



Zug-Fahrplan Applikation ÜK Modul 318

Von Pirmin Wegner



24. APRIL 2018
BÉNÉDICT SCHULE LUZERN

Inhaltsverzeichnis

1. Einleitung.....	1
2. Vorbereitung.....	2
2.1. Skizze (Mockup)	2
2.2. Use-Case/ Activity-Diagramm	3
3. Funktionen der Applikation	5
3.1. Geforderte Funktionen	5
3.2. Testfälle.....	6
4. Installations- und Deinstallationsanleitung.....	6
5. Fehleranalyse.....	9
6. Fazit	10

1. Einleitung

In dem ÜK 318 ging es um das Objektorientierte Programmieren mit Windows Forms. Hierbei bekamen wir das Projekt, 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. Vorbereitung

2.1. Skizze (Mockup)

Fahrplan:

Fahrplan-Auskunft

Abfahrtsort

Q Adligenswil

Adligenswil, Dorf

Adligenswil, Stuben

Luzern,

Luzern, Bahnhof

Fahrziele

Q Luzern

Adligenswil, Dorf

Adligenswil, Stuben

Luzern,

Luzern, Bahnhof

Fahrten Suchen

Erweiterte Suche

Abfahrtszeiten:

14:00

15:00

16:00

Ankunftszeiten:

14:30

15:30

16:30

Abfahrtstafel:

Fahrplan-Auskunft

Ab Ort:

Abfahrtsort

Q Adligenswil

Adligenswil, Dorf

Adligenswil, Stuben

Luzern,

Luzern, Bahnhof

Ab:

Abfahrtszeit:

09:00

10:00

11:00

12:00

An:

Ankunftszeit:

10:00

11:00

12:00

13:00

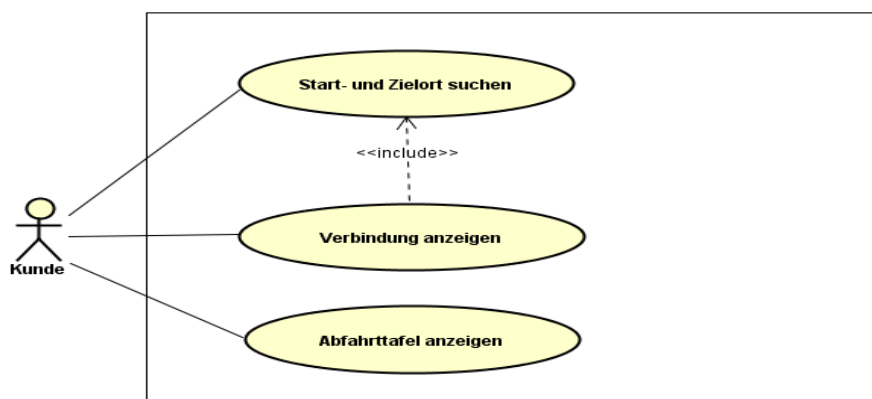
Fahrzeiten Suchen

Abfahrtsort	Umstiege	Ab	An
Von Adligenswil (Stuben) nach Luzern (Luzernerhof)	1	12:12	12:42
Von Adligenswil (Dorf) nach Luzern (Luzernerhof)	1	12:46	13:17
Von Adligenswil (Stuben) nach Luzern (Luzernerhof)	1	13:12	13:42
Von Adligenswil (Stuben) nach Luzern (Luzernerhof)	1	13:46	14:17

Wenn man nun das fertige Programm ansieht, kann man beim Fahrplan leichte Änderungen erkennen, jedoch gravierende bei der Abfahrtstafel. Dies liegt vor allem daran, dass ich meist die Angewohnheit habe große Ideen zu haben, die jedoch viel komplizierter in der Projektphase zu erfassen sind. Will man nun die Funktionen einbauen macht man zuerst sogenannte Use-Case und Activity Diagramme.

2.2.Use-Case/ Activity-Diagramm

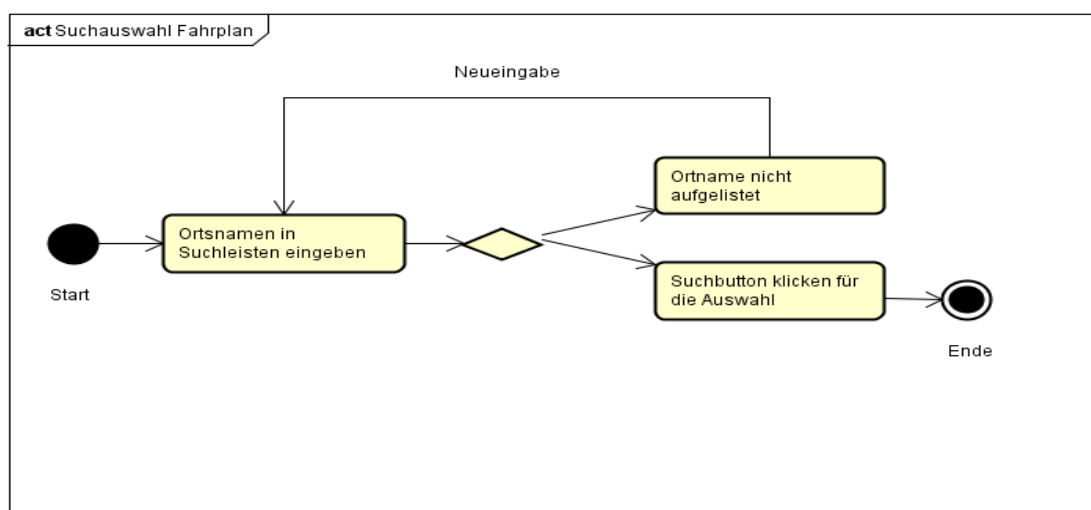
Use Case:



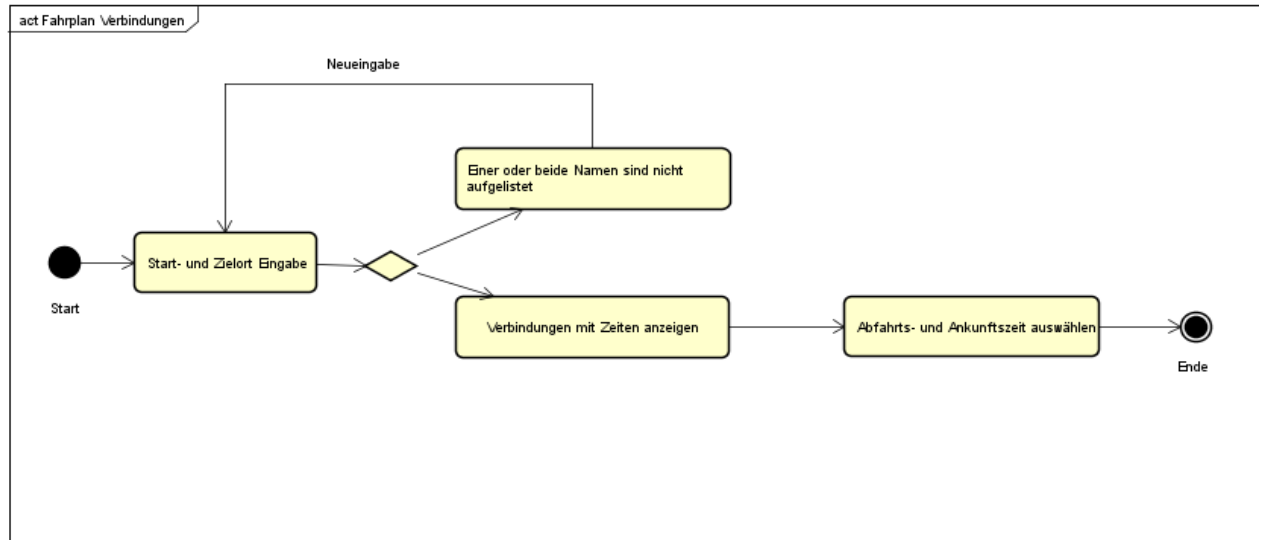
Hier erkennt man die drei Aktivitäten, welche der User mit meiner Applikation ausführen kann. Er kann nach Orten suchen, deren Verbindungen anzeigen lassen sowie mithilfe einer Abfahrtsstafel sich mögliche Zielorte und Abfahrtszeiten dahin ansehen.

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

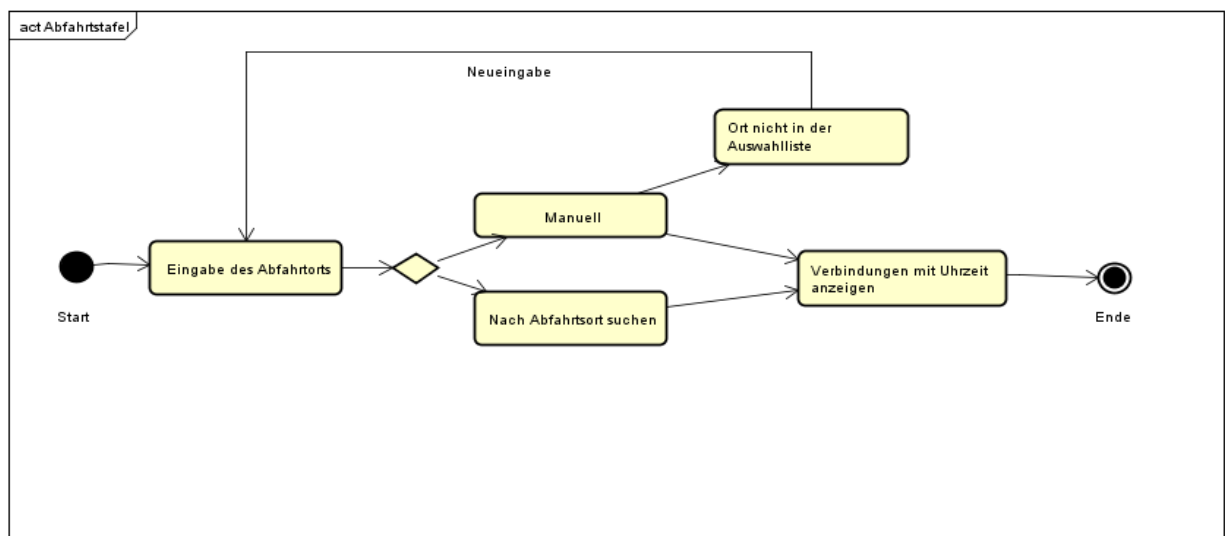
1. Abfahrts- und Zielort suchen



2. Verbindungen anzeigen



3. Abfahrtstafel anzeigen



3. Funktionen der Applikation

3.1. Geforderte Funktionen

A001	Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss.	1
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.	1
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.	1
A004	Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann.	2
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.	2
A006	Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht.	3
A007	Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann.	3
A008	Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können.	3

A001 Suchfunktion

- Erkennbar an den zwei Comboboxen und den dazugehörigen Suchbuttons

A002 Verbindung Suche

- Erkennbar an der DataGridView auf dem unteren Bildschirm des Fahrplans
- Aktualisiert sich mit geänderten Eingabewerten

A003 Abfahrtsplan

- Mittels dem Button Abfahrtsplan kann der User sich eine Auflistung von Zielorten von seinem angegebenen Wert anzeigen lassen

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

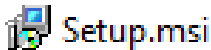
3.2. Testfälle

Schritt	Testfall	Erwartetes Resultat
1	Stadt ist bekannt, jedoch nicht die richtige Schreibweise	Suchfeld zeigt mögliche Treffer auf
2	Verbindung suchen zwischen Zug und Luzern	4 Ergebnisse ab der aktuellen Uhrzeit werden angegeben
3	Man kennt die Stadt, aber nicht die Haltestelle	Durch Suchfunktion werden 10 Vorschläge für Haltestellen angegeben
4	Man möchte alle Verbindungen sehen der aktuellen Ortschaft	Es werden die nächsten 10 Verbindungen vom Abfahrtsort angezeigt

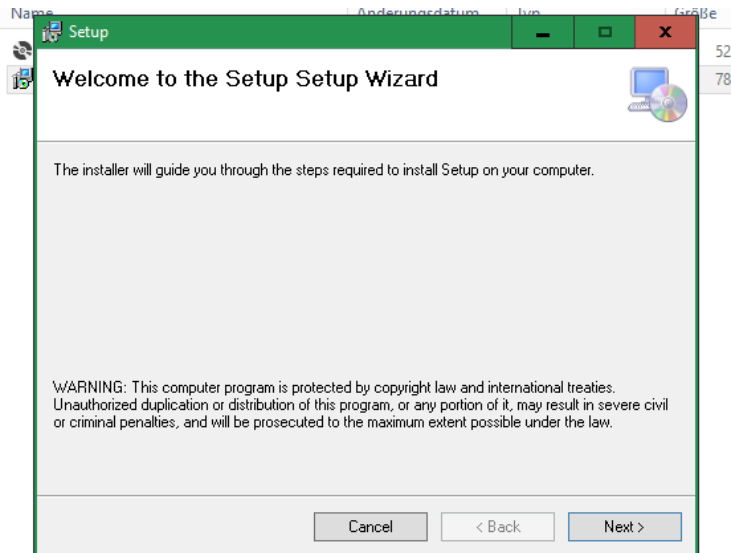
4. Installations- und Deinstallationsanleitung

Da man das Programm auch als Setup Datei abspeichern kann, ist es für den User nun viel einfacher diese Applikation zu verwenden. Um dieses Programm nun ausführen zu können, muss man folgende Schritte befolgen:

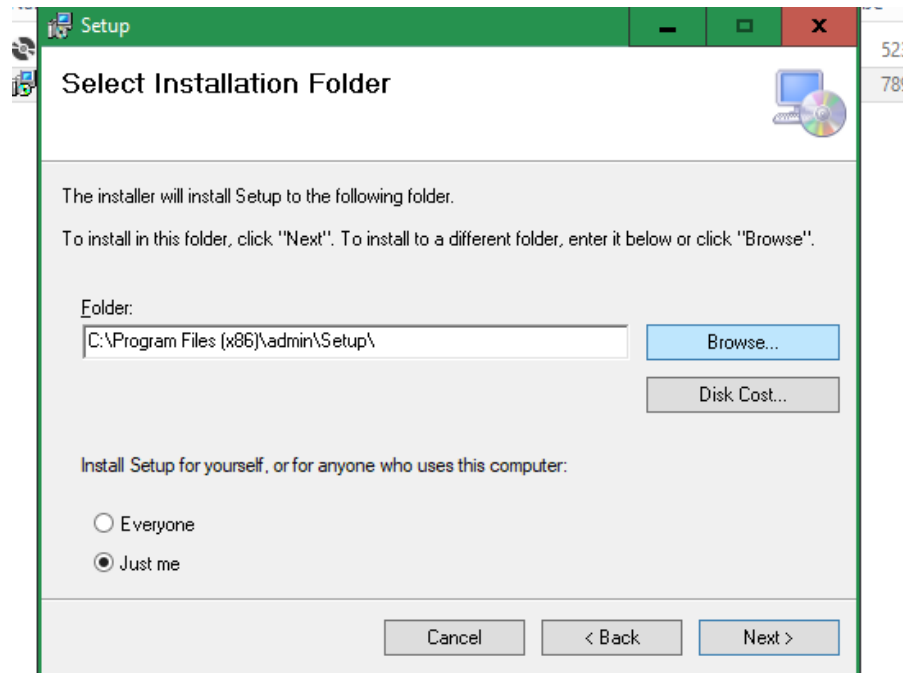
1. Setup.msi ausführen per Doppelklick



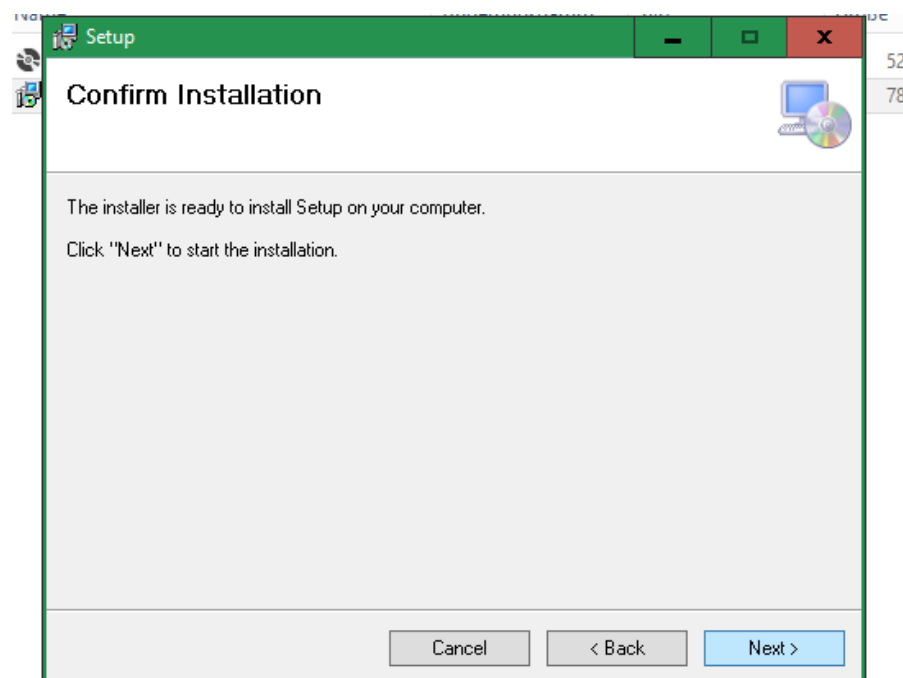
2. 'Next' Klicken



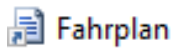
3. Installationsordner per 'Browse' einstellen und anschliessend 'Next' klicken



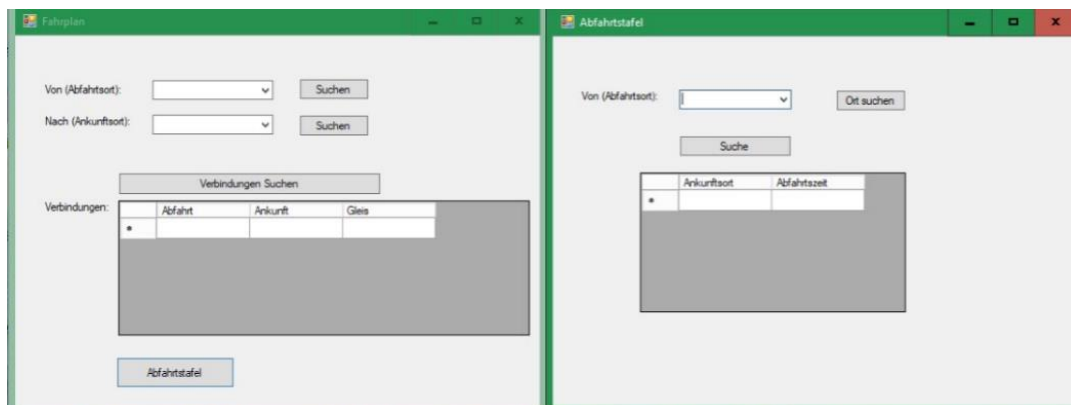
4. Installation abschliessen per Klick auf 'Next'



Sollte man nun die Installation erfolgreich durchgeführt haben, so müsste am zuvor angegebenen Ordner folgende Datei zu finden sein:

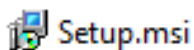


Führt man die Datei aus, öffnet sich das gewünschte Fahrplan-Programm, welches folgendermassen aussieht:

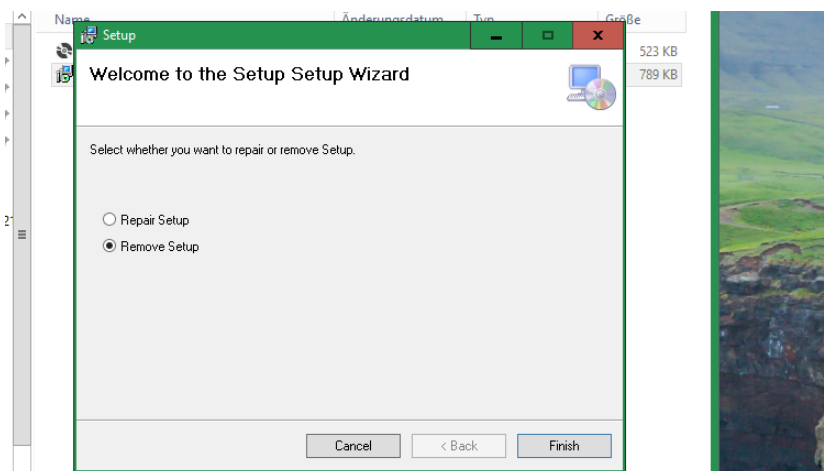


Möchte man nun diese Applikation nicht mehr auf dem Computer, so führt man die Deinstallation folgender Massen aus:

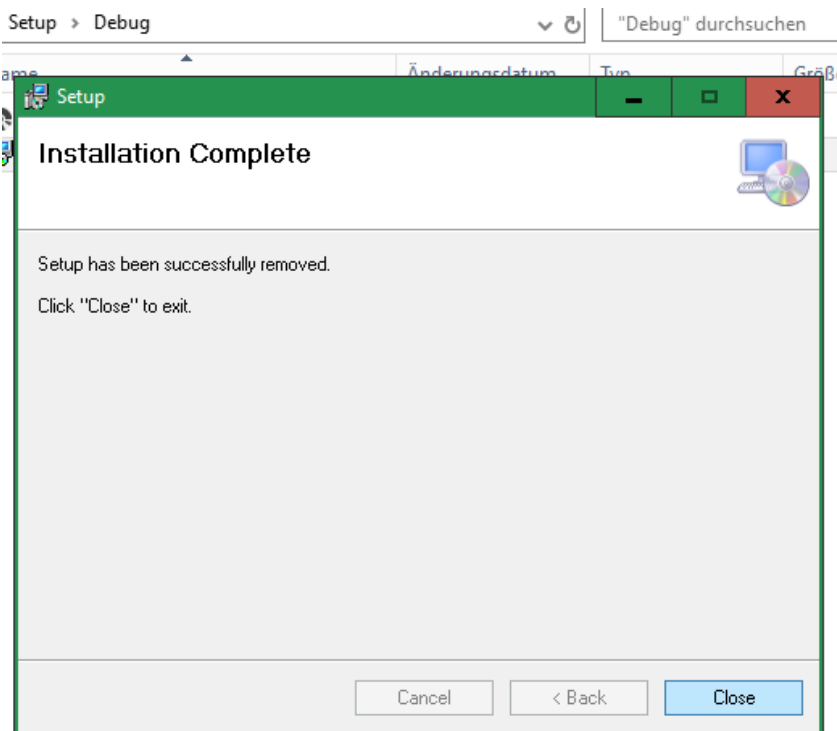
1. Doppelklick auf die Setup.msi Datei



2. Remove Setup anwählen und auf 'Finish' klicken



3. Deinstallation abschliessen per Klick auf 'Close'



'88 KB)

5. Fehleranalyse

Da Ich nicht auf dem gleichen Wissensstand im Gebiet Objektorientiertes Programmieren habe wie der grösste Teil der restlichen Klasse, fehlte mir am Ende die Zeit, um noch Zusatzfeatures wie zum Beispiel das Aufrufen einer Google Maps Seite oder das Versenden der Ergebnisse für die Verbindungen per Email zu versenden in mein Programm einzubauen. Da jedoch die Hauptanforderungen bestens geprüft wurden, kam es sehr selten zu Fehlern.

Einer der kleineren wäre zum Beispiel, dass das Programm keine Angabe gibt, falls die Zielhaltestelle per Bus zu erreichen ist. Das ist aber nicht schlimm, da wie der Name schon sagt es eine SBB-Fahrplan Applikation ist und es hierbei hauptsächlich um die Zugverbindungen geht.

Ein weiterer minimaler Fehler ist die ungenaue Einstellung der Abfahrtszeiten, da hierbei nur die aktuelle Uhrzeit vom Benutzersystem überprüft wird und ab dieser Zeit vier Verbindungen herausgesucht werden.

6. 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 5/7 😊