

Softwarepraktikum 2022

Meilenstein 1

1. Kurzbeschreibung

In unserem Projekt wird eine mobile Campus App für Endgeräte, mit einem Android Betriebssystem (V. 7.1 o. höher), entwickelt. Die Studierenden und Mitarbeiter der Universität Marburg sollen mit der App schneller Informationen erlangen und sich besser organisieren können.

Die App soll Funktionen bereitstellen mit denen der Benutzer auf das Ilias Forum der Universität Marburg zugreifen kann und die Funktionen des Forums in Gänze nutzen kann. Der Benutzer soll auf Veranstaltungspläne oder Lagepläne von Vorlesungsräumen der Universität zugreifen können, es soll eine Feedbackfunktion für Vorlesungen geben. Außerdem soll die App dem Benutzer eine Such- und Filterfunktion bieten, mit dem er Studierende aus demselben Kurs auffinden kann.

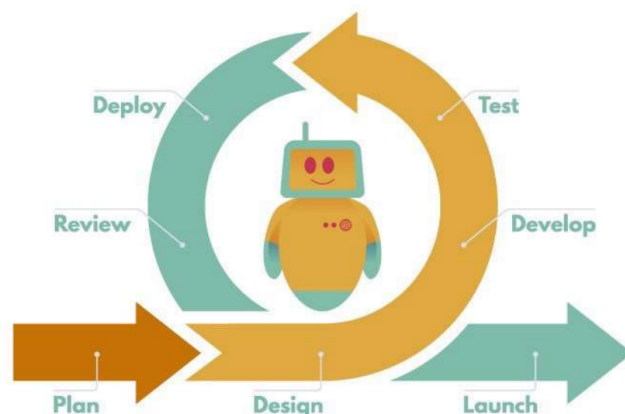
Für die verfügbaren Daten aus Ilias soll die REST API der Pegasus App als Datenbank dienen. Die Benutzeroberfläche der App soll die Sprachen Deutsch und Englisch unterstützen, sie soll aber auch eine unkomplizierte Erweiterung der Sprachen bieten. Der Benutzer soll eine möglichst intuitive Benutzung da geboten bekommen, die an den Design- Guidelines für Android orientiert sind.

Des Weiteren steht für die Zukunft noch eine Chat- Funktion aus, die eingebaut werden könnte. Mit dieser können Benutzer untereinander kommunizieren, ohne dass die Telefonnummern verwendet werden müssen. Außerdem soll es Funktionen für Bus- und Speisepläne und für Hausaufgabenabgaben geben.

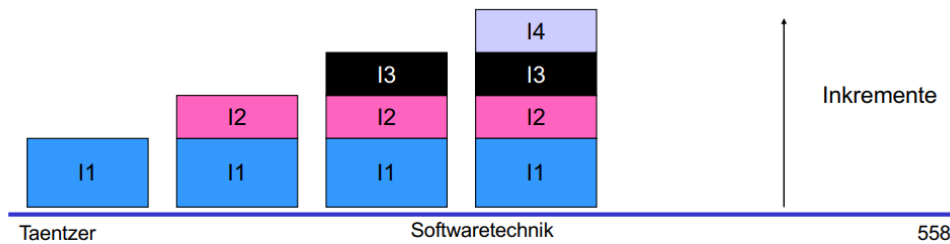
Das gesamte Projekt soll bis zum 29.07.2022 realisiert sein. Der erste Meilenstein soll bis zum 22.05.2022 erarbeitet sein, der darauf folgende zweite Meilenstein soll bis zum 17.06.2022 ausgearbeitet sein. Der dritte und letzte Meilenstein soll bis zum 08.07.2022 realisiert sein.

2. Entwicklungsprozess

- Als **Prozessmodell** werden wir die Agile Softwareentwicklung verwenden. Die basiert auf der Inkrementeller sowie Iterativer Systementwicklung, wird oft für Webanwendungen verwendet.



- **Inkrementelle Systementwicklung** bedeutet, mit einem (i.a. relativ kleinen) Kernsystem zu beginnen und diesen Kern schrittweise zu erweitern, bis die gewünschte Funktionalität erreicht ist.



Die Anforderungen zu den Hauptfunktionalitäten des Systems sind schon am Anfang des Projekts meistens fixiert und in der Anforderungsspezifikation beschrieben. Wir fangen mit den einfachen Funktionalitäten (Login, Logout usw.) an und werden dann weitere Bausteine entwickeln. Nach dem Hinzufügen jedes Bausteins wird das System getestet. Jeder durchgeführte Testfall wird durch Eingabe, erwartetes Verhalten und tatsächliches Verhalten dokumentiert.

Iterative Entwicklung ist eine Strategie zur Überarbeitungsplanung, die Zeit für laufende Revision und Verbesserung der Teile des Systems vorsieht. Das Ergebnis einer Iteration wird auf notwendige Änderungen untersucht, vor allem hinsichtlich einer Anpassung der Ziele späterer Iterationen.

- **Aufgabenverteilung über die Teammitglieder**

Ganz grob können wir die Aufgabenverteilung so darstellen:

Emre: Algorithmen, Backend, Layout

Valeria: Firebase, Datenmanipulation

Niklas: Projektmanagement

Adrian: Layout

Aber da unser Team nicht groß ist, wird jeder Teammitglied die Vorstellung über Backend des Apps oder über die Datenbank (vorübergehend Firebase) haben und sich zum Teil damit beschäftigen.

- Der Projektmanager ist für Projektplanung-Dokumente, To-Do Liste, Zeitmanagement verantwortlich. Die Mitglieder, die mit dem Layout arbeiten, sind für das korrekt laufende Layout verantwortlich. Die Mitglieder, die mit der Database arbeiten, sind für die korrekt laufende Database, Anfragen zur Database verantwortlich. Jeder Code-Entwickler ist unter anderem für die Tests nach jedem Baustein verantwortlich.

- Bei der Entwicklung (des Codes) werden die Java Code-Konventionen eingehalten. Der Code wird mit Javadoc (alle Klassen und alle öffentlichen Methoden) dokumentiert.

Model visually mittels UML Diagramme verwenden wir auch schon im Meilenstein 1.

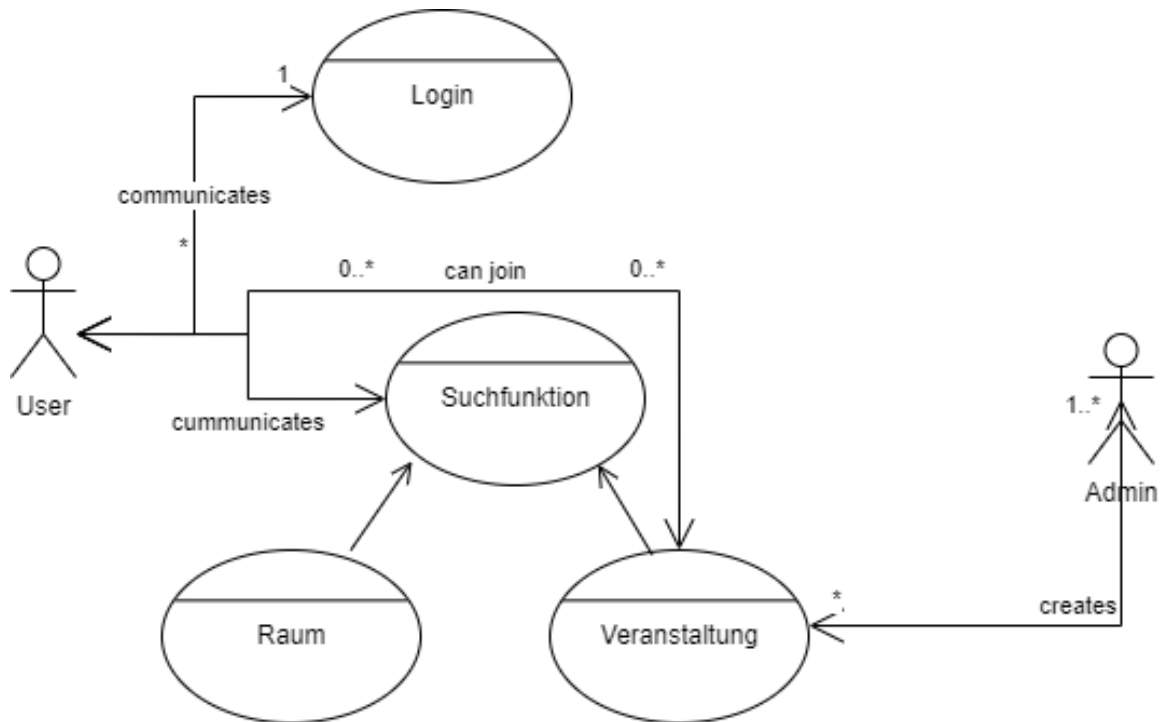
Requirements management wird auch wichtig sein. Die Anforderungen werden priorisiert und es kann sein dass die Anforderungen im Laufe der Entwicklung etwas (aber ohne grundlegende Änderungen an der Systemfunktionalität) an das Development angepasst werden.

Wahrscheinlich wird auch Pair-Programming per IntelliJ Ultimate angewandt.

Da unser System nicht groß ist, Collective code ownership wird auch meistens relevant sein.

3. Anforderungen

- Definition des Zielsystems: Endgeräte, mit einem Android Betriebssystem (V. 7.1 o. höher)
- **Geschäftsanforderungen:** da unser Projekt kein Geschäftsprojekt ist, wird es hier nicht ausführlich beschrieben. Die Zielgruppe sind Studierende der Uni Marburg, mit dem Fokus auf Studierenden der FB12 für die Testphase. Das Ziel des Projekts ist die Übertragung des Ilias-Systems für Studenten auf das Android App (vielleicht werden in der Zukunft die erweiterten Funktionalitäten des Systems auch für die Lehrkräfte weiterentwickelt).
- **UML Use Case Diagram**



- **Benutzeranforderungen:**

Allgemein: Der Benutzer (der Student) kann das Ilias System (von Anfang nur die begrenzten Funktionalitäten) im Android App mit seinem Uni-Account verwenden.

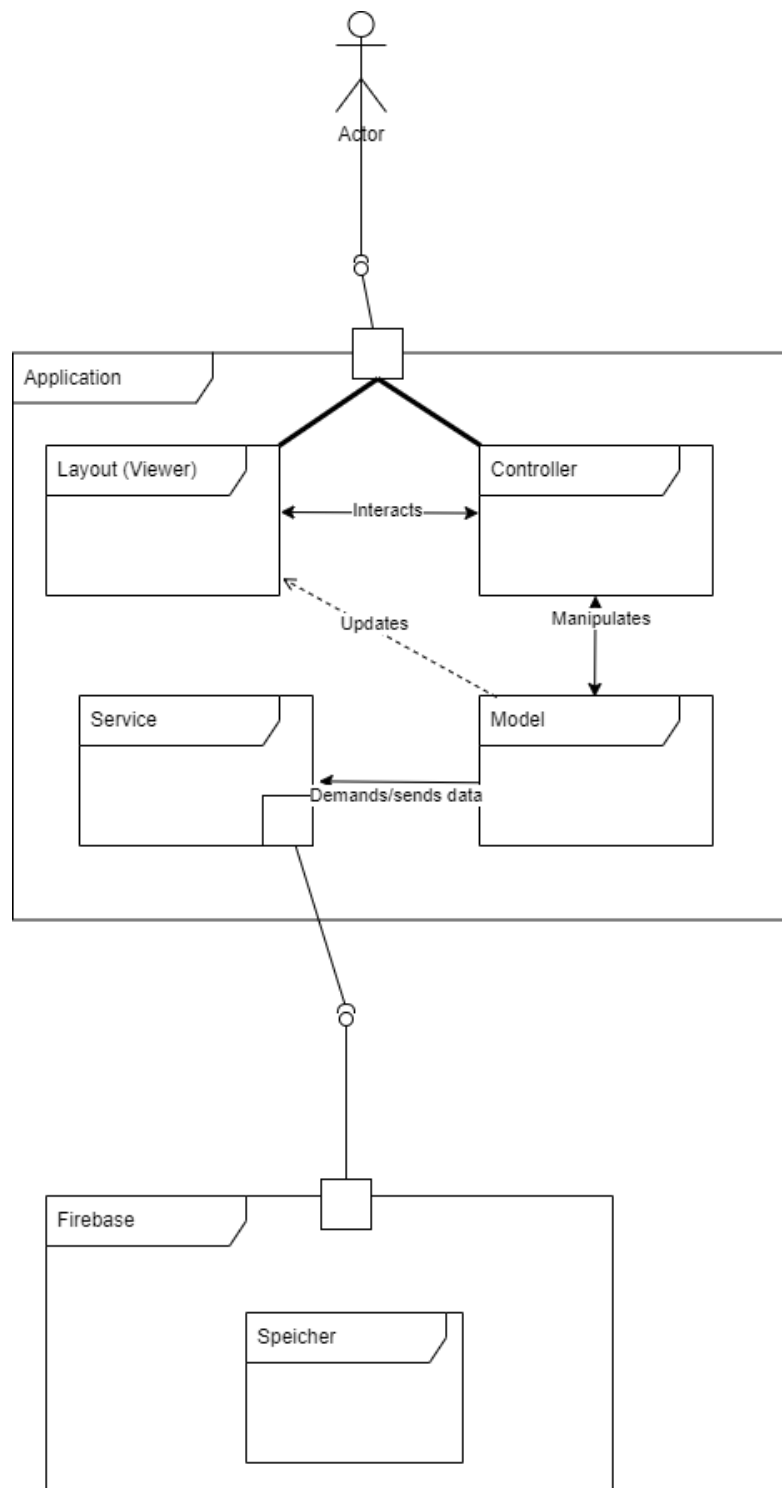
Detaillierter:

- Der Benutzer kann sich im App mit seinem Uni-Account anmelden
- Der Benutzer kann sich, wenn er im App eingeloggt ist, abmelden
- Der Benutzer kann die Liste aller Veranstaltungen des FB 12 anschauen
- Der Benutzer kann die Veranstaltungen des FB 12 mit dem Suchterm oder mit den Filter-Einstellungen aussortieren
- Der Benutzer kann die Details der Veranstaltungen anschauen (z.B. Typ der Veranstaltung (Vorlesung/Übung/Sonstiges...), Anbieter, Verfügbarkeit usw.)
- Der Benutzer kann verfügbare Inhalte der Veranstaltungen anschauen (z.B. Kursforum, Vorlesungsmaterial, Übungsmaterial, Teilnehmer)
- Der Benutzer kann über die "Details der Veranstaltung" den Zeitplan des Raums dieser Veranstaltung anschauen
- Der Benutzer kann einen Teilnehmer der Veranstaltung auswählen und das Chatfenster öffnen (um in der zukünftigen Funktionalität den Chat zu starten)
- Der Benutzer kann die Teilnehmer der Veranstaltungen, die er selbst besucht, anschauen und diese Teilnehmer (z.B. mit einem Suchterm) aussortieren

5. Entwurf

- Grober Entwurf / Architektur

Als die Architektur verwenden wir die Model-View-Controller Architektur. Dabei wird sie voraussichtlich über Service Paket mit der Datenbank kommunizieren.



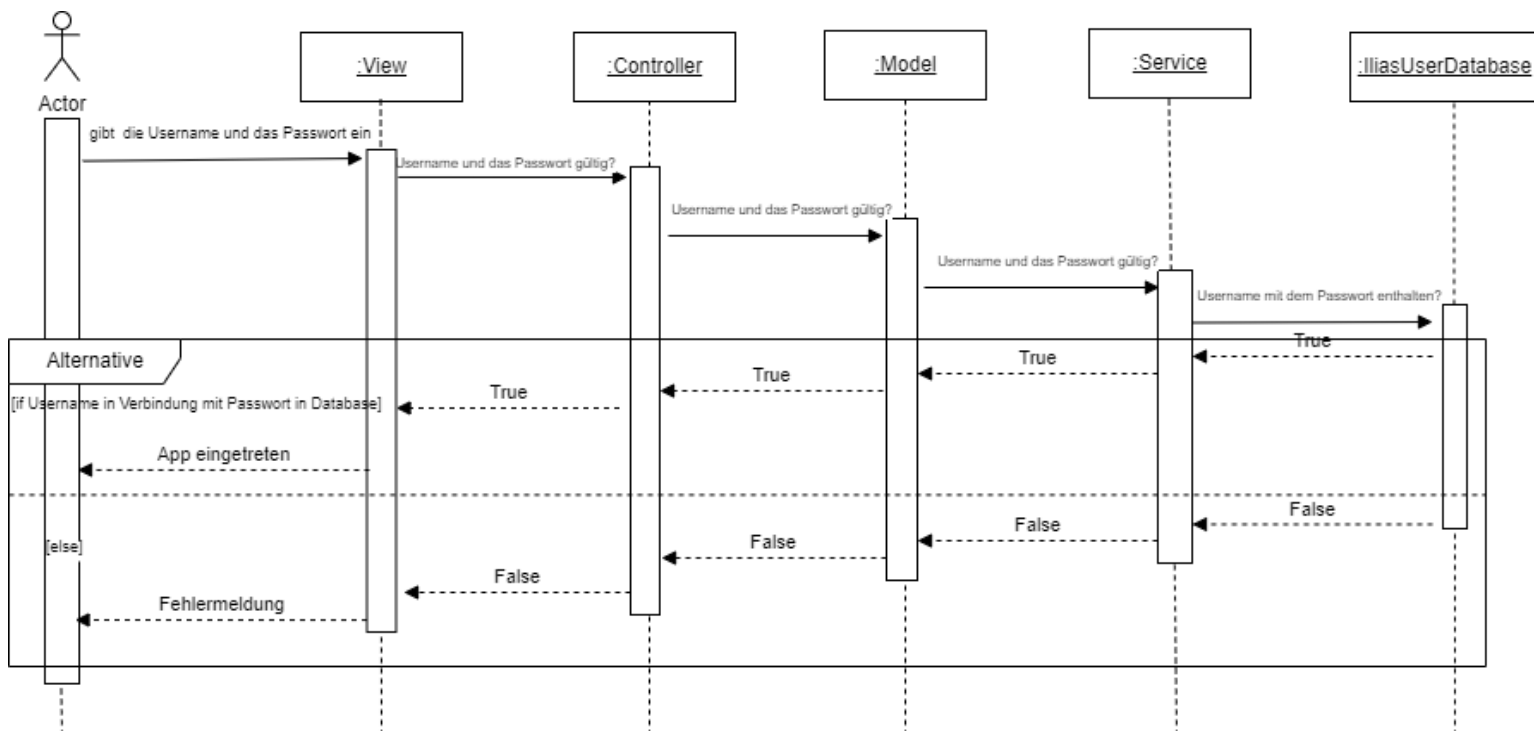
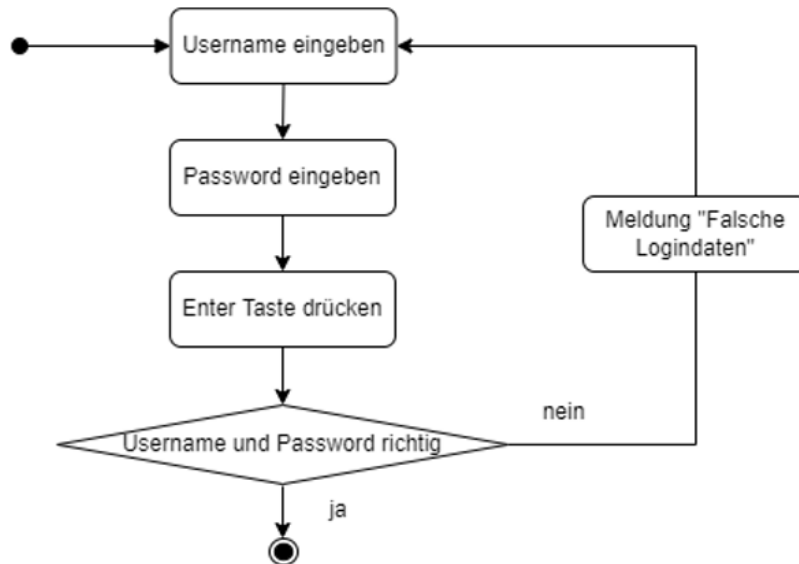
- Detaillierter Entwurf der Struktur

Da wir nicht viel Erfahrung mit dem App haben und aus diesem Grund jetzt wahrscheinlich nicht genau projektieren können, an welche Klassen wir Bedarf haben werden, haben wir die Klassenstruktur bis jetzt nicht ausführlich modelliert, weil die sich dann stark von der tatsächlichen wahrscheinlich unterscheiden wird. Wir haben weiter gezeigt dass wir die Model View Controller Architektur verwenden, und im Rahmen der Pakete dieser Architektur werden weiter mit Klassen arbeiten.

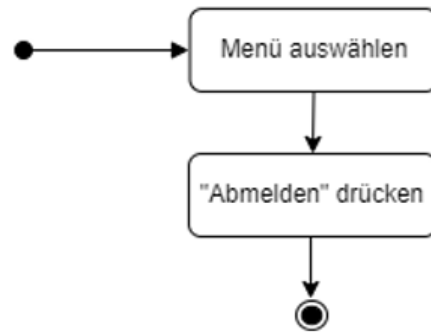
- Detaillierter Entwurf des Verhaltens

Im Rahmen des Projekts für den Entwurf des Verhaltens haben wir die Aktivitätsdiagramme und die Sequenzdiagramme für die Modellierung verwendet.

- Login

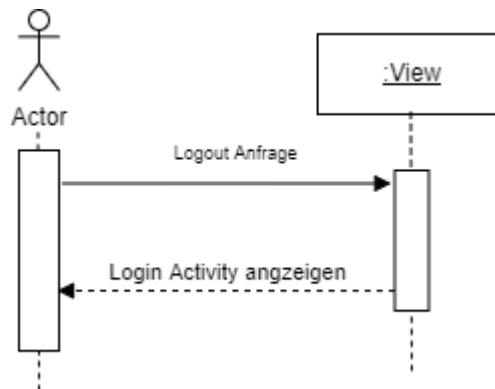


- Logout

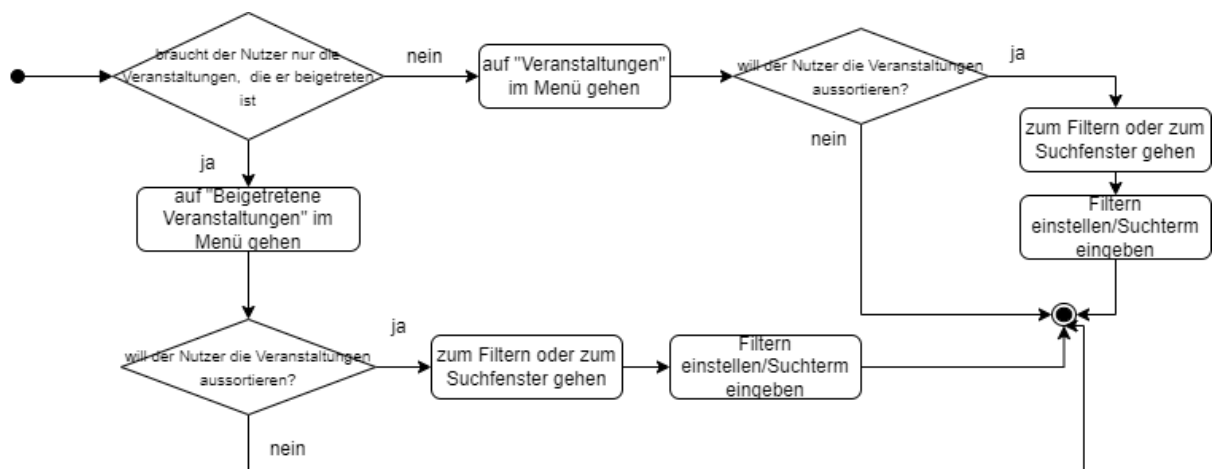


Bei dem Logout werden wir nichts im Modell ändern/speichern (z.B. wie bei dem Logout aus evault Database, wenn es gefragt wird "Änderungen speichern?"), weil der Nutzer bei seiner Aktivitäten im App alles speichert.

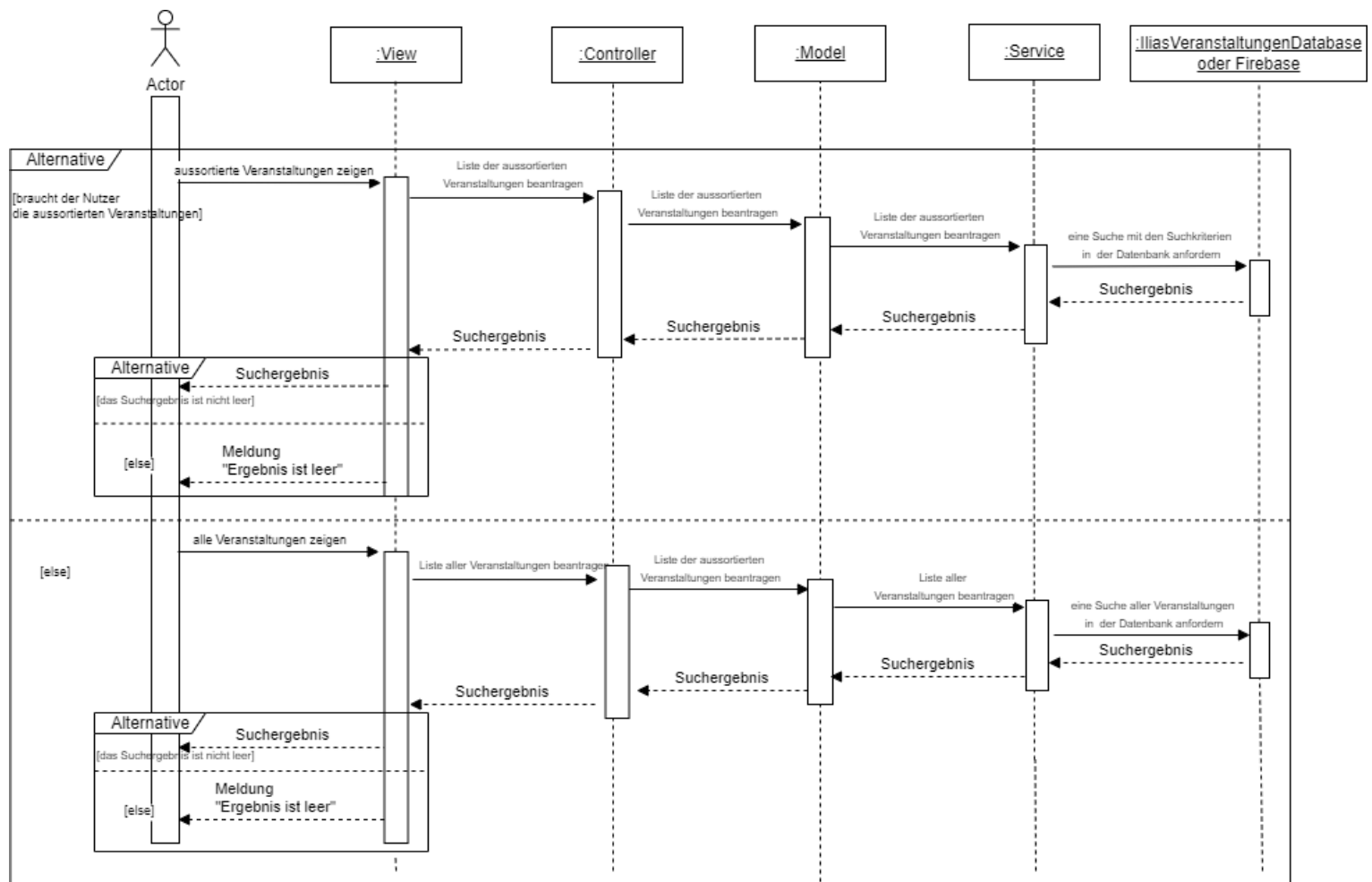
Falls es aber nötig sein wird (wenn wir die Klassen spezifizieren) z.B. reset() für das Modell bei dem Logout zu tun (oder etwas anderes im Modell), wird das Sequenzdiagramm ausführlicher sein.



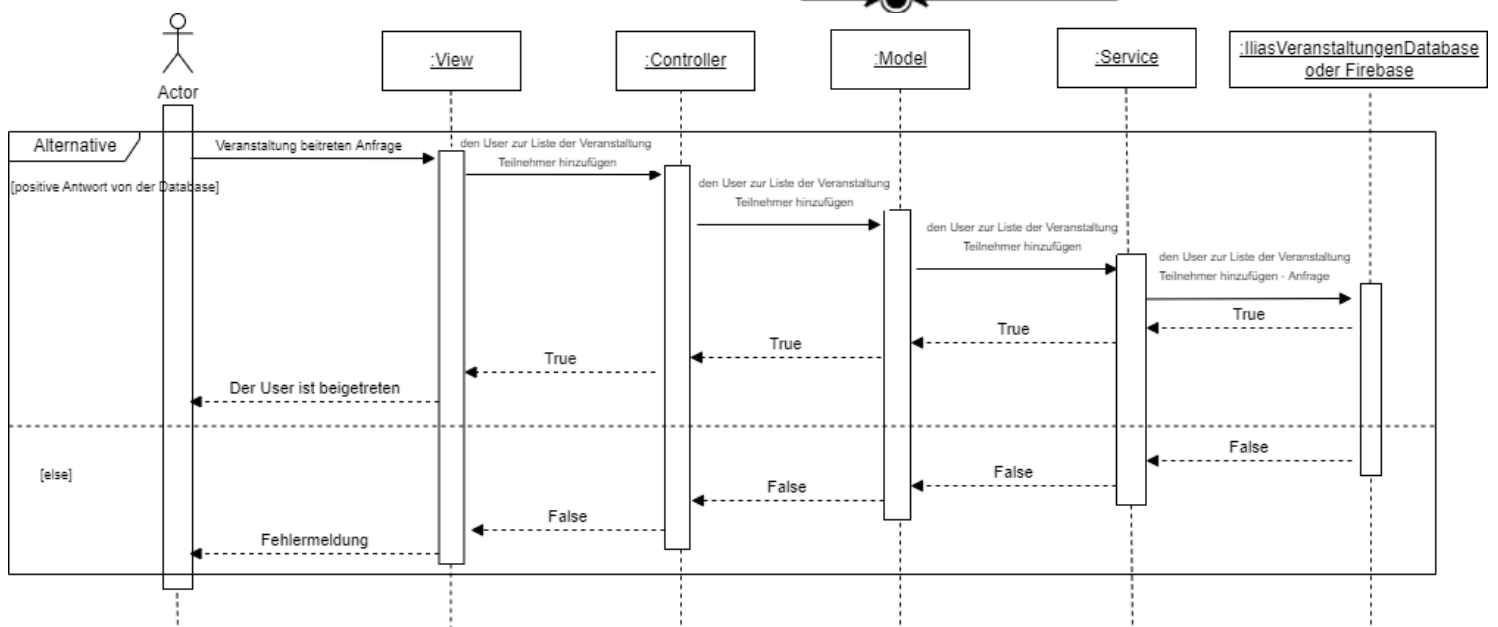
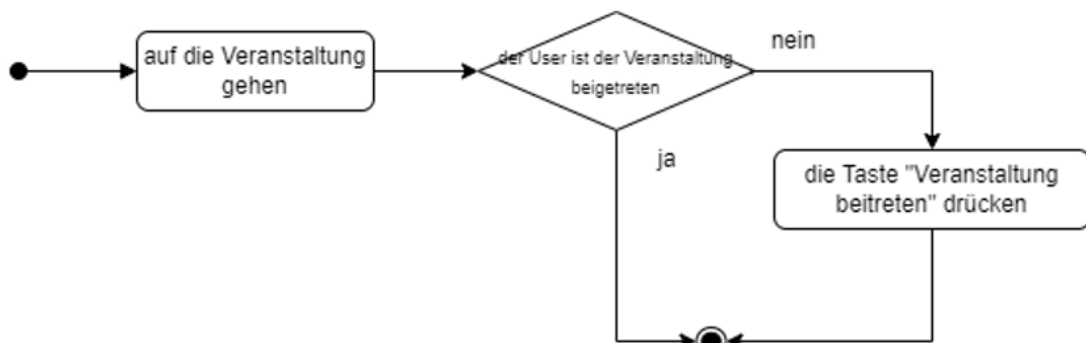
- Liste der Veranstaltungen (optional nur beigetretener) zeigen, komplett oder mit den Filtern oder einem Suchterm aussortierte



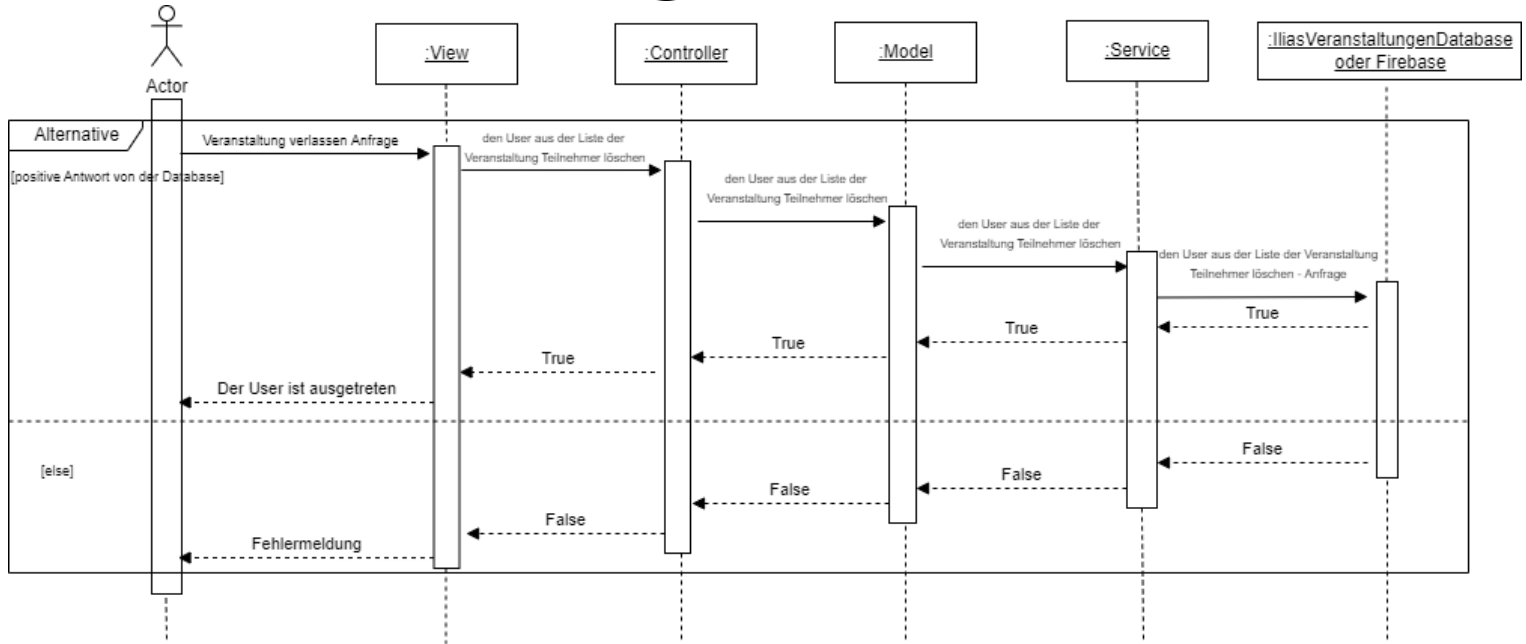
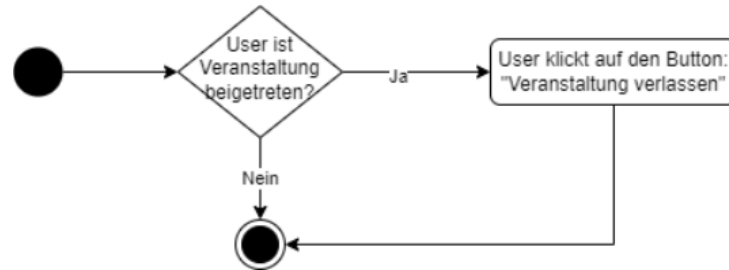
Das Sequenzdiagramm ist für "alle Veranstaltungen" angepasst, für die "beigetrete Veranstaltungen" wird es ähnlich sein: die Suchanfrage an Database wird sich unterscheiden und das Ergebnis im anderen Layout gezeigt



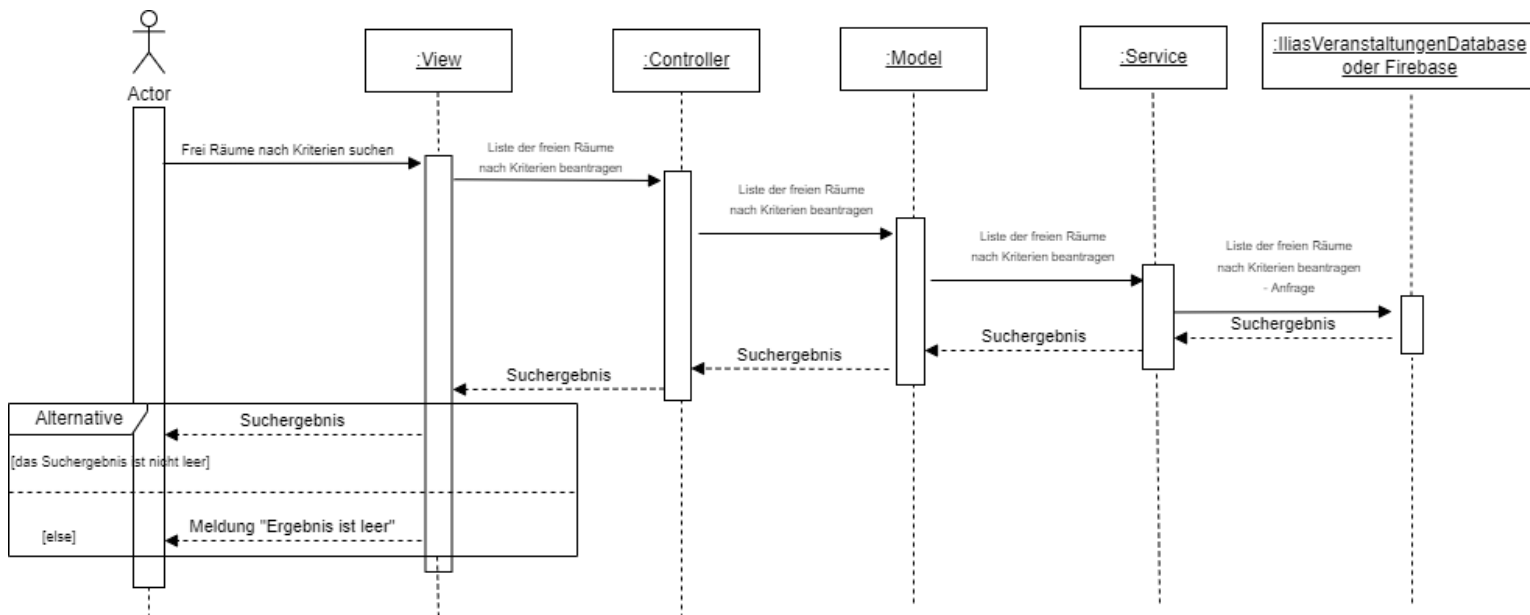
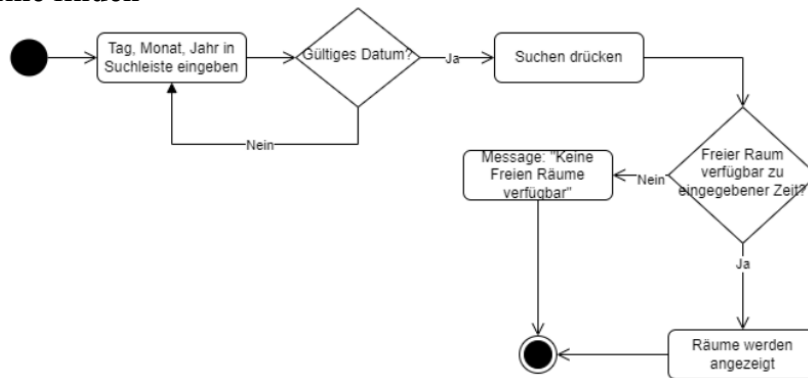
- Veranstaltung beitreten



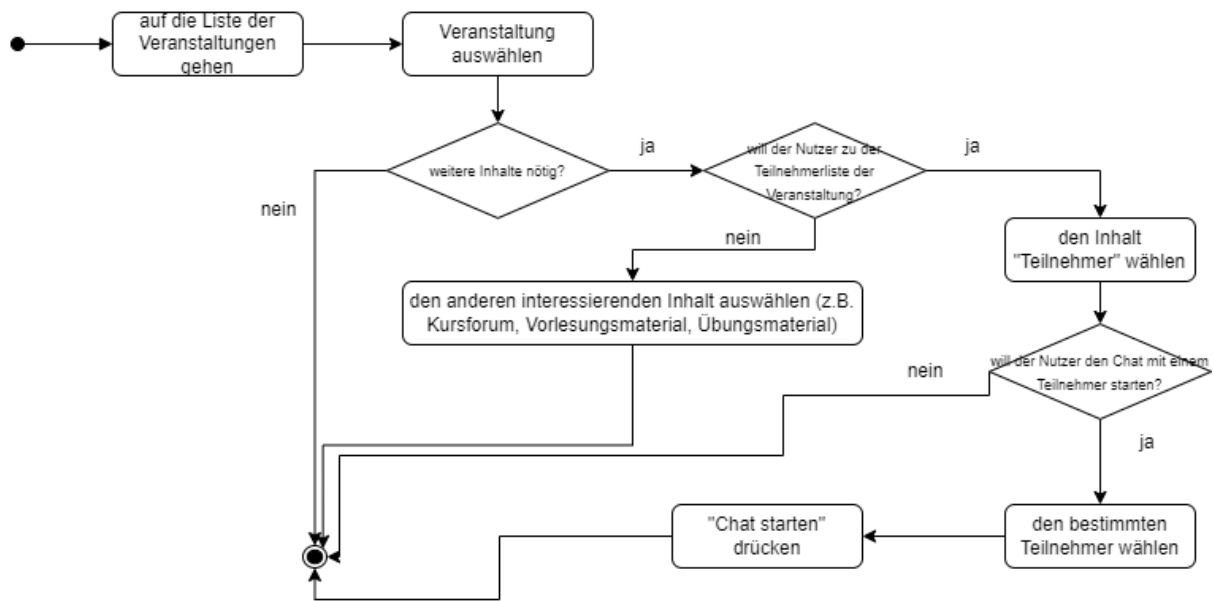
- Veranstaltung verlassen



- Freie Räume finden



- Details der Veranstaltung (Anbieter, Semester, Zeit, Tag, Raum) und weitere Inhalte zeigen (z.B. Teilnehmer, Kursforum, Vorlesungsmaterial, Übungsmaterial, auch Chatfenster mit einem Teilnehmer)



- Zeitplan für einen Raum zeigen: über alle Räume im Menü oder über die Details der Veranstaltung

