David Rüegger & Jonathan Fritz

Pendenz-Verwaltung

Android App

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc72415098)

[1.1 Ziele 2](#_Toc72415099)

[1.2 GitHub Struktur 2](#_Toc72415100)

[2 Planung 3](#_Toc72415101)

[2.1 Mockups 4](#_Toc72415102)

[3 Technischer Teil 6](#_Toc72415103)

[3.1 Spezifikationen 6](#_Toc72415104)

[4 Testing 7](#_Toc72415105)

[4.1 Manuelle UI-Tests 7](#_Toc72415106)

[4.2 Testauswertung 8](#_Toc72415107)

[5 Fazit 9](#_Toc72415108)

# Abstract (Kurzbeschreibung)

Ziel dieses Projekts ist es, eine Android-App für Smartphones zu erstellen, über welche ein Lernender Pendenzen erstellen kann und dann in einer Liste (oder ähnlichem) alle erfassten Pendenzen angezeigt werden. Es gibt Pflichtanforderungen und optionale Anforderungen. Zu den Pflichtanforderungen gehört der ganze Ablauf mit dem Erstellen der Pendenzen und das Anzeigen. Zusätzlich soll auch die Dringlichkeit jeder Pendenz angezeigt werden. Alle Pendenzen müssen ausserdem in eine lokale Datenbank gespeichert werden. Die App wird auf Deutsch sein. Zu den optionalen Anforderungen gehört eine Detailansicht von den einzelnen Pendenzen.

## Ziele

Unser Ziel ist es, dass unsere App dem Mockup entspricht und wir alle Designs übernehmen können, ohne etwas am Mockup zu verändern. Zusätzlich sollte der Code sauber sein und die ganze Filestruktur eingehalten werden. Wir wollen ausserdem, dass wir die optionalen Anforderungen ebenfalls implementieren.

Die Dokumentation soll sehr detailreich sein und alles beinhalten. Mit dem ganzen wollen wir am besten bereits ein oder zwei Tage früher sein, damit wir nochmals alles durchgehen können und additiv noch Zusatzfeautures hinzufügen können.

## GitHub Struktur

Unser Git-Repository ist unter folgendem Link   
erreichbar: <https://github.com/chas1e/uek335_PendenzVerwaltung.git>.

Der Aufbau ist wie folgt:

Dokumentation:

<https://github.com/chas1e/uek335_PendenzVerwaltung/blob/main/documentation/Dokumetation_PendenzVerwaltung_david_jonathan.docx>

Projekt:

<https://github.com/chas1e/uek335_PendenzVerwaltung/tree/main/project>

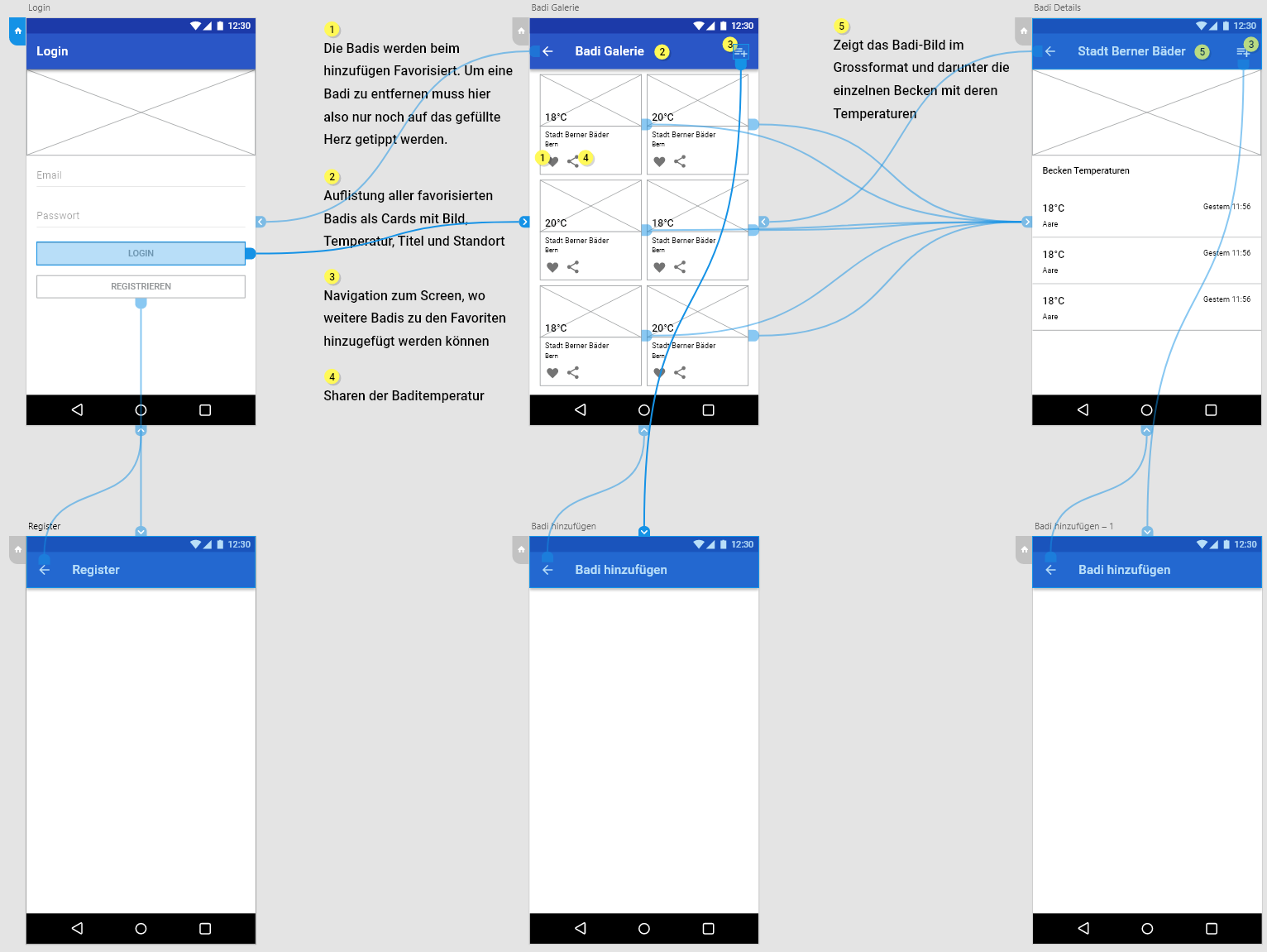
Im README File ist die Struktur nochmals erklärt:

<https://github.com/chas1e/uek335_PendenzVerwaltung/blob/main/README.md>

# Planung

|  |  |  |
| --- | --- | --- |
| Tag | David Rüegger | Jonathan Fritz |
| 19.05.21 | Mit Mockup begonnen | Mit Dokumentation begonnen |
| 20.05.21 |  |  |
| 21.05.21 |  |  |
| 26.05.21 |  |  |
| 27.05.21 |  |  |
| 28.05.21 |  |  |

## 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**…

# Technischer Teil

## Spezifikationen

* 1. Funktionale Anforderungen
  2. Benutzereingaben

**Titel:** Pflichtfeld, max. 50 Zeichen

**Beschreibung:** Optional, max. 500 Zeichen

**Zu erledigen bis:** Optional, Datumsfeld, Datepicker, Format dd.mm.yyyy

**Dringlichkeit:** Pflichtfeld, Dropdown, Werte Niedrig, Normal und Hoch (Normal default selektiert)

**Überprüfung aller Eingaben bevor speichern.**

# Testing

## Manuelle UI-Tests

In diesem Kapitel definiert ihr die Tests die Ihr macht.

|  |  |
| --- | --- |
| Abschnitt | Innhalt |
| 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 | Innhalt |
| 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.