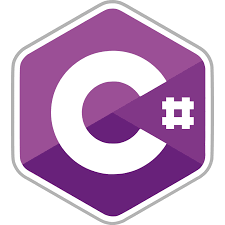
Dokumentation

M318 Analysieren und objektbasiert programmieren

24. April 2020

Noah Götschi



Inhaltsverzeichnis

Inhalt

[1. Funktionalität 2](#_Toc38630890)

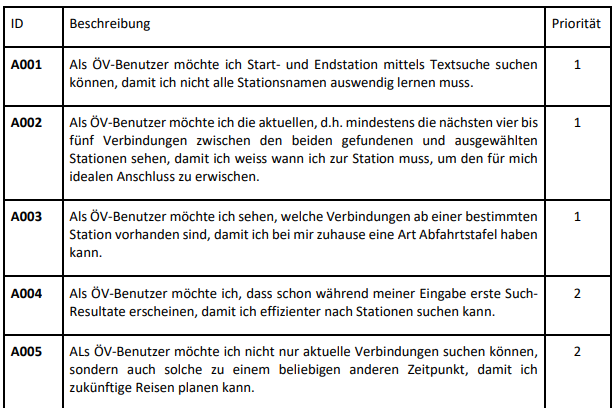
[2. GUI Mockup 3](#_Toc38630891)

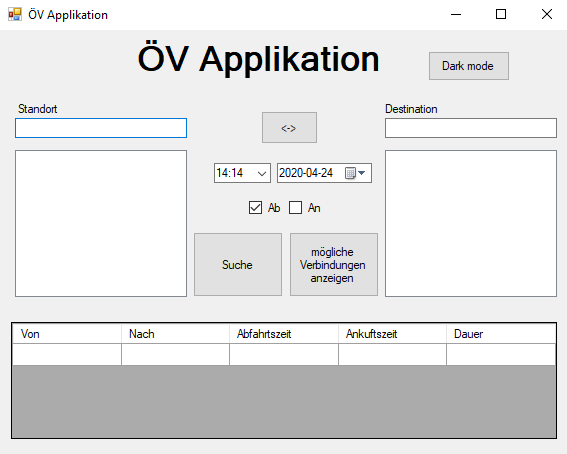
[3. Use Case Diagramm 4](#_Toc38630892)

[4. Aktivitätsdiagramm 6](#_Toc38630893)

[5. Programmrichtlinien 7](#_Toc38630894)

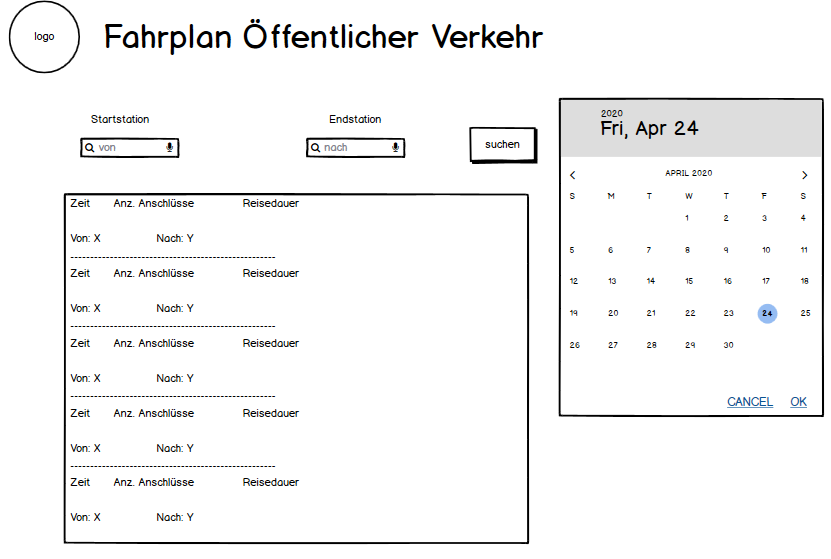
# Funktionalität

Die Anforderungen A001-A005 wurden erfolgreich im Programm umgesetzt.

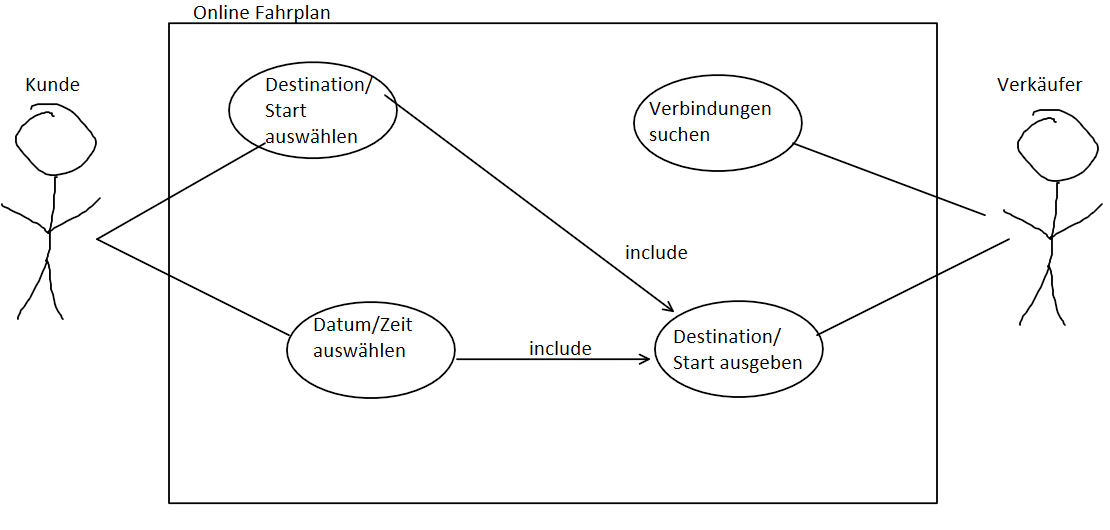
Screenshot der geforderten Applikation

# GUI Mockup

Erstes Mockup des Programms gegen Anfang des Kurses, wie das Programm hätte aussehen können.



# Use Case Diagramm

  
Entwurf des ÖV Programms für die Anforderungen A001-A003 anhand eines Use-Case Diagramms.  
Hier wird beschrieben welche Leistungen der Kunde erwartet vom Programm und wie der Verkäufer, (also wir, die Lernenden) mit dem Programm erfüllen müssen.

Die zugehörigen Use-Case Beschreibungen für die gefragten Anforderungen in Detail.

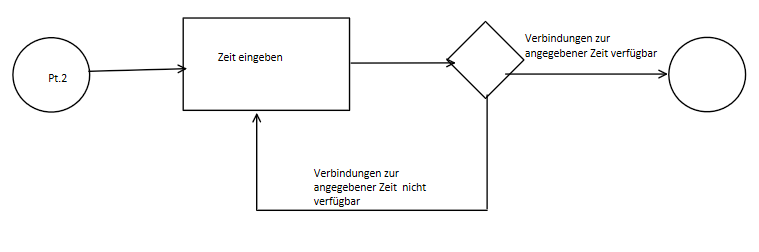
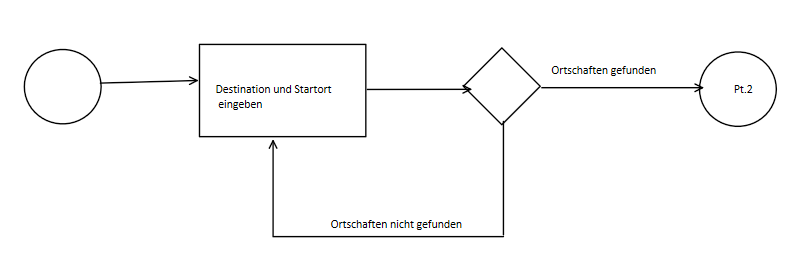
|  |  |
| --- | --- |
| Use Case | Destination/Start ausgeben |
| Beschreibung | Ausgewählte beliebige Standorte für den Transport ausgeben können |
| Akteur | Noah Götschi |
| Auslöser | Verlangen auf ein Online Fahrplan |
| Vorbedingungen | Der User muss eingeben welche Standorte verlangt sind |
| Ablauf | 1. Input des Users für Destination und Start annehmen 2. Nach möglichen Anschlüssen filtern |
| Alternativer Ablauf |  |
| Ergebnis | Es werden die möglichen Verbindungen ausgegeben |

|  |  |
| --- | --- |
| Use Case | Datum/Zeit auswählen |
| Beschreibung | Beliebige Uhrzeit/Datum für den Transport auswählen können |
| Akteur | Noah Götschi |
| Auslöser | Verlangen auf ein Online Fahrplan |
| Vorbedingungen | Es muss ein Kalender & eine Uhr vorhanden sein |
| Ablauf | 1. Klicken aufs Objekt 2. Auswahl des Datums/Uhrzeit |
| Alternativer Ablauf | - |
| Ergebnis | Es wird mit der angegebenen Zeit nachher gesucht |

|  |  |
| --- | --- |
| Use Case | Verbindungen suchen |
| Beschreibung | Mögliche Verbindungen ausgeben, vom Startpunkt aus |
| Akteur | Noah Götschi |
| Auslöser | Verlangen auf ein Online Fahrplan |
| Vorbedingungen | Der Startpunkt muss von User definiert werden |
| Ablauf | 1. Aufnahme des Startpunkts durch Input des Users 2. Nach möglichen Verbindungen filtern |
| Alternativer Ablauf | - |
| Ergebnis | Mögliche Verbindungen werden ausgegeben |

|  |  |
| --- | --- |
| Use Case | Destination/Start auswählen |
| Beschreibung | Beliebige Standorte für den Transport auswählen können |
| Akteur | Noah Götschi |
| Auslöser | Verlangen auf ein Online Fahrplan |
| Vorbedingungen | Der Akteur muss Zugriff auf eine grosse Auswahl von Standorten haben |
| Ablauf | 1. Programm öffnen 2. Start, Destination auswählen und suchen 3. Überblick vom gesuchten Fahrplan |
| Alternativer Ablauf |  |
| Ergebnis | Man kann solang möglich jede Verbindung auswählen |

# Aktivitätsdiagramm

Mit diesem Diagramm wird illustriert was im Programm geschehen sollte bei Verzweigungen, wenn es nicht das erwartete Resultat erhält.

# Programmrichtlinien

Kurze Beschreibung nach welchen Richtlinien programmiert werden soll, um die Arbeit zu vereinfachen und besseren Überblick zu behalten

* Naming Conventions: Camel Casing brauchen
* Lokal deklarieren
* Kommentare vor Methoden/allgemein beschreiben
* Sinnvolle Namen geben
* Statement Klammer immer auf einer neuen Zeile ->

{

}

# Testfälle

Titelbild Quelle: <https://worldvectorlogo.com/logo/c--4>