

# Dokumentation ÖV-Fahrplan

ÖV-Fahrplan

Verbindungen suchen

Von:

Nach:

Datum: 18.12.2019

Uhrzeit: 13:53

☒ Verbindungsplan ☐ Verbindungen ab Station

Verbindungen

Von	Nach	Zeit	Zug/Bus Nr	Dauer	

**Florian Stettler**

ÜK M318

18.12.2019

## Inhaltsverzeichnis

Management Summary .....	3
Zweck .....	3
Anforderungen des Programms .....	3
Anforderungen Funktionen .....	3
Nicht umgesetzte Anforderungen .....	3
UI-Mockups .....	4
Hauptseite .....	4
Maps-Fenster .....	5
Unterschiede zum Endprodukt .....	5
Use Cases .....	6
Use Case Diagramm .....	6
Use Case Beschreibung .....	6
Anforderung 001/004 .....	6
Anforderung 002/005 .....	7
Anforderung 003 .....	7
Aktivitätendiagramme .....	7
Anforderung 001 .....	7
Anforderung 002 .....	7
Anforderung 003 .....	8
Anforderung 004 .....	8
Anforderung 005 .....	8
Anforderung 006 .....	8
Testing .....	9
Testfall 1: Stationen suchen .....	9
Testfall 2: Gefundene Verbindungen anzeigen .....	9
Testfall 3: Verbindungen ab Station anzeigen .....	10
Testfall 4: Stationen auf Karte anzeigen .....	10
Testfall 5: Auf Karte nach Stationen suchen .....	10
Ergebnis der Tests .....	11
Installationsanleitung .....	11
Deinstallieren .....	11
Einbindung der Karte in das Programm .....	12
Schlusswort .....	12

## Management Summary

Im Rahmen des ÜK 318 mussten wir eine Fahrplan Applikation auf Basis der zur Verfügung gestellten Bibliothek erstellen. Die Bibliothek greift auf die Daten der «Swiss Public Transport API» von OpenData zu. Das Ziel des Projektes war es, die Kenntnisse in der OOP zu verbessern und einen Einblick in die Entwicklung einer Applikation (mit Planung, etc.) zu erhalten. Im Fokus stand auch die Verwendung der oben erwähnten Web-API und wie man damit in einem Windows Forms Programm umgeht.

## Zweck

Der Zweck dieses Dokuments ist es, das von mir entwickelte Programm «ÖV-Fahrplan» zu Erklären und die Entstehung zu dokumentieren. Auf den nachfolgenden Seiten werden Sie mein Programm näher kennenlernen und die nötigen Informationen für die Bedienung des Programms erhalten.

## Anforderungen des Programms

Um das Programm zu entwickeln haben wir eine Liste mit Anforderungen, die der Kunde erstellt hat, erhalten. Es gab drei Anforderungen der Priorität eins, die umgesetzt werden müssen. Weiter gab es noch Anforderungen der Priorität zwei, die ebenfalls umgesetzt werden müssen und die Funktionalität und Bedienbarkeit des Programms deutlich verbessern. Die Anforderungen der Priorität drei sind «nice to have» und müssen nicht zwingend umgesetzt werden.

## Anforderungen Funktionen

Ich habe folgende Funktionen in meinem Programm umgesetzt.

- **A001:** Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. Eine Verbindungsanzeige welche Verbindungen zwischen 2 gewählten Standorten zu einer gewünschten Zeit anzeigt.
- **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.
- **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.
- **A004:** Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such Resultate erscheinen, damit ich effizienter nach Stationen suchen kann.
- **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.
- **A006:** Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht.

## Nicht umgesetzte Anforderungen

Folgende Funktionen habe ich nicht umgesetzt.

- **A007:** Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann.
- **A008:** Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können.

## UI-Mockups

Bevor ich mit dem Erstellen des Programms gestartet habe, habe ich Mockups für die Oberfläche (UI) des Programms erstellt. Das sollte mir die Arbeit vereinfachen und schon ab Anfang aufzeigen, wie das Endprodukt schlussendlich aussehen wird.

### Hauptseite


ÖV - Fahrplan

Von:

Nach:

Datum:

Zeit:



Verbindungsplan

Verbindungen ab Station

Verbindung suchen

Verbindungen von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

Von XXX nach XXX

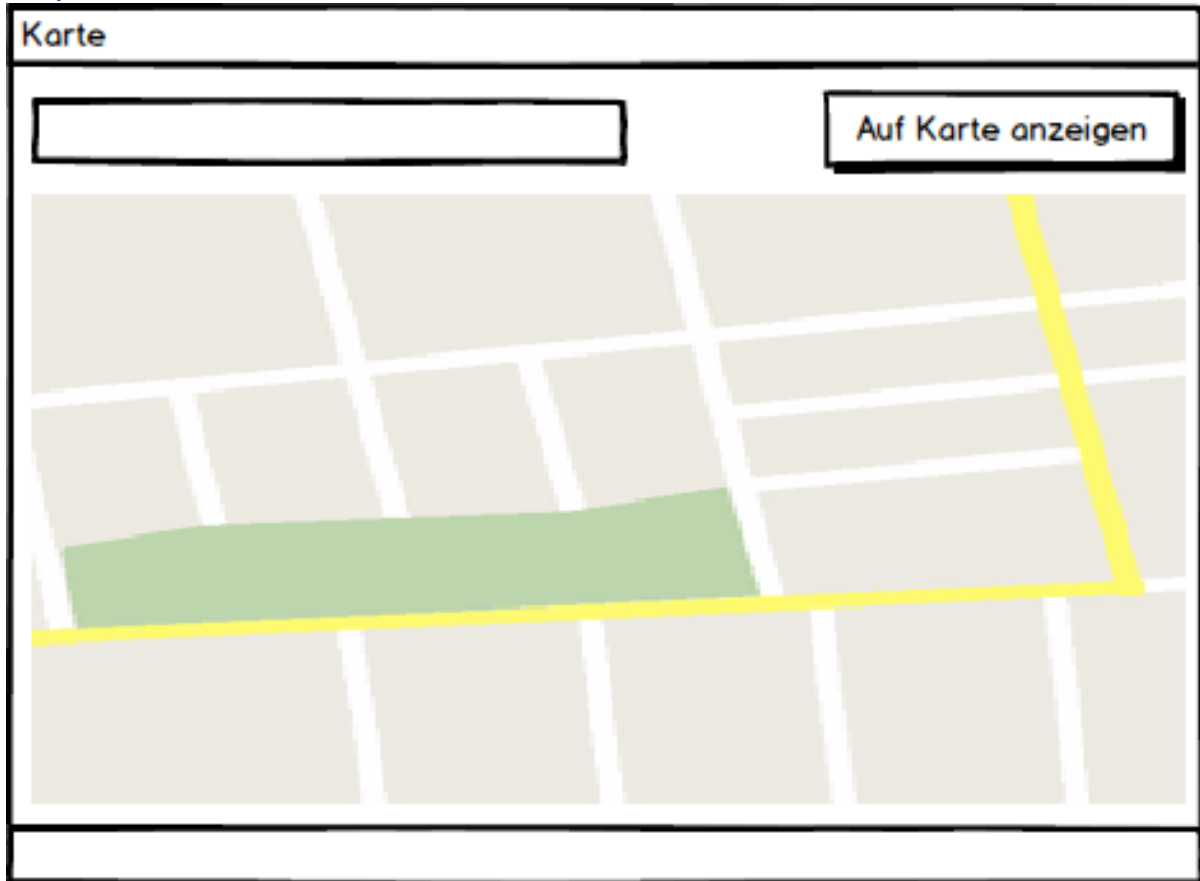
Auf Karte suchen

per Mail versenden

Beenden

Unterhalb der TextBox für den Startort und den Zielort, ist noch eine ListBox, die ausgeblendet ist und eingeblendet wird, sobald man mit der Suche nach Stationen beginnt. Nachdem man die richtige Station gefunden hat, wird diese ListBox wieder ausgeblendet.

## Maps-Fenster



Wie bei der Hauptseite, hat es auch Unterhalb der TextBox um den Ort einzugeben noch eine ListBox, die ebenfalls ausgeblendet ist und nur bei der Suche nach Stationen eingeblendet wird.

### Unterschiede zum Endprodukt

Die Mockups und das Endprodukt unterscheiden sich nur geringfügig. Kleinere Unterschiede sind z.B.

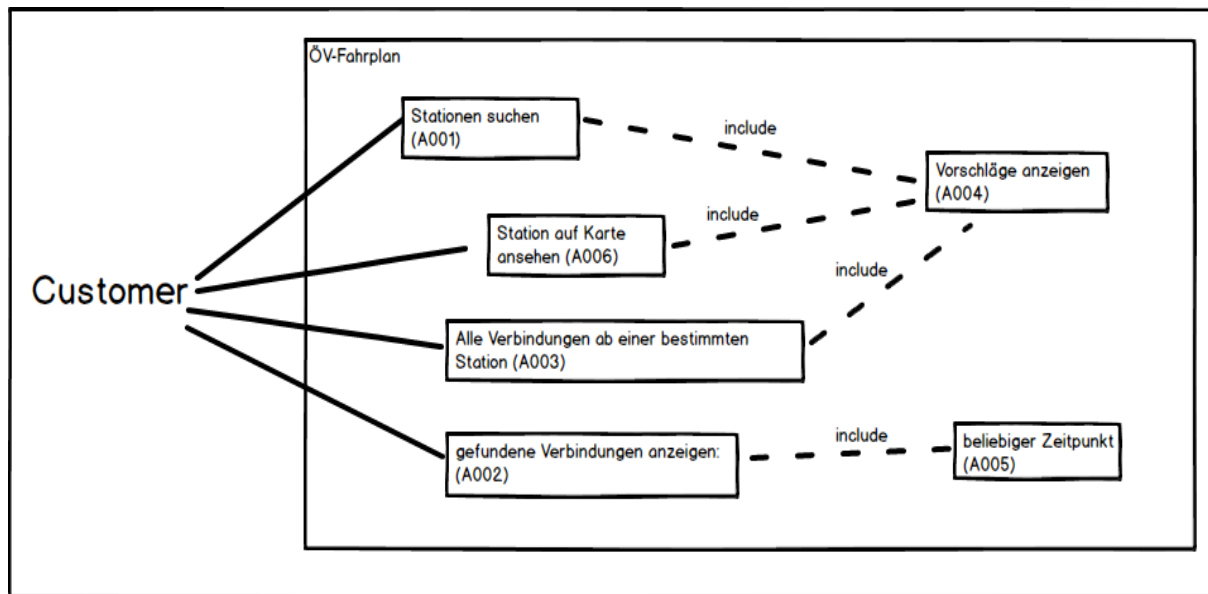
- Der Button um die gefunden Verbindungen anzuzeigen ist verschwunden, weil ich diese Anforderung nicht umgesetzt habe.
- Die ListView in der alle Verbindungen angezeigt werden, ist im Endprodukt noch in einer GroupBox und nicht wie im Mockup einfach so auf dem Form.

## Use Cases

Um das Programmieren einfacher zu machen, kann man die Use-Cases, für die das Programm gebraucht wird, in einem Diagramm darstellen und auch eine detailliertere Beschreibung machen. Für mein Programm ÖV-Fahrplan habe ich das so gemacht und nachfolgend sehen Sie meine Ergebnisse.

### Use Case Diagramm

Aus der Übersicht der Anforderungen kann man ein Use-Case Diagramm machen. Dort werden alle Anforderungen und Funktionen, die ich im Programm implementiert habe,



übersichtlich dargestellt.

### Use Case Beschreibung

In der Beschreibung der Use Cases, sieht man detailliert welche Funktion wie gebraucht wird und kann sie dann so nach diesen Anforderungen programmieren.

#### Anforderung 001/004

Use Case A001	Stationen suchen
<b>Beschreibung</b>	Der Benutzer möchte nach Stationen suchen und es sollen Ergebnisse dafür angezeigt werden.
<b>Akteur(e)</b>	Der Benutzer der ÖV-Fahrplan Applikation.
<b>Auslöser</b>	Die Person möchte eine Station suchen.
<b>Vorbedingung</b>	Eine Internetverbindung ist notwendig.
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Radio Button „Verbindungsplan anklicken</li> <li>2. Stationsnamen eingeben</li> <li>3. Passende Vorschläge für eingegeben Station werden angezeigt.</li> <li>4. Gewünschte Station auswählen</li> </ol>
<b>Alternativer Ablauf</b>	Kein
<b>Ergebnis</b>	Passende Stationsvorschläge des eingegebenen Textes werden angezeigt und ausgewählte Station wird in TextBox geschrieben.

## Anforderung 002/005

Use Case A002	Gefundene Verbindungen anzeigen
<b>Beschreibung</b>	Der Benutzer möchte 4-5 Verbindungen von der Start zur Endstation angezeigt bekommen
<b>Akteur(e)</b>	Der Benutzer der ÖV-Fahrplan Applikation.
<b>Auslöser</b>	Der Benutzer möchte einen idealen Anschluss bekommen
<b>Vorbedingung</b>	Eine Internetverbindung ist notwendig.
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Datum und Uhrzeit festlegen</li> <li>2. Radio Button „Verbindungen ab Station“ anklicken</li> <li>3. Start Station eingeben</li> <li>4. Button „Abfahrt suchen“ klicken</li> <li>5. Die nächsten Verbindungen ab der angegebenen Station werden angezeigt</li> </ol>
<b>Alternativer Ablauf</b>	Kein
<b>Ergebnis</b>	In der ListView werden die nächsten 5 Verbindungen ab dieser Station angezeigt.

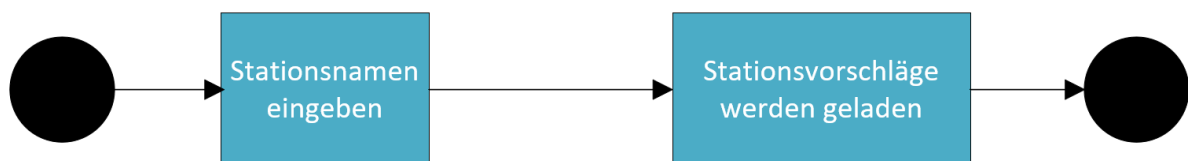
## Anforderung 003

Use Case A003	Alle Verbindung ab einer Station
<b>Beschreibung</b>	Der Benutzer sucht nach einer beliebigen Station und alle Verbindungen ab dieser Station werden angezeigt.
<b>Akteur(e)</b>	Der Benutzer der ÖV-Fahrplan Applikation.
<b>Auslöser</b>	Die Person möchte eine Zug/Bus Verbindung suchen
<b>Vorbedingung</b>	Eine Internetverbindung ist notwendig.
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>6. Stationsnamen eingeben</li> <li>7. Button „Verbindungen suchen“ klicken</li> <li>8. Verbindungen ab der gewünschten Station werden angezeigt</li> </ol>
<b>Alternativer Ablauf</b>	Kein
<b>Ergebnis</b>	Alle Verbindungen ab der gewünschten Station werden angezeigt.

## Aktivitätendiagramme

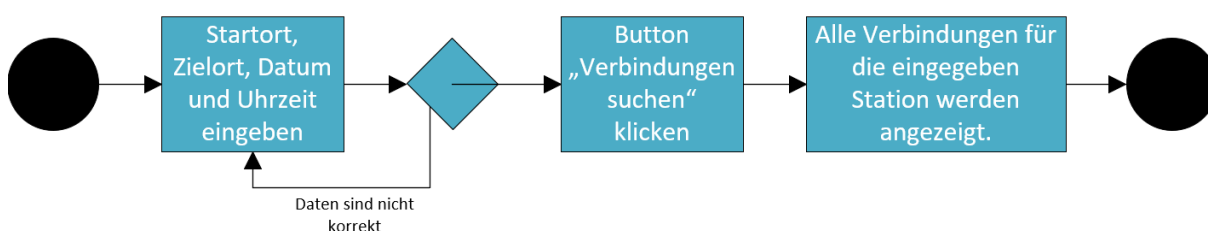
## Anforderung 001

Der Benutzer möchte nach Stationen suchen und es sollen Ergebnisse dafür angezeigt werden.



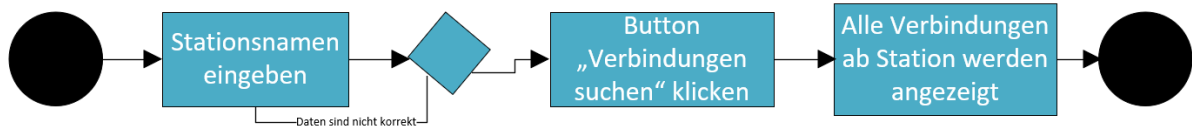
## Anforderung 002

Der Benutzer möchte 4-5 Verbindungen vom Start



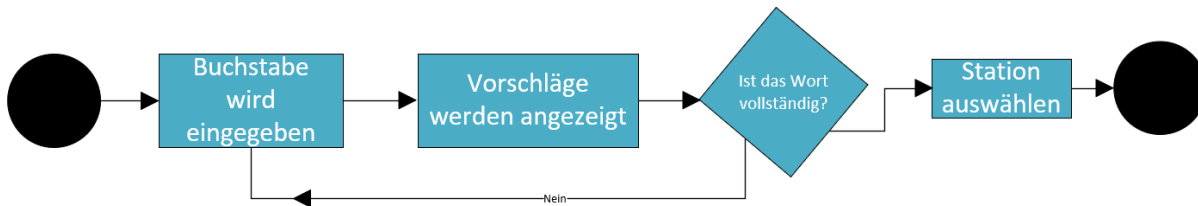
### Anforderung 003

Der Benutzer sucht nach einer beliebigen Station und alle Verbindungen ab dieser Station werden angezeigt.



### Anforderung 004

Der Benutzer möchte schon beim Eintippen die ersten Ergebnisse für Stationen erhalten.



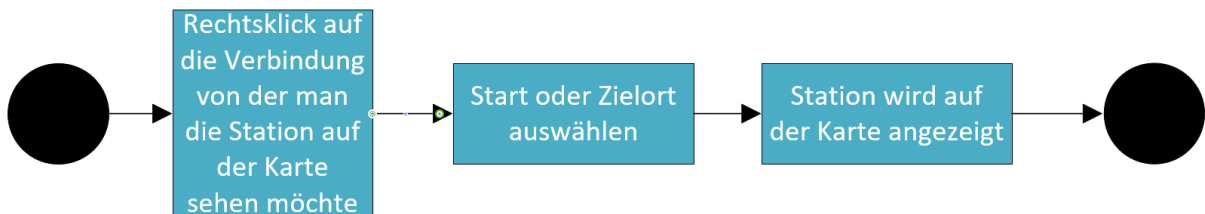
### Anforderung 005

Der Benutzer möchte eine Verbindung zu einem beliebigen Zeitpunkt suchen, damit er seine Reisen besser planen kann.

Diese Anforderung wurde bereits in das Aktivitätendiagramm der Anforderung 002 implementiert.

### Anforderung 006

Der Benutzer möchte die Station auf der Karte anschauen, damit er sich einen Überblick schaffen kann.









## Testing

Damit das Programm möglichst keine Fehler hat, sollte es getestet werden. Dafür werden verschiedene Testfälle definiert und anschliessend ausgeführt. Es ist wichtig, dass für jeden Testfall das erwartete Resultat dasselbe ist, wie das tatsächliche Resultat. Das Testing sollte nicht erst am Ende gemacht werden, sondern immer wieder während dem Schreiben des Codes.

### Testfall 1: Stationen suchen

Durchgeführt von: Yazdan Musa



Durchgeführt am: 18.12.2019


Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	Benutzer beginnt den Namen der Station einzugeben.	Es werden Stationsvorschläge angezeigt, die zu den eingegebenen Buchstaben passen.	Passende Stationsvorschläge werden angezeigt.	
2	Bei jedem weiteren Eingegebenen Buchstaben wird die Auswahl an Vorschlägen automatisch aktualisiert.	Automatische Aktualisierung der Auswahl	Die Auswahl wird automatisch aktualisiert beim Eingeben von weiteren Buchstaben.	
3	Gewünschte Station wird aus dem DropDown Menu ausgewählt und erscheint in der TextBox.	Station erscheint in der TextBox	TextBox wird mit dem Stationsnamen befüllt.	
4	DropDown verschwindet, nachdem die Station ausgewählt wurde.	DropDown verschwindet.	DropDown verschwindet	

### Testfall 2: Gefundene Verbindungen anzeigen

Durchgeführt von: Yazdan Musa

Durchgeführt am: 18.12.2019



Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	Benutzer gibt Abfahrtort und Zielort ein	Auswahl von Stationen wird angezeigt.	Stationsauswahl wird angezeigt.	
2	Benutzer gibt Datum und Uhrzeit ein	Datum und Uhrzeit wird in TextBox angezeigt.	Datum und Uhrzeit wird in TextBox angezeigt.	

3	Benutzer klickt auf Verbindungen suchen	4 Verbindungen mit den eingegebenen Angaben werden angezeigt.	4 Verbindungen werden angezeigt.	
---	---	---	----------------------------------	---

### Testfall 3: Verbindungen ab Station anzeigen

Durchgeführt von: Yazdan Musa



Durchgeführt am: 18.12.2019

Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	Benutzer gibt Namen der Station ein	Auswahl von Stationen wird angezeigt.	Stationsauswahl wird angezeigt.	
2	Benutzer klickt auf Button «Abfahrt suchen»	Verbindungen ab gewünschter Station werden angezeigt.	Verbindungen werden angezeigt.	

### Testfall 4: Stationen auf Karte anzeigen

Durchgeführt von: Yazdan Musa



Durchgeführt am: 18.12.2019


Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	Rechtsklick auf die gewünschte Verbindung	Kontext Menü erscheint	Kontext Menü erscheint.	
2	Benutzer klickt entweder auf «Startort auf Karte anzeigen» oder auf «Zielort auf Karte anzeigen»	Zweites Fenster öffnet sich und die gewünschte Station wird auf der Karte angezeigt.	Station wird auf der Karte im zweiten Fenster angezeigt.	

### Testfall 5: Auf Karte nach Stationen suchen

Durchgeführt von: Yazdan Musa

Durchgeführt am: 18.12.2019

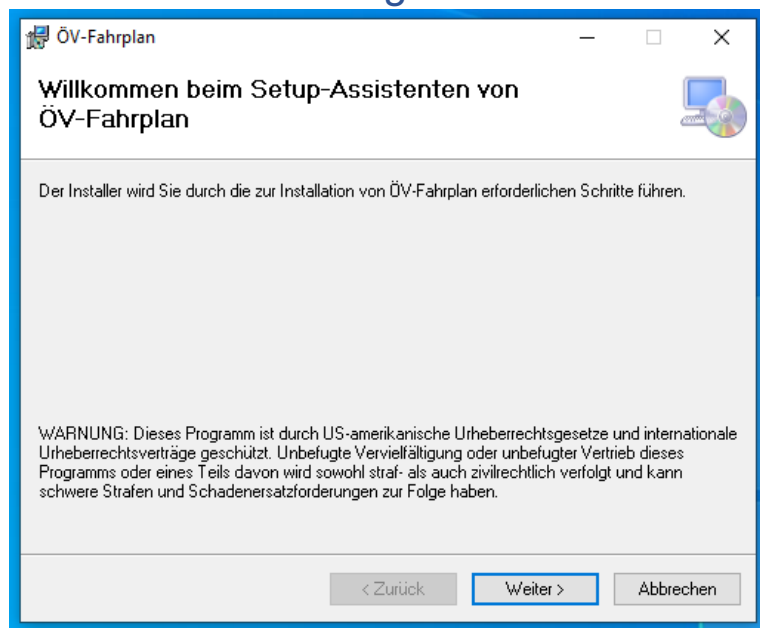
Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	Benutzer klickt auf den Button «Auf Karte suchen»	Zweites Fenster mit einer Karte öffnet sich.	Zweites Fenster hat sich geöffnet.	
2	In TextBox gibt der Benutzer die gewünschte Station ein	Auswahl an Station, die ähnlich wie die Eingabe sind, werden angezeigt.	Stationsauswahl wird angezeigt.	

3	Benutzer klickt auf Button «Auf Karte anzeigen»	Die gewünschte Station wird auf der Karte angezeigt.	Station wird auf der Karte angezeigt.	
---	---	--	---------------------------------------	---

## Ergebnis der Tests

Alle Tests waren erfolgreich. Alles hat so funktioniert wie ich es erwartet habe und es gab keine Fehler. Grund dafür war auch, dass ich während dem Schreiben des Codes immer wieder die Funktionen getestet habe. Dadurch konnte ich die Fehler immer sofort beheben oder ein Workaround finden.

## Installationsanleitung



Um das Programm «ÖV-Fahrplan» zu installieren, öffnen Sie die Datei «ÖV-Fahrplan.msi».

Dann öffnet sich das Fenster auf der linken Seite. Hier klicken sie auf weiter.

Im nächsten Schritt können Sie den Installationspfad wählen und dann auf weiter klicken.

Wenn Sie im nächsten Fenster auf weiter klicken, wird das Programm installiert und es kann benutzt werden.

## Deinstallieren

Um das Programm zu deinstallieren gehen sie in die Systemsteuerung > Programme > Programme und Features. Dort finden Sie in der Liste das Programm «ÖV-Fahrplan». Machen Sie einen Rechtsklick auf den Eintrag und wählen Sie Deinstallieren. Es kommt noch einmal ein Dialog, den Sie mit Ja bestätigen und das Programm ist dann deinstalliert.

## Einbindung der Karte in das Programm

Eine Anforderung (A006) an das Programm war es, dass man die Station auf einer Karte anschauen kann, um sich ein besseres Bild davon zu machen. Zuerst wollte ich diese Anforderung mit dem Steuerelement «WebBrowser» umsetzen. Doch bei der Umsetzung habe ich bemerkt, dass wenn man Google Maps damit aufruft, nicht nur die Karte, sondern auch noch die anderen Elemente auf der Website von Google Maps angezeigt werden. Ich fand diese Lösung nicht sehr schön und wollte Google Maps auch nicht im Browser öffnen, wenn man den Standort der Station ansehen möchte.

Deshalb habe ich im Internet nach einer anderen Lösung gesucht und ein NuGet-Paket namens [GMap.Net.WinForms](#) gefunden. Durch dieses Paket erhält man ein Steuerelement, das die Karte direkt anzeigen kann. Die Daten kann es von einigen verschiedenen Karten-Anbietern holen. Ich habe mich für Google Maps entschieden.

Die Benutzung davon ist sehr einfach gemacht und mit wenigen Zeilen Code kann die Station auf der Karte anzeigen. Dafür sind nur die Koordinaten nötig. Durch die Implementation des NuGet-Pakets konnte ich die Karte direkt in das Programm einbinden und ohne störende Elemente anzeigen.

## Schlusswort

Mir hat das Projekt sehr Spass gemacht. Ich konnte meine Fähigkeiten in C# verbessern und viele neue Dinge lernen. Leider konnte ich nicht alle acht Anforderung in das Projekt implementieren. Doch die wichtigsten, das heisst die mit Priorität eins und zwei, konnte ich gut umsetzen.