ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 408
Innovative System Design and Development II

MOVIE RECOMMENDATION SYSTEM

Doruk Gürsey
201511029
Kaan Önder
201511044
Kemal Berke Saka
201411051
Önen Emre Can
201511008

Advisor: *Abdül Kadir Görür*

Abstract

The aim of this project is to develop a system that can offer movie advice to people looking for movies they like or looking for similar ones they love. In this document, we will present to you what is our project will be look like, what are the algorithms that we may use and which softwares, systems uses these algorithms.

1. Introduction

Over the past 5 years, watching movies and series on digital platforms is now a part of our lives. While the film industry is slowly losing its influence, watching movies on the Internet and paying less for it is becoming a better and easier method for everyone. With the increase in our internet speeds, we are now able to watch movies or series from various platforms very comfortably. Some companies that foreseen this development have increased their investments in this field and have become the pioneers of this sector[1].

As we can see from the Figure 1, the internet movie watching sector has increased over the years and will continue to increase at the same speed.

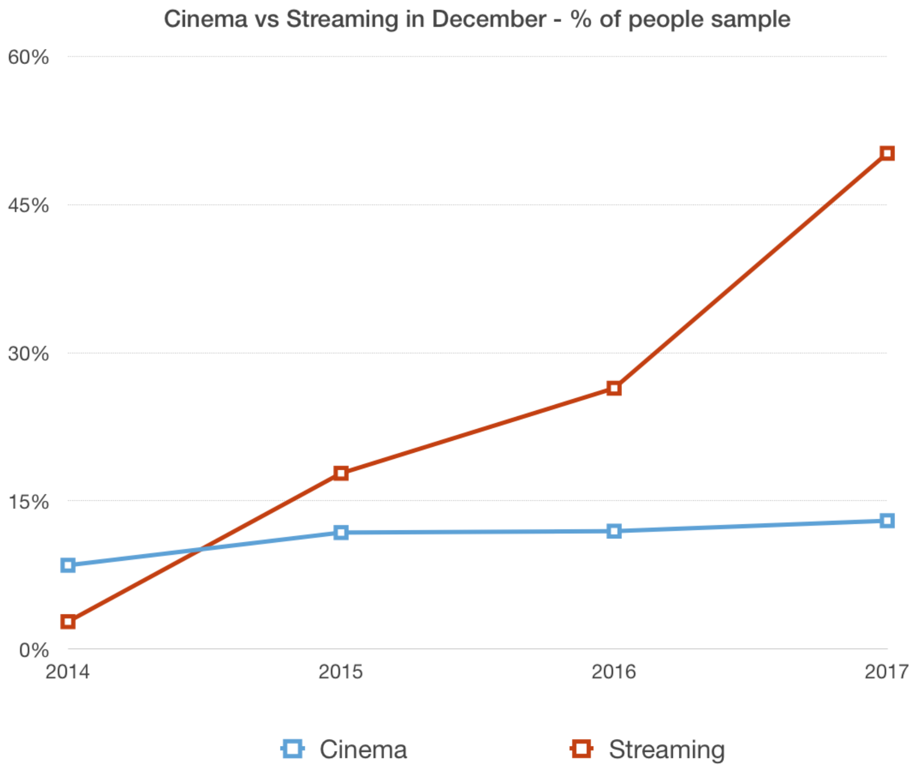


Figure 1

1.1 What is a Recommendation System?

State the problem to be solved. Why are you doing this work and what significance does it have in the relevant literature? Even if your project is applied (as opposed to research-oriented), you are building a system because a problem, requiring a solution in the form of a computer program, exists.

1.2 Recommendation System Algorithms

There are basically 4 important recommendation systems.

- Content-Based Filtering Systems
- Collaborative Filtering Systems
- Popularity Based Recommendation Systems
- Hybrid Recommendation Systems

1.3 Content Based Filtering Systems

This method is based on the users' preferred options and the content of those options. It attempts to recommend products that are similar to the products the user has liked in the past. So it works as follows; "If you like product A, you like product B" [3].

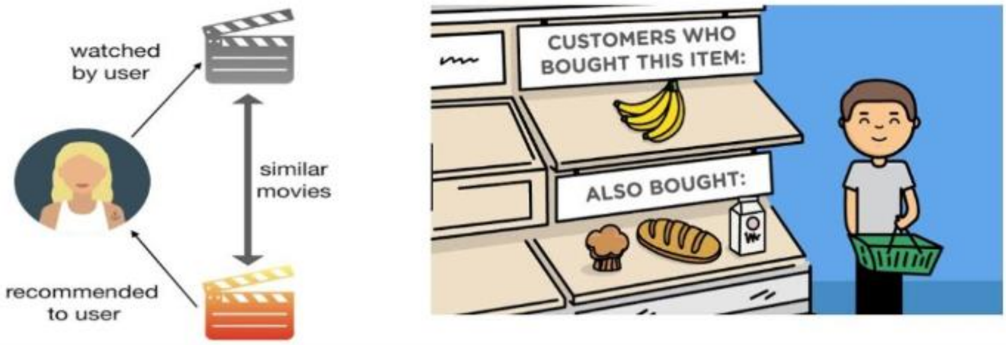


Figure 2

1.4 Collaborative Filtering Systems

This method is based on analyzing the user's activities, behaviors or preferences and gathering information about them. Its most important feature is that it combines the information it collects and analyzes, based on its similarity with other users, to predict what they will love. The underlying theory is; It is assumed that individuals who have similar preferences in the past will make the same choices in the future[4].

If they want to understand better, let me think you an example. User1 chose A,B,C and User2 choose B,C,D. It is obvious that User1 will prefer D and User2 will prefer A in the future. This is exactly what the system does.

There are 2 types of Collaborative Filtering Systems :

- User-Based Collaborative Filtering

This algorithm is based on associating similar customers. It works quite well but takes a long time to work. It is difficult to implement this algorithm for large platforms.

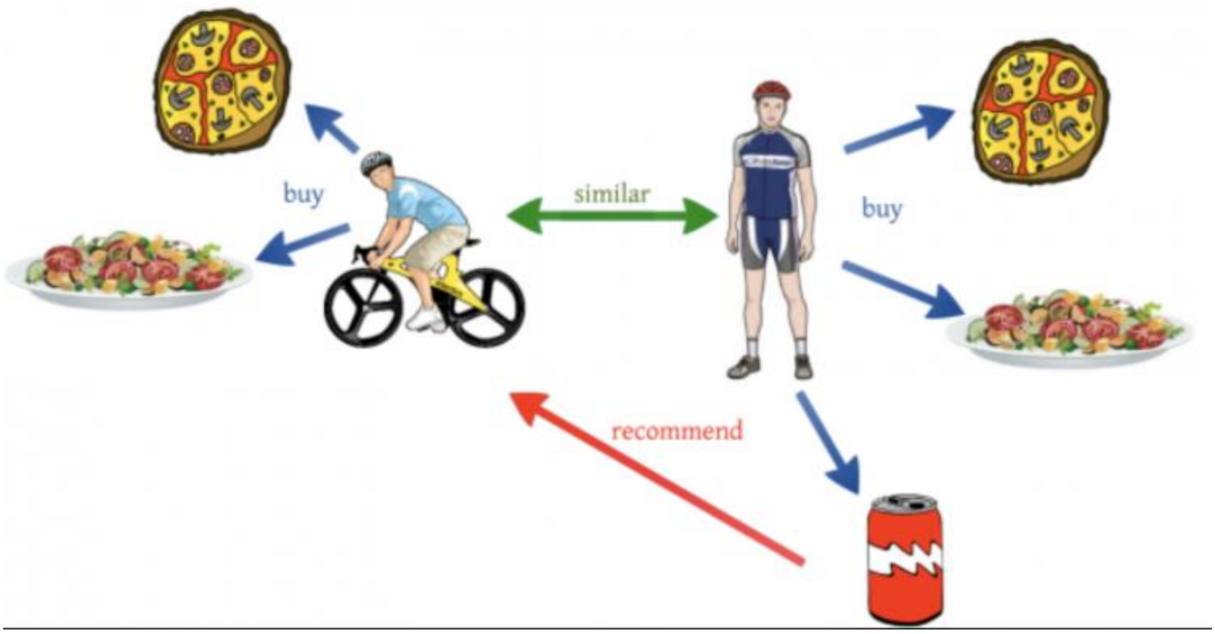


Figure 3

- Item Based Collaborative Filtering

So instead of finding the similarity among users, we focus on product similarity. With this algorithm, we can easily recommend similar items to the customer who purchases any item. It works faster than User-User Collaborative Filtering and easy to implement to the platforms.

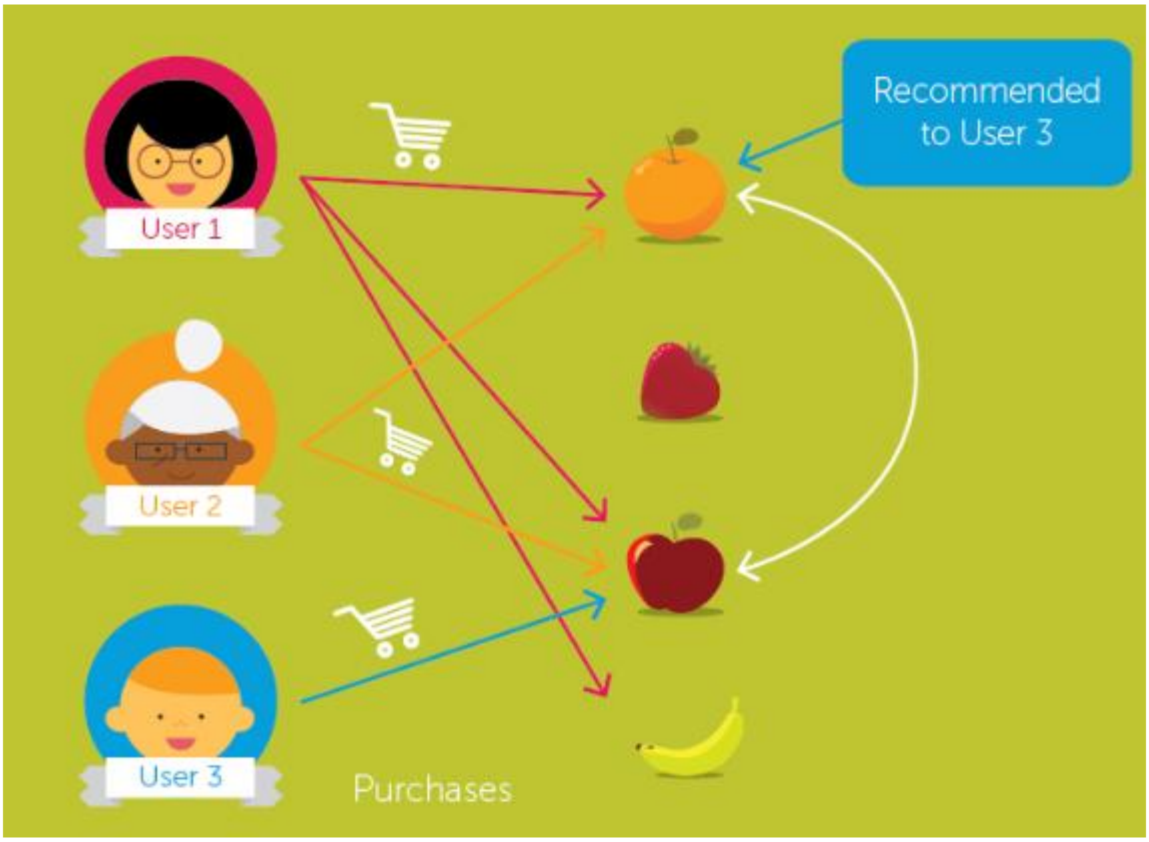


Figure 4

1.5 Popularity Based Recommendation Systems

Works based on popularity. Current products that are instant trending are used. For example, if 90% of newly registered members have purchased A, this product may be offered to the other newly registered user.

1.6 Hybrid Recommendation Systems

As a result of recent researches, it has been found that combining Collaborative Filtering Systems and Content-Based Filtering Systems will be more effective. It is not very common yet.

2. Who Uses Recommendation Systems?

2.1 Netflix

Netflix, according to the Statista report, reached approximately 150 million available in the first quarter of 2019. What Netflix does is it makes predictions based on users' viewing histories and the scores they have given to movies. Taking advantage of big data and artificial intelligence, Netflix; analyzing the monitoring activities of millions of users every day; yet you can really guess what you want to watch, even when you don't even know what you want to watch. Its most important innovation in recent years is; while offering a film to users with different picture covers[5]. If User1 likes romantic movies, they recommend an action film with a romantic picture cover. The cover pictures of the films are made by AI Models, not physically from the employees [6].

2.2 Amazon

Amazon is one of the companies that do Recommendation System best and earn the best money from it. 35% of this company's revenue comes from the Recommendation Systems. Because the existing algorithms are too hard to perform in a huge data like Amazons', They have developed their own algorithms. How Amazon's algorithm works are still unknown. This algorithm based on the Item Based Collaborative Filtering[7].

2.3 Spotify

Spotify is also at the top of the music industry. It gives each audience a personalized listening experience. It has more than 100 million users. Each week, Spotify presents a list of songs that users are likely to like, called _Weekly Discover_. It uses Collaborative Filtering in order to recommend a song to the people. Also, they use Natural Language Processing for analyzing the lyrics of the songs in order to good suggestions [8].

3. Software Requirements Specification

3.1 Introduction

3.1.1 Purpose

This document describes the Movie Recommendation System which is a web app that uses various prediction algorithms to propose their users' new movies according to their movie tastes. Also, the app will be compatible with mobile devices too. In this document, our main goal is to explain our project's requirements and functionalities. The purpose of this project is to help people to find new movies to watch and share their movie interests with other people and with their friends.

3.1.2 Scope of Project

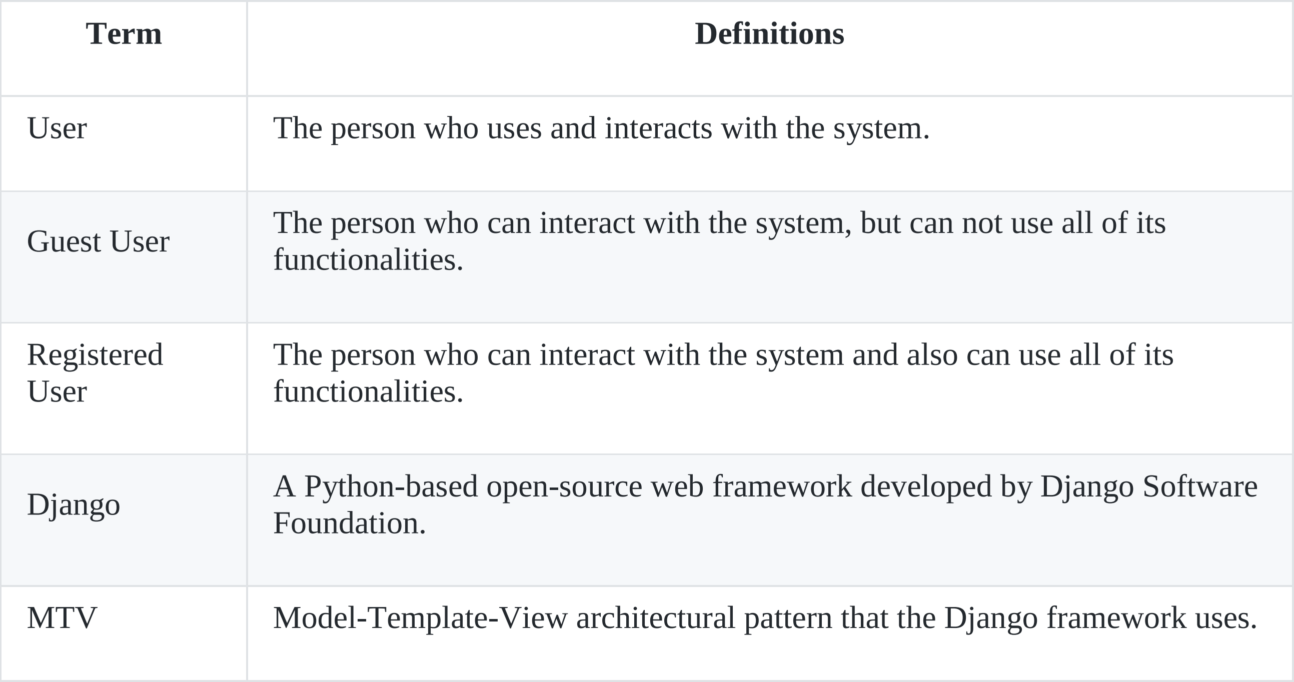
People used to have a hard time finding a movie that suits them. Because until 10 years ago, how could we set up a fast algorithm when our computers were not running fast? Even though people found such algorithms, we couldn't implement them because our calculations powers were slow. With the development of technology, such algorithms that could work fast began to be developed. Even though we don't realize this, they offer something every day.

The aim of this project is to propose movies using specific algorithms according to the movies people like. Recently, similar systems have been used by large companies such as Amazon, Netflix, and Spotify. Netflix uses a system similar to this, but only for movies on its own platform. IMBD also uses this system and they have large datasets of movies, users can find similar movies that they searched for. Our system will also include those approaches and benefit from those platforms to develop a unique movie recommendation system.

User will have 2 choices:

- Find movies similar to the movie that the user writes to the system.
- The system recommends movies based on user's movie preferences(whether he/she pressed the like button of a movie) and proposes similar movies to them.

3.1.3 Glossary



Term	Definitions
User	The person who uses and interacts with the system.
Guest User	The person who can interact with the system, but can not use all of its functionalities.
Registered User	The person who can interact with the system and also can use all of its functionalities.
Django	A Python-based open-source web framework developed by Django Software Foundation.
MTV	Model-Template-View architectural pattern that the Django framework uses.

3.2 Overall Description

3.2.1 Product Perspective

Movie Recommendation System is a website that recommends movies to people. Users registered can see the similar movies they watched and liked. Also, they can get new movie recommendations based on what they like from the system.They can easily find the movies they want and watch from other platforms.

3.2.2 Memory Constrains

Movie Recommendation System can be reached from any computer that connects to the internet.

3.2.3 Operations

Users can use the system even if they are not registered. If the user is not registered to the system, he/she can only search for a movie that he/she wants to and browse among all movies but that is it. If the user registered to the system, he/she can rate(like) movies they want to and after the user likes at least 8 movies. The system recommends movies that are similar to those of previous rated movies.

3.2.4 Site Adaptation Requirements

People can use our system even if they are not registered. If they are registered, they can be able to use more features of the system.

3.2.5 Product Functions

3.2.5.1 HomePage

- Users can use the system even if they don't register or log in to the system.
- If they write the name of the movie they watch and enjoy, the system will give them similar movies with the percentage of similarity.
- With the register button, they can register to the system from the register page.
- If the user already has an account he/she can log in with the login button from the login page.
- Users can logout from the system.
- Users can browseamong several movie genres.

3.2.5.2 Register Page

- Users that want to use the system's all functionalities enter their information here and it is recorded on the database.

3.2.5.3 Login Page

- Users can log to the system using their predefined information at the registration process on the register page.

3.2.5.4 Movie Page

- Users can display information about the selected movie.
- Registered users can rate(like) the selected movie in order to get more recommendations.

3.2.6 Constraints

- The movie Recommendation system is designed for users from all around the world. So our system will support only the English language.

3.2.7 User Characteristics

- Our system's language is English. So, users should understand English if they want to use the system.

3.2.8 Use Case Diagram

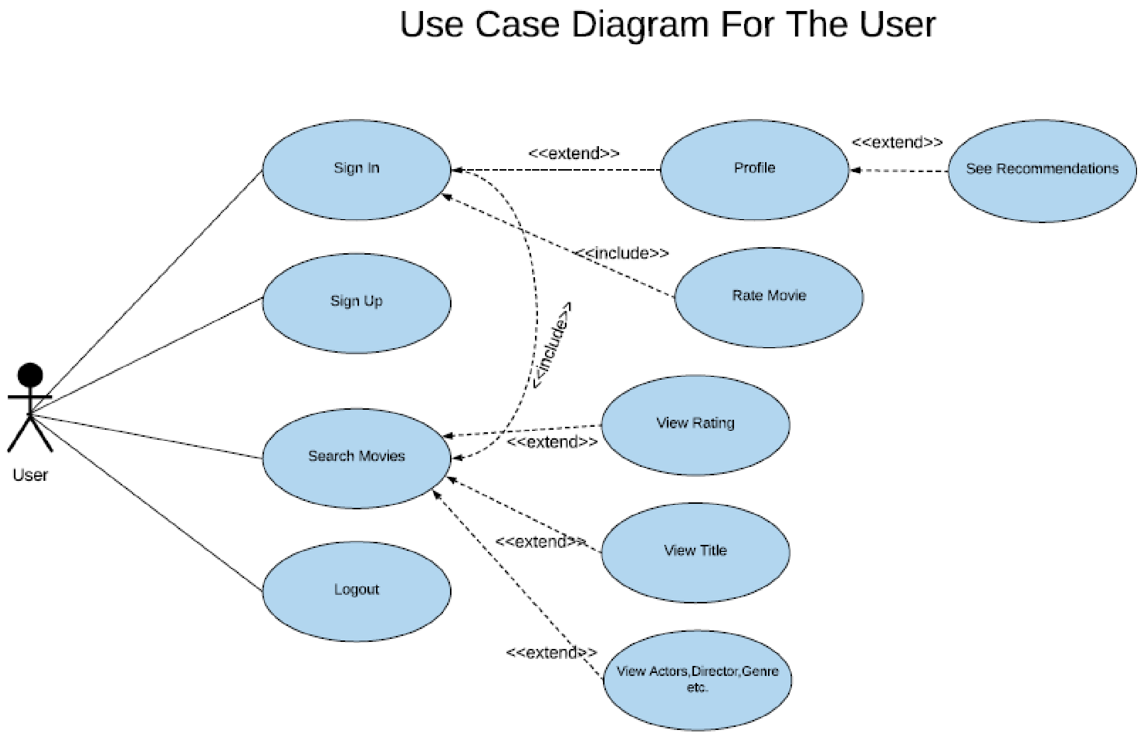


Figure 5 User Use Case Diagram

3.3 Requirements Specification

3.3.1 External Interface Requirements

3.3.1.1 User interfaces

Movie Recommendation System is a web-based project. We will use bootstrap for better User Interface.

3.3.1.2 Hardware interfaces

Users that want to use the Recommendation System must have :

- At least 1 GB of Ram
- Minimum 1 CPU
- Internet Connection

3.3.1.3 Software interfaces

Any device that can access the internet can use our system. We recommend using Google Chrome while using the system.

3.3.1.4 Communications interfaces

User needs to have an internet connection and a web browser.

3.3.2 Performance Requirements

Internet connection speed is not so important to use our system. Having a connection is the most important thing.

3.3.3 Software system attributes

3.3.3.1 Portability

In Movie Recommendation System, we will use Django, Vue.js, and Bootstrap. Users can access our web-based application from their devices both mobile or stationary.

3.3.3.2 Performance

Performance is based on the server density and internet connection.

3.3.3.3 Adaptability

Users can use our system from the time they log in to our website. Registered users will have more features available.

3.3.3.4 Safety Reqirements

There is no big deal for safety because users will not share all their pieces of information to the system. They just want to get recommended movies.

4. Conclusion For SRS

This software document contains more details about the project. The software we will use, the movements of the users who will use the system, which team member is responsible for which task are stated in this document. Thanks to this document, it will be easier for us to do the project, and when we forget something or want to remember, we will be able to go back and continue the project by looking at this documentation.

5. Software Design Description

5.1 Introduction

5.1.1 Purpose

The aim of this document is to test the behavior of a Movie Recommendation System and more details. At first, our target users are the ones who find movies that are closer to their movie tastes. Our biggest goal is to help people after successfully setting up our algorithm. We think we will change the way people view movies. There will be a dataset for our system to use effectively. Users can enter their information into the system and keep them in a database. With registration, users will be able to rate(like) movies they want and get recommendations based on their ratings.

On the other hand, our system is also free to use without registering. If you do not register, you can only browse among movies that the system has.

5.1.2 Scope

We are going to use Python and its frameworks for this project because Python is a programming language that works very well in machine learning algorithms and has a very good interface within websites. Python is one of the easiest languages to learn and process. Clear definition of syntax; its simple interface makes it easier to read.

Another advantage of Python is its ability to access standard libraries. UNIX can be moved to platforms such as Windows and Macintosh without requiring interface changes. Python's source code is easy to maintain and allows the use of interactive modes. Another reason why Python is so popular is its wide range of operations, including data mining and deep learning.

5.1.3 Glossary

Term	Definitions
Django	A Python-based open-source web framework developed by Django Software Foundation.
MVT	Model-View-Template architectural pattern that the Django framework uses.
SQLite	A lightweight embedded database management system.
SCRUM	An agile software development methodology based on iterative and incremental approach.

5.1.4 Overview of document

More detailed information of the rest of the content is clarified in the below sections. Section 2.1. is the "Architecture Design Approach" which contains information about the development methodology of the project. Section 2.2 contains information about which technologies will our project have and how will we use them. Section 2.3 is the Class diagram that represents the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. Section 2.4 contains information about the different roles of the system's architectural components. Such as "Admin Operations", "User Operations", "Prediction System ". Section 2.5 contains the activity diagram of the project that represents the workflows of activities and actions.

5.2 Architecture design

5.2.1 Architecture Design Approach

For developing the project, we have planned to use the scrum method. Scrum is a term used by people who are considered to be the development method of an application and who are working on software development. With Scrum, it will be beneficial to ensure that the observer, who possesses the characteristics aimed at developing the main objective, has a developmental structure, and to obtain a repetitive infrastructure in this process and to pay attention to the development of a modern system.

It will be possible to try to reduce this complex order as one of the most important factors will be the thought or assumption that it will be difficult to ensure that all stages are planned in a planned manner. At this point, it would be useful to evaluate the following three main items. The roles in the scrum process basically start as interaction, the product owner. The Scrum administrator is determined and the work is started considering the user or the masses of users. A system focuses on systems of interaction with managers and teams, taking into account the risk of abuse of roles.

5.2.2 Technologies Used

We have planned to use Django Framework. The main purpose of the Django Framework is to facilitate the use of complex web applications. Django is designed with a policy of reusability, modularity and rapid development process. Django also provides a dynamic management panel that incorporates the basic record insertion, deletion, editing and update functions for each project.

A convenient web framework for those who want to get rid of complex databases. Django is an open-source web framework that uses an MVT architect structure written in Python programming language. The structure we call MVT is Model, View, and Template. It shows us that Django was coded in the development stage by paying attention to this structure. Also, we are going to use SQLite for database design because it is reliable and reading and writing operations are fast. The database has a very important place here because the application runs completely through the database. Thanks to the database we will be able to collect information about users and give them the information they want. For the background, we are thinking of using HTML, CSS, and javascript.

5.2.3 Architecture Design Of Movie Recommendation System

Position	Registered User
Login Page	The user can log in to the system
Home Page	The user can list and browse the movies in the system
Home Page	The users can rate(like) movies.
Movie Page	The users can rate(like) movies.
Home Page	The users can list the movies that are similar to the previous rated(liked) movies with the details.
Home Page	The users can select and list the movies that they rated(liked)

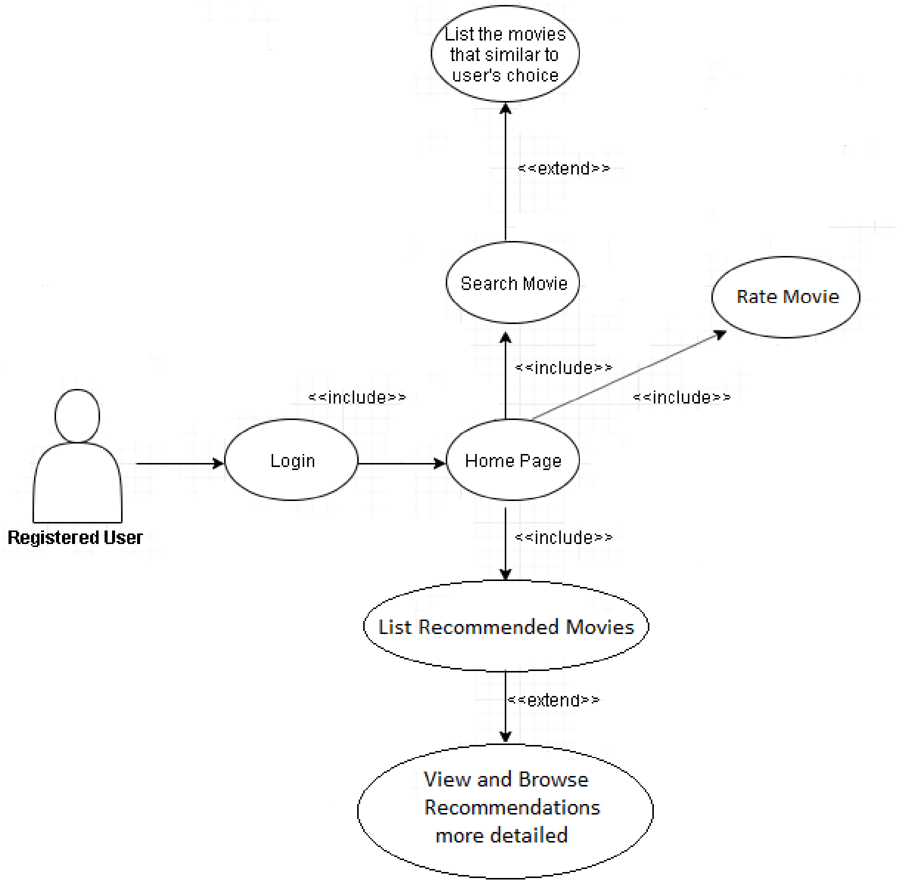


Figure 6 Registered User

Position	Unregistered User
Login Page	The user can log in to the system
Register Page	The user can register to the system
Home Page	The user can list and browse the movies in the system

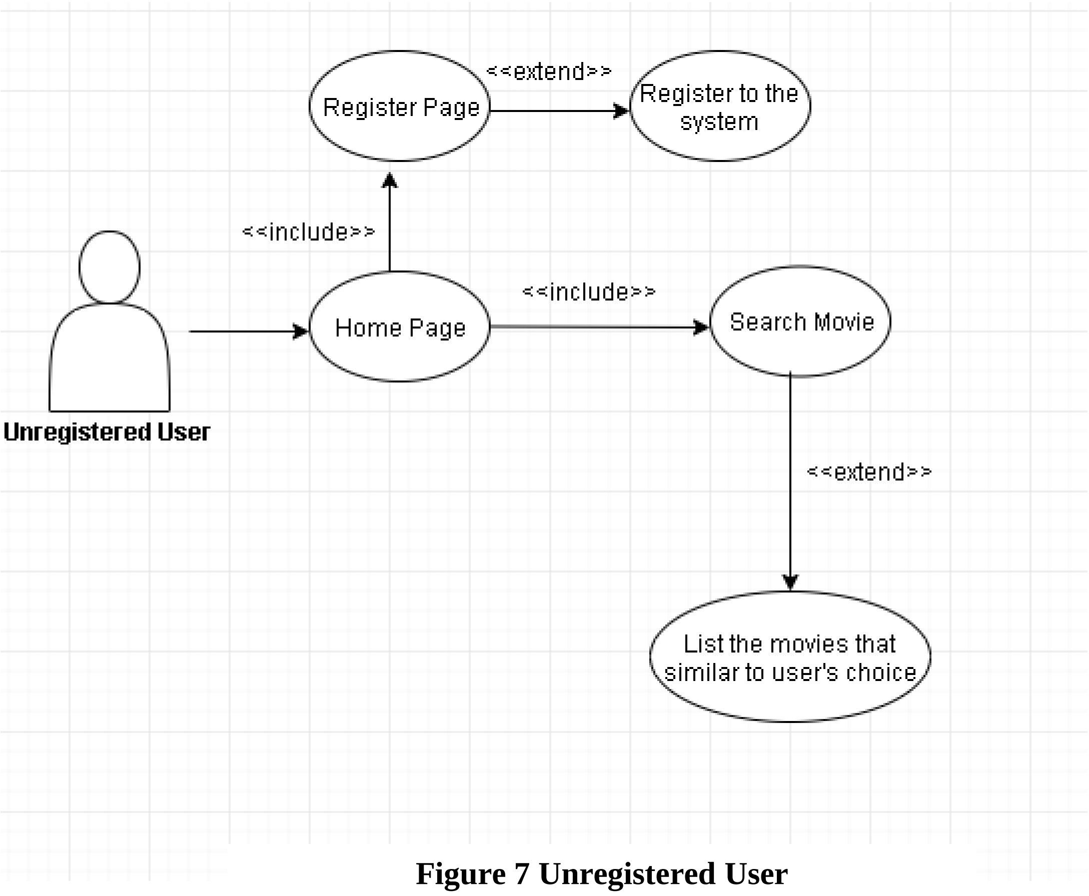


Figure 7 Unregistered User

Position	Admin
Panel	The admin can update the algorithm and dataset for better recommendations
Home Page	The admin can delete users

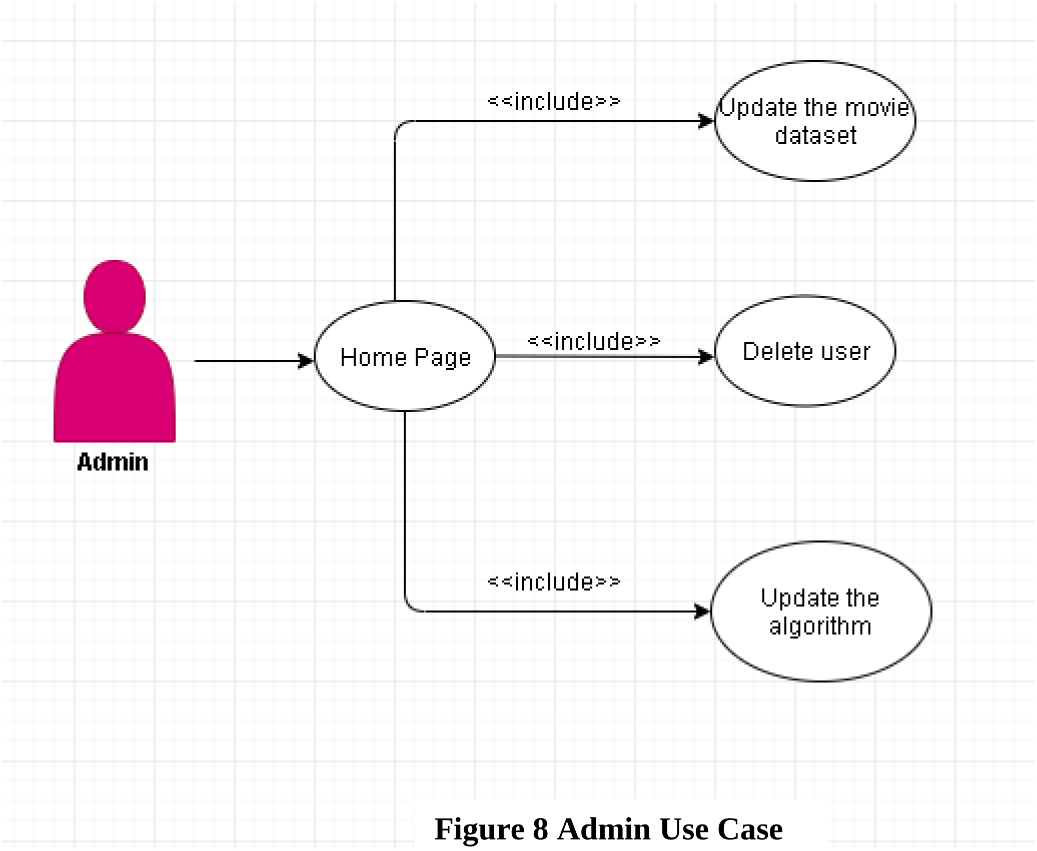


Figure 8 Admin Use Case

5.3 Class Diagram

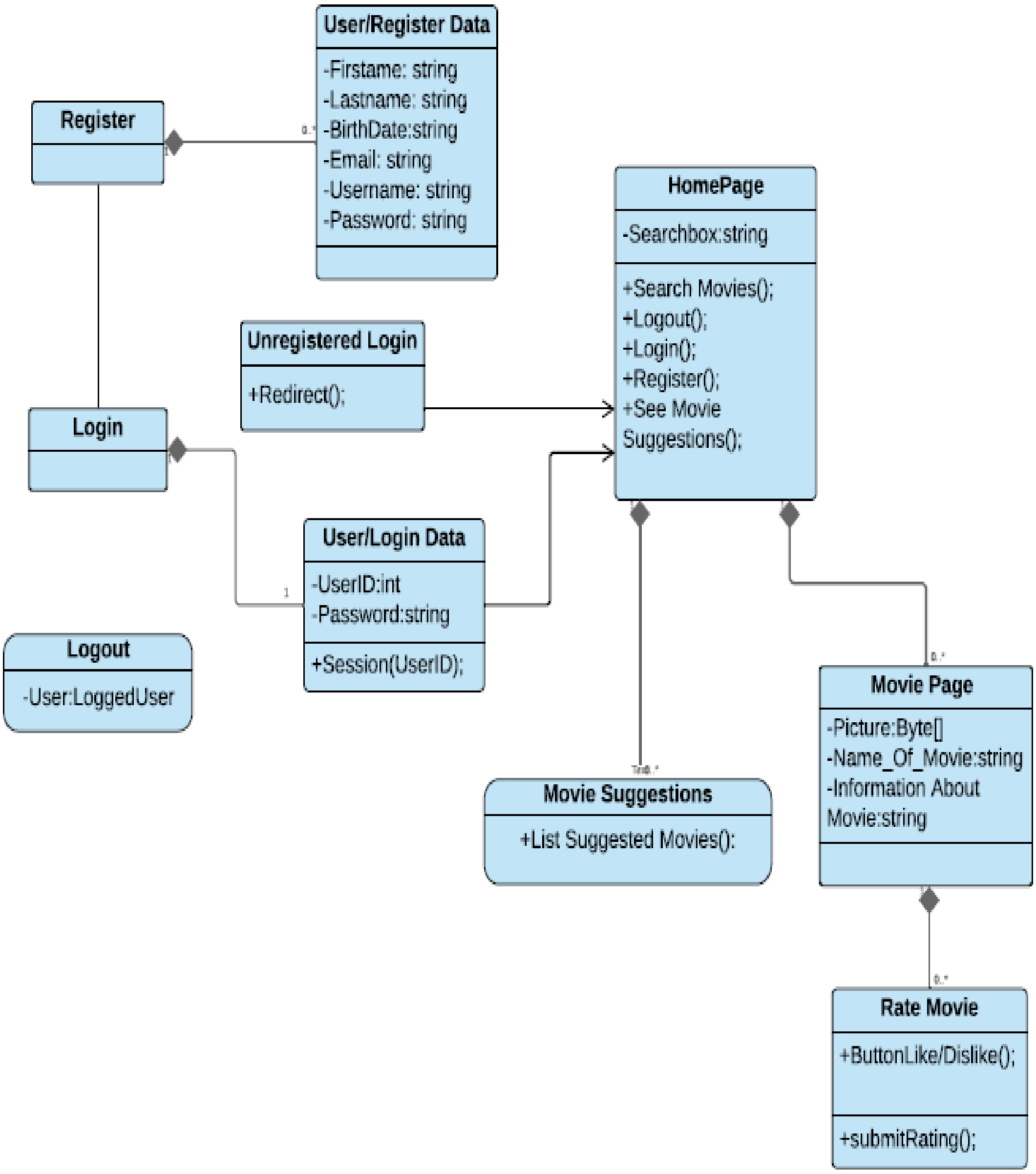


Figure 9 Class Diagram

5.4 Use case realizations

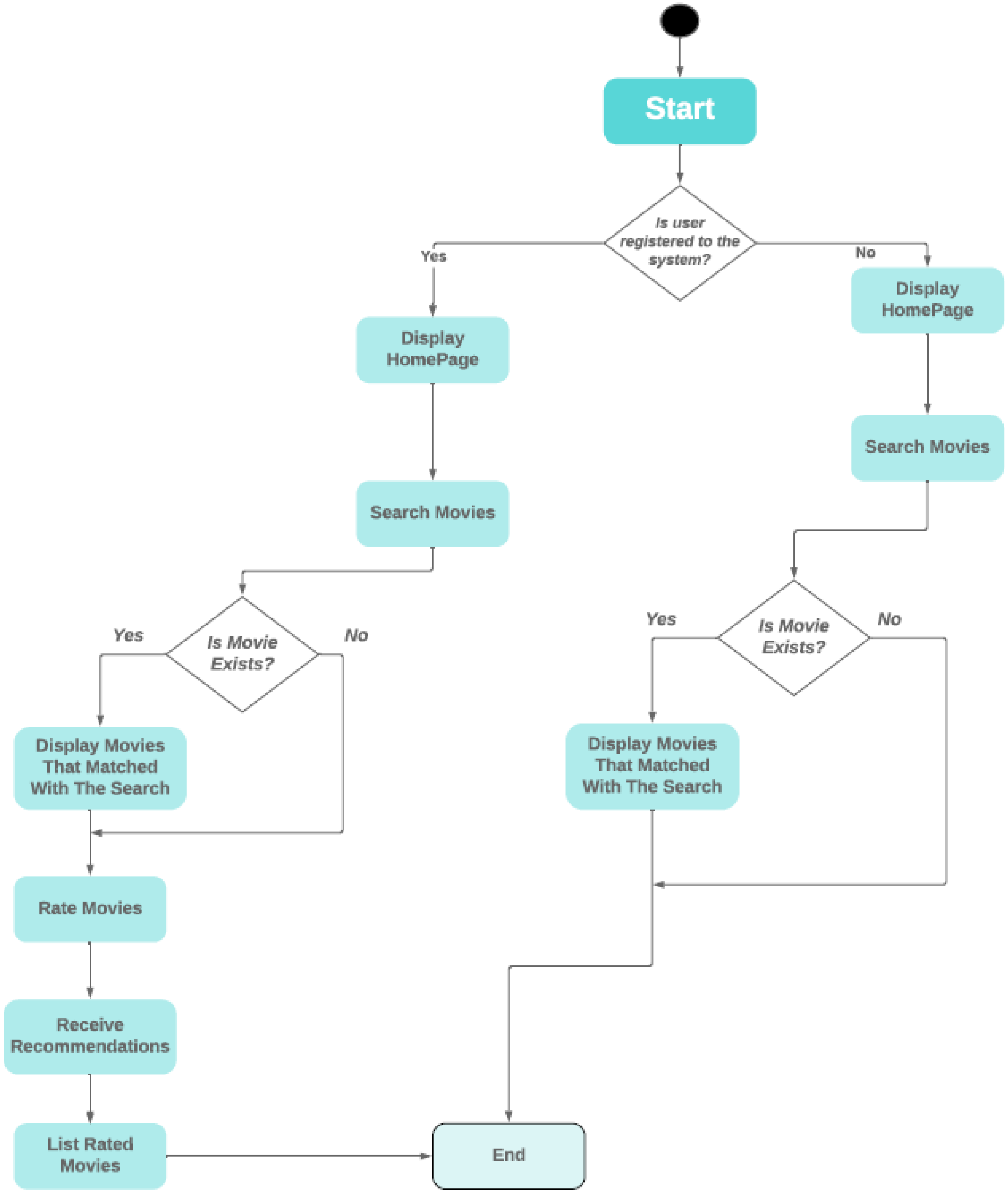


Figure 10 System Activity Diagram

5.4.1 Movie Recommendation System Components

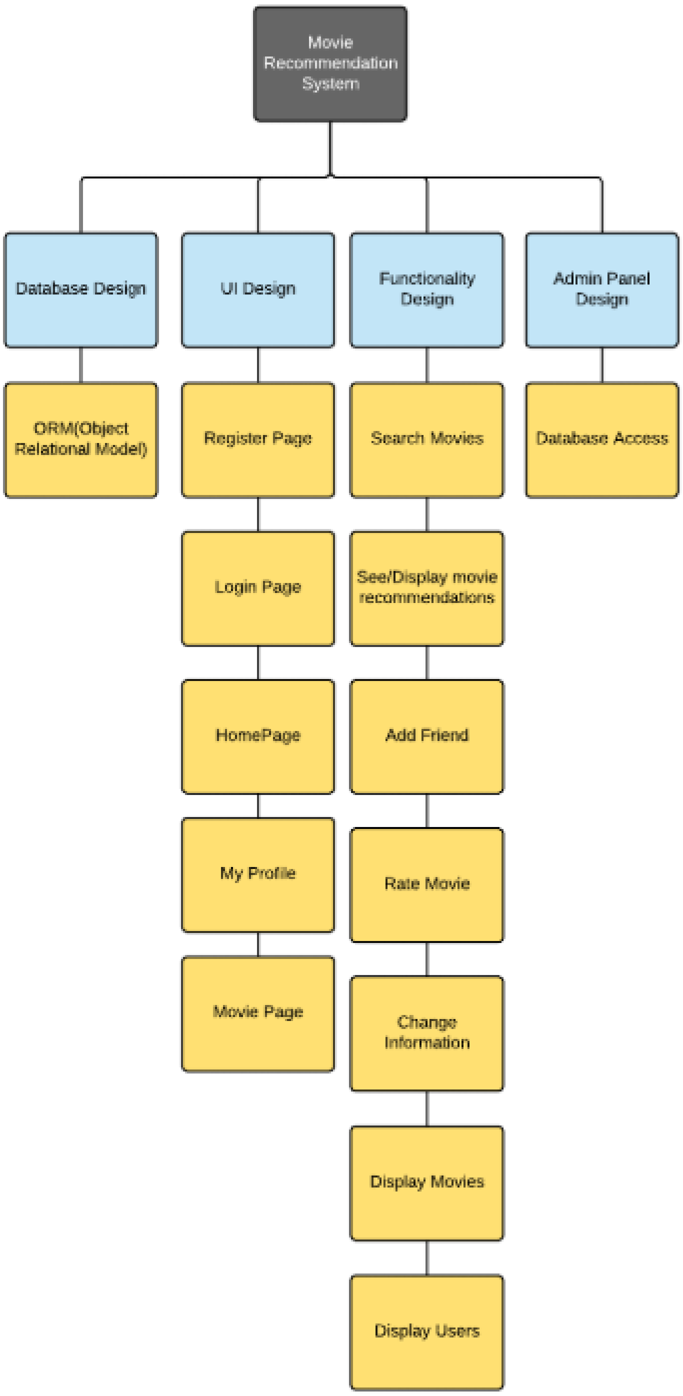


Figure 11 System Components Diagram

5.4.2 UI Design Description

Our project's UI (User Interface) Design will contain 5 different user interactable sub-parts. They are all explained below.

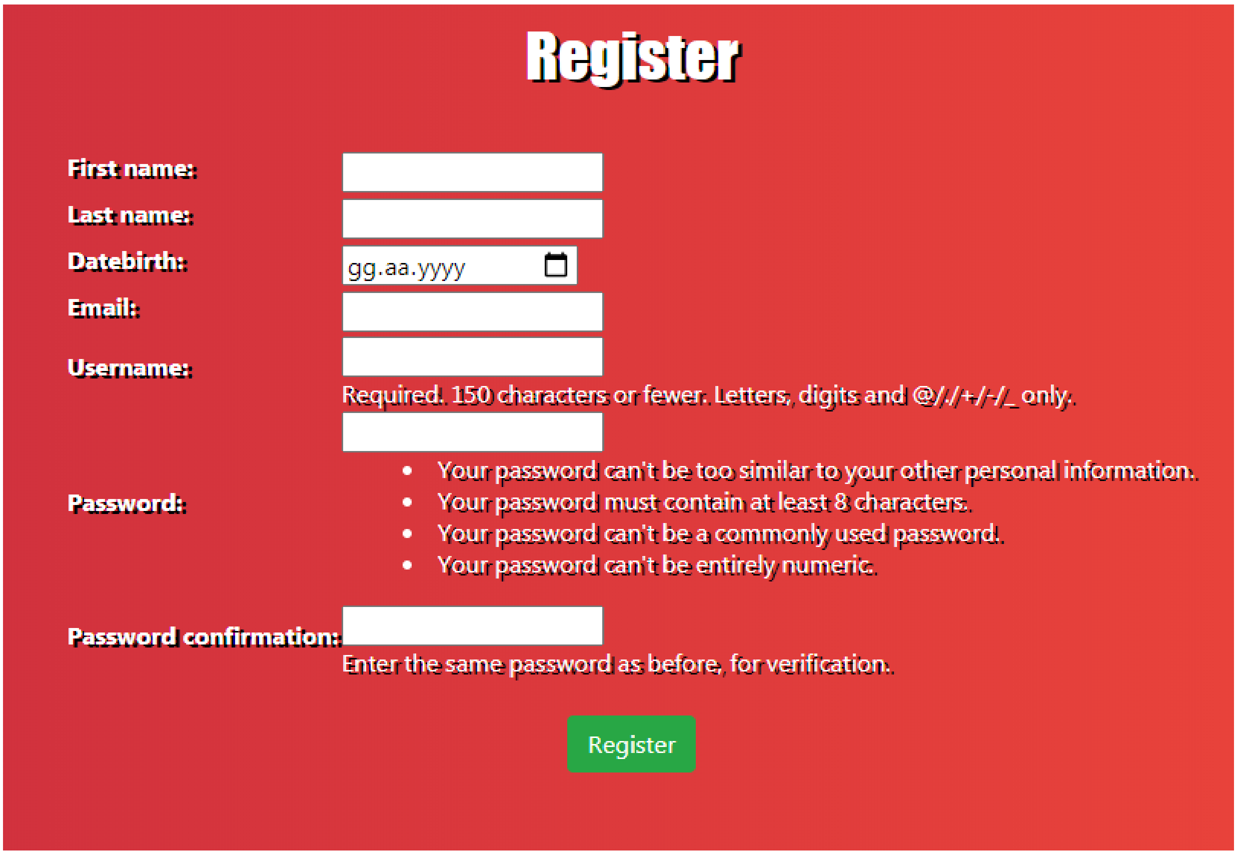
6. **Login Page:** Registered users logs in to the system by validating their own username and password.(Figure 2)

7. **Register Page:** Users register to the system by entering their information. After this process, the system saves the user to the database as a registered user. (Figure 1)

8. **Home Page:** The system redirects registered or unregistered users to this page. Most of the system's functionalities can be accessed from this page. (Figure 3)

9. **Movie Page:** Registered or unregistered users can access this type of page by selecting any movie they want and can examine the information about the movie. Registered users can rate movies and receive suggestions from the system based on their decisions.(Figure 4)

5.5 Interface Design



Register

First name:

Last name:

Datebirth: gg.aa.yyyy

Email:

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

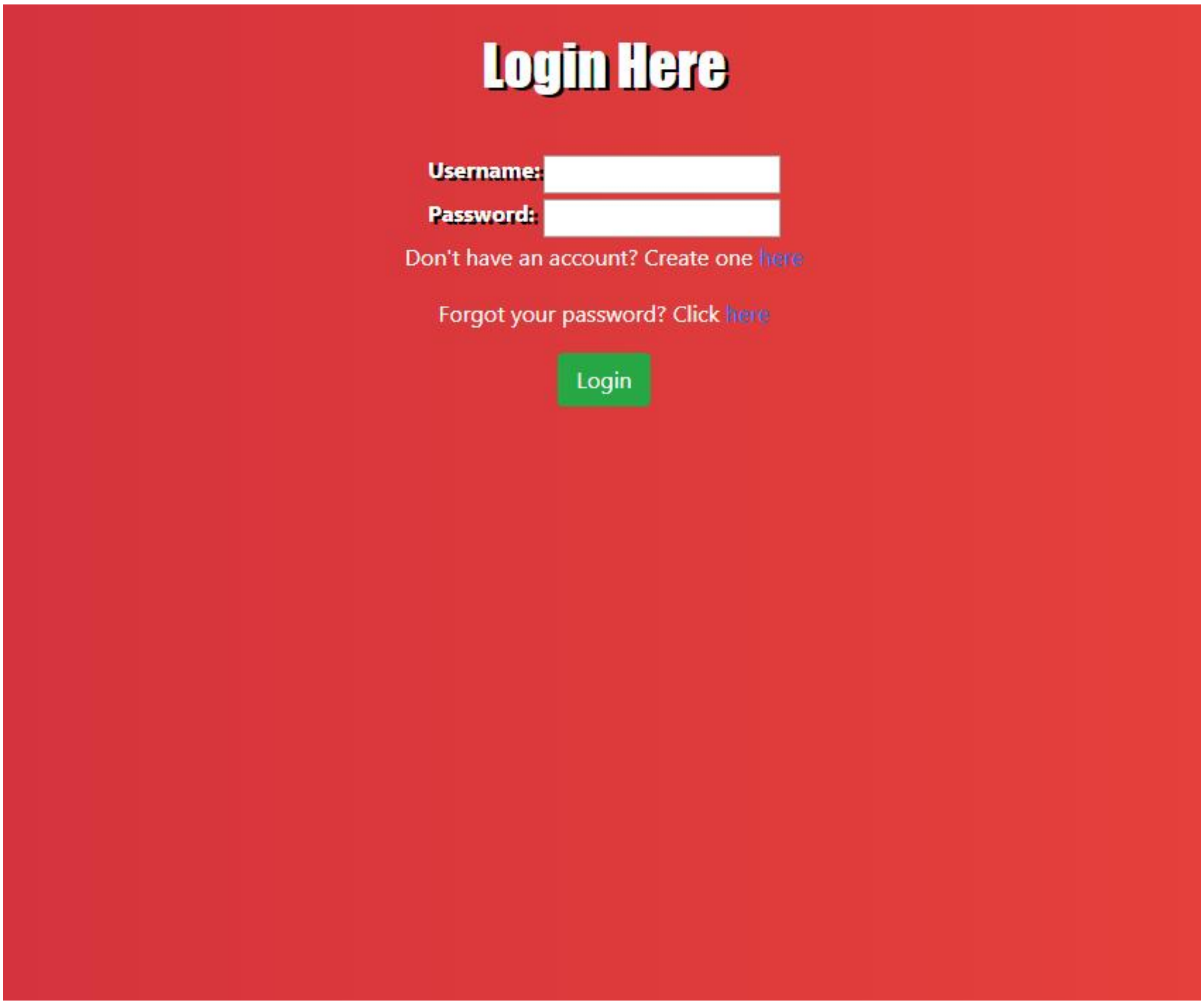
Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

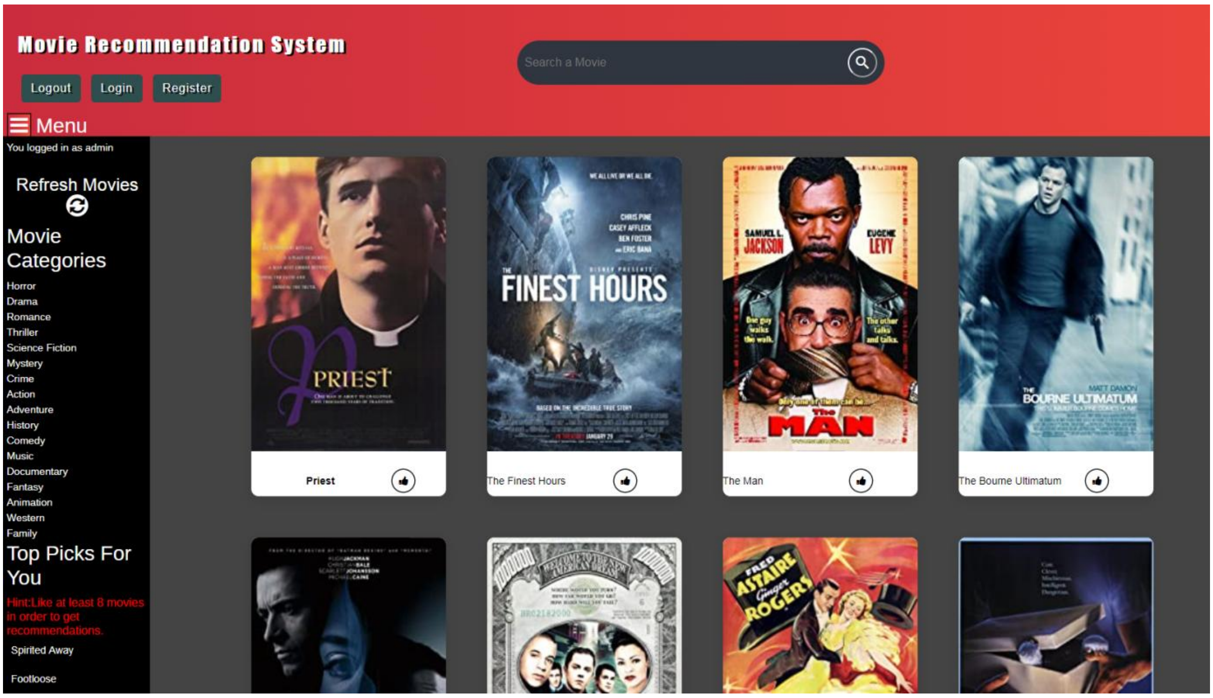
Figure 12 Register Page Design



Don't have an account? Create one here

Forgot your password? Click here

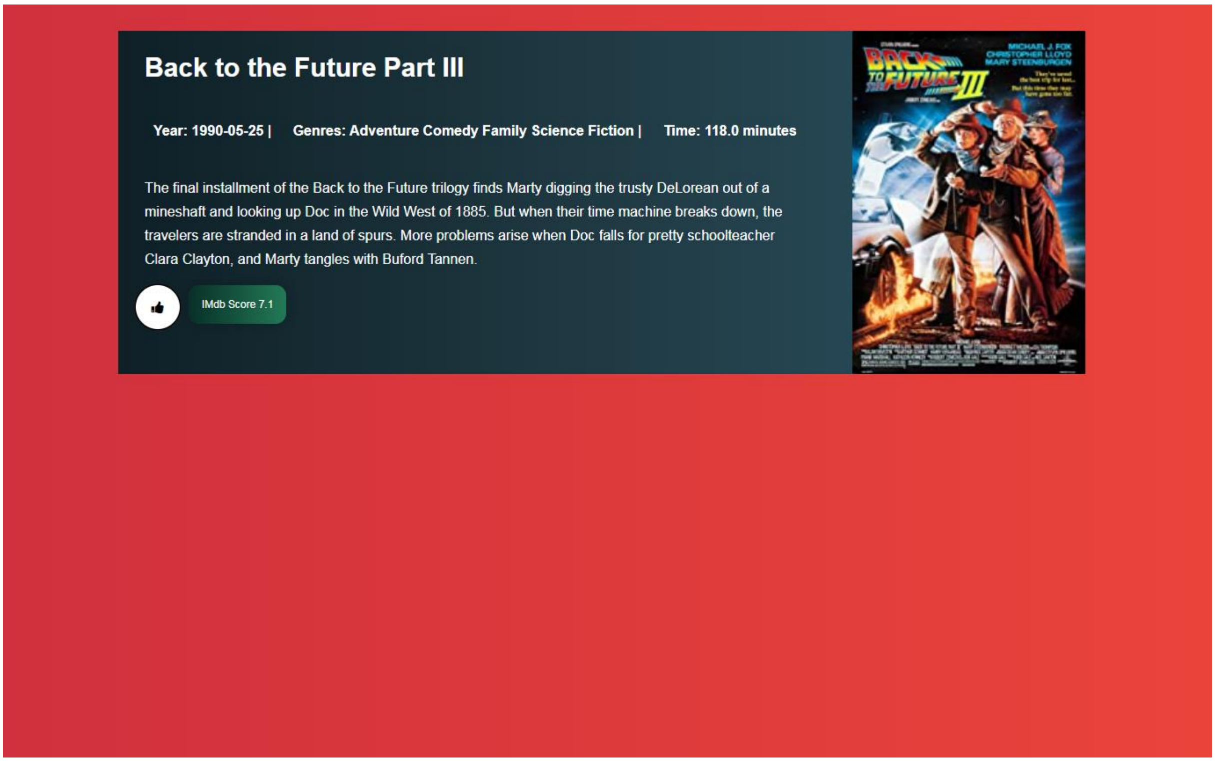
Figure 13 Login Page Design



You logged in as admin

Hint:Like at least 8 movies in order to get recommendations.

Figure 14 Home Page Design



Back to the Future Part III

Year: 1990-05-25 | Genres: Adventure Comedy Family Science Fiction | Time: 118.0 minutes

The final installment of the Back to the Future trilogy finds Marty digging the trusty DeLorean out of a mineshaft and looking up Doc in the Wild West of 1885. But when their time machine breaks down, the travelers are stranded in a land of spurs. More problems arise when Doc falls for pretty schoolteacher Clara Clayton, and Marty tangles with Buford Tannen.

Figure 15 Movie Page Design

5.6 Conclusion

Even though we could not complete everything we wanted to do at the end of our project, we have completed the vast majority. We had to remove some features unfortunately. Our system runs well most of the time and we successfuly implemented the cosine-based similarity method in our project.Users can easily register to the system and receive mostly accurate recommendations based on the movies they rated(liked) and easily browse,search and get more specific information among thousands of movies.(Movie overview,genre,runtime,score of the movie etc.)This project improved our both object oriented programming skills and how machine learning,prediction algorithm methods work.

6. Test Plan

6.1. Introduction

In this document, we will define our actions and use cases with test plans.

6.2. Overview

System features and use cased of the Movie Recommendation System are planned to be tested. Requirements, functions and more details had been determined in SRS and SDD documents before.

6.3. Scope

This document includes existing interfaces, integration of all functions.

6.4. Terminology

Term	Definition
GUI	Graphical User Interface
UI	User Interface
ML	Machine Learning
DB	DataBase

6.5. Features To be Tested

This section gives a brief description and lists of all the major features to be tested. Movie Recommendation System has 2 main parts which are Registered, User and Non-Registered User. For each major feature, there will be a Test Design Specifications added as a section.

6.6. Graphical User Interface

Most of the features of our project related to GUI. GUI contains a login page, register page, movie pages, and movie detail pages. There are 2 main parts of GUI which are Registered and Non-Registered parts. If the user registered to the system, they can access all of the pages with Movie Like Button, User Similarity and Add Friend. They should like the movies in order to get recommendations related to their likes. They should also add friends and like a movie in order to see the movie taste similarity between their friends and also they should add friends to see this. If users are non-registered, they can't access Like Buttons and User

Similarity. They have to write the movie name one by one and choose one of them and see the details of that movie. Of course, they can't see the movie similarity between users because they are not registered.

6.7. Item Pass/Fail Criteria

This section describes the general rule to use to decide then a test case passes and when it fails.

6.8. Exit Criteria

The product will be considered successful when we accomplish the functionalities that we pre-determined. Those functionalities can be found in our SRS and SDD documents.

- %100 of the test cases are executed
- %99.9 of the test cases are passed
- All High (H) and Medium (M) priorities of test cases passed

6.9. Test Design Specifications

6.10. Subfeatures to be tested

Register Button (GUI.REG.BTN)

If a user wants to be registered to the system and use its exclusive contents, he/she could do it by entering the specified information that the system requires. Such as First name, last name, email, password.If the inputted values are valid. (System checks whether they are valid or not) The user registers to the system successfully. This button also redirects the user to the homepage after a successful registration.

Login Button (GUI.LOG.BTN)

If the user has an account for the system already(Registered already), he/she can access the system's exclusive contents by inputting the required information. These pieces of information are the username and password of the user. If the inputted values match the values in the database, the user logs in.

Don't have an account Link (GUI.ACC.LINK)

This link only redirects the user from the login page to the register page.

Forgot Password Link (GUI.FORGOT.LINK)

If the user has an account but can not remember his/her password, he/she can reset his/her own password by clicking this link that is on the login page. With this link, the user enters its email and then the system sends a link to that inputted email address. When the user clicks that link that sent by the system. He/she encounters with a password reset page where he/she can reset the account's password. This is a multiple-step process, it has several sub-functionalities. Thus, more buttons such as;

- Email Submit Button(In order to send the link to the inputted mail)
- Change Password Button

Like Button (GUI.LIKE.BTN)

Users can use Like Button only if the user has an account. With this button, the user can like the movies and the movie name is written to the database. Then, in order to run the recommendation algorithms, parse the data from that table and perform the algorithm.

Get Recommendation (GUI.RECOMMENDATION.LINK)

When the user likes movies, there will be a link which is Get Recommendation button. With this button, users can get movie recommendations related to their likes.

Add Friend (GUI.FRIEND.LINK)

Users can add a friend. This feature provides us to perform the machine learning algorithm which is about user similarity. With this feature, we can provide users to see their similarities between their friends.

6.11. Test Cases

TC ID	Requirements	Priority	Scenario

TC ID	Requirements	Priority	Scenario
GUI.REG.BTN	3.1	H	Select 'Register' button and there will open register page

TC ID	Requirements	Priority	Scenario
GUI.LOG.BTN	3.1	H	Select 'Login' button and there will be opened login page

TC ID	Requirements	Priority	Scenario
GUI.ACC.LINK	3.1	L	Select 'Don't have an account' link and there will be opened register page

TC ID	Requirements	Priority	Scenario
GUI.FORGOT.LINK	Graphical User Interface	H	Select 'Forgot Password' button and we will send a link to email in order to change passwords if users forgot

TC ID	Requirements	Priority	Scenario
GUI.LIKE.BTN	3.1	H	Select 'Like' button in an order running the algorithm of the recommendation which is in the 'Get Recommendation' Page

TC ID	Requirements	Priority	Scenario
GUI.RECOMMENDATION.LINK	3.1	H	Select 'Get Recommendation' link

TC ID	Requirements	Priority	Scenario
			and there will be opened a page which people can see the suggested movies according to the movies they like

6.1. Database(DB)

LoginQuery (DBS.LGNQRY)

The user will write the username and password. If that is correct, they will see the main page.

RegisterQuery (DBS.REGQRY)

The user will write the username, password and more details about him. The query will go to the database and keep it there.

LikeQuery (DBS.LIKEQRY)

Users will like the movies. In each like, the movie name will be written to the database in order to get movie recommendations.

AddFriendQuery (DBS.FRIENDQRY)

User will add a friend and that will be written to the database in order to get user similarity.

Test Cases

TC ID	Requirements	Priority	Scenario
DBS.LGNQRY	3.1	H	This query is for logins

TC ID	Requirements	Priority	Scenario
DBS.DBS.REGQRY	3.1	H	This query performs user to register
DBS.LIKEQRY	3.1	H	This query is for movie likes and in order to get movie recommendations
DBS.FRIENDQRY	3.1	H	This query is kept the user's friends and then give them a similarity between their friends

6.2. Machine Learning

Cosine Similarity (ML.COS_SIM)

This is all about the backend of the program.

Test Cases

TC ID	Requirements	Priority	Scenario
ML.COS_SIM.1	3.1	H	If the user writes the name of the movie to the textbox, they will get the highest 10 movies that similar they write.
ML.COS_SIM.2	3.1	H	If users add friends, they will get similarities between him/her. This is also performed by the cosine similarity
ML.COS_SIM.3	3.1	H	If the user registered, they will like movies. Then if they select 'get recommendations', they will get the highest 8 movies that they liked before

6.3. Detailed Test Cases

GUI.REG.BTN

TC ID	GUI.REG.BTN
Purpose	Register the user to the system
Requirements	3.1
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Server must be running
Setup	-
Procedure	Click the register button from the homepage to redirect to the registration page, then click "Register" button after inputted your information validly
Cleanup	Successful registration / Exit

GUI.LOG.BTN

TC ID	GUI.LOG.BTN
Purpose	Log the user to the system
Requirements	3.1
Priority	High

TC ID	GUI.LOG.BTN
Estimated Time Needed	Less than 1 minute
Dependency	Server must be running
Setup	-
Procedure	Click the login button from the homepage to redirect to the login page, then click the "Login" button after inputted your information validly
Cleanup	Successful user validation from database / Exit

GUI.ACC.LINK

TC ID	GUI.ACC.LINK
Purpose	Redirect the user to the registration page
Requirements	3.1
Priority	Low
Estimated Time Needed	Less than 1 minute
Dependency	Server must be running
Setup	-

TC ID	GUI.ACC.LINK
Procedure	Click the "Don't Have An Account" link from the Login page
Cleanup	Redirection to the page

GUI.FORGOT.LINK

TC ID	GUI.FORGOT.LINK
Purpose	Reset the password of the user if the user wants to change the password of his/her account
Requirements	3.1
Priority	High
Estimated Time Needed	1 - 5 minutes
Dependency	Server must be running / Email of the user must be valid
Setup	-
Procedure	Click the "forgot your password" link from the Login page
Cleanup	Successful password reset / Exit

GUI.LIKE.BTN

TC ID	GUI.LIKE.BTN
Purpose	If user wants to get recommendations, he/she has to like the movies
Requirements	3.1
Priority	High
Estimated Time Needed	Less than 2 seconds
Dependency	Server must be running
Setup	-
Procedure	Click 'Like' sign
Cleanup	Like / If you liked before; dislike

GUI.RECOMMENDATION.LINK

TC ID	**GUI.RECOMMENDATION.LINK**
Purpose	When user likes movies, if they want to get recommendations, they should use this feature from the webpage
Requirements	3.1
Priority	High
Estimated Time Needed	Less than 10 seconds
Dependency	User must like at least 8 movies
Setup	-
Procedure	Click 'Get Recommendations' link
Cleanup	User goes to another page which they get recommendations

7. References

- [1] https://www.sheffieldav.com/education/how-internet-changing-filmmaking-industry
- [2] https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
- [3] https://medium.com/@cfpinela/content-based-recommender-systems-a68c2aee2235
- [4] https://realpython.com/build-recommendation-engine-collaborative-filtering/
- [5] https://help.netflix.com/tr/node/100639
- [6] https://www.wired.co.uk/article/netflix-data-personalisation-watching
- [7] https://aws.amazon.com/tr/blogs/big-data/building-and-running-a-recommendation-engine-at-any-scale/
- [8] https://medium.com/@ericboam/i-decoded-the-spotify-recommendation-algorithm-heres-what-i-found-4b0f3654035b
- https://link.springer.com/article/10.1007/s11257-019-09231-w
- http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/
- https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26