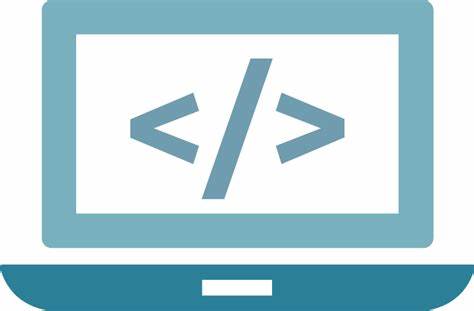
Modul 318

Analysieren und objektbasiert programmieren mit Komponenten

**Projekt Dokumentation**



**Von Manuel Marbacher**

Inf19aL

Dezember 2020

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc59110927)

[1.1 Zweck der Doku 3](#_Toc59110928)

[1.2 Zu dem Projekt 3](#_Toc59110929)

[2 Design und Analyse 3](#_Toc59110930)

[2.1 Mockup 3](#_Toc59110931)

[2.2 User Story 5](#_Toc59110932)

[2.3 Ablaufdiagramm 7](#_Toc59110933)

[3 Testing 8](#_Toc59110934)

[3.1 Testfälle 9](#_Toc59110935)

[4 Anleitung 11](#_Toc59110936)

[4.1 Installation 11](#_Toc59110937)

[4.2 Deinstallation 12](#_Toc59110938)

[5 Abschluss 12](#_Toc59110939)

# Einleitung

Im ÜK 318 war es die Aufgabe ein Projekt zu erstellen und dieses zu dokumentieren. In dieser Dokumentation finden sie mein Vorgehen und eine Anleitung zu der Installation und dem Programm selbst.

## Zweck der Doku

In dieser Doku finden sie die Anforderungen des Projektes und meine Vorgehensweise wären dem Projekt (vom ersten Mockup bis zum fertigen Programm mit allen Features).

## Zu dem Projekt

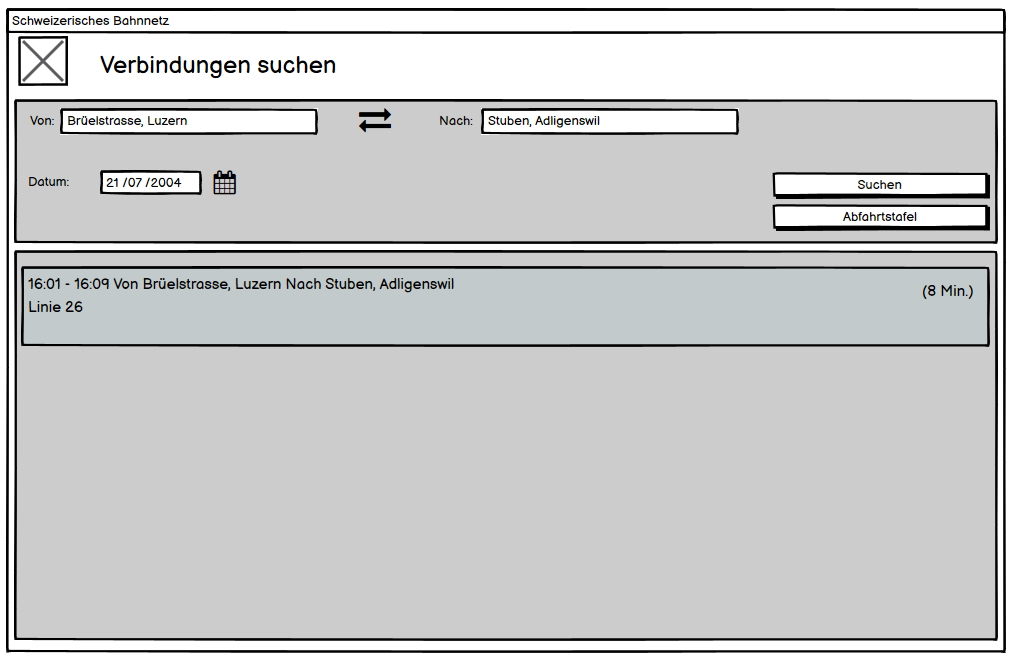
Das Ziel war es eine Applikation für den ÖV zu erstellen. Mit dieser Applikation kann man Verbindungen suchen (auch zukünftige Verbindungen) und man hat eine Abfahrtstaffel, welche alle Verbindungen von einer beliebigen Station aus geben.

# Design und Analyse

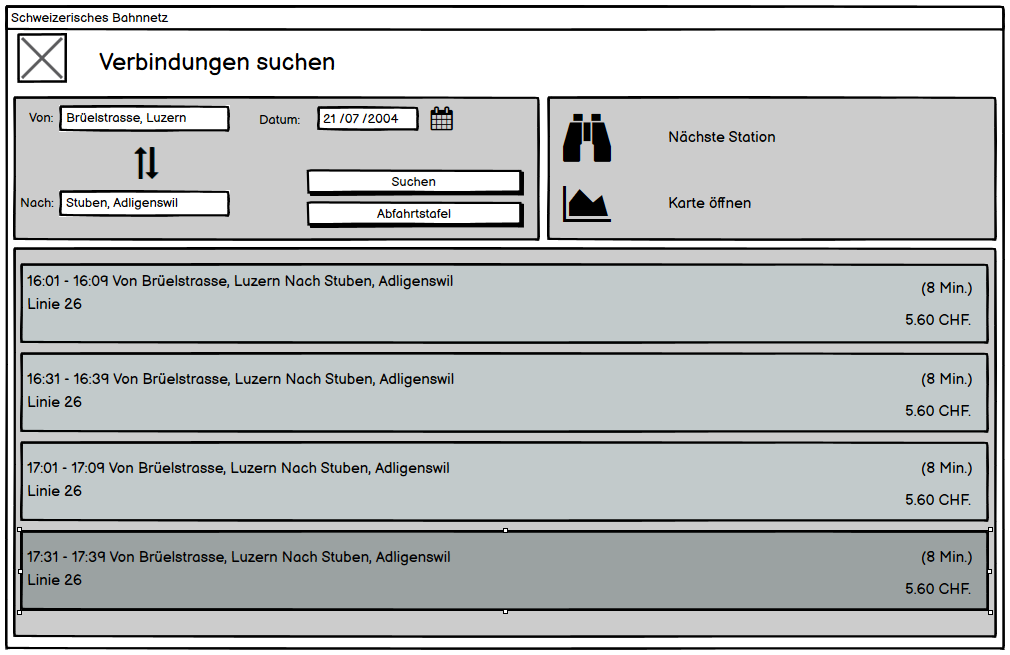
## Mockup

Das Mockup ist ein grobes design wie die die Ausgabe später aussehen sollte. Ich habe dies in 2 Stufen gemacht

Stufe 1: Alle Aufgaben der Priorität 1 und 2



Stufe 2: Alle Aufgaben der Prioritäten 1 – 3



Wenn man das Mockup nun mit dem Resultat vergleicht findet man einige Kleinigkeiten, welche sich verschoben haben oder welche es nicht mehr gibt. Dies liegt einerseits daran, dass man diese Elemente in der Toolbox nicht findet andererseits habe ich auch Elemente verschoben für die User Freundlichkeit. Trotz den vielen kleinen Änderungen konnte das Mockup aufs grosse und ganze umgesetzt werden.

## User Story

Die User Storys sind in 3 Teile aufgeteilt:

1. Anforderungen des Kunden
2. Die Akzeptanz (wann diese User Story erfüllt ist)
3. Die Priorität

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Verbindungen suchen** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 001 | Als User möchte ich den Fahrplan zu meinem nächsten Aussendienst ansehen können, um zu meinen Terminen zu gelangen | * Keine leere Eingabe möglich * Bei richtiger Eingabe Anzeige von Verbindungen (die Funktion wird in der User Story „[Mehrere](file:///C:\Users\manma\Downloads\Dokumentation_TransportApp%20(1).docx#_Nächsten_vier_Verbindungen) Verbindungen anzeigen“ behandelt) * Falls keine Verbindungen vorhanden sind wird eine Fehlermeldung ausgegeben | High | Done |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mehrere Verbindungen anzeigen** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 002 | Als User möchte ich mindestens 4 Verbindungen sehen, um verschiedene Möglichkeiten zu haben. | * Falls keine Verbindung vorhanden wird eine Fehlermeldung ausgegeben * Falls Verbindung vorhanden werden mindestens 4 Verbindungen ausgegeben * Die Ausgabe Enthält Start-/Endstation, Plattform Nummer, Abfahrt-/Ankunftszeit und die fahrt zeit | High | Done |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Abfahrtstafel** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 003 | Als User möchte ich alle möglichen Verbindungen von der aktuellen Station sehen, damit ich weiss welche Verbindungen es gibt. | * Bei Eingabe werden mindestens die nächsten 4 Verbindungen dieser Station ausgegeben falls diese vorhanden sind * Die Ausgabe Enthält Start-/Endstation, Zug Name und die Uhrzeit | High | Done |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Zukünftige Verbindungen** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 004 | Als User möchte ich die Zeiten verschieben können, um im Voraus planen zu können. | * Eingabe von Datum und Zeit ist einfach möglich * Es werden mindestens 4 Verbindungen nach dem angegebenen Datum angezeigt | Medium | Done |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Suchvorschläge** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 005 | Als User möchte ich das die Busstationen vorgeschlagen werden, damit ich mir nicht alle Station Namen merken muss. | * Beim Klicken auf einen Vorschlag wird das Feld automatisch gefüllt * Die Vorschläge werden dynamisch angepasst * Die Vorschläge behindern den Nutzer nicht beim Schreiben * Ist das Feld leer gibt es keine Vorschläge | Medium | Done |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Abfahrts-/Zielort switch** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 006 | Als User möchte ich abfahrts-/Zielort wechseln können damit ich direkt den Rückweg anschauen kann. | * Bei einem Klick auf den switch Knopf werden die Texte von Abfahrtsort und Ankunft Ort getauscht. * Wenn man danach auf Suchen Klickt funktioniert alles wie davor | Medium | Done |

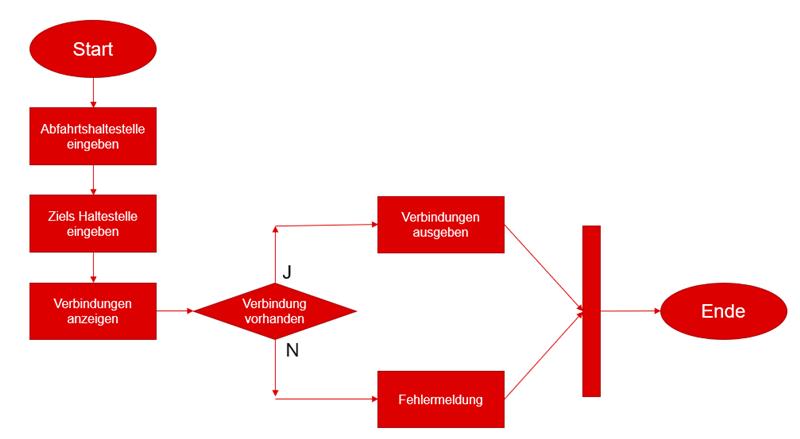
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nächste Station anzeigen** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 007 | Als User sollen mir die nächsten Stationen von meiner momentanen Lage angezeigt werden. Es dient dazu, schnell eine nahe Station von überall in der Schweiz zu finden. | * Falls GPS nicht vorhanden ist, gibt es eine Fehlermeldung aus * Der aktuelle Standort und die nächsten 4 Stationen werden angezeigt. | Low | To Do |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stationen auf Karte Anzeigen** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 008 | Als User möchte eine Station auf einer Karte suchen können. Somit kann ich auf einer Karte meine Route einfach planen. | * Falls die Station nicht existiert wird eine Fehlermeldung ausgegeben * Die Adresse einer Station wird mithilfe einer Karte dargestellt * Beim Starten der Applikation wird noch keine Karte angezeigt | Low | To Do |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ticketpreise angeben** | | | | |
| **ID** | **Beschreibung** | **Akzeptanzkriterien** | **Priorität** | **Status** |
| 009 | Als User möchte ich die Ticketpreise bereits sehen, damit ich weiss wie viel Geld ich mitnehmen muss. | * Wenn man nach einer Verbindung sucht werden in der Tabelle die jeweiligen Preise ausgegeben * Falls es mit einem Tagesticket günstiger ist wird das berücksichtigt | Low | To Do |

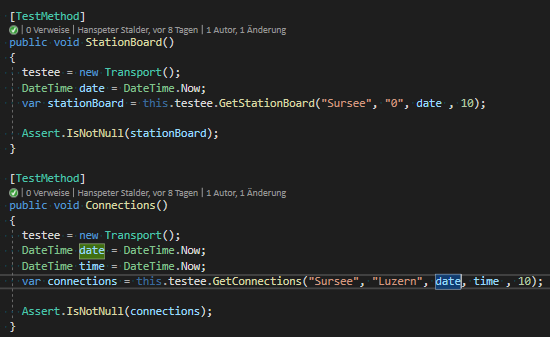
## Ablaufdiagramm

Dies ist ein Beispiel Ablauf der User Story «Verbindungen suchen»

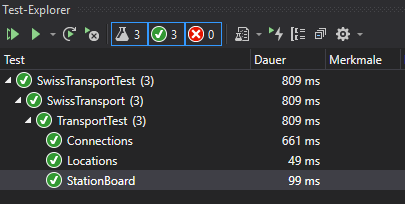
****

# Testing

Das Testing ist ein wichtiger Prozess bei einem Projekt damit die Applikation auch möglichst fehlerfrei läuft. Die ersten zwei Tests sind bereits vordefiniert. Diese sehen wie folgt aus.



Ob die Tests nun fehlerfrei liefen sieht man an den grünen Haken. Falls ein Test ein rotes Kreuz hat wäre dieser schief gelaufen.



## Testfälle

Zusätzlich zu dem vordefinierten Test habe ich zu jeder User-Story mit der Priorität 1 einen Testfall geschrieben.

Testfall 1: Verbindungen suchen, switch Button und Auto Completion

Start Zustand: Programm ist noch nicht gestartet und Internet ist vorhanden

|  |  |  |  |
| --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Erfüllt |
| 1 | Programm starten | Programm öffnet sich mit dem Titel «Verbindungen suchen». Felder für das Suchen und Anzeigen von Verbindungen werden angezeigt |  |
| 2 | Auf das Feld Von klicken und «Zug» eingeben | Im runter geklappten Feld werden Vorschläge angezeigt (Zug; Zug, Postplatz; Zug, Chollemüli; etc.) |  |
| 3 | «Zug, Postplatz» anklicken | Das Vorschlagsfenster wird geschlossen und im Feld Abfahrtsort steht «Zug, Postplatz» |  |
| 4 | Auf das Feld Nach klicken und «Luzern» eingeben | Im runter geklappten Feld werden Vorschläge angezeigt (Luzern; Luzern, Bahnhof; Luzern, Kantonalbank etc.) |  |
| 5 | «Luzern, Bahnhof» anklicken und Tabulator-Taste drücken | Fokus geht auf Datum, das Vorschlagsfenster wird geschlossen und im Feld Nach steht «Luzern, Bahnhof» |  |
| 6 | Datum 23.12.2020 wählen Tabulator-Taste drücken und Zeit auf 14:30 setzen | Im Feld Datum steht nun das gewollte Datum und im Feld Zeit die gewollte Zeit |  |
| 7 | Auf Button mit Pfeilen klicken | Im Feld Abfahrtsort steht neu «Luzern, Bahnhof» und im Feld Ankunftsort «Zug, Postplatz» |  |
| 9 | Button «suchen» klicken | 10 Verbindungen von «Luzern, Bahnhof» nach «Zug, Postplatz» mit Abfahrtsdatum und Uhrzeit nach 23.12.2020 14:30 werden angezeigt |  |

Tester: Manuel Marbacher

Testfall 2: Abfahrtstafel anzeigen

Start Zustand: Programm ist gestartet auf der Seite Verbindungen suchen

|  |  |  |  |
| --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Erfüllt |
| 1 | Auf «Abfahrtstafel» klicken | Der Titel wird zu «Abfahrtstafel» geändert. Die Felder nach, Datum und Zeit sind nicht mehr aufrufbar. |  |
| 2 | Auf das Feld «Von» klicken und «Littau, Gasshof» eingeben und suchen Klicken | In der Tabelle werden 10 mögliche Zielorte ausgegeben mit dem dazugehörigen Fahrzeugnamen und der passenden Abfahrtszeit |  |

Testfall 3: Eingabe von nicht vorhandenen Ortschaften

Start Zustand: Programm ist noch nicht gestartet und Internet ist vorhanden

|  |  |  |  |
| --- | --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat | Erfüllt |
| 1 | Programm starten | Programm öffnet sich mit dem Titel «Verbindungen suchen». Felder für das Suchen und Anzeigen von Verbindungen werden angezeigt |  |
| 2 | Auf das Feld Abfahrtsort klicken und «Test» eingeben | Es werden keine Vorschläge ausgegeben. Bei den Vorschlägen steht lediglich «Keine Treffer» |  |
| 3 | Keine Treffer wählen und in das Feld Ankunftsort «Luzern» eingeben | Es Wird «Luzern» vorgeschlagen |  |
| 4 | «Luzern» anwählen und «Verbindungen suchen» klicken | Es wird ein Fehler mit «Es wurden keine Verbindungen gefunden» angezeigt |  |

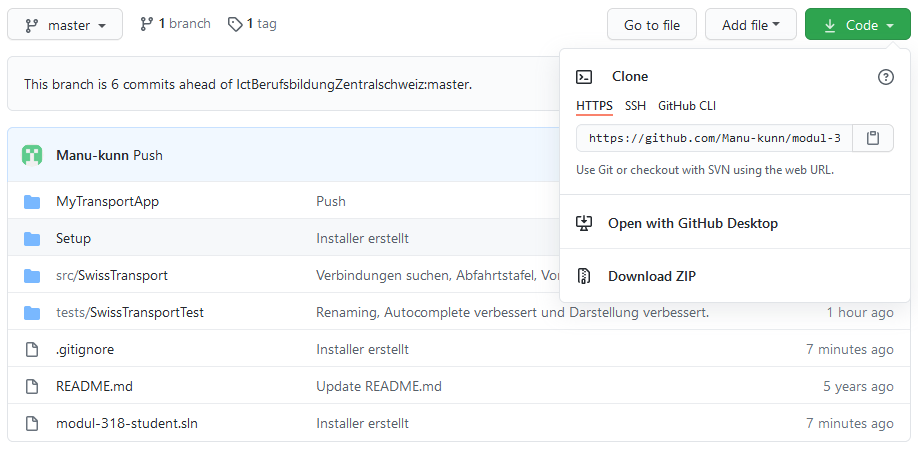
Tester: Manuel Marbacher

# Anleitung

In dieser Anleitung erkläre ich wie man die Applikation installiert, diese anwendet und danach wieder deinstalliert.

## Installation

Zu Beginn man die Zip-Datei von github.com herunterladen. Um dies zu tuen muss man folgenden Link öffnen: <https://github.com/Manu-kunn/modul-318-student>

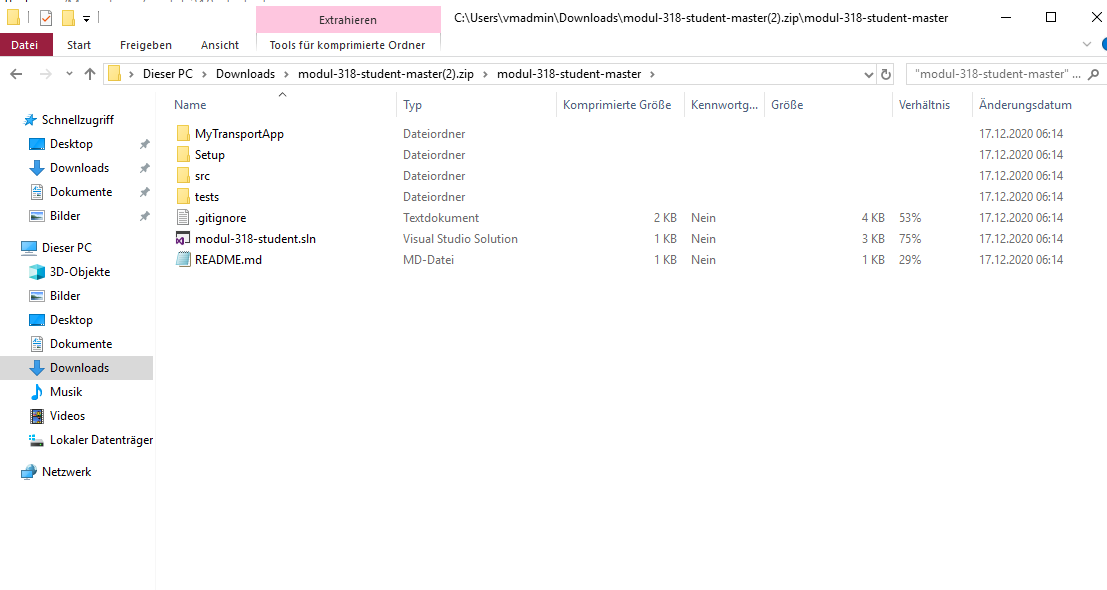


**2..**

**1.**

Nun kann man oben rechts das Projekt als Zip-Datei herunterladen.

Danach muss man den heruntergeladenen Zip-Ordner entpacken dies sieht dann wie folgt aus

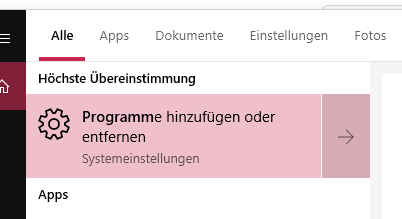


Nun geht man auf «modul-318-student-master\Setup\Release\setup.exe» und öffnet dies.

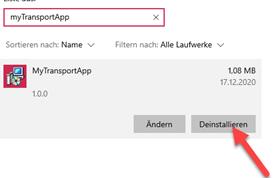
Falls beim Öffnen eine Warnung zur Computersicherheit kommt kann man diese ignorieren.

## Deinstallation

Für die Deinstallation öffnet man zuerst die App «Programme hinzufügen oder entfernen»



Danach sucht man nach der «myTransportApp» und deinstalliert diese.



# Abschluss

Bei diesem Projekt habe ich viel Neues gelernt. Nun weiss ich wie ich ein Projekt plane (Userstory, Mockup), wie man externe Daten in eine Applikation einbindet und welche vor- und Nachteile dies hat. Dazu konnte ich einige Erfahrungen sammeln, welche ich bei weiteren Projekten sehr gut nutzen kann.

Adligenswil, 17.21.2020



Manuel Marbacher manuel\_marbacher@sluz.ch