2018

Vasily Kozlov

BiCA AG Rothenburg

4.12.2018

Modul 318 ÖV App erstellen



Inhalt

[1.0 Einführung 2](#_Toc531688830)

[2.0 Fehler/Bugs 2](#_Toc531688831)

[2.0.1 Bug 2](#_Toc531688832)

[2.0.2 Lösung 2](#_Toc531688833)

[2.0.3 Bug 3](#_Toc531688834)

[2.1 Lösung 3](#_Toc531688835)

[2.2 Bekannte Bugs 3](#_Toc531688836)

[3.0 GUI 3](#_Toc531688837)

[3.1 Vorstellung / Realität 4](#_Toc531688838)

[4.0 Welche Funktionen wurden umgesetzt? 5](#_Toc531688839)

[5.0 Aktivitätsdiagramm 6](#_Toc531688840)

[6.0 Use Case 7](#_Toc531688841)

[7.0 System-Tests 8](#_Toc531688842)

[7.1 Anforderung A001 8](#_Toc531688843)

[7.2 Anforderung A002 9](#_Toc531688844)

[7.3 Anforderung A003 10](#_Toc531688845)

[7.3 Anforderung A004 11](#_Toc531688846)

[7.4 Anforderung A005 12](#_Toc531688847)

[7.5 Anforderung A008 13](#_Toc531688848)

[8.0 Programmier-Richtlinien 14](#_Toc531688849)

[9.0 Installations-Anleitung 17](#_Toc531688850)

# Einführung

Die Aufgabe war es eine ÖV Applikation umzusetzen.

In dieser Dokumentation möchte ich euch aufzeigen wie ich bei der Arbeit vorgegangen bin. Unteranderem will ich meine Erfahrung mit euch teilen. Auch will ich euch beschreiben welche Probleme ich während der Arbeit hatte, was ich tat, um diese zu lösen und wie am Schluss das gesamt Resultat geworden ist.

# Fehler/Bugs

Ich würde nicht wirklich sagen, dass ich viele Probleme während des Programmierens hatte. Es ist eher so dass es einfach teils sehr mühsam war für mich die ein oder andere Methode umzusetzen. Nichts destotrotz, will ich paar meiner Fehler und vor allem deren Lösung mit euch teilen.

# 2.0.1 Bug

Mein wohl grösstes und das welches ich wohl am längsten dran war, ist das Problem, dass der Benutzer nach der Eingabe des Textes (Abfahrtsstation) einen Auto Fill Vorschlag bekam. Jedoch musste er diesen nicht mal auswählen, ohne Grund nahm das Programm den obersten Vorschlag, und setzte es rechts von der Eingabe des Benutzers.

# 2.0.2 Lösung

Die Lösung ist mir zwar jetzt mehr als logisch, dennoch habe ich davor lange nach einer Lösung suchen müssen. Ich habe es folgendermassen gelöst:

Nach der Eingabe vom Text durch Benutzer, rufe ich im TextChanged, eine Methode auf, welcher ich als Parameter unteranderem den Text von der Eingabe des Benutzers übergebe. Am Schluss von der Methode setze ich noch den Text (die Eingabe des Benutzers) wieder in das Eingabefeld. So hat er immer den Text, welchen er auch wirklich eingegeben hat.

# 2.0.3 Bug

Unteranderem hatte ich das Problem, das der Courser immer an den Anfang gerutscht ist, hat also der Benutzer das Wort «Lu» eingeben, so war der Courser wieder am Anfang vom Text, sprich «|Lu».

# 2.0.4 Lösung

Auch eine Lösung, welche mir nicht sofort in den Sinn gekommen ist. Die Lösung, welche ich genommen habe, ist glaube ich wirklich die simpelste, die man überhaupt finden kann. Wie in der Lösung 2.2 schon erklärt, habe ich als Parameter den Text vom Benutzer übergeben. Ich habe es sehr effizient genutzt, und ich mache eigentlich nicht mehr wie die Länge des Textes auslesen. Zum Schluss setze ich einfach nur noch den Courser an die letzte Stelle.

# 2.1 Bekannte Bugs

Mir ist leider nicht gelungen die Anforderung A008 einwandfrei zum laufen zu bringen. Es ist mir zwar bereits mal gelungen das mir das Resultat auf meine Mail geschickt wurde, jedoch funktioniert es nicht mehr.

Dieser ist der einzige Bug in meinem Programm. Ansonsten ist mir aber alles gut gelungen.

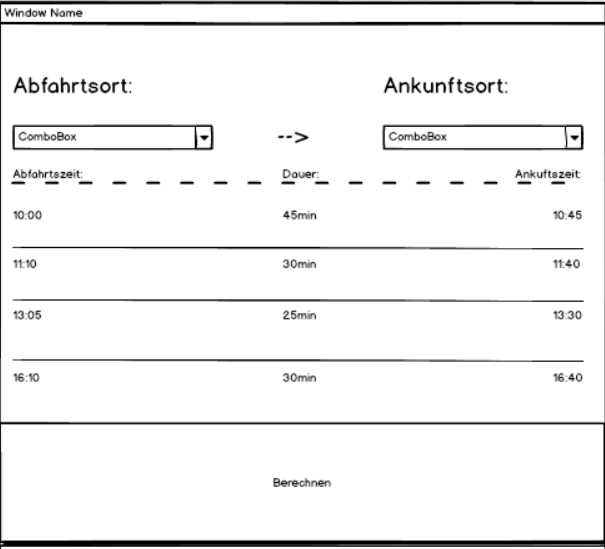
2.2 Fazit

Ehrlich gesagt fand ich die Probleme zu lösen gar nicht so schwierig, denn dank diesen habe ich so viel lernen können.

# GUI

Beim Designen von meinem Graphik User Interface hatte ich ein Motto: Schlicht und möglichst selbsterklärend. Meiner Meinung nach ist mir dies auch gelungen.

Auch während der Umsetzung bin ich meiner Vorstellung treu geblieben, denn meiner Meinung nach, Ist es trotz den Sachen, die ich zusätzlich mit in mein neues GUI eingebaut habe, immer noch sehr einfach zum Verstehen.



# Vorstellung / Realität

Während des Programmierens meiner App, habe ich aber weitere Elemente hinzugefügt, wie:

* Date Time
* Ein weiterer Button

Den einen Button habe ich hinzugefügt, da ich diesen in einer weiteren Anforderung brauche. Es soll dazu dienen das der Benutzer die Resultate via Email weiterleiten kann.

Auch habe ich eine «Uhr» eingebaut, welche es dem Benutzer ermöglicht, die Abfahrtszeit zu ändern.

Ansonsten bin ich meinem Mockup treugeblieben und habe auch nichts weiteres eingebaut. Das vor allem aus dem Grund, da ich den Benutzer nicht mit vielen Elementen verwirren möchte. Mein Ziel ist es nämlich das Programm möglichst selbsterklärend zu gestallten.

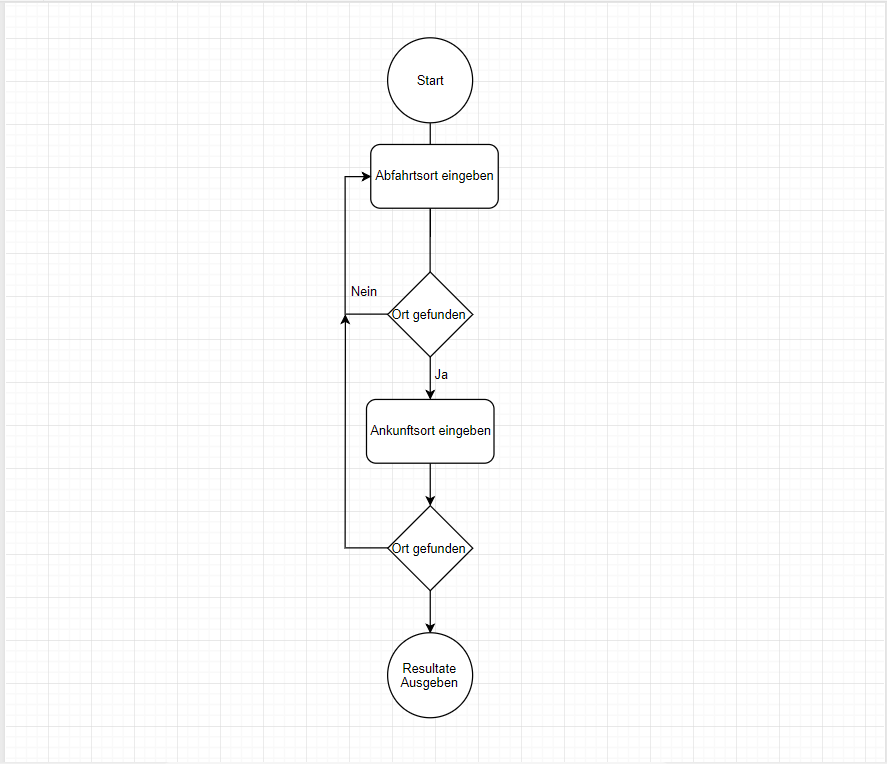


Einige Farben wurden während der Arbeit verändert. Jedoch ist das oben gezeigte Bild mein fertiges GUI.

# Welche Funktionen wurden umgesetzt?

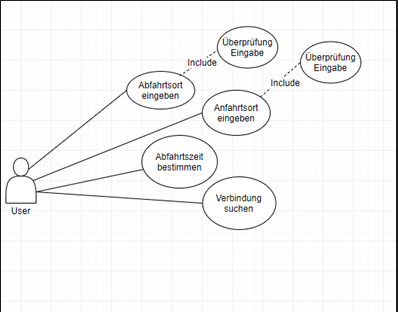
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Beschreibung** | **Umgesetzt? (Ja/nein)** | **Wie umgesetzt?** |
| A001 | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | Ja | Ich habe ein Auto Fill programmiert |
| 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 | Ich habe eine List Box im GUI, auf welcher die nächsten vier zutreffenden Verbindungen erscheinen. |
| 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 | Ich habe ein 2. Form mit nur einem DataGrid erstellt. Sollte der Benutzer nur eine Abfahrt aber keinen Ankunftsort ausgewählt haben, so erscheint Form 2 mit allen Verbindungen ab dem eingegebenen Abfahrtsort. |
| A004 | Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann. | Ja | Im Text Change Event rufe ich eine Methode auf, welche für den Auto Fill zuständig ist. |
| 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 | Ich habe ein DateTimePicker in mein GUI hinzugefügt. So kann der Benutzer die Abfahrtszeit eingeben und es werden Ihm nur die Resultate ab seiner eingegebenen Zeit angezeigt. |
| A006 | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. | Nein |  |
| A007 | Als ÖV-Benutzer möchte 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. | teilweise | Ich habe oben einen Button hinzugefügt, welcher eine mail an, die von mir hart gecodete Mail mit den Resultaten aus der List box schickt. |

# 5.0 Aktivitätsdiagramm

Mein Aktivitätsdiagramm ist ebenfalls recht simpel, da das was es jetzt macht, meiner Meinung nach genug ist. Sollte mir auffallen das mir etwas fehlt oder sollte ich am Schluss noch genug Zeit haben, so werde ich bestimmt noch weitere Features in das Programm einfügen.

# Use Case

Ähnlich meinem Aktivitätsdiagramm ist auch mein Use Case dementsprechend einfach gestaltet, da ich noch wenig Funktionen geplant habe.



# System-Tests

In diesem Abschnitt werde ich meine System-Tests durchführen und Dokumentieren. Pro Anforderung, welche ich jeweils erfüllt habe, werde ich ein System-Test machen.

# Anforderung A001

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| **1** | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «Lu» ein. | Es sollen Stationen welche «Lu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **2** | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Zu» ein. | Es sollen Stationen welche «Zu» enthalten vorgeschlagen werden.  Es muss so aussehen: |

# Anforderung A002

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| **1** | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «Lu» ein. | Es sollen Stationen welche «Lu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **2** | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Zu» ein. | Es sollen Stationen welche «Zu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **3** | Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | In der List Box über der «Suchen»-Taste werden die nächsten vier Verbindungen angezeigt (in diesem Beispiel die Verbindungen zwischen Luzern und Zürich) |

# 7.3 Anforderung A003

|  |  |  |
| --- | --- | --- |
| **1** | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «Lu» ein. | Es sollen Stationen welche «Lu» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **2** | Der Benutzer kann direkt auf «Suchen» klicken, ohne einen Ankunftsort eingegeben zu haben. | Es muss ein neues Fenster aufgehen, wo sich alle Verbindungen ab der Abfahrtstation anzeigen lassen.  Hier mit Beispiel als Abfahrtort Luzern: |

# Anforderung A004

|  |  |  |
| --- | --- | --- |
| **1** | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «L» ein. Die Suche beginnt ab dem ersten Char. | Autofill wird direkt gestartet. Es werden alle möglichen Vorschläge, welche zur Suche passen angezeigt.  Es muss so aussehen: |
| **2** | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Z» ein. Die Suche beginnt ab dem ersten Char. | Es sollen Stationen welche «Z» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **3** | Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | In der List Box über der «Suchen»-Taste werden die nächsten vier Verbindungen angezeigt (in diesem Beispiel die Verbindungen zwischen Luzern und Zürich) |

# Anforderung A005

|  |  |  |
| --- | --- | --- |
| **1** | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «L» ein. Die Suche beginnt ab dem ersten Char. | Autofill wird direkt gestartet. Es werden alle möglichen Vorschläge, welche zur Suche passen angezeigt.  Es muss so aussehen: |
| **2** | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Z» ein. Die Suche beginnt ab dem ersten Char. | Es sollen Stationen welche «Z» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **3** | Der Benutzer kann auch die Abfahrtszeit Beliebig ändern. Er kann also die ersten zwei stellen mit der Stunde befüllen und die nächsten zwei mit den Minuten befüllen.  Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | Es sollen nur Verbindungen angezeigt werden, welche innerhalb der Zeiteingabe des Benutzers liegen. (Hier ein Beispiel falls der Benutzer als Abfahrtszeit «09.35» eingegeben hat) |

# Anforderung A008

|  |  |  |
| --- | --- | --- |
| **1** | Der Benutzer klickt auf die Combobox «Abfahrtsort» und gibt «L» ein. Die Suche beginnt ab dem ersten Char. | Autofill wird direkt gestartet. Es werden alle möglichen Vorschläge, welche zur Suche passen angezeigt.  Es muss so aussehen: |
| **2** | Der Benutzer klickt auf die Combobox «Ankunftsort» und gibt «Z» ein. Die Suche beginnt ab dem ersten Char. | Es sollen Stationen welche «Z» enthalten vorgeschlagen werden.  Es muss so aussehen: |
| **3** | Der Benutzer kann auch die Abfahrtszeit Beliebig ändern. Er kann also die ersten zwei stellen mit der Stunde befüllen und die nächsten zwei mit den Minuten befüllen.  Der Benutzer muss auf die «Suchen»-Taste klicken, und ihm werden die Nächsten vier Verbindungen zwischen seinem Abfahrtsort und Ankunftsort angezeigt. | Es sollen nur Verbindungen angezeigt werden, welche innerhalb der Zeiteingabe des Benutzers liegen. (Hier ein Beispiel falls der Benutzer als Abfahrtszeit «09.35» eingegeben hat) |
| **4** | Danach kann der Benutzer seine gefundene Resultate via Mail teilen. | Diese Funktion ist mir zuhause bereits einmal gelungen. Ich habe die Zieladresse und den Absender HardCoden müssen. |

# Features

Zusätzlich zu den Anforderungen, welche wir alle bekommen haben, habe ich drei weitere Elemente in mein Programm eingebaut. Hier kurz die Zusammenfassung:

1. Sobald der User anfängt eine Eingabe zu tätigen, so fängt eine Musik an zu spielen
2. Einen Button mit Hilfe, wobei beim Klick auf dem Google geöffnet wird
3. Einen Sponsoring Button, bei welchem eine weitre Web-Seite geöffnet wird

# 9.1 Hilfe Button

|  |  |  |
| --- | --- | --- |
| 1 | Sobald der User eine Frage hat, so kann er einfach auf den «Hilfe»-Button klicken. | Sobald der «Hilfe»-Knopf gedrückt wurde, so wird im Browser ein Tab mit der Google-Webseite geöffnet. |

# 9.2 Sponsoring Button

|  |  |  |
| --- | --- | --- |
| 1 | Sollte der User wissen wollen wer diese großartige Webseite sponsort, so kann er auf den Sponsor-Knopf klicken. | Drückt der User auf den Sponsoring Knopf, so wird eine Webseite von unserem Sponsor geöffnet. |

# 9.3 Musik

|  |  |  |
| --- | --- | --- |
| 1 | Sobald der User anfängt seine Abfahrtsstation einzugeben. | So wird fängt eine Musik(Swan Lake) im Hintergrund an zu spielen. |

# Programmier-Richtlinien

Hier sind die Programmier-Richtlinien, welche ich mit Gerry für uns abgemacht haben:

Wieso diese Richtlinien?

Der Zweck der Richtlinien ist es Erleichterung der Arbeit von Teammitglieder im Projekt. Unter Erleichterung versteht man die Lesbarkeit, Verständlichkeit und Wartbarkeit vom Code. Einhaltung vom vorgängig definierten Rechtlinien für variablen, Prozeduren und Klassen Namen, kann Lesbarkeit und Wartbarkeit des Codes wesentlich verbessern. Während der Wartung ist die Einhaltung eines definierten Programmstiels noch wichtiger als während der Entwicklung. Ein Programmiersteil sollte nicht unbedingt wie eine programmier-Richtlinien ausgelegt werden. Verstösse dagegen sollten erlaubt sein, sofern diese gut begründet sind.

Unsere Richtlinien

* Immer bevor man eine Klammer nach einem Funktionsaufruf setzt, einen Abstand davor setzen z.B. / MachEineRolle (a)
* Am Anfang und am Ende einer Klammer wird *kein* Abstand gemacht. z.B. so: MachEinRad (a) und nicht so: machEinrad ( a )
* Sollte man einen generischen Typ nutzen, so wird der Datentyp ohne Abstand direkt angehängt z.B. <int>List Beispiel = new List<int>
* Eine geschweifte klammer bekommt immer eine einzelne Zeile. So fängt man immer eine neue Zeile an sobald man eine geschweift klammer platziert hat.

if (a)

{

code ();

code ();

}

if (a) {

code ();

code ();}

* Klassennamen werden immer mit einem sinnvollen Namen und mit einem Grossbuchstaben definiert.

Public Class Car -> Klasse über Autos

Public Class tastatur -> Klasse über Mäuse

* Die Funktion sollte nur das machen, wie sie auch heisst.

Private void getAuto()

{

String hallo;

Return Hallo;

}

Private void GetAuto()

{

String auto;

Return auto;

}

* Funktionen sollten so kurz wie möglich gemacht werden. (max. 20 Zeilen)

Private void getAuto()

{

String auto;

Return auto;

}

Private void getAuto()

{

String auto;

String auto;

String auto;

String auto;

String auto;

String auto;

String auto;

String auto;

…

Return auto;

}

* Methoden sowie Funktionen fangen immer mit einem Grossbuchstaben an.

Private void GetAuto()

{

String auto;

Return auto;

}

Private void GETAUTO()

{

String auto;

Return auto;

}

* Kommentare möglichst kurz und sinnvoll schreiben.

//Hier habe ich einen leeren string initialisiert

//Autocomplete wird erstellt

* Variabeln werden immer kleingeschrieben.

String Eingabe = «»;

String input = «»;

* Code darf keine Warnings enthalten.

Am Schluss dürfen auf der Fehlerlistekeine Warnings enthalten sein.

* Code wird «auf Englisch geschrieben» d.h. dass Methoden, Variabeln und Funktionen bekommen einen Englischen Namen. (#Kommentare dürfen auf Englisch verfasst werden.)

string Bnutzereingabe = ComboBox\_Input.text;

string userinput = ComboBox\_Input.text;

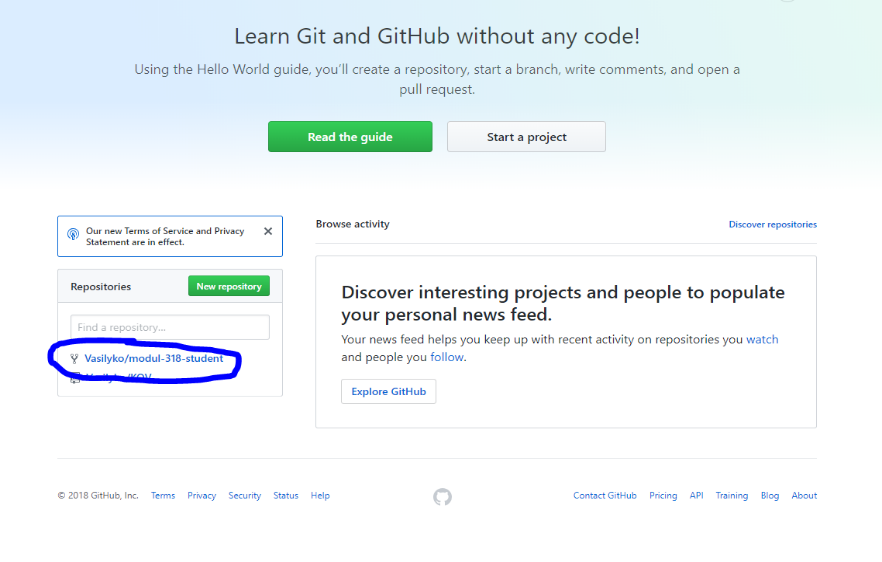
* Variabeln müssen sinnvoll benannt werden. So soll jeder der sie zum ersten Mal sieht direkt wissen für was diese Variabel eingesetzt werden.

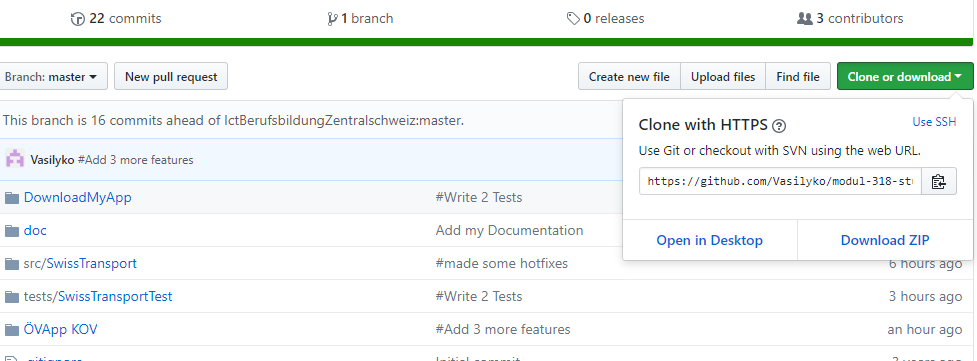
int x = ID;

int userid = ID;

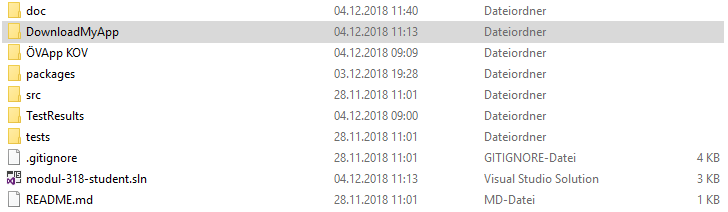
# Installations-Anleitung

Hier eine «Schritt für Schritt» Anleitung, wie man sich meine «SBB»-App lokal herunterladen kann:

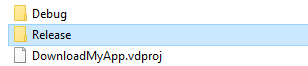
1. Man geht auf github.com und klickt auf «vasilyko/modul-318-student»
2. Danach muss man es einfach nur noch als Zip-Datei runterladen



1. Man öffnet den Speicherort, in dem von der Dtei welche man als zip heruntergeladen hat
2. Dort öffnet man den Ordner «DownloadMyApp»



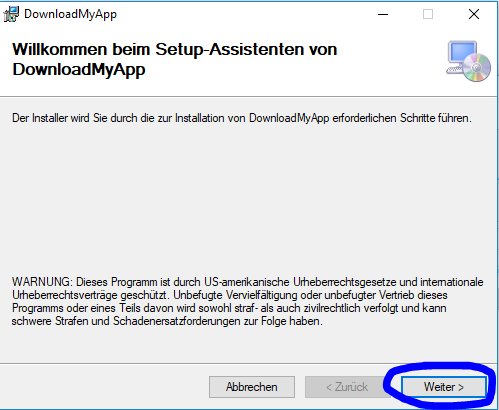
1. Danach öffnet man den Ordner «Release»



1. Als nächstes doppelklickt man die Datei «DownloadMyApp»



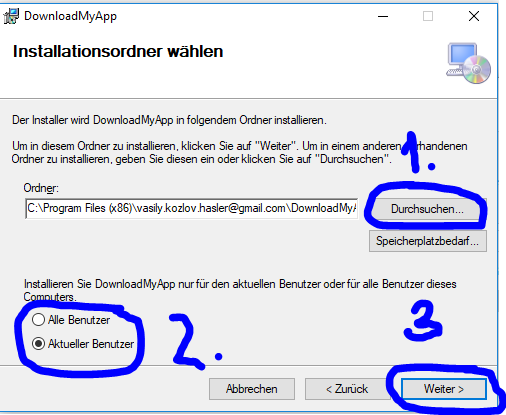
1. Man klickt auf «Weiter»



1. Benutzereinstellungen

Entweder man kann seinen eigenen Speicher-Pfad Entweder man kann seinen eigenen Speicher-Pfad angeben oder den vom Programm vorgeschlagenen lassen.

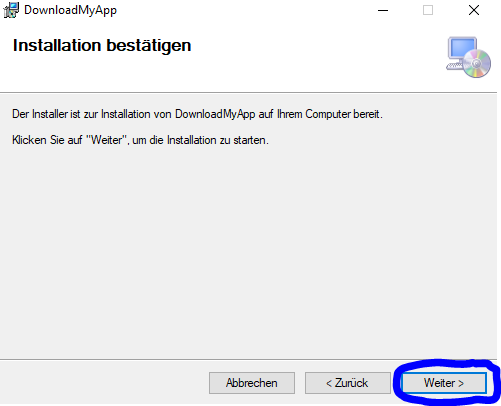
angeben oder den vom Programm vorgeschlagenen lassen.



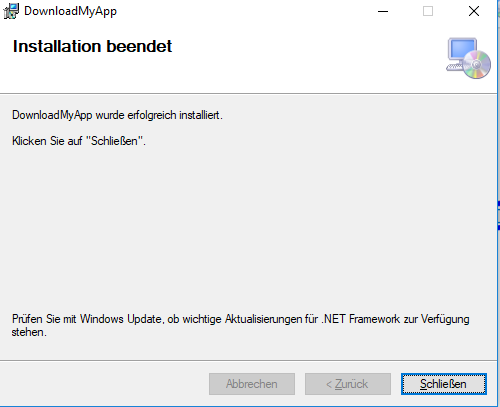
Hier kann man einstellen ob alle Benutzer auf dem Computer/Laptop von meiner «SBB»-App profitieren, oder nur du selbst.

Sollte man die Einstellungen bereits gemacht haben/auf den Standard Einstellungen gelassen haben und man ist zufrieden mit dem, so kann man auf «Weiter» klicken

1. Danach noch auf Weiter klicken



1. Schlussendlich nur noch auf schliessen klicken und schon hat man meine Wunderschöne App heruntergeladen.



Ich hoffe ich konnte dir weiter helfen 😉