

ÜK318

Fahrplan Applikation



Autor: Martin Eberhart

Programmiert mit C#

ÜK Modul 318

Inhalt

1. Einleitung.....	2
2. Zweck des Dokumentes.....	2
3. Richtlinien.....	2
4. Umgesetzte Funktionen	3
5. Fehlende Funktionen.....	4
6. Planungsphase.....	4
6.1 Mockup.....	5
6.1.1 Umsetzung des Mockups	5
6.2 Use Case Diagramm.....	7
6.3 Aktivitäten Diagramm	7
7. Systemtests	8
8. Installationsanleitung	9
9. Deinstallation	10

1. Einleitung

Wir bekamen den Auftrag eine SBB Fahrplan Desktopapplikation zu entwickeln und sauber zu dokumentieren. In diesem Dokument zeige ich mein Vorgehen wie ich die Applikation entwickelt habe und bereitstelle. Ebenso findet man eine Installationsanleitung wie man die Applikation erfolgreich installieren kann.

2. Zweck des Dokumentes

Durch dieses Dokument erhält man eine Übersicht über meine Applikation. Es zeigt wie die Applikation aufgebaut ist.

3. Richtlinien

Namensgebung von Variablen/Klassen/Methoden

Klassen:

-immer gross schreiben

GUI- Controls

Namensgebung von Labels, Buttons, Textboxen:

-Sollen mit Abkürzungen benannt werden

Z.B txt bei Textboxen, btn bei Buttons

-Name Soll Eindeutig, verständlich und grossgeschrieben sein:

txtEingabe, btnAnzeigen/btnSuchen

Abstand (white spaces)

Beispiel:

-Allgemein nach einem Komma und einem Semikolon ein Leerzeichen setzen

Aber nicht nach dem Öffnen und vor dem Schliessen einer Klammer.

Gutes Beispiel:

```
TestMethod(a, b, c);
```

Schlechtes Beispiel:

```
TestMethod(a,b,c);
```

```
TestMethod( a, b, c );
```

Zwischen Operatoren soll auch ein Leerzeichen sein

Gutes Beispiel:

```
a = b;  
for (int i = 0; i < 10; i++)
```

Schlechtes Beispiel:

```
a=b;  
for(int i=0; i<10; i++)
```

Kommentieren

Nicht in jeder Zeile soll ein Kommentar geschrieben werden.

Kommentare sinnvoll verwenden (kurz, Beschreibung von einer Variabel)

Komentare immer vor dem kommentierten Code

Gutes Beispiel:

```
//Variabel i wenn kleiner als Zehn wird nochmals eins dazugezählt//  
a = b;  
for (int i = 0; i < 10; i++)
```

Schlechtes Beispiel:

```
a = b;  
for (int i = 0; i < 10; i++) //Variabel i wenn kleiner als Zehn wird nochmals eins  
dazugezählt//
```

4. Umgesetzte Funktionen

Es ist möglich mittels Text suche Start und Endstationen einzugeben und dazu die Fahrpläne anzuzeigen (A001). Nach der suche werden die nächsten 4-5 mögliche Verbindungen angezeigt. Man kann auch nach schauen welche Verbindungen ab einer bestimmten Station vorhanden sind in verschiedene Richtungen. Bei der Eingabe der Station werden schon verschiedene Vorschläge angezeigt. Das Datum und die Zeit der Abfahrt können beliebig gewählt werden. Es ist möglich auf Google Maps den Standort der Station anzuzeigen. Die Liste der gefundenen Verbindungen kann mit einer Email verschickt werden.

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

5. Fehlende Funktionen

Es ist leider nicht möglich einen Ort einzugeben und dass dann die Stationen in der Nähe anzuzeigen.

6. Planungsphase

In der Planungsphase habe ich Mockups erstellt, wie die Applikation aussehen könnte. Da habe ich die Aufgabe A008 mit dem Email versenden noch gar nicht beachtet. Dies habe ich später einfach im GUI hinzugefügt. Später ist mir auch aufgefallen das ich beim Mockup noch keine Buttons für den Abfahrtsplan und für Google Maps erstellt hatte. Das Layout habe ich am Schluss auch ein bisschen abgeändert. Ich wollte am Anfang alles in einem Fenster machen, habe aber später die Google Maps Funktion in einem neuen Fenster gemacht. Damit die Applikation übersichtlicher wird.

6.1 Mockup


Abfahrtsort:

Ziel:

Uhrzeit: von: Datum:

Nächste fahrten vom aktuellen standort:

6.1.1 Umsetzung des Mockups

 Fahrplan — □ ×

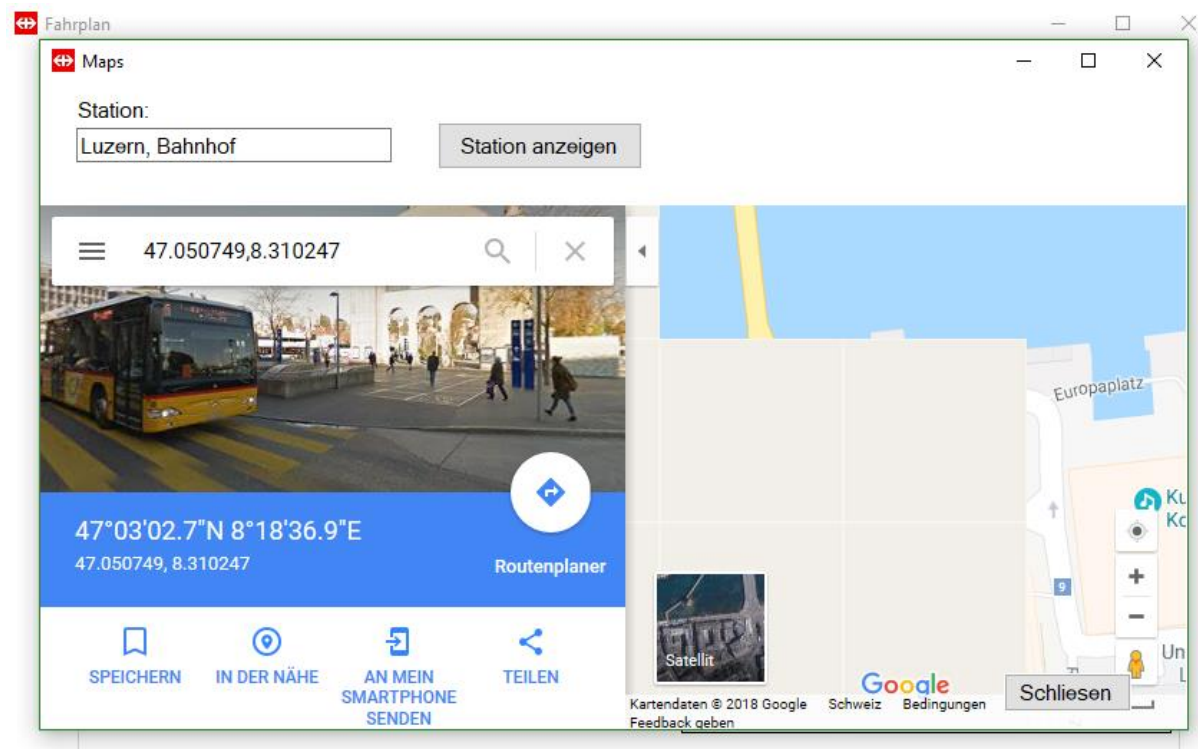
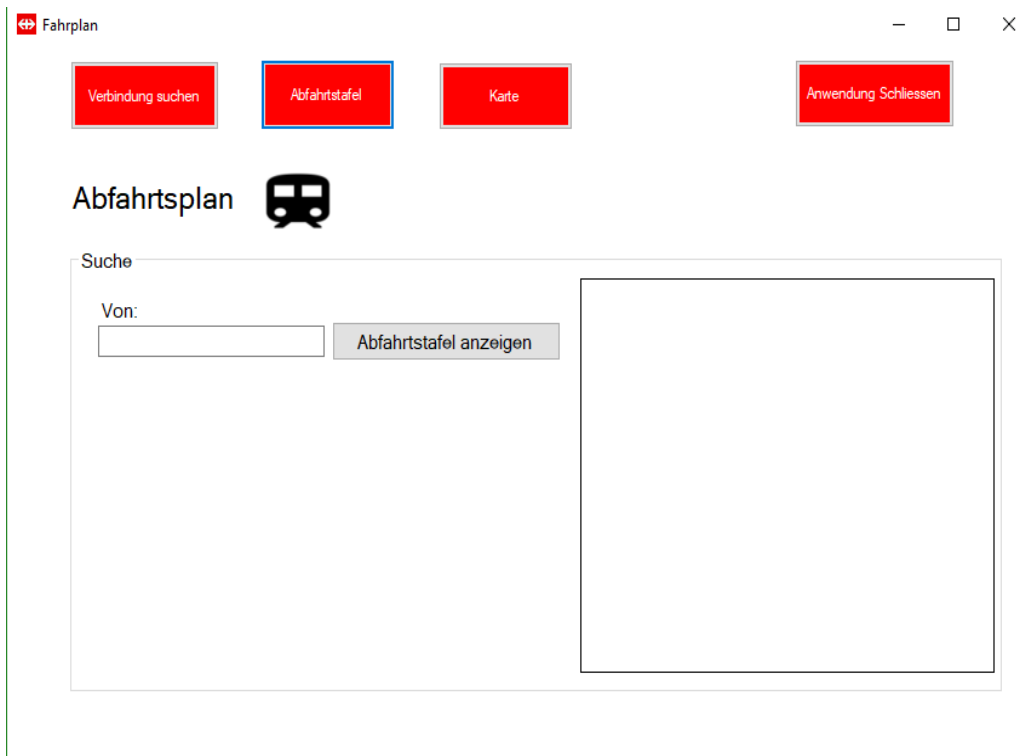
Suche

Abfahrtsort: Ziel:

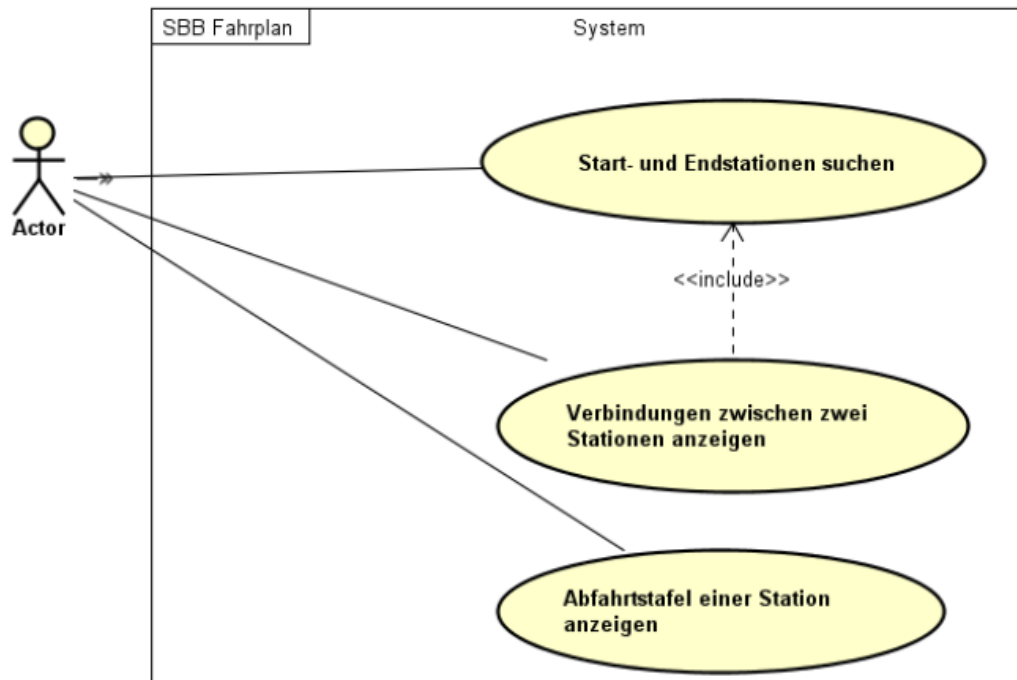
Uhrzeit: Datum:

Resultat

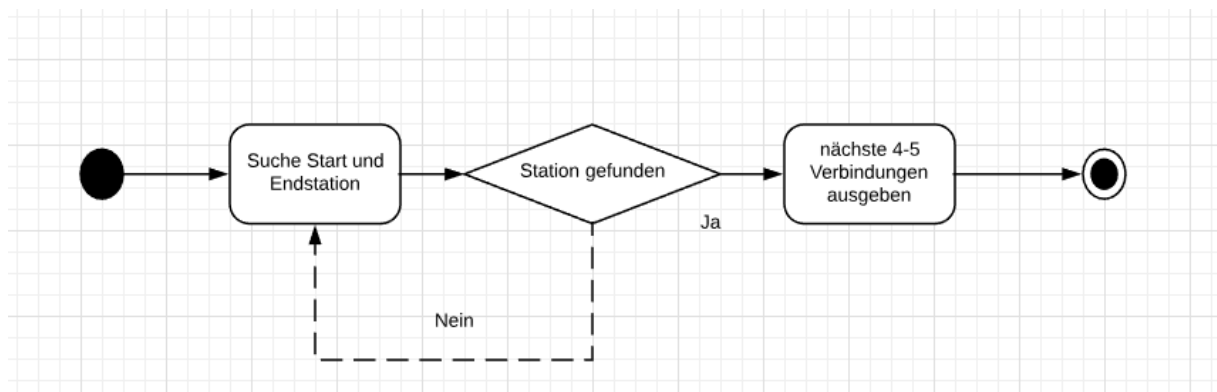
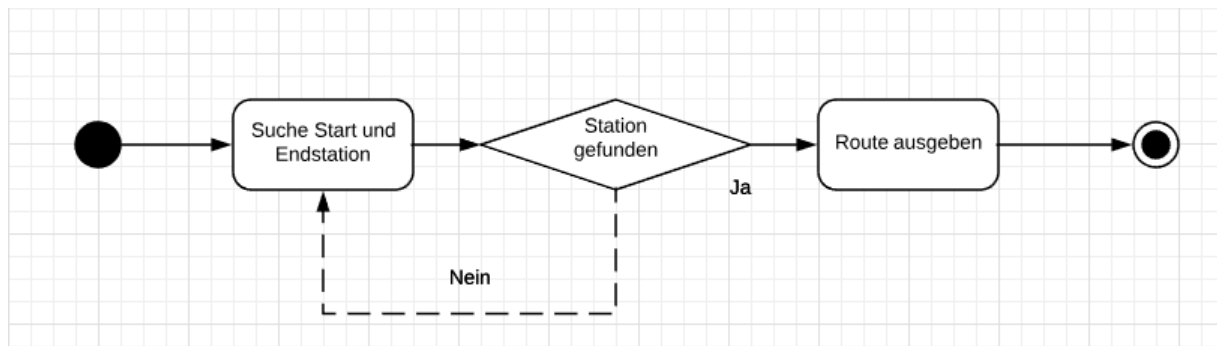
Liste als Mail Schicken:

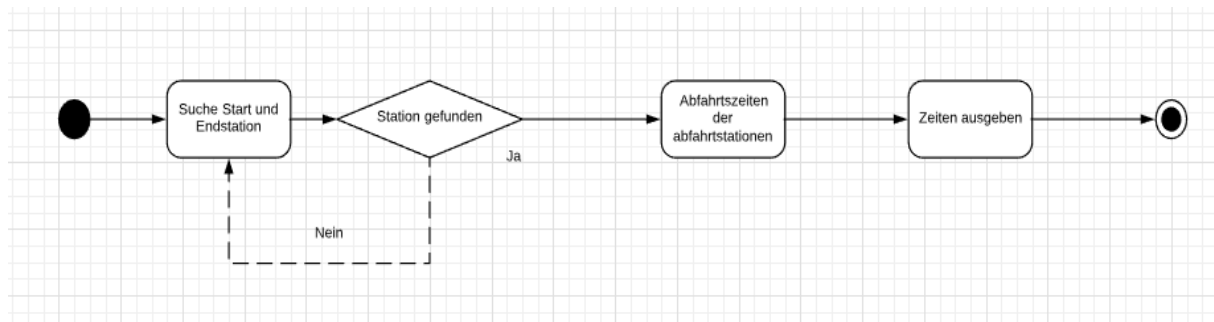


6.2 Use Case Diagramm



6.3 Aktivitäten Diagramm





7. Systemtests

Schritt	Aktivität	Erwartetes Resultat
1	User wählt die Option Verbindung suchen.	Das Panel Verbindung Suchen erscheint.
2	User gibt folgende Daten ein: -Abfahrtort: Brunnen, Bahnhof -Ziel: Luzern, Bahnhof -Uhrzeit: 09.24 -Datum: 21.12.2018 User klickt auf: Verbindung Suchen	Eine Liste der Verbindungen wird angezeigt.
3	User klickt auf Schicken	MessageBox erscheint «Bitte geben Sie eine E-Mail-Adresse ein!»
4	User klickt auf OK	MessageBox verschwindet.
5	User gibt eine E-Mail-Adresse ein: - Mart.t@gmx.ch User klickt auf «Schicken»	MessageBox erscheint mit dem Inhalt «Email wurde erfolgreich gesendet»

Schritt	Aktivität	Erwartetes Resultat
1	User gibt bei Abfahrtsort folgendes ein - Bru	Vorschläge zum aussuchen erscheinen
2	User Doppelklickt auf Brunnen	Die TextBox wird mit «Brunnen» aufgefüllt
3	User gibt folgende Daten ein: - Ziel: Luzern - Uhrzeit 09.35 - Datum 21.12.2018 User klickt auf «Verbindung suchen»	Ein Liste der Verbindungen wird angezeigt.

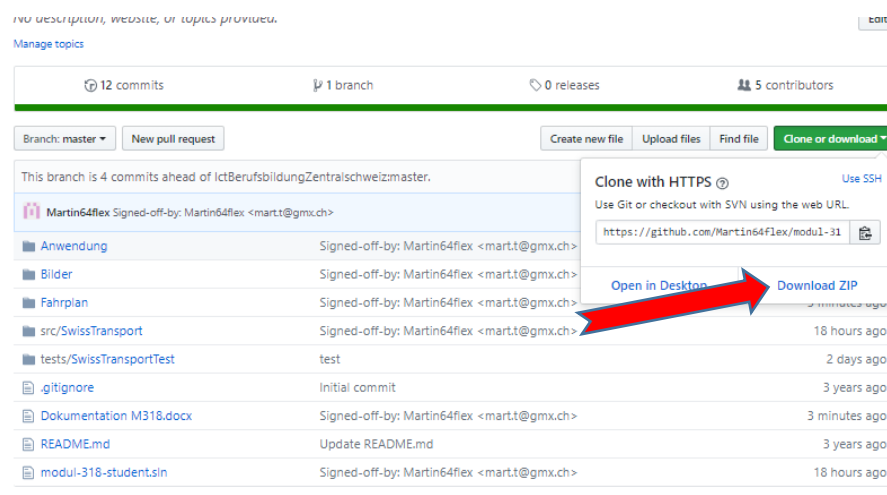
Schritt	Aktivität	Erwartetes Resultat
1	User klickt auf «Abfahrtstafel»	Panel Abfahrtstafel erscheint
2	User gibt folgende Daten ein: - Brunnen, Bahnhof User klickt auf «Abfahrtstafel anzeigen»	Es werden die Nächsten Verbindungen ab dieser Station angezeigt

Schritt	Aktivität	Erwartetes Resultat
1	Interessent wählt die Option «Karte»	Ein neues Fenster erscheint
2	User gibt Folgende Daten ein: - Station: Brunnen, Bahnhof User klickt auf «Station anzeigen»	Es wird im Integrierten Webbrowser ein Google Maps ausschnitt angezeigt

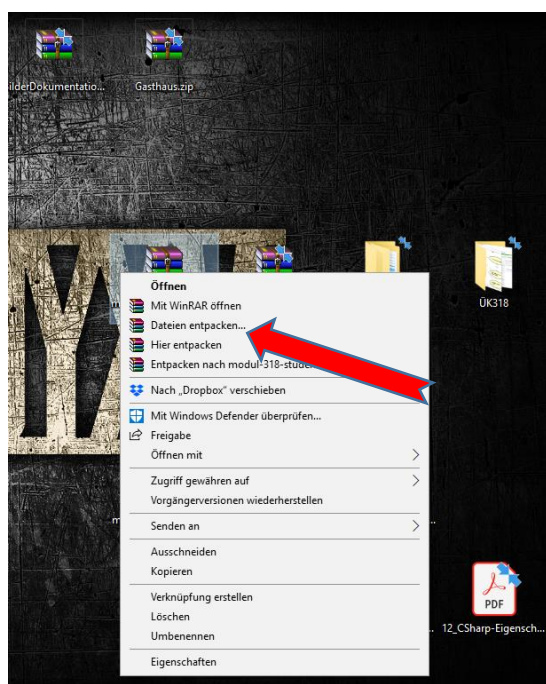
8. Installationsanleitung

1. Öffnen sie folgenden Link: <https://github.com/Martin64flex/modul-318-student>

Klicken sie auf Clone or Download und dann Download ZIP

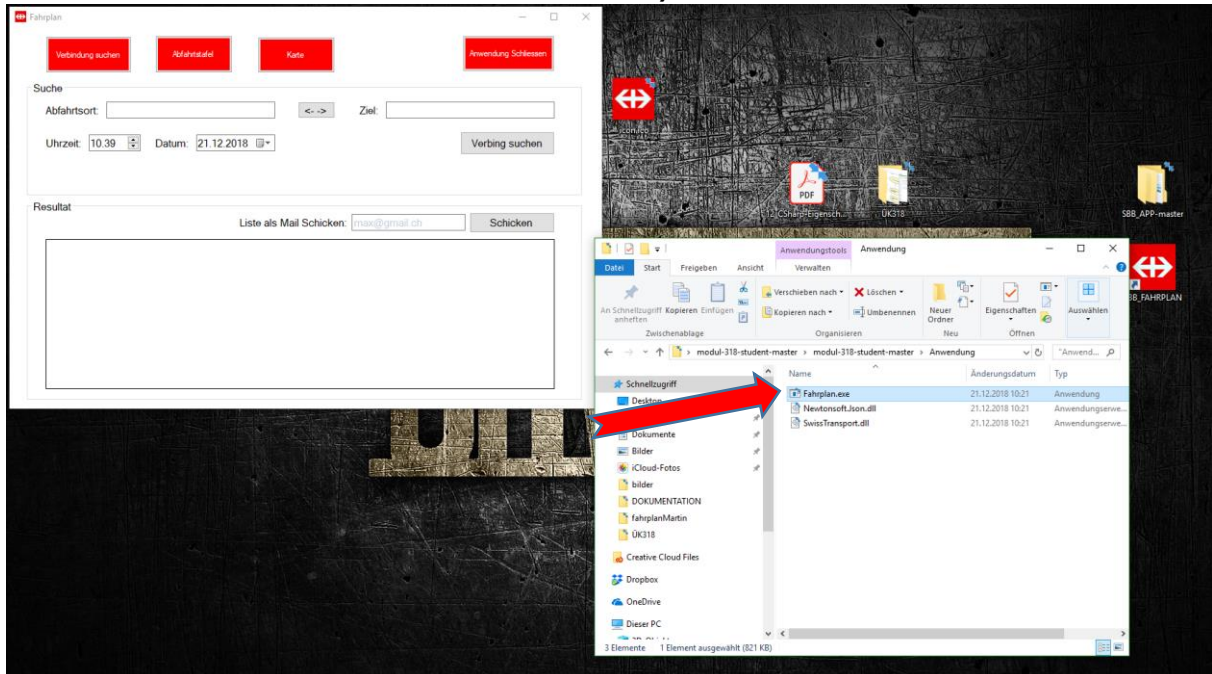


2. Rechtsklick auf die ZIP Datei und klicken sie auf Datei entpacken



3. Öffnen sie dann den Ordner «modul-318-student-master» dann nochmal auf «modul-318-student-master» dann auf den Ordner Anwendung.

4. Im Ordner Anwendung haben sie eine exe Datei und zwei weitere Dateien, mit der exe Datei kann das Programm gestartet werden. **(Wichtig! Es müssen sich alle drei Dateien immer am selben Ort befinden)**



9.Deinstalation

Da es eine EXE Datei ist kann die Datei einfach gelöscht werden und muss nicht extra deinstalliert werden. (Rechtsklick auf Fahrplan.exe und auf Löschen klicken)