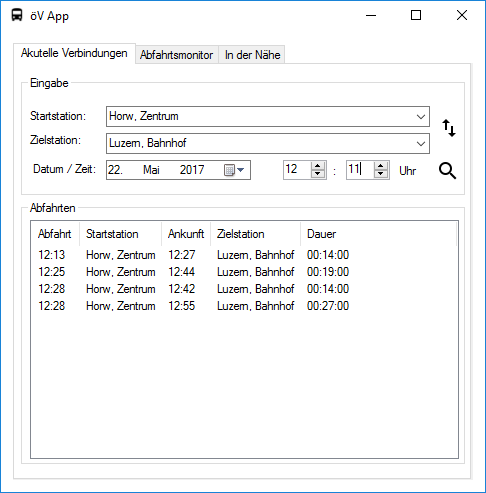
öv app



Projektarbeit M318

Mauro Küpfer

23.05.2017

Inhaltsverzeichnis

[Management Summary 4](#_Toc483309754)

[Zweck des Dokuments 5](#_Toc483309755)

[Umgesetzte Funktionen 6](#_Toc483309756)

[UML – Diagramme 7](#_Toc483309757)

[Use Cases 7](#_Toc483309758)

[Aktivitätendiagramme 7](#_Toc483309759)

[Programmierrichtlinien 8](#_Toc483309760)

[Naming Conventions 8](#_Toc483309761)

[Variablen 8](#_Toc483309762)

[Eigenschaften 8](#_Toc483309763)

[Methoden 8](#_Toc483309764)

[Klassen 8](#_Toc483309765)

[GUI-Controls 8](#_Toc483309766)

[Declaration 8](#_Toc483309767)

[Lokale Variablen 8](#_Toc483309768)

[Globale Variablen 8](#_Toc483309769)

[Comments 8](#_Toc483309770)

[Statements(new Lines, Indentation) 8](#_Toc483309771)

[Testfälle 10](#_Toc483309772)

[Testfall «A001» 10](#_Toc483309773)

[Vorbedingungen 10](#_Toc483309774)

[Testszenario 10](#_Toc483309775)

[Resultat 10](#_Toc483309776)

[Testfall «A002» 11](#_Toc483309777)

[Vorbedingungen 11](#_Toc483309778)

[Testszenario 11](#_Toc483309779)

[Resultat 11](#_Toc483309780)

[Testfall «A003» 12](#_Toc483309781)

[Vorbedingungen 12](#_Toc483309782)

[Testszenario 12](#_Toc483309783)

[Resultat 12](#_Toc483309784)

[Testfall «A005» 13](#_Toc483309785)

[Vorbedingungen 13](#_Toc483309786)

[Testszenario 13](#_Toc483309787)

[Resultat 13](#_Toc483309788)

[Testfall «A006» 14](#_Toc483309789)

[Vorbedingungen 14](#_Toc483309790)

[Testszenario 14](#_Toc483309791)

[Resultat 14](#_Toc483309792)

[Testfall «A007» 15](#_Toc483309793)

[Vorbedingungen 15](#_Toc483309794)

[Testszenario 15](#_Toc483309795)

[Resultat 15](#_Toc483309796)

[Testfall Zusatzfeature 16](#_Toc483309797)

[Vorbedingungen 16](#_Toc483309798)

[Testszenario 16](#_Toc483309799)

[Resultat 16](#_Toc483309800)

[Installationsanleitung 17](#_Toc483309801)

[Installation 17](#_Toc483309802)

[Deinstallation 17](#_Toc483309803)

[Anhang 18](#_Toc483309804)

[Mockup 18](#_Toc483309805)

# Management Summary

Im Rahmen des überbetrieblichen Kurses zum Modul 318 «Analysieren und objektbasiert programmieren mit Komponenten» ist das Programm «öV App» entstanden. Der Zweck dieses Programmes ist, aktuelle und zukünftige Verbindungen mit den öffentlichen Verkehrsmitteln der Schweiz abrufbar zu machen. Zusätzlich werden Funktionen zur Verfügung gestellt, welche Stationen in der Nähe anzeigen, damit eine möglichst bequeme Reise ohne grosse Recherche geplant und in Angriff genommen werden kann.

# Zweck des Dokuments

Dieses Dokument beinhaltet sämtliche Informationen, welche zur Analyse, Planung, Durchführung und Überprüfung des Programmes zusammengetragen wurden. Es soll dem Instruktor die Möglichkeit geben, eine genaue Bewertung abzugeben. Zusätzlich dient es als kleines Benutzerhandbuch zur Installation und zur effizienten Anwendung des Programmes.

# Umgesetzte Funktionen

Der aktuelle Release deckt folgende Anforderungen ab:

|  |  |
| --- | --- |
| Anforderung | Beschreibung |
| A001 | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. |
| 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. |
| 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. |
| 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. |
| A006 | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. |
| A007 | Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann. |
| Zusatzfeature | Die Start- und Zielstationen können gewechselt werden |

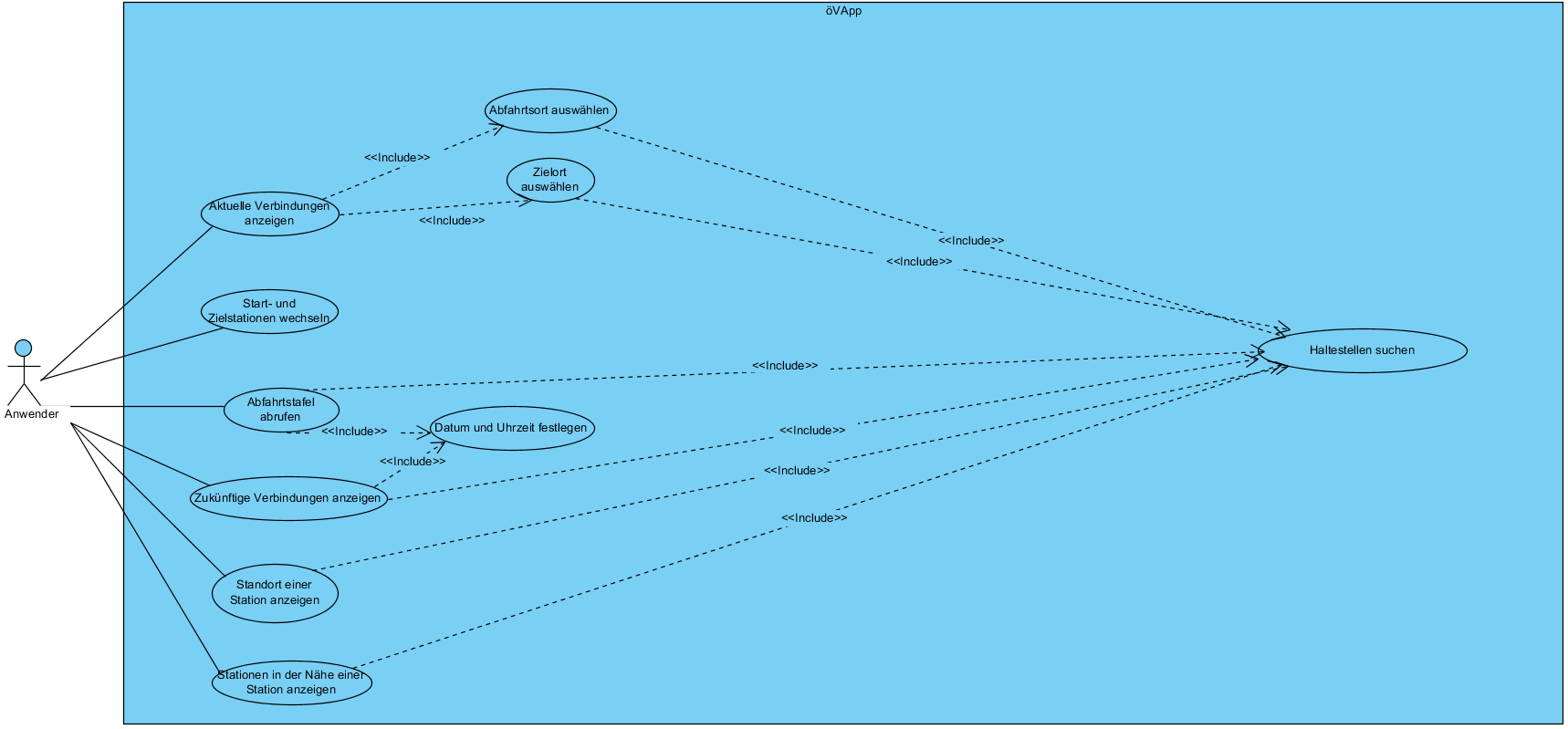
Der aktuelle Release deckt die folgenden Anforderungen nicht ab:

|  |  |  |
| --- | --- | --- |
| Anforderung | Beschreibung | Begründung |
| A004 | Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Suchresultate erscheinen, damit ich effizienter nach Stationen suchen kann. | Mit der aktuell verwendeten Control «ComboBox» ist diese Anforderung nicht abbildbar. Der Aufwand, das bestehende Konstrukt umzubauen, konnte ich nicht mit der Zeit vereinbaren, sodass die Funktion für später aufgeschoben wurde. |
| A008 | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. | Da diese Anforderung optional ist, und die Zeit nicht gereicht hat, habe ich diese Funktion für einen späteren Release aufgeschoben. |

# UML – Diagramme

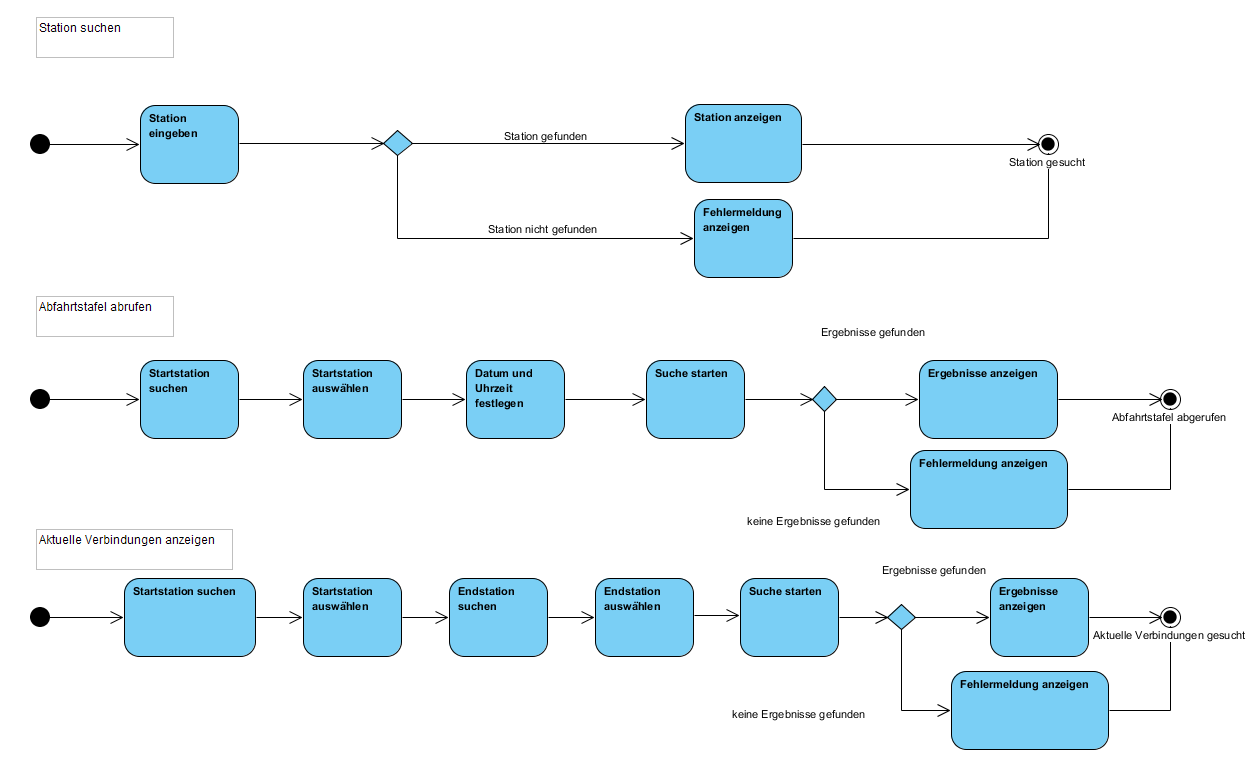
## Use Cases

Nachfolgend sind sämtliche Use Cases für die umgesetzten Funktionen im neusten Release abgebildet.



## Aktivitätendiagramme

Nachfolgend sind die drei Aktivitätsdiagramme für die Anforderungen A001 – A003 abgebildet.



# Programmierrichtlinien

An folgende Programmierrichtlinien habe ich mich während des Projektverlaufs orientiert.

## Naming Conventions

### Variablen

Die lokalen Variablen werden Camel-Case geschrieben (dieseVariable).

Globale Variablen (Members) schreibe ich zusätzlich mit dem Prefix «m\_» (m\_dieseVariable).

### Eigenschaften

Eigenschaften werden Pascal-Case geschrieben (DiesesProperty).

### Methoden

Grundsätzlich werden die Methoden Pascal-Case geschrieben (DieseMethode). Die einzige Ausnahme bezieht sich auf Events. Da alle Controls klein Beginnen, wird auch die Methode klein begonnen (btnMeinKnopf\_Click).

### Klassen

Klassen werden grundsätzlich Pascal-Case geschrieben. Wenn der Klassenname jedoch von einem GUI-Control abgeleitet wird, wird dieser jedoch Camel-Case geschrieben (frmMain).

### GUI-Controls

Der Name der GUI-Controls entspricht jeweils aus einem dreistelligen Prefix, welcher den Typ des Control abkürzt. Er wird Camel-Case geschrieben (btnMeinKnopf).

## Declaration

### Lokale Variablen

Lokale Variablen werden dann definiert, wenn diese Benötigt werden. Sie müssen nicht direkt initialisiert werden.

### Globale Variablen

Globale Variablen werden immer zu Beginn der Klasse definiert. Sie müssen nicht direkt initialisiert werden.

## Comments

Methoden, die Public sind, müssen immer am Anfang der Funktion kommentiert werden. Bei Privaten Methoden ist dies nicht zwingend und liegt im Ermessen des Entwicklers.

Innerhalb von Funktionen werden Kommentare bei Notwendigkeit eine Zeile über dem zu kommentierenden Block geschrieben.

Der Inhalt der Kommentare beschreibt, was nachfolgend passiert. Es soll jedoch nicht «übersetzt» werden, sondern Zusatzinfos für die Verständlichkeit geliefert werden.

## Statements(new Lines, Indentation)

Nach If, For, Foreach, While und Switch werden die geschweiften Klammern auf eine neue Linie gesetzt. Der Inhalt zwischen diesen Klammern ist um einen Tabstopp weiter eingerückt.

Wenn nur eine Zeile Code innerhalb diesen Klammern stehen würde, können die Klammern weggelassen werden und das Kommando eine Zeile unterhalb der Keywörter geschrieben.

# Testfälle

## Testfall «A001»

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste die Eingabezeile «Startstation» auswählen | Die Eingabezeile «Startstation» ist fokussiert.  Der Cursor blinkt darin |
| 2 | Mit der Tastatur den folgenden Text eintippen: «Horw, » | In der Eingabezeile «Startstation» steht der Text «Horw, » |
| 3 | Mit der linken Maustaste auf den **einzelnen** Pfeil nach unten, in der Eingabezeile «Startstation» klicken | Das DropDown wird aufgeklappt  Darin werden diverse Stationsvorschläge angezeigt  Die ersten drei Vorschläge sind:   1. Horw, Rank 2. Horw, Spier 3. Horw, Spitz   Im Eingabefeld steht der Text «Horw, Rank» |
| 4 | Im DropDown den Eintrag «Horw, Spitz» mit der linken Maustaste auswählen | Das DropDown wird zugeklappt  Im Eingabefeld steht der Text «Horw, Spitz» |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

## Testfall «A002»

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste die Eingabezeile «Startstation» auswählen | Die Eingabezeile «Startstation» ist fokussiert.  Der Cursor blinkt darin |
| 2 | Mit der Tastatur den folgenden Text eintippen: «Horw» | In der Eingabezeile «Startstation» steht der Text «Horw» |
| 3 | Mit der Tabulator-Taste zum Eingabefeld «Zielstation» wechseln | Die Eingabezeile «Zielstation» ist fokussiert.  Der Cursor blinkt darin |
| 4 | Mit der Tastatur den folgenden Text eintippen: «Luzern» | In der Eingabezeile «Zielstation» steht der Text «Luzern» |
| 5 | Die Entertaste drücken | In der Anzeige «Abfahrten» werden die fünf aktuellen Verbindungen angezeigt  In der Eingabezeile «Startstation» steht «Horw, Zentrum»  In der Eingabezeile «Zielstation» steht «Luzern, Bahnhof» |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

## Testfall «A003»

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste den Tab «Abfahrtsmonitor» auswählen | Der Tab «Abfahrtsmonitor» ist selektiert |
| 2 | Die Eingabezeile «Startstation» mit der linken Maustaste auswählen | Der Cursor blinkt in der Eingabezeile «Startstation» |
| 3 | Mit der Tastatur den folgenden Text eintippen: «Horw, Zentrum» | In der Eingabezeile «Startstation» steht der Text «Horw, Zentrum» |
| 4 | Die Entertaste drücken | In der Anzeige «Abfahrten» werden die nächsten Abfahrten angezeigt |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

## Testfall «A005»

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Den Testfall «A002» durchführen | Siehe Testfall |
| 2 | In die Nummernfelder für die Zeit folgenden Wert eingeben: 21:00 | Die Eingabe wurde akzeptiert |
| 3 | Die Entertaste drücken | In der Anzeige «Abfahrten» werden fünf Verbindungen nach 21:00 Uhr angezeigt |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

## Testfall «A006»

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung. Es ist ein Webbrowser installiert.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Den Testfall «A001» durchführen | Siehe Testfall |
| 2 | Auf die Beschriftung «Startstation:» mit der linken Maustaste klicken | Google Maps wird im Webbrowser geöffnet  Ein Pin ist in der Nähe des Schulhauses «Spitz» gesetzt |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

## Testfall «A007»

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Mit der linken Maustaste auf den Tab «In der Nähe» anklicken | Der Tab «In der Nähe» wird geöffnet |
| 2 | In das Eingabefeld «Suchstation» folgenden Wert eingeben: «Horw, Zentrum» | Der Text «Horw, Zentrum» steht im Eingabefeld «Suchstation» |
| 5 | Die Entertaste drücken | In der Anzeige «Stationen in der Nähe» werden die Stationen bis 1km Distanz angezeigt |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

## Testfall Zusatzfeature

### Vorbedingungen

Das Programm ist gestartet. Der Tab «Aktuelle Verbindungen» ist aktiv. Es besteht eine Internetverbindung.

### Testszenario

|  |  |  |
| --- | --- | --- |
| Schritt | Aktivität | Erwartetes Resultat |
| 1 | Den Testfall «A002» druchführen | Siehe Testfall |
| 2 | Mit der linken Maustaste auf den Wechsel-Icon klicken (Zwei Pfeile) | Der Inhalt der beiden Eingabefelder wurde gewechselt |

### Resultat

|  |  |
| --- | --- |
| Test bestanden | Test nicht bestanden |
|  |  |

# Installationsanleitung

## Installation

Für die Installation der öV App sind folgende Schritte notwendig:

|  |  |
| --- | --- |
| Schritt | Beschreibung |
| 1 | Das folgende Repos klonen:  <https://github.com/deeznutshaha/modul-318-student/> |
| 2 | Im Root-Verzeichnis liegt die aktuelle Setup-Datei mit der Dateiendung «.msi» |
| 3 | Diese Datei muss mit lokalen Administrationsrechten gestartet werden |
| 4 | Danach muss durch den Wizzard navigiert werden. Dieser wird alle notwendigen Informationen abfragen. |
| 5 | Die öV App wurde installiert und kann nun auf dem Desktop und im Startmenü von Windows gefunden werden. |

## Deinstallation

Für die Deinstallation der öV App sind folgende Schritte notwendig:

|  |  |
| --- | --- |
| Schritt | Beschreibung |
| 1 | Im Windows nach «Programme und Features» suchen und dieses Fenster öffnen |
| 2 | Die öV App suchen |
| 3 | Rechtsklick auf die öV App 🡪 Deinstallieren auswählen |
| 4 | Allenfalls die Deinstallationsnachfrage bestätigen |
| 5 | Die öV App wurde deinstalliert |

# Anhang

## Mockup

Dies ist der erste Layoutentwurf meiner öV App, welcher sämtliche Funktionen bereits abdeckt.

