



# FAHRPLAN APP

ÜK318 - Analysieren und objektbasiert  
programmieren

## Inhalt

1	Einführung .....	2
1.1	Einleitung .....	2
1.2	Zweck des Dokumentes .....	2
1.3	Ziel .....	2
2	GUI-Design .....	2
2.1	Einleitung .....	2
2.2	Ursprüngliches GUI-Design .....	2
2.3	Finales GUI .....	3
3	Anforderungen .....	4
3.1	Vorgegebene Anforderungen .....	4
3.2	Umgesetzte Anforderungen .....	4
3.2.1	A001 / A004 .....	4
3.2.2	A002 / A005 .....	5
3.2.3	A003 .....	5
3.2.4	A006 .....	5
4	Testfälle .....	6
4.1	Einleitung .....	6
4.2	Verbindungen suchen .....	6
4.3	Alle Verbindungen ab einer Station suchen .....	6
4.4	Stationskarte anzeigen .....	6
5	Installation .....	7
6	Diagramme .....	7
6.1	Use-Case Diagramm .....	7
6.2	Aktivitätsdiagramm .....	7
7	Schlusswort .....	8
7.1	Fazit .....	8

# 1 Einführung

## 1.1 Einleitung

In diesem ÜK haben wir alle notwendigen Sachen gelernt, um dieses Projekt zu vervollständigen. Das Ziel ist es mit der öffentlichen API-Schnittstelle ÖV-Daten zu holen und mit diesen Daten zu arbeiten. Um diese Projektarbeit zu verwirklichen mussten viele Theorieblöcke gemacht werden. Zum einen mussten zuerst die Grundlagen der Programmiersprache C# erlernt werden, mit der das Projekt realisiert wurde. Danach mussten auch noch verschiedene Diagramme besprochen und gemacht werden. Das sind Aktivitätsdiagramme, Use Case Diagramme und Klassendiagramme. Dies war wichtig, um solche Diagramme am Schluss für die Dokumentation zu erstellen.

## 1.2 Zweck des Dokumentes

Dieses Dokument soll das wichtigste meiner Projektarbeit beschreiben und verständlich machen. Es soll aufzeigen wie meine Applikation aufgebaut ist, und wie diese funktioniert. Verschiedene Aspekte des Programmes sollen auch visuell mit Diagrammen dargestellt werden.

## 1.3 Ziel

Das Ziel bei meiner Projektarbeit war es eine Applikation zu erstellen, die den Anforderungen entspricht oder sogar besser ist. Die Applikation sollte aber nicht nur funktionieren, sondern auch noch weitere Punkte wie das Exception-Handling sauber machen.

# 2 GUI-Design

## 2.1 Einleitung

Zuerst habe ich versucht ein möglichst gutes GUI zu gestalten, bevor ich angefangen habe zu programmieren. Ich versuchte möglichst exakt nach meinem GUI-Entwurf zu arbeiten. Durch diverse Verbesserungen und Optimierungen, die ich erst sah als ich das Programm ausführte und testete, habe ich dann aber doch noch einiges abgeändert.

## 2.2 Ursprüngliches GUI-Design

Window Name

Von Haltestelle, Ort, usw. Nach Haltestelle, Ort, usw. Verbindung suchen

Datum / / Zeit

Luzern BH Gleis 12  
12:06 ————— 12:51

Fahrstrecke

Luzern //bolt 12:06  
|  
Sarnen 12:24  
|  
Sachseln 12:30  
|  
Giswil 12:38  
|  
Kaiserstuhl 12:43  
|  
Lungern //bolt 12:51

Dieses ist auch nicht für alle Funktionen geeignet, die ich schlussendlich implementiert habe. Einige Sachen die ich zu darstellen versuchte, konnte man in Windows Forms nur sehr schwer umsetzen und ergaben dadurch auch keinen Sinn.

## 2.3 Finales GUI

Zeit	Von	Zeit	Nach	Dauer
11:05	Lungem - Gleis 1	11:55	Luzern - Gleis 1	00:50:00
12:05	Lungem - Gleis 1	12:55	Luzern - Gleis 1	00:50:00
13:05	Lungem - Gleis 1	13:55	Luzern - Gleis 1	00:50:00
14:05	Lungem - Gleis 1	14:55	Luzern - Gleis 1	00:50:00

Zeit	Nach	Name	Betreiber
10:51	Baar	S1 21132	SBB
10:54	Basel SBB	IR 2316	SBB
10:57	Langenthal	RE 4468	BLS-bl
10:57	Bern	RE 4468	BLS-bl
10:57	Stans	S4 21440	ZB
11:00	Genève-Aéroport	IR 2518	SBB
11:02	Lenzburg	S9 21940	SBB
11:05	Olten	RE 4718	SBB
11:06	Interlaken Ost	IR 2922	ZB
11:06	Brunnen	S3 21341	SBB
11:10	Zürich HB	IR 2620	SBB
11:10	Engelberg	IR 2968	ZB
11:12	Giswil	S5 21540	ZB
11:14	Sursee	S1 21133	SBB
11:16	Langenthal	S6 21640	BLS-bl

Da ich ein Fan von minimalistischen Designs bin, versuchte ich alles irgendwie so zu gestalten, dass man sich sofort damit zurechtfindet. Eine Group Box für die Eingabe, eine für die Ausgabe und eine, um die Funktion auszuwählen. Die Controls in der Group Box für die Eingabe und in der Group Box für die Ausgabe passen sich der Funktion an, die in der Group Box Funktion ausgewählt ist. Controls werden ausgeblendet, Tabellenspalten sind anders angeordnet und haben andere Namen oder eine Karte wird anstelle von der Tabelle angezeigt.

### 3 Anforderungen

#### 3.1 Vorgegebene Anforderungen

ID	Beschreibung	Priorität
A001	Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.	1
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.	1
A003	Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station voran den sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann.	1
A004	Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann.	2
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.	2
A006	Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht.	3
A007	Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann.	3
A008	Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können.	3

Anforderungen mit Priorität 1 müssen gemacht werden, solche mit Priorität 2 sollten auch gemacht werden und solche mit Priorität 3 sind «nice to have».

#### 3.2 Umgesetzte Anforderungen

##### 3.2.1 A001 / A004

Use Case	Station suchen
<b>Beschreibung</b>	User kann nach Station suchen. Die Namen werden automatisch vorgeschlagen.
<b>Akteur(e)</b>	User
<b>Auslöser</b>	User möchte eine Station eingeben.
<b>Vorbedingung</b>	Programm offen, Verbindung zum Internet, Verbindung zur API
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Textfeld Start oder Endstation auswählen</li> <li>2. Station / Suchbegriff eingeben</li> <li>3. Die richtige vorgeschlagene Station auswählen</li> <li>4. Enter drücken</li> </ol>
<b>Alternativer Ablauf #1</b>	<ol style="list-style-type: none"> <li>1. Textfeld Start oder Endstation auswählen</li> </ol>

	<ol style="list-style-type: none"> <li>2. Station / Suchbegriff eingeben</li> <li>3. Die richtige vorgeschlagene Station auswählen</li> <li>4. Auf die gewünschte Station klicken</li> </ol>
<b>Ergebnis</b>	Die Station wird ausgewählt und in das Textfeld komplett hineingeschrieben, Der Focus liegt beim nächsten Control.

## 3.2.2 A002 / A005

<b>Use Case</b>	<b>Verbindungen suchen und anzeigen</b>
<b>Beschreibung</b>	User kann 4 Verbindungen anzeigen lassen.
<b>Akteur(e)</b>	User
<b>Auslöser</b>	User möchte Fahrplan Infos haben.
<b>Vorbedingung</b>	Programm offen, Verbindung zum Internet, Verbindung zur API
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Textfeld Start und Endstation auswählen</li> <li>2. Jeweils Station / Suchbegriff eingeben und Station auswählen</li> <li>3. Enter drücken</li> </ol>
<b>Alternativer Ablauf #1</b>	<ol style="list-style-type: none"> <li>1. Textfeld Start und Endstation auswählen</li> <li>2. Jeweils Station / Suchbegriff eingeben und Station auswählen</li> <li>3. Zeit und Datum auswählen</li> <li>4. Auf «Suchen» klicken</li> </ol>
<b>Ergebnis</b>	Es werden die nächsten 4 Verbindungen angezeigt.

## 3.2.3 A003

<b>Use Case</b>	<b>Abfahrtsplan anzeigen</b>
<b>Beschreibung</b>	Eine Liste von Verbindungen ab einer bestimmten Station anzeigen.
<b>Akteur(e)</b>	User
<b>Auslöser</b>	User möchte eine Art Abfahrtstafel haben.
<b>Vorbedingung</b>	Programm offen, Verbindung zum Internet, Verbindung zur API
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Funktion «Abfahrtsplan» auswählen</li> <li>2. Station eingeben</li> <li>3. Enter drücken</li> </ol>
<b>Alternativer Ablauf #1</b>	<ol style="list-style-type: none"> <li>5. Station eingeben</li> <li>6. Funktion «Abfahrtsplan» auswählen</li> <li>7. Auf «Suchen» klicken</li> </ol>
<b>Ergebnis</b>	Es werden die nächsten paar Verbindungen angezeigt.

## 3.2.4 A006

<b>Use Case</b>	<b>Stationskarte anzeigen</b>
<b>Beschreibung</b>	Eine Karte von der Umgebung eine Station anzeigen.
<b>Akteur(e)</b>	User
<b>Auslöser</b>	User möchte sich anhand einer Karte zu einer Station einen Überblick verschaffen.
<b>Vorbedingung</b>	Programm offen, Verbindung zum Internet, Verbindung zur API
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>4. Funktion «Stationskarte» auswählen</li> <li>5. Station eingeben</li> <li>6. Enter drücken</li> </ol>
<b>Alternativer Ablauf #1</b>	<ol style="list-style-type: none"> <li>8. Station eingeben</li> </ol>

	9. Funktion «Stationskarte» auswählen 10. Auf «Suchen» klicken
<b>Ergebnis</b>	Es wird eine Karte von der Station gezeigt.

## 4 Testfälle

### 4.1 Einleitung

Für die das Testen gibt es gewisse Vorbedingungen. Diese Vorbedingungen sind bei allen Tests gleich. Der Benutzer muss eine laufende Internetverbindung haben und das Programm muss vollständig und korrekt auf dem Gerät des Users installiert sein. Auch sind natürlich alle Testergebnisse erfolgreich, sonst wäre mein Programm ja nicht fertig.

### 4.2 Verbindungen suchen

SCHRITT	AKTIVITÄT	ERWARTETES RESULTAT
1	Applikation starten	Die Applikation soll, ohne abzustürzen und ohne Fehlermeldung starten.
2	Stationsnamen eingeben	Direkt bei der Eingabe sollte sich die automatische Vervollständigung aktivieren und Vorschläge für Stationen machen. Man soll die Station auswählen können.
3	Datum und Zeit eingeben	Datum sollte mit DatePicker auswählbar sein, Zeit kann in TextBox eingegeben werden.
4	Suchen klicken / mit Enter bestätigen	Es werden die nächsten 4 Verbindungen angezeigt mit Abfahrt- / Ankunftszeit, Dauer, Stationen und Gleis Informationen.

### 4.3 Alle Verbindungen ab einer Station suchen

SCHRITT	AKTIVITÄT	ERWARTETES RESULTAT
1	Applikation starten	Die Applikation soll, ohne abzustürzen und ohne Fehlermeldung starten.
2	Funktion «Abfahrtsplan» auswählen	Alle Controls die nicht mehr benötigt werden sollen ausgeblendet werden. Falls im «Von»-Feld noch etwas drinstand, soll dies immer noch drinstehen.
3	Stationsname eingeben	Es soll auch mit der automatischen Vervollständigung gearbeitet werden können.
4	Suchen klicken / mit Enter bestätigen	Alle Abfahrten ab der bestimmten Station sollen gezeigt werden.

### 4.4 Stationskarte anzeigen

SCHRITT	AKTIVITÄT	ERWARTETES RESULTAT
1	Applikation starten	Die Applikation soll, ohne abzustürzen und ohne Fehlermeldung starten.

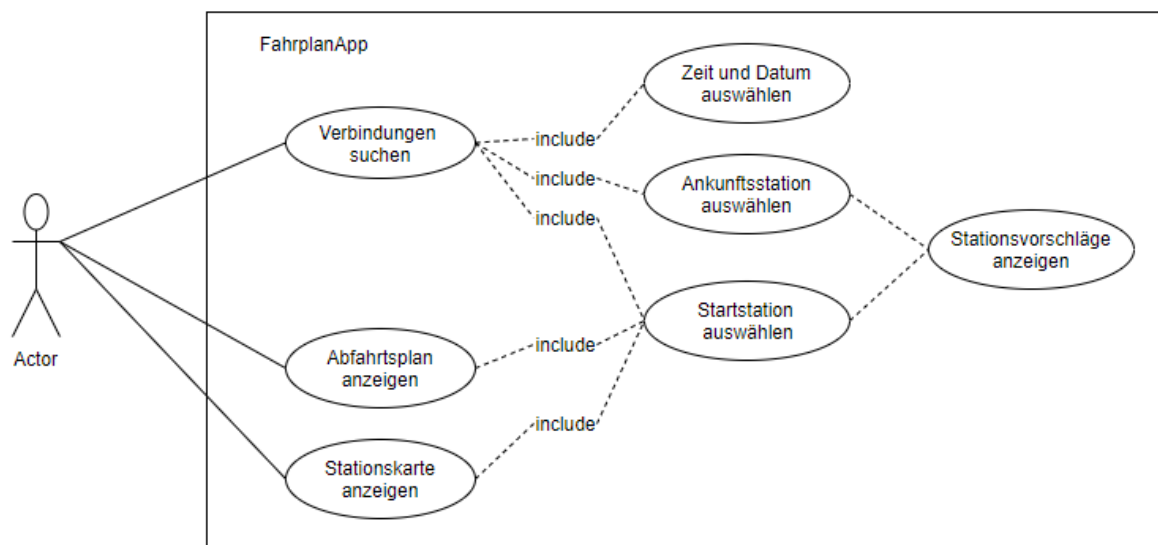
2	Funktion «Stationskarte» auswählen	Alle Controls die nicht mehr benötigt werden sollen ausgeblendet werden. Falls im «Von»-Feld noch etwas drinstand, soll dies immer noch drinstehen.
3	Stationsname eingeben	Es soll auch mit der automatischen Vervollständigung gearbeitet werden können.
4	Suchen klicken / mit Enter bestätigen	Eine Karte von der Umgebung der Station soll angezeigt werden.

## 5 Installation

## 6 Diagramme

### 6.1 Use-Case Diagramm

Dieses Diagramm zeigt die einzelnen Use-Cases von einem User, also die Verwendungsfällen eines Users und die dazugehörige Komponente.



### 6.2 Aktivitätsdiagramm

Die Aktivitätsdiagramme zeigen den Ablauf bei den drei Hauptfunktionen meines Programmes.

#### Verbindungen anzeigen





### Abfahrtsplan anzeigen



### Stationskarte anzeigen



## 7 Schlusswort

### 7.1 Fazit

Dieser ÜK war bis jetzt einer der interessantesten, vor allem weil es endlich so richtig ums Programmieren ging, auch wenn mit Windows Forms. Die meisten Grundlagen für diesen ÜK waren schon vorhanden. Aber bis jetzt haben wir in der Schule noch nie so ein grösseres Projekt umgesetzt. Auch einige Tools wie GitHub oder solche zum Diagramme machen haben wir neu kennen gelernt.