

# Softwarepraktikum 2022

## Meilenstein 2

### 1. Anforderungen

#### - **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
- Der Benutzer kann den Zeitplan für den Raum für den bestimmten Zeitraum anschauen
- Der Benutzer kann mit der Eingabe der Zeit und des Datums nach leeren Räumen (für FB12) suchen
- Der Benutzer kann die Veranstaltung beitreten, austreten, und die Liste von Veranstaltungen, die er momentan belegt, anschauen (?? das steht so nicht direkt in “Softwarepraktikum 2022 Anforderungsbeschreibung.pdf”, aber es wäre wahrscheinlich für die anderen Funktionalitäten bequem, wenn wir diese Anforderung auch implementieren könnten)
- Die **detaillierte Anforderungsspezifikation** (funktionale und nicht funktionale Anforderungen) zu den minimalen Funktionsanforderungen (Suchfunktionen, Details der Veranstaltung zeigen) ist im separaten PDF Dokument “Anforderungen-Designentscheidung.pdf” präsentiert.
- Die Unterteilung der Anforderungen nach Wichtigkeit und Komplexität steht in dem separaten Dokument Anforderungen\_SOAP\_Firebase.pdf. In diesem Dokument ist es auch dargestellt, welche Daten für welche Anforderungen geholt werden (Test-Ilias mit SOAP oder Firebase dummy Daten)

## 2. Entwurf

Das Klassendiagramm mit den Assoziationen und Vererbungen ist im separaten Dokument "Klassendiagramm.png" dargestellt (die Datei ist ganz groß, deshalb als .png, um die Qualität nicht zu verlieren). Es beschreibt die Struktur der Klassen etwas grober, als es tatsächlich ist. Da haben wir mehrere Hilfsklassen ausgeschlossen, um die Struktur etwas übersichtlicher zu machen.

Um mit SOAP zu arbeiten, wurden die Klassen mit dem EasyWSDL Plugin generiert (die sind auf dem Klassendiagramm auch nicht dargestellt).

Die Aktivitätsdiagramme und Sequenzdiagramme sind am Ende dieses Dokuments präsentiert.

## 3. Planung

- Die folgende Tabelle zeigt ziemlich grob, wie viel Zeit bereits für die Implementierung der Anforderungen aufgewendet wurde (dark blau) und wieviel Zeit voraussichtlich die restlichen Kernanforderungen noch brauchen werden.

			June																												July							
			6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8			
Anforderung	Priorität	Tage																																				
Der User kann sich über seinen Uni Account einloggen	1	5																																				
Dem User wird eine Liste der von allen Veranstaltungen des FB12 angezeigt	2	6																																				
Der User kann die Veranstaltungen mit einem Suchterm aussortieren	2	5																																				
Der User kann die Veranstaltungen mit den Filtereinstellungen aussortieren	2	4																																				
Der User kann den Zeitplan bzw. Belegung eines Raumes sehen	3	7																																				
Mit Eingabe einer definierten Zeit kann der User leere Räume suchen	3	5																																				
Der User kann die Teilnehmer aus ILIAS einer Veranstaltung einsehen	2	6																																				
Der User kann mit dem Suchterm die Teilnehmer aller Veranstaltungen, die er selbst besucht, aussortieren	4	6																																				
Dem User wird eine Liste von Veranstaltungen gezeigt, die er momentan belegt	2	6																																				
Der User kann einer Veranstaltung beitreten	2	4																																				
Der User kann eine Veranstaltung verlassen	2	4																																				
Details der Veranstaltungen werden visualisiert (Anbieter, Semester, Zeit, Raum)	2	5																																				
Der User kann den Kursforum der Veranstaltung anschauen	4	8																																				
Der User kann das Vorlesungsmaterial der Veranstaltung anschauen	4	8																																				
Der User kann das Übungsmaterial der Veranstaltung anschauen	4	8																																				
Man kann von der "Details der Veranstaltung" Ansicht zu dem Detailansicht "Raum" (-> Zeitplan des Raums)	3	8																																				
Man kann von der "Details der Veranstaltung" Ansicht zu den "Teilnehmer" (-> Chat) durchklicken	4	8																																				
User kann sich über seinen Uni Account ausloggen	1	5																																				

## Entwurf Diagramme

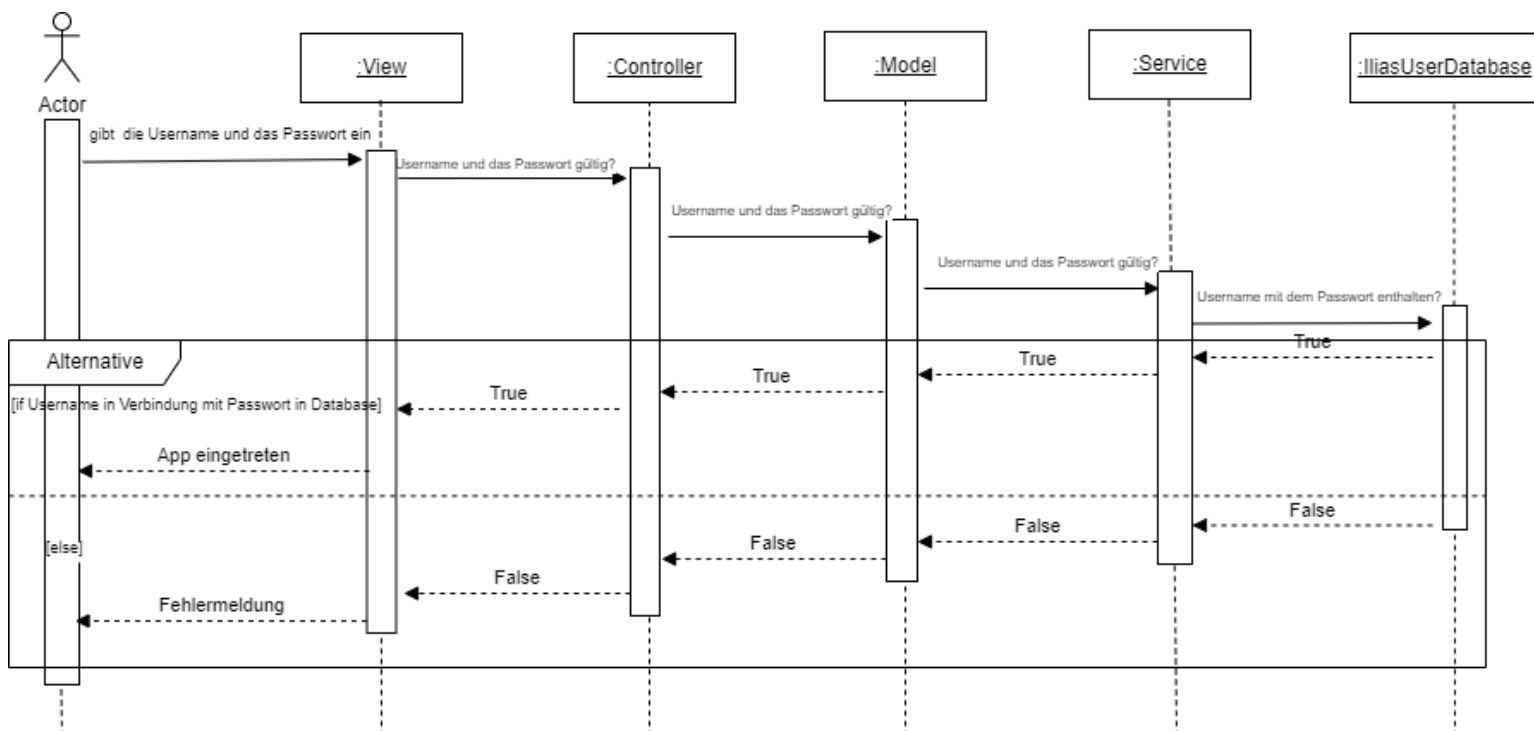
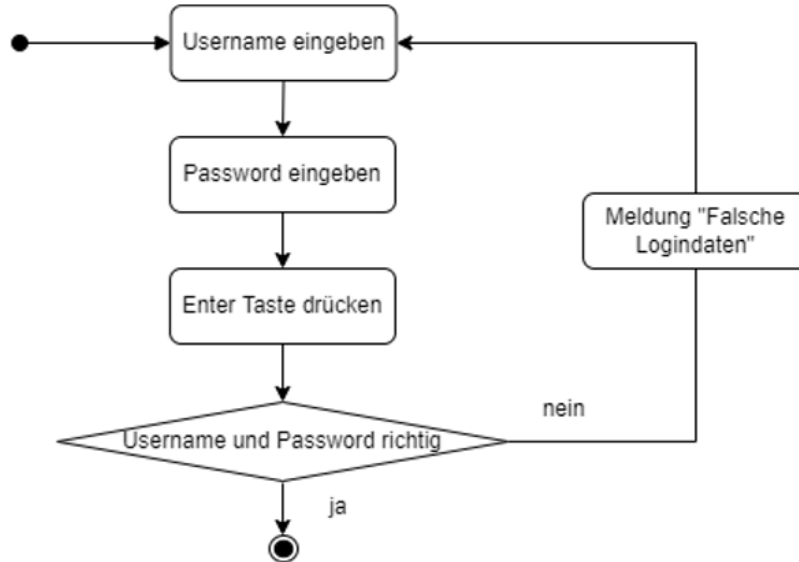
### - Grober Entwurf / Architektur

Als die Architektur verwenden wir die Model-View-Controller Architektur. Dabei kommuniziert sie über Service Paket (SoapAPI) mit der Datenbank (Firebase und SOAP).

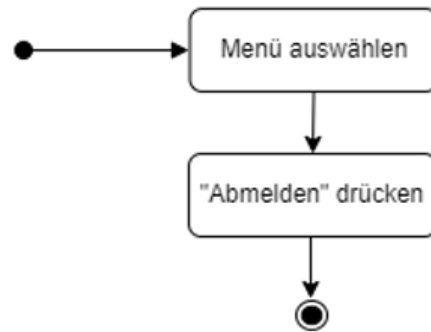
## - Detaillierter Entwurf des Verhaltens

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

### - Login

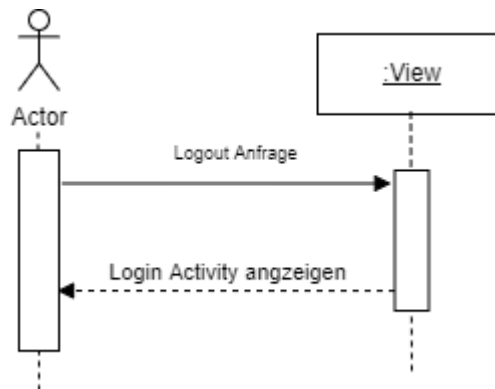


## - Logout

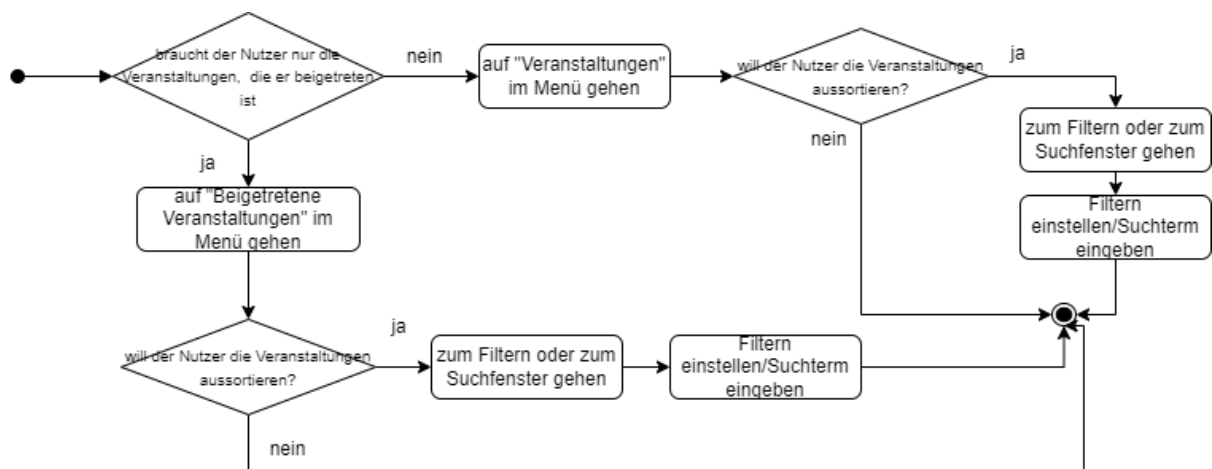


Bei dem Logout werden wir nichts im Modell ändern/speichern (z.B. wie bei dem Logout aus evault Database, wenn es gefragt wird "Änderungen speichern?"), weil der Nutzer bei seiner Aktivitäten im App alles speichert.

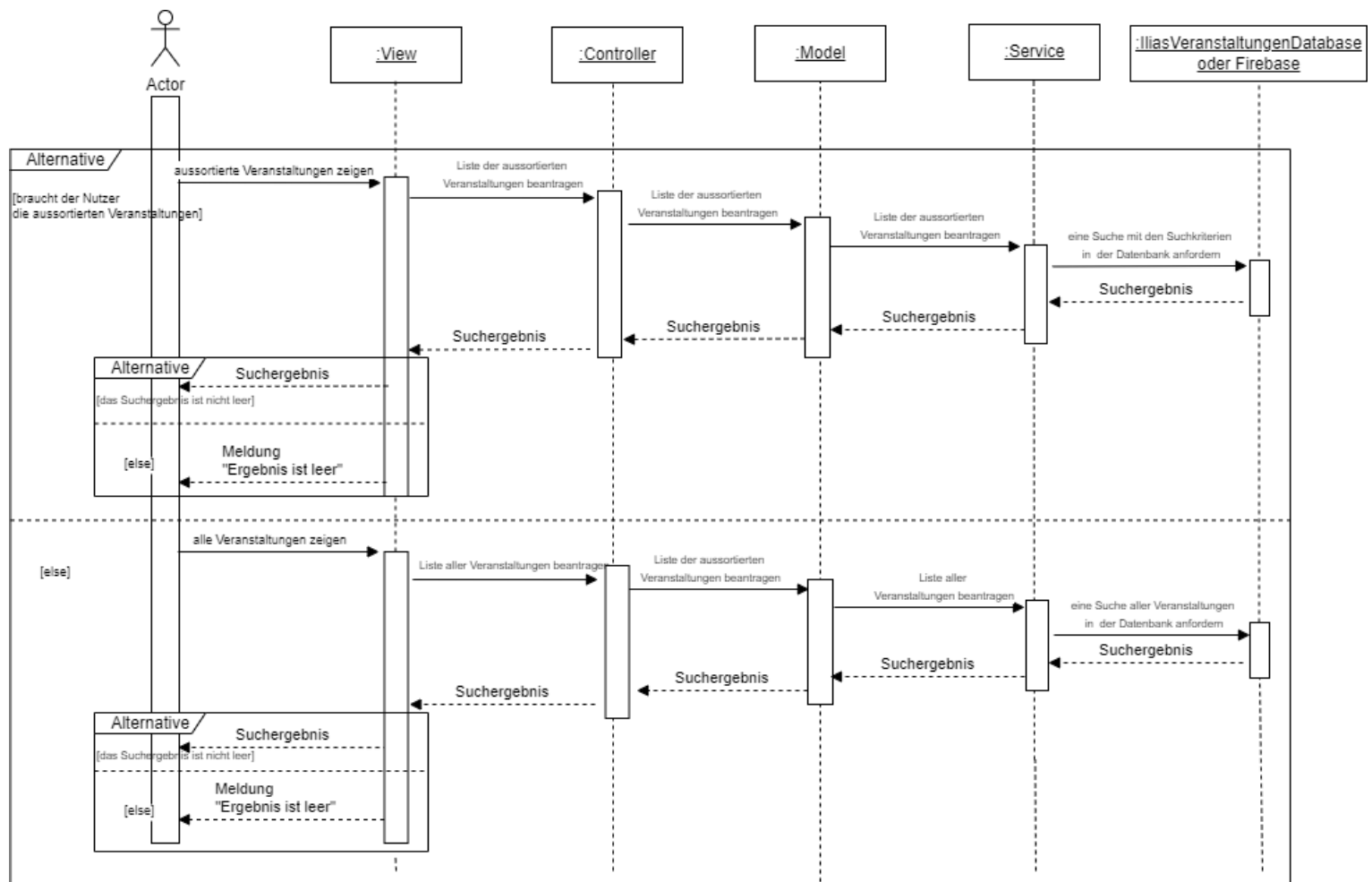
Falls es aber nötig sein wird (wenn wir die Klassen spezifizieren) z.B. reset() für das Modell bei dem Logout zu tun (oder etwas anderes im Modell), wird das Sequenzdiagramm ausführlicher sein.



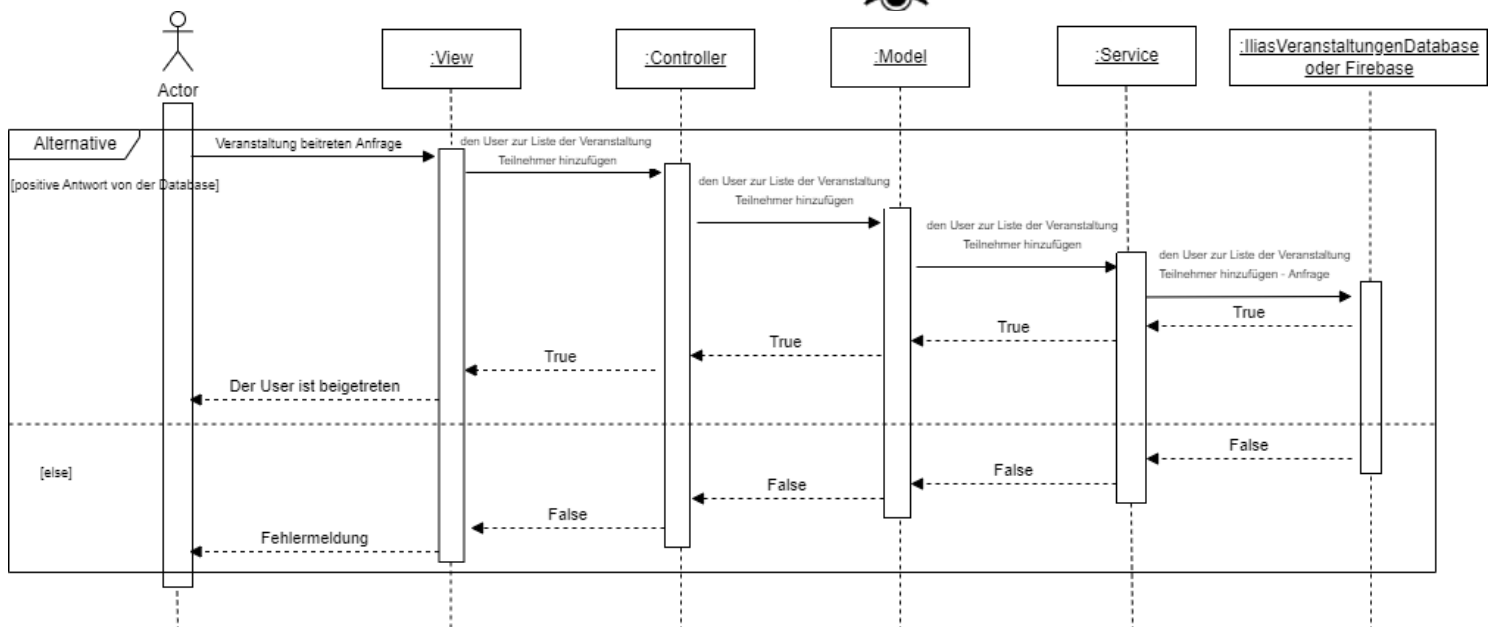
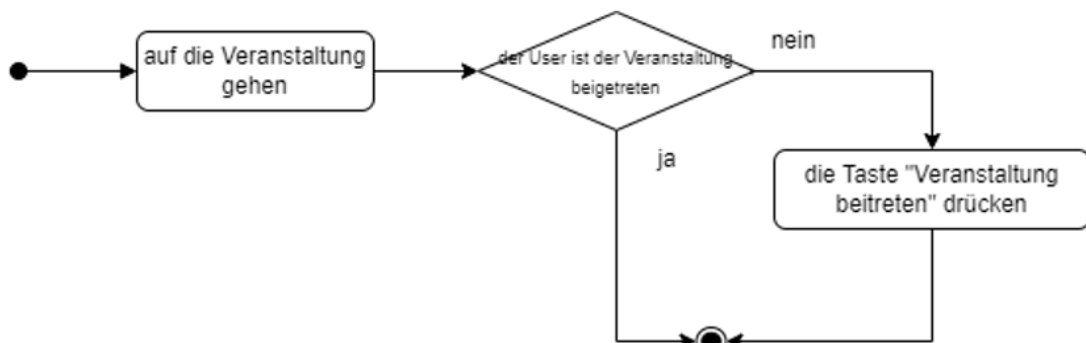
## - Liste der Veranstaltungen (optional nur beigetretener) zeigen, komplett oder mit den Filtern oder einem Suchterm aussortierte



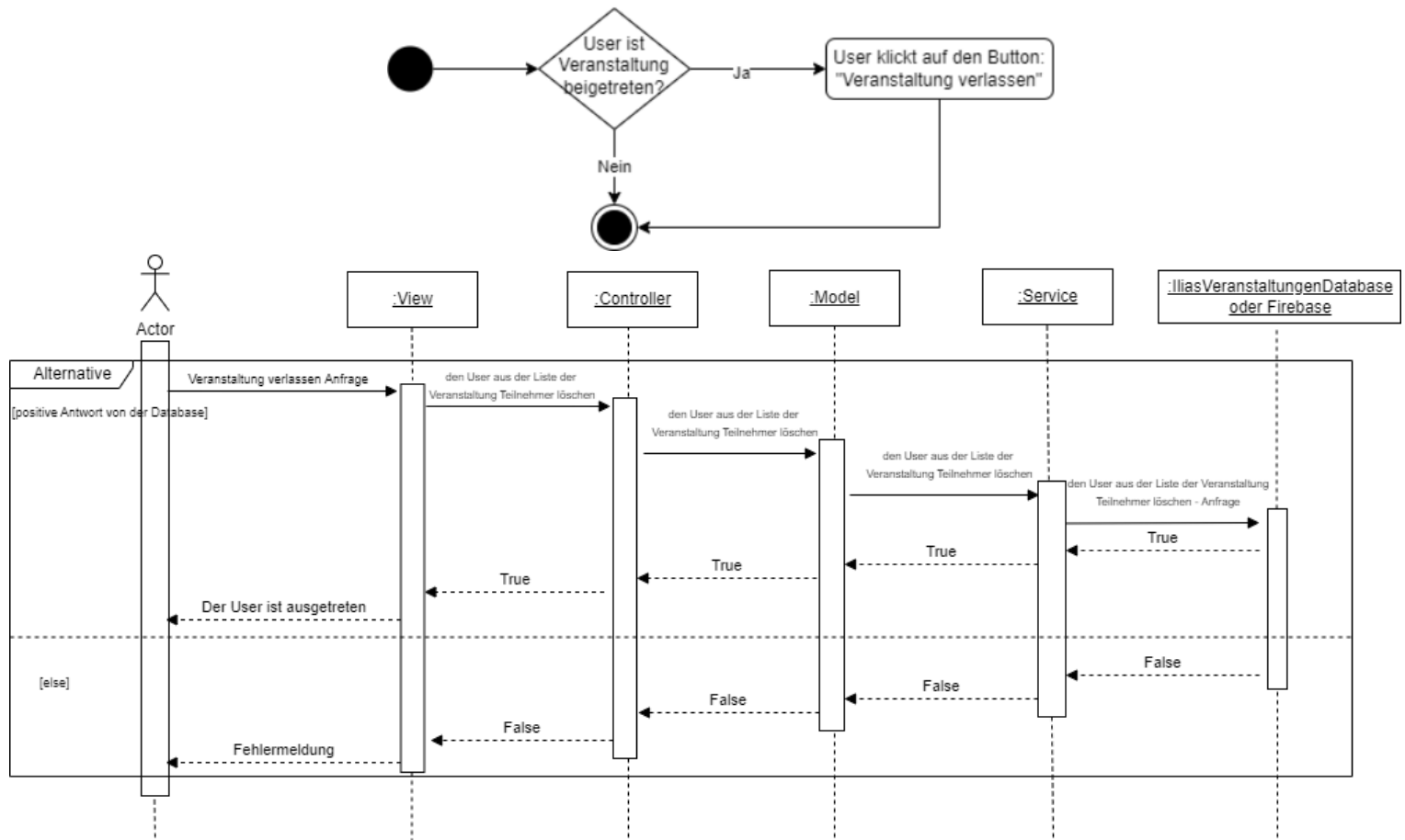
Das Sequenzdiagramm ist für "alle Veranstaltungen" angepasst, für die "beigetrete Veranstaltungen" wird es ähnlich sein: die Suchanfrage an Database wird sich unterscheiden und das Ergebnis im anderen Layout gezeigt



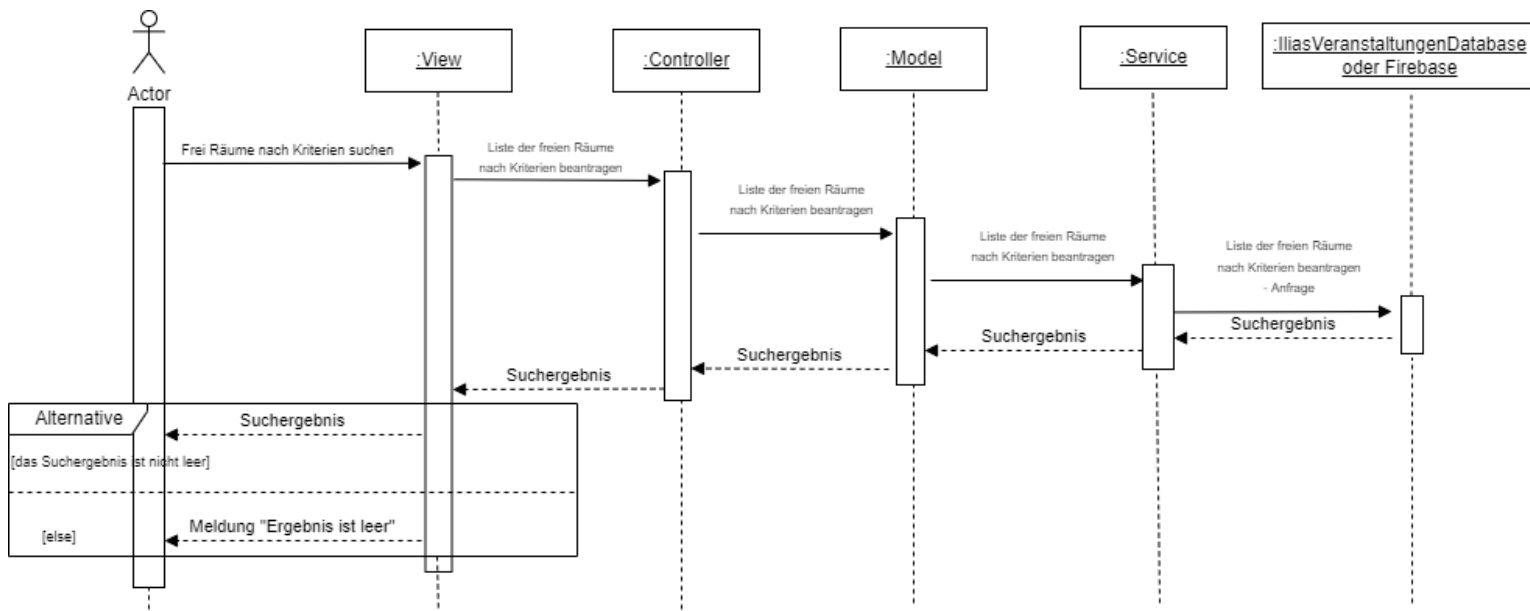
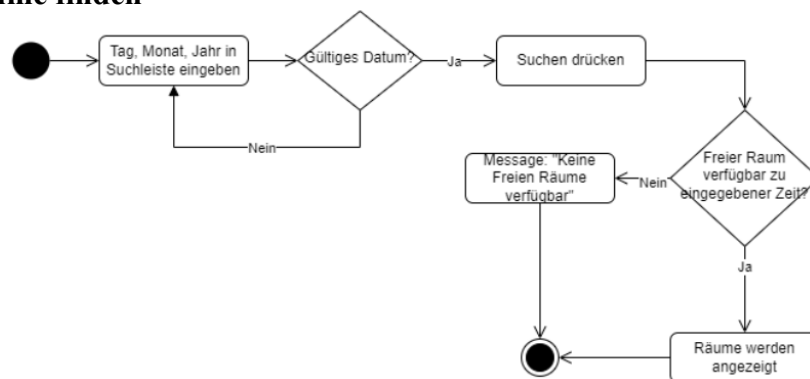
## - Veranstaltung beitreten



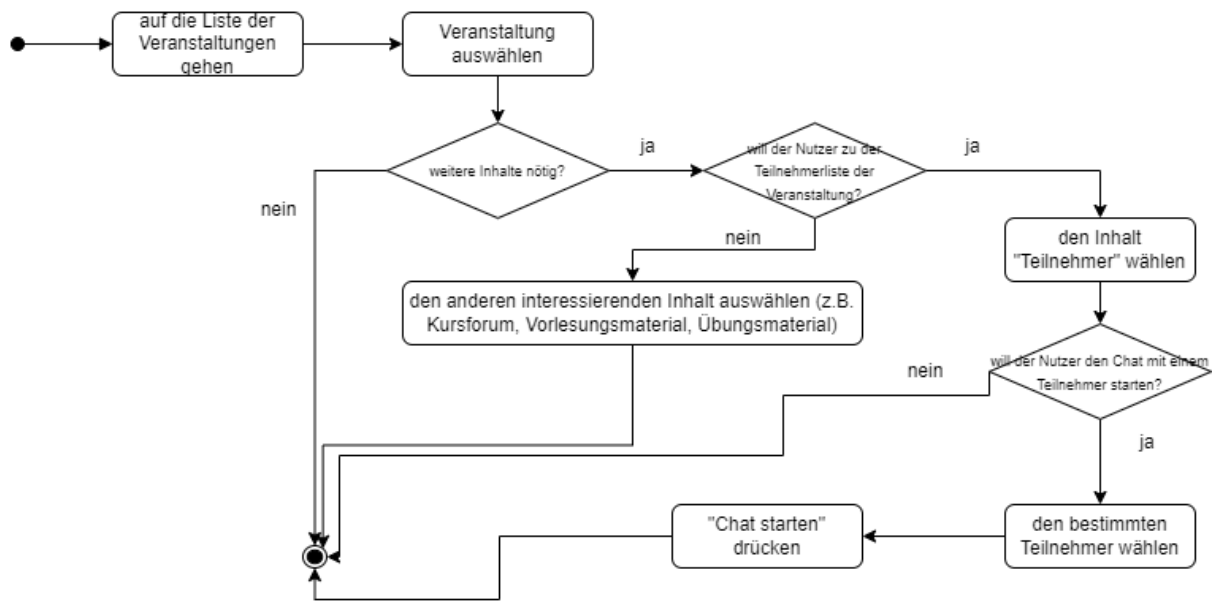
## - Veranstaltung verlassen



## - 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

