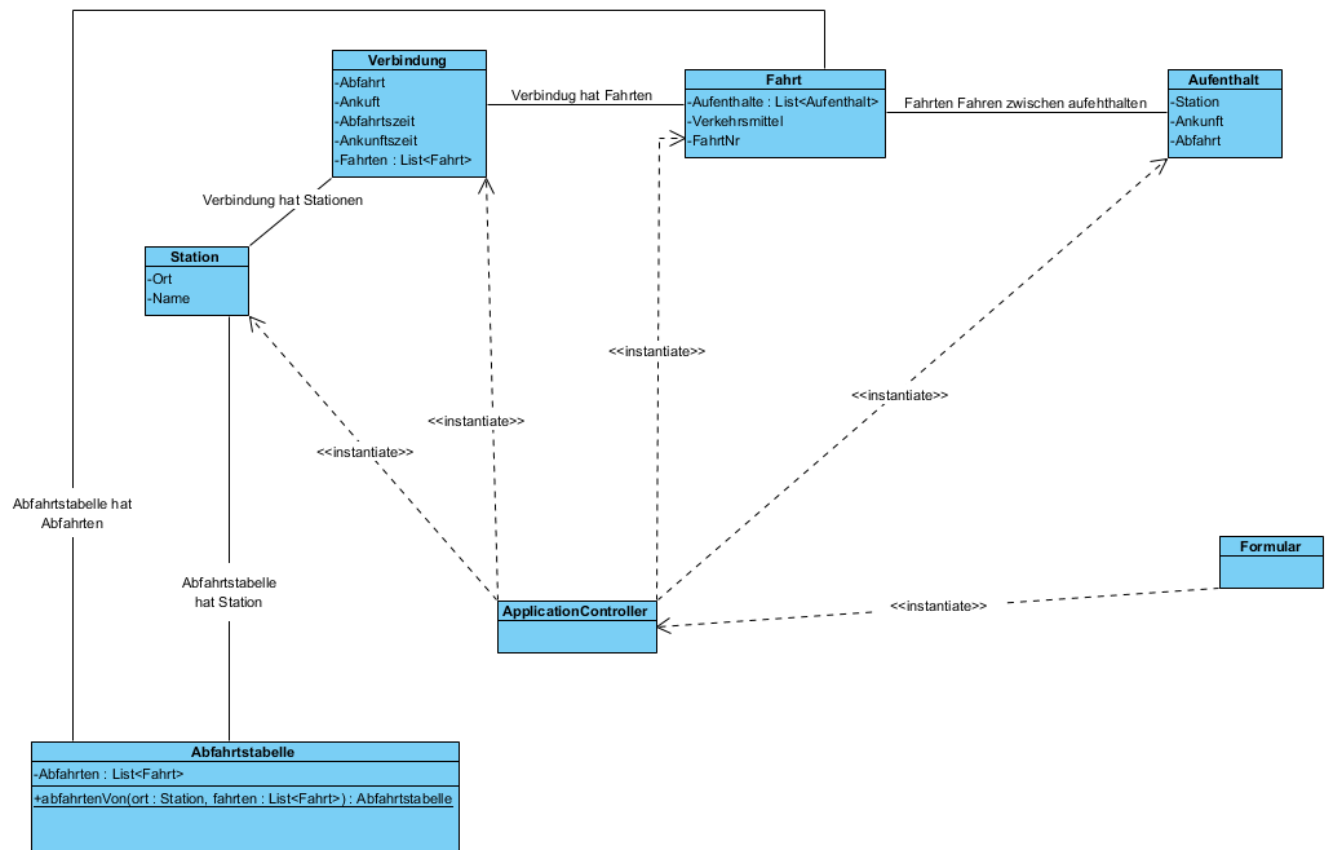
<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Vorhersage</b>	<b>4</b>
2.1	Klassendiagramm	4
2.2	Mockup	5
<b>3</b>	<b>Code Guidelines</b>	<b>7</b>
3.1	Klassennamen	7
3.2	Variabel-Namen	7
3.3	Methoden	8
3.4	Geschweifte Klammern	8
3.5	Files	8
3.6	WinForms Controls	9
<b>4</b>	<b>Umgesetzte Anforderungen</b>	<b>10</b>
4.1	A001	10
4.2	A002	10
4.3	A003	10
4.4	A005	10
4.5	A006	10
<b>5</b>	<b>Use Cases</b>	<b>11</b>
5.1	Diagramm	11
5.2	Beschreibungen	11
5.2.1	Station auswählen	11
5.2.2	Verbindung suchen	12
5.2.3	Abfahrten anzeigen	12
<b>6</b>	<b>Testfälle</b>	<b>13</b>
6.1	UnitTests	13
6.1.1	TransportTest	13
6.1.2	DurationTest	13
6.2	Manuelle Tests	14
6.2.1	A001	14
6.2.2	A002	14
6.2.3	A003	15
6.2.4	A005	15
6.2.5	A006	15
<b>7</b>	<b>Installationsanleitung</b>	<b>Fehler! Textmarke nicht definiert.</b>
<b>8</b>	<b>Schlusswort</b>	<b>17</b>

## 1 Einleitung

In diesem Projekt geht es darum, ein Projekt basierend auf gegebenen Anforderungen zu führen. Das Ziel ist es, ein Windows-Programm zu entwickeln, das mindestens die ersten 3 Anforderungen erfüllt. Zählend ist aber nicht nur das Programm am Schluss sondern auch die Punkte Programmcode, Testing und Dokumentation.

## 2 Vorhersage

### 2.1 Klassendiagramm



## 2.2 Mockup

ÖV-Applikation

Ich möchte: ☒ Verbindungen suchen ☐ Abfahrten suchen

Startstation auswählen

Luzern

Luzern, Bahnhof  
Luzern, Pilatusplatz  
<weitere Vorschläge>

Endstation auswählen

Adligenswil

Adligenswil, Dorf  
Adligenswil, Stuben  
<Weitere Vorschläge>

Wann?

28.11.2018 8:00

Verbindungen suchen

Verbindung 1

Abfahrt: Luzern, Bahnhof  
26.11.2018 um 8:05 Uhr

30 min

Ankunft: Adligenswil, Stuben  
26.11.2018 um 8:35 Uhr

Verbindung 2

Abfahrt: Luzern, Bahnhof  
26.11.2018 um 8:35 Uhr

25 min

Ankunft: Adligenswil, Stuben  
26.11.2018 um 9:00 Uhr

Verbindung 3

Abfahrt: Luzern, Bahnhof  
26.11.2018 um 9:05 Uhr

30 min

Ankunft: Adligenswil, Stuben  
26.11.2018 um 9:35 Uhr

ÖV-Applikation

Ich möchte: ☐ Verbindungen suchen ☒ Abfahrten suchen

Startstation auswählen

Luzern

Luzern, Bahnhof  
Luzern, Pilatusplatz  
<weitere Vorschläge>

Endstation auswählen


Wann?

28.11.2018 8:00

Abfahrten suchen

S1

Luzern, Bahnhof  
26.11.2018 um 8:05 Uhr



### 3 Code Guidelines

Alles was nicht mit Ausgaben an den Benutzer zu tun hat, ist in Englisch zu halten. (Klassen, Variablen, Methoden, ...)

#### 3.1 Klassennamen

Klassennamen werden in CamelCase geschrieben und beginnen mit einem Grossbuchstaben. Der Name der Klasse sollte ein Nomen sein.

```
0 Verweise | 0 Änderungen | 0 Autoren, 0 Änderungen
class DataSearcher
{
    // Code ...
}
```

#### 3.2 Variabel-Namen

Variabel-Namen sind ebenfalls in CamelCase geschrieben, beginnen jedoch mit einem kleinen Buchstaben. Hierbei spielt es keine Rolle, ob die Variable global, lokal oder ein Attribut einer Methode ist. Die einzige Ausnahme bildet die Konstante, welche nur in Grossbuchstaben geschrieben wird. In einer Konstante werden die einzelnen Wörter mit Underscore (\_) getrennt.

```
0 Verweise | 0 Änderungen | 0 Autoren, 0 Änderungen
class DataSearcher
{
    private static Transport swissTransportApi = new Transport();
    private const int LIMIT_OF_ROWS = 16;

    /// <summary>
    /// Search a list of Stations with a query and returns them
    /// </summary>
    /// <param name="stationName">query</param>
    /// <returns>stations</returns>
    0 Verweise | 0 Änderungen | 0 Autoren, 0 Änderungen
    public static List<string> searchStations(string stationName)
    {
        List<string> stations = new List<string>();
    }
}
```

### 3.3 Methoden

Auch Methoden-Namen werden in CamelCase geschrieben. Auch hier beginnt der Name mit einem kleinen Buchstaben. Eine Methoden-Name sollte ein Zusammenspiel zwischen einem Verb und einem Nomen sein. Wenn die Aktion in der Methode etwas mit dem Objekt selber macht, kann auch nur ein Verb verwendet werden (z.B. `Duration.parse(...)`). Zwischen zwei Methoden ist immer eine Linie Abstand. Alle public-Methoden haben oberhalb der Methodendeklaration einen XML-Kommentar (///).

```
private const int LIMIT_OF_ROWS = 16;

/// <summary>
/// Search a list of Stations with a query and returns them
/// </summary>
/// <param name="stationName">query</param>
/// <returns>stations</returns>
0 Verweise | 0 Änderungen | 0 Autoren, 0 Änderungen
public static List<string> searchStations(string stationName)
{
```

### 3.4 Geschweifte Klammern

Die geschweiften Klammern kommen immer unterhalb des Kopfes. Ein else und else if steht auf einer Linie zwischen zwei geschweiften Klammern. Die geschweiften klammern werden auch dann verwendet, wenn danach nur eine Zeile Code ausgeführt wird.

### 3.5 Files

Jedes File soll gleich heissen, wie die Klasse darin (z.B. bei der Klasse Transport soll das File Transport.cs heissen).

```
foreach (string station in stations)
{
    // Code ...
}

if (stations.Count > 1)
{
    // Code if ...
}
else if (stations.Count == 1)
{
    // Code else if ...
}
else
{
    //Code else ...
}
```



### 3.6 WinForms Controls

Auch die WinForms Controls werden in CamelCase beginnend mit kleinen Buchstaben geschrieben. Voraus kommt aber immer noch ein dreibuchstabiger identifier:

Identifier	WinForms Control
<b>btn</b>	Button
<b>clm</b>	Spalte einer DataGridView
<b>dtp</b>	DateTimePicker
<b>grb</b>	GroupBox
<b>lbl</b>	Label
<b>lsb</b>	ListBox
<b>pnl</b>	Panel
<b>rdb</b>	RadioButton
<b>tbl</b>	DataGridView (Tabelle)
<b>txb</b>	TextBox
<b>wbb</b>	WebBrowser

## **4 Umgesetzte Anforderungen**

### **4.1 A001**

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

### **4.2 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.

### **4.3 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.

### **4.4 A005**

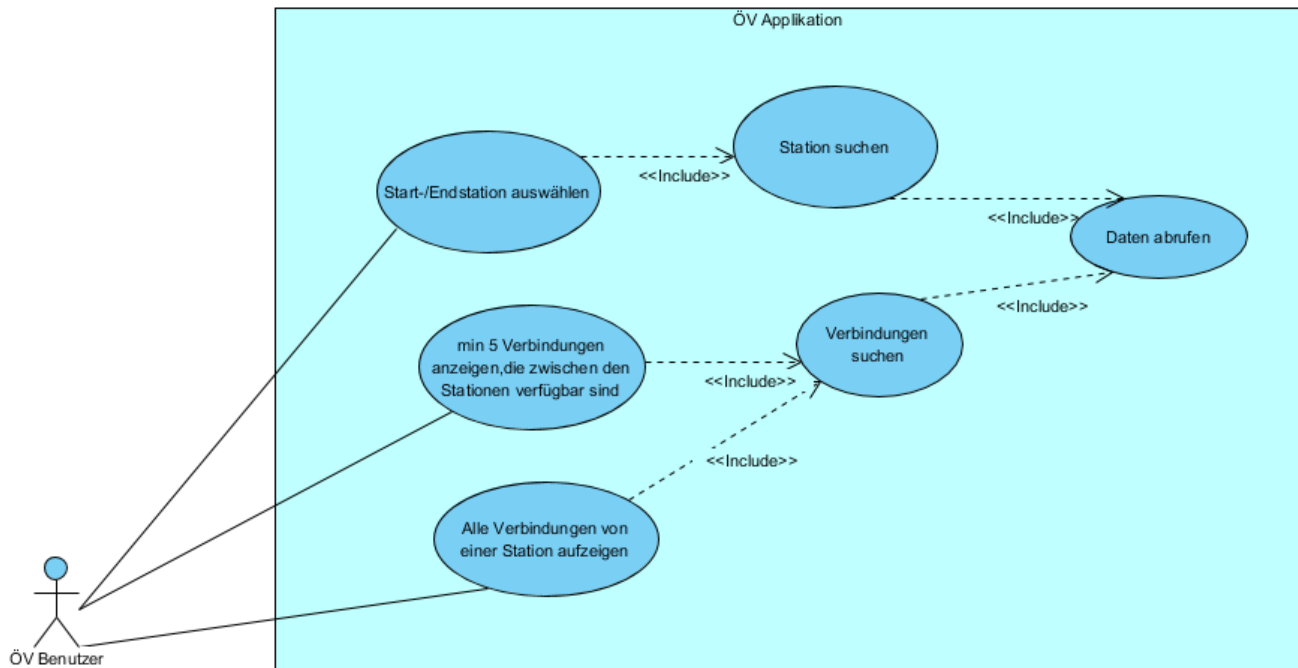
Als ÖV-Benutzer möchte ich nicht nur aktuelle Verbindungen suchen können, sondern auch solche zu einem beliebigen anderen Zeitpunkt, damit ich zukünftige Reisen planen kann.

### **4.5 A006**

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

## 5 Use Cases

### 5.1 Diagramm



## 5.2 Beschreibungen

### 5.2.1 Station auswählen

Use Case Beschreibung	
<b>Beschreibung</b>	Der ÖV-Benutzer möchte eine Station auswählen
<b>Voraussetzungen</b>	Internetverbindung
<b>Aktor</b>	ÖV-Benutzer
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Programm öffnen</li> <li>2. Textsuche nach Station</li> <li>3. Station aus Vorschlägen anzeigen</li> </ol>
<b>Varianten</b>	Start-/Endstation
<b>Ergebnis</b>	Station wurde ausgewählt

### 5.2.2 Verbindung suchen

Use Case Beschreibung	
<b>Beschreibung</b>	Der ÖV-Benutzer möchte eine Verbindung suchen, um von einer Station zur nächsten zu kommen.
<b>Voraussetzungen</b>	Internetverbindung
<b>Aktor</b>	ÖV-Benutzer
<b>Ablauf</b>	<ol style="list-style-type: none"><li>1. Stationen auswählen (siehe Use Case Station auswählen)</li><li>2. Befehl zum suchen geben</li></ol>
<b>Varianten</b>	-
<b>Ergebnis</b>	Liste von mindestens 5 Verbindungen zwischen den jeweiligen Stationen

### 5.2.3 Abfahrten anzeigen

Use Case Beschreibung	
<b>Beschreibung</b>	Der ÖV-Benutzer möchte alle abfahren von einer Station aufgelistet haben
<b>Voraussetzungen</b>	Internetverbindung
<b>Aktor</b>	ÖV-Benutzer
<b>Ablauf</b>	<ol style="list-style-type: none"><li>1. Startstation auswählen (siehe Use Case Station auswählen)</li><li>2. Befehl zum suchen geben</li></ol>
<b>Varianten</b>	-
<b>Ergebnis</b>	Liste von allen Verbindungen, die als Abfahrtsort die Startstation haben.

## 6 Testfälle

### 6.1 UnitTests

#### 6.1.1 TransportTest

Die Klasse Transport ist für die Kommunikation zwischen dem API und der ÖV-Applikation zuständig.

##### 6.1.1.1 transport\_searchConnections\_connections()

Getestete Methode: `Transport.searchConnections()`

Als ÖV-Benutzer suche ich 4 Verbindungen von Sursee nach Luzern.

Ich erwarte, dass ich 4 Verbindungen bekomme und diese die Startstation Sursee und die Endstation Luzern haben.

##### 6.1.1.2 transport\_searchStationboards\_stationBoardRoot()

Getestete Methode: `Transport.searchStationboards();`

Als ÖV-Benutzer möchte ich mich über die nächsten Abfahrten von Sursee informieren.

Ich erwarte, dass alle Abfahrten von Sursee her gehen.

#### 6.1.2 DurationTest

Die Klasse Duration wandelt den Duration String der Api um und hat eine Formatierte Ausgabe.

##### 6.1.2.1 apiDuration\_parse\_timeSpan()

Getestete Methode: `Duration.parse()`

Ich bekomme von der API nur einen String, der mir die Zetispanne angibt. Damit ich damit besser arbeiten kann, möchte ich diesen in eine TimeSpan umwandeln.

Ich bekomme von der API den Duration-Strin "02d:05:33:33" und gebe diesen der Methode mit.

Ich erwarte eine TimeSpan, die mir diesen String repräsentiert:

2 Tage, 5 Stunden, 33 Minuten und 33 Sekunden

##### 6.1.2.2 TimeSpan\_userOutput\_output

Getestete Methode: `Duration.userOutput()`

Ich habe eine TimeSpan und möchte diese in eine gut lesbare Ausgabe konvertieren. Ich benötige nur Tage, Stunden und Minuten.

Ich habe eine TimeSpan von 35 Tage, 8 Stunden, 1 Minute und 35 Sekunden.

Ich erwarte folgende Ausgabe: "35 Tage, 8 Stunden, 1 Minute"

## 6.2 Manuelle Tests

### 6.2.1 A001

Folgende Schritte sollten auf dem Hauptfenster bei beiden Textfeldern funktionieren:

Schritt	Ereignis	Ergebnis
1	F1 drücken (Autocomplete) ohne Suchbegriff zu verwenden	Warnmeldung, die einem darüber informiert, dass man einen Suchbegriff eingeben soll
2	Eine wirre Zeichenfolge eingeben und F1 drücken	Warnmeldung, dass keine Ergebnisse gefunden wurden
3	Einen Suchbegriff eingeben und F1 drücken	Suchergebnisse werden in die ListBox unterhalb eingefügt und können ausgewählt werden
4	Ich möchte den Suchbegriff ändern und drücke ESCAPE	Die ListBox wird geleert und deaktiviert. Die TextBox steht wieder im Fokus
5	Schritt 2	Schritt 2
6	Ich kann mich mit den Pfeiltasten zum richtigen Begriff navigieren und drücke ENTER	<p>Der Selektierte Text wird in die TextBox übernommen und die ListBox wird deaktiviert.</p> <p>Wenn ich bei der 1. TextBox war und ich Verbindungen suche, steht jetzt die 2. TextBox im Fokus.</p> <p>Wenn ich bei der 1. TextBox war und ich Abfahrten suche, steht jetzt die Datumseingabe im Fokus.</p> <p>Wenn ich bei der 2. TextBox war, kann ich jetzt mit den Pfeiltasten zwischen Abfahrt und Ankunft auswählen.</p>

### 6.2.2 A002

Folgende Schritte sollten auf dem Hauptfenster möglich sein:

Schritt	Ereignis	Ergebnis
1	ENTER drücken (ohne jede Eingabe/Eingabe nur bei einem Textfeld)	Warnmeldung, die darauf hinweist, dass keine Ergebnisse gefunden wurden. Man wird aufgefordert, die Suche anzupassen.
2	Eine wirre Zeichenfolge eingeben und ENTER drücken	Warnmeldung, dass keine Ergebnisse gefunden wurden
3	Zwei Stationen eingeben (evtl. mit Autocomplete) und ENTER drücken oder auf "Verbindungen suchen" klicken	Falls vorhanden, sollen 10 Verbindungen zwischen den 2 Stationen aufgezeigt werden. Falls es keine Ergebnisse gibt, erscheint die Warnmeldung von Schritt 2.

### 6.2.3 A003

Folgende Schritte sollten auf dem Hauptfenster möglich sein:

Schritt	Ereignis	Ergebnis
1	Mit F2 oder mit durch wechseln des Radiobuttons die Aktion wechseln	Warnmeldung, die darauf hinweist, dass keine Ergebnisse gefunden wurden. Man wird aufgefordert, die Suche anzupassen.
2	Schritt 1 und 2 von A002	Schritt 1 und 2 von A002
3	Eine eingeben (evtl. mit Autocomplete) und ENTER drücken oder auf "Abfahrten suchen" klicken.	Falls vorhanden, sollen Abfahrten der Station aufgezeigt werden. Falls es keine Ergebnisse gibt, erscheint die Warnmeldung von A002 Schritt 2.

### 6.2.4 A005

Folgende Schritte sollen auf dem Hauptfenster möglich sein:

Schritt	Ereignis	Ergebnis
1	Alle Schritte von A001 – A003	Alle Schritte von A001 – A003
2	Angabe zweier Stationen eines Datums und einer Uhrzeit bei der Funktion Verbindung suchen. Bestätigen mit ENTER.	Die Verbindungen sollen (je nachdem, welcher Radiobutton (Abfahrt/Ankunft) selektiert ist) von einem bestimmten Zeitpunkt angezeigt werden.
3	Angabe zweier Stationen eines Datums und einer Uhrzeit bei der Funktion Abfahrten suchen. Bestätigen mit ENTER.	Die Abfahrten sollen von einem bestimmten Zeitpunkt angezeigt werden.

### 6.2.5 A006

Folgende Schritte sollen auf dem Hauptfenster möglich sein:

Schritt	Ereignis	Ergebnis
1	Alle Schritte von A001, A002 und A005	Alle Schritte von A001, A002 und A005
2	Bei Verbindungsergebnissen ENTER drücken oder Doppelklick auf ein Ergebnis machen.	Detailinformationen zur Verbindung erscheinen in einem separaten Fenster.

Folgende Schritte sollen auf dem Verbindungsfenster möglich sein:

3	Klick auf einen der "Ort anzeigen" Buttons oder drücken von F1/F2	Im Browser wird die Position der Station angezeigt.
---	---	---

## **7      Anleitung**



## **8 Schlusswort**