|  |
| --- |
| Software Design Specification  <Movie Tracker and Recommendation System>  Project Code:  CS Regular 7 (CSR7)  Internal Advisor:  External Advisor:  Project Manager:  Sir Saad Razzaq  Project Team:  Taha Hussain Shah (BSCSF18M008) (Team Lead)  Hafiz Osama Sultan (BSCSF18M023) (Team Member)  Muzammil Ashraf (BSCSF18M001) (Team Member)  Submission Date:  10-01-2022  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Project Manager’s Signature |

**Document Information**

| Category | Information |
| --- | --- |
| University | UOS |
| Project | < Movie Tracker and Recommendation System> |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Identifier |  |
| Status | Final |
| Author(s) | <Names of all the authors of this document> |
| Approver(s) | PM |
| Issue Date | Jan. 10, 2022 |
| Document Location |  |
| Distribution | 1. Advisor 2. PM 3. Project Office |

**Definition of Terms, Acronyms and Abbreviations**

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

| Term | Description |
| --- | --- |
| ASP | Active Server Pages |
| DD | Design Specification |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table of Contents

[Software Design Specification 1](#_Toc93189278)

[1. Abstract 4](#_Toc93189279)

[2. Background and Justification 4](#_Toc93189280)

[3. Project Methodology 4](#_Toc93189281)

[4. Introduction 5](#_Toc93189282)

[4.1 Purpose of Document 5](#_Toc93189283)

[4.2 Project Overview 5](#_Toc93189284)

[4.3 Scope 5](#_Toc93189285)

[5. Overall System Description 5](#_Toc93189286)

[5.1 User characteristics 5](#_Toc93189289)

[5.2 Operating environment 5](#_Toc93189290)

[5.3 System constraints 6](#_Toc93189291)

[6. External Interface Requirements 6](#_Toc93189292)

[6.1 Hardware Interfaces 6](#_Toc93189293)

[6.2 Software Interfaces 6](#_Toc93189294)

[7. Functional Requirements 6](#_Toc93189295)

[8. Non-functional Requirements 7](#_Toc93189296)

[8.1 Performance Requirements 7](#_Toc93189297)

[9. Design Considerations 8](#_Toc93189298)

[9.1 Assumptions and Dependencies 8](#_Toc93189299)

[9.2 Risks and Volatile Areas 8](#_Toc93189300)

[10. System Architecture 8](#_Toc93189301)

[10.1 System Level Architecture 8](#_Toc93189302)

[10.2 Dataflow Diagram: 9](#_Toc93189307)

[11. Detailed System Design 10](#_Toc93189308)

[11.1 Class Diagram: 11](#_Toc93189309)

[11.2 Sequence Diagram: 13](#_Toc93189310)

[11.3 Use Case Diagram: 14](#_Toc93189311)

[11.4 Entity Relationship Diagram: 21](#_Toc93189318)

[12. Cost Estimation Using Cocomo Model: 21](#_Toc93189320)

[13. References 23](#_Toc93189321)

# Abstract

Humans always tend to need some kind of entertainment in their life. Sometimes that entertainment is in the form of movies. But when you decide to watch a movie you often can’t find the right movie to watch. To solve that problem we suggest a movie recommendation system. This recommendation system will recommend its users movies that are according to their interests or the movies that their friends or peers are watching.

Our system will help users in finding the movies that they want to watch and will like. To recommend movies to users according to their interest we will gather data on interests of the users and then recommend the movies according to that. Another approach that we will take in recommending the right movies to our users is by looking at the interests of their friends and what kind of movies they watch.

# Background and Justification

A recommender system is a simple algorithm whose aim is to provide the most relevant information to a user by discovering patterns in a dataset. The algorithm rates the items and shows the user the items that they would rate highly. An example of recommendation in action is when you visit Amazon and you notice that some items are being recommended to you or when Netflix recommends certain movies to you or they are used by music streaming applications such as Spotify.

Machine learning algorithms in recommender systems typically fit into two categories: content-based systems and collaborative filtering systems. Modern recommender systems combine both approaches.

Content-based methods are based on the similarity of movie attributes. Using this type of recommender system, if a user watches one movie, similar movies are recommended. For example, if a user watches a comedy movie starring Adam Sandler, the system will recommend them movies in the same genre or starring the same actor, or both. With this in mind, the input for building a content-based recommender system is movie attributes. With collaborative filtering, the system is based on past interactions between users and movies. With this in mind, the input for a collaborative filtering system is made up of past data of user interactions with the movies they watch.

# Project Methodology

The proposed system is a web-based system in which when a user opens up the website. Users can see and scroll between the movies shown on the screen. By hovering over the movie icon option of reviewing and rating of the movie is visible to the user and further details about the specific movie are available to the user after clicking on the image icons. Furthermore, different people's reviews will be shown to the user and movies can also be recommended on the basis of the celebrity that the user has already liked.

To achieve our goals following an action plan and steps we will adopt to build up our comprehensive and responsive website.

1. The front-end of the system will be first created in ReactJs.
2. The middle wear will be required to construct a live working website through which we connect the back end with the front end. Middlewear will be NodeJS.
3. Data will be fetched from a database constructed in the MongoDB database environment.
4. For the recommendation machine learning will be used to suggest to the user what they might like to watch.
5. One of the most common ways to build a recommendation system is to use Python Machine Learning. Python offers probably the most popular and powerful interpreted language. So, python will be used for the recommendations.
6. Users can perform multiple things within a single website.

# Introduction

## 4.1 Purpose of Document

Purpose of the document is to explain every single detail of the project to the reader. So, whenever a new person wants to know about the idea of the project that we are going to do. The reader can just read out the introductory part of the document and he will get the whole overview of the project and why and what is going to be performed in this project. This document will provide details about what our project will be about what resources we will be needing. What are the possible constraints in our project? For example if we have a limited amount of time and or a low budget available by our client. Who is going to interact with our system?

## 4.2 Project Overview

The proposed system will help users in keeping track of the movies they are watching or thinking of watching and will help users by recommending movies of their interests. Sometimes when a person is looking for some entertainment they turn to things like movies and dramas. Our system will help user in doing that.

## 4.3 Scope

1. Films can be added to watch list
2. Film can be added to watched list
3. User can give ratings
4. User can add reviews
5. User can view a films ratings and reviews
6. Users can select different genres and interests to see films.
7. Users can be recommended film on the basis of the people they follow.

# Overall System Description



## User characteristics

User class that will use our system can be from the age of 15-50 years or more. The most important user class for our system will be teenagers or people who are looking for some kind of entertainment in the form of movies.

## Operating environment

System will operate on all kinds of devices which support an internet connection and a web browser since our system will be web based.

## System constraints

Identify any constraints or limitations on the system. Constraints may include the following:

* There are not really any hardware constraints for our system. Only hardware constraint might be availability of an internet connection to our website/system.
* Cultural constraints will include limited language support as our system will only support English language.

# External Interface Requirements

## Hardware Interfaces

There is no hardware interface in our system. The system that we are going to construct is purely web-based and has no such hardware interface with which the user can interact.

## Software Interfaces

When the user opens up the website first thing that he will be going to interact with is the page that shows different categories of the movie. User can scroll down between the different option provided on the main page screen but if the user wants to add the movie to the watch list and want to review any movie and to check the reviews of the different people about a particular movie that he/she is interested in. for that user has to login with us if the user is already registered. Otherwise, the user has to first register him and then can access more features of the website.

# Functional Requirements

All the possible modules that our system will be going to have are listed below step by step. These modules will explain the different interfaces with which the user is going to interact and a bit details about each module on our webpage.

1. **Login/Signup:**

Two things that the user is going to see when he first opens up the website are login and signup options. Users will either already our registered members so there will be no need for the user to sign up again. Otherwise, the user first has to sign up and give credentials and register himself and then can access the other features of the website.

In the login section if the user forgets the email id or password there will be the option of forget button user can click on that and then change the password by using the email that he entered at the time of the creation of the account. From there password can be changed.

1. **Recommendation screen:**

This screen of the webpage will show the movies according to user liking and disliking, recommendations will be given either by collaborative filtering or content-based filtering. Users can select any of the recommended movies and can go to the next page if there are more recommendations according to user taste.

1. **Detailed movie screen:**

After the recommendation screen when the user selects or clicks on any of the movies next screen will be of detailing. On this screen, everything about the movie is displayed. For example, the cast (cast names), genre, movie title, movie rating, trailer, release date, and a button will be provided if the user wants to add the opened movie to the watch list so that the user can watch it later.

1. **Watch-list screen:**

All of the movies that the user has added to the watch list will be shown in this area.

1. **Search Bar:**

Users can search for any movie about which he has to get some details. Movies will be searched by the title of the movie and then a detailed screen can be opened if the movie of that title is found.

1. **Rating & Reviewing:**

Rating and review feature will be provided in the detailed screen where user can rate the movie and can give personal reviews about the movie for giving review to the movie user must first login to the website to access this feature. Rating will be given out of 5 stars. Where user can select any number between 1-5. Reviews of the other people about the will also be provided below the details of the movie.

1. **Sentimental analysis of user review:**

After the user has given his review. We will use sentiment analysis api to evaluate the sentiments of the user through their review.

1. **User profile:**

In the user dashboard user can edit the basic credential can change the profile picture, change the name, and all the basic information except email provided during the creation of an account.

# Non-functional Requirements

## Performance Requirements

Following bullet points show the performance requirements of our project.

● The server shall be capable of supporting no less than 200 concurrent connections.

● It will provide no limit on how many users are currently using the system.

● There must be alternate server in case of some server breakdown.

● Speed of the website is the main focus as waiting during moving from different module is not a good user experience

● Rating and reviews from different people must be shown correctly and timely to the user.

# Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution. In other words, this section is used to formally set the groundwork for the system design.

## Assumptions and Dependencies

Assumptions and dependencies for the system and project are already captured in the FS document. This section should not repeat those issues. Instead it should deal with previously stated issues in the context of design, if appropriate, or bring up new issues that are only relevant to design.

## Risks and Volatile Areas

Discuss the most likely sources of change and risks (new requirements, technology, etc.) that would impact the design of the system. If appropriate, describe how the system will be designed to allow timely response to changes or what the contingency path is for changes.

# System Architecture

This section should provide a high-level overview of how the functionality and responsibilities of the system are partitioned and then assigned to subsystems or components. The main purpose is to gain a general understanding of how the system is decomposed, and how the individual parts work together to provide the desired functionality.

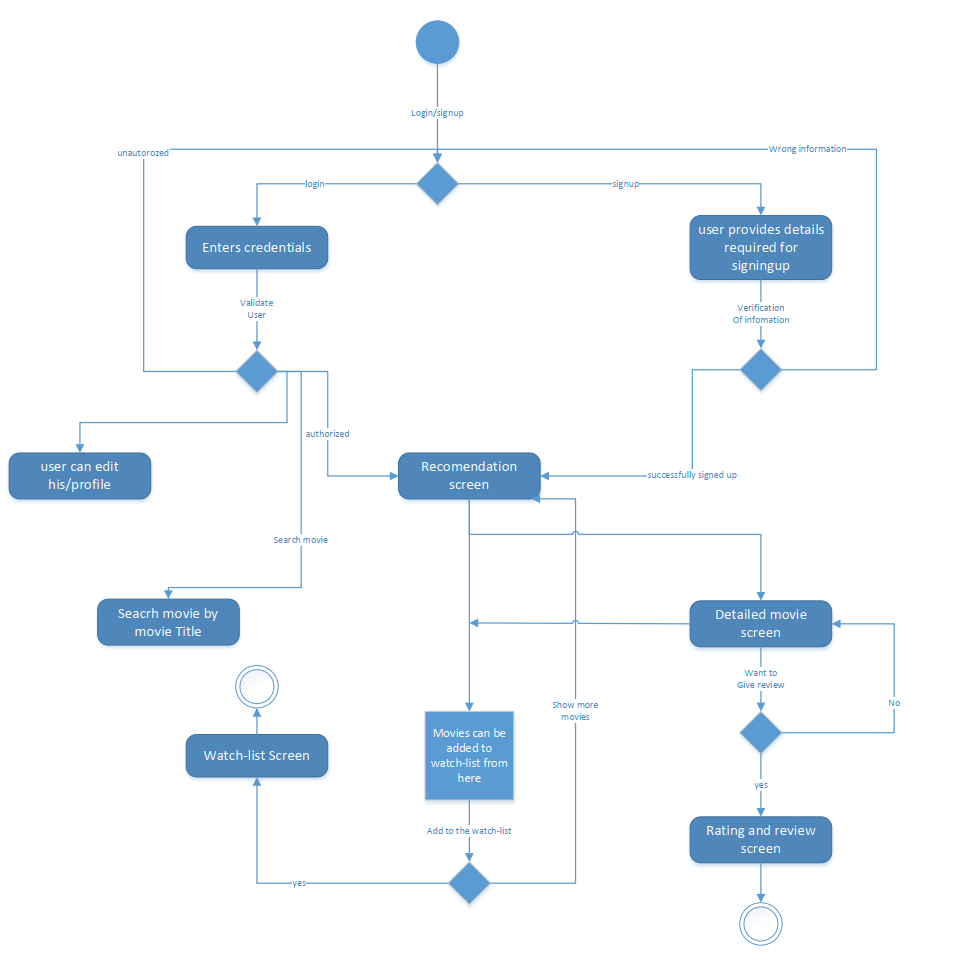
## System Level Architecture

The architecture should decompose the system at a top level in a way that provides a foundation for more detailed design work. The architecture discusses relationships and roles of system elements (subsystems, components, modules, etc.), but does not provide internal details. Areas for consideration are:

* System decomposition into elements
* The relationship between the elements
* Interfaces to external systems
* Major physical design issues such as where elements will execute
* Global design strategies such as error handling



## Dataflow Diagram:



Figure

# Detailed System Design

A detailed design should include the following:

* Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.
* Detailed Sequence diagram with parameter list
* State Transaction Diagram
* Logical data model (E/R model)
* Physical data models
* Detailed GUIs

## Class Diagram:

Figure

**Class Diagram Descriptions:**

* **User Class:**
* **Attributes:**

1. ID
2. Name
3. Email
4. Password

* **Methods:**

1. User will be able to create their name and email.(createName() , createEmail())
2. User will be able to verify their login credentials.(verifyLogin())
3. User can change their email or password.(changeEmail(), changePassword())

* **List Class:**
* **Attributes:**

1. ListType

* **Methods:**

1. User will be able to view their lists whether recommended list or watch list.(viewList())

* **Movie Class:**
* **Attributes:**

1. Name
2. Rating
3. Review
4. Watched

* **Methods:**

1. User will be able to give their review and rating.(giveRating(), giveReview())
2. User can add a movie to their watch list.(addToWatchlist())
3. User can add a movie to their watched list.(watched())
4. User can view detailed movie screen.(viewDetailedScreen())

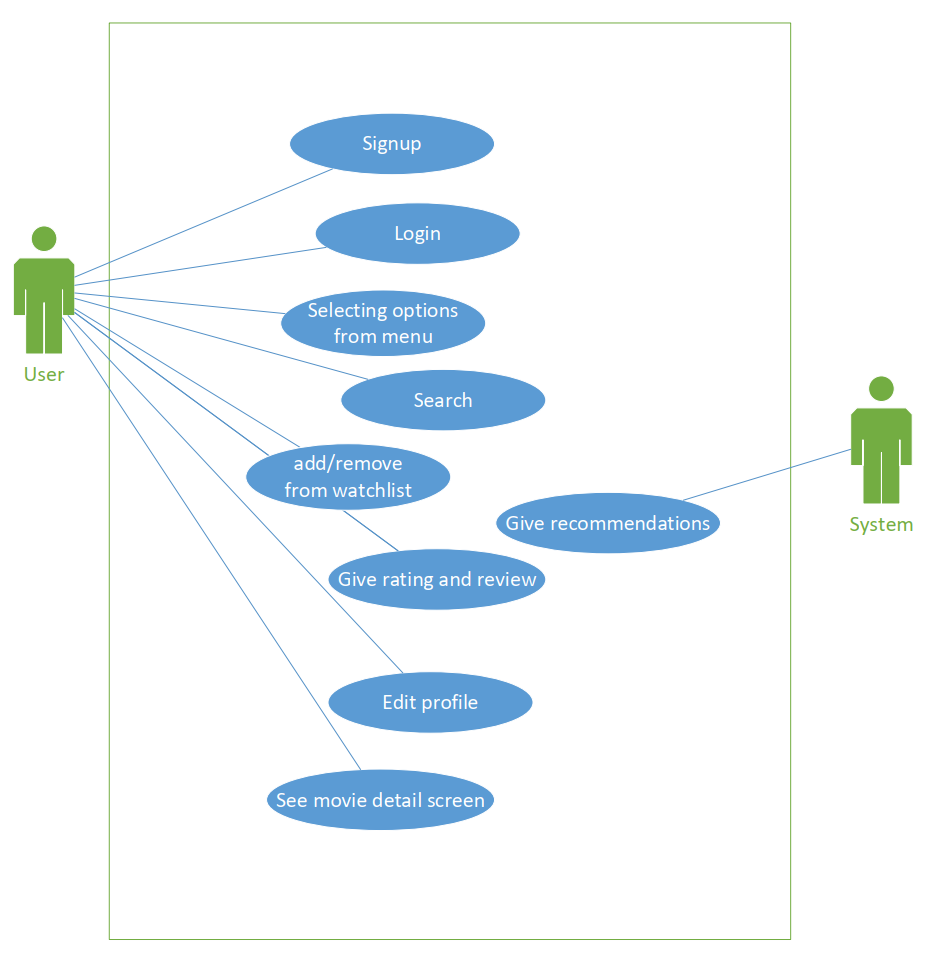
## Screenshot (17) **Sequence Diagram**:

Figure

**Sequence Diagram Description:**

In this we have a single actor that is our user. So, user can either be already registered member or new user. The alternative box is used firstly in case user is already a member then he/she directly login into the system and alternatively if new user then has to go through the sign up process and then can login into the system. Once login is done then user can access different screen and perform the function showed in the sequence diagram.

## Use Case Diagram:



**Use Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 1: Signup** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Signup | | | | |
| **Use case Id:** | | 01 | | |
| **Pre-condition:** | | User needs to open the website | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | User enters username | | | Software checks availability of username |
| **2.1** | User enters email | | | Software checks validity or pattern of email |
| **3.2** | User enters password | | |  |
| **4.n** | User re-enters password to confirm password | | | Software checks if both passwords are matching |
| **4.** | User clicks on signup button | | | User is taken to the login screen after successful account creation |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** | User has successfully signed up and now can login to their account | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | This use case uses login use case(id:02) after its completion | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 2: Login** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Login | | | | |
| **Use case Id:** | | 02 | | |
| **Pre-condition:** | | User needs to open the website | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | User enters username or email | | | Software checks validity or pattern of email |
| **2.2** | User enters password | | |  |
| **3.n** | User clicks on login button | | | Software checks credentials and authenticates |
| **4.** | User can also click on forgot password | | | Software will take user to a new screen where they can enter their username or email to reset password by getting verified |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
| **3a: If the user credentials are right they will be taken to the home screen of website**  **3b: If the users credentials are wrong. They will be notified and reset password** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** | User has successfully logged in and now can access other features of the website | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 3: Selecting different options from home screen** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Home screen menu | | | | |
| **Use case Id:** | | 03 | | |
| **Pre-condition:** | | User needs to be logged in to their account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | User can see various features from their home screen like seeing watch list, recommendations tab, search button, user profile | | |  |
| **2.2** | User can see recently released list of movies here on this screen | | |  |
| **3.** | User can click on any movie to see the detail about it | | | If user clicks on any movie they will be taken to detailed screen of the movie |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** |  | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 4: Searching movies** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Search | | | | |
| **Use case Id:** | | 04 | | |
| **Pre-condition:** | | User needs to be logged in to their account | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | User can click on search bar and type name of the movie to find detail and information about the movie | | | After clicking on a certain movie user will be taken to the detailed movie screen |
| **2.2** |  | | |  |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** | User will be on detailed movie screen | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 5: Adding or removing movies from watch list** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Watch list | | | | |
| **Use case Id:** | | 05 | | |
| **Pre-condition:** | | User needs to be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | Users can add movie to their watch list from either home screen or by searching or from recommendation screen. | | | Software will add the movie to database of watch list against the user’s profile. |
| **2.2** | Users can remove movie from their watch list from either home screen or by searching or from recommendation screen. | | | Software will remove the movie from database of watch list against the user’s profile. |
| **3.** | User can add the movie to their watched list | | | Software will add the movie to database of list of watched movies against the user’s profile. |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** |  | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 6: Seeing movie details** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Movie detail | | | | |
| **Use case Id:** | | 06 | | |
| **Pre-condition:** | | User needs to be logged in and has clicked on a specific movie to see it details | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | Users can here see details about movies like their synopsis, cast, rating and reviews of movie | | |  |
| **2.2** |  | | |  |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** |  | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 7: Providing rating and review** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Rating and reviewing | | | | |
| **Use case Id:** | | 07 | | |
| **Pre-condition:** | | User needs to be on detailed movie screen | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | If the user has watched the movie they can give rating out of 5. | | |  |
| **2.2** | If the user has watched the movie they can give review of the movie | | |  |
| **3.** | User will be shown net review based on their review after sentimental analysis of their feedback | | |  |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** |  | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 8: Editing profile** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Profile settings | | | | |
| **Use case Id:** | | 08 | | |
| **Pre-condition:** | | User needs to be on profile settings screen | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | User can change their profile picture | | | All these changes made will be edit user attributes in the database against user’s profile. |
| **2.2** | User can change their email address | | |
| **3.** | User can change their password | | |
| **4.** | User can add details about themselves in their bio | | |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** |  | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

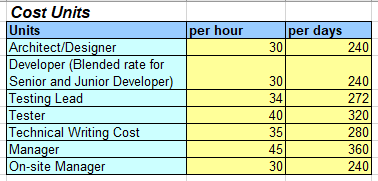
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case 9: Getting recommendations** | | | | |
| **Actors:**  User | | | | |
| **Feature:** Recommendations | | | | |
| **Use case Id:** | | 09 | | |
| **Pre-condition:** | | User needs to be on the recommendation screen | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.1** | User can select different genres to get movie recommendations through content filtering | | |  |
| **2.2** | Users will also see list of recommended movies through collaborative filtering | | |
| **Alternate Scenarios:** Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability. | | | | |
|  | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1.1** |  | | | |
| **2** |  | | | |
| **n** |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |
| **User Interface reference** | | | List user interface(s) that are related to this use case. Use numbered list in case of more than one user interface elements. | |

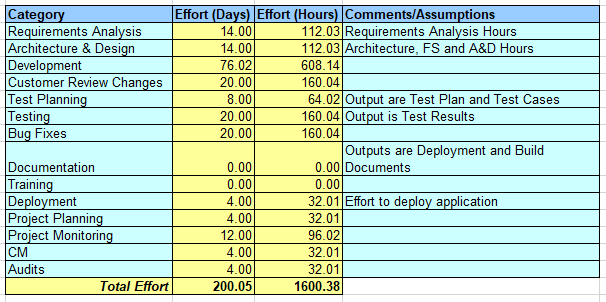


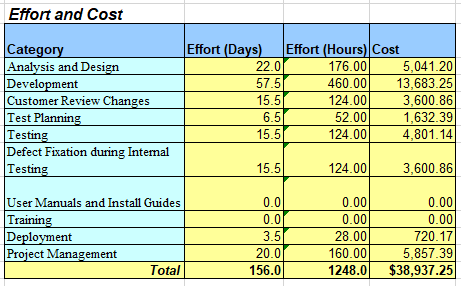
## Entity Relationship Diagram:

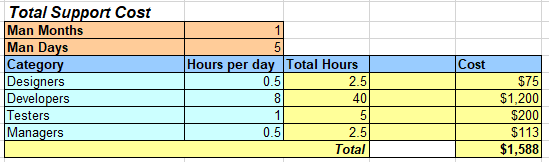
## 

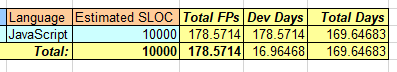
# Cost Estimation Using Cocomo Model:











# References

| Ref. No. | References |
| --- | --- |
| 1 | <https://www.rottentomatoes.com/> |
| 2 | <https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/> |
| 3 | <https://www.sciencedirect.com/science/article/pii/S1110866515000341> |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |