

Marco Rocchinotti

Bénédict  [Firmenadresse]

Swiss Transport

Inhalt

[UseCase 2](#_Toc533151816)

[Mockup 3](#_Toc533151817)

[Aktivitätsdiagram 5](#_Toc533151818)

[Programmierrichtlinien 6](#_Toc533151819)

[Umgesetzte Anforderungen 7](#_Toc533151820)

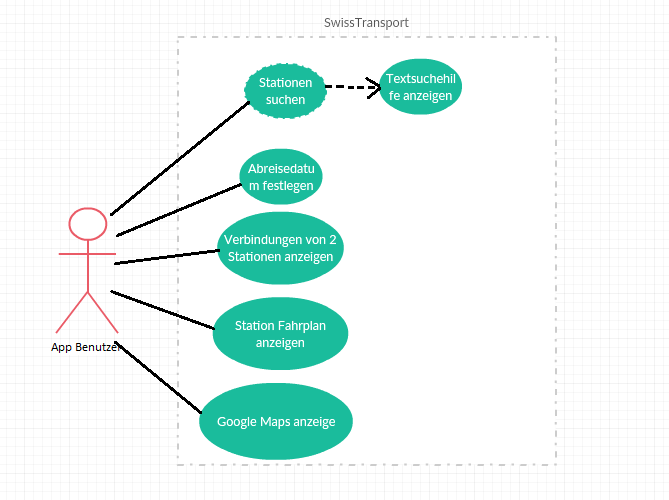
[Bekannte Fehler 8](#_Toc533151821)

[Installationsanleitung 8](#_Toc533151822)

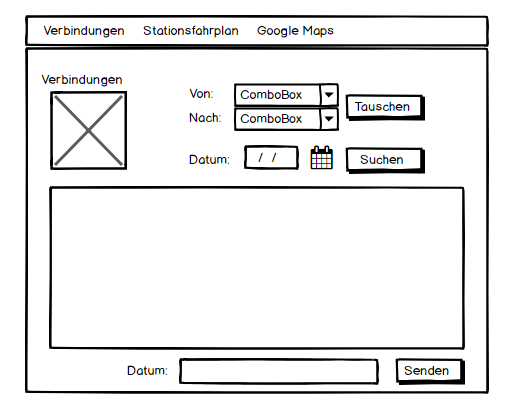
[Deinstallationsanleitung 12](#_Toc533151823)

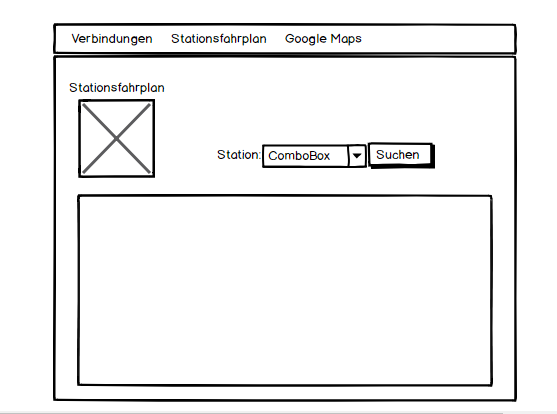
# UseCase

Basis der Funktionen:

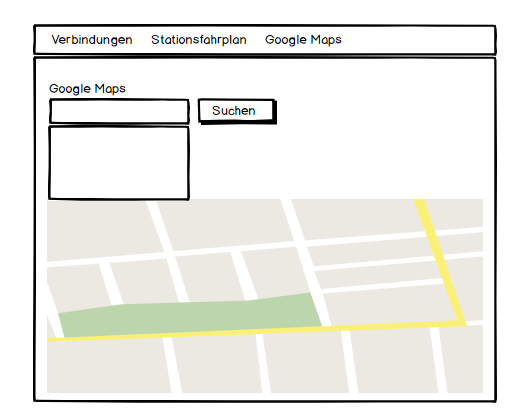


# Mockup

Verbindungen: 

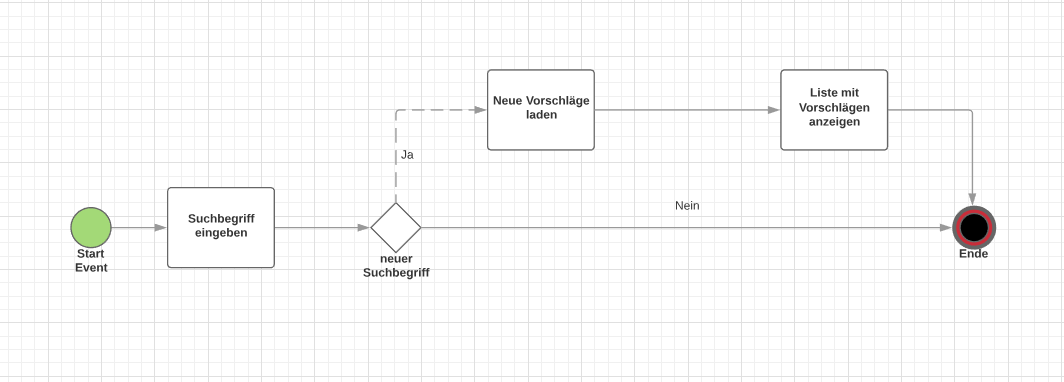
Stationsfahrplan:

Google Maps:

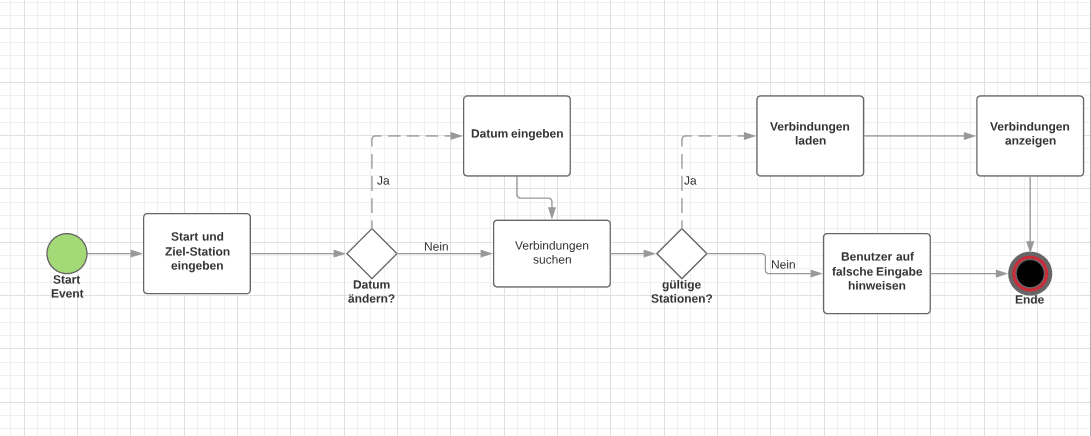


# Aktivitätsdiagram

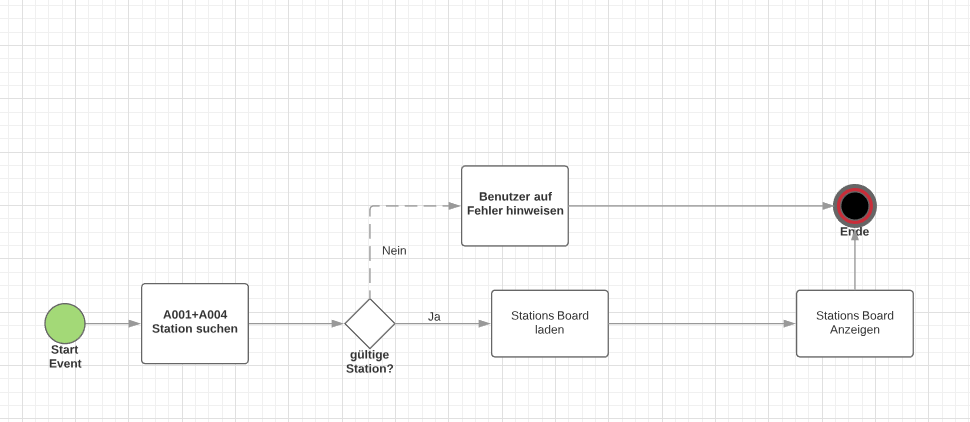
**A001+A004:**



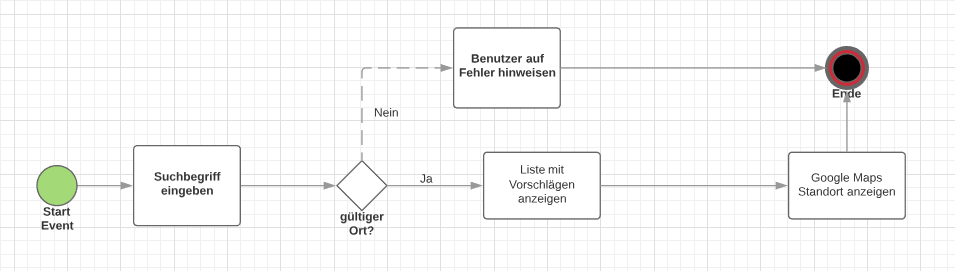
**A002+A005:**



**A003:**



**A006:**



# Programmierrichtlinien

**Naming Conventions**

Lokale variablen beginnen mit einem Kleinbuchstaben, globale Variablen mit (\_).

Klassen und Methoden beginnen gross.

Falls ein Name aus mehr als 2 Wörtern besteht wird jedes neue Wort grossgeschrieben.

GUI Elemente beginnen mit einem Kürzel zB. cb für ComboBox oder lb für ListBox.

Namen der Methoden müssen bereits die Aktion, welche sie ausführt beschreiben.

**Deklaration**

Variablen dürfen leer deklariert werden, müssen jedoch einen im Kontext eindeutigen Namen besitzen.

Methoden dürfen lokale Variablen besitzen.

**Comments**

Jede public aufrufbare, nicht generierte Methode benötigt vorgehend einen Kommentar welcher den Sinn der Methode nochmals zusammenfasst. In der Methode fassen Kommentare einzelne Aktionen zusammen falls dies benötigt wird.

**Statements**

Falls ein Statement ausser Switch nicht bereits durch die Namensgebung der Variable logisch ist zB. if(nameVorhanden==true) wird vorgehend Kommentiert welche Bedingungen erfüllt erden werden müssen. Bei switch-Statements muss bei unklaren Optionen einen Kommentar vorgehen welcher die Aktion beschreibt oder ein allgemeiner Kommentar welcher das verhalten aller Optionen beschreibt. Falls die Anweisung nach einem Statement nur eine Zeile benötigt, werden die geschweiften Klammern weggelassen. Statements werden mit dem C#

# Umgesetzte Anforderungen

A001 + A004:

Die Textsuche schlägt automatisch, nachdem sich der Text 500ms nicht verändert hat, alle gefundenen Treffer vor.

A002:

Nach dem betätigen des Suchbuttons werden die 4 nächsten Verbindungen angezeigt.

A003:

Durch drücken des Suchbuttons wird das Stationsboard angezeigt.

A005:

Durch das ändern der Datums und Zeitanzeige wird die Suche 4 Verbindungen ab dem angegebenen Datum ausgeben.

Switch Button:

Ein Button welcher den Zielort mit dem Startort tauscht, falls man eine Verbindung in die andere Richtung sucht.

A006:

Man kann den Ort auf dem Google Maps Tab suchen und anschauen.

A008:

Funktioniert teilweise ab und zu wird die Liste mit einem Code ersetzt.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Anforderung: | Beschreibung: | Anleitung: | Erwartetes Ergebnis: | Bugs: | Erfüllt: |
| A001+A004 | Automatische Textsuchhilfe | 1. In eine ComboBox des Programmes Klicken.  2. Standort eingeben  3. warten | Eine Liste von mindestens 10 Stationen welche Luzern enthalten. | Bei Schreibfehler gibt’s ein Error |  |
| A002+A005 | Verbindungen von 2 Stationen ab einem Datum Anzeigen. | 1. In der ComboBox neben dem Label Von: Wunsch Ort eingeben  2. Enter drücken  3. Tab drücken  4. Ziel Ort eingeben  5. Mit der Pfeiltaste nach unten Ziel Ort auswählen  6. Enter drücken  7. Datum setzen  8. Zeit setzen  9 Suchen Klicken | Verbindungen von Wunsch Ort nach Ziel Ort anzeigen. | Bei Schreibfehler gibt’s ein Error |  |
| A003 | Das Stationsbrett einer Station anzeigen. | 1. ComboBox neben dem Label Station anklicken  2. Wunsch Ort eingeben  3. Enter drücken  4. Suchen drücken | Das aktuelle Board von Wunsch Ort, startend ab der momentanen Zeit | Bei Schreibfehler gibt’s ein Error |  |
| A006: | Station auf Google Maps anzeigen | 1.Textbox unter dem Label Google Maps anklicken.  2.Wunschort eingeben.  3.Suchen drücken | Die Station wird auf Google Maps angezeigt. | Es gibt eine Fehler Meldung die man einfach weg klicken kann. |  |
| A008 | Der Fahrplan wird per Mail geschickt. | 1.Email in der Textbox neben Email: eingeben  2.Senden klicken | Der Fahrplan wird per mail geschickt | Es kann sein das die Liste nicht geschickt wird. |  |

# Bekannte Fehler

Automatische Suchhilfe:

Die automatische Vervollständigung von Texteingaben der Methode SearchStationsHelper() der Klasse GUIFunctions führt durch einen unbekannten Fehler dazu, dass der Mauszeiger verschwindet, wenn die automatischen Vorschläge angezeigt werden, bis ein Item der Liste ausgewählt wird.

Wenn ein unbekannter Ort eingegeben wird oder bei einem Schreibfehler wird ein Error angezeigt den kann man einfach weg klicken.

Das Senden von Mails geht aber die Liste wird nicht geschickt.

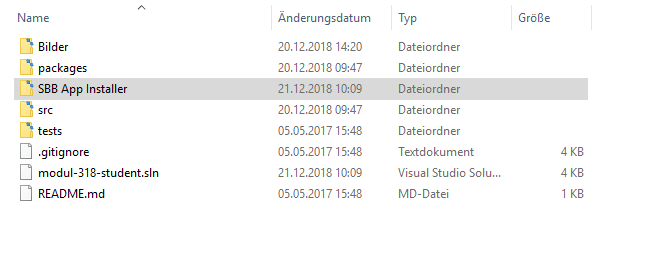
Wenn man auf dem Google Maps Tab ist kommt eine Fehler Meldung.

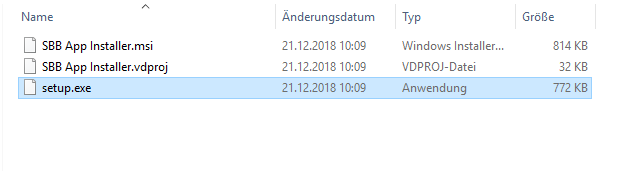
# Installationsanleitung

Als erstes laden Sie sich die Installationsdaten herunter indem Sie auf folgenden Link Klicken:

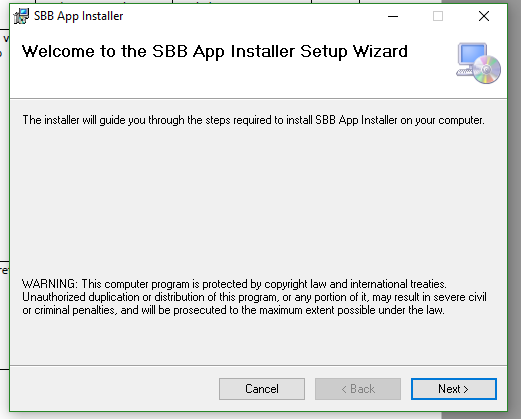
<https://github.com/ggthis272/modul-318-student>

Der Ortner heisst SBB App Installer

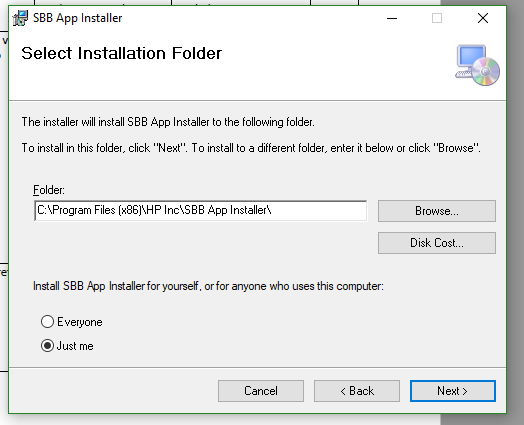


Als nächstes führen sie die Datei setup.exe falls nötig mit administrativen Rechten aus. 

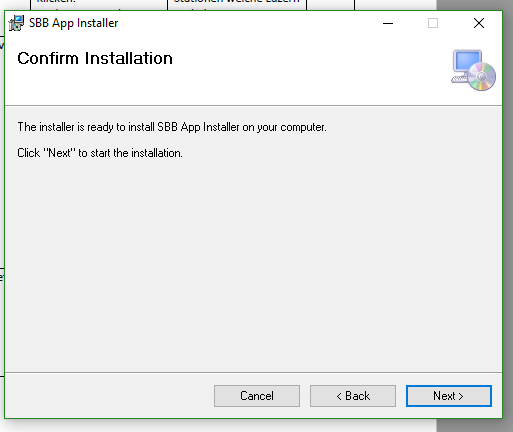
Drücken sie «Next >» um mit der Installation anzufangen.



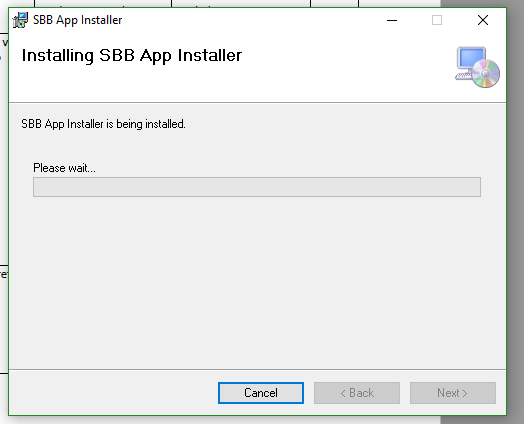
Wählen sie einen Installation Ordner aus wenn sie wollen, es wird immer ein Shortcut auf dem Desktop erstellt.



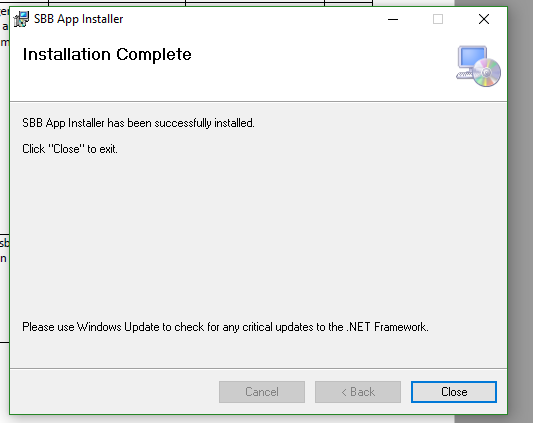
Drücken sie Next um mit der Installation fortzufahren.



Warten bis die Installation abgeschlossen ist.



Drücken sie im nun auf Close um das Setup zu verlassen. Die heruntergeladene Setupdatei benötigen Sie nichtmehr.



Auf dem Desktop wurde Automatisch eine .exe mit dem Namen SBB App erstellt die kann mit einem Doppelklick geöffnet werden.

# Deinstallationsanleitung

Um das Programm deinstallieren zu können öffnen sie wieder die Installation Datei setup.exe

Es öffnet sich ein Fenster mit zwei Optionen wählen sie Remove SBB App Installer aus und klicken sie auf Finish.

