

Meine App

Timetable

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

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc7422480)

[2 Konkurrenzanalyse 3](#_Toc7422481)

[3 User Stories 4](#_Toc7422482)

[4 Mockups 5](#_Toc7422483)

[5 Technische Realisierung 6](#_Toc7422484)

[6 Testing 7](#_Toc7422485)

[6.1 Manuelle UI-Tests 7](#_Toc7422486)

[6.2 Testauswertung 8](#_Toc7422487)

[7 Fazit 9](#_Toc7422488)

# 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.

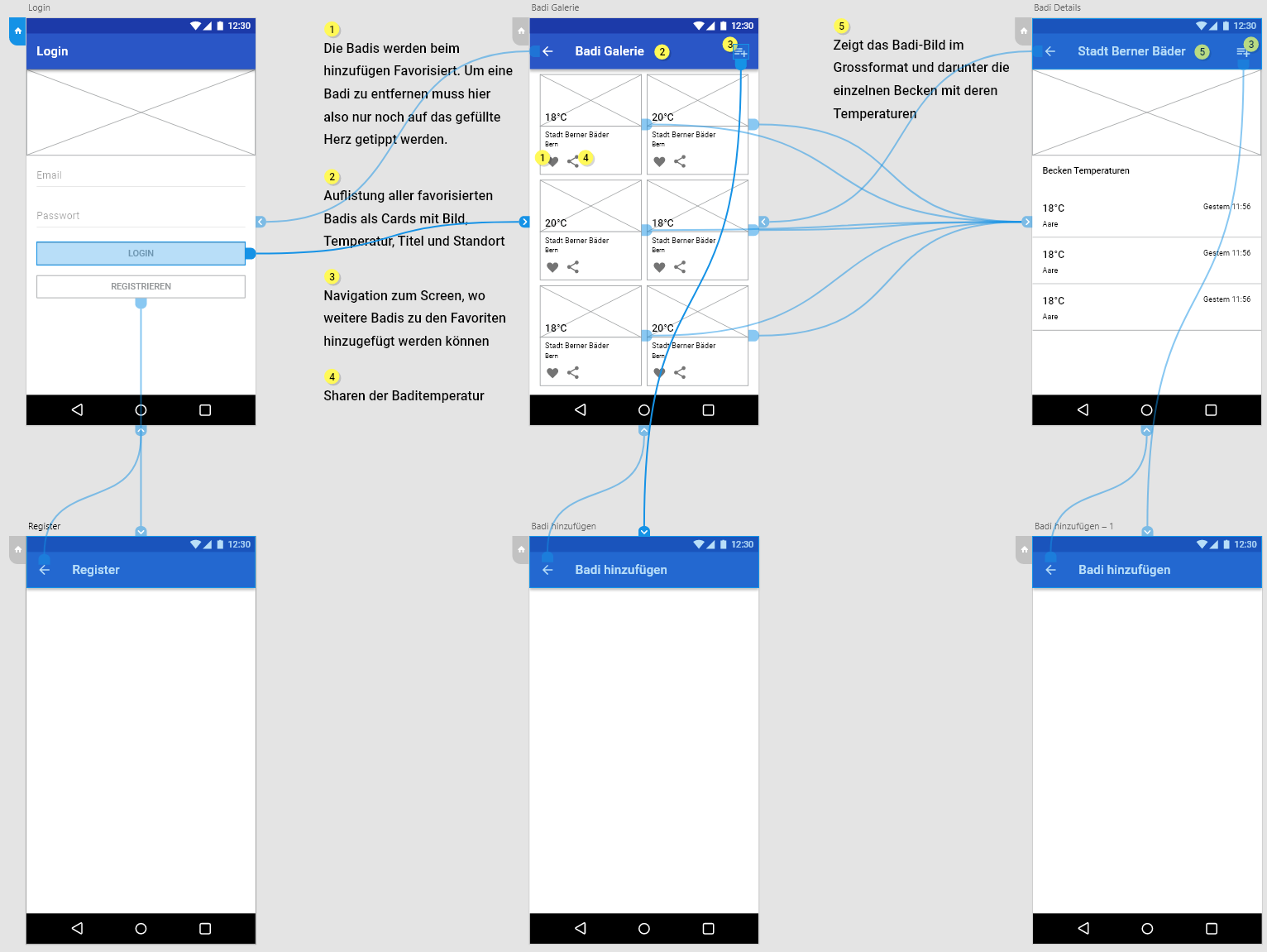
Als Benutzer möchte ich bei einem falsch eingegebenen Bahnhof oder keiner Internetverbindung eine Fehlermeldung erhalten, um genau zu wissen was das Problem ist.

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



1. **Startactivity**Innerhalb der StartActivity wird dem Benutzer zentral ein Login-Formular präsentiert. Oberhalb des Formulars wird das Logo der App platziert. Die Eingabefelder sollten möglichst in den oberen zwei Drittel des Bildschirms platziert werden damit die Tastatur diese nicht überdeckt.  
   Unter den Eingabefelder sind zwei Buttons platziert, «Login» und «Registrieren». Dabei soll der Login-Button farblich hervorgehoben werden da dieser öfters benutzt wird. Der «Registrieren»-Button wird im Normalfall einmal benutzt deshalb soll dieser neutral oder sogar weniger prominent dargestellt werden.
2. **Badi-Galerie**In der «Badi-Galerie»-Ansicht wird dem Benutzer ein Grid mit den favorisierten Schwimmbäder angezeigt. Jedes einzelne Schwimmbad wird mit einem Bild präsentiert. In der in der unteren Ecke jedes Bildes wird die aktuelle Temperatur angezeigt. Darunter den Namen der Badi und der Ort. In der Auflistung muss auf und ab navigiert werden können da man mehr Schwimmbäder hinzufügen kann als auf dem Display Platz haben. Ein Floating-Action-Button mit einem Plus-Icon ist unten rechts am Screen platziert um neue Schwimmbäder in die eigene Liste aufzunehmen. In der Actionbar am oberen Rand wird rechts ein Kontextmenü platziert welches mit dem entsprechenden Button geöffnet werden kann. Darin sind die Punkte Einstellungen, Hilfe und Logout zu finden.
3. **Badidetails**…
4. **Badiauswahl**…
5. **Registration**…

# 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.