

Meine App

NOA «Pendenz-Verwaltung»

Version 1.0.0, 17.06.2020 | Jimmy Lee Sager | Joël Oppliger

Inhalt

[1 Abstract (Kurzbeschreibung) 2](#_Toc43827263)

[2 Mockups 3](#_Toc43827264)

[3 Technische Realisierung 4](#_Toc43827265)

[3.1 UML 4](#_Toc43827266)

[3.2 Programmierung 4](#_Toc43827267)

[4 Testing 5](#_Toc43827268)

[4.1 Manuelle UI-Tests 5](#_Toc43827269)

[4.2 Testauswertung 6](#_Toc43827270)

[5 Fazit 7](#_Toc43827271)

# Abstract (Kurzbeschreibung)

Unsere Aufgabe war es, eine Pendenz-Verwaltung zu entwerfen, in der man nicht nur die Pendenzen anschauen kann. Die verschiedenen Pendenzen haben verschiedenen Wichtigkeiten, ein «Ablaufdatum», und eine Beschreibung. Der Benutzer kann auch neue Pendenzen erstellen und alle Attribute dort mitgeben. Um eine neue Pendenz zu erstellen, kann man einfach auf einen offensichtlichen Button klicken, sodass der Benutzer nicht verwirrt wird.

# Mockups

1. **Startactivity / Pendenzen anzeigen**

**Der User sieht hier seine erstellten Pendenzen und kann diese als Erledigt oder Offen einstellen. Jede Pendenz hat eine Wichtigkeitsstufe, die in drei Farben (grün, gelb und rot) aufgeteilt sind. Unten rechts kann der User auf den Butten klicken, um eine neue Pendenz zu erfassen.**

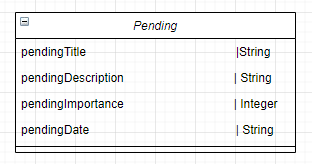
1. **Pendenzen erfassen**

**Hier kann der User eine Pendenz erfassen. Dort kann er einen Titel, Beschreibung, Ablaufdatum und eine Wichtigkeit setzten. Sobald er fertig ist, kann der User die Pendenz mit einem Buttonklick speichern. Während dem erstellen kann er auch mit der Rücktaste oben Links oder der Abbruchtaste oben Rechts den Vorgang abbrechen.**

1. **Pendenzen erfassen Kalender  
   Beim Datumsfeld wird ein Android Spezifisches Feld geöffnet, wo der User sein gewünschtes Datum festlegen kann.**

# Technische Realisierung

## UML



Da wir nur eine Tabelle erstellen musste, ist das UML-Diagramm auch nicht wirklich gross geraten. Wie man direkt erkennen kann, verwenden wir vier Datensätze in unserer «Pending» Tabelle. Der PendingTitle wird als einfacher String abgespeichert und ist genau gleich aufgebaut, wie die pendingDescription Spalte. Die pendingImportance wird als Integer Wert abgespeichert, sodass man zwischen 0, 1 und 2 unterscheiden kann. Das pendingDate ist wiederum ein String, da man so die Daten besser verwalten kann.

## Programmierung

Die Mobile App wurde mit Java und XML realisiert. Die Datenbank wurde mit sqlite erstellt. Es gibt insgesamt vier verschiedene Java Class files. Zwei davon sind die Java Dateien für die Views und der Rest ist für die Persistenz.

# 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 | Die App ist auf dem Gerät installiert |
| Ablauf | * Die App wird gestartet damit die Pendenzübersicht erscheint * Der Runde Button in der rechten unteren wird gedrückt. * Einen Titel, Beschreibung, Datum sowie Priorität werden ausgewählt beziehungsweise eingetragen * Der Button mit dem Label „Fertig“ wird geklickt |
| Erwartetes Resultat | Der Benutzer wird auf die Pendenzübersicht weitergeleitet. Dort sieht er nun eine neue Pendenz. |

|  |  |
| --- | --- |
| Abschnitt | Innhalt |
| ID | ST-02 |
| Anforderungen | US-01; US-03 |
| Vorbedingungen | Die App ist auf dem Gerät installiert |
| Ablauf | 1. Die App wird gestartet damit die Pendenzübersicht erscheint 2. Der Runde Button in der rechten unteren wird gedrückt. 3. Einen Titel, Beschreibung, Datum sowie Priorität werden ausgewählt beziehungsweise eingetragen 4. Der Button mit dem Label „Fertig“ wird geklickt 5. Der Benutzer erstellt eine neue Pendenz für jede Priorität. (Hoch, Mittel, Tief) |
| Erwartetes Resultat | Das Bild Links ist Bei Hoher Priorität Rot, bei Mittlerer Priorität Gelb und bei tiefer Priorität grün. |

|  |  |
| --- | --- |
| Abschnitt | Innhalt |
| ID | ST-03 |
| Anforderungen | US-01; US-03 |
| Vorbedingungen | Die App ist auf dem Gerät installiert |
| Ablauf | 1. Die App wird gestartet damit die Pendenzübersicht erscheint 2. Der Runde Button in der rechten unteren wird gedrückt. 3. Der Button mit dem Label „Fertig“ wird geklickt |
| Erwartetes Resultat | Ein Toast erscheint mit der Meldung “Please fill out all the fields” |

## 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. |
| ST-02 | Ja | Der Testfall ist gescheitert. Die Bilder sind überall nur grün. |
| ST-03 | Ja | Der Testfall war erfolgreich. Dem Tester ist aufgefallen das der Toast auf Englisch ist und nicht wie der Rest der Applikation auf Deutsch |

# Fazit

Unser Projekt hatte am Anfang kleinere Probleme, da Jimmy nicht mit dem Android Studio Programm arbeiten konnte. Sobald wir aber alle Probleme gelöst hatten, konnten wir voll im Team arbeiten. Der Programmieranfang viel uns etwas schwer, da wir schon seit längerem nicht mit Java gearbeitet haben. Jedoch konnten wir uns nach einigen Stunden uns wieder voll einleben und ohne Probleme unseren Auftrag beginnen. Beim Entwerfen des Mockups entstanden gar keine Probleme und wir konnten schnell alles fertigstellen. Es ergaben sich nur kleine Verbesserungsvorschläge bei der Abgabe.