M318 Projektarbeit

Florian Beqiraj, 02. Dezember 2020

# Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc57818736)

[Einleitung 3](#_Toc57818737)

[Vorbereitung 4](#_Toc57818738)

[Use-Cases 4](#_Toc57818739)

[User-Stories 5](#_Toc57818740)

[Kriterien 6](#_Toc57818741)

[Mockup 8](#_Toc57818742)

[Umsetzung 9](#_Toc57818743)

[Testing 11](#_Toc57818744)

# Einleitung

In diesem Dokument wird praktische Projektarbeit des Moduls 318 präsentiert. Der Sinn der Arbeit ist die Umsetzung eines realen Entwicklerauftragsszenario. Es waren Analysen der Anforderungen, Einschätzungen des Aufwandes und Entwicklungsarbeiten durchzuführen.

Konkret mussten wir für einen fiktiven Arbeitsgeber eine ÖV-Software entwickeln, die seinen Anforderungen gerecht ist. Dabei mussten wir von einer öffentlichen Web-API Verkehrsdaten beziehen und diese in unserer Applikation konsumieren.

Wir mussten die Anforderungen des Arbeitgebers in verschiedene User-Stories unterteilen und diese der Priorität nach umsetzen.

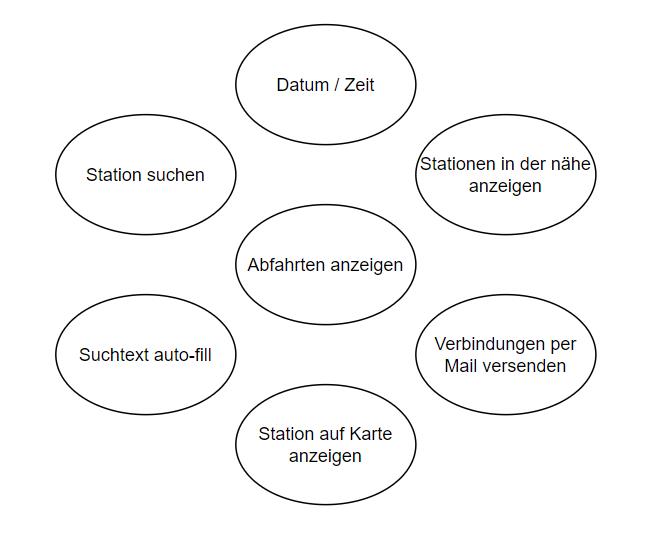
Wir mussten ein GUI erstellen, welches den heutigen Standarten der Userfreundlichkeit getreu war. Dazu haben wir zu Beginn des Kurses verschiedene Aufgaben gelöst.

# Vorbereitung

Wir bekamen den Auftrag in Form von einer fiktiven E-Mail des Arbeitgebers. Dieser ist unter folgender Adresse zu findend: <https://m318.ict-bz.ch/tag-2/projektanforderungen-analysieren>

## Use-Cases

Nach der Analyse der Nachricht ergaben sich folgende Use-Cases:



## User-Stories

Anhand der Use-Cases konnten wir folgende User-Stories definieren:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Wer | Was | Warum | Priorität | Umgesetzt |
| Aussendienst Mitarbeiter | Station suchen (Start und End) | Verbindung suchen | 1 | Ja |
| Abfahrt anzeigen | Beste Verbindung auswählen | 1 | Ja |
| Verbindungen suchen/anzeigen | Bei Unwissenheit beste Verbindung anzeigen oder Vorschlagen lassen | 1 | Ja |
| Datum / Zeit | Mögliche Veränderung Fahrplan (Sonntag) | 2 | Ja |
| Station in der Nähe anzeigen | Ort in der Nähe finden, für nächste mögliche Station | 3 | Nein |
| Suchtext auto-Vorschlag mit vervollständigen | Damit man nicht genau wissen muss wie die Station heisst, gibt es eine auto-vervollständigung | 2 | Ja |
| Station auf Karte anzeigen | Wegen Unwissenheit, um herauszufinden wo die nächste Station ist oder die Endstation | 3 | Nein |
| Verbindung per Mail versenden | Falls notwendig per Mail versendbar um den Kunden zu informieren | 3 | Nein |
| Ankunft / Abfahrt bei Zeit aus wählbar | Um selber zu entscheiden, ob man um diese Zeit ankommen möchte oder abfahren möchte | 4 | Ja |

## Kriterien

Anhand der User-Stories ergaben sich folgende Kriterien:

**Station suchen (Start und End)**

Als Aussendienst Mitarbeiter möchte ich eine Station nach Start und End Station suchen können, weil ich mir die Verbindungen nicht merken kann.

* Folgende Angaben können vom Benutzer eingegeben werden:
  + Startstation
  + Endstation

**Abfahrt anzeigen:**

Als Aussendienst Mitarbeiter möchte ich die Abfahrt anzeigen lassen, um daraus die bestmöglichen Verbindungen oder Vorschläge anzeigen zu lassen.

* Bei der Eingabe von Start- und Endstation soll die bestmögliche Abfahrt angezeigt werden.

**Verbindungen suchen/anzeigen:**

Als Aussendienst Mitarbeiter möchte ich Verbindungen suchen und anzeigen lassen, um nicht alle Details wissen zu müssen und somit eine Auswahl von Möglichkeiten zu finden.

* Bei der alleinigen Eingabe des Startstation sollen alle möglichen Verbindungen angezeigt werden.

**Datum/Uhrzeit:**

Als Aussendienst Mitarbeiter möchte ich das Datum und die Zeit eingeben können, um exakte Angaben über meinen nächsten Verbindungen zu erhalten, falls ein Feiertag oder Sonntag an diesem Tag ist und sich der Fahrplan veränderte.

* Das genaue Datum und die genaue Uhrzeit einer Verbindung sollen dem Benutzer angezeigt werden.
* Eine Uhrzeit soll vom Benutzer eingegeben werden können. Anhand dieser Eingabe sollen passende Verbindungen angezeigt werden.

**Station in der Nähe anzeigen:**

Als Aussendienst Mitarbeiter möchte ich, dass die Station, die sich in meiner Nähe befindet, angezeigt wird, um herauszufinden, ob ich in einem mir unbekannten Ort eine Station in der Nähe habe.

* Die nächstmöglichen Stationen sollen dem Benutzer angezeigt werden.

**Suchtext auto-Vorschlag mit vervollständigen:**

Als Aussendienst Mitarbeiter möchte ich, dass der Text, den ich eingebe, automatisch eine Liste mit möglichen Orten, die mit diesen eigegebenen Buchstaben beginnen, anzeigt, damit ich mir nicht Komplete Namen merken muss.

* Bei der Eingabe von einzelnen Suchbegriffen oder Buchstaben sollen dem Benutzer Suchvorschläge zu Stationen gemacht werden.

**Station auf der Karte anzeigen:**

Als Aussendienst Mitarbeiter möchte ich, dass Start und End Station angezeigt werden oder alle Stationen in der Nähe, um genauere Ortsangaben zu haben.

* Die vom Benutzer gesuchte Station soll auf einer Karte angezeigt werden.

**Verbindung per Mail versenden:**

Als Aussendienst Mitarbeiter möchte ich, dass ich die Verbindung per Mail versenden kann, um meine Mitarbeiter zu informieren oder den Kunden.

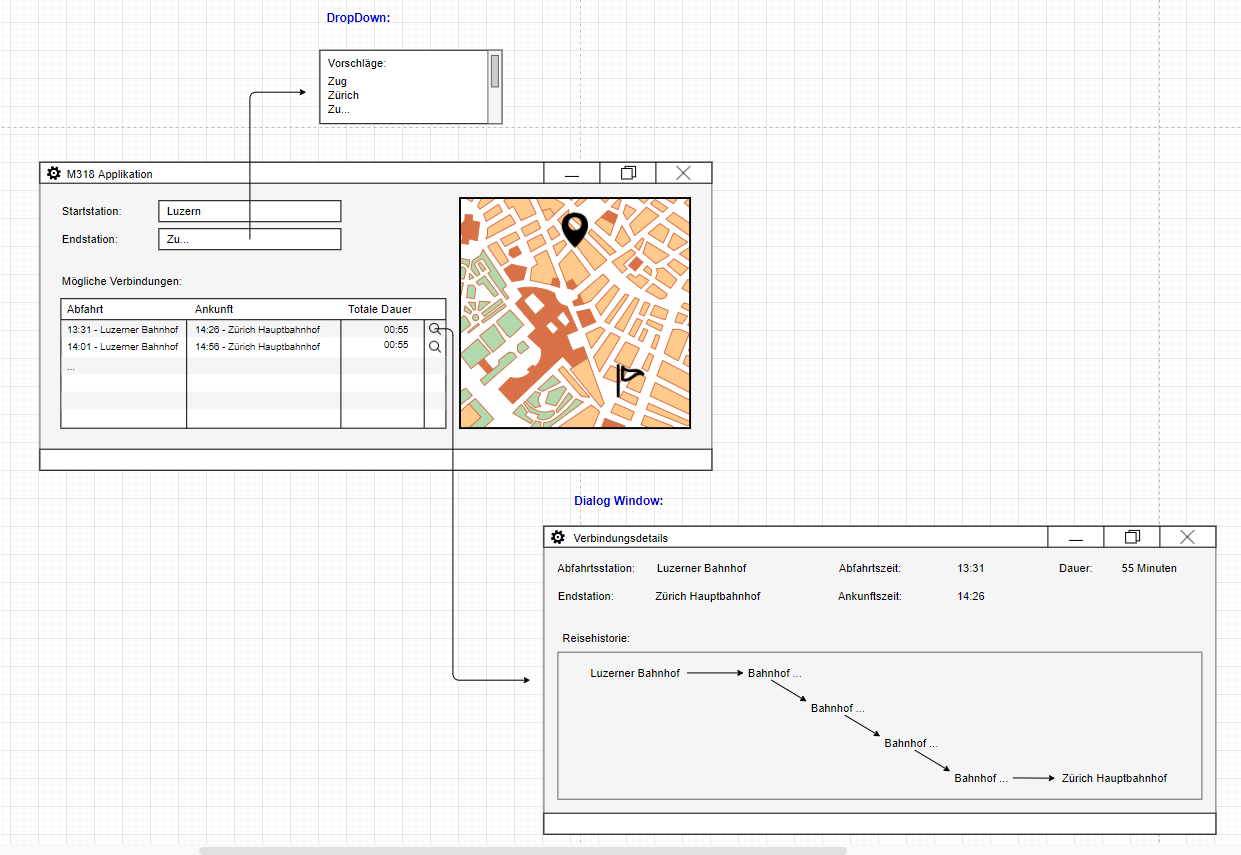
* Die Applikation soll dem Benutzer eine Funktion bieten, welche die ermittelte Verbindung per Mail absenden kann.

**Ankunft / Abfahrt bei Zeit aus wählbar:**

Als Aussendienst Mitarbeiter möchte ich, dass ich bei der Zeit auswählen kann, ob ich um diese Zeit ankomme oder abfahre.

* Der Benutzer soll auswählen können, ob die Zeit die Ankunftszeit oder Abfahrtzeit ist, um das Ergebnis für Ihn besser zu definieren.

## Mockup

Vor der Umsetzung des GUIs mussten wir ein Mockup erstellen, welches eine visuelle Erscheinung der Applikation vorstellungsmässig abbildet. Dabei bekam ich schlussendlich folgendes Resultat:

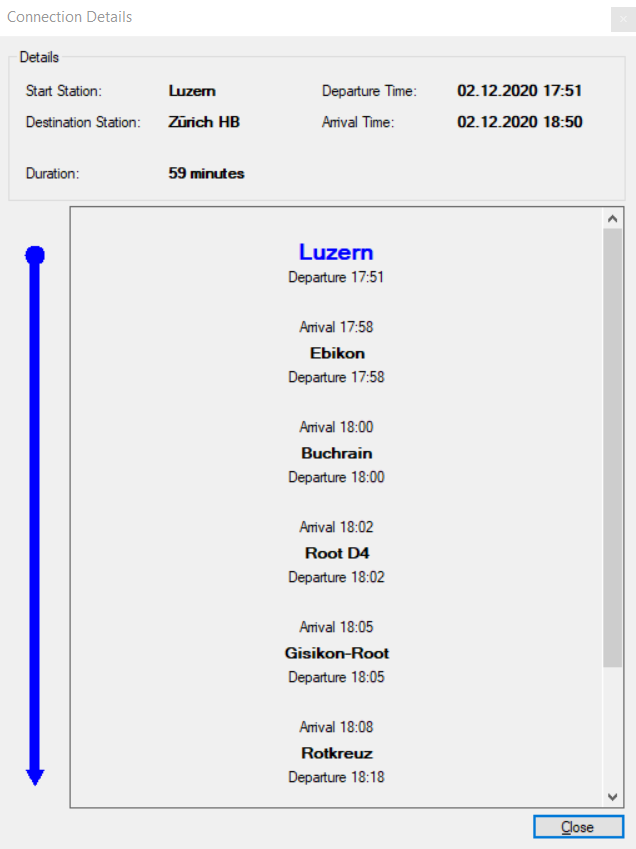
# Umsetzung

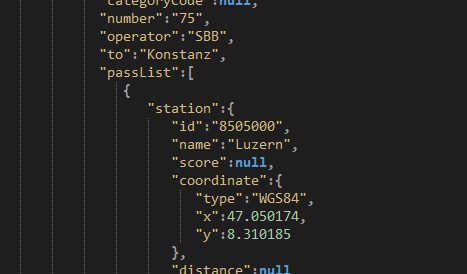
Da wir einen Request auf eine API machen mussten, bekamen wir vom Kurs eine Komponente, welche den Web Request machte.

Die Umsetzung dieses Projektes erlief sehr gut. Ich konnte die grössten meisten Dinge umsetzen und auch sehr gut implementieren. Ich kann hinter meinen Code mit guten Gewissen stehen.

Gegen Ende war ich noch etwas unter Zeitdruck, da ich mich in etwas bestimmtes sehr fest hineingearbeitet habe:

Ich hatte für mein Detailformular vorgesehen, dass jede Zwischenstation, dessen Abfahrt- und dessen Ankunftszeit angezeigt wird. Dabei habe ich vieles überstürzt und ich investierte viel Zeit in das GUI. Als ich dann realisierte, dass die Klassen, welche in der für uns zugestellte Komponente beinhaltet sind, nicht mit meiner Idee harmonieren, war ich sehr schockiert. Ich suchte in der Dokumentation der API nach, liess diese mehrmals durch, und doch fand ich nichts zu Zwischenstationen. Als ich dann über den Browser einen Web Request machte, und ich mir die Response genau unter die Lupe nahm, sah ich, dass es ein Attribut «passList» gab, welches die Zwischenstationen beinhaltet. Ich war sehr stolz auf mich, da ich meiner Ansicht nach etwas implementiert habe, was offiziell nicht machbar wäre. Ich konnte meine Vision realisieren.





# Testing

Beim Testing habe ich mich für dieses Projekt auf User-Acceptance-Tests fokussiert. Da ich keinen Business-Layer habe, und die wichtigsten Methoden in Verbindung mit der API im SwissTransport Projekt schon getestet wurden, verzichte ich auf Unit-Testing.

Die Tests meiner Software wurden von Fabian, meinem Klassenkollegen, durchgeführt. Folgendes Feedback bekam ich von ihm:

«Ich durfte Florians Software durchtesten. Dabei habe ich die verschiedenen Funktionen auf Richtigkeit und auf Fehleranfälligkeit geprüft. Er hat ein sehr gutes Exception-Handling und das Programm hält auch unerwarteten oder noch vorgesehenen Aktionen sehr gut aus. Die Useability seines Programmes ist auch sehr gut. Man kann das ganze Tool benutzen ohne die Maus anzufassen.»