

# 1. Kurzbeschreibung

Im Rahmen des Softwarepraktikum erstellen wir eine Android-App. Das Softwarepraktikum ist ein Praxismodul an der Universität Marburg.

Die App soll auf allen Android Versionen  $\geq 8.0$  (abweichend von ursprünglich 7.1) laufen.

Sie soll nützliche Funktionen für den Unialltag bieten. Dazu gehört z.B. ein Zugriff auf Informationen und Funktionalitäten von Ilias und Marvin aber auch Raumpläne und die aktuelle Speisekarte der Mensa sollen verfügbar sein.

Die App soll in verschiedenen Meilensteinen entwickelt werden. Die Abgabe des letzten Meilensteins ist am 29.7.22

## 2. Entwicklungsprozess

Als Entwicklungsmodell verwenden wir Scrum. Da unser Team klein ist werden wir alle auch als Entwickler sein. Weiterhin benötigen wir wegen der Größe des Teams auch keinen dedizierten Scrum Master. Da es sich nicht um ein kommerzielles Produkt handelt ist auch die Rolle des Product Owner hier nur bedingt sinnvoll.

Grundsätzlich wollen wir an verschiedenen Komponenten arbeiten, statt zu viert an einer gemeinsamen Komponente zu arbeiten. Der entsprechende Entwickler ist dann auch verantwortlich für die Dokumentation und das Testen der entsprechenden Komponente

### Tools, Konventionen und Vorgehen bei der Entwicklung

Versionsverwaltung mit Git über Github

Im Ordner src auf dem Branch main soll immer eine lauffähige und möglichst aktuelle Version des Systems liegen.

Wenn ein neues Feature implementiert werden soll, wird ein neuer Branch erstellt. Dieser wird, sobald das Feature und entsprechende Test implementiert wurden, dann wieder in den main-Branch gemerged.

### Tests

Komponenten sollten, wenn sinnvoll möglich mit JUnit-tests getestet werden. Für größere Komponenten sollten auch ein paar Espresso-tests definiert werden.

Das System als Ganzes sollte auch nach der Implementierung einer sinnvollen Einheit mithilfe des Emulators einem Praxistest unterzogen werden.

Modellierung sollte visuell gemacht werden.

Klassenmodelle, Interaktionsmodelle und ähnliche Strukturmodelle sollten mit draw.io angelegt

werden und dann sollten sowohl die .drawio Sourcedatei als auch ein gleichnamiger .pdf Export in der Dokumentation abgelegt werden.

Textuelle Notizen sollten in einer .odt Datei angelegt werden. Hier ist kein .pdf Export nötig.

Kleinere Notizen, wie z.B. das Aussehen einer Aktivität kann auch mal als händische Skizze angelegt werden.

## **Codeconventions**

Benennung von Klassen, Methoden, Variablen usw. machen wir auf Englisch

Kommentare können in Deutsch und Englisch verfasst werden.

Util-Klassen sollten Singleton sein

## **3. Anforderungen**

Wir wollen die REST Apis von Ilias und Marvin benutzen.

Des Weiteren werden wir die Mensainformationen von der Seite der Studentenwerke Marburg beziehen. (Webscrape)

## **Bereiche und Subkategorien**

Themen für die lokale App:

Startseite

Profil

Suche

Räume

Veranstaltungen

Karten

Kalender

Mensa

Datenverwaltung (lokale Persistierung)

## **Glossar**

Menü:

Das Menü befindet sich auf jeder Seite oben links in der Ecke. In diesem kann man zwischen den Verschiedenen Seiten wechseln.

Startbildschirm:

Dies ist die Seite, auf der die App startet. Ähnlich wie im Menü sind hier Buttons, mit denen man zu den anderen Seiten wechseln kann, allerdings noch weitere Einträge, wie „Einstellungen“, „Logout“ sowie „Nutzerprofil“. (siehe Layout Entwurf)

Nutzer:

Ein Nutzer ist eine in der App Registrierte Person (Hat ein Nutzerkonto).

Applogin-Daten:

Dies sind die Daten, mit denen sich ein Nutzer in der App registriert hat.

Student:

Ein Student ist eine Person mit Ilias-Account.

Unilogin-Daten:

Dies sind die Daten, mit denen sich ein Student in Ilias anmeldet. Ein Nutzer kann Uni-Login Daten in der App hinterlegen, sodass die App auf Iliasfunktionen zugreifen kann.

Bereiche:

Sind die Verschiedenen Funktionalitätsbereiche der App. Dies sind:

Startseite, Meine Veranstaltungen, Kalender, Mensa, Karten, Suche, Datenverwaltung

## **Anwendungsfälle**

### **Bereich: Meine Veranstaltungen**

#### **1.1 Veranstaltung Hinzufügen/Löschen**

Vorbedingung: Seite „Meine Veranstaltungen“ ist geöffnet

Ablauf: Nutzer klickt auf Feld „Veranstaltung suchen“ und gibt Suchbegriff ein.

In der Liste der gefundenen Veranstaltung kann die gesuchte Veranstaltung angeklickt werden, worauf sich die Seite der Veranstaltung öffnet. Hier kann über einen Button die Veranstaltung zu „Meine Veranstaltungen“ hinzugefügt werden. Sollte die Veranstaltung bereits bei „Meine Veranstaltungen“ vorhanden sein, dient der Button stattdessen zum Entfernen der Veranstaltung aus „Meine Veranstaltungen“

#### **1.2 Veranstaltung einsehen**

Vorbedingung: Seite „Meine Veranstaltungen“ ist geöffnet

Ablauf: Der Nutzer sieht eine Liste seiner ausgewählten Veranstaltungen. Aus dieser kann er eine Veranstaltung anklicken, worauf sich die Seite zu der Veranstaltung öffnet.

Alternativ kann eine Veranstaltung, die der Nutzer noch nicht gespeichert hat, über die Suche gesucht werden und dann angeklickt werden.

Auf dieser Seite werden generelle Informationen zu der Veranstaltung angezeigt (Dozent, Veranstaltungstermine, Semesterwochenstunden, etc.), ein Button, der zur Iliasseite dieser Veranstaltung führt und eine Einsicht über die Bewertungen dieser Veranstaltung.

## **Bereich: Karte**

### **2.1 Auf Karte navigieren**

Vorbedingung: Die Seite „Karte“ ist ausgewählt.

Ablauf: Der Nutzer kann auf der Karte die verschiedenen Gebiete der Uni (Lahnberge, Innenstadt) anklicken. Wird ein Bereich angeklickt, öffnet sich eine neue Karte mit einer detaillierteren Ansicht dieses Bereichs, auf dem wieder einzelne Bereiche angeklickt werden können (Biegenstraße, Renthof, Mehrzweckgebäude, etc.) von denen sich dann wieder eine detailliertere Ansicht öffnet. Über einen zurück Knopf kann zu der vorherigen Karte zurückgekehrt werden.

### **2.2 Räume/Gebäude Suchen**

Vorbedingung: Die Seite „Karte“, oder die Seite „Suche“, mit aktiviertem „Raum“-Flag ist geöffnet.

Ablauf: Der Nutzer kann nun einen Raum- oder Gebäudenamen eingeben. Daraufhin werden ihm alle passenden Räume/Gebäude angezeigt. Diese können jeweils angeklickt werden, woraufhin sich die Hierarchie bis zu dem ausgewählten Objekt öffnet (Siehe Beispiel). Der Nutzer kann die Gebiete in der Hierarchie anklicken, woraufhin sich eine Karte öffnet auf der man die Position des jeweiligen Gebiets sieht.

Beispiel: Suchbegriff „PC Pool A3“ wird eingegeben. Der PC Pool A3 auf den Lahnbergen wird als Ergebnis vorgeschlagen. Klickt man diesen an, so wird die Hierarchie „>Lahnberge>Hans-Meerwein-Straße>Mehrzweckgebäude>A319“ angezeigt.

## **Bereich: Suche**

### **3.1 Objekte Suchen**

Vorbedingung: die Suche-Seite ist geöffnet.

Ablauf: der Benutzer klickt auf Suchbegriff und gibt eine Zeichenkette ein. Es werden alle Objekte angezeigt, die eine Übereinstimmung mit der eingegebenen Zeichenkette haben.

Abbruchszenarien: keine Objektbezeichnung stimmt mit der Eingabe überein.

Nachbedingung: Man kann auf das angezeigte Objekt klicken, um die Übersicht davon unter dem passenden Bereich zu sehen.

## **Bereich: Mensa**

## **4.1 Essensplan aufrufen**

Vorbedingung: Die Mensa-Seite ist geöffnet.

Ablauf: Der Nutzer wählt den Zeitraum und die Mensa aus. Dann werden alle Mahlzeiten angezeigt, die in dem Zeitraum in der Mensa zur Auswahl stehen.

## **Bereich: Kalender**

### **5.1 Veranstaltung eintragen / entfernen**

Vorbedingung: Die Seite „Meine Veranstaltungen“ ist geöffnet.

Ablauf: Der Nutzer klickt eine Veranstaltung aus der Übersicht in „Meine Veranstaltungen“ an oder er sucht eine Veranstaltung und klickt sie an. Die Veranstaltung öffnet sich und der Nutzer klickt auf den Button „zu Kalender hinzufügen“ / „aus Kalender entfernen“.

Nachbedingung: Alle mit der ausgewählten Veranstaltung verbundenen Termine wurden in den Kalender eingetragen / wurden aus dem Kalender entfernt.

### **5.2 Termin eintragen / entfernen**

Vorbedingung: Die Seite „Kalender“ ist geöffnet.

Ablauf: Der Nutzer klickt (in der Wochenübersicht) auf den gewünschten Zeit Slot. Es öffnet sich ein Pop-Up, in dem die aktuellen Informationen dieses Zeitslots angezeigt werden. Der Nutzer klickt auf den Button „Termin hinzufügen“ / „Termin löschen“. Daraufhin kann er den gewünschten Termin eintragen bzw. der Termin wird aus dem Kalender gelöscht.

## **Bereich: Login**

### **6.1 Anmelden**

Vorbedingung: Der Login-Screen der App ist geöffnet.

Ablauf: Der Nutzer gibt in die Felder „Username“ und „Password“ seine App-Login Daten ein und klickt dann auf den Button „Login“.

Abbruchszenario: Sind die eingetragenen App-Login Daten nicht valide (z.B. es fehlen Daten oder die Daten sind so nicht im System vermerkt), wird eine Nachricht zum aufgetretenen Fehler auf dem Login-Screen ausgegeben.

Nachbedingung: Der Startbildschirm des angemeldeten Accounts wird angezeigt.

### **6.2 Registrieren**

Vorbedingung: Der Login-Screen der App ist geöffnet.

Ablauf: Der Nutzer klickt auf den Button „noch keinen Account?“ und wird auf den Registrierungs-Screen weitergeleitet. Dort gibt er die gewünschten Anmeldedaten in die Felder „Username“ und „Password“ ein und klickt auf den Button „Register“. (Ein Username muss mindestens 3 Zeichen lang sein, ein Passwort mindestens 8 Zeichen lang)

Abbruchszenario: Sind die angegebenen Registrierungsdaten nicht valide (z.B. es fehlen Daten oder der Nutzernamen ist bereits vergeben), wird eine Nachricht zum aufgetretenen Fehler auf dem Login-Screen ausgegeben.

Nachbedingung: Der Home-Screen des angemeldeten Accounts wird angezeigt, die Registrierungsdaten sind als App-Login Daten des gerade erstellten Accounts gespeichert.

## 4. Umsetzung

Meilenstein 1 (22.05.):

Überblick über das Projekt verschaffen

Dokumentation des Projekts erstellen

Erstellen diverser Diagramme zur Hilfe

Meilenstein 2 (17.06.):

Login implementieren (hohe Priorität)

lokale Datenerhaltung implementieren (hohe Priorität)

Grundgerüst der einzelnen Bereiche (siehe Glossar) der App implementieren

Meilenstein 3 (08.07.):

Implementierung der verbleibenden Bereiche (siehe Glossar) der App

Schreiben von ausführlichen Tests

Dokumentation der Implementierung erstellen

Abrundung (29.07.):

Beheben von letzten Problemen

eventuelles Hinzufügen von weiteren Funktionalitäten

## 5. Kurzer Überblick über erreichte Funktionalität

### **Login und Register:**

Funktionieren wie erwartet.

Beim ersten Login/Register nach Installation der App werden die Mockdaten eingelesen und die lokale Datenbank wird angelegt, dies kann einen Moment dauern.

Bei jedem erfolgreichen Login/Register werden im Hintergrund die Mensadaten geladen.

**Map:**

Man kann sich einen Überblick über die Lage der Gebäude verschaffen, indem man sich durch Übersichtskarten klickt.

**Kalender:**

Man kann eigene Termine (Event genannt) im Kalender eintragen und auch wieder entfernen (long-click auf den Termin).

Man kann LectureInstances pinnen (und wieder unpinnen). Dabei wird für jeden Termin ein Event in den Kalender eingetragen.

**Meine Veranstaltungen:**

Zeigt alle Veranstaltungen (Lecture) an, die der User mal hinzugefügt hat.

Ein Klick auf die Lecture öffnet eine Detailansicht zur Lecture.

**Mensa:**

Zeigt für den ausgewählten Zeitraum und die ausgewählte Mensa die Gerichte an.

**Suche**

Man nach allen Eigenschaften von Lectures und LectureInstances suchen, die Ergebnisse werden entsprechend in einer Übersicht angezeigt.

Dies umfasst z.B. Raum, Uhrzeit, Name der Veranstaltung, Dozent.

Beide Arten von Ergebnissen können angeklickt werden und führen zu einer Übersicht über die entsprechende Lecture.