



Zug-Fahrplan Applikation ÜK Modul 318

Von Lynn Jung



21. Dezember 2018
BÉNÉDICT SCHULE LUZERN

Inhaltsverzeichnis

1.	Einleitung.....	2
2.	Zweck des Dokuments	3
3.	Vorbereitung.....	3
3.1.	Skizze (Mockup).....	3
3.2.	Use-Case/ Activity-Diagramm	5
4.	Welche Funktionen wurden umgesetzt?	6
5.	Bekannte Bugs	7
6.	Testfälle	8
6.1.	Abänderung Testmethode.....	9
7.	Installations- und Deinstallationsanleitung	9
7.1.	Installation.....	9
7.2.	Deinstallation.....	13
8.	Fazit.....	14

1. Einleitung

In dem ÜK 318 ging es um das Objektorientierte Programmieren mit Windows Forms. Hierbei bekamen wir den Auftrag, eine SBB-Fahrplan Applikation zu erstellen. Der Sinn und Zweck dahinter bestand darin, dass wir unsere erlernten C#- und Visual Studio Kenntnisse optimal verbinden konnten und ein lauffähiges Programm erstellen konnten, welches sowohl Informatiker als auch Laien verstehen können.

In den nachfolgenden Zeilen beschreibe Ich nun meine komplette Arbeit der letzten 4 Tage, indem Ich meinen Fortschritt von der Idee bis hin zur Finalisierung der fertigen Applikation beschreibe und zudem bekannte Fehler erläutere.

2. Zweck des Dokuments

Diese Dokumentation soll Ihnen zeigen wie ich bei der Entwicklung der Desktopapplikation vorgegangen bin und wie diese strukturell aufgebaut ist.

3. Vorbereitung

3.1. Skizze (Mockup)

Fahrplan:

Verbindung suchen

Verbindung suchen

Fahrten

Maps

Schliessen

Von :
Zürich, Flughafen

Nach :
Luzern
Luzern, Bahnhof
Luzern, Gütsch

Datum :
/ /

Verbindung suchen

Datum	Von	Nach	Abfahrt	Ankunft	Gleis	Weiterleiten
19.12.2018	Zürich Flughafen	Luzern	13:40	14:50	Gl. 3	
19.12.2018	Zürich Flughafen	Luzern	14:05	15:12	Gl. 4	
19.12.2018	Zürich Flughafen	Luzern	14:40	15:50	Gl. 3	
19.12.2018	Zürich Flughafen	Luzern	15:05	16:12	Gl. 4	
19.12.2018	Zürich Flughafen	Luzern	15:40	16:50	Gl. 3	

Abfahrtstafel:


Abfahrtstafel

Verbindung suchen

Abfahrtstafel

Maps

Schliessen

 **Abfahrtstafel**

Von :

Luzern

Luzern, Bahnhof
Luzern, Gütsch

Verbindungen suchen

Datum	Von	Nach	Abfahrt	Ankunft	Gleis	Weiterleiten
19.12.2018	Zürich Flughafen	Luzern	13:40	14:50	Gl. 3	
19.12.2018	Zürich Flughafen	Luzern	14:05	15:12	Gl. 4	
19.12.2018	Zürich Flughafen	Luzern	14:40	15:50	Gl. 3	
19.12.2018	Zürich Flughafen	Luzern	15:05	16:12	Gl. 4	
19.12.2018	Zürich Flughafen	Luzern	15:40	16:50	Gl. 3	

Maps:


Maps

Verbindung suchen

Abfahrtstafel

Maps





Schliessen

 **Maps**


Mein Standort :

Verbindungen suchen

Google Maps



<https://www.google.ch/maps/preview>

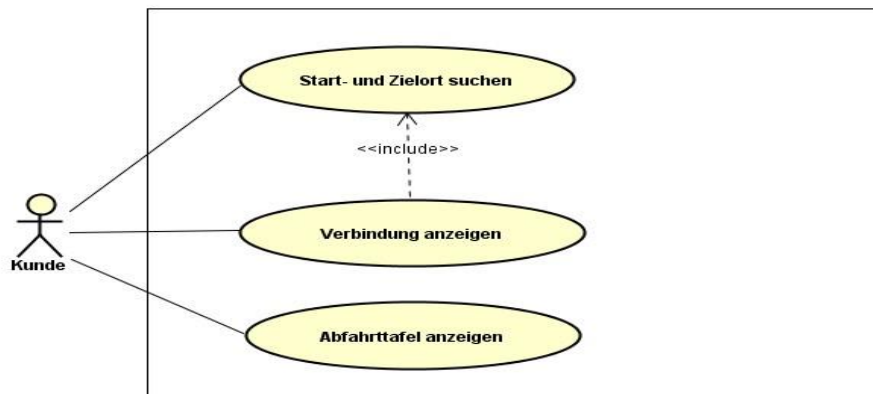


Wenn man nun das fertige Programm ansieht, kann man über alle Seiten gravierende Änderungen sehen dies liegt daran das ich mit dem Design nicht zufrieden war und auch gewisse Features fehlten.

Bevor ich mich dann auf das coden stürzte erstellte ich sogenanntes Use-Case und Activity Diagramme um

3.2. Use-Case/ Activity-Diagramm

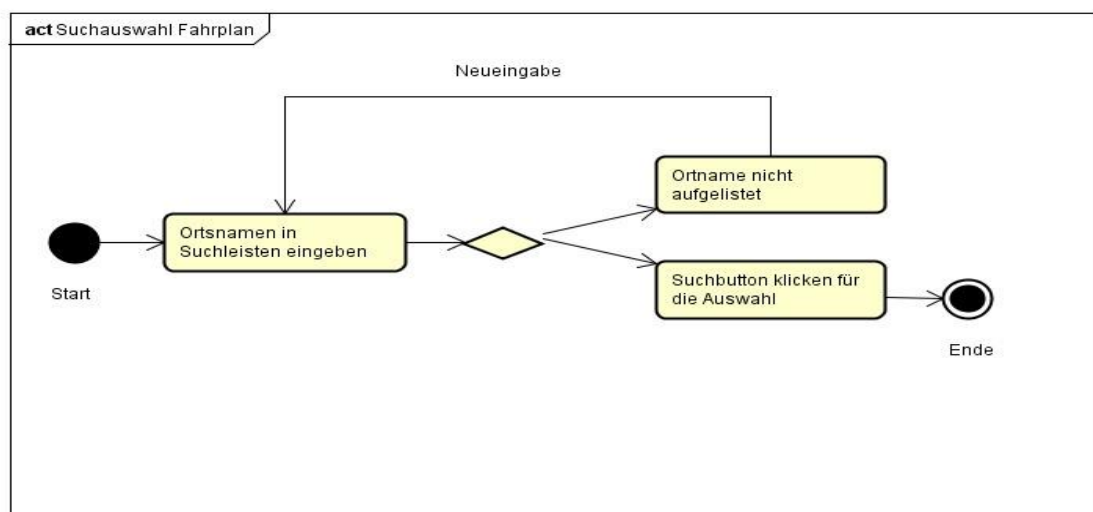
Use Case:



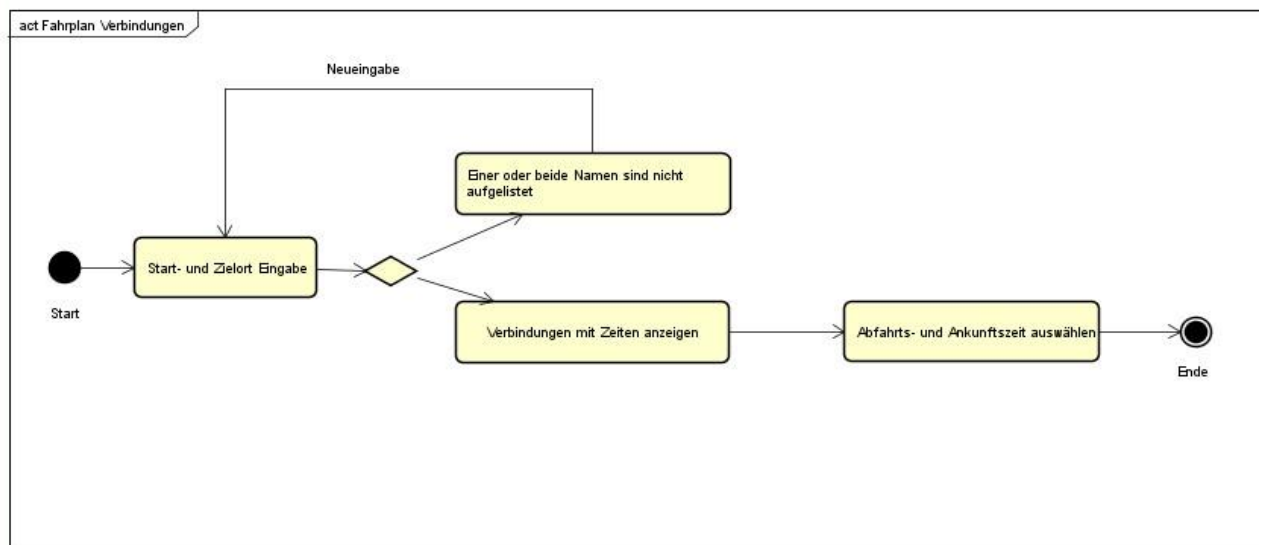
Hier erkennt man die drei Aktivitäten, welche der User mit meiner Applikation ausführen kann. Er kann nach Orten suchen, deren Verbindungen anzeigen lassen sowie mithilfe einer Abfahrtsfel sich mögliche Zielorte und Abfahrtszeiten dahin ansehen.

Die jeweiligen 3 Aktionen sind in den Activity- Diagrammen erkennbar: 1.

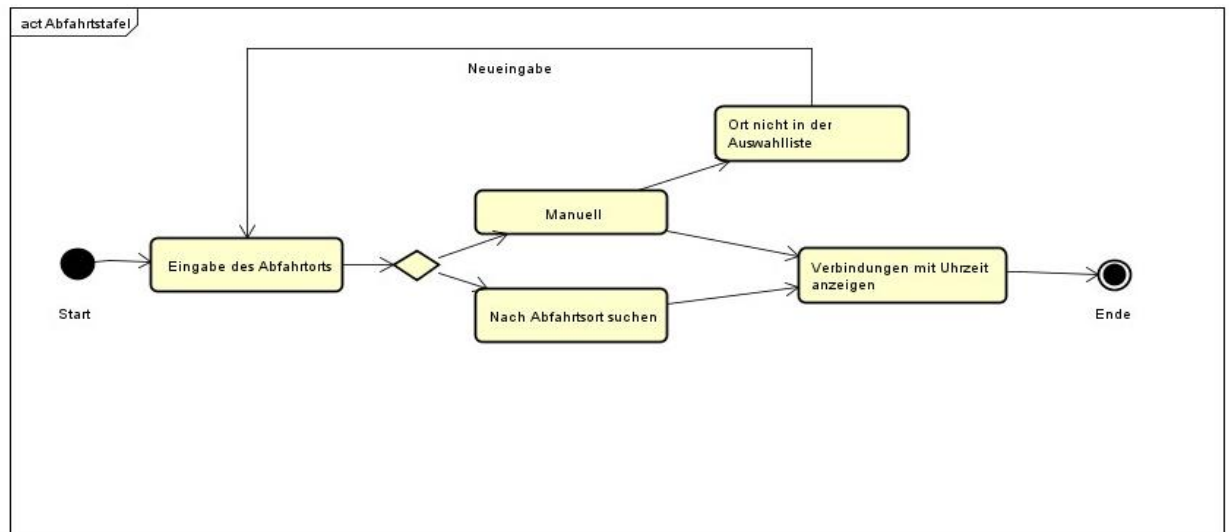
Abfahrts- und Zielort suchen



2. Verbindungen anzeigen



3. Abfahrtstafel anzeigen



4. Welche Funktionen wurden umgesetzt?

ID	Beschreibung	Priorität
A001	Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.	✓
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.	✓
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.	✓

Ich konnte alle Funktionen erfolgreich umsetzen.

5. Bekannte Bugs

- Ich konnte leider in der Combobox vom Abfahrtsort die Liste aller Stationen nicht direkt einfügen, weil ich einfach nicht wusste wie und ich keine Zeit mehr hatte um noch weiter nachzuforschen.
-

Alle weiteren Funktionen musste Ich zeitbedingt weggelassen, da Ich mich darauf fixiert habe, alle von mir erstellten Funktionen sauber ablaufen lassen zu können.

6. Testfälle

Schritt	Testfall	Erwartetes Resultat
1	Kunde gibt folgende Daten ein: - Von: Luzern - Nach: Zürich HB - Datum: 24.04.2018 - Zeit: 08:51 Kunde klickt auf «Verbindung suchen»	In Verbindungen werden 4 Ergebnisse angezeigt.
2	Kunde klickt auf «Per Email versenden»	MessageBox erscheint mit dem Inhalt «Bitte geben Sie eine Email-Adresse ein!»
3	Kunde klickt auf «OK»	MessageBox verschwindet.
4	Kunde gibt eine Email-Adresse ein: - lynn.jung@eduone.ch Kunde klickt auf «Fahrplan per Email senden»	MessageBox erscheint mit dem Inhalt «Email wurde erfolgreich gesendet»

Schritt	Testfall	Erwartetes Resultat
1	Kunde wählt die Option «Abfahrtstafel»	Das Form «Abfahrtstafel» erscheint
2	Kunde gibt folgende Daten ein: - Von: Luz Und drückt auf Ort suchen	Feld wird mit Luzern aufgefüllt und Dropdown erscheint mit weiteren Vorschlägen.
3	Interessent klickt auf «Suche»	Datagrid erscheint mit allen Verbindungen ab Luzern

Schritt	Testfall	Erwartetes Resultat
1	Kunde wählt die Option «Ort auf der Karte anzeigen» aus.	MessageBox erscheint mit Nachricht «Bitte Station Auswählen»
2	Kunde gibt folgende Daten ein: - Von: Luz Und drückt auf Ort suchen	Feld wird mit Luzern aufgefüllt und Dropdown erscheint mit weiteren Vorschlägen.
3	Kunde wählt die Option «Ort auf der Karte anzeigen» aus.	Form Maps erscheint mit Google Maps und dem gesuchten Ort.

Schritt	Testfall	Erwartetes Resultat
1	Kunde wählt die Option «Stationen in der Nähe» aus.	Form Standortmaps erscheint und zeigt aktuellen Standort mit allen Stationen in der Nähe an.

6.1. Abänderung Testmethode

```
// Ich musste den Test abändern weil der GetConnections 4 Strings erwartet.  
// Date und Time hinzugefügt.  
[TestMethod]  
0 Verweise | LynnJung, Vor 22 Stunden | 1 Autor, 1 Änderung  
public void Connections()  
{  
    testee = new Transport();  
    var connections = testee.GetConnections("Sursee", "Luzern", "2018-12-21", "19:00");  
  
    Assert.IsNotNull(connections);  
}
```

7. Installations- und Deinstallationsanleitung

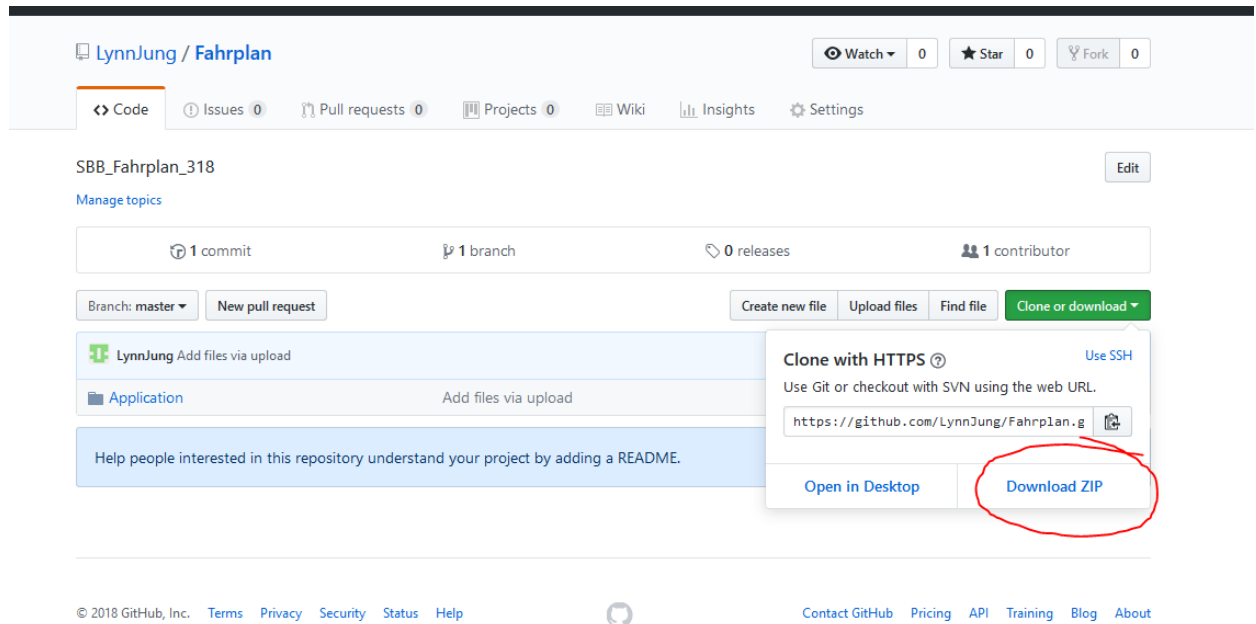
Ich erkläre Ihnen nun wie Sie diese Applikation installieren und deinstallieren können.

7.1. Installation

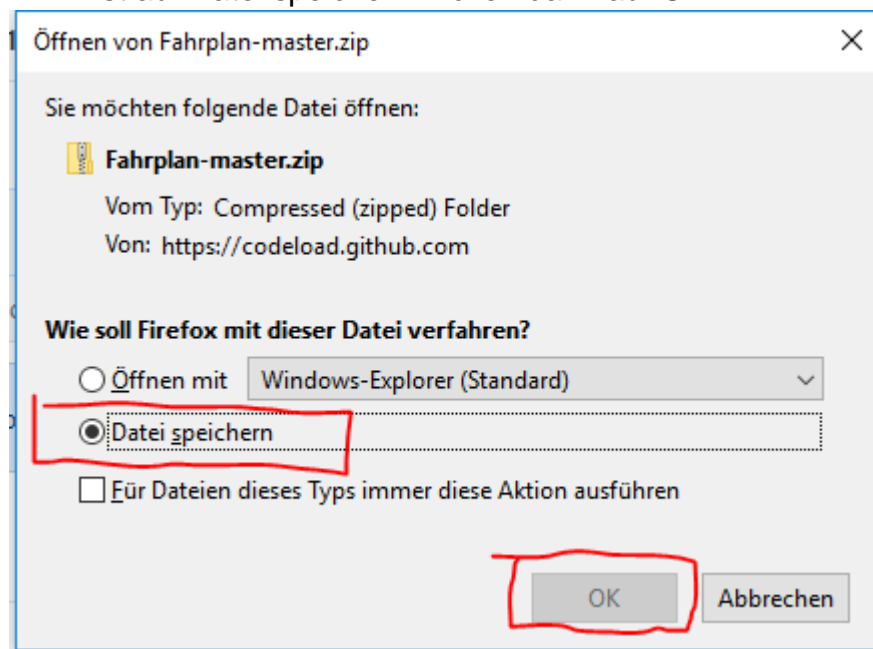
1. Auf den Link klicken. <https://github.com/LynnJung/Fahrplan>
2. Auf Clone or download klicken.

The screenshot shows the GitHub repository page for 'LynnJung / Fahrplan'. At the top, there are buttons for 'Watch', 'Star', and 'Fork'. Below these are tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The repository name 'SBB_Fahrplan_318' is displayed. Below this, there are statistics: '1 commit', '1 branch', '0 releases', and '1 contributor'. A 'Branch: master' dropdown and a 'New pull request' button are visible. The 'Clone or download' button is highlighted with a red circle. Below the repository information, there are sections for 'Add files via upload' and 'Add files via upload' with a '26 minutes ago' timestamp. At the bottom, there is a section for 'Help people interested in this repository understand your project by adding a README.' with an 'Add a README' button.

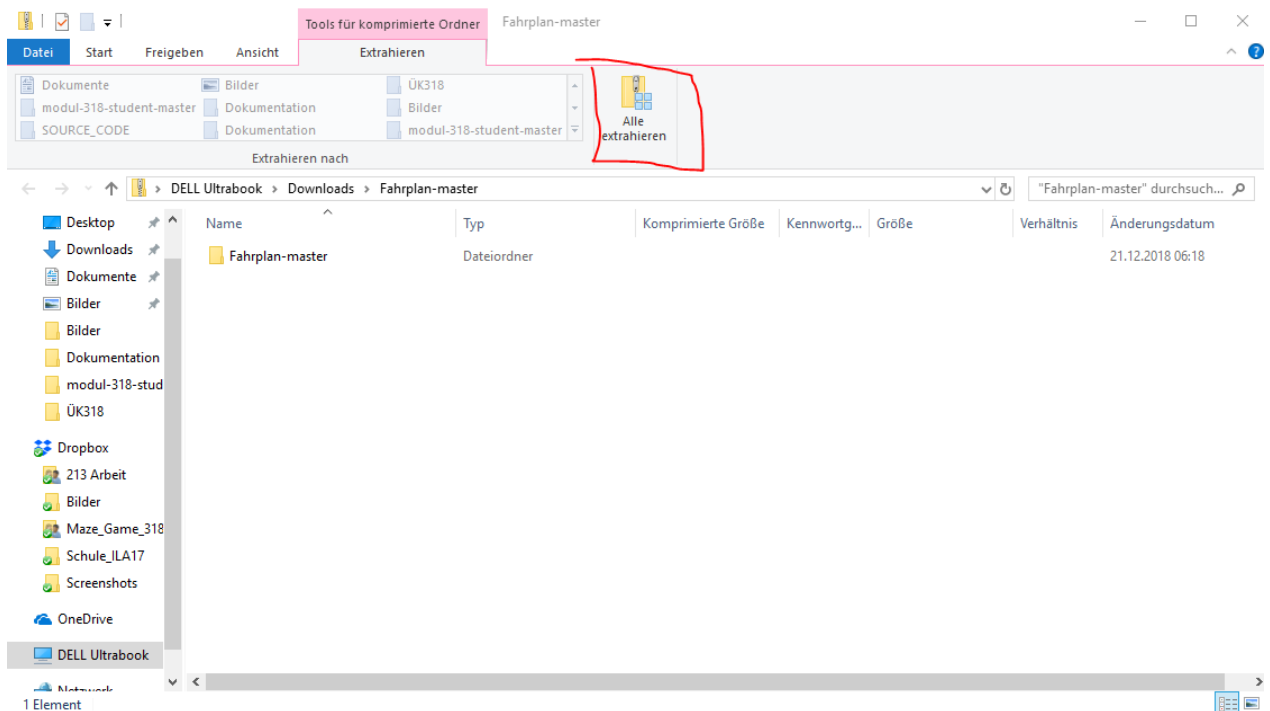
3. Auf Download ZIP klicken und ZIP runterladen.



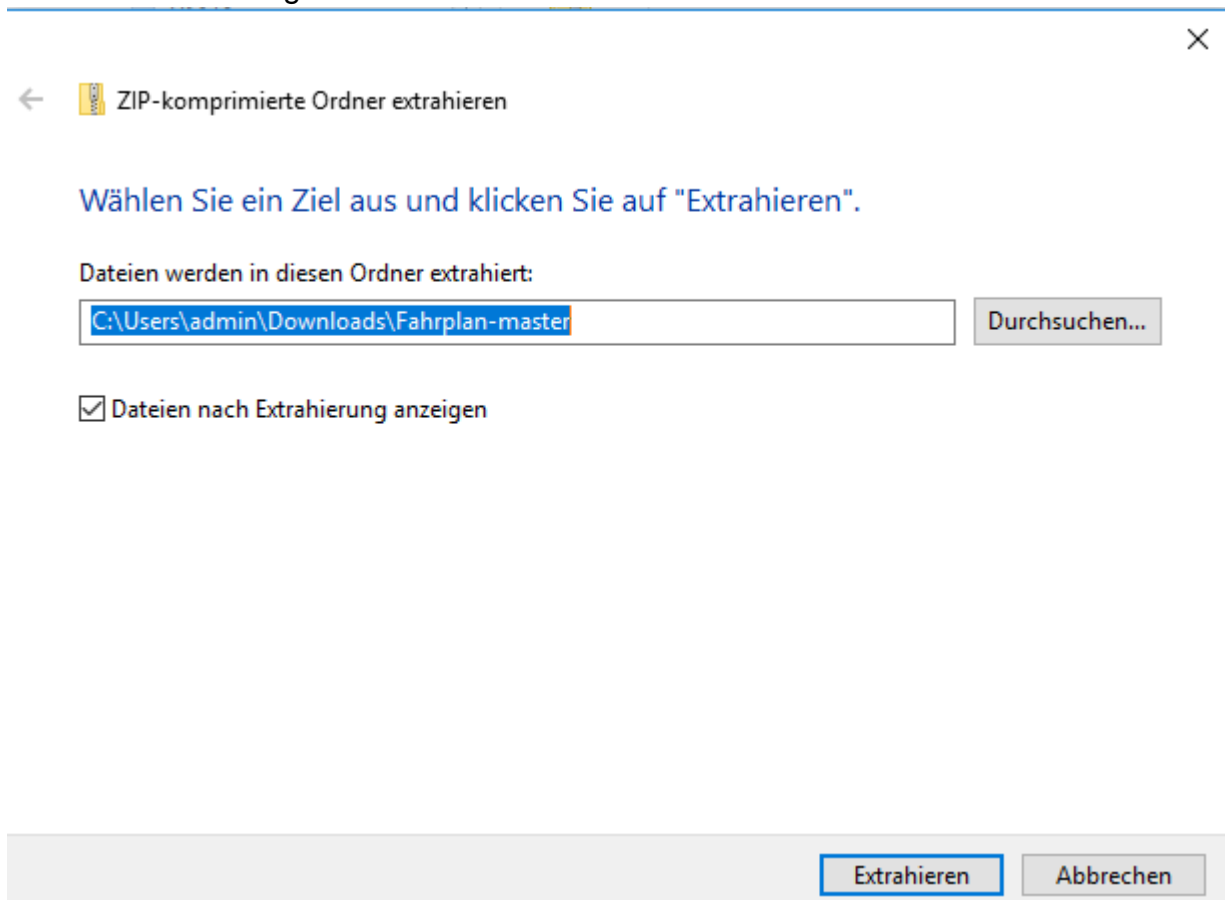
4. Erst auf Datei speichern klicken dann auf OK.



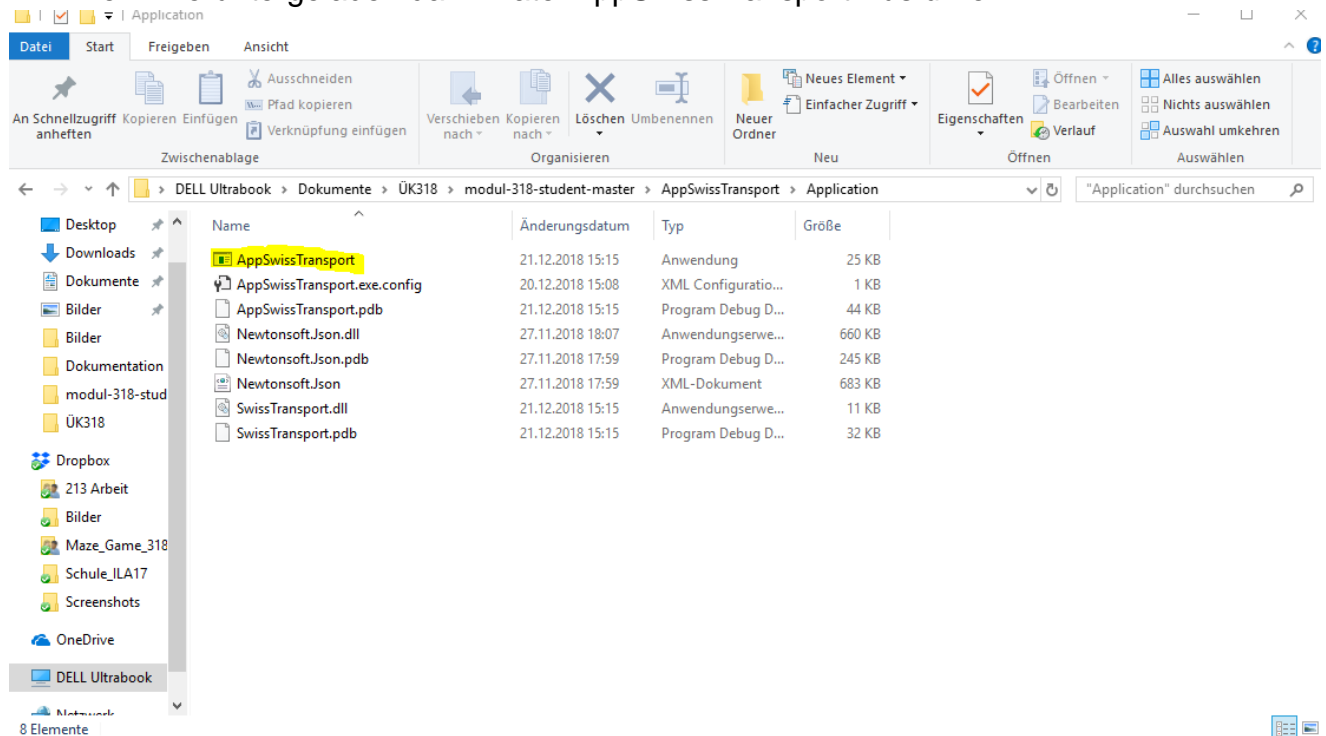
5. Den ZIP Ortner extrahieren.



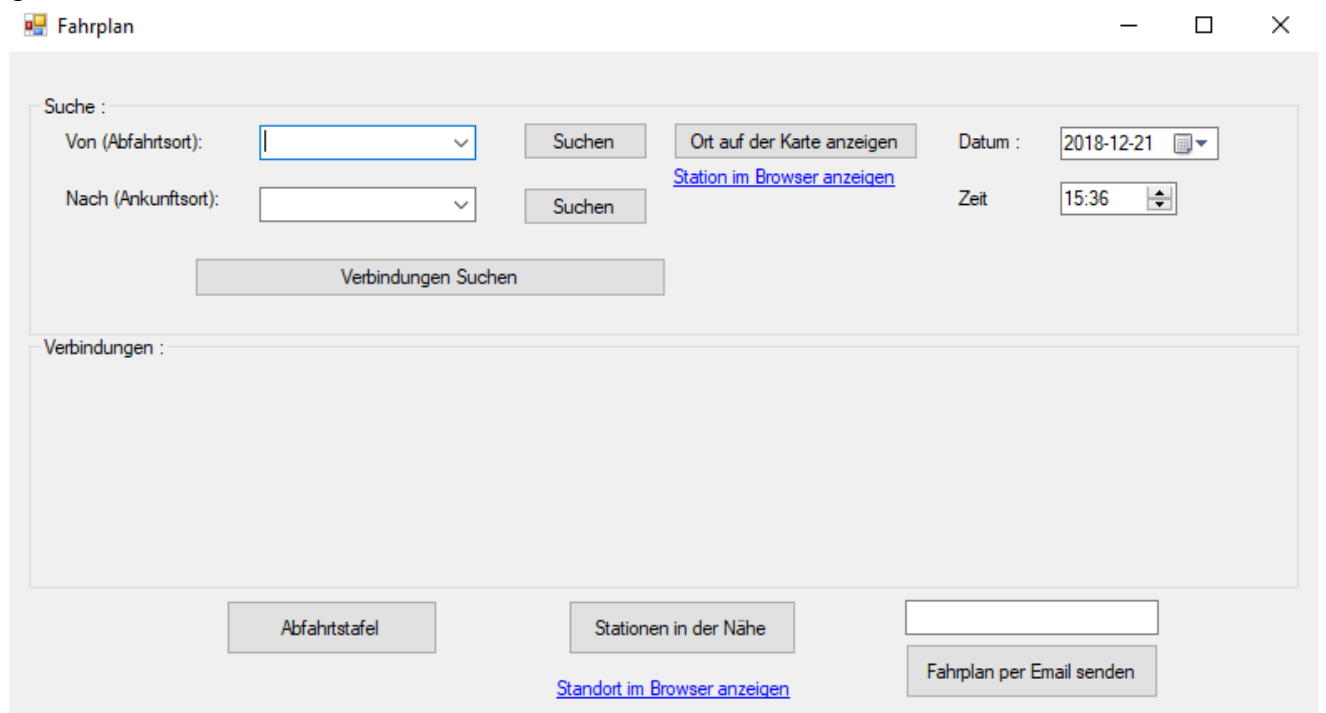
6. Dateien in den gewünschten Ortner laden.



7. Wenn Heruntergeladen dann Datei AppSwissTransport Ausführen.



8. Wenn alles richtig gelaufen ist dann solltest du jetzt im Stande sein meine App zu genießen.

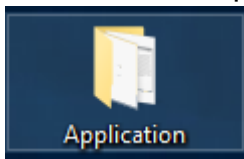


Viel Spaß ;D

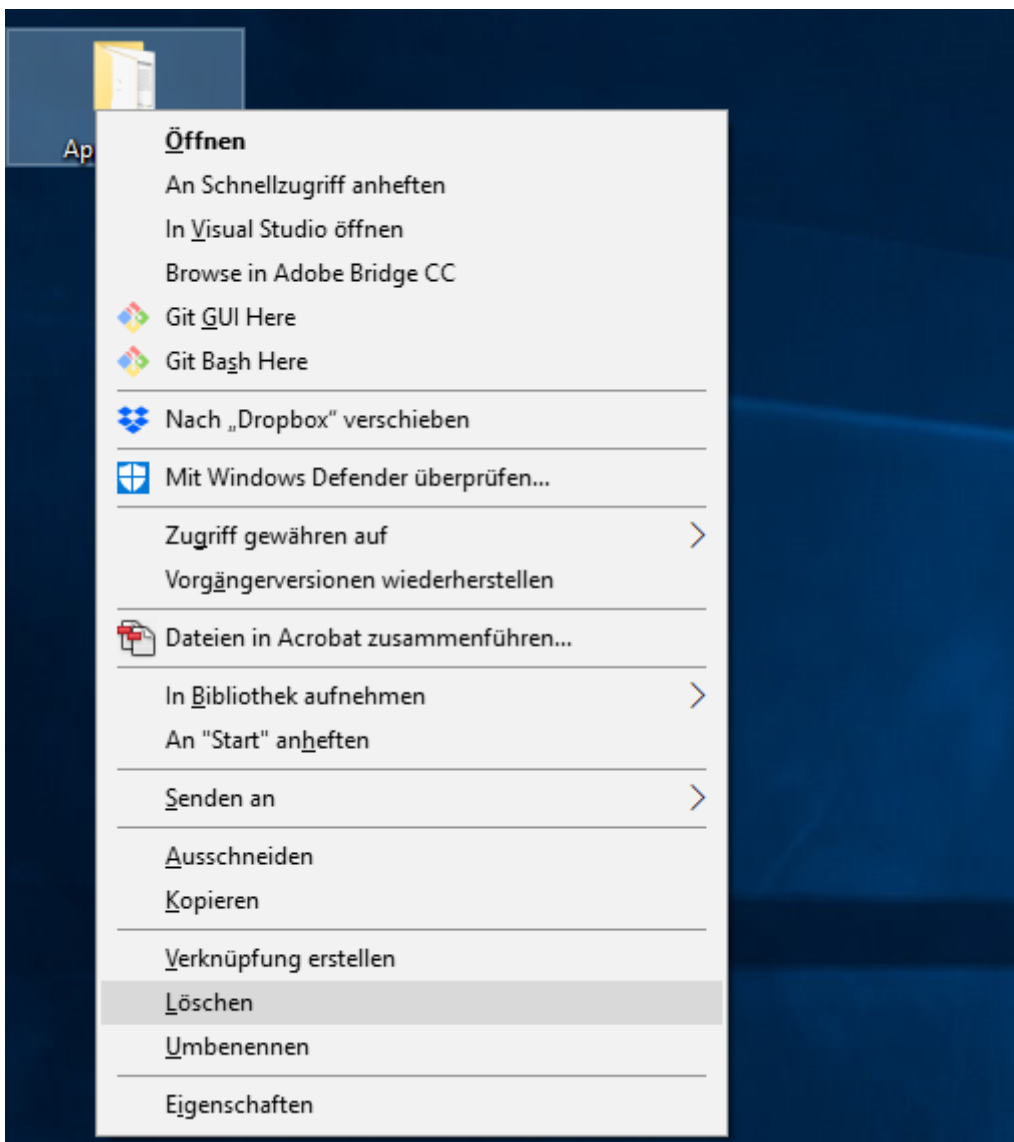
7.2. Deinstallation

Solltest du keinen Gebrauch mehr für die App haben oder keine Freude daran dann erkläre ich dir nun wie du diese wieder Deinstallieren kannst.

1. Ortner mit der App Auswählen



2. Rechte Maustaste klicken und Löschen auswählen.



Und schon ist die Applikation wieder gelöscht.

8. Fazit

Mir hat der ÜK mit dem Thema Objektorientiertes Programmieren teilweise sehr gefallen, da Ich die Erfolge sofort an meiner Applikation erkennen konnte. Für mich ist es immer noch erstaunlich, wie nur durch das Programmieren eine Windows Forms den Code visualisiert und man die Funktion, falls die Codierung «clean» war, auch sofort erkennen kann.

Ich hoffe mein gefestigtes Basiswissen weiterhin in Zukunft oft zu gebrauchen und mehr über Clean Code in die Praxis umsetzen zu können, um noch genauer und damit auch noch schneller zu programmieren.

Alles in Allem: Eine saubere 7/10 😊