Dokumentation Projekt des ÜK318-1



Lino Bucher

05.12.2018

ÜK 318-1

Abschlussprojekt Dokument

Inhaltsverzeichnis

[1. Einleitung 3](#_Toc531789463)

[2. Zweck dieses Dokuments 3](#_Toc531789464)

[3. Welche Funktionen wurden umgesetzt 3](#_Toc531789465)

[4. Was ist noch fehlerhaft 4](#_Toc531789466)

[5. Diagramme der Priorität 1 Aufgaben 4](#_Toc531789467)

[Klassendiagramm 4](#_Toc531789468)

[Use-Case Diagramm 5](#_Toc531789469)

[Aktivitätsdiagramm 5](#_Toc531789470)

[6. GUI 6](#_Toc531789471)

[7. Programmierrichtlinien 7](#_Toc531789472)

[8. Testfälle 11](#_Toc531789473)

[Station-Suchen per Button 11](#_Toc531789474)

[Station Table anzeigen 11](#_Toc531789475)

[Verbindungen anzeigen 11](#_Toc531789476)

[Autocomplete Checkbox 11](#_Toc531789477)

[Routenplaner anzeigen 11](#_Toc531789478)

[9. Installationsanleitung 12](#_Toc531789479)

[10. Reflexion 13](#_Toc531789480)

[Positives 13](#_Toc531789481)

[Negatives 13](#_Toc531789482)

# Einleitung

Im ÜK 318-1 behandelt man das analysieren und das objektorientierte Programmieren. In diesem ÜK haben wir diverse Übungen zum objektorientiertem Programmieren gelöst, welche uns nachher im Abschlussprojekt helfen können. Ausserdem haben wir viel Theorie behandelt, welche auch abgelegt wurde und uns helfen kann.

Das oben genannte Abschlussprojekt besteht darin eine Applikation zu entwickeln, welche einen Fahrplan visualisiert und durch Such-/Filteroptionen vereinfacht. Dazu verwendet man die API-Schnittstelle, welche von SBB zur Verfügung gestellt wird.

# Zweck dieses Dokuments

Das Dokument dient zum besseren Verständnis des Vorgehens und des Codes. Ausserdem soll es meine Gedankensprünge sowie mein Vorgehen aufzeigen.

# Welche Funktionen wurden umgesetzt

In meiner Umsetzung des Projekts wurden folgende Funktionen umgesetzt.

* Auf Knopfdruck werden alle Stationen angegeben, welche auf die Eingabe in eine Combobox des Benutzers zutreffen.
* Nachdem man ausser der Startstation einen Ankunftsort auf dieselbe Weise wie der Startpunkt definiert hat, kann man auf einen Knopf drücken und es werden einem die nächsten vier bis fünf zutreffenden Verbindungen zwischen den beiden Orten angezeigt.

Dies geschieht durch eine DataGridView.

* Wenn man nur eine Startstation eingegeben hat, hat man die Möglichkeit ein Fahrplan aller Verbindungen ausgehend von diesem Ort aus anzeigen zu lassen.

Dies wird auch in einer neuen DataGridView angezeigt, welche erst sichtbar wird, wenn auf den dafür vorgesehenen Knopf gedrückt wird.

* Für das noch einfachere Handling habe ich noch die Autocompletion in den beiden Comboboxen des Abfahrtsortes und des Ankunftsortes eingebaut. Damit fällt eigentlich der erste genannte Punkt weg, jedoch liess ich ihn trotzdem in dem Programm, da doch die eine oder andere Zeile Code dahinterstecken und ich es recht sauber gelöst finde.
* Wenn man sich die Verbindungen anzeigen lässt, kann man auf eine Spalte der Gridview klicken und man gelangt auf Google Maps mit der angezeigten Route. Dies wird durch ein Label gezeigt.
* Meine Zusatzfunktion ist das ein und ausschalten der Autocompletion. Ich habe dies mithilfe von einer Checkbox gelöst. Ist sie angekreuzt, ist die Autocompletion an, ist sie nicht angekreuzt, so ist sie aus.

Weitere Anforderungsziele habe ich noch nicht weiter programmiert, da ich mir selber die Prioritäten nicht nur auf die Quantität setzte, sondern mehr auf die Qualitä

# Was ist noch fehlerhaft

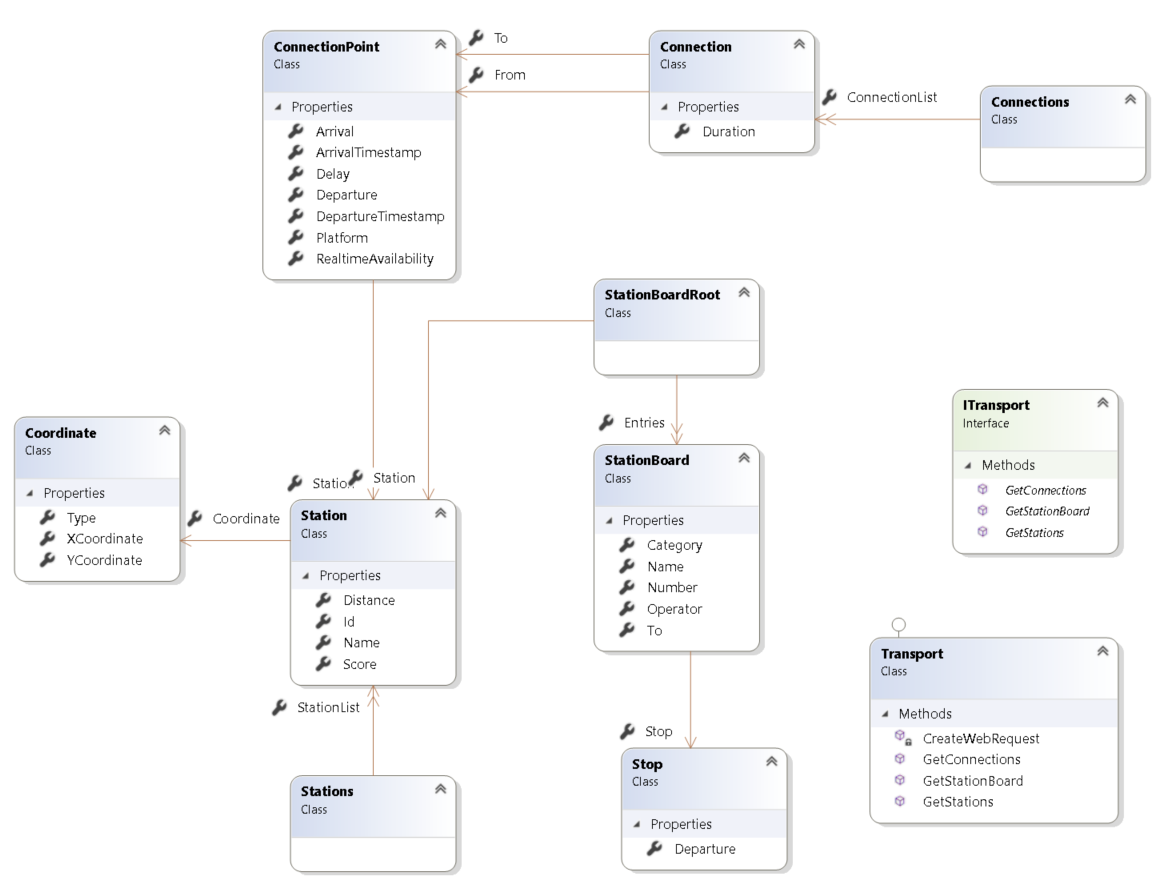
Einer meiner grössten Probleme ist ein Bug, welcher ich nicht aus meinem Programm bringe.

Meine Autocompletion läuft, jedoch kommt ein Fehler auf, wenn man etwas eingibt, dies dann wieder löscht und anschliessend die Combobox wechselt. Dann stürzt sogar das ganze Programm ab. Diesen Fehler konnte ich mit verschiedenen Methoden und Events nicht beheben. Da man den Fehler aktiv auslösen muss, ist er bei mir nicht auf erster Stelle.

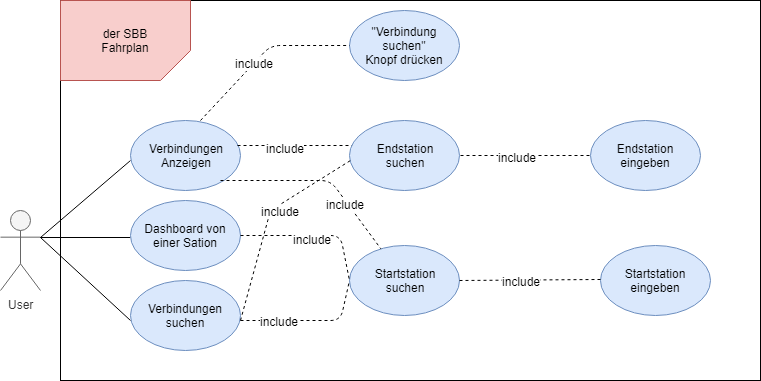
# Diagramme der Priorität 1 Aufgaben

In diesem Absatz meiner Dokumentation finden Sie meine erstellten Diagramme zu den drei Priorität 1 Aufgaben.

## Klassendiagramm



## Use-Case Diagramm



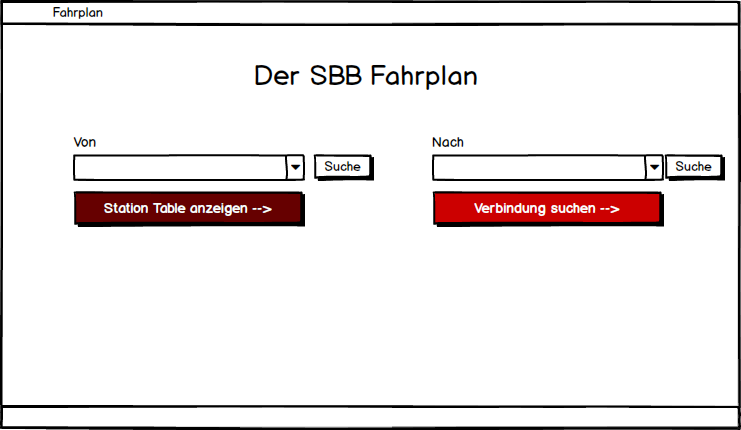
## Aktivitätsdiagramm



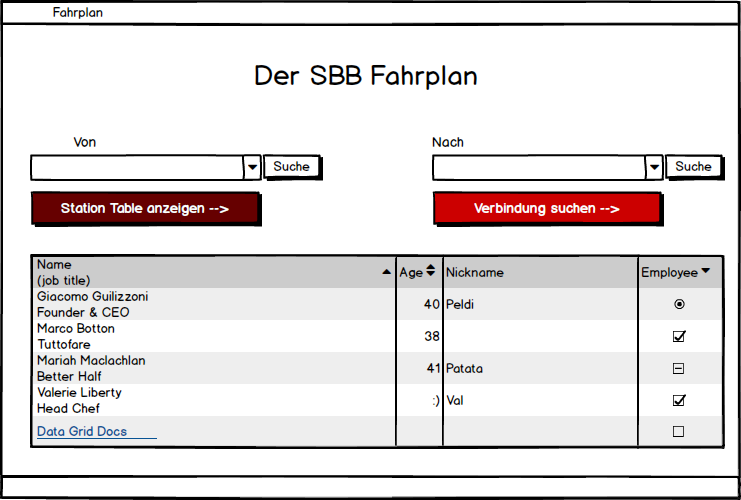
# GUI

In diesem Absatz können Sie mehr über meine Gedanken hinter meinem GUI-Design erfahren. Dazu habe ich in Balsamiq Mockups 3 mein GUI designt und möchte gerne die einzelnen Ausschnitte erläutern.

Zu meinem User Interface ist noch zu sagen, dass es sehr einfach gehalten wurde und es deshalb auch nur zwei Ausschnitte davon gibt.



Wenn man das Tool öffnet, erscheint das obige Fenster. Wenn man in der ersten Combobox beginnt zu schreiben, kommt sofort die Autocompletion und hilft einem bei der Suche nach dem gewünschten Ort. Man kann auch sein Ziel eingeben und dann auf den Button drücken, welcher jeweils rechts von der Combobox liegt. Wenn man nun nur eine Startstation eigegeben hat, erscheint einem der dunkelrote Butten als Verfügbar und man kann Ihn drücken, vorher nicht. Den anderen grossen Button kann man erst drücken, wenn in beiden Comboboxen etwas drinsteht.



Auf diesem Ausschnitt sieht man was passiert, wenn einer der Buttons ausgelöst wird. Es erscheint die jeweilige Datagridview zu dem gedrückten Button. Auf den Button «Station Table anzeigen 🡪» erscheint eine Datagridview mit den dazugehörigen Inhalten, genau wie bei dem Button «Verbindung suchen 🡪»

# Programmierrichtlinien

Was ist der Zweck von gemeinsamen Programmierrichtlinien:

**Wartbarkeit, Lesbarkeit, Ästhetik**

Bei der Erstellung von Software sollten immer folgende Faktoren in Erinnerung gerufen werden:

* Kann ich diesen Source-Code auch in einem Jahr noch verstehen?
* Ist meine Software ordentlich strukturiert/modularisiert oder gibt's haufenweise dubiose

Querverbindungen zwischen den Modulen?

* Was passiert, wenn jemand anderer meinen Code betrachtet ("Code–Review") oder Änderungen vornehmen muss – wird er sich zurechtfinden?
* Erscheint meine Software aus "einem Guss" (konsequentes Anwenden von Kommentierungen, Einrückungen, Variablennamen...)?

**Software-Aktualität**

Wird eine Software erstellt oder geändert, so muss diese wie auch alle Dokumente und andere Unterlagen auf dem Server gespeichert werden. Nach einer eventuellen Software-Änderung ausser Hause und muss diese Software sofort bei der Rückkehr (und nicht erst Stunden später oder nie) auf den gemeinsamen Server.

**Firmen-Name, Firmen-Logo**

Wird in einer Software der Firmen-Name irgendwo verwendet (innerhalb eines Hilfe-Fensters o. ä.) so lautet dieser

TEAM ELECTRONICS

und nicht etwa "TEAM Electronics" oder "Team Electronics" oder gar "TEAM Elektronic".

Das Firmenlogo findet sich im Dokument D0393 – bitte keine Eigenkreationen ersinnen.

**Sprache**

Grundsätzlich orientieren sich viele Variablennamen an der englischen Sprache und die Kommentare sind in Deutsch verfasst, sofern nicht Kundenanforderungen dies verhindern.

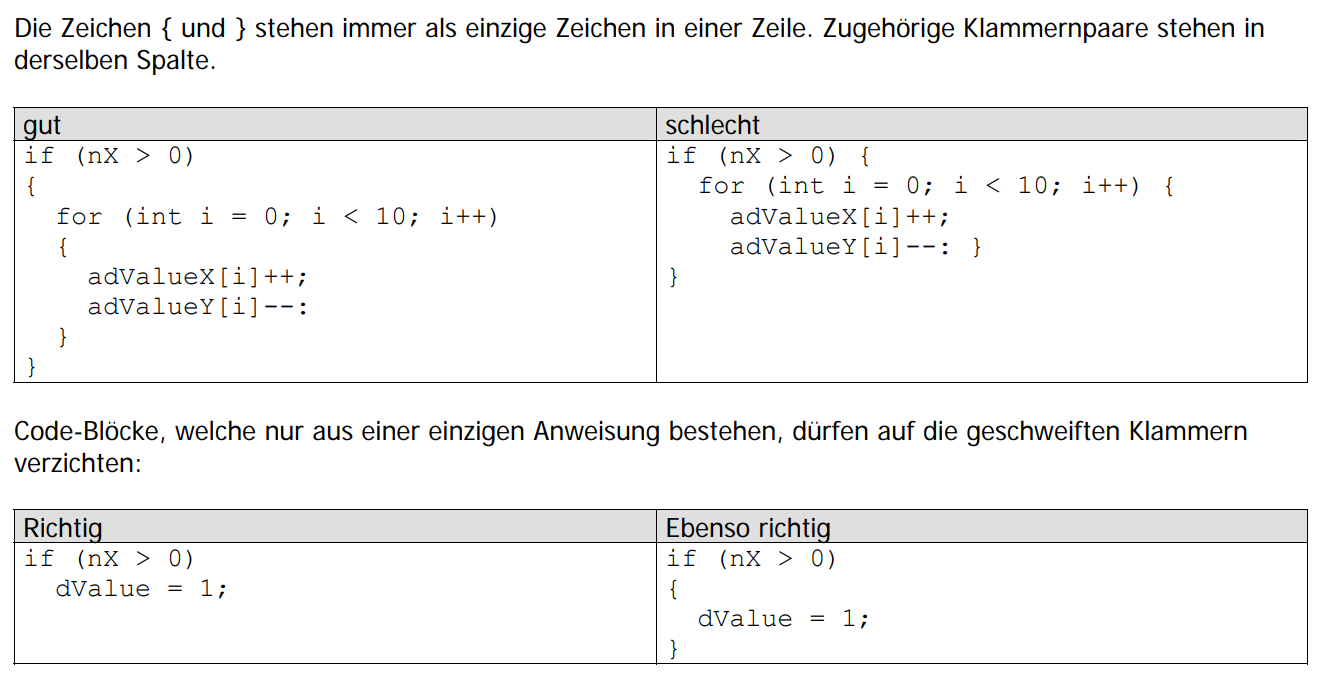
Wie in der Programmierung üblich, ergibt sich oft ein Mix aus deutsch/englisch, was aber im Rahmen der natürlichen Empfindung zulässig ist. Es gelten allgemein die Regeln der neuen deutschen Rechtschreibung.

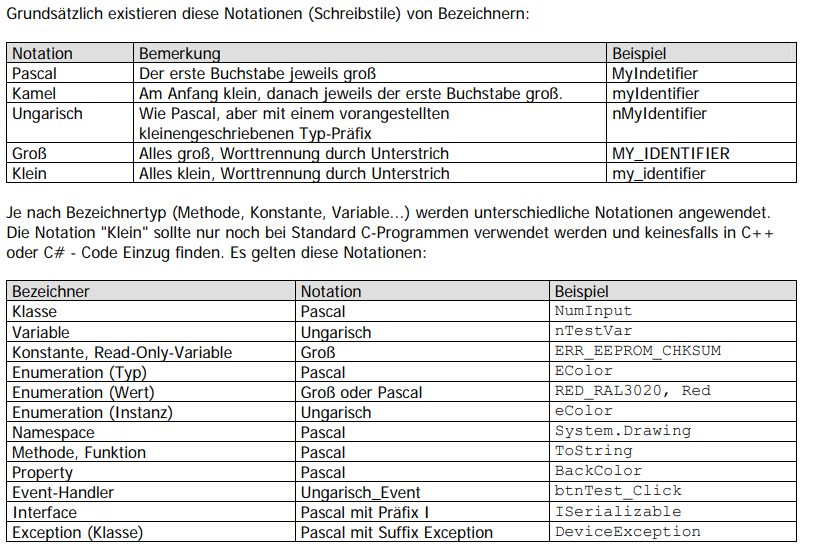
**Dokumentation**

Lästig, aber notwendig: Inbetriebnahmeanleitungen, Beschreibung von Funktionsbibilotheken,

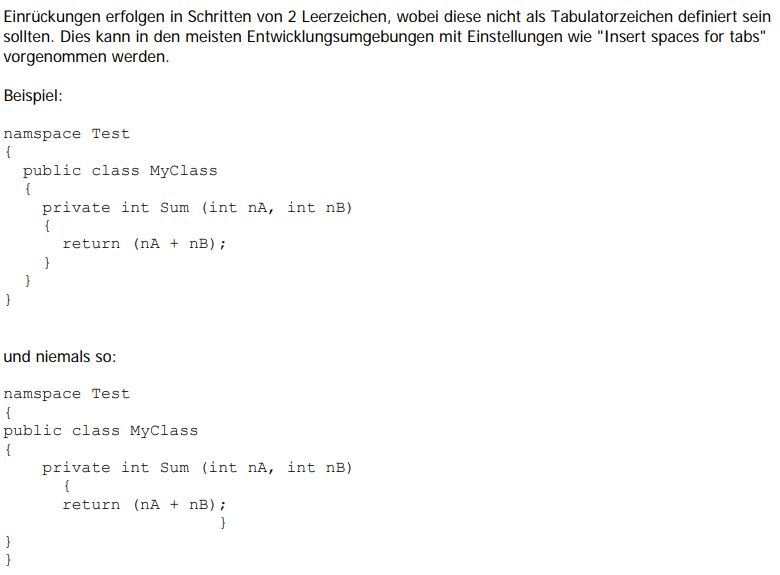
Bedienungsanleitungen, Berechnungs-Dokumentationen (Übersetzungsverhältnisse...) usw.

Programmierrichtlinien:

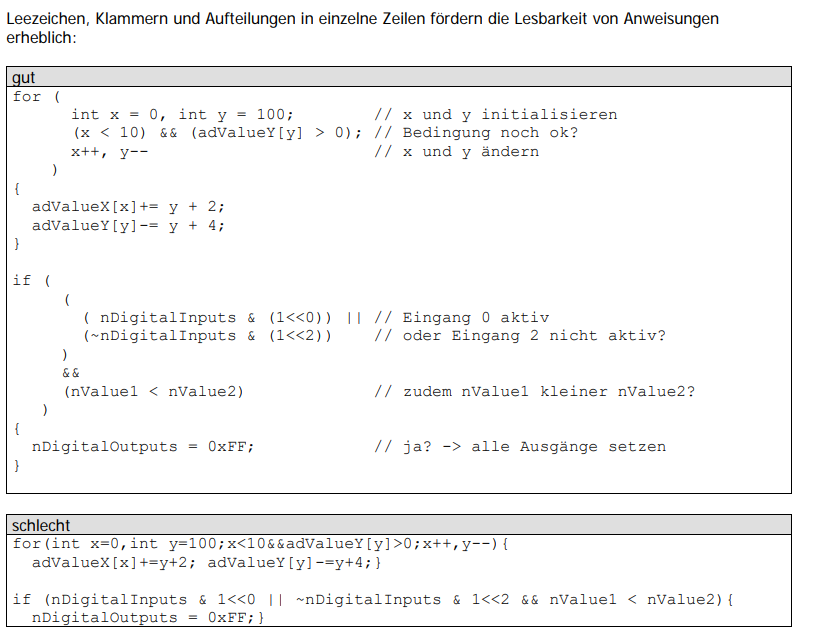
**Verwendung der geschweiften Klammern**

**Notationen**:

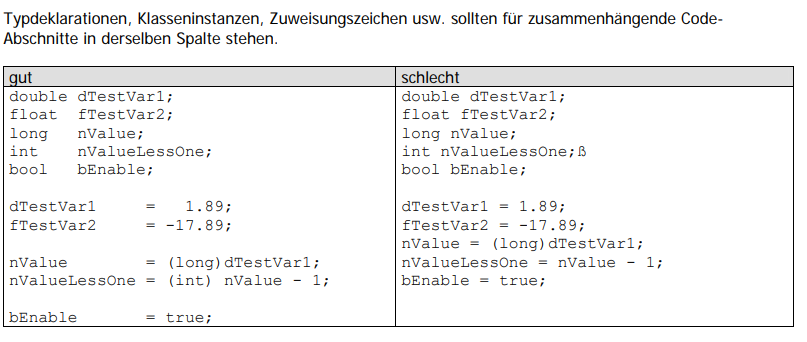
**Einrückungen:**



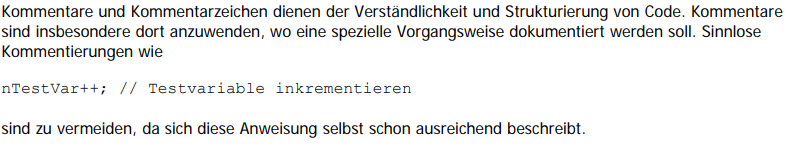
**Verwendung von Leerzeichen und Klammern:**



**Deklarationen und Zuweisungszeichen in derselben Spalte:**



**Kommentare:**



# Testfälle

## Station-Suchen per Button

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | User klickt auf die erste Combobox und drückt auf «Suche» | Die Combobox droppt auf und beinhaltet die relevantesten Stationen zu dem eingegebenen Text des Users |

## Station Table anzeigen

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | User gibt in beiden Comboboxen eine gewünschte Station ein | Die Suche wird durch den Button oder die Autocompletion unterstützt |
| 2 | Der User klickt auf den Button «Station Table anzeigen 🡪» | Es erscheint eine Datagridview mit den Inhalten des Station Tables |

## Verbindungen anzeigen

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | User gibt in beiden Comboboxen eine gewünschte Station ein | Die Suche wird durch den Button oder die Autocompletion unterstützt |
| 2 | Der User klickt auf den Button «Verbindung anzeigen 🡪» | Es erscheint eine Datagridview mit den Verbindungen |

## Autocomplete Checkbox

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | User gibt in die Comboboxen die gewünschten Orte ein ohne Autocompletion | Die Stationen erscheinen erst, wenn der User auf den Button Suchen drückt. |
| 2 | User drückt auf die Checkbox «Autocompletion» | Checkbox kreuzt sich an. |
| 3 | User gibt in die Comboboxen die gewünschten Orte ein ohne Autocompletion | Währenddem der User in die Comboboxen schreibt, werden ihm die Stationen laufend angezeigt. |
| 4 | User drückt erneut auf die Checkbox | Autocompletion schaltet sich wieder ab |

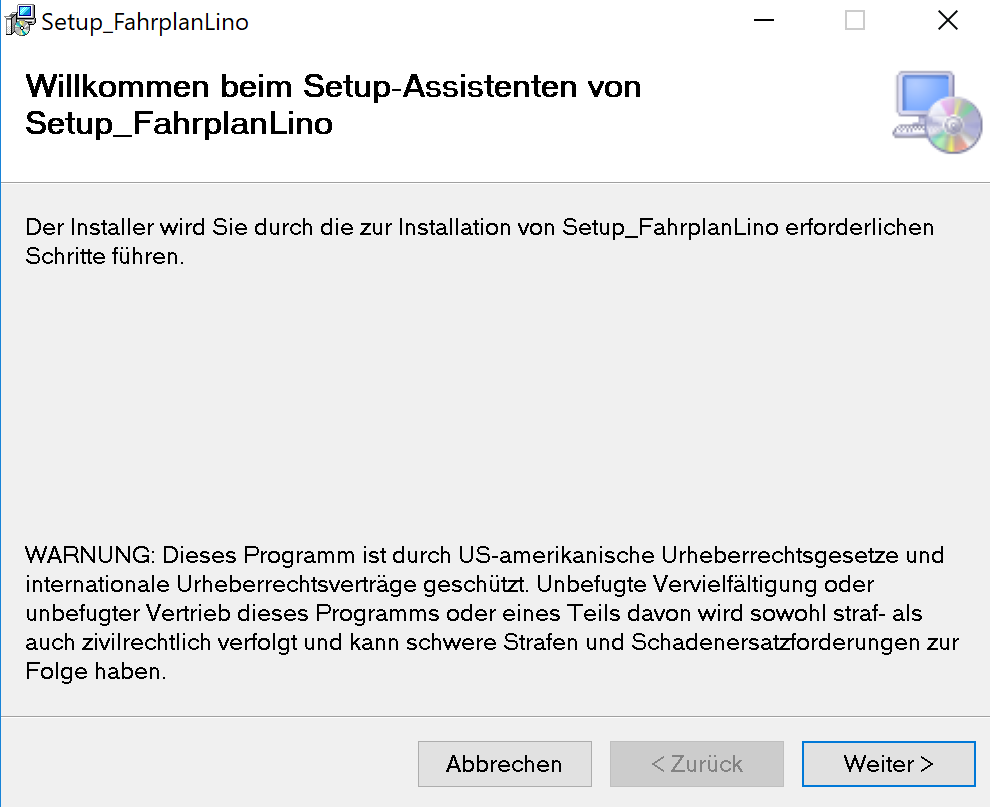
## Routenplaner anzeigen

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | User gibt in beiden Comboboxen eine gewünschte Station ein | Die Suche wird durch den Button oder die Autocompletion unterstützt |
| 2 | Der User klickt auf den Button «Verbindung anzeigen 🡪» | Es erscheint eine Datagridview mit den Verbindungen |
| 3 | User doppelklickt auf eine Spalte der Datagridview | Es öffnet sich ein Browserfenster und es erscheint die Route welche doppelklickt wurde. |

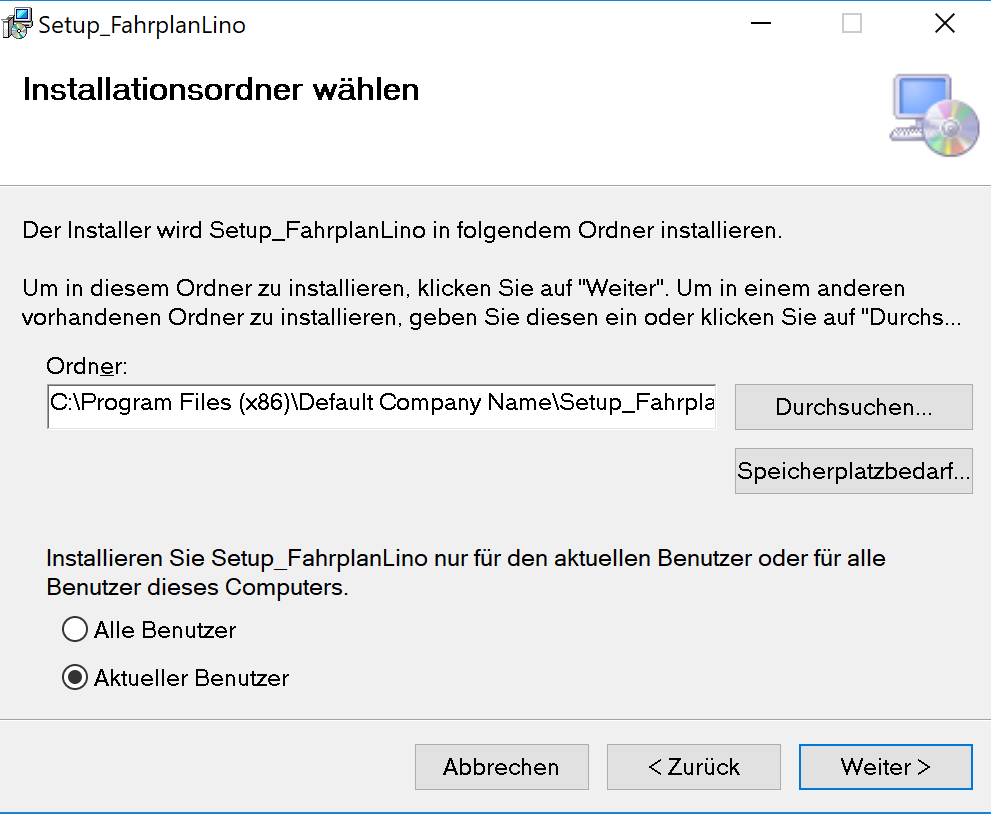
# Installationsanleitung

Um mein Programm zu verwenden, kann man einfach den Installationswizard herunterladen und diesen dann durchklicken. Der Installer ist auf dem meinem Github als Release zu finden.

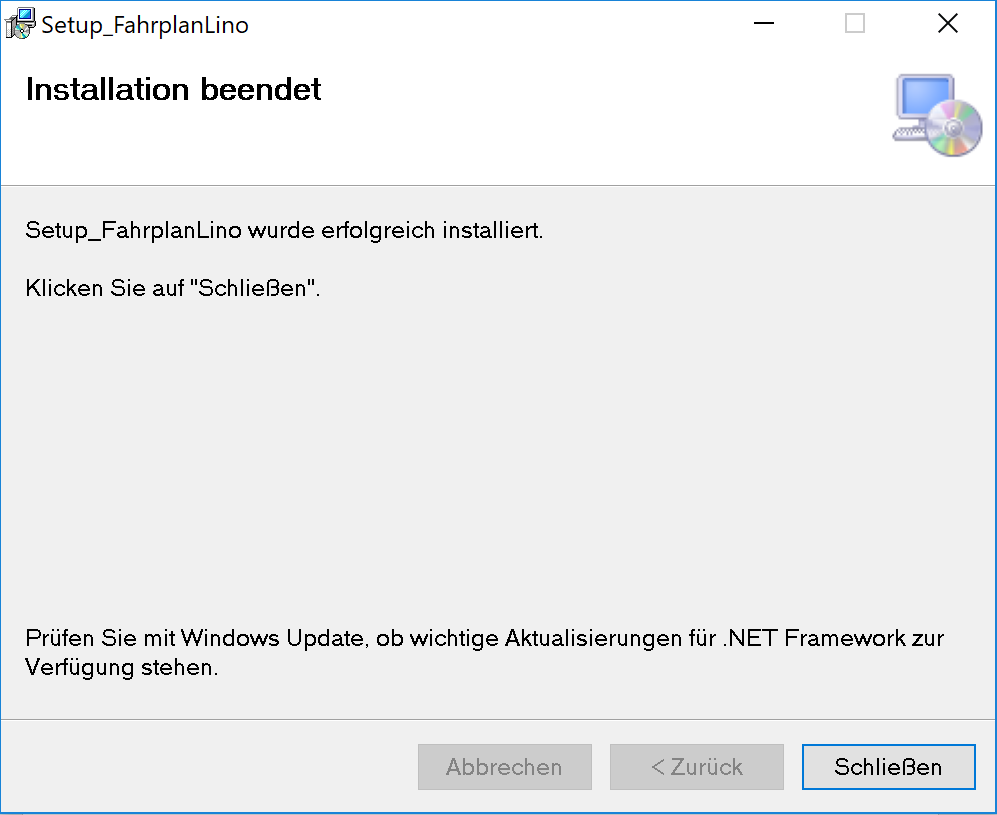
1. Als erstes startet man den Wizard und man sieht folgendes Bild. Wenn man alles gelesen hat, klickt man auf weiter.



1. Danach kann man auf folgendem Bild den Pfad auswählen und aussuchen auf welchem Benutzer die Applikation installieren will.



1. Im nächsten Bild nochmal auf weiter klicken.
2. Nun die Benutzerkontensteurung akzeptieren und dann wird die App installiert.
3. Als letzter Schritt nur noch auf schliessen drücken und man hat erfolgreich meine Applikation heruntergeladen.



# Reflexion

In diesem Abschnitt meiner Dokumentation finden Sie meine persönliche Meinung zu dem Projekt. Sie erfahren, ob ich selber damit zufrieden bin und was ich von der Idee halte.

## Positives

Ich finde die Punkte, welche ich eingebaut habe sind mir gelungen und mit dem was ich gemacht habe bin ich zufrieden. Auch die Idee eine öffentliche API zu verwenden finde ich eine sehr spannende und interessante Idee.

## Negatives

Mit was ich nicht so zufrieden bin ist, dass ich nicht so viele der Punkte erfüllen konnte. Jedoch ist das bereits alles.