

## **Funktionale Anforderungen**

### **1. Einloggen**

**Vorbedingung** der Login Bildschirm ist auf

**Ablauf** der Nutzer gibt die Daten seines Uni Accounts ein und drückt die Bestätigungstaste.

**Abbruchscenario:** Werden die Daten falsch eingegeben, kriegt der Nutzer die Meldung und bleibt auf dem Login-Bildschirm

**Nachbedingung** der Nutzer ist eingeloggt und landet auf dem main page Bildschirm

Designentscheidung: LoginActivity

### **2. Alle Veranstaltungen des FB12 anzeigen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü

**Abbruchscenario:** der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** dem Nutzer wird die Liste von allen Veranstaltungen des FB12 angezeigt

Designentscheidung: wenn der Nutzer in der App angemeldet ist, sieht er (außer in der Login Activity) oben links im "Menü" die Button "Alle Veranstaltungen"

### **3. Veranstaltung nach Angaben eines Suchterms suchen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü. Im Suchfenster (Texteingabefenster) gibt der Nutzer den Suchterm ein

**Abbruchscenario:** bei der Eingabe des Suchterms, also bevor der Nutzer den Suchterm mit der 'Search' Button bestätigt, kann der Nutzer den schon angegebenen Suchterm im Suchfenster löschen

**Nachbedingung** der Nutzer sieht alle Veranstaltungen, die den Suchterm enthalten. Die Ergebnisse zeigen auch, wo, wann, und ob die Veranstaltung in diesem Semester stattfindet

Designentscheidung: an der "Alle Veranstaltungen" Seite (LectureSearchPageFragment) gibt's oben ein Suchfeld

### **4. Veranstaltung mit den Filtern aussortieren**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü. Der Nutzer drückt den Filter Button. Verfügbare Filtereinstellungen werden angezeigt. Dann wählt der Nutzer die entsprechenden Einstellungen und bestätigt sie.

**Abbruchscenario:** bei der Eingabe der Filtereinstellungen, also bevor der Nutzer sie bestätigt, kann er die schon ausgewählte Filtereinstellungen zurücksetzen

**Nachbedingung** der Nutzer sieht alle Veranstaltungen, die den Filterkriterien entsprechen. Die Ergebnisse zeigen auch, wo, wann, und ob die Veranstaltung in diesem Semester stattfindet

Designentscheidung: an der "Alle Veranstaltungen" Seite (LectureSearchPageFragment) gibt's eine Filterbutton, bei dem Drucken kann der Nutzer die Filters einstellen

## **5. Belegungszeitplan für einen Raum finden**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Räumen" im Menü. Der Nutzer wählt den Raum und den Zeitraum

**Abbruchscenario:** der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer sieht den Belegungszeitplan für den gewählten Raum, mit dem ausgewählten Zeitraum

Designentscheidung: wenn der Nutzer im App angemeldet ist, sieht er (außer in der Login Activity) oben links im "Menü" die Button "Raum suchen". Der Nutzer landet dann auf der Seite mit den Räumen, wo er über den Suchterm oder mit den Filtern den Raum und das Datum einstellen kann.

## **6. Leere Räume suchen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Räumen" im Menü und weiter zu "Freie Räume finden". Mit den Filtern gibt der Nutzer das Datum und den Zeitraum ein.

**Abbruchscenario:** bevor der Nutzer die Filtereingebe bestätigt kann er diese zurücksetzen. Der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer sieht leere Räume an dem angegebenen Datum und in der angegebenen Zeit

Designentscheidung: wenn der Nutzer in der App angemeldet ist, sieht er (außer in der Login Activity) oben links im "Menü" die Button "Raum suchen". Der Nutzer landet dann auf der Seite mit den Räumen, wo er über den Suchterm oder mit den Filtern den Raum und das Datum einstellen kann. Über ein extra Häkchen in den Filtern kann der Nutzer nur freie Räume am bestimmten Tag auswählen.

## **7. Teilnehmerliste einer Veranstaltung, die der Nutzer beigetreten ist, anschauen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms oder "Meine Veranstaltungen"). Der Nutzer befindet sich auf der Seite der ausgewählten Veranstaltung. Falls der Nutzer die Veranstaltung schon beigetreten ist, sieht er die 'Teilnehmer' Button. Falls nicht, kann der Nutzer die Veranstaltung beitreten und sieht dann auch die 'Teilnehmer' Button. Der Nutzer drückt den 'Teilnehmer' Button

**Abbruchscenario:** der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer sieht die Teilnehmerliste (Mitglieder) der ausgewählten ILIAS Veranstaltung

Designentscheidung: wenn der Nutzer die Details einer Veranstaltung (LectureDetailsPageFragment) anschaut, gibt's unter 'Inhalt' die Button "Teilnehmerliste", die der Nutzer sieht, falls er die Veranstaltung beigetreten ist

## **8. Teilnehmer in den Veranstaltungen, die der Nutzer selbst besucht, suchen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Beigetretenen Veranstaltungen" im Menü und weiter zu den Teilnehmern. Der Nutzer sieht alle Teilnehmer der Veranstaltungen, die er beigetreten ist. Der Nutzer gibt den Suchterm ein.

**AbbruchszENARIO:** bevor der Nutzer den Suchterm bestätigt kann er diesen zurücksetzen. Der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer sieht die nach dem Suchterm die aussortierte Teilnehmerliste (Mitglieder) der beigetretenen Veranstaltungen.

Designentscheidung: wenn der Nutzer die Teilnehmer einer Veranstaltung (ParticipantPageFragment) anschaut, kann er oben im Suchfenster bestimmte Nutzer suchen.

## **9. Veranstaltung beitreten**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms). Falls der Nutzer diese Veranstaltung nicht beigetreten ist, sieht er den 'Beitreten' Button und drückt sie.

**AbbruchszENARIO:** bevor der Nutzer „Beitreten“ drückt, kann der Nutzer jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer ist der Veranstaltung beigetreten und kann sie in der Liste der beigetretenen Veranstaltungen auffinden

Designentscheidung: wenn der Nutzer die Details einer Veranstaltung (LectureDetailsPageFragment) anschaut, gibt's die Button "Veranstaltung beitreten", falls der Nutzer diese Veranstaltung noch nicht beigetreten ist (sonst sieht der Nutzer die 'Inhalt' Button)

## **10. Beigetretene Veranstaltungen anzeigen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Beigetretene Veranstaltungen" im Menü

**AbbruchszENARIO:** der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** dem Nutzer wird die Liste von allen Veranstaltungen des FB12, die er beigetreten ist, angezeigt

Designentscheidung: wenn der Nutzer in der App angemeldet ist, sieht er (außer in der Login Activity) oben links im "Menü" die Button "Meine Veranstaltungen"

## **11. Veranstaltung verlassen**

**Vorbedingung** der Nutzer ist eingeloggt

**Ablauf** der Nutzer geht zu den "Beigetretene Veranstaltungen" im Menü und wählt eine Veranstaltung (der Nutzer kann eine Veranstaltung, die er beigetreten ist, auch über "Menü"->"Veranstaltungen" finden, auch möglich über Filter oder nach Eingabe eines Suchterms). Der Nutzer drückt die 'Verlassen' Taste auf der Seite der Veranstaltung.

**AbbruchszENARIO:** bevor der Nutzer „Verlassen“ drückt, kann der Nutzer jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer hat die Veranstaltung verlassen und die Veranstaltung wird nicht

mehr in der Liste der bereits beigetretenen Veranstaltungen angezeigt.

Designentscheidung: wenn der Nutzer die Details einer Veranstaltung (LectureDetailsPageFragment) anschaut, gibt's da die Button "Inhalt", falls der Nutzer diese Veranstaltung beigetreten ist. Unter dem 'Inhalt' steht die Button "Veranstaltung austreten" zur Verfügung

## **12. Details der Veranstaltung zeigen**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms).

**Abbruchscenario:** der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** auf der Seite der ausgewählten Veranstaltung sieht der Nutzer die Details der Veranstaltung (Name der Veranstaltung, Typ der Veranstaltung (Vorlesung/Übung/Sonstiges...), Anbieter, Semester, Zeit, Tag, Raum)

\*"Teilnehmer der Veranstaltung anschauen" wurde im Punkt 7 beschrieben

Designentscheidung: über 'Meine Veranstaltungen' oder 'Alle Veranstaltungen' kann der Nutzer eine Veranstaltung auswählen und dann landet der Nutzer auf der Seite (LectureDetailsPageFragment), wo der Nutzer die Details der bestimmten Veranstaltung sieht

## **13. Kursforum der Veranstaltung, die der Nutzer beigetreten ist, zeigen**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms) ODER er geht zu den "Beigetretenen Veranstaltungen" im Menü und wählt eine Veranstaltung. Falls der Nutzer die Veranstaltung schon beigetreten ist, sieht er die 'Inhalt' Button. Falls nicht, kann der Nutzer die Veranstaltung beitreten und sieht dann auch die 'Inhalt' Button. Der Nutzer drückt die 'Inhalt' Button. Der Nutzer sieht weiter unter den anderen Möglichkeiten die „Kursforum“ Button.

**Abbruchscenario:** der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** auf der geöffneten Seite sieht der Nutzer das Forum der Veranstaltung

Designentscheidung: wenn der Nutzer die Details einer Veranstaltung (LectureDetailsPageFragment) anschaut, gibts da die Button "Inhalt", falls der Nutzer diese Veranstaltung beigetreten ist. Unter dem 'Inhalt' steht die Button "Kursforum" zur Verfügung

## **14. Vorlesungsmaterial der Veranstaltung, die der Nutzer beigetreten ist, zeigen**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms) ODER er geht zu den "Meine Veranstaltungen" im Menü und wählt eine Veranstaltung. Falls der Nutzer die Veranstaltung schon beigetreten ist, sieht er die 'Inhalt' Button. Falls nicht, kann der Nutzer die Veranstaltung beitreten und sieht dann auch die 'Inhalt' Button. Der Nutzer drückt die 'Inhalt' Button. Der Nutzer sieht weiter unter den anderen Möglichkeiten die „Vorlesungsmaterial“ Button.

*Abbruchszenario:* der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** auf der geöffneten Seite sieht der Nutzer das Vorlesungsmaterial (z.B. wie eine Liste der verfügbaren Dokumente) der Veranstaltung

Designentscheidung: wenn der Nutzer die Details einer Veranstaltung (LectureDetailsPageFragment) anschaut, gibts da die Button "Inhalt", falls der Nutzer diese Veranstaltung beigetreten ist. Unter dem 'Inhalt' steht die Button "Vorlesungsmaterial" zur Verfügung

## **15. Übungsmaterial der Veranstaltung, die der Nutzer beigetreten ist, zeigen**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms) ODER er geht zu den "Meine Veranstaltungen" im Menü und wählt eine Veranstaltung. Falls der Nutzer die Veranstaltung schon beigetreten ist, sieht er die 'Inhalt' Button. Falls nicht, kann der Nutzer die Veranstaltung beitreten und sieht dann auch die 'Inhalt' Button. Der Nutzer drückt die 'Inhalt' Button. Der Nutzer sieht weiter unter den anderen Möglichkeiten die „Übungsmaterial“ Button.

*Abbruchszenario:* der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** auf der geöffneten Seite sieht der Nutzer das Übungsmaterial (z.B. wie eine Liste der verfügbaren Dokumente) der Veranstaltung

Designentscheidung: wenn der Nutzer die Details einer Veranstaltung (LectureDetailsPageFragment) anschaut, gibts da die Button "Inhalt", falls der Nutzer diese Veranstaltung beigetreten ist. Unter dem 'Inhalt' steht die Button "Übungsmaterial" zur Verfügung

## **16. Von der "Details des Raums" die Details der Veranstaltung anschauen**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** der Nutzer geht zu den "Raum suchen" im Menü, wählt das Datum und dann einen Raum, der an diesem Tag rot gefärbt ist. Der Nutzer sieht den Zeitraum, in dem diese Veranstaltung in diesem Raum stattfindet. Der Nutzer drückt auf die Veranstaltung in diesem Zeitraum.

*Abbruchszenario:* der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** der Nutzer sieht die Details der Veranstaltung

Designentscheidung: über 'Raum suchen' im Menü, wie in der Anforderung beschrieben

## **17. Chat über die Veranstaltung**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** der Nutzer geht zu den "Veranstaltungen" im Menü, wählt eine Veranstaltung (auch möglich über Filter oder nach Eingabe eines Suchterms) ODER er geht zu den "Beigetretenen Veranstaltungen" im Menü und wählt eine Veranstaltung. Der Nutzer sieht die Details der Veranstaltung und drückt auf die Taste "Teilnehmer" der Veranstaltung. Auf der geöffneten Seite wählt der Nutzer einen Teilnehmer und wählt die Funktionalität „Chat starten“

*Abbruchszenario:* der Nutzer kann jederzeit eine andere Seite oder App-Funktionalität auswählen

**Nachbedingung** auf der geöffneten Seite sieht der Nutzer das Chatfenster

Designentscheidung: wenn der Nutzer die Teilnehmerliste einer Veranstaltung (die er selbst beigetreten ist) sieht, kann er auf einen Teilnehmer drücken und somit den Chat mit ihm starten

## **18. Ausloggen**

**Vorbedingung** Der Nutzer ist eingeloggt. Eine beliebige Seite der App ist geöffnet

**Ablauf** Der Nutzer geht zu "Abmelden" im Menü

*Abbruchszenario:* keine

**Nachbedingung** Der Nutzer wurde erfolgreich aus seinem Account abgemeldet und sieht die entsprechende Meldung der App

Designentscheidung: wenn der Nutzer in der App angemeldet ist, sieht er an jeder Seite (außer Login Activity) die "Logout" Button oben rechts (oder auch im Menü gibt's die "Sign out " Button)

## **Nichtfunktionale Anforderungen**

### **1. Daten**

Die Veranstaltungsdaten werden aus der Ilias Datenbank gespeist

### **2. Datensicherheit**

Die Userprofile haben die selben Attribute, wie der jeweilige Uni Account.

### **3. Datenbank**

Die meisten Daten werden aktuell aus der Datenbank (Firebase) mit dummy Daten geholt. Die Login und Logout Funktionalitäten laufen jetzt mit den Ilias Daten über SOAP (ILIAS Test-Instanz).

In der Zukunft für die aus ILIAS verfügbaren Daten (Produkt-Instanz) kann das SOAP API und für die aus MARVIN verfügbaren Daten kann das REST API der Pegasus App (<https://github.com/fluxapps/ILIAS-Pegasus>) verwendet werden.

### **4. Sprachen**

Die Benutzeroberfläche der Applikation soll mindestens die Sprachen Deutsch und Englisch unterstützen. Zudem soll eine unkomplizierte Erweiterung um zusätzliche Sprachen durch das Design der Applikation ermöglicht werden.

### **5. Portabilität**

Unterstützt werden sollen grundsätzlich alle Android-Endgeräte, die über die Android- Version 7.1 oder höher verfügen.

## **Glossar**

Keine