David Cvetkovic

Modul 318 Projekt

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc69479219)

[2 Zweck des Dokumentes 2](#_Toc69479220)

[3 Mockup 2](#_Toc69479221)

[4 User Stories 3](#_Toc69479222)

[4.1 Höchste Priorität 3](#_Toc69479223)

[4.2 Zweithöchste Priorität 3](#_Toc69479224)

[4.3 Dritthöchste Priorität 4](#_Toc69479225)

[4.4 Mögliche Vorschläge 4](#_Toc69479226)

[5 Aktivitätsdiagramm 4](#_Toc69479227)

[6 Testfälle 5](#_Toc69479228)

[7 Testprotokoll 6](#_Toc69479229)

[8 Installationsanleitung 7](#_Toc69479230)

[9 Fazit 7](#_Toc69479231)

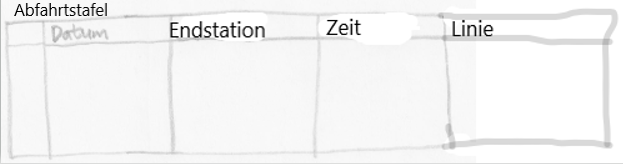
# Einleitung

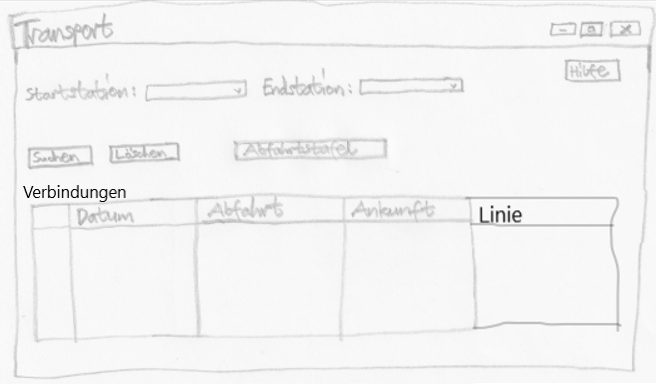
Industriespionage können geschehen, und deshalb ist es wichtig, dass man sich davor schützt. Eine Option wäre, dass man ein Tablet als Arbeitsgerät benutzt, das aus Sicherheitsgründen keinen Browser installiert hat. Die Aussendienstmitarbeitende müssen somit dieses Tablets benutzen und nicht ein anderes Arbeitsgerät, da sie sich schützen sollen. Sie müssen mit dem Tablet schnell eine ÖV-Verbindung suchen können. Ziel ist es, ein Programm zu erstellen, dass dies erlaubt. Mit diesem Programm kann man ohne Browser nach einer ÖV-Verbindung suchen.

# Zweck des Dokumentes

Mit dieser Dokumentation will ich zeigen, wie die Entwicklung des Programms von Anfang bis Ende ausgesehen hat. Auch ist es wichtig zu zeigen, was denn die Erwartungen für das Programm sind, um diese dann auch umsetzen zu können. Schlussendlich soll der Leser verstehen, was ich genau gemacht habe, um sich einen Überblick über das Ganze zu verschaffen.

# Mockup





# User Stories

## Höchste Priorität

Verbindungen (Story 1): Als Benutzer möchte ich die nächsten vier Verbindungen zwischen Start- und Endstation sehen, um mich für eines entscheiden zu können, dass mir besser passt.

Abnahmekriterien:

* Der Programmierer hat die nächsten vier Verbindungen programmiert
* Der Benutzer hat die Start- und Endstation eingegeben

*Story 1 wurde umgesetzt.*

Stationssuche (Story 2): Als Benutzer möchte ich eine Stationssuche, um nach einer Station zu suchen.

Abnahmekriterien:

* Der Programmierer hat die Stationssuche programmiert
* Der Benutzer hat eine Station eingegeben

*Story 2 wurde umgesetzt.*

Abfahrtstafel (Story 3): Als Benutzer möchte ich eine Abfahrtstafel, um direkt zu sehen, welche Verbindungen von dort weg gehen.

Abnahmekriterien:

* Der Programmierer hat die Abfahrtstafel programmiert
* Der Benutzer hat eine Station eingegeben

*Story 3 wurde umgesetzt.*

## Zweithöchste Priorität

Vorschläge (Story 4): Als Benutzer möchte ich während dem Eintippen Vorschläge von Stationen erhalten, um schneller die richtige Station auszuwählen.

Abnahmekriterien:

* Der Programmierer hat die Vorschläge programmiert
* Der Benutzer ist die Station am Eingeben

*Story 4 wurde umgesetzt.*

Datum/Uhrzeit (Story 5): Als Benutzer möchte ich ein Datum und eine Uhrzeit eingeben können, um Verbindungen in der Zukunft zu suchen.

Abnahmekriterien:

* Der Programmierer hat Datum und Uhrzeit für die Zukunft programmiert
* Der Benutzer hat ein Datum und eine Uhrzeit in der Zukunft eingegeben

*Story 5 wurde nicht umgesetzt.*

## Dritthöchste Priorität

Karte (Story 6): Als Benutzer möchte ich eine Karte, um zu sehen, wo sich die Station befindet.

Abnahmekriterien:

* Der Programmierer hat eine Karte mit den Stationen programmiert
* Der Benutzer hat eine Station eingegeben und die Karte geöffnet

*Story 6 wurde nicht umgesetzt.*

Nächstgelegene Stationen (Story 7): Als Benutzer möchte ich vom aktuellen Standort die nächsten Stationen sehen, um mich für die nächstgelegene Station entscheiden zu können.

Abnahmekriterien:

* Der Programmierer hat die nächstgelegenen Stationen vom aktuellen Standort programmiert
* Der Benutzer hat die Karte geöffnet, um die nächstgelegenen Stationen zu sehen

*Story 7 wurde nicht umgesetzt.*

E-Mail (Story 8): Als Benutzer möchte ich jemanden direkt per Mail über die Zugverbindung informieren, um sicherzustellen, dass der Empfänger über mögliche Anpassungen der Fahrten Bescheid weiss.

Abnahmekriterien:

* Der Programmierer hat eine Mail-Funktion programmiert
* Der Benutzer hat die Zugverbindung gewählt und diese einem Empfänger geschickt

*Story 8 wurde nicht umgesetzt.*

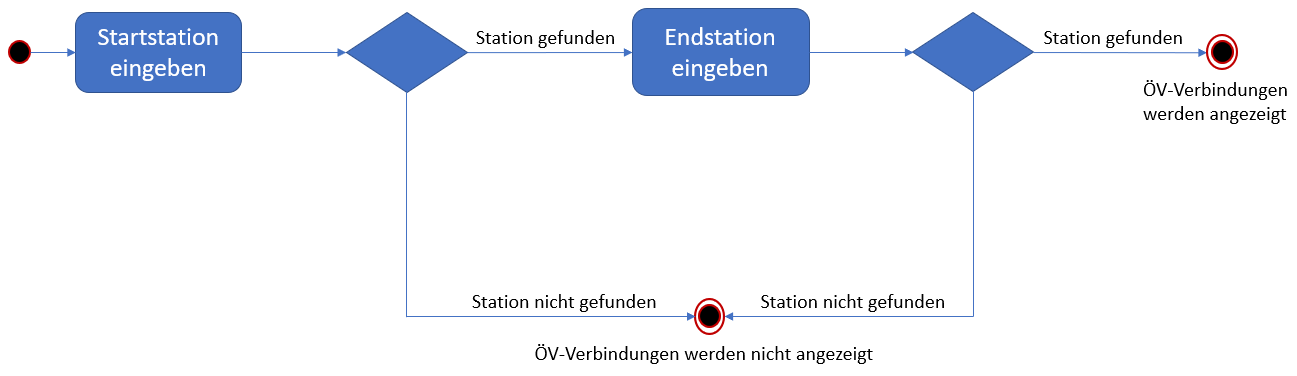
## Mögliche Vorschläge

Story 9: Als Benutzer möchte ich den Suchverlauf der vorherig eingegebenen Verbindungen sehen, um direkt diese wieder auswählen zu können, falls ich dieselbe Verbindung wieder nehme.

Story 10: Als Benutzer möchte ich auf einer Karte selbst ein Standort festlegen, um dort die nächstgelegenen Stationen zu sehen.

# Aktivitätsdiagramm

Hier sieht man ein Aktivitätsdiagramm von Story 1:



# Testfälle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Abw. Resultat** | **Erfüllt** |
| 1 | Programm öffnen. | Das Programm soll geöffnet werden. |  |  |
| 2 | Den Button «Hilfe» drücken. | Ein kleines Fenster wird geöffnet. Es erklärt, wie man das Programm verwendet. |  |  |
| 3 | Das kleine «x» oben rechts oder «OK» drücken. | «Hilfe»-Fenster wird geschlossen. |  |  |
| 4 | Start- und Endstation eingeben. | Während dem Eintippen werden Vorschläge angezeigt. Bei nicht eingegebenen Stationen erscheint ein kleines Fenster, dass den Benutzer darauf hinweist. |  |  |
| 5 | Den Button «Suchen» drücken. | Die nächsten vier Verbindungen werden angezeigt. |  |  |
| 6 | Den Button «Löschen» drücken. | Die im Textfeld eingegebenen Stationen und die Verbindungen werden gelöscht. |  |  |
| 7 | Den Button «Abfahrtstafel» drücken. | Eine Meldung soll erscheinen, dass man eine Startstation eingeben soll. |  |  |
| 8 | Das kleine «x» oben rechts oder «OK» drücken. | Meldungsfenster wird geschlossen. |  |  |
| 9 | Startstation eingeben. | Vorschläge sollen während dem Eintippen erscheinen. |  |  |
| 10 | Den Button «Abfahrtstafel» drücken. | Die Abfahrten werden angezeigt. |  |  |
| 11 | Den Button «Löschen» drücken. | Die Abfahrten und die eingegebene Startstation werden gelöscht. |  |  |

# Testprotokoll

Durchführungsdatum: 16.04.2021

Tester: David Cvetkovic

Umgebung: Visual Studio 2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** | **Abw. Resultat** | **Erfüllt** |
| 1 | Programm öffnen. | Das Programm soll geöffnet werden. |  | ☑ |
| 2 | Den Button «Hilfe» drücken. | Ein kleines Fenster wird geöffnet. Es erklärt, wie man das Programm verwendet. |  | ☑ |
| 3 | Das kleine «x» oben rechts oder «OK» drücken. | «Hilfe»-Fenster wird geschlossen. |  | ☑ |
| 4 | Start- und Endstation eingeben. | Während dem Eintippen werden Vorschläge angezeigt. Bei nicht eingegebenen Stationen erscheint ein kleines Fenster, dass den Benutzer darauf hinweist. |  | ☑ |
| 5 | Den Button «Suchen» drücken. | Die nächsten vier Verbindungen werden angezeigt. |  | ☑ |
| 6 | Den Button «Löschen» drücken. | Die im Textfeld eingegebenen Stationen und die Verbindungen werden gelöscht. |  | ☑ |
| 7 | Den Button «Abfahrtstafel» drücken. | Eine Meldung soll erscheinen, dass man eine Startstation eingeben soll. |  | ☑ |
| 8 | Das kleine «x» oben rechts oder «OK» drücken. | Meldungsfenster wird geschlossen. |  | ☑ |
| 9 | Startstation eingeben. | Vorschläge sollen während dem Eintippen erscheinen. |  | ☑ |
| 10 | Den Button «Abfahrtstafel» drücken. | Die Abfahrten werden angezeigt. |  | ☑ |
| 11 | Den Button «Löschen» drücken. | Die Abfahrten und die eingegebene Startstation werden gelöscht. |  | ☑ |

# Installationsanleitung

Im ZIP-Ordner befindet sich eine .exe-Datei. Um das Programm benutzen zu können, muss man diese .exe-Datei öffnen.

Wenn man die Software nicht mehr will, kann man direkt alle Dateien vom ZIP-Ordner löschen.

# Fazit

Am interessantesten fand ich die Verbindungen und die Abfahrtstafel. Ich habe gelernt, wie man diese Funktionen programmiert. Mit dieser Dokumentation konnte ich immer alles dokumentieren, um mir einen klareren Überblick zu verschaffen. Mein Ziel war es, alle Funktionen der ersten Priorität zu programmieren und einen möglichst benutzerfreundlichen GUI zu erstellen, und das Ziel habe ich auch erreicht.