Film Plattform

Inhalt

[Planung: 1](#_Toc86745305)

[Diagramme: 1](#_Toc86745306)

[Beschreibung: 3](#_Toc86745307)

[Ablauf des Programmes: 3](#_Toc86745308)

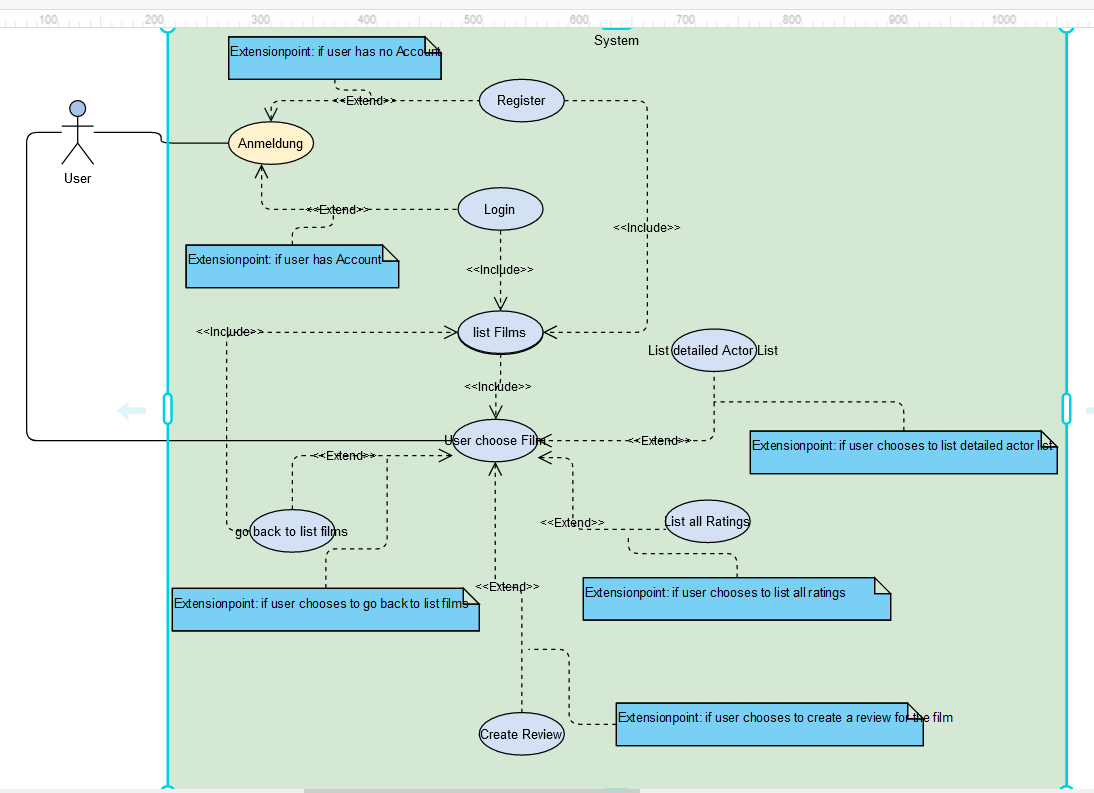
[Reflexion: 4](#_Toc86745309)

[Testing: 4](#_Toc86745310)

## Planung:

Unsere Idee war es eine kleine Filmplattform zu erstellen, in welcher man Filme auflisten-, genaue Beschreibungen anzeigen lassen- und bewerten kann. Wir haben zuerst eine Skizze auf einem Blatt Papier erstellt, um uns einen Überblick zu verschaffen. Auf der vorherig erwähnten Skizze waren die einzelnen Objekte, welche wir in unserer App haben wollten mit den jeweiligen Feldern und Assoziationen abgebildet.

### Diagramme:

UseCase:   


##### Class Diagramm:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Beschreibung:

Als erstes haben wir ein kleines Login/Register System entwickelt. Dieses haben wir am Ende jedoch noch umgeschrieben. Wir kamen zum Punkt, uns entscheiden zu müssen, wie wir die Objekte speichern wollten. Wir haben es zuerst mit JAXB probiert, dies hatte aber nicht gut funktioniert und hat uns zu viel Zeit gekostet. Schlussendlich haben wir uns für das Persistieren in einem JSON File entschieden mit JSON Simple. Wir erstellten dazu auch Manager wie zum Beispiel der JSONFileManager. Wir haben auch viel mit UUID gearbeitet. In dieser Zeit haben wir viel erstellt, viel gelöscht und viel verbessert. Es gab dann einen Durchbruch und wir kamen dann sehr gut voran. Zuletzt haben wir JAVAdoc hinzugefügt.

In actors.json und ratings.json wird einem actor/rating eine random UUID zugewiesen und Objekte wie namen vorname anzahl sterne gespeichert. In films.json wird auf diese UUID zugewiesen. So wird auch erkannt welcher Actor in einem Film mitspielt und was für Ratings ein Film hat.

So sieht es in films.json aus:¨

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ausserdem haben wir eine ObjCache Klasse in dieser wird alles aus den .json files gelesen/gespeichert. Wir haben in der genannten Klasse mehrere Methoden für die verschiedenen Möglichkeiten.

Hier als Beispiel für Actors:  
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Ablauf des Programmes:

Hier ist der Ablauf unseres Programmes: Zuerst kommt man zur Anmeldung dort kann der User, wenn er noch keinen Account hat einen erstellen oder sich einloggen. Ist der User eingeloggt werden Filme aufgelistet. Der User kann aus diesen Filmen einen auswählen. Der ausgewählte Film wird dann angezeigt also Titel Beschreibung etc. Zu diesem Film hat der User einige weitere Funktionen. Er kann eine genauere Liste der Actors anzeigen welche zum Beispiel das Geburtsdatum der Actors anzeigt und in welchen anderen Filmen dieser mitspielt. Die zweite Funktion ist alle Ratings des Filmes anzuzeigen. Dort werden die Ratings angezeigt auch mit Kommentar. Der User selbst kann auch ein Rating zu diesem Film abgeben. Die letzte Funktion ist zurück zu der Auflistung der Filme zu gehen.

## Reflexion:

Henrik: Ich habe in diesem Projekt sehr viel dazu gelernt. Ich habe zum ersten Mal mit UUID und JSON gearbeitet. Wir haben Fehler gemacht, so aber auch dazu gelernt. Ich bin sehr zufrieden mit dem Resultat.

Flurin: Ich habe JSON richtig kennengelernt und auch die Struktur eines Projektes. Ich bin auch zufrieden mit dem Resultat.

## Testing:

Am Ende haben wir alle Funktionen nochmals getestet. Sollte also Funktionieren 😉.