Inhaltsverzeichnis

[1 Management Summary 3](#_Toc34901330)

[2 Zweck 4](#_Toc34901331)

[3 Bewertung 5](#_Toc34901332)

[4 Funktion 6](#_Toc34901333)

[5 Mockups 7](#_Toc34901334)

[6 Use Case 8](#_Toc34901335)

[7 Aktivitätendiagramme 9](#_Toc34901336)

[8 Systemtests 10](#_Toc34901337)

[9 Testprotokoll 11](#_Toc34901338)

[10 Installationsanleitung 12](#_Toc34901339)

[11 Reflexion 13](#_Toc34901340)

# Management Summary

Im ÜK M318 wurde das Projekt «Swiss Transport App» erstellt und entwickelt. Bei diesem Projekt sollte eine Applikation entwickelt werden welche HTTP Web APIs verwendet, um Daten zu holen. Diese Daten sollten mit Hilfe von C# und Visual Studio in ein GUI eingebunden werden.

# Zweck

Durch dieses Dokument wird das Programm «Swiss Transport App» beschrieben sowie die Entstehung erklärt. Dieses Dokument beinhaltet die Planung sowie die Umsetzung des Projektes und der Umgang mit der Web API. Dieses Projekt wurde von Simon Krieger im Rahmen des Kurses M318 entwickelt und fertiggestellt.

Der Zweck des Kurses war es, die Kenntnisse im Umgang mit C# und WPF/WinForms zu verbessern sowie die Kenntnisse im Objektorientierten Programmieren zu verbessern.

# Bewertung

Die Applikation musste mindestens die Aufgaben A001 – A003 erfüllen.

Das Projekt wird folgendermassen bewertet:

## Qualität

### Funktionalität

Hier wird geprüft, ob die Software richtig funktioniert.

### Zuverlässigkeit

Hier schaut man ob die Software zuverlässig funktioniert, was in Fehler-Situationen passiert, ob verständliche Fehlermeldungen angezeigt werden, ob sich Daten wiederherstellen lassen usw.

### Benutzbarkeit

Kann man die Software einfach bedienen? Ist die Benutzeroberfläche übersichtlich und selbsterklärend gestaltet? Versteht man die Funktionen?

### Effizienz

Definiert die Performance (das Zeitverhalten) der Applikation, den Speicherverbrauch usw.

### Wartbarkeit

Kann man die Software einfach/problemlos erweitern? Wie einfach ist es, Fehler einzugrenzen und zu lokalisieren? Wie stabil verhält sich die Software bei Änderungen (Stichwort: Side Effects)? Ist der Source Code gut und verständlich kommentiert? Wurden die Programmierrichtlinien durchgängig eingehalten?

### Portierbarkeit

Kann man die Software einfach installieren? Kann sie mit anderen Programmen koexistieren? Welche Auswirkungen hat die Installation der Software auf andere Programme? usw.

## Qualitätsziele

### Funktionalität

Die Funktionalität der Software soll den Anforderungen entsprechen, welche laut Dokumentation umgesetzt wurden.

Die Funktionen sollen korrekt umgesetzt sein.

Die richtige Funktionsweise soll durch Testfälle sichergestellt sein.

### Zuverlässigkeit

Die Software soll zuverlässig funktionieren und tun, was man von ihr erwartet.

Die Software soll in Fehlerfällen den Benutzer korrekt (mit verständlichen Fehlermeldungen) informieren, ohne abzustürzen.

Nach Auftreten eines Fehlers sollte die Software, wenn möglich, normal weiterlaufen.

### Benutzbarkeit

Die Bedienung der Software soll selbsterklärend sein.

Benutzereingaben sollen validiert werden.

Die grafische Benutzeroberfläche (GUI) soll übersichtlich gestaltet sein.

Das GUI soll sich an UI Standards halten.

### Wartbarkeit

Der Source Code soll sich an die Programmierrichtlinien (Coding Guidelines) halten.

Variablen, Klassen, Methoden und andere Elemente sollen sprechende Namen haben.

Der Source Code soll verständlich kommentiert sein, insbesondere sollen alle öffentlichen Methoden inklusive ihrer Parameter kommentiert sein.

Es soll einfach und problemlos möglich sein, Änderungen an der Software zu machen. Dazu gehört: kein repetitiver Code (copy paste), kein Spaghetticode, Beachtung des Kohäsionszprinzips und eine möglichst lose Kopplung der einzelnen Module und Klassen.

### Portierbarkeit

Die Software soll einfach zu installieren sein.

Die Software soll auch einfach wieder entfernt werden können.

Die Software soll nach der Installation auch auf einem Rechner ohne Entwicklungsumgebung lauffähig sein.

Zusammen mit der Software soll eine Installationsanleitung ausgeliefert werden.

## Dokumentation

* Autor, Datum.
* Eine Einleitung (Management Summary).
* Zweck des Dokuments.
* Was (d.h. welche Funktionen) wurde umgesetzt?
* Falls bestimmte Funktionen nur teilweise umgesetzt wurden: Welche? Welcher Teil der Funktionalität fehlt noch? Bekannte Fehler/Bugs?
* Mockups für alle Formulare/Dialoge (GUI) , die benötigt werden um die Anforderungen A001, A002 und A003 abzudecken.
* Use Cases, Use Case Beschreibungen und Aktivitätendiagramme mindestens für die mit Priorität 1 klassifizierten Anforderungen.
* Testfälle (Systemtests), verständlich und eindeutig nachvollziehbar geschrieben, so dass ein Tester diese ohne weiteren Erklärungen durchführen kann.
* Testprotokoll mit Durchführungsdatum der Tests, Name des Testers und Protokollierung der tatsächlichen Testresultate.
* Installationsanleitung: Wie wird die Software installiert? Wie wird die Software deinstalliert?
* Andere spannende Informationen für die Bewertung.

# MOCKUPS

Die nachfolgenden Mockups wurden bei der Planung erstellt.

## Verbindungen

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

## Abfahrtstafel

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

## Stationen

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

## Mockups vs Endprodukt

Beim Vergleich zum Endprodukt sind die Unterschiede nur minimal. Die Unterschiede sind folgende:

* E-Mail-Export Button
* E-Mail-Export Window
* Spalten in den Tabellen

# Funktion

## Anforderungen

Die folgende Tabelle zeigt welche Anforderungen und Aufgaben dieses Projekt beinhaltet.

* 1: Must
* 2: Should
* 3: Nice to have

|  |  |  |  |
| --- | --- | --- | --- |
| **A001** | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | Erledigt | 1 |
| **A002** | Als ÖV-Benutzer möchte ich die aktuellen, d.h. mindestens die nächsten vier bis fünf Verbindungen zwischen den beiden gefundenen und ausgewählten Stationen sehen, damit ich weiss wann ich zur Station muss, um den für mich idealen Anschluss zu erwischen. | Erledigt | 1 |
| **A003** | Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann. | Erledigt | 1 |
| **A004** | Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann. | Erledigt | 2 |
| **A005** | ALs ÖV-Benutzer möchte ich nicht nur aktuelle Verbindungen suchen können, sondern auch solche zu einem beliebigen anderen Zeitpunkt, damit ich zukünftige Reisen planen kann. | Erledigt | 2 |
| **A006** | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. | Erledigt | 3 |
| **A007** | Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann. | Erledigt | 3 |
| **A008** | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. | Erledigt | 3 |

## A001

Bei den Textboxen wird ein Vorschlag des eingegebenen Wortes angezeigt um den den richtigen Stationsnamen auszuwählen ohne den Namen auswendig lernen zu müssen.

## A002

Beim Tab “Verbindungen” werden mindestens vier bzw. fünf Verbindungen zwischen der Start- und Endstation im DataGrid angezeigt.

## A003

Beim Tab “Abfahrtstafel” wird eine Abfahrtstafel für die eingegebene Station im DataGrid angezeigt.

## A004

Beim Tab “Abfahrtstafel” und “Verbindungen” werden die verfügbaren Stationen in der Textbox mit einem Dropdown angezeigt sobald der Benutzer mehr als zwei Zeichen eingetippt hat.

## A005

Durch die Datums und Zeit Auswahlbox kann ein Datum und eine Uhrzeit ausgeählt werden. Dazu kann noch nach der Ankunfts- oder der Abfahrtszeit gesucht werden.

## A006

Beim Tab “Stationen” kann nach Stationen gesucht werden. Die Suchergebnisse werden im DataGrid angezeigt. Mit dem Button “Map öffnen” wird ein Browser-Fenster geöffnet mit dem Standort der ausgewählten Station.

## A007

Beim Tab “Stationen” kann auf den Button “In meiner Nähe” geklickt werden um die Stationen in der Nähe anzuzeigen. Die Resultate werden im DataGrid angzeigt. Mit dem Button “Map öffnen” wird ein Browser-Fenster geöffnet mit dem Standort der ausgewählten Station.

## A008

Beim Tab “Verbindungen” kann nach Verbindungen gesucht werden. Diese gefunden Verbindungen können per Mail versendet werden. Dazu klickt man auf den Button “E-Mail-Export”. Es öffnet sich ein neues Fenster, bei diesem Fenster kann der Empfänger, Betreff und der Mail-Text eingegeben werden. Der Mail-Text wird mit allen gefunden Verbindungen automatisch ausgefüllt. Beim klick auf den Button “Mail-To” wird ein Mail im Standard Mail-Programm verfasst und geöffnet. Dieses Mail kann dan verschickt werden.

# Use Case

## Use Case Diagramm

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Use Case Beschreibung

### A001

|  |  |
| --- | --- |
| Use Case | 1. Start- und Endstation suchen |
| Beschreibung | Der ÖV-Benutzer sucht nach Start- und Endstation. |
| Akteur | ÖV-Benutzer |
| Auslöser | Der ÖV-Benutzer benötigt den Stationsnamen für Start- oder Endstation für die Verbindungssuche oder die Abfahrtstafel. |
| Vorbedingung | Keine |
| Ablauf | 1. Swiss Transport App öffnen  2. Suchbergriff eingeben  3. Suche klicken |
| Ergebnis | Die Stationen werden dem Benutzer angezeigt. |

### A002

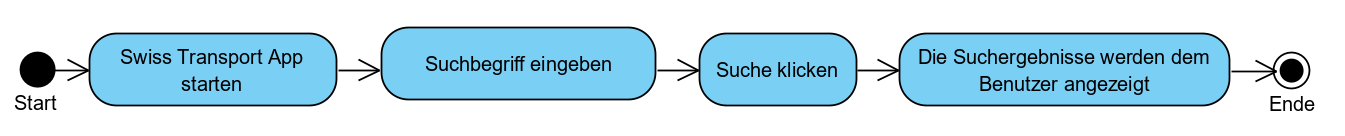
|  |  |
| --- | --- |
| Use Case | 2. Verbindungen anzeigen |
| Beschreibung | Der ÖV-Benutzer sucht nach einer Verbindung zwischen zwei eingegebenen Stationen. |
| Akteur | ÖV-Benutzer |
| Auslöser | Der ÖV-Benutzer benötigt eine Verbindung zwischen zwei Stationen. |
| Vorbedingung | Keine |
| Ablauf | 1. Swiss Transport App öffnen  2. Startstation eingeben  3. Endstation eingeben  4. Suche klicken |
| Ergebnis | Die Verbindungen werden dem ÖV-Benutzer angezeigt |

### A003

|  |  |
| --- | --- |
| Use Case | 3. Abfahrtstafel anzeigen |
| Beschreibung | Der ÖV-Benutzer sucht nach den nächsten Fahrten einer Station. |
| Akteur | ÖV-Benutzer |
| Auslöser | Der ÖV-Benutzer benötigt eine Abfahrtstafel einer Station |
| Vorbedingung | Keine |
| Ablauf | 1. Swiss Transport App öffnen  2. Station eingeben  3. Fahrten suchen |
| Ergebnis | Die Abfahrtstafel wird dem ÖV-Benutzer angezeigt |

# Aktivitätsdiagramm

## A001

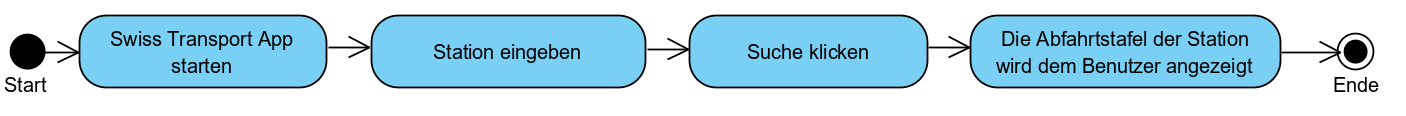


## A002

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

## A003



# Systemtests

## Voraussetzungen

# Testprotokoll

# Installationsanleitung

# Reflexion