David Rüegger & Jonathan Fritz

Pendenz-Verwaltung

Android App

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc72485438)

[1.1 Ziele 2](#_Toc72485439)

[1.2 GitHub Struktur 2](#_Toc72485440)

[2 Planung 3](#_Toc72485441)

[2.1 Arbeitsaufteilung 3](#_Toc72485442)

[2.2 Mockups 4](#_Toc72485443)

[3 Technische Realisierung 6](#_Toc72485444)

[3.1 Spezifikationen 6](#_Toc72485445)

[3.1.1 Benutzereingaben 6](#_Toc72485446)

[3.1.2 Anzeige der Pendenzen 6](#_Toc72485447)

[3.2 UML-Diagramm 7](#_Toc72485448)

[3.2.1 Beschreibung 7](#_Toc72485449)

[4 Testing 8](#_Toc72485450)

[4.1 Manuelle UI-Tests 8](#_Toc72485451)

[4.2 Testauswertung 9](#_Toc72485452)

[5 Fazit 10](#_Toc72485453)

# 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 Zusatzfeatures 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>

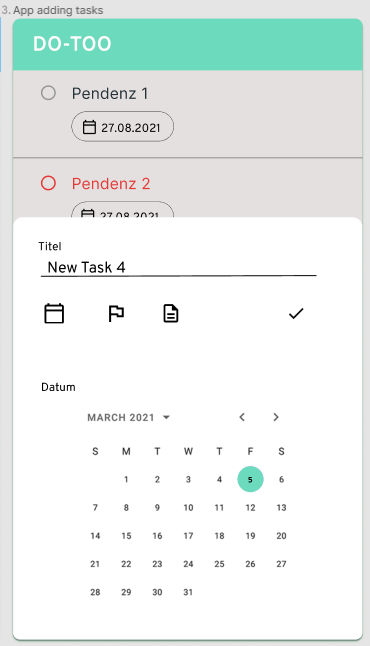
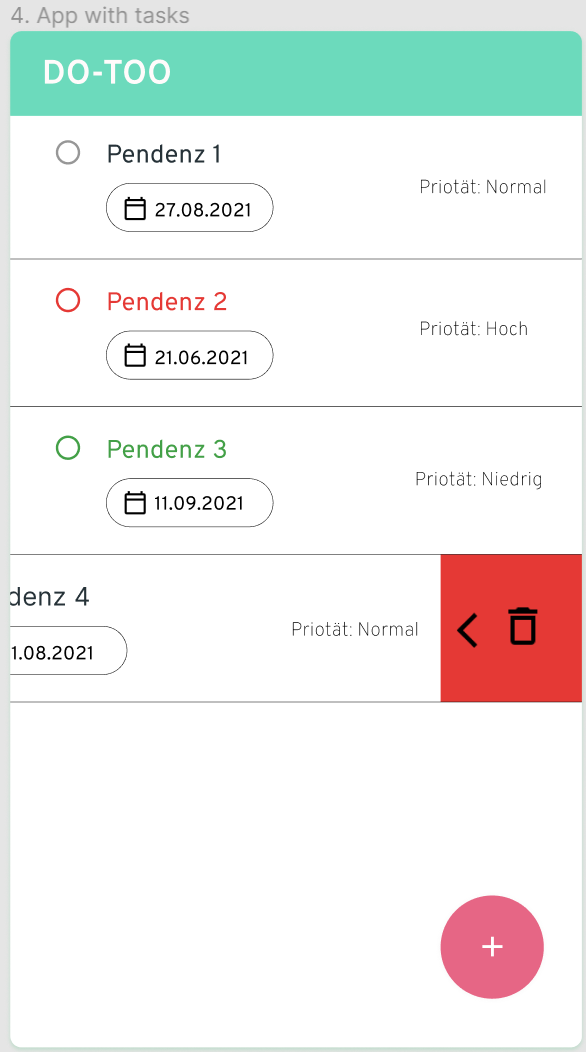
# Planung

## Arbeitsaufteilung

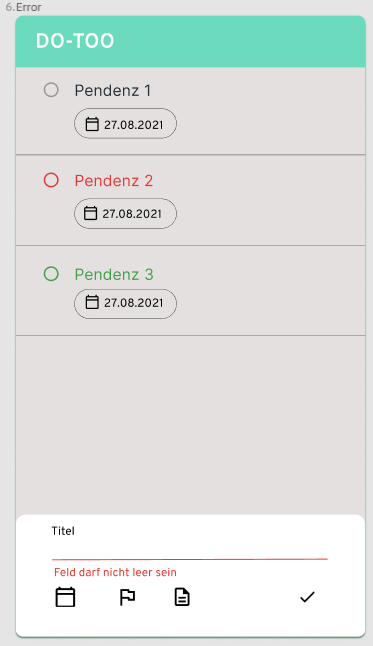
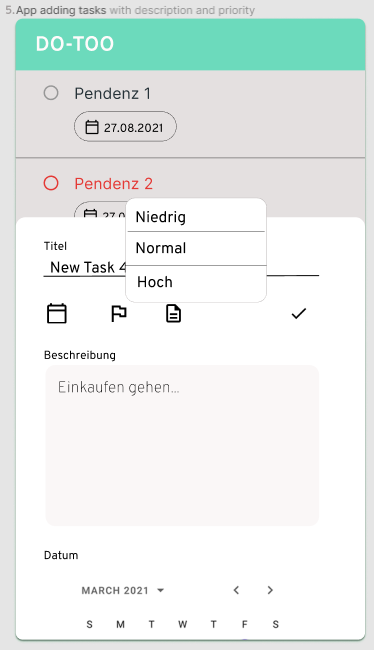
|  |  |  |
| --- | --- | --- |
| Tag | David Rüegger | Jonathan Fritz |
| 19.05.21 | Mit Mockup begonnen | Mit Dokumentation begonnen |
| 20.05.21 | Mocku |  |
| 21.05.21 |  |  |
| 26.05.21 |  |  |
| 27.05.21 |  |  |
| 28.05.21 |  |  |

## Mockups

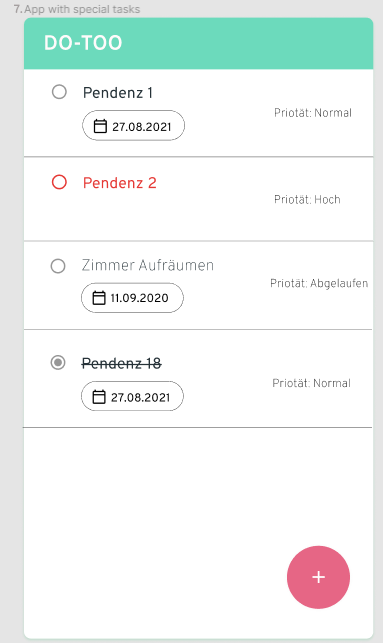
1. Dies ist die Startseite unserer App. Wenn man das erste Mal die App öffnet, wird man von einer Nachricht erwartet, welche besagt das noch keine Pendenzen vorhanden sind. Ganz oben links, sieht man den Schriftzug «DO-TOO». Das ist unser App Name und ist von dem Microsoft Programm To-Do abgeleitet. Es soll also bei uns heissen auch noch tun oder mach auch. Unten rechts sieht man einen runden Button mit einem Plus, welcher zum Erstellen einer Pendenz gedacht ist. Die Farbauswahl besteht aus einem hellen Grünton und einem Rot-Pinkt Ton, welche komplementär zueinander stehen.
2. Bei der zweiten Ansicht sieht man bereits erstellte Pendenzen und unten ein Menu zum Erstellen einer Pendenz, welches sich beim Klicken auf das Plus öffnet. Jetzt könnte man theoretisch einen Titel eingeben und die Pendenz über den Hacken speichern, da der Titel das einzige Pflichtfeld ist. Man sieht hier noch ein Kalender, ein Flaggen und ein Dokumenten Logo, welches ebenfalls weitere Eingabemöglichkeiten beinhalten, die aber alle optional sind oder einen Default Wert haben.



1. Man hat jetzt zusätzlich die Möglichkeit das geöffnete Fenster beim Betätigen des runden plus Knopfes zu vergrössern, indem man es am oberen Rand hoch zieht. Beim Hochziehen kommt die Datumsauswahl zum Vorschein, welche ebenfalls beim Betätigen durch das Kalender Icon ganz links erscheint.
2. Wenn man eine Pendenz löschen will, kann man dies durch ein Einfaches nach links wischen auf der gewünschten Pendenz machen. Es erscheint dann ein rotes Icon mit einem Abfalleimer drauf.



1. Die Priorität einer Pendenz kann man über das Flaggen-Icon festlegen. Standardmässig ist diese auf «Normal» gestellt. Normale Pendenzen werden in der Listenansicht grau gekennzeichnet. Stellt man die Pendenz auf Niedrig ist die Pendenz grün gekennzeichnet und ist die Priorität hoch ist sie rot.
2. Da der Titel das einzige Pflichtfeld ist, kommt dort eine rote Fehlermeldung, wenn man die Pendenz ohne einen Titel speichern will. Eine Fehlermeldung kommt ausserdem ebenfalls wenn man ein Datum aus der Vergangenheit auswählt.



1. Wenn eine Pendenz abgelaufen ist, wird die Priorität auf abgelaufen gesetzt und der Titel ist etwas blasser. Will man eine Pendenz als erledigt markieren, klickt man auf den Punkt neben dem Titel und die Pendenz wird durchgestrichen. Gibt man der Pendenz kein Fälligkeitsdatum wird dort nichts angezeigt.

# Technische Realisierung

## Spezifikationen

* 1. Pflicht-Anforderungen

### 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), (statischer Spinner und Einträge aus Array-XML)

**Überprüfung aller Eingaben bevor speichern. (Toast)**

* Datum darf nicht in Vergangenheit sein
* Pflichtfelder dürfen nicht leer sein

Wenn Pendenz abgelaufen ist, wird sie gekennzeichnet.

**Falls alles korrekt in SQLite-DB speichern.**

**Die App ist nur auf Deutsch verfügbar.**

### Anzeige der Pendenzen

Titel, Zu erledigen bis und Dringlichkeit werden in einer Liste angezeigt. Wenn «Zu erledigen bis» nicht

angegeben wurde, wird in der Liste die Pendenz ohne Datum angezeigt.

* 1. Optionale Anforderungen

Details einer Pendenz wird angezeigt, wenn auf diese ausgewählt wird. Titel, die Beschreibung, zu erledigen bis und die Dringlichkeit werden dann angezeigt.

Optional werden wir noch eine die Möglichkeit bieten, eine Pendenz zu editieren und zu löschen.

## UML-Diagramm

### Beschreibung

# 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.