17.12.2020

Projekt 318

Luca Rickenbach

Inhalt

[1. Einleitung 2](#_Toc58918540)

[2. Zweck 2](#_Toc58918541)

[3. UserStory 2](#_Toc58918542)

[1.1 Stationssuche - Story01 2](#_Toc58918543)

[1.2 Verbindungen anzeigen - Story02 2](#_Toc58918544)

[1.3 Abfahrtstafel - Story03 3](#_Toc58918545)

[1.4 Vorschlag während dem eingeben von Ortschaft - Story04 3](#_Toc58918546)

[1.5 Datum und Zeit Eingabe, um Verbindungen in der Zukunft zu finden - Story05 3](#_Toc58918547)

[1.6 Station auf der Karte anzeigen - Story06 3](#_Toc58918548)

[1.7 Nächste Station anzeigen lassen - Story07 3](#_Toc58918549)

[1.8 Die Verbindung via Mail versenden - Story 08 4](#_Toc58918550)

[1.9 Eigene User Story -> Auswahl Ankunft/Abfahrt Story09 4](#_Toc58918551)

[4. Aktivitätsdiagramm zu Story01 4](#_Toc58918552)

[5. Mockup 5](#_Toc58918553)

[6. Testfälle 6](#_Toc58918554)

[7. Testprotokoll 6](#_Toc58918555)

# Einleitung

In diesem Modul lernen wir das Objektorientierte Programmieren kennen. Die meisten von uns hatten schon Erfahrungen mit diesem Thema. Jedoch haben wir auch noch neues gelernt, wie z.B. Testing mit Unit-Tests.

Diese Dokumentation beschreibt ein Projekt, welches wir im ÜK-318 gemacht haben. Dieses Projekt müssen wir am Ende des ÜK dem Instruktor abgeben. Er wird das Projekt und diese Dokumentation bewerten uns eine Modulnote dafür geben. Das fertige Projekt sollte etwas Ähnliches wie eine SBB App werden. Der Punkt drei, die UserStorys, sind Anforderungen oder Wünsche, die uns der Kunde mitgeteilt hat. Mit diesen Anforderungen müssen wir das Projekt individuell gestalten.

# Zweck

Dieses Projekt soll unsere Fähigkeiten im Objektorientierten Programmieren testen. Ebenfalls, soll uns diese Dokumentation lernen, wie wir am besten vorgehen können. Wir benutzten eine gute App für die Vorbereitung des GUI, dass wir erstellen müssen. Beim Punkt fünf sieht man ein Mockup. Ein Mockup ist eine Grafik, welche meine Vorstellungen, des GUI aufzeigt. So können wir herausfinden, wie es uns am besten passen würde. Der Haupt-Zweck dieses Dokuments ist, dass wir lernen wie eine gute Dokumentation/Vorgehen aussieht. Da das Dokumentieren eine grosse Rolle in unserem Beruf spielt.

# UserStory

## 3.1 Stationssuche - Story01

Als User möchte ich herausfinden ob beim eingegebenen Standort, eine Station vorhanden ist, um sicher zu gehen, dass es den Standort oder Station gibt.

**Abnahmekriterien**

* Es muss ein Textfeld erhältlich sein, damit der User die Ortschaft eingeben kann.

**Priorität: 1**

**Status: Fertig**

## 3.2 Verbindungen anzeigen - Story02

Als User möchte ich eine Start-Station und eine End-Station eingeben können, um mir sicher die nächsten 4 Verbindungen anzeigen zu lassen.

**Abnahmekriterien**

* Es muss eine Tabelle erhältlich sein, in der alle Verbindungen angezeigt werden.
* Es müssen mindestens die nächsten 4 Verbindungen angezeigt werden.
* In diese Tabelle, muss das Resultat der Stationssuche – Story01 ausgegeben werden

**Priorität: 1**

**Status: Fertig**

## 3.3 Abfahrtstafel - Story03

Als User möchte ich alle Verbindungen ab der eingegebenen Station angezeigt bekommen, um alle Informationen über die Fahrt auf einen Blick zu sehen.

**Abnahmekriterien**

* Es muss eine Tabelle erhältlich sein, welche
* Die End-Station der Fahrten muss nicht angezeigt werden.

**Priorität: 1**

**Status: Fertig**

## 3.4 Vorschlag während dem eingeben von Ortschaft - Story04

Als User möchte ich beim Eingeben einer Station Vorschläge erhalten, um es mir leichter zu machen, den Namen einer Station zu finden.

**Abnahmekriterien**

* Dafür verwenden wir die Textfelder in welcher wir die Ortschaften eingeben

**Priorität: 2**

**Status: Fertig**

## 3.5 Datum und Zeit Eingabe, um Verbindungen in der Zukunft zu finden - Story05

Als User möchte ich eine Zeit und ein Datum eingeben können, um Verbindungen ab diesem Zeitpunkt zu finden.

**Abnahmekriterien**

* Eine Box, in der das Datum und die Zeit eingegeben werden kann.

**Priorität: 2**

**Status: Fertig**

## 3.6 Station auf der Karte anzeigen - Story06

Als User möchte ich die Angegebenen Stationen/Ortschaften auf einer Karte angezeigt bekommen, um zu sehen wo die Stationen/Orte liegen.

**Abnahmekriterien**

* Ein Knopf der eine Karte öffnet, auf welcher der Standort der Station markiert ist.

**Priorität: 3**

## 3.7 Nächste Station anzeigen lassen - Story07

Als User möchte ich die naheliegendste Station angezeigt bekommen, um zu sehen welche Station die nächste ist.

**Abnahmekriterien**

* Ein Knopf welcher entweder die Nächste Station einfach in ein Feld ausgibt, oder sogar auf der Karte anzeigt.

**Priorität: 3**

## 3.8 Die Verbindung via Mail versenden - Story 08

Als User möchte ich die Verbindung via Mail versenden können, um sie mit meinen Arbeitskollegen zu teilen.

**Akzeptanzkriterien**

* Ich kann jede E-Mail-Adresse eingeben.
* Ich kann eine beliebige Verbindung versenden.
* Es können mehrere Verbindungen auf einmal geteilt werden.

**Priorität: 3**

**Status: In Bearbeitung**

## 3.9 Eigene User Story -> Auswahl Ankunft/Abfahrt Story09

Als User möchte ich auswählen können, ob das Verkehrsmittel um die definierte Zeit ankommen oder abfahren sollte. Um die Suche, nach der besten Verbindung, zu vereinfachen.

**Akzeptanzkriterien**

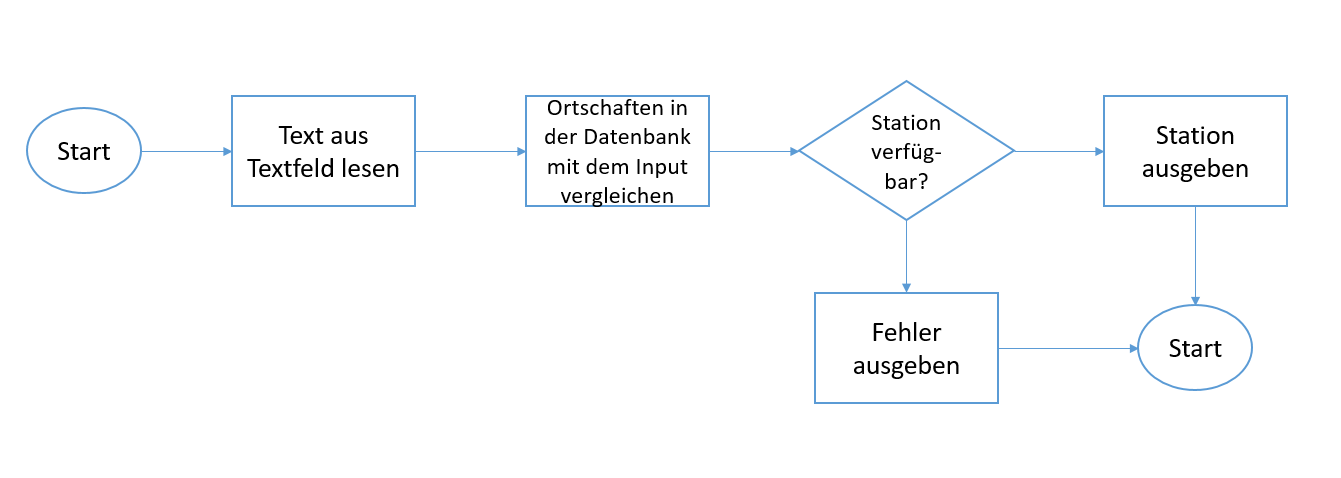
* Radiobuttons wo ich zwischen Ankunft und Abfahrt auswählen kann.

**Priorität: 3**

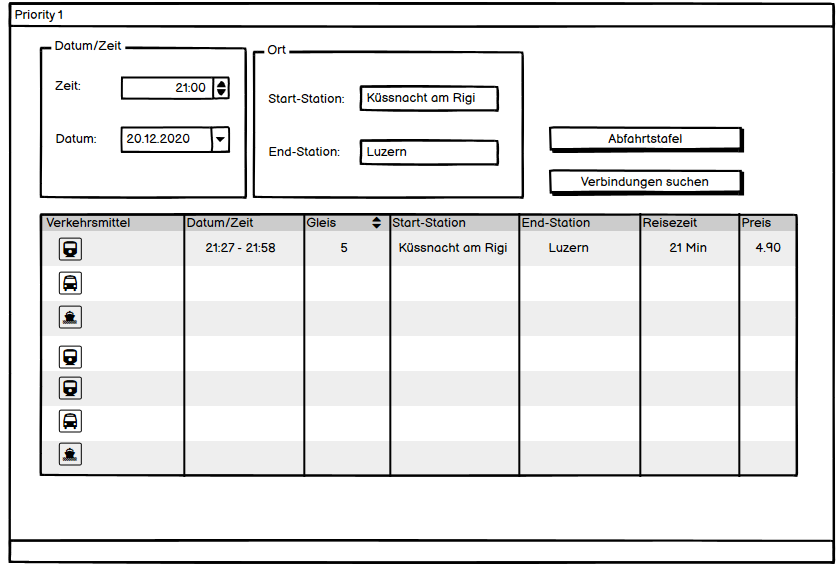
**Status: In Bearbeitung**

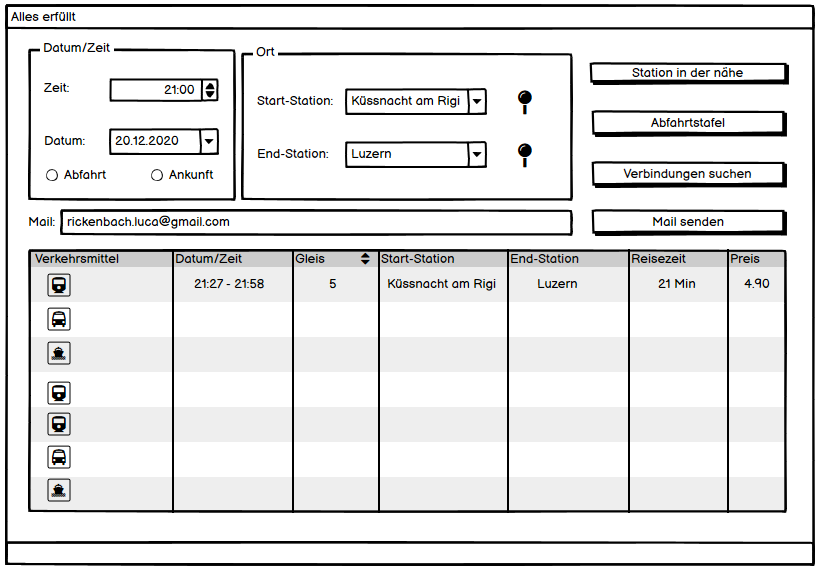
# Aktivitätsdiagramm zu Story01

Das Aktivitätsdiagramm zeigt kurz auf wie das Programm funktionieren soll. Es dient als Hilfe, dass sich der Entwickler während dem Programmieren besser vorstellen kann, wie er vorgehen muss.



# Mockup

Priority 1:

Nice to Have: 

# Testfälle

## 6.1 Story02

Als ich das GUI fertiggestellt habe, habe ich mit der Story02 begonnen. Dafür habe ich den Code in «ITransport» und «Transport» angepasst und den benötigten Code für den Button geschrieben. Anschliessend habe ich es das erste Mal getestet. Es hat funktioniert. Jedoch habe ich fürs erste nur das Datum, die Start-Station und die End-Station ausgegeben. Die Zeit hat noch nicht funktioniert. Danach habe ich Kolonnen in der Tabelle hinzugefügt und im Code noch den Rest ergänzt. Danach gab es zusätzlich noch die Plattform, Ankunftszeit und Reisezeit aus. Es hat funktioniert!

## 6.2 Story03

Anschliessend habe ich mit der Story03 begonnen. Diese war ähnlich wie die Story02. Ich musste nur kleine Änderungen vornehmen und dann hat diese ebenfalls schon funktioniert. Das Programm gibt nun das Datum, die Start-Station, End-Station und der Name des Zuges aus. Dies hat ebenfalls funktioniert. Im Anschluss habe ich mich noch dafür entschieden, dies in einer separaten Form zu machen.

## 6.3 Story05

Anschliessend hatte ich noch Probleme mit der Zeiteingabe. Als ich es zum ersten Mal getestet haben, hat es das Datum wie immer übernommen, aber die Zeit war falsch. Ich habe gemerkt das ich die Zeit in einem string gespeichert hatte. Ich habe dies dann zu «DateTime» gewechselt und das «ToString()» mit «Value» ersetzt. Danach hat es bei der Verbindungen Suchen funktioniert. Jedoch nicht bei der Abfahrtstafel. Ich habe es mehrmals kontrolliert und verglichen, jedoch konnte ich keinen Fehler entdecken. Darum habe ich es so eingestellt, dass es einfach immer die jetzige Zeit nimmt.

# Testprotokoll

