

Meine App

Timetable

Version 1.0.0, 4. Juli 2022 | Nathan Richard & Veselin Stanchev

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc107914558)

[2 User Stories 3](#_Toc107914559)

[3 App-Struktur 3](#_Toc107914560)

[3](#_Toc107914561)

[4 Mockups 4](#_Toc107914562)

[5 Testing 7](#_Toc107914563)

[5.1 Manuelle UI-Tests 7](#_Toc107914564)

[5.2 Testauswertung 8](#_Toc107914565)

[6 Fazit 9](#_Toc107914566)

# Abstract (Kurzbeschreibung)

An dieser Stelle kommt eine kurze Einführung, die in 5-10 Sätzen das Projekt beschreibt.

# User Stories

Als Benutzer möchte ich eine Zugverbindung auswählen können, um genauere Informationen anzeigen zu lassen.

Als Benutzer möchte ich einen Bahnhof eingeben können, um die nächsten Zugverbindungen ab diesen Bahnhof zu sehen.

Als Benutzer möchte ich die nächsten Zugverbindungen als chronologisch geordnete Liste sehen, um mir einen schnellen und klaren Überblick zu verschaffen.

Als Benutzer möchte ich aufs Fragezeichen klicken können, um Informationen und Hilfe zu App zu erhalten.

Als Benutzer möchte ich bei einem falsch eingegebenen Bahnhof oder keiner Internetverbindung eine Fehlermeldung erhalten, um genau zu wissen was das Problem ist.

Als Benutzer möchte ich von der Listenansicht sowie von der Einzelansicht auf dem Up button klicken, um zur letzten Seite zu gelangen.

Als Benutzer möchte ich das Datum und die Zeit meiner Ausgabe verändern können, um die Ausgabe meiner Bedürfnisse anzupassen.

# App-Struktur

## 

# Mockups



Die Reihenfolge der Ansichten entspricht der Reihenfolge nach dem Uhrzeigersinn, wenn oben rechts gestartet wird.

1. **Startactivity**Innerhalb der StartActivity wird dem Benutzer zentral ein Eingabefeld, indem er einem Bahnhof suchen kann, präsentiert. Oberhalb des Formulars wird ein Willkommenstext ausgegeben, der den Benutzer begrüssen soll. Unterhalb des Willkommen Textes haben wir einen kleinen Text, der dem Benutzer erklärt, wie er die Applikation benutzen soll.   
   Unter dem Eingabefeld befindet sich einen Button mit der Beschriftung «GO». Dieser leitet den Benutzer auf die Listenansicht / den Abfahrtsplan. Dieser ist in der Farbe Violet damit man ihn sieht und damit er zur Applikation passt.
2. **Listenansicht**In der Listenansicht wird dem Benutzer eine Liste mit den nächsten Verbindungen ab den ausgewählten Bahnhof angezeigt. Jede Verbindung bedeckt eine Reihe in der Liste. In jede Reihe steht die Abfahrtszeit, der Endbahnhof und das Gleis. Oberhalb der Liste haben wir noch den gesuchten Bahnhof als Text ausgegeben. Dabei haben wir in der oberen linken Ecke noch einen Pfeil, welcher den Benutzer zurück zur StartActivity führt.
3. **Fehleransicht**

Falls beim Laden der Listenansicht ein Fehler auftritt. Z.B. durch keine Internetverbindung oder nicht vorhandenem Bahnhof, so wird diese Angezeigt. Die Ansicht besteht eigentlich nur aus einem Dialog, in welchem als Überschrift Fehler steht und dann als Text der Grund für den Fehler. Unter rechts kann man auf okay drücken oder einfach neben den Dialog und kommt danach auf die leere Listenansicht.

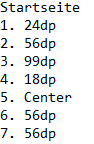
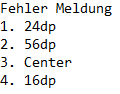
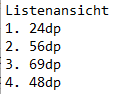
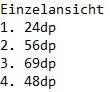
1. **Einzelansicht**

In der Einzelansicht werden wir wieder eine Liste sehen. Diese Liste ist die Detailansicht der Ausgewählten Verbindung aus der Listenansicht. Die Einzelansicht beinhaltet das Gleis, an dem man ankommt und abfahrt, die Abfahrtszeit sowie die Ankunftszeit und der Endbahnhof. Oberhalb von dieser Liste befindet sich der Titel der Ansicht, der einem Aussagt, welche Verbindung man in der Listenansicht gewählt hat. In der oberen linken Ecke haben wir einen Pfeil, der einem zur Listenansicht zurückführt.

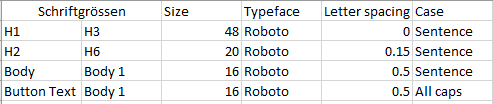
1. Verbindungen
2. Bei einer Betätigung des Buttons wird nach dem eingegebenen Bahnhof gesucht. Wenn der Bahnhof gefunden wurde, geht’s zur Listenansicht, wenn nicht wird eine Fehlermeldung angezeigt, der den Benutzer bittet einen validen Bahnhof anzugeben.
3. Mit der Betätigung des Pfeiles wird der Benutzer von der Listenansicht zur StartActivity zurückgeführt.
4. Mit einem Klick auf einer Reihe in der Listenansicht wird die Einzelansicht von dieser Reihe beziehungsweise Verbindung aufgerufen.
5. Mit dem Pfeilbutton wird der Benutzer zurück zur Listenansicht zurückgebracht.
6. Wie bei Punkt eins schon erwähnt wird der Benutzer bei einem Fehler eine Fehlermeldung angezeigt, welche ihn dazu bittet, einen richtigen Bahnhof auszuwählen.
7. Mit einem Klick auf den Button wird der Benutzer zurück auf die StartActivity gebracht.
8. Abstände & Schrift

Die Abstände der verschiedenen Ansichten sind orange angezeigt und nummeriert. Der Seitenabstand beträgt auf beiden Seiten 16dp.

Hier noch die verschiedenen Abstände:



Hier sind auch die Schriftgrössen, die wir brauchen wollen. In der ersten Spalte sind unsere Bezeichnungen, in der Zweiten die, die uns vorgegeben wurden, die dritte Spalte ist die Grösse, die Vierte ist die Schriftart, in der Fünften ist der Buchstabenabstand und in der letzen Spalte haben wir die Schreibweise.



1. Farbkonzept

|  |  |  |  |
| --- | --- | --- | --- |
| Anwendung | HEX-Code | Beschreibung | Farbe |
| Titel | #323437 | Sticht aus dem Gelb heraus und sieht gut aus. |  |
| Button | #E2B714 | Sticht aus dem Weiss hinaus und ist gut ersichtlich. |  |
| Statusbar | #E2B714 | Sieht gut aus und durch die dunklere Farbe nicht augenreizend für den Benutzer. |  |
| Navigationsbar | #E2B714 | Sieht gut aus und durch die dunklere Farbe nicht augenreizend für den Benutzer. Dabei ist es einheitlich mit der Statusbar. |  |
| Upbutton | #FFFFFF | Ist in der gelben Navigationsbar gut ersichtlich. |  |
| Button Text | #323437 | Sticht aus dem Gelb hinaus. |  |
| Text | #323437 | Sieht man gut auf der weissen Seite. |  |

1. Technische Realisierung

Beschreibt hier, wie ihr eure komplexe Komponente technisch umgesetzt habt. Zur Darstellung der technischen Umsetzung wird ein UML-Diagramm empfohlen, welches zusätzlich in Textform beschrieben wird. Erklärt kurz die wichtigsten Klassen und Methoden und deren Zusammenspiel. Eine Fachperson, welche dieses Kapitel liest, sollte schnell nachvollziehen können, wie die externe Komponente realisiert wurde.

# Testing

## Manuelle UI-Tests

In diesem Kapitel definiert ihr die Tests die Ihr macht.   
Es müssen minimal 5 Unit-Tests, 3 automatische UI-Tests (Espresso) und 2 manuelle UI-Tests gemacht werden. Auf die Unit-Tests und die automatischen UI-Tests soll hier verwiesen werden, die manuellen UI-Tests hier definiert werden.

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | Testfallnummer (ST = Systemtest) |
| Anforderungen | Welche Anforderungen werden durch diesen Testfall abgedeckt. (User Stories) |
| Vorbedingungen | Was muss gegeben sein, damit dieser Test durchgeführt werden kann? |
| Ablauf | Welche Schritte werden bei der Durchführung des Tests durchlaufen? |
| Erwartetes Resultat | Was sollte nun passiert sein? |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | ST-01 |
| Anforderungen | US-01; US-03 |
| Vorbedingungen | In der Datenbank existiert ein Benutzer, welcher gesperrt ist. |
| Ablauf | 1. Die App wird gestartet damit das Login-Formular erscheint 2. Der korrekte Benutzername sowie das korrekte Passwort werden eingegeben. 3. Der Button mit dem Label „Login“ wird geklickt |
| Erwartetes Resultat | Ein Toast mit dem Text «Login erfolgreich» wird angezeigt. Die App wechselt zu der Ansicht mit den favorisierten Schwimmbäder |

## Testauswertung

Zusammenfassung aller durchgeführten Tests. Nur fehlgeschlagene Tests und Tests mit Bemerkungen müssen in der folgenden Tabelle aufgelistet werden.

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| ST-01 | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| … | … | … |
|  |  |  |

# Fazit

Hier kommt eure Reflexion zum Projekt.

* Was lief gut/schlecht?
* Wie seid ihr mit dem Endergebnis zufrieden?
* Was habt ihr gelernt?
* War alles vorhanden oder was fehlte noch?
* Usw.