## Dokumentation M318 Alejandro Facal

Inhalt

[Dokumentation M318 Alejandro Facal 1](#_Toc100844570)

[Einleitung Projekt Alejandro Facal 2](#_Toc100844571)

[User Stories: 3](#_Toc100844572)

[Aktivitätsdiagramm zu: Als User möchte ich die nächsten 4 Verbindungen zwischen Start- und Endstation finden können 5](#_Toc100844573)

[5](#_Toc100844574)

[Umgesetzte Funktionen: 6](#_Toc100844575)

[Mockups 7](#_Toc100844576)

[Testfälle 8](#_Toc100844577)

[Testprotokoll 10](#_Toc100844578)

# Einleitung Projekt Alejandro Facal

In diesem Projekt geht es um die Umsetzung von verschiedene User-Stories und Anforderungen, die uns als Form vom ÜK M318 gestellt wurde. Meine Applikation in Windows-Forms, soll durch Stationssuche, Abfahrtstafeln und Verbindungen-Suche allgemeine Transport-Hilfe anbieten. Die Daten von Stationen, Verbindungen usw. bekomme ich von einer bereitgestellten API. Meine eigentliche Aufgabe ist es, durch die vorher erwähnten Anforderungen einen funktionierenden GUI zu programmieren.

# User Stories:

|  |  |  |
| --- | --- | --- |
| User-Stories | Abnahmekriterien | Prio |
| Als User möchte ich die nächsten 4 Verbindungen zwischen Start- und Endstation finden können | * Eingabe Start- und Endstation * Min 4 Verbindungen zwischen Start- und Endstatoin werden ausgegeben | Prio 1 |
| Als User möchte ich Vorschläge von Stationen bekommen, auch wenn ich nicht den kompletten Namen eingegeben habe | * Stationen welche mit der Eingabe übereinstimmen werden angezeigt * Bei Click auf Vorschlag ist der Value von Vorschlag die neue Eingabe | Prio 2 |
| Als User möchte ich eine Abfahrtstafel zur Verfügung haben, die mir alle Verbindungen einer bestimmten Station anzeigt, die von dort weggehen | * Eingabe von Station * Bei Click Verbindungen die von der Eingabe-Station weggehen anzeigen | Prio 1 |
| Als User möchte ich zukünftige Verbindungen mit Datum und Uhrzeit finden können | * Eingabe von Datum und Zeit * Bei Click werden Verbindungen im eingegeben Datum angezeigt | Prio 2 |
| Als User möchte ich auf einer Karte den Standort einer bestimmten Station sehen/finden. | * Eingabe von Station * Standort von Eingabe-Station wird angezeigt | Prio 3 |
| Als User möchte ich die nahesten Stationen an meinem Standort sehen | * Erkennung von eigenen Standort * Naheliegende Standorte werden angezeigt | Prio 3 |
| Als User möchte ich jemanden direkt per Mail über eine bestimmte Zugverbindung informieren können | * Eingabe Content, Sender- und Empfängermail | Prio 3 |
| Als User möchte ich durch einen Click, die Start- und Endstationen miteinander vertauschen, da es für die Heimreise praktisch wäre | * Eingabe Start- und Endstationen * Eingabe-Values von beiden Textboxen werden vertauscht | Eigene Idee |
| Als User möchte ich bestimmte Stationen suchen und finden können | * Eingabe von Station * Durch Click werden alle Stationen, die mit der Eingabe-Station übereinstimmen angezeigt | Prio 1 |

# Aktivitätsdiagramm zu: Als User möchte ich die nächsten 4 Verbindungen zwischen Start- und Endstation finden können

# 

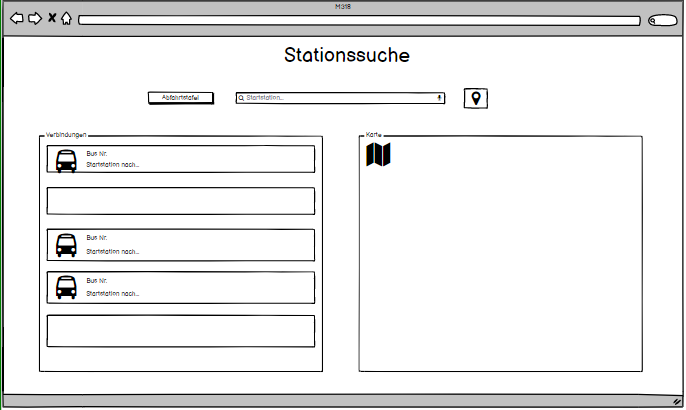
# Umgesetzte Funktionen:

|  |  |  |
| --- | --- | --- |
| Funktion | Status | ID |
| Verbindungen zwischen Start- und Endstationen | Umgesetzt | 1 |
| Abfahrtstafel zu bestimmter Station | Umgesetzt | 2 |
| Suche von bestimmten Stationen | Umgesetzt | 3 |
| Vorschläge anzeigen bei Wählen von Start- und Endstationen | Umgesetzt | 4 |
| Verbindungen in der Zukunft anzeigen | Umgesetzt | 5 |
| Standort einer bestimmten Station auf der Karte | Nicht umgesetzt | 6 |
| Stationen in der Nähe per Karte anzeigen | Nicht umgesetzt | 7 |
| Per Mail über Zugverbindung informieren können | Nicht umgesetzt | 8 |
| Start- und Endstationen können miteinander vertauscht werden (Eigene Idee) | Umgesetzt | 9 |

# Mockups

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



# Testfälle

**Testfälle zu folgenden Funktionen:**

* ID 4
* ID 1
* ID 5
* ID 9

**Test 1:**

Vorschläge bei Eingabe von Start- und Endstation

|  |  |
| --- | --- |
| Aktion | Erwartetes Resultat |
| User gibt in Textbox 1 (AnzeigeStStation) oder in Textbox 2 (AnzeigeEndStation) den ersten Buchstaben ein. | Auf einen Datagridview werden alle Stationen anzeigt, welche mit dem Text in der Textbox übereinstimmen |

**Test 2:**

Verbindungen zwischen Start- und Endstationen anzeigen

|  |  |
| --- | --- |
| Aktion | Erwartetes Resultat |
| User gibt in Textbox 1 (AnzeigenStStation) die Startstation ein und in Textbox 2 (AnzeigenEndeStation) die EndStation ein. User clickt dann auf den Suchen Button | Alle Verbindungen zwischen Start- und Endstation werden angezeigt. |
|  |  |

**Test 3:**

Zukünftige Verbindungen anzeigen

|  |  |
| --- | --- |
| Aktion | Erwartetes Resultat |
| User wählt nach Eingabe von Start- und Endstation ein Datum in der Zukunft ein (Im Datetimepicker). Dann drückt er auf den Suchen-Button | Alle Verbindungen zwischen Start- und Endstation welche mit dem eingegebenen Datum übereinstimmen werden angezeigt |

**Test 4:**

Startstation mit Endstation wechseln

|  |  |
| --- | --- |
| Aktion | Erwartetes Resultat |
| User drückt nach Eingabe von Start- und Endstation in Textbox 1 und 2 den Change-Button () | Textbox 1 bekommt den Inhalt von Textbox 2 und Textbox 2 bekommt den Inhalt von Textbox 1 |

**Testfälle zu folgenden User-Stories:**

* ID 3

**Test 5:**

Stationen suchen

|  |  |
| --- | --- |
| Aktion | Erwartetes Resultat |
| User gibt in der Textbox im Stationssuchen-Tab den Namen einer beliebigen Station ein. Dann clickt der User auf den Suchen-Button | Alle Stationen die mit dem Namen in der Textbox übereinstimmen werden angezeigt. |

**Testfälle zu folgenden User-Stories:**

* ID 2

**Test 6:**

Abfahrtstafel zu einer bestimmten Station

|  |  |
| --- | --- |
| Aktion | Erwartetes Resultat |
| User gibt in der Textbox im AbfahrtsTafel-Tab den Namen einer bestimmten Station. Dann clickt der User auf den Suchen-Button. | Alle Verbindungen, die von der Start-Station (Eingabe von User) weggehen, werden angezeigt. |

# Testprotokoll

Protokoll zu Test 1

|  |  |  |
| --- | --- | --- |
| Durchführungsdatum | Name des Testers | Testresultate |
| 14.04.2022 | Alejandro Facal | Erfolgreich |

Protokoll zu Test 2

|  |  |  |
| --- | --- | --- |
| Durchführungsdatum | Name des Testers | Testresultate |
| 14.04.2022 | Alejandro Facal | Erfolgreich |

Protokoll zu Test 3

|  |  |  |
| --- | --- | --- |
| Durchführungsdatum | Name des Testers | Testresultate |
| 14.04.2022 | Alejandro Facal | Erfolgreich |

Protokoll zu Test 4

|  |  |  |
| --- | --- | --- |
| Durchführungsdatum | Name des Testers | Testresultate |
| 14.04.2022 | Alejandro Facal | Erfolgreich |

Protokoll zu Test 5

|  |  |  |
| --- | --- | --- |
| Durchführungsdatum | Name des Testers | Testresultate |
| 14.04.2022 | Alejandro Facal | Erfolgreich |

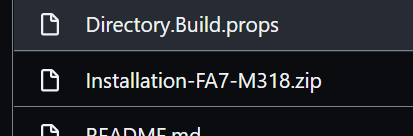
Protokoll zu Test 6

|  |  |  |
| --- | --- | --- |
| Durchführungsdatum | Name des Testers | Testresultate |
| 14.04.2022 | Alejandro Facal | Erfolgreich |

# Installationsanleitung

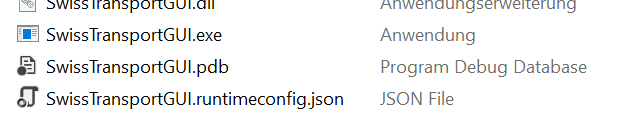
Schritt 1:

Im GitHub Repositorie Zip-File herunterladen



Schritt 2:

.exe File ausführen



Autor: Alejandro Facal