

SBB APP

ÜK 318



Einleitung

Liebe Leserinnen und Leser

Ich habe im 2. Jahr meiner Informatik Lehre den ÜK 318 bei Urs Nussbaumer & Lukas Zingerli besucht. Zur Bewertung dieses Kurses musste ich ein SBB App mithilfe einer API in C# schreiben. Das Ziel dieser Arbeit ist so viel wie möglich aus dem Projekt zu lernen. Ich werde die Applikation mit WPF anstelle von Windows Forms schreiben um mehr zu profitieren da wir Forms in der Schule lernen.

In diesem Dokument werde ich mein Projekt grob erläutern. Ich werde beschreiben welche Funktionen ich umgesetzt habe, wo ich Probleme hatte und welche Bugs evtl. noch vorhanden sind. Sie finden in meiner Dokumentation ebenfalls die Use Cases und die Aktivitäts Diagramme für die ersten 3 Anforderungen.

Am Schluss des Dokumentes sind meine Testfälle die ich verständlich und eindeutig nachvollziehbar geschrieben habe.

Ich bin mir sicher, dass ich Ihr Interesse an meiner Arbeit geweckt habe und Sie Freude und Spass bei der Bewertung haben werden.

Bei Fragen und Verbesserungsvorschlägen wenden Sie sich bitte an mich.

Vielen Dank für Ihr Interesse



Iman Lünsmann, Informatiklernende der CSS Versicherung

Inhalt

Zweck des Dokuments.....	4
Anforderungen	4
Umsetzung.....	5
Planung.....	5
Anforderung A005	5
Validierung	5
Fehler/ Bugs.....	5
Mockups	6
Use Cases.....	7
Aktivitätendiagramme.....	8
Testfälle	9
Testfall «Station suchen»	9
Testfall «Verbindung suchen»	9
Testfall «Abfahrtstafel»	9
Programmierrichtlinien	10
Naming Conventions	10
Declaration	10
Comments	10
Installationsanleitung Installation	11
Deinstallation	12

Zweck des Dokuments

Das Dokument dient der Bewertung des Ük's 318. Im folgenden Teil sind die Anforderungen an das Programm aufgelistet.

Anforderungen

Hier hat der Kunde seine Anforderungen nach Priorität definiert.

Prioritäten: 1 = must / 2 = should / 3 = nice to have

ID	Beschreibung	Priorität
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

Umsetzung

Planung

Zu Beginn meiner Arbeit habe ich mit den ersten 3 Anforderungen begonnen. Ich habe das Mockup dafür gezeichnet und dann mit der Zeit immer angepasst.

Die Mockups befinden sich auf Seite 6.

Funktionen

Umgesetzt: A001, A002, A003, A004

Ich habe die Anforderung A001 umgesetzt und etwas später durch A004 ersetzt. Hier ging es darum die Station direkt bei der Eingabe zu suchen.

Die zweite Anforderung habe ich auch erfüllt. Der Benutzer gibt eine Abfahrt und eine End Station ein. Wenn der Benutzer dann auf den «Verbindung suchen» Button klickt erscheinen die nächsten paar Verbindungen.

Die Anforderung A004 habe ich ebenfalls erfüllt. Der Benutzer kann auf den nächsten Tab «Abfahrtstafel» klicken und seine Station eingeben. Es erscheinen anschliessend alle Verbindungen ab dieser Station.

Anforderung A005

Die Anforderung 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.» konnte ich aus Zeitgründen nicht mehr umsetzen.

Ich habe am Dienstagmorgen (24.04.18, letzter Kurstag) einen DatePicker in mein WPF eingefügt um die Verbindungen nach Datum Abzufragen.

Ich musste anschliessend die API so anpassen dass man bei den Verbindungen auch nach Datum Suchen kann. Dies hat aber anfangs nicht so geklappt wie ich wollte. Ich habe anschliessend ein wenig Zeit dafür investiert um alles so anzupassen sodass es funktioniert. Schlussendlich habe ich alles wieder rückgängig gemacht da ich aus Zeitgründen die Validation noch schreiben musste. In meinem Code befinden sich keine Testversuche der Anforderung A005 da dies gegen CleanCode verstösst.

5

Validierung

Mein Programm gibt keine Rückmeldung falls eine Eingabe komplett falsch eingegeben wurde.

Definition komplett falsch: Luzern -> Lü>ern

Das Programm stürzt auch nicht ab. Der User versteht anschliessend von alleine dass die Eingabe falsch ist.

Fehler/ Bugs

Mein Programm beinhaltet keine bekannten Fehler oder Bugs

Ich habe alles mit den Testfällen auf der Seite 9 geprüft.

Mockups

Mockup für die Fahrplanseite.

Fahrplan
⌵ ⌵ ⌵

Fahrplan

From

Luzern ▾

Luzern, Bahnhof
Luzern, Gütsch
Luzern, Verkehrshaus

Suchen

Verbindung suchen

To

Basel ▾

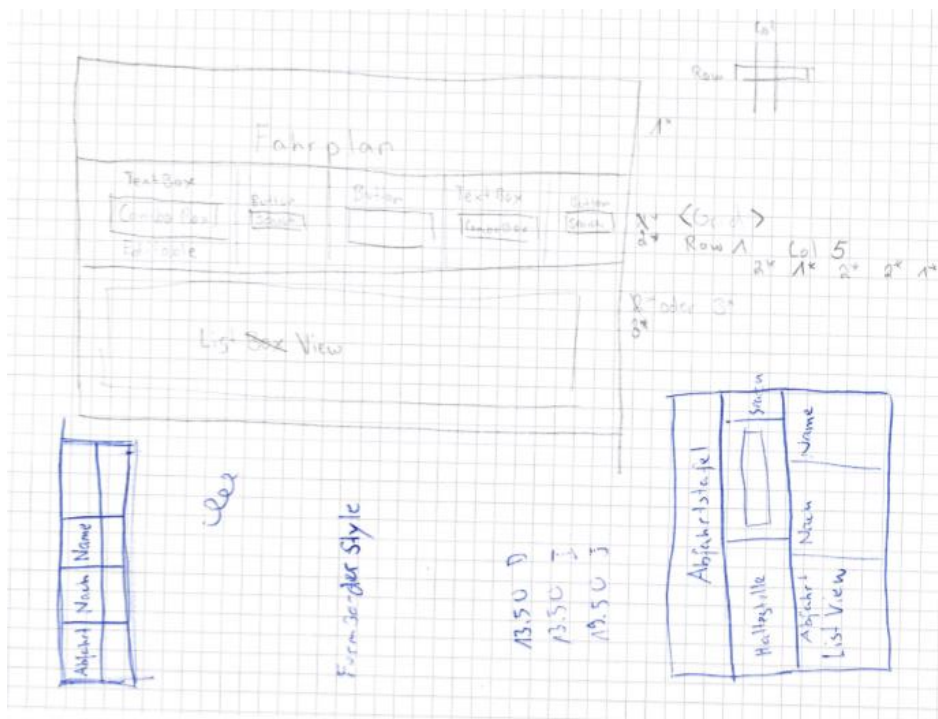
Basel, SBB
Basel, Badischer Bhf

Suchen

Abfahrt	Ankunft	Richtung	Dauer	Gleis
13.00	14.03	Basel SBB	1h 3min	7
15.25	16.00	Olten	35min	5

Mockup für die Planung der Seite mit einem Grid View und zweites Mockup unten rechts in der Ecke für die Abfahrtstafel.

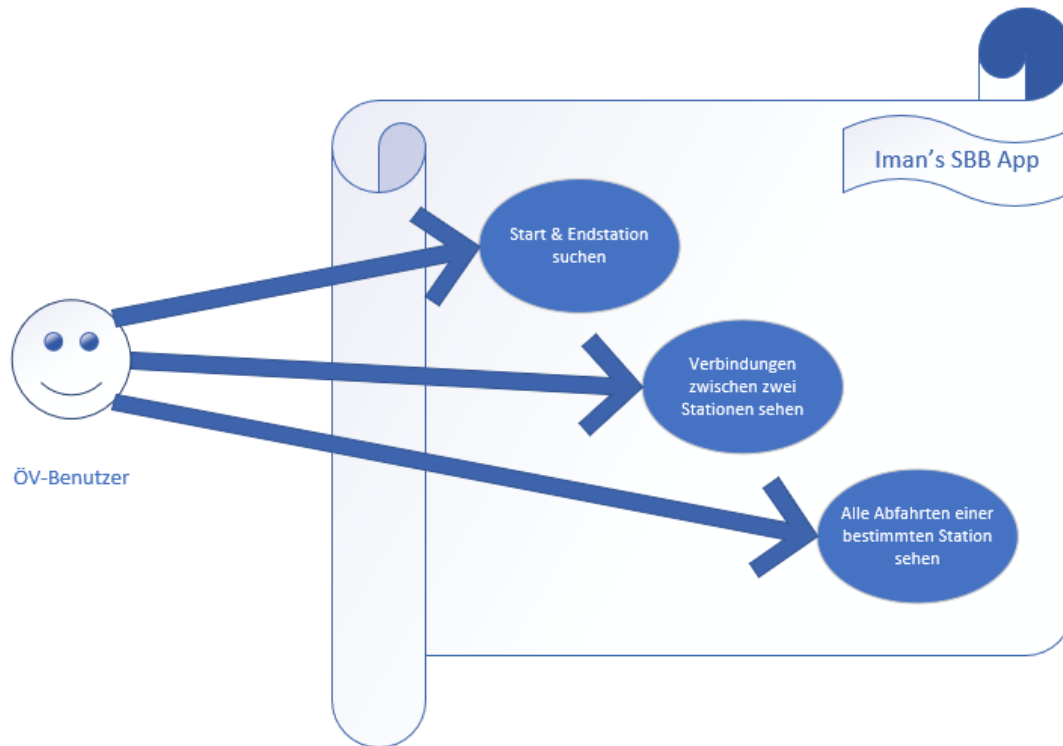
6



Use Cases

Ein Use Case ist das nach außen sichtbare Verhalten eines Systems aus Sicht der Nutzer.

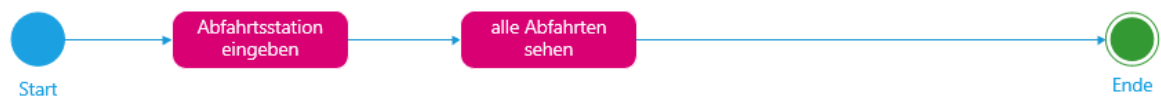
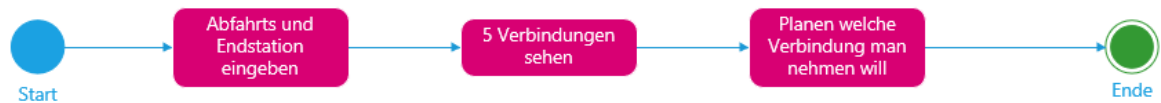
Hier befinden sich die Use Cases der ersten 3 Anforderungen.



Aktivitätendiagramme

Ein Aktivitätsdiagramm ist ein Verhaltensdiagramm der UML. Diese stellt die Vernetzung der Aktionen und deren Verbindungen grafisch dar.

Hier befinden sich die Aktivitätsdiagramme für die ersten 3 Anforderungen.



Testfälle

Testfall «Station suchen»

Vorbedingungen

Der Benutzer muss die Applikation gestartet haben und befindet sich auf der Fahrplan oder Abfahrtstafel Seite.

Schritt	Aktivität	Erwartetes Resultat
1	Der Applikation Benutzer tippt einen Buchstaben in eine der Comboboxen ein.	Die Combobox öffnet sich sobald der Fokus auf sie gesetzt wurde und zeigt die Vorschläge an
2	Der Applikation Benutzer gibt eine Falsche Eingabe ein Von: ^^--	Die Combobox zeigt keine Vorschläge an und wirft keinen Fehler.

Testfall «Verbindung suchen»

Vorbedingungen

Der Benutzer der Applikation befindet sich auf der Fahrplan Seite und hat die Abfahrts und Endstation eingegeben.

Schritt	Aktivität	Erwartetes Resultat
1	Der Benutzer klickt auf den «Verbindung suchen» Button.	Die ListBox wird mit den ersten paar Verbindungen abgefüllt. Es werden die Abfahrts- und Ankunftszeiten mit dem heutigen Datum angezeigt. Man sieht ebenfalls wie lange die Fahrt dauert und auf welches Gleis man muss.
2	Der Benutzer klickt auf den «Verbindung suchen» Button und hat falsche Angaben eingegeben.	Keine Fehlermeldung, es passiert nichts. Der Benutzer merkt anschliessend das er einen Fehler gemacht hat.
3	Der Benutzer klickt auf den «Verbindung suchen» Button und hat keine Internetverbindung.	Es erscheint eine Fehlermeldung das der User keine funktionierende Internetverbindung besitzt.

9

Testfall «Abfahrtstafel»

Vorbedingungen

Der Benutzer der Applikation befindet sich auf der Abfahrtstafel Seite und hat seine Station eingegeben.

Schritt	Aktivität	Erwartetes Resultat
1	Der Benutzer klickt auf den «suchen» Button.	Die ListBox wird mit allen Verbindungen die in nächster Zeit abfahren abgefüllt. Man sieht die Anfahrtszeit, die Verbindung wohin und was für ein Zug dorthin fährt (S-Bahn, Regio Express usw.).
2	Der Benutzer klickt auf den «Verbindung suchen» Button und hat falsche Angaben eingegeben.	Keine Fehlermeldung, es passiert nichts. Der Benutzer merkt anschliessend das er einen Fehler gemacht hat.
3	Der Benutzer klickt auf den «Verbindung suchen» Button und hat keine Internetverbindung.	Es erscheint eine Fehlermeldung das der User keine funktionierende Internetverbindung besitzt.

Programmierrichtlinien

Naming Conventions

Variablen: PascalCase, keine Unterscheidung von private public oder protected

Eigenschaften: CamelCase

Methoden: CamelCase, Klammern nach C# Richtlinien,

Klassen: Erster Buchstabe gross ansonsten C# Richtlinien

GUI-Controls: CamelCase

Declaration

Lokale und globale Variablen immer ganz oben in der Methode und in der Klasse.

Constructor nach den Variablen deklariert.

Comments

Kommentare nur wenn nötig bei unklaren Sachen



Statements

If-else / for / foreach / while / switch etc.: Nach C# Syntax Klammern auf neuer Linie.

Installationsanleitung

Installation

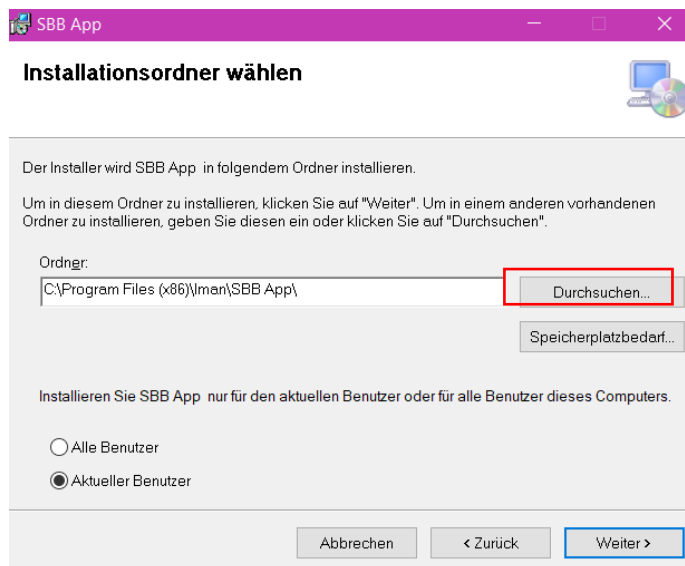
Nach dem Download und des entzippen des Projekts begibt man sich zum SBB App Setup Ordner und anschliessend zum Installations Ordner.

> Dokumente > GitHub > modul-318-student > SBB App Setup > Installation				▼ ↺
Name	Änderungsdatum	Typ	Größe	
 SBB App Setup.msi	24.04.2018 12:50	Windows Installer-...	1'119 KB	
 setup.exe	24.04.2018 12:50	Anwendung	752 KB	

Hier sieht man ein setup.exe und ein SBB App Setup.msi.

Um das Programm zu installieren doppelklickt man die Setup.exe Anwendung.

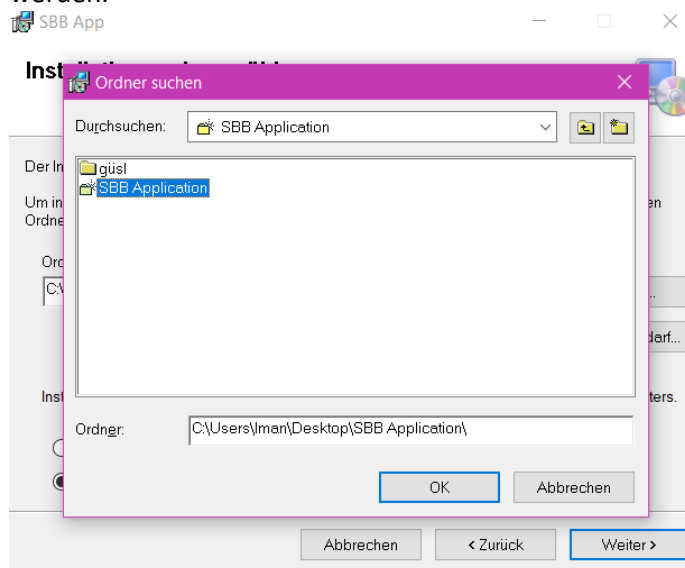
Man klickt sich weiter bis man den Installationspfad angeben muss.



Nun klickt man auf durchsuchen

11

und wählt den Desktop und erstellt einen neuen Ordner für alle Dateien die anschliessend installiert werden.









Hinweis: Die Applikation funktioniert nicht ohne die Dateien die ebenfalls installiert werden.

Nun bestätigt man die Destination mit «OK» und klickt sich weiter.

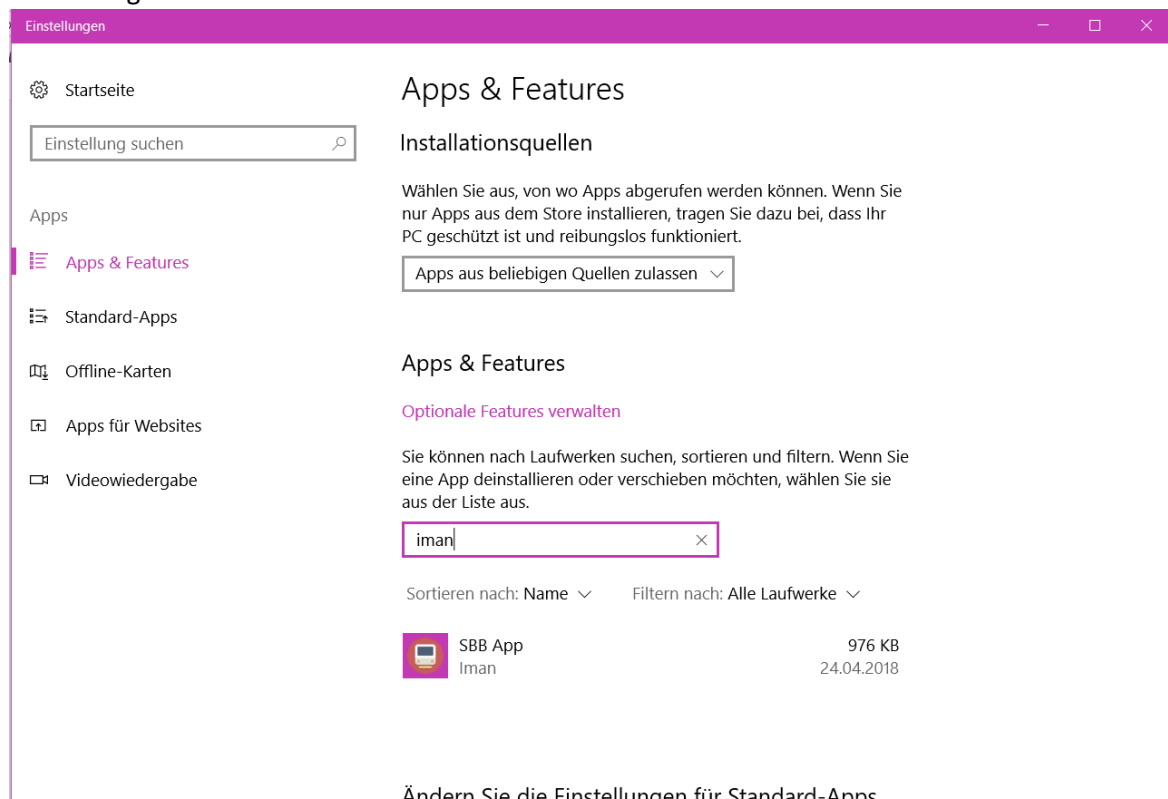
Die Meldung über die Änderungen von einem Unbekanntem herausgeben Akzeptieren.

Nun startet man die ImanSbbApp.exe

 ImanSbbApp.exe	24.04.2018 12:50	Anwendung	19 KB
 ImanSbbApp.exe.config	18.04.2018 08:39	XML Configuration...	1 KB
 Newtonsoft.Json.dll	09.01.2016 13:50	Anwendungserwei...	509 KB
 SwissTransport.dll	24.04.2018 12:50	Anwendungserwei...	12 KB
 System.Net.Http.dll	05.11.2015 19:36	Anwendungserwei...	84 KB
 transportation.ico	24.04.2018 11:57	ICO-Datei	362 KB



Deinstallation

Um die Anwendung zu deinstallieren sucht man entweder in den Programmen in den Windows Einstellungen danach.

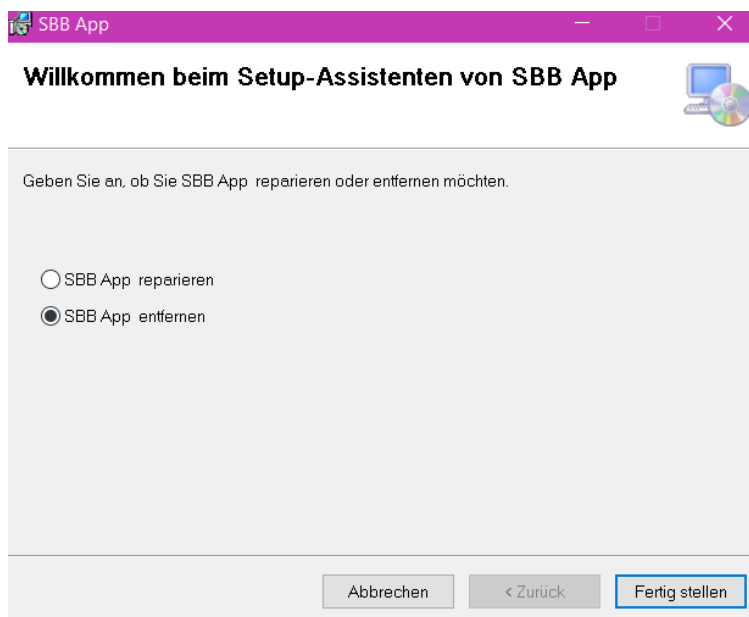


12

Oder man begibt sich wieder in den Installations Ordner und wählt das setup.exe oder das SBB App Setup.msi aus.

> Dokumente > GitHub > modul-318-student > SBB App Setup > Installation				▼	↺
Name	Änderungsdatum	Typ	Größe		
 SBB App Setup.msi	24.04.2018 12:50	Windows Installer...	1'119 KB		
 setup.exe	24.04.2018 12:50	Anwendung	752 KB		

Man wählt dann die Option SBB App entfernen und anschliessend klickt man auf fertigstellen. Auch



hier bestätigt man die Änderung von einem unbekannten Herausgeber.

Evaluierung

Ich habe mein Ziel mit dem viel lernen erreicht. Ich konnte WPF kennenlernen und habe es mir schwieriger vorgestellt als es tatsächlich ist. Ich habe zudem das MVVM Pattern kennengelernt. Dies konnte ich in meinem Projekt aber nur teilweise einsetzen da es sonst zu viel Zeit gebraucht hätte um alles zu lernen. Ich habe in meinem Projekt eine MainView erstellt und zwei Klassen verwendet.

Am Anfang des ÜK's hatte ich Probleme mit dem Mac zu arbeiten. Ich musste eine Windows VM erstellen und habe somit einen ganzen Tag verloren.

Am zweiten Tag habe ich mich anschliessend wieder mit C# eingearbeitet. Anfangs kam ich nicht so gut voran aber mit der Zeit war ich wieder voll drinnen.

Im Großen und Ganzen hat mir der ÜK sehr gefallen. Wir konnten selbständig am Projekt arbeiten und konnten fragen wann wir wollten.

Teilweise fand ich die Theorieinputs nicht besonders spannend da ich die ähnlichen Themen schon mehrmals während meiner Lehre gehört habe.