

PROM02 Computer Master's Project

Movie Recommender System

Design Document

Name: Chu Siu Kay Alan

Student No: 189222006

Version: 1.0

Date: Feb 19, 2021

1 Purpose

This purpose of this document is to illustrate the design of the movie recommender system.

2 Architecture Design

This section discusses the architecture design of the movie recommender system. Figure 2.1 shows the proposed architecture for the production movie recommender system.

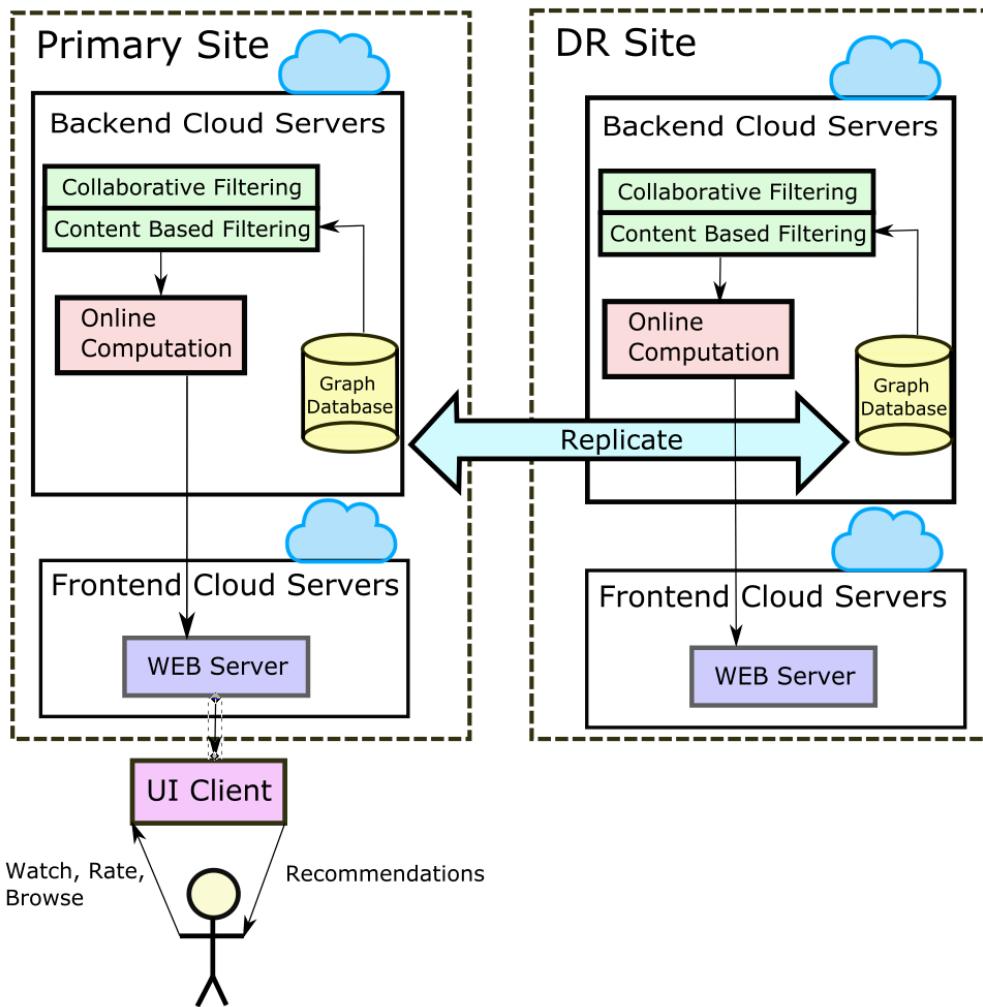


Figure 2.1 Architecture of the production system

In order to satisfy the resilience and availability requirements, the proposed architecture design consists of a primary site and a disaster recovery (DR) site. The data is synchronised from primary to DR in real time to guarantee no data lost after failover. In the frontend, there is a WEB server to serve the WEB application to the customers. When the frontend servers need to obtain the recommendations, it will request the backend server to perform the online computation by applying both collaborative filtering and content based filtering to the graph database. Once the recommendations are generated, the result will be transferred back to the frontend server. The online computation module is also responsible for updating the graph database so that the graph data can be kept up to date all the time. For example, the customer may rate a new movie during the session. The online computation will re-compute the recommendation based on the updated data. This proposed design is inspired by the architecture design used in Netflix (Xavier A., 2019).

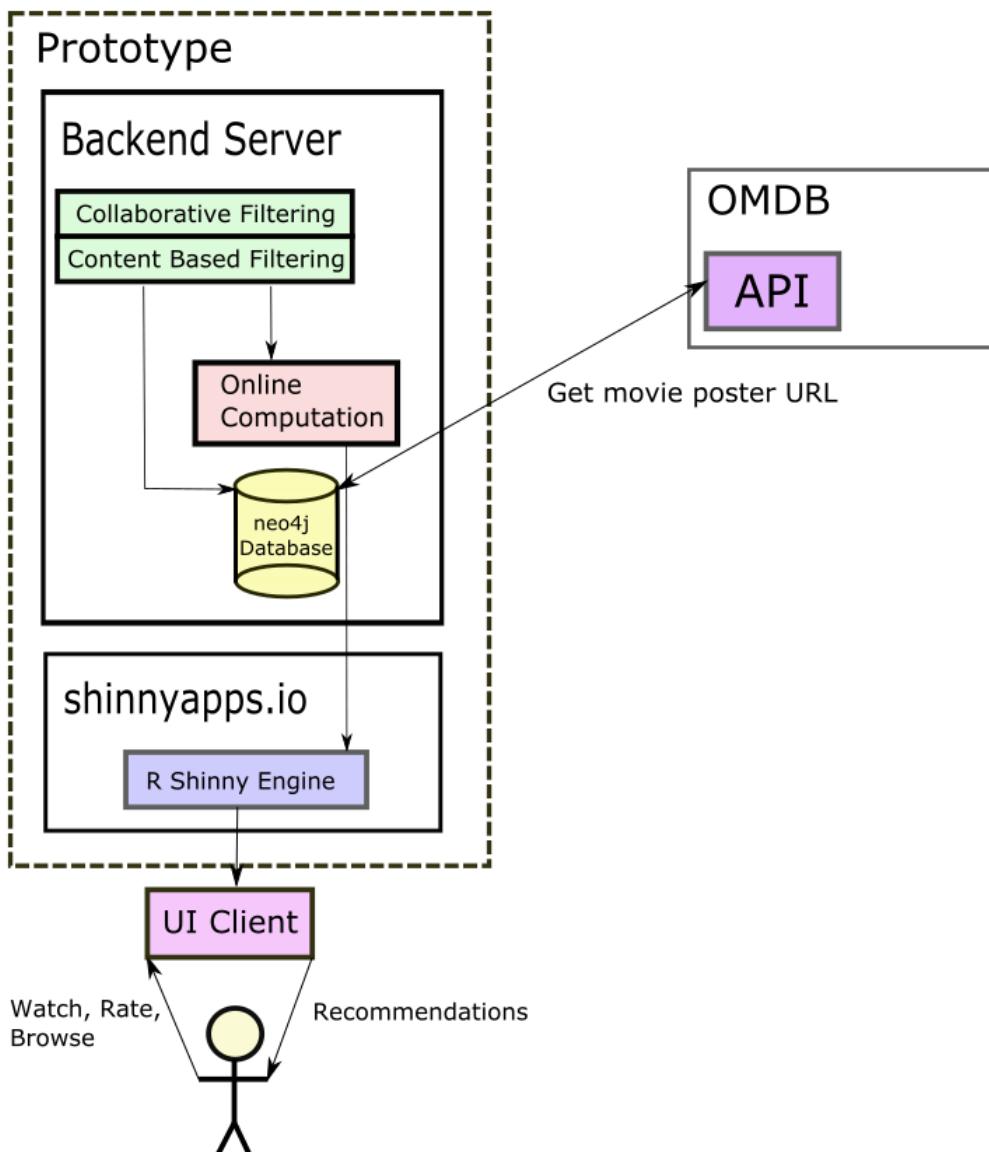


Figure 2.2 Architecture of the prototype

As the prototype is to mainly demonstrate the functional requirements of the movie recommender system, the architecture of the prototype is simplified as shown in Figure 2.2. Since the original data set does not include the movie poster images, a third party API provided by OMDB is used in the backend server to obtain the URLs retrieving the movie poster images. The URLs are then loaded to the Movie objects in the graph database. The R Shiny server connects directly to the backend server to access the neo4j graph database.

3 Schema Design

A neo4j database is used as the backend database for storing the movie, user and review data in a graph representation. Figure 3.1 shows the meta graph of the database. The meta graph represents the labels and relationship-types defined in the database. From it, one could study how the labels are connected. Please refer to (neo4j, 2020) for more information about meta graphs.

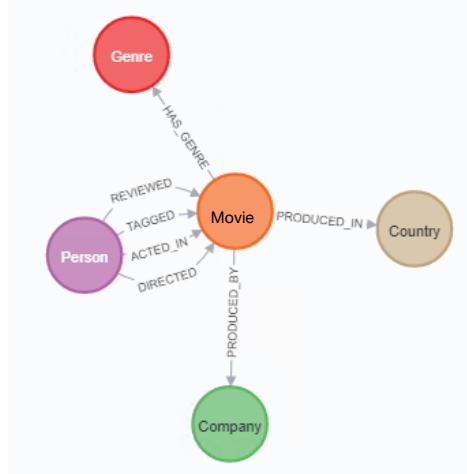


Figure 3.1 Meta graph

As shown in Figure 3.1, there are 5 labels defined. They are Movie, Person, Genre, Company and Country. The Movie label is for labelling nodes that contain movie properties such as movie title, production year, etc. The Genre label is for labelling nodes that contain movie genres. The purpose of genres is to categorise movies into different types such as action, comedy, drama, etc. The Country label mainly serves as indicating which country the movie is produced in. The Company label is for labelling nodes that contain information of the movie production companies. The Person label is used to label any nodes that represent persons. A person can have different relationships with the movie. He/she can be an actor, a director or just a user who has reviewed a particular movie. Figure 3.2 shows the different roles a person could be.

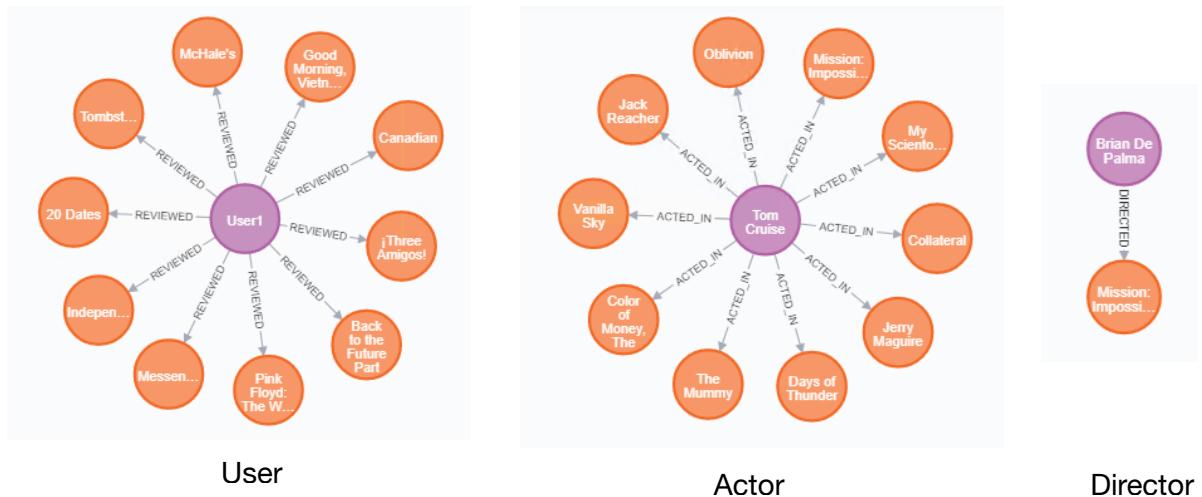


Figure 3.2 Different relationships between persons and movies

Table 3.1 below further describes the nodes and relationships by showing their attributes defined.

Nodes	
Movie	
<id>	neo4j internal ID
movielid	Movie ID to uniquely identify a movie
title	Title of a movie
year	Production year of a movie
poster	Poster URL
imdbId	Movie ID in IMDb (for linkage to IMDb)
tmbdId	Movie ID in TMDb (for linkage to TMDb)
Genre	
<id>	neo4j internal ID
name	Name of the person
Company	
<id>	neo4j internal ID
name	Company name
Country	
<id>	neo4j internal ID
name	Country name
Person	
<id>	neo4j internal ID
name	Name of the person
loginId	Movie Recommendation System Login ID of the person
Relationships	
REVIEWED	
<id>	neo4j internal ID
rating	Movie rating given by a user
review_date	Date of the movie review
timestamp	UTC timestamp of the movie review
ACTED_IN	
<id>	neo4j internal ID
DIRECTED	
<id>	neo4j internal ID
HAS_GENRE	
<id>	neo4j internal ID
PRODUCED_BY	
<id>	neo4j internal ID
PRODUCED_IN	
<id>	neo4j internal ID

Table 3.1 Attribute of nodes and relationships

4 User Interface Design

This section discusses the user interface design of the movie recommender system.

4.1 Overall User Interface Design

The movie recommender system has three pages in total. The first page is the movie recommendation page which is accessible by customers. This page shows all the movies recommended by the system once the customers have logged in to the system.

There are five rows of movies listed in the page. The first row displays the ten most popular movies. The second row displays the ten most recent movies rated by the users. The five stars displayed underneath the movie posters indicate the rating the users have given. The third row displays the top ten movies that are similar to the users' most favourite movies. The stars displayed underneath the movie posters indicate the average rating of the movies. This row of movies are actually recommended by using the content-based recommendation algorithm. The similarity between the recommended movies and the user's favourite movies is calculated by using the Jaccard Similarity algorithm based on the common properties found in all types of relationships including REVIEWED, ACTED_IN, DIRECTED, HAS_GENRE, PRODUCED_BY and PRODUCED_IN. The fourth row displays the movies liked by the people who are similar to the user. This row of movies are recommended by using the collaborative filtering algorithm. The similarity between the user and the others are calculated using the Pearson Similarity algorithm based on the ratings given on the common movies that they have reviewed. The fifth row shows the movies that are acted by the actors or actresses that appear in the user's favourite movies.

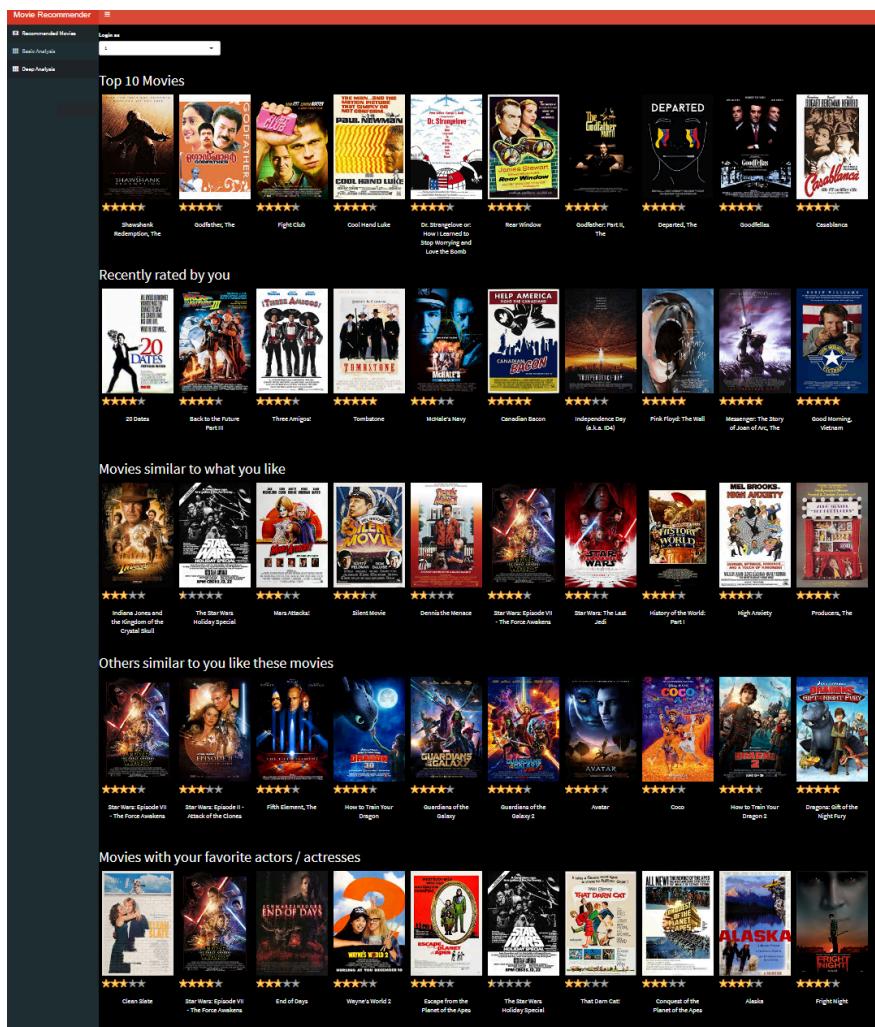


Figure 4.1 Movie Recommendation Page

Besides the movie recommendation page, there are two maintenance pages which are accessible only to staff. These pages allow the staff to perform analysis on the movie data and the movie recommendations generated by the system. The first maintenance page as shown in Figure 4.2 provides some basic analysis tools to staff for analysing the movie data.

The first section displays the top ten movies that are rated within the specified year range and have specified minimum number of ratings given by the users. The second section displays the Word Clouds of movie genres and movie tags of the movies that were produced within the given year range and possess of the specified minimum average rating. Besides, the staff could adjust minimum frequency and maximum number of words to control the amount of information to be displayed. The third section shows the histograms of the movie ratings and average movie ratings. The last section shows the average number of ratings given by the users each year.

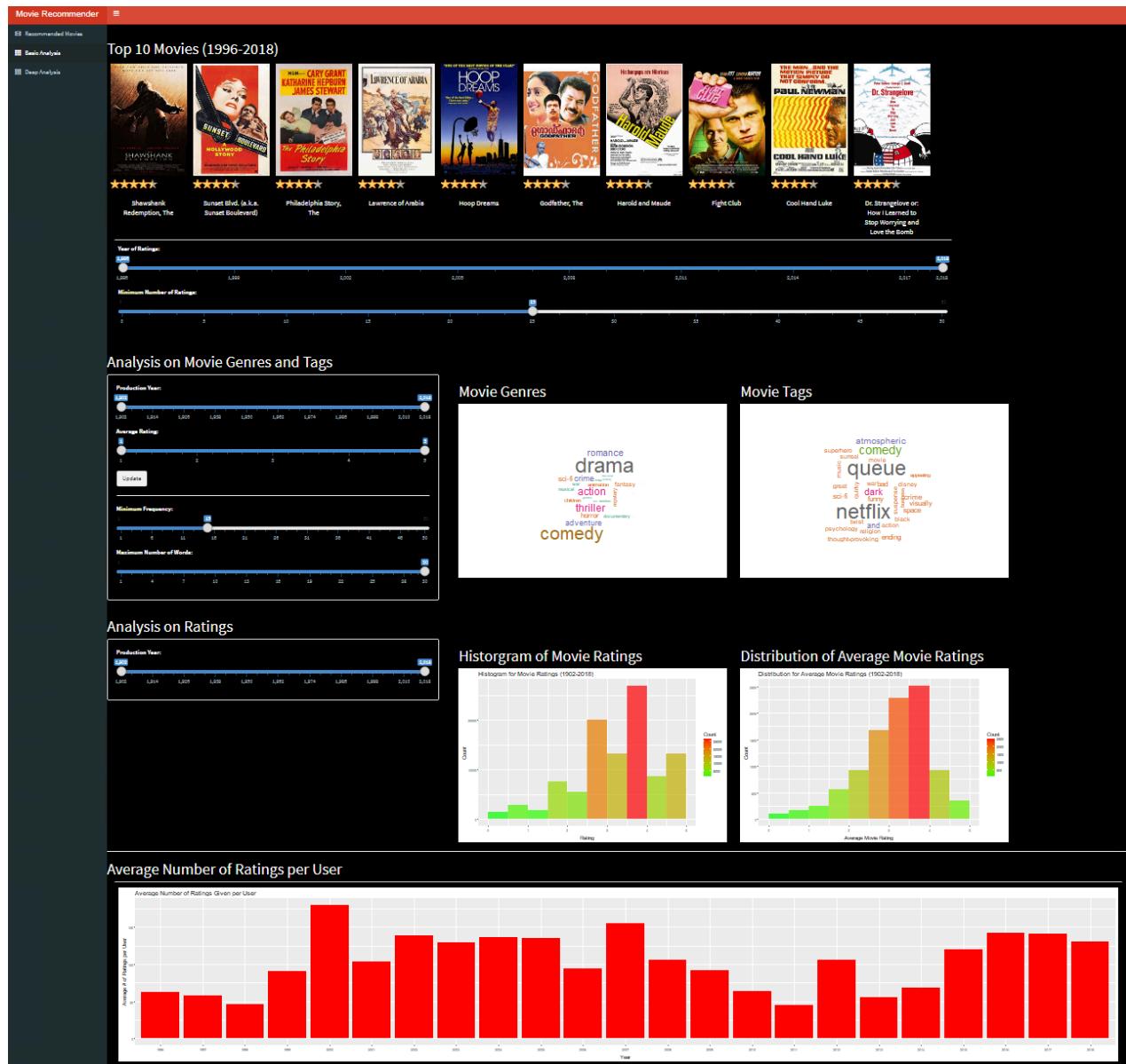


Figure 4.2 Basic Analysis Page

The second maintenance page allows staff to dive in deeper to do analysis on the recommendations suggested by the system. This page allows the staff choose a particular recommended movie by selecting login ID, recommendation category and the movie that he / she would like to do the analysis on. Once the criteria have been chosen, a graph that describes how the system suggests the recommendation will be displayed.

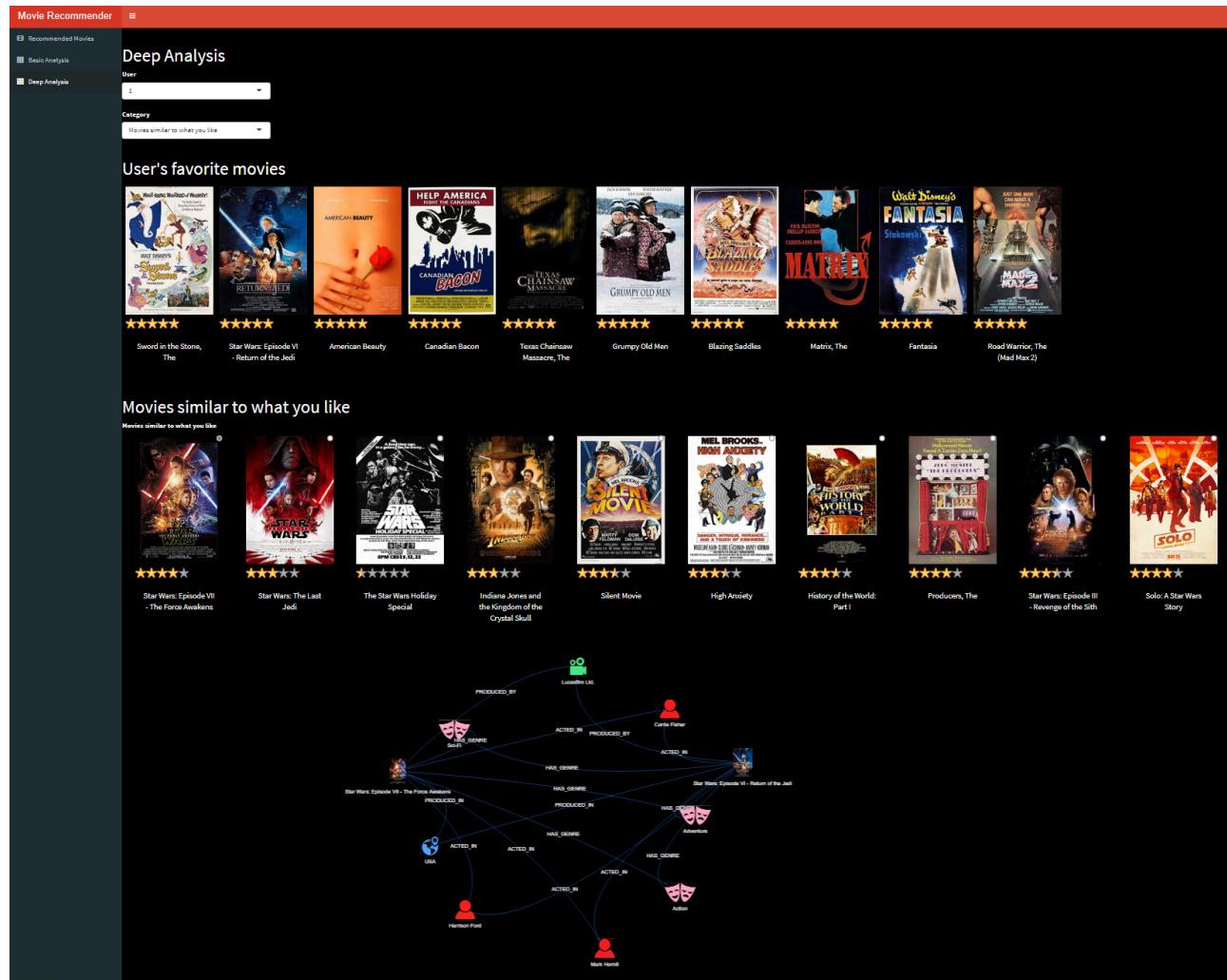


Figure 4.3 Deep Analysis Page

4.2 User Interface Design Details

This section discusses the functionality of the features provided by each page in details.

User Story ID	As A <Type of user>	I want to <perform some task>	So that I can <achieve some goal>
C1	As a customer	I want to be given a set of recommendations based on the ratings that I have given in the past.	So that I can select the movies that I like without much navigation.
C2	As a customer	I want to be given a set of recommendations based on the popularity of the movies.	So that I can select the movies that I like without much navigation.
C3	As a customer	I want to be given a set of recommendations based on the similarity of the movies that I liked before	So that I can select the movies that I like without much navigation.
C4	As a customer	I want to see the recent movies that I have rated with my ratings	So that I can be reminded which movies I have rated with good ratings or bad ratings
C5	As a customer	I want to see the average ratings of the recommended movies	So that I can have an ideas how the others rated the recommended movies
C6	As a customer	I want to be given a set of recommendations based on what the others similar to me like	So that I can select the movies that are liked by the people who are similar to me
S1	As a staff	I want to check the top 10 movies at a specific period. Also the number of ratings given has to be at least equal to the minimum number of ratings specified	So that I can know the top 10 movies that are popular at what period of time
S2	As a staff	I want to check which movie genres and tags are popular / unpopular	So that I can determine which types of movies the company should upload to the system
S3	As a staff	I want to check the distribution of the movie ratings	So that I can have an ideas what ratings are generally given by the customers
S4	As a staff	I want to check the number of ratings given per customer for each year	So that I can have an ideas how active customers are regarding to rating movies
S5	As a staff	I want to see the reasoning behind the recommendations	So that I can determine whether the recommendations make any sense

Table 4.1 Functional Requirement

Table 4.1 shows a set of user stories which describe the functional requirement of the movie recommender system.

4.2.1 Recommendation Page

This section shows how the features in the recommendation page satisfy the functional requirement defined in the Table 4.1.

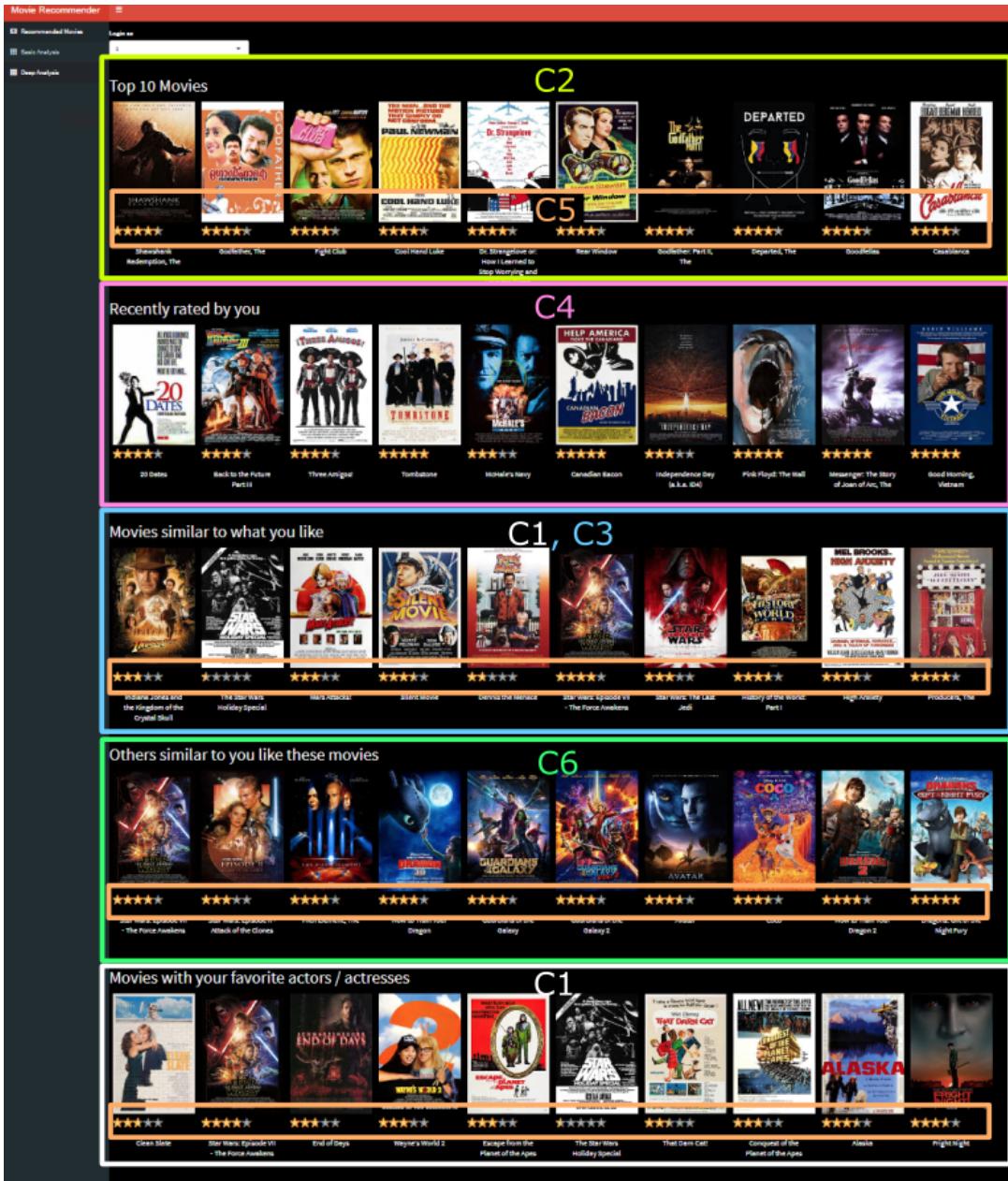


Figure 4.1

As shown in Figure 4.1, the areas that satisfy different functional requirements are visually highlighted in boxes in different colours. Each colour denotes a user story. The recommendation page contains 5 rows of movie recommendations. The first row displays the top 10 movies which satisfy the user story C2 as it recommends movies based on the popularity. The second row shows the movies that have been rated by the user recently with the ratings given by the user. This satisfies the user story C4. The third row recommends movies that are similar to the user's favourite movies, which satisfies the user story C3. The fourth row recommends movies that are liked by the others who are similar to the user, which satisfies the user story C6. The last row recommends movies that are based on the actors or actresses that have also acted in user's favourite movies. As a result, both the third row and the fifth row suggest movies based on the user's favourite movies, which satisfies user story C1. Finally, the star ratings in the first, third, fourth and fifth rows denote the average rating of the movies, which satisfy the user story C5.

4.2.2 Basic Analysis Page

This section shows how the features in the basic analysis page satisfy the functional requirement defined in the Table 4.1.



Figure 4.2

Similar to Figure 4.1, Figure 4.2 indicates which parts of the basic analysis page satisfy which functional requirements (user stories) using colour boxes. The top section of the page displays the top ten movies at a specific period of time. There are 2 slide bars beneath the movie posters which allow the staffs to adjust the time period and the minimum number of ratings given to the movies. Hence the first section meets the functional requirement specified in the user story S1. The second section shows the word cloud diagrams for both movie genres and movie tags. From these diagrams, staffs are able to find which movie genres and tags are popular or unpopular. The slide bars on the left allow the staff the adjust the production year of the movies, minimum average ratings, minimum frequency of the appearance of the movie genres or tags and the maximum number of words to be show in the word cloud diagrams. Hence, the second section meets the functional requirement specified in the user story S2. The third section displays the histograms of ratings and distribution of average movie ratings. This satisfies the functional requirement specified in the user story S3. The bottom section shows the histogram of average

number of ratings given by user in each year, which satisfies the functional requirement specified in the user story S4.

4.2.3 Deep Analysis Page

This section shows how the features in the deep analysis page satisfy the functional requirement defined in the Table 4.1.

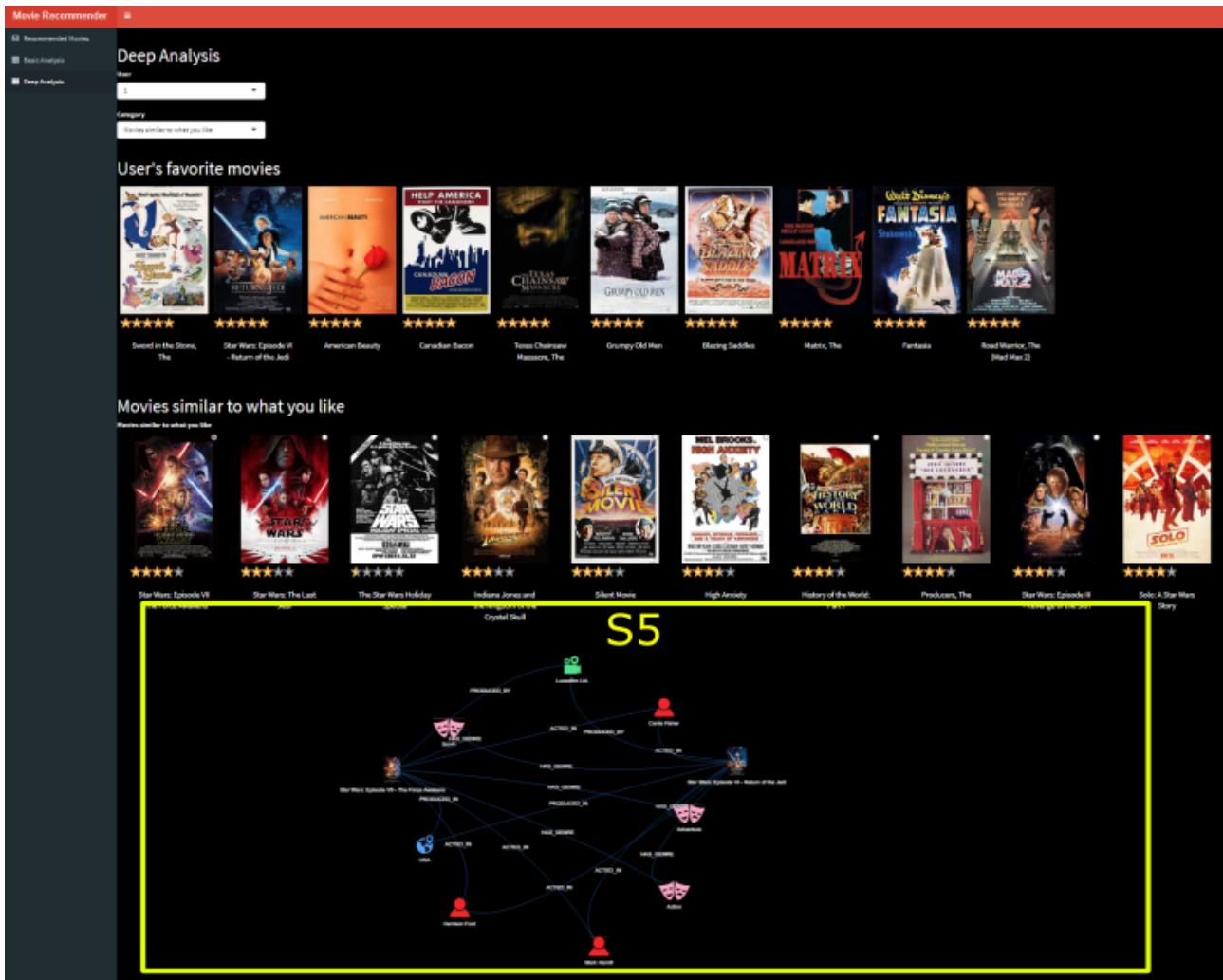


Figure 4.3

The deep analysis page allows staff to pinpoint a particular movie recommendation and examine the reasoning behind it. The first drop down box allows a staff to choose a particular user. The second drop down box allows a staff to choose a particular movie recommendation category for conducting analysis. The movie recommendation categories include the following:

1. Movies similar to what you like
2. Others similar to you like these movies
3. Movie with your favourite actors / actresses

Below the drop down boxes, the top ten movies that the chosen user like most are displayed. Once the staff has chosen the values from the two drop down boxes, the system will display the top ten movies that satisfy the chosen criteria. As shown in Figure 4.3, the category “Movies similar to what you like” is chosen and the top ten movies that are similar to user’s favourite movies are displayed. Finally, the bottom section displays the graph that explains the reasoning behind the recommendation. This satisfy the functional requirement specified in the user story S5.

5 References

neo4j (2020). Meta Graph. *APOC User Guide 4.1.* <<https://neo4j.com/labs/apoc/4.1/database-introspection/meta/>>