**Dokumentation m318**

Analysieren und objektbasiert programmieren mit Komponenten

Inhalt

[Einleitung 2](#_Toc69465509)

[Zweck des Dokuments 2](#_Toc69465510)

[Mockup 2](#_Toc69465511)

[User Stories 3](#_Toc69465512)

[Aktivitätsdiagramm 5](#_Toc69465513)

[Testing 6](#_Toc69465514)

[Testfälle 6](#_Toc69465515)

[Testprotokoll 6](#_Toc69465516)

[Installation 6](#_Toc69465517)

# Einleitung

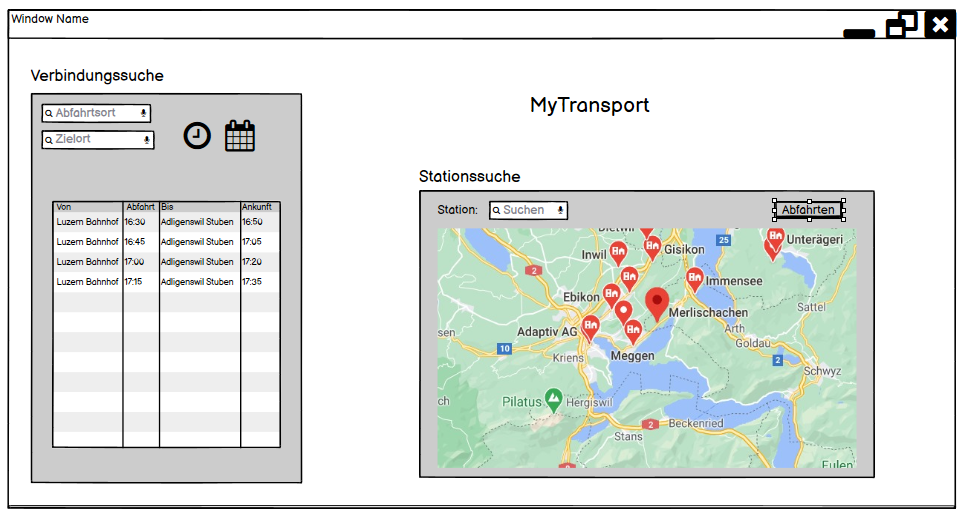
Die «Transportapp» ist die Umsetzung eines Kunden-Auftrags. Anhand des «Gesprächs» mit Hans Kunde sind User Stories entstanden mit den gewünschten Funktionen. Die Abnahmekriterien mit Priorität 1 und 2 müssen im Programm vorhanden sein. Der Rest sind nice-to-have. Aus den User Stories ergab sich ein Mockup design mit den meisten Funktionen. Das Design hat sich nicht bis zum Ende bewahrt, da ich während dem Programmieren einige Anpassungen vorgenommen habe.

Ich hatte knapp eine Woche Zeit für die Umsetzung des Projekts.

# Zweck des Dokuments

Dies ist eine Art Dokumentation für das vorher genannte Projekt. Im ÜK m318 haben wir den Auftrag bekommen. Das Dokument beinhaltet KEINE Arbeitsbeschreibung, sondern die Planung (Mockup, User Stories). Es dient als Beilage zum Programm, und ist essenziell für die Bewertung des Instruktors.

# Mockup



# User Stories

Prioritäten: P1 höchste, P2, P3 niedrigste

Abnahmekriterien: AK

Status: X = Nicht umgesetzt, √ = Erledigt (Abnahmekriterien stimmen)

1. Stationssuche [P1] √

*Als Benutzer möchte ich Abfahrt/Ziel-ort eingeben, um mir eine Busverbindung auszusuchen.*

AK: Es gibt jeweils ein Suchfeld (Start/Ziel)

2. Verbindungen [P1] √

*Als Benutzer möchte ich die nächsten vier Verbindungen angezeigt bekommen, die der Eingabe entsprechen, um einen Bus zu Wählen.*

AK: Die nächsten vier Verbindungen werden unter der Suchleiste angezeigt.

Die gesuchten Orte Stimmen über ein.

3. Abfahrtstafel [P1] √

*Als Benutzer möchte ich eine Station eingeben, um ausgehende Busse anzeigen zu lassen.*

AK: Es gibt eine Stationssuchleiste für dieses feature

Alle ausgehenden Busse werden angezeigt

4. Vervollständigung [P2] √

*Als Benutzer möchte ich bei der Eingabe Vorschläge bekommen, um nicht den ganzen*  *Namen wissen zu müssen.*

AK: Während dem Eintippen erscheinen such Vorschläge

Wenn man zb. Eine Stadt eintippt werden verschiedene Stationen vorgeschlagen

5.Datums/Zeiteingabe [P2] X

*Als Benutzer möchte ich ein Späteres Datum/Uhrzeit eingeben, um in Zukunft eine*  *Verbindung aussuchen zu können.*

AK: Unter der Suchleiste gibt es einen time/date Picker

6. Hilfe-button [P2] (Mein Vorschlag) √

*Als Benutzer möchte ich einen Hilfe Button klicken, um mir die verschiedenen Funktionen zu beschreiben.*

AK: Es gibt einen Hilfe-button

Es werden kleine «Sprechblasen» angezeigt, die mir Infos über das Steuerelement anzeigen, wenn ich drüberfahre

7. Karte [P3] X

*Als Benutzer möchte ich von einer ausgewählten Station den Standort auf der Karte* *angezeigt bekommen, um zu wissen wo diese liegt.*

AK: Es gibt unter dem Eingabe Feld ein «Karte» Button

Beim Klicken wird eine Karte aufgerufen mit dem entsprechenden Standort

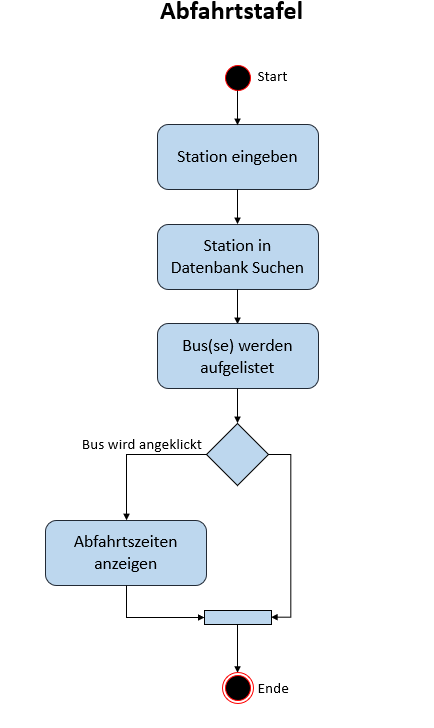
8. Teilen über E-Mail [P3] X

*Als Benutzer möchte ich eine Verbindung über E-Mail teilen, um andere zu informieren.*

AK: Bei der jeweiligen Verbindung gibt es ein «Teilen» Button

Beim Klicken wird eine Mail Vorlage geöffnet mit der entsprechenden Verbindung.

# Aktivitätsdiagramm

****Das Diagramm gehört zur User Story:

3. **Abfahrtstafel**

**!**

Das geplante Diagramm entspricht

nicht 1:1 der schlussendlichen Implementierung.

# Testing

## Testplan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartets Resultat** | **Abw. Resultat** | **Erfüllt?** |
| 1 | Den Tab «Abfahrtstafel» klicken (oben links) | Die Ansicht ändert sich zum Abfahrtsplan. Oben links steht «Abfahrten» |  |  |
| 2 | Den Tab «Verbindungssuche» klicken (oben links) | Die Ansicht ändert sich zum Abfahrtsplan. Oben links steht «Verbindungen Suchen» |  |  |
| 3 | «Hilfe?» Button klicken, und mit der Maus drauf bleiben | Es erscheint eine Sprechblase, die eine Information beinhaltet |  |  |
| 4 | Maus auf «⇄» bewegen | Es erscheint eine Sprechblase, die eine Beschreibung der Funktion (des Buttons) gibt |  |  |
| 5 | Maus auf «⨯» bewegen | **↑↑** |  |  |
| 6 | Maus auf die Tab Leiste bewegen | **↑↑** |  |  |
| 7 | Maus auf «Suchen» bewegen | **↑↑** |  |  |
| 8 | In das «Abfahrtsort» Feld «Lu» eintippen | In der List box unten dran werden «Luzern»,  «Luzern, Bahnhof», «Lugano» vorgeschlagen |  |  |
| 9 | In das «Zielort» Feld «Lu» eintippen | **↑↑** |  |  |
| 10 | Im linken Vorschlag Feld «Lugano» klicken | Die Vorschläge verschwinden wieder, und in der Suchleiste steht «Lugano» |  |  |
| 11 | Im rechten Vorschlag Feld «Luzern, Bahnhof» klicken | Die Vorschläge verschwinden wieder, und in der Suchleiste steht «Luzern, Bahnhof» |  |  |
| 12 | Den «⇄» Button klicken | «Lugano» und «Luzern, Bahnhof» werden getauscht |  |  |
| 13 | Den «Suchen» Button klicken | In der unteren Tabelle kommen 4 Datensätze (Uhrzeiten müssen nicht übereinstimmen) |  |  |
| 14 | Den «⇄» Button klicken | «Lugano» und «Luzern, Bahnhof» werden wieder getauscht und die Suchergebnisse sind (Uhrzeiten müssen nicht übereinstimmen): |  |  |
| 15 | Den Tab «Abfahrtstafel» klicken (oben links) | Die Ansicht ändert sich zum Abfahrtsplan. Oben links steht «Abfahrten» |  |  |
| 16 | In das «Station» Feld «Adlig, St» | In der unteren List box steht nur «Adligenswil, Stuben» |  |  |
| 17 | «Adligenswil, Stuben» anklicken | «Adligenswil, Stuben» Steht nun in der Suchleiste |  |  |
| 18 | «Suchen» Botton klicken | Es erschein eine Liste der Ausgehenden Verbindungen (Uhrzeiten müssen nicht übereinstimmen) |  |  |
| 19 | Neben dem Suchfeld «x» klicken | Eingabefeld, Vorschlagliste, Verbindungsliste sind jetzt leer |  |  |
| 20 | Internet ausschalten, Aktivitäten nochmals Durchgehen | Die erwarteten Resultate (1-19) sind bei diesem Schritt nicht zu beachten.  Das Programm darf lediglich nicht absürzen |  |  |

## Testprotokoll

# Installation