Gasser Cedric

INF18a / BMLT19a

fahrplan app

ÜK318 - Analysieren und objektbasiert programmieren

Inhalt

[1 Einführung 2](#_Toc27569856)

[1.1 Einleitung 2](#_Toc27569857)

[1.2 Zweck des Dokumentes 2](#_Toc27569858)

[1.3 Ziel 2](#_Toc27569859)

[2 GUI-Design 2](#_Toc27569860)

[2.1 Einleitung 2](#_Toc27569861)

[2.2 Ursprüngliches GUI-Design 2](#_Toc27569862)

[2.3 Finales GUI 3](#_Toc27569863)

[3 Anforderungen 4](#_Toc27569864)

[3.1 Vorgegebene Anforderungen 4](#_Toc27569865)

[3.2 Umgesetzte Anforderungen 4](#_Toc27569866)

[3.2.1 A001 / A004 4](#_Toc27569867)

[3.2.2 A002 / A005 5](#_Toc27569868)

[3.2.3 A003 5](#_Toc27569869)

[3.2.4 A006 5](#_Toc27569870)

[4 Testfälle 6](#_Toc27569871)

[4.1 Einleitung 6](#_Toc27569872)

[4.2 Verbindungen suchen 6](#_Toc27569873)

[4.3 Alle Verbindungen ab einer Station suchen 6](#_Toc27569874)

[4.4 Stationskarte anzeigen 6](#_Toc27569875)

[5 Installation 7](#_Toc27569876)

[6 Deinstallation 7](#_Toc27569877)

[7 Diagramme 7](#_Toc27569878)

[7.1 Use-Case Diagramm 7](#_Toc27569879)

[7.2 Aktivitätsdiagramm 7](#_Toc27569880)

[7.3 Klassendiagramm 7](#_Toc27569881)

# Einführung

## 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 Cas Diagramme und Klassendiagramme. Dies war wichtig, um solche Diagramme am Schluss für die Dokumentation zu erstellen.

## 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.

## Ziel

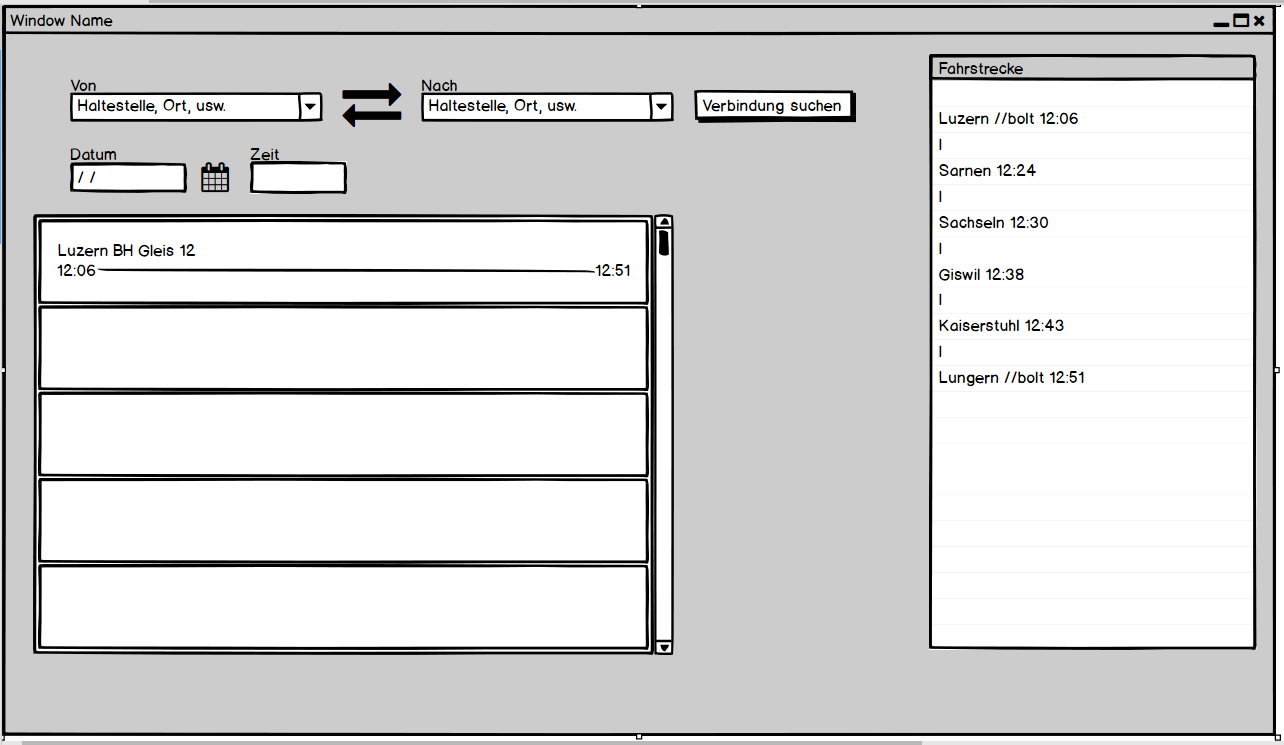
Das Ziel bei meiner Projektarbeit war es eine Applikation zu erstellen, die den Anforderungen entspricht oder sogar besser ist. Die Applikation sollte aber nicht nur funktionieren, sondern auch noch weitere Punkte wie das Exception-Handling sauber machen.

# GUI-Design

## Einleitung

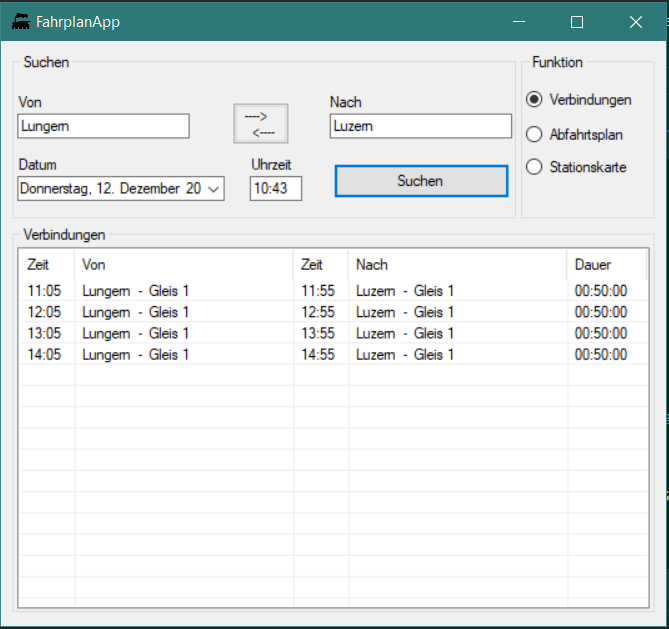
Zuerst habe ich versucht ein möglichst gutes GUI zu gestalten, bevor ich angefangen habe zu programmieren. Ich versuchte möglichst exakt nach meinem GUI-Entwurf zu arbeiten. Durch diverse Verbesserungen und Optimierungen, die ich erst sah als ich das Programm ausführte und testete, habe ich dann aber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

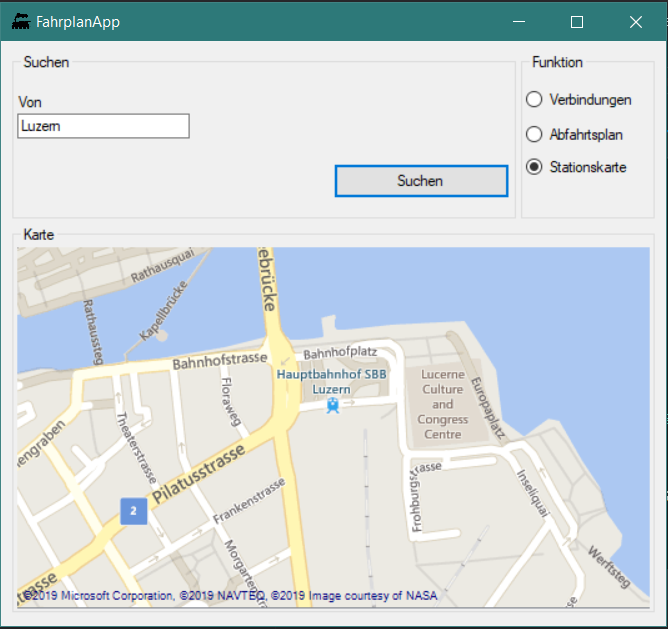
## Ursprüngliches GUI-Design



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.

## Finales GUI





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.

# Anforderungen

## 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».

## Umgesetzte Anforderungen

### A001 / A004

|  |  |
| --- | --- |
| Use Case | Station suchen |
| Beschreibung | User kann nach Station suchen. Die Namen werden automatisch vorgeschlagen. |
| Akteur(e) | User |
| Auslöser | User möchte eine Station eingeben. |
| Vorbedingung | Programm offen, Verbindung zum Internet, Verbindung zur API |
| Ablauf | 1. Textfeld Start oder Endstation auswählen 2. Station / Suchbegriff eingeben 3. Die richtige vorgeschlagene Station auswählen 4. Enter drücken |
| Alternativer Ablauf #1 | 1. Textfeld Start oder Endstation auswählen 2. Station / Suchbegriff eingeben 3. Die richtige vorgeschlagene Station auswählen 4. Auf die gewünschte Station klicken |
| Ergebnis | Die Station wird ausgewählt und in das Textfeld komplett hineingeschrieben, Der Focus liegt beim nächsten Control. |

### A002 / A005

|  |  |
| --- | --- |
| Use Case | Verbindungen suchen und anzeigen |
| Beschreibung | User kann 4 Verbindungen anzeigen lassen. |
| Akteur(e) | User |
| Auslöser | User möchte Fahrplan Infos haben. |
| Vorbedingung | Programm offen, Verbindung zum Internet, Verbindung zur API |
| Ablauf | 1. Textfeld Start und Endstation auswählen 2. Jeweils Station / Suchbegriff eingeben und Station auswählen 3. Enter drücken |
| Alternativer Ablauf #1 | 1. Textfeld Start und Endstation auswählen 2. Jeweils Station / Suchbegriff eingeben und Station auswählen 3. Zeit und Datum auswählen 4. Auf «Suchen» klicken |
| Ergebnis | Es werden die nächsten 4 Verbindungen angezeigt. |

### A003

|  |  |
| --- | --- |
| Use Case | Abfahrtsplan anzeigen |
| Beschreibung | Eine Liste von Verbindungen ab einer bestimmten Station anzeigen. |
| Akteur(e) | User |
| Auslöser | User möchte eine Art Abfahrtstafel haben. |
| Vorbedingung | Programm offen, Verbindung zum Internet, Verbindung zur API |
| Ablauf | 1. Funktion «Abfahrtsplan» auswählen 2. Station eingeben 3. Enter drücken |
| Alternativer Ablauf #1 | 1. Station eingeben 2. Funktion «Abfahrtsplan» auswählen 3. Auf «Suchen» klicken |
| Ergebnis | Es werden die nächsten paar Verbindungen angezeigt. |

### A006

|  |  |
| --- | --- |
| Use Case | Stationskarte anzeigen |
| Beschreibung | Eine Karte von der Umgebung eine Station anzeigen. |
| Akteur(e) | User |
| Auslöser | User möchte sich anhand einer Karte zu einer Station einen Überblick verschaffen. |
| Vorbedingung | Programm offen, Verbindung zum Internet, Verbindung zur API |
| Ablauf | 1. Funktion «Stationskarte» auswählen 2. Station eingeben 3. Enter drücken |
| Alternativer Ablauf #1 | 1. Station eingeben 2. Funktion «Stationskarte» auswählen 3. Auf «Suchen» klicken |
| Ergebnis | Es wird eine Karte von der Station gezeigt. |

# Testfälle

## Einleitung

## 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. |

## 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. |

## 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. |

# Installation

# Deinstallation

# Diagramme

## Use-Case Diagramm

## Aktivitätsdiagramm

## Klassendiagramm