Chapter 1

Introduction

There is one famous quote about customer's relationship "customers don't know what they want until we show them". If we succeed in showing something which customers may like business profit will skyrocket. In today's digital world where there is an endless variety of content to be consumed like books, videos, articles, movies, etc., finding the content of one's liking has become an irksome task. Here recommendation engines will help customers find information, products, and services they might not have thought of. Recommendation applications are helpful in a wide variety of industries and Business. A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and make suggests based on these preferences. A TV Series recommendation system provides a level of comfort and personalization that helps the user interact better with the system and watch Series that are personalized to his wants.

1.1 Description

This project aims to build a TV Series recommendation system which will be the combination of content based filtering with the help of TFID vectorization and collaborative filtering with the help of correlation to predict the TV series. The System analyzes the genre preferability of the user, to recommend him shows based on his genre preferability. The system makes use of TFID vectorization technique for the same. For unique and personalized recommendations, the system uses a Hybrid approach. In the first part we analyze user's content preferences, and then use correlation to find out similar TV shows which a user may like. The hybrid methods thus used can provide more accurate recommendations than pure approaches (Content and Collaborative filtering). These methods can also be used to overcome some of the common problems in TV Series recommendation systems such as sparsity, reduced coverage, scalability, and cold start problems.

1.2 Problem Formulation

Choosing what to watch has always been a question for many. A TV Series Recommendation Engine can be helpful for many as to have machine tell what to watch next. This Online Tv

Series Streaming Website helps to watch online Series with Recommendation system which is one of the stronger tools to increase profit and retaining buyer. Recommendation are already an integral part of many streaming websites like Netflix. Giving recommendation to the users after predicting if the user is likely to be interested in that product has helped a lot of the companies in boosting their sales. Recommendation systems are widely used to recommend products to the end users that are most appropriate. We aim at developing a model for a web-based personalized tv series recommender system where clustering algorithms will be based on their rating, content and reviews by the user, which exploits various aspects of giving recommendation apart from the regular collaborative and content-based filtering approaches. Scrapping information from the web and using the information obtained from this process can be equally useful in making recommendations.

1.3 Motivation

In the past years, a lot of people have taken to watching TV series online. For that a lot of websites such as Netflix, Hulu, Amazon and Sony Liv have been deployed. As it goes without saying a lot of data is feeded to a user when he searches online. So what a user wants is a personalized search, where he gets recommendations based on his likes/ needs. We plan to gain a better understanding of TV, paying particular attention to the type of data likely to be used in choosing a Series to watch, evaluate existing methods (if any) used to help in the recommendation of TV series; Research and evaluate potential software development methodologies and programming languages which can be used in the system, justifying all decisions which are made.

1.4 Proposed System

In this project we are proposing a solution for recommender system. The user when opens the Application user has to Register and login with right credentials. This system uses rather Hybrid approach to recommend the series to the user. This system analyzes the genre preferability of the user, puts the users of similar taste into a cluster using K-Means and finally uses Correlation to provide a recommendation. We combine the features of both content-based filtering technique and also collaborative based filtering. There mainly are 2 algorithms in use TF-id vectorization for content-based filtering and correlation for collaborative filtering. Our hybrid recommendation system combines the output given by these two algorithms and provides the user with a personalized recommendation.

1.5 Scope of the project:

This project helps to recommend TV series for the user saving his time. The application of this system is very wide.

1. Hybrid Model for recommendation
2. We propose to add personalized recommendation if needed by including demographic parameters like user's age, gender, country, language and tune the results further if user needs
3. Using Web Scraping which gives top trending TV-Series some from each genres to the user after parsing from web

Chapter 2

Review of Literature

2.1 Introduction

Recommendation system is used to find user interest and improves user experience. Recommendation systems or recommendation systems are a subclass of information filtering system that seek to predict the ‘rating’ or ‘preference’ that a user would give to an item. Recommendation systems have become extremely common in recent years and are applied in variety of applications. The most popular ones are probably movies, series, books, music, search queries, social tags and products in general. However, there are also recommendation systems for expert, jokes, financial services, life insurance, persons(online-dating), etc.

The paper [1] proposes a way of recommending TV series by analyzing the user’s genre preferability of movies, the genre of TV series and the number of episodes. This system analyzes the genre preferability of the user from the movie data using Fuzzy Inference System, puts the users of similar taste into a cluster using K-Means and finally applies Adaptive Fuzzy Neuro Inference System in the Cluster to predict the rating of that TV series the user might give in real life. The system developed in this paper is the first TV series recommendation system that considers the number of episodes of TV series as an input. The result is promising as the average deviation from the actual rating is significantly lower, but more research can improve the result even further

2.1.1 K-Means algorithm for Clustering:

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K group based on the features that are provided.

2.1.2 Correlation for collaborative:

For collaborative filtering we use Correlation algorithm to find similar users based on their series preference. Correlation is a statistical technique that can show whether and how strongly pairs of variables are related to each other. The main result of a correlation is called the correlation coefficient (or "r"). It ranges from -1.0 to +1.0. The closer r is to +1 or -1, the more closely the two variables are related. We take list of movies watched by the two users and correlation would tell us whether the two users have similar movie preference or not.

The paper [2] proposes a way of recommending TV series to newly registered users, thus dealing with the cold-start problem which normally arises. This paper, brings forward a distributed hybrid recommendation framework. First, current user characteristics, user context and operating records are used to classify the user type. Then, suitable recommendation algorithms are dynamically selected based on the current user type i.e. Collaborative Filtering, Content Similarity, Popularity Estimation. Finally, individual recommendation lists are then, merged into a consolidated list as the final result.

2.1.3 Content Based Filtering:

Content based Filtering methods are based on description of the item and a profile of the user's preference. In a content-based recommendation system, keywords are used to describe the items; besides, a user profile is created to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past. In particular, various candidate items are composed with items previously rated by the user and the best-matching items are recommended.

2.1.4 Demographic Filtering:

DF uses demographic data such as age, gender, education, etc. for identifying categories of users. It does not suffer from the new user problem as is doesn't use ratings to provide recommendations. However, it is difficult today to collect enough demographic information that is needed because of online privacy concerns, limiting the utilization of DF. It is still combined with other recommenders as a reinforcing technique for better quality.

The paper [3] presents a literature overview of TV program recommender systems. Based on related research, a smart and social TV program recommender framework is proposed which

consists of TV program content analysis module, user profile analysis module and user preference learning module. The proposed framework not only processes TV program content and users' direct feedback, but also suggests extracting related information such as TV watching statistics information, users' preference/interest for the other contents from social media or relevant organization. In the preference learning module, they suggest three user preference learning approaches: leaning from individual's past experience, learning from implicit network and learning form explicit network. This paper also addresses several issues, which are important in the building of a TV program recommender system, such as accuracy, diversity, novelty, explanation and group recommendation and show some corresponding solutions for these issues. The proposed framework could be used to help designers/developers to build TV program recommender systems/engines for smart TV.

2.2 Comparative Study:

The table 2.1 is used to compare the 2 approaches content based filtering and collaborative filtering. Also Hybrid approach is studied to eliminate the disadvantages of both the methods.

Table 2.1 Comparative Study

Content based filtering	Collaborative filtering	Hybrid Approach
Content based filtering uses information filtering for their recommendation. It analyzes the preferred item's content in order to predict the unseen items relevance.	Collaborative filtering algorithm provide items recommendations or predict based on the opinions of the active user's neighbor	In hybrid approaches both the approaches for recommender system can be applied independently and then can be added in an appropriate manner
Content based filtering face cold start problem. Their learning stages are based on users' information ,in most cases a user has to input their rating or preferences manually and therefore the collection of this kind of information is hard to be achieved	Collaborative filtering also face cold start problem their learning stages are based on user's information in most cases a user has to input their ratings or preferences manually and therefore the collection of this kind of information is hard to be achieved	Boosting algorithm used in hybrid recommendation can remove cold start problem by using combination of collaborative and demographic algorithm.
Content based filtering stability versus plasticity problem. Stability/plasticity problem means that it is sometimes hard to change establishment user's profiles which have been established after a given period of time using the systems.	Collaborative filtering also face stability versus plasticity problem. Stability/plasticity problem means that it is sometimes hard to change establishment after a given period of time using the systems.	This problem can be solved with the hybrid approach because different type of recommendation technique like knowledge based algorithm can be less affected by the problem. One of the solutions for the latter problem is temporal discount ,which make older rating with less influence

Chapter 3

System Analysis

This chapter gives an idea about various functional and non-functional requirements of proposed solution. It also gives an overview of the hardware and software requirements of the system both on client as well as server side.

3.1 Functional requirements:

Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use case.

1. The User must firstly register and login in the system with right credentials.
2. Users can then select TV-Series from the categories given.
3. User can search for intended Series by providing its name on search bar.
4. User can add Series to their Favorites list.
5. User Can Rate and comment on the Series and express their liking and disliking.

3.2 Non- Functional requirements:

Non-functional requirements will describe how a system should behave and what limits there are on its functionality. It generally specifies the system's quality attributes or characteristics.

1. The system should be compliant to optimize compatibility with other systems.
2. All app pages generated by the system shall be fully downloadable in no more than 10 seconds over 40 Kbps modem speed.
3. Responses to query should not take more than 7 sec to load onto the screen after the user submits the query.

3.3 Specific requirements:

3.3.1 Hardware Requirements:

Client Side

1. 512 MB RAM
2. Android Phone with jellybean 4.0 or above

Server Side

1. Computer with 2TB HDD
2. 3.12 GHz Processor
3. 16GB RAM
4. Must be able to connect to 100 Clients at a time

3.3.2 Software Requirements:

Client Side

1. Our TV show app installed on their Phone.

Server Side

2. OS: Windows, Ubuntu
3. Apache Server
4. MySQL
5. XAMPP
6. Python
7. Android Studio

3.4 Use-Case diagram and Description:

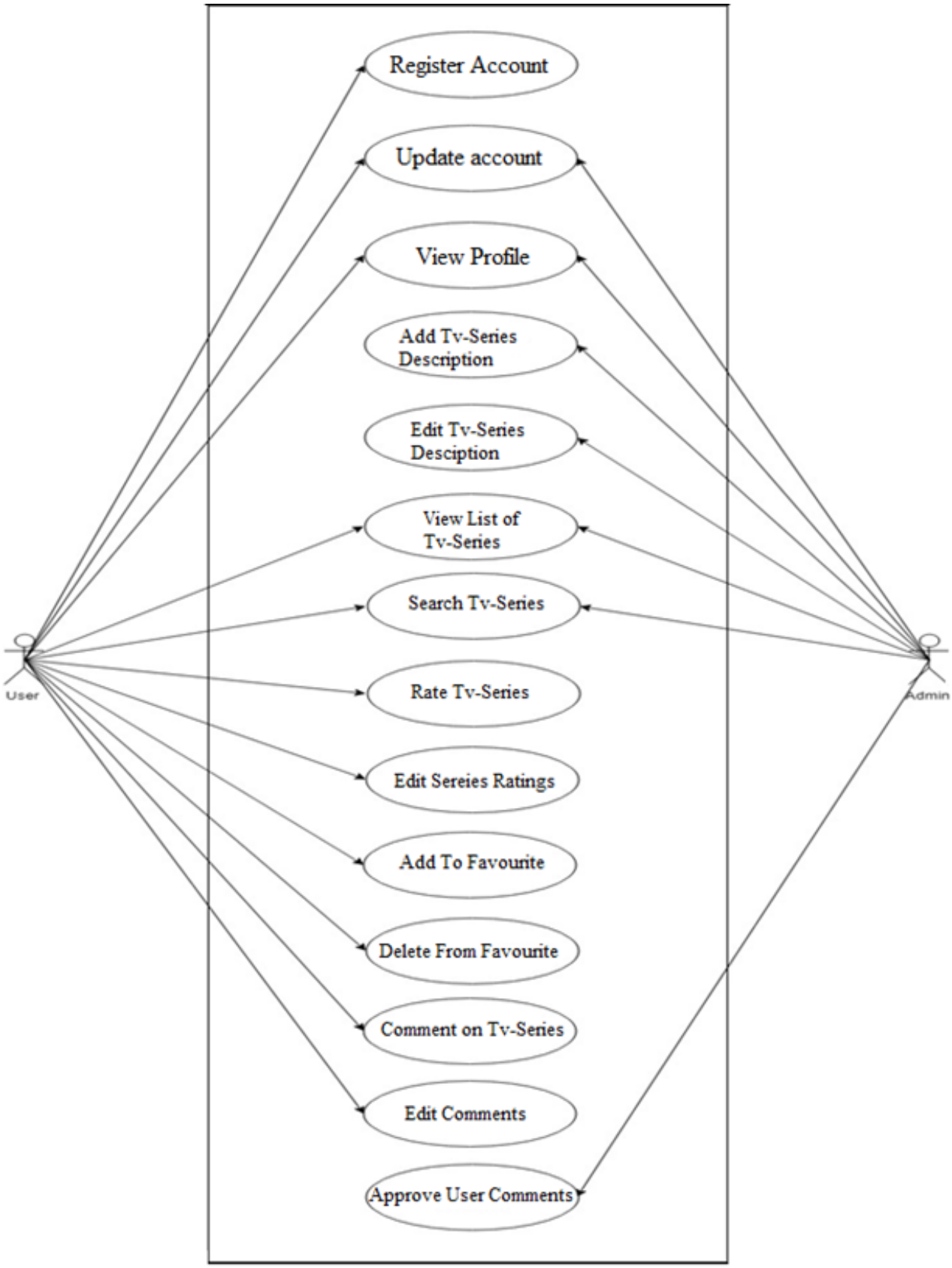


Fig 3.1 Use Case Diagram

The above figure 3.1 illustrates the use case diagram for the TV-Series Recommendation System

1. User will register to the system. Admin will verify the credentials and give access to the user.
2. User will then select or search from the series available.
3. Based on liking user can rate the TV-series from 1 to 5.
4. User can add liked TV-series to favorites.
5. User can comment on the TV-series and the admin can approve users comments.

Table 3.1-UseCase Description

Use case name:	TV-Series Recommendation System
Actor:	User
Brief description:	The user will enter their credentials and search for Tv-series and will get recommended Tv- series.
Precondition:	User must register to the system with right credentials
Post condition:	After getting in the system user will get recommended series.
Priority:	The use case has the highest priority.
Non-behavioral requirements:	High speed internet connection and a desktop.
Assumptions:	The User data and status uploaded is up to date.

3.5 Sequence Diagram:

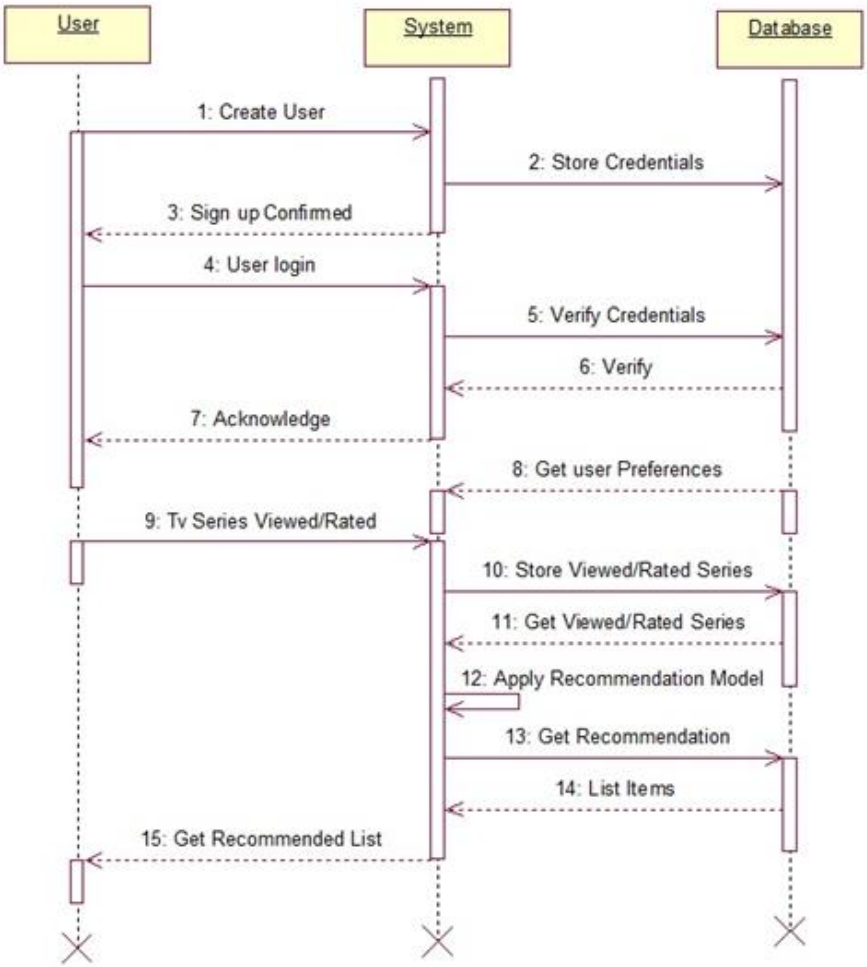


Fig 3.2 Sequence Diagram

1. The above figure 3.2 explains the sequence of how the operations would be performed by our system.
2. At first the user will clear his profile where the system stores his credentials in the database.
3. During login the user will enter his details and the system will verify with the database to authenticate the user and display his content accordingly.
4. As the Users keeps on giving new rating system will keep adding that to the database.
5. All the store data about the user will be later used to improve the recommendation system to provide a better personalized to the user.
6. Finally the generated list of recommended TV shows by the system for the user will be displayed to him.

Chapter 4

Analysis Modeling

4.1. Entity- Relationship Diagram:

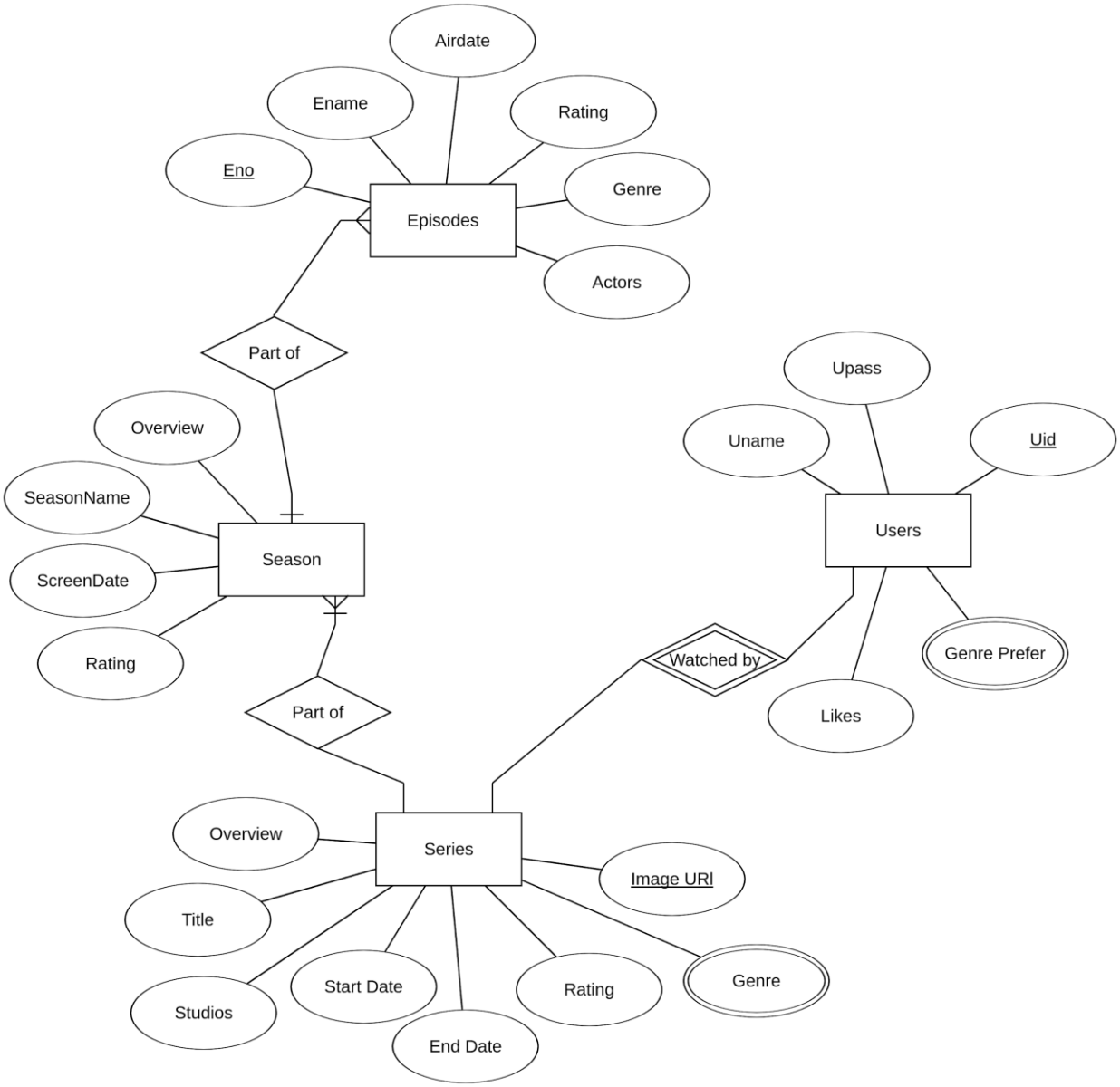


Fig.4.1 ER Diagram

An Entity-Relationship diagram provides a relevant method of displaying this understanding at this stage as it" provides a useful perspective, especially for the purpose of the initial database design". In order to capture the requirements of the database at its most basic level the initial E-R Diagram was taken into account. Already at this high-level decision were made regarding the relationships between entities in terms of cardinality -" the number of

relationship instances that an entity can participate in and dependence -" whether the existence of entity depends on its being related to another entity". Many Episodes for each Season and each Season has many episodes. Accordingly, a one-to-many relationship was created for these entities.

4.2. Data flow Diagram

4.2.1 Level -0 DFD

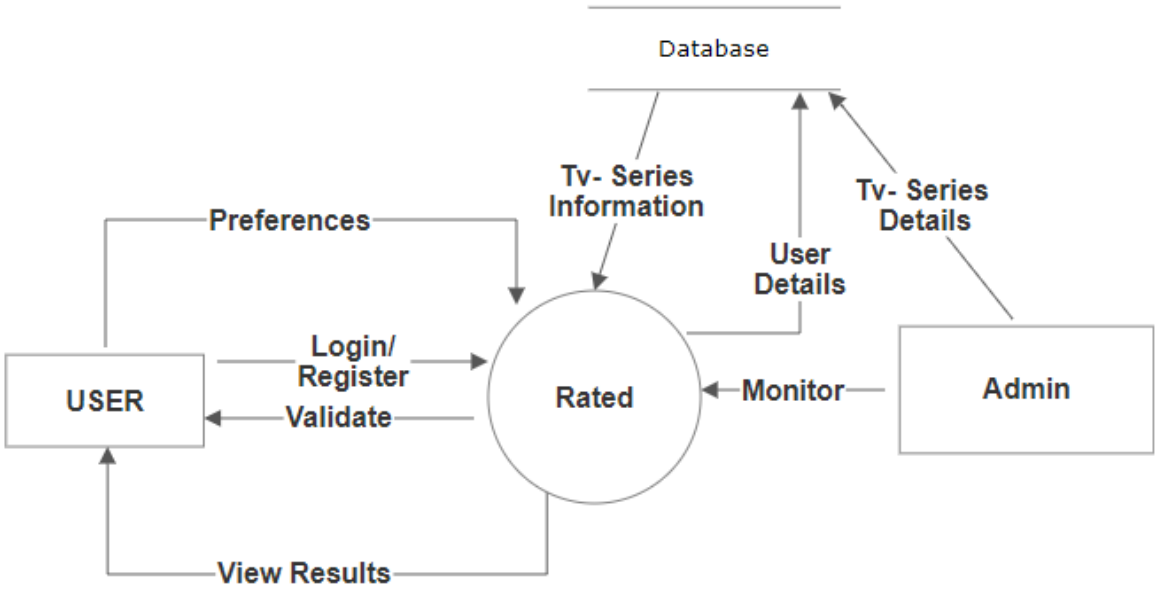


Fig.4.2 Level-0 DFD

The above Fig.4.2 Level-0 DFD explains the basic data flow of TV-Series recommendation system "Rated". The User Registers/logs in the system with right credentials. The Admin can update the TV series details in the database which is retrieved by the system as per need. The Admin Also monitors the recommendation system for right responses and bugs and errors. While registering the user can select their TV-series Preferences. The user will view recommendation from the system.

4.2.2 Level -1 DFD

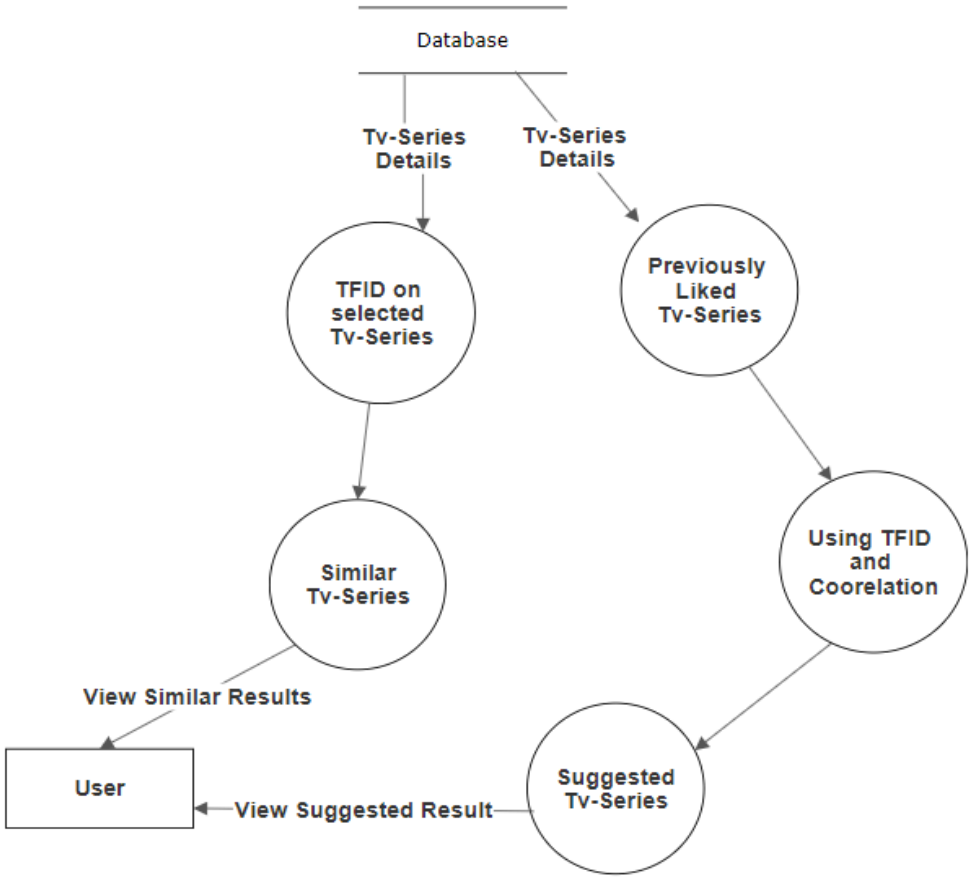


Fig.4.3. Level-1 DFD

The above Fig.4.3 Level-1 DFD explains the data flow of TV-Series recommendation system "Rated" . The system uses the information about TV Series from their database and get previously liked TV-series then using TFID algorithm and correlation on those shows gives best recommendation of TV series for the user. When any user selects any show in the system the system also shows Similar TV-shows by using TFID algorithm on selected series.

4.3. Activity Diagram:

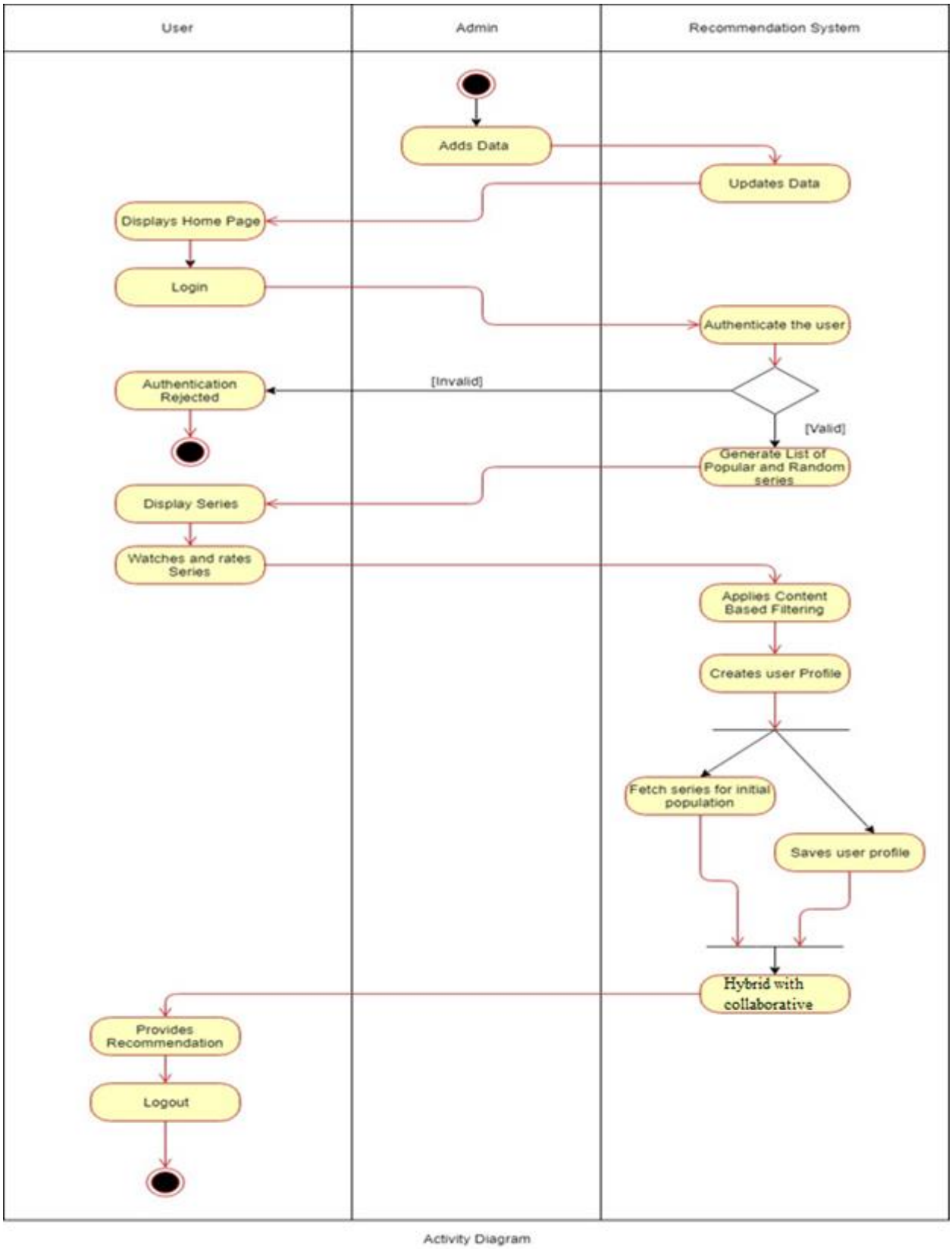


Fig.4.4 Activity Diagram

Figure 4.4 is the activity diagram for the recommendation system that we have proposed. The Admin handles the data input in the system. The system works to authenticate the user, handles data updating and uses the various algorithms to filter and recommend the perfect search for a user.

4.4. Timeline Chart:

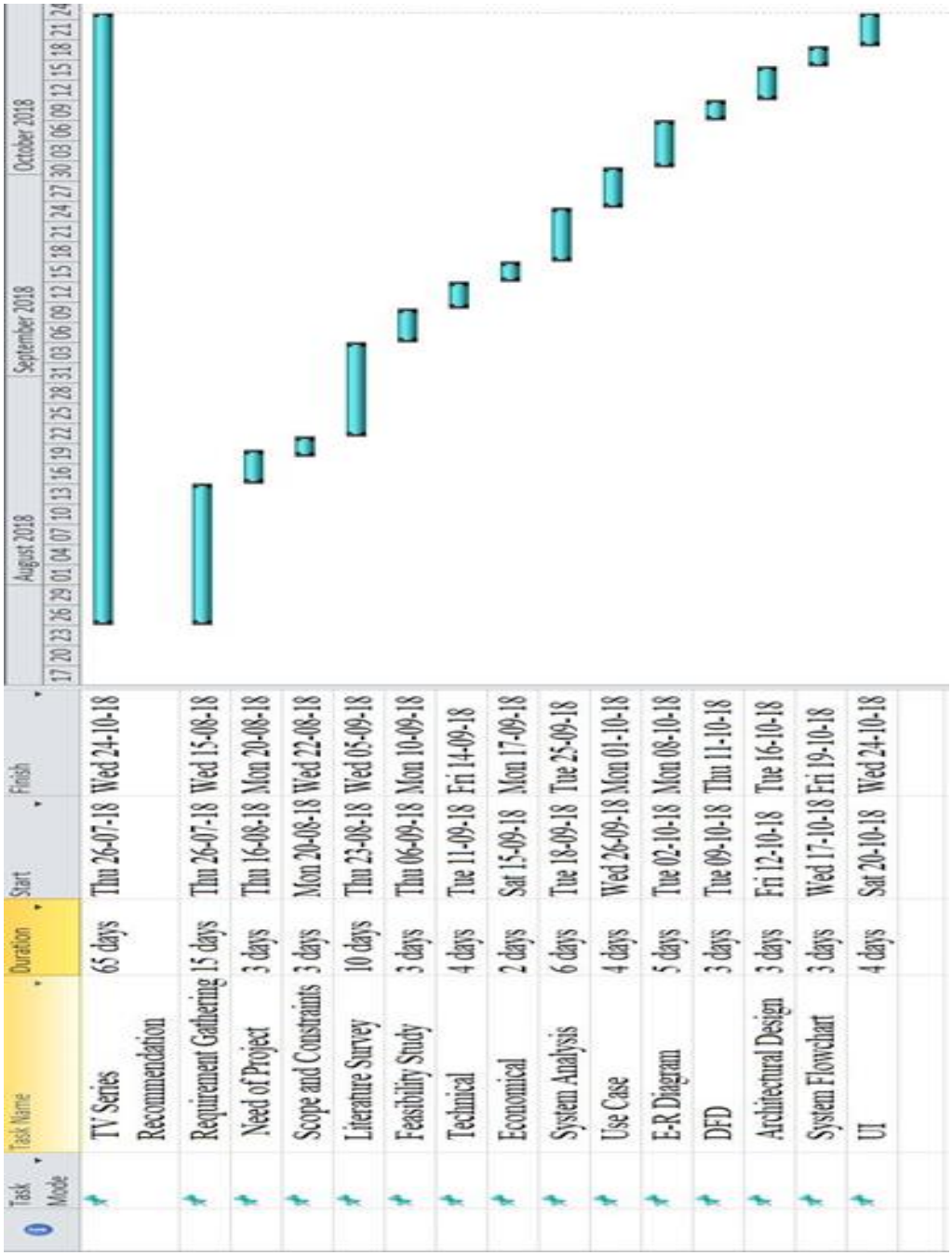
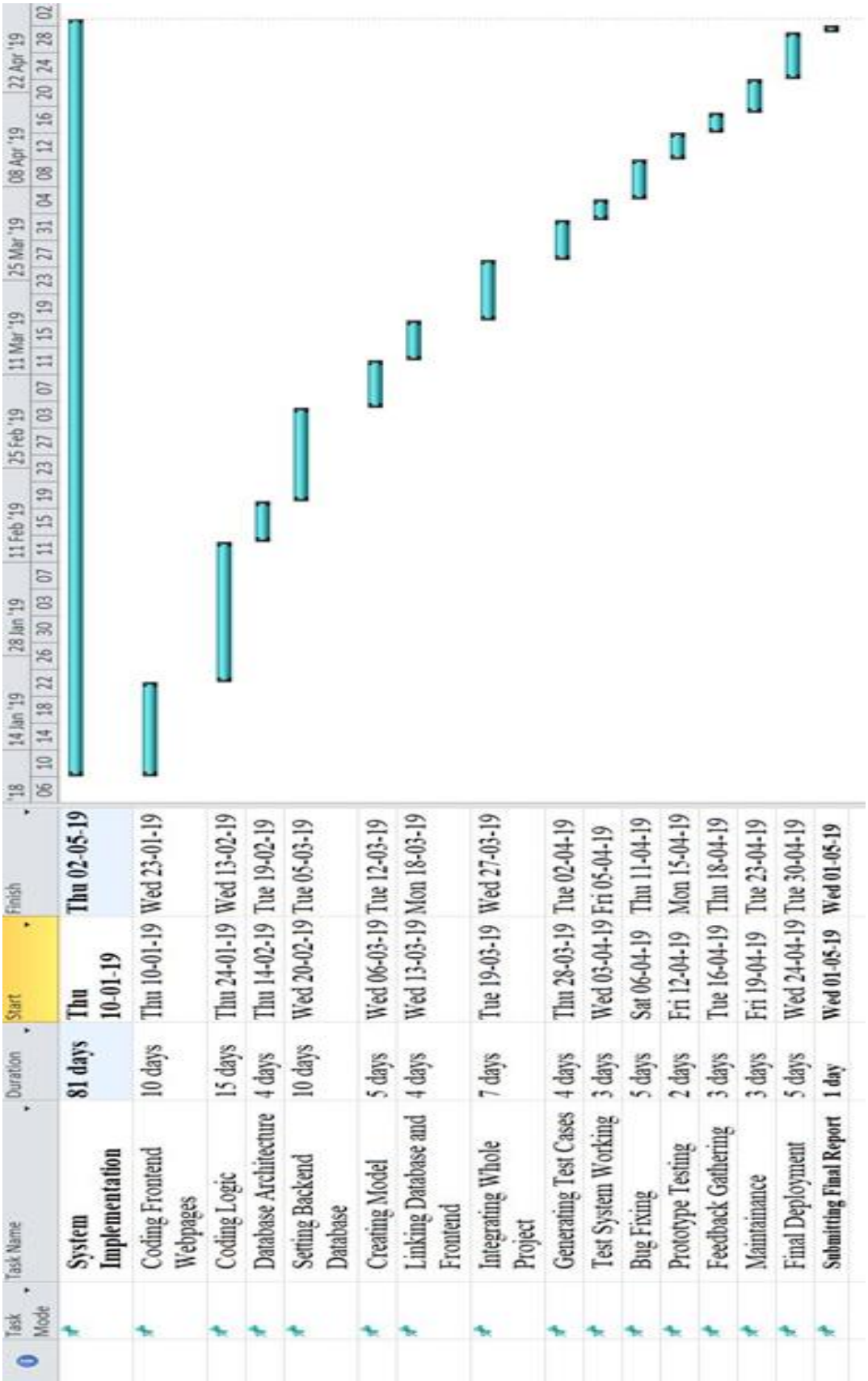


Fig.4.5 Review Timeline Chart



Task Mode	Task Name	Duration	Start	Finish
	System Implementation	**81 days**	**Thu 10-01-19**	**Thu 02-05-19**
	Coding Frontend Webpages	10 days	Thu 10-01-19	Wed 23-01-19
	Coding Logic	15 days	Thu 24-01-19	Wed 13-02-19
	Database Architecture	4 days	Thu 14-02-19	Tue 19-02-19
	Setting Backend Database	10 days	Wed 20-02-19	Tue 05-03-19
	Creating Model	5 days	Wed 06-03-19	Tue 12-03-19
	Linking Database and Frontend	4 days	Wed 13-03-19	Mon 18-03-19
	Integrating Whole Project	7 days	Tue 19-03-19	Wed 27-03-19
	Generating Test Cases	4 days	Thu 28-03-19	Tue 02-04-19
	Test System Working	3 days	Wed 03-04-19	Fri 05-04-19
	Bug Fixing	5 days	Sat 06-04-19	Thu 11-04-19
	Prototype Testing	2 days	Fri 12-04-19	Mon 15-04-19
	Feedback Gathering	3 days	Tue 16-04-19	Thu 18-04-19
	Maintainance	3 days	Fri 19-04-19	Tue 23-04-19
	Final Deployment	5 days	Wed 24-04-19	Tue 30-04-19
	Submitting Final Report	**1 day**	**Wed 01-05-19**	**Wed 01-05-19**

Fig.4.6 Implementation Timeline Chart

Chapter 5

Design

5.1 Architectural Design

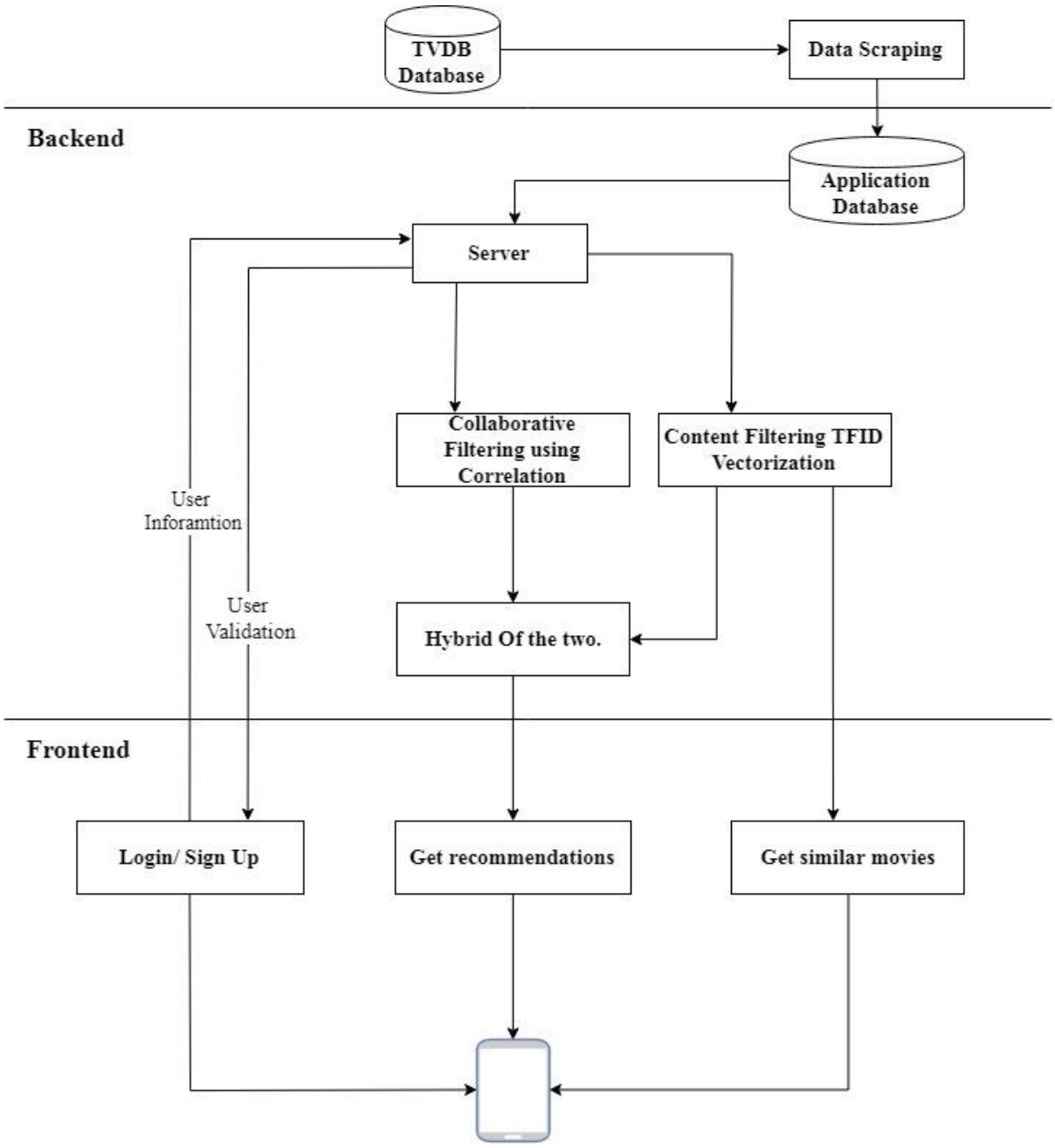


Fig 5.1 System architecture design.

In this Figure 5.1, we describe the system architecture. The user when opens the Application user has to Register and login with right credentials. This system uses rather Hybrid approach to recommend the series to the user. This system analyzes the genre preferability of the user, puts the users of similar taste into a cluster using Correlation to provide a recommendation. We combine the features of both content-based filtering technique and also collaborative based filtering. There mainly are 2 algorithms in use TF-id vectorization for content-based filtering which gives similar shows to user preference genres and correlation for collaborative filtering to recommend.

5.1.1 System Flow:

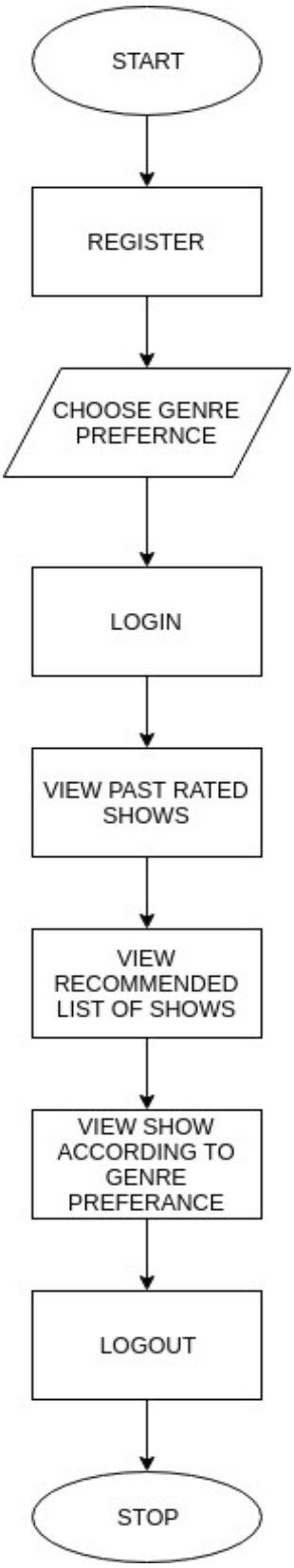


Fig 5.2 System Flow

The above figure 5.2 shows the system flow of our project. The user will first have to register and choose genre preference if the user is new else the user can login in the system using username and password. The user can view his past rated shows, also the list of recommended shows and also view shows according to his genre preference .The user can Logout at the end.

5.2. User Interface Design GUI

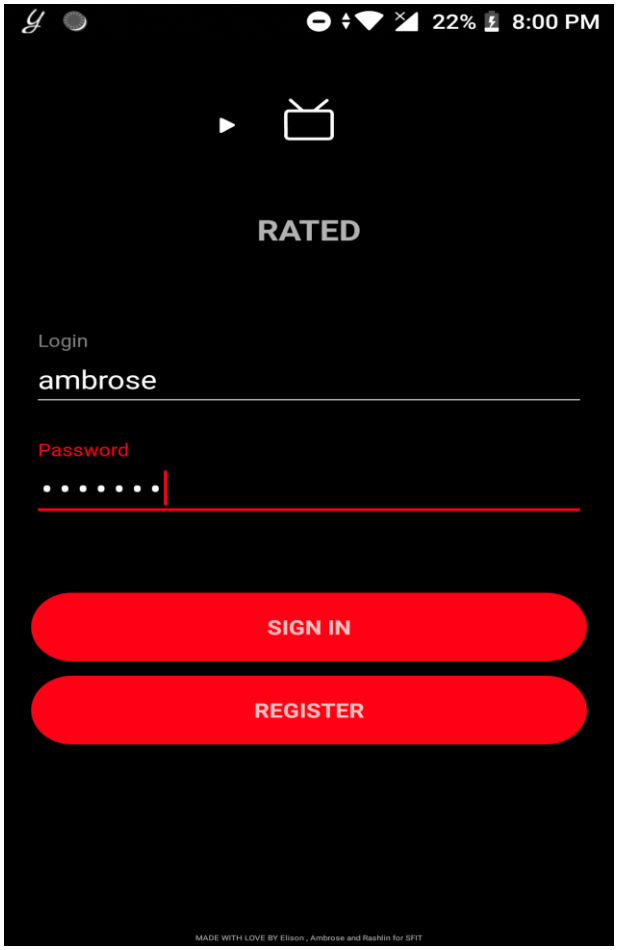


Fig 5.3 Sign in/ Register

In the above figure 5.3 one can sign in with an already created account or register for a new account via the respective buttons. Here we are using an existing account to sign in.

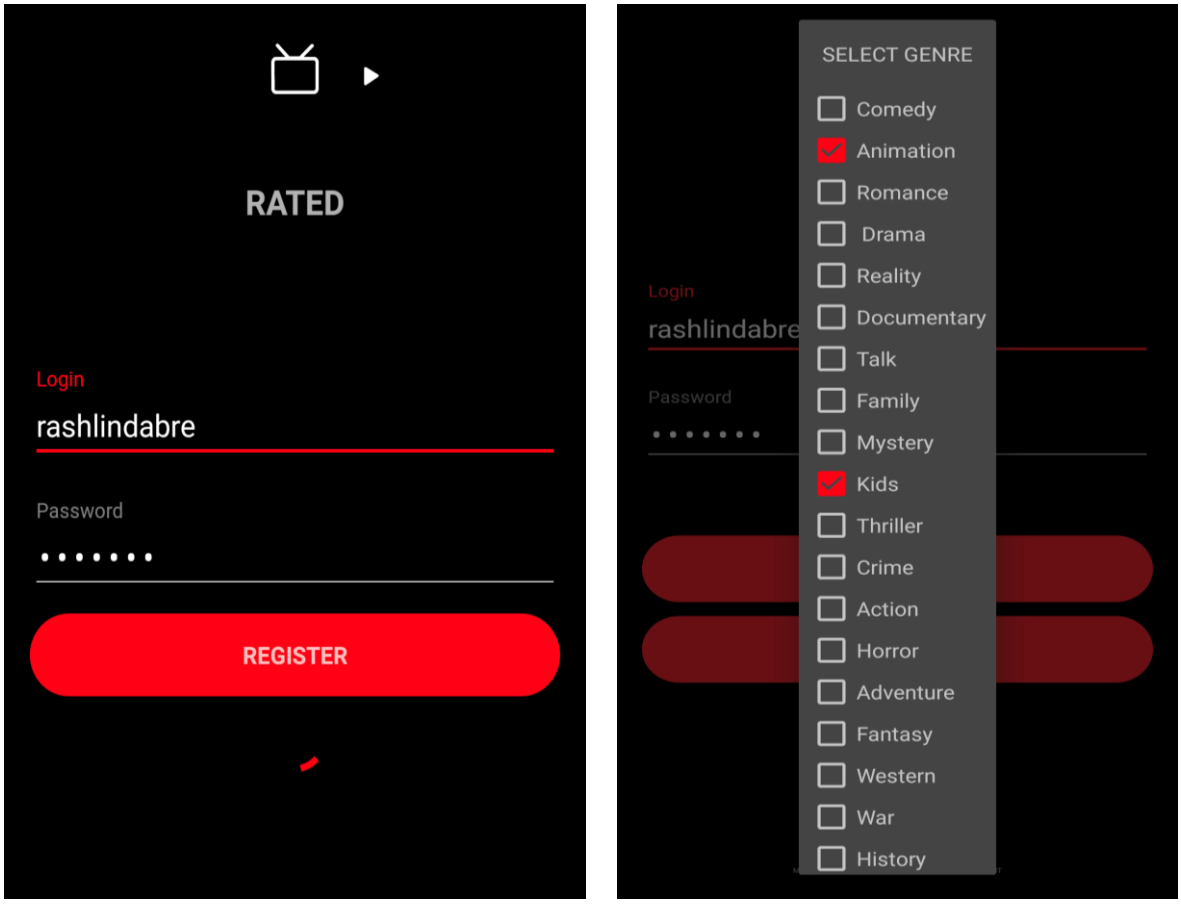


Fig 5.4 Registering New User

In figure 5.4 new User can sign in by pressing on the Register button, users get to choose from a list of Genres so that the system gets to know their preferences better.

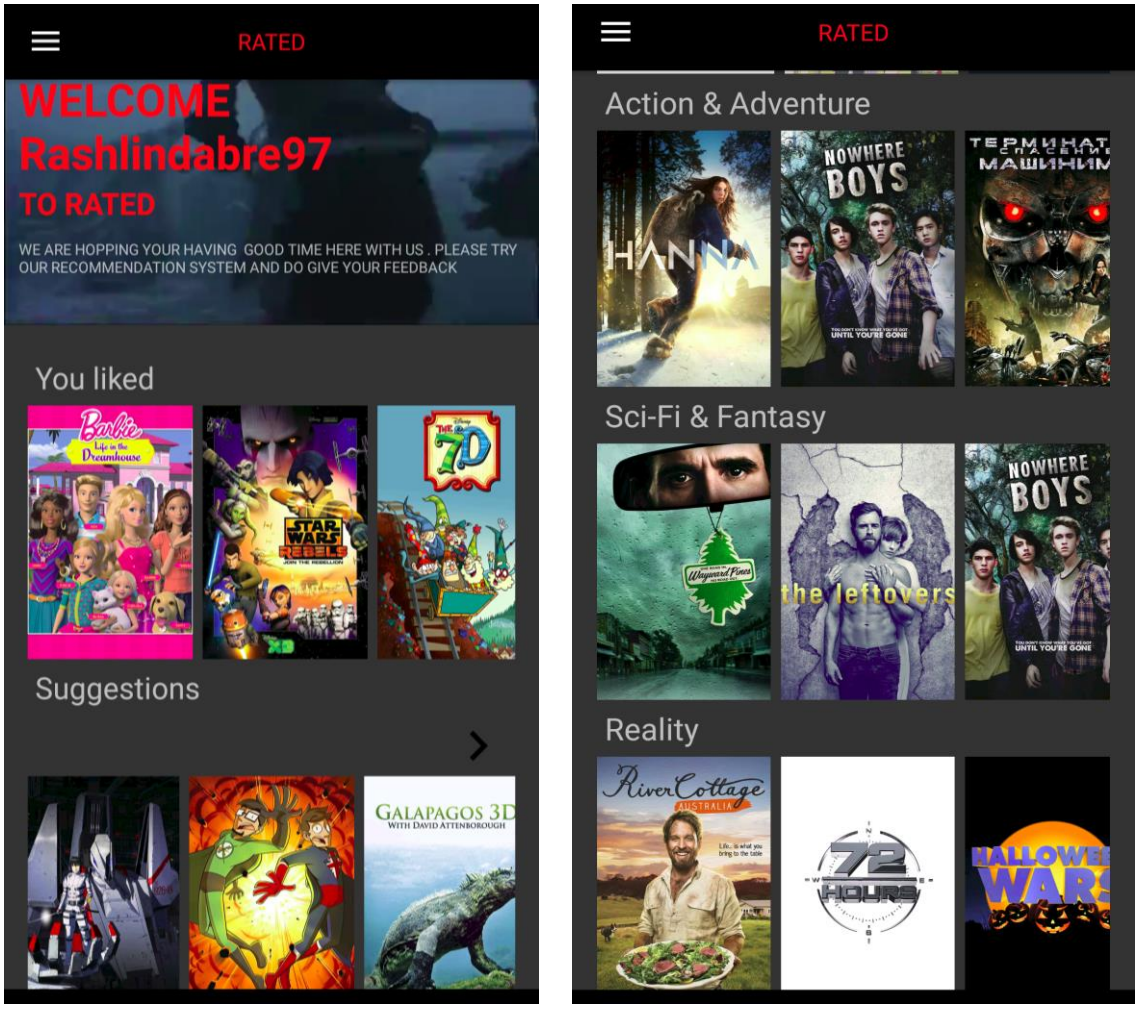


Fig 5.5 Homepage

In above figure 5.5 the homepage lists out TV Series on various categories as per their genres. Also unique suggestions are shown to users based on his previous likes.

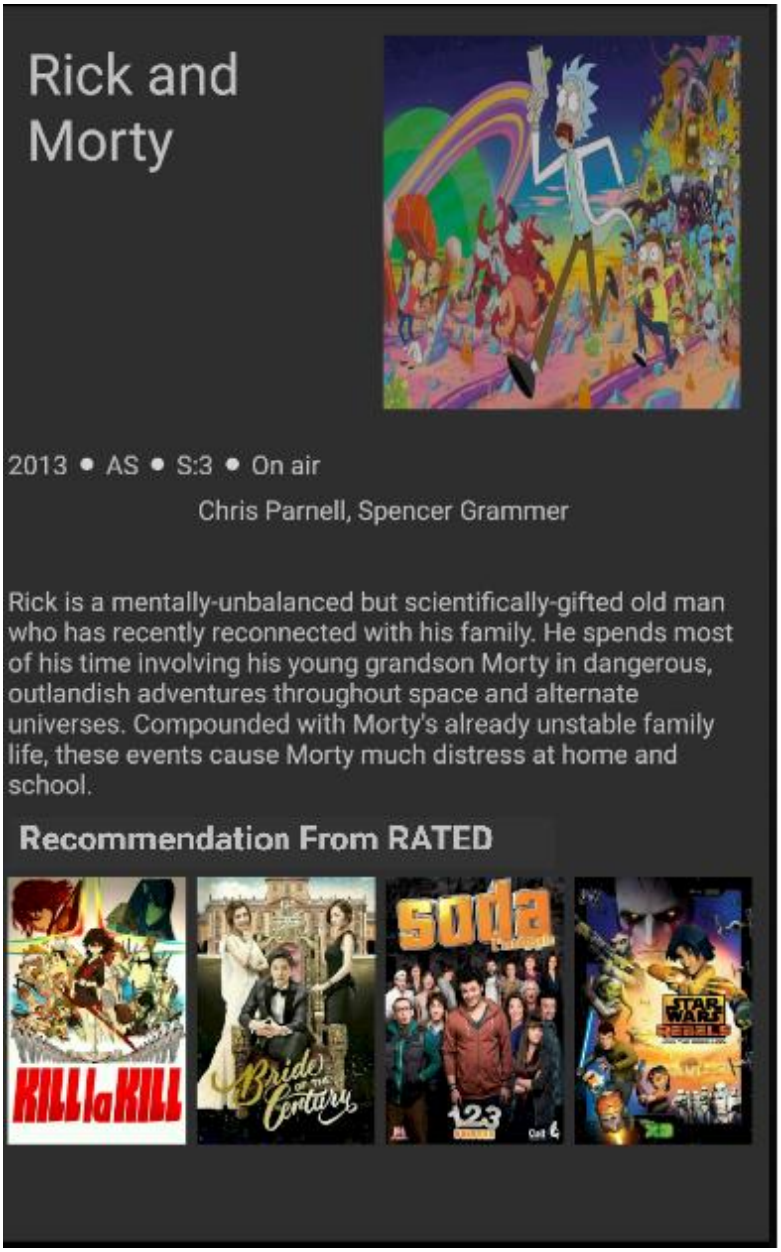


Fig 5.6: Content Filtering for specific TV Show

The above figure 5.6 shows details of the TV show user clicks on and does content filtering for same. Here we have showed example for Rick and Morty.

Chapter 6

Implementation

6.1 Algorithms / Methods Used:

To construct a Hybrid recommendation system, we need to combine the features of both content-based filtering technique and also collaborative based filtering. Combining these two techniques will give us an efficient recommendation system. There mainly are 2 algorithms in use TF-id vectorization for content based filtering and correlation for collaborative filtering. Our hybrid recommendation system combines the output given by these two algorithms and provides the user with a personalized recommendation. The steps to be followed are given below.

Step 1: Scraping data and cleaning for use

The data required to construct a recommendation system for TV shows was taken from The TVdb website. They provide and open api for use. Using the api we scrap the data and use it accordingly in our system. The scraped data is huge which contains features like genre, cast, crew, overview, etc. of TV shows. All though we can use majority of it in our system we would reduce the unnecessary features and clean it for faster and better computation in our system. Now the clean data is used in our algorithms to provide us with appropriate output.

Step 2: TF-id for content-based filtering

Content based filtering is a filtering technique where the for the given input data the output is given such that they show similarity in content such as genre, overview, cast etc. We use TF-id vectorization algorithm to do content-based filtering. Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

Tf-idf uses product of Tf (t) and Idf(t) to find weight of the word. where t stands for the word.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
IDF (t) = log_e (Total number of documents / Number of documents with term t in it).
Here a particular TV show and all its features are given as a input to the system example a TV show called friends which lies in comedy genre has a funny overview about 6 friends .The system would take these features such as genre, overview, number of episodes, cast and find TV shows who are similar in content with these features and provide us with a list of similar shows. Example for above input friends we would get output such as Big band theory, how I met your mother, etc.

Step 3: Correlation for collaborative filtering

Collaborative filtering takes users past watched TV shows into consideration and gives output accordingly. In collaborative filtering we check what all TV shows the user has watched liked and disliked and find similar users in order to recommend. The idea behind this filtering technique is that two similar users would have same taste of series so what one user has liked can be recommended to another user who is yet to watch that show.

We use Correlation algorithm to implement collaborative filtering in our system. Correlation is a statistical technique that can show whether and how strongly pairs of variables are related to each other. The main result of a correlation is called the correlation coefficient (or "r"). It ranges from -1.0 to +1.0. The closer r is to +1 or -1, the more closely the two variables are related. We take an array of TV shows previously watched by user and liked. Then we compare them with the array of watched shows of all the other users in our system. Putting these 2 array in our correlation algorithm we get output whether they are similar or nor . accordingly a list of similar users a found by our system .

Step 4: Hybrid approach

Now we have implemented content-based filtering with the help of Tf-if vectorization and collaborative with the help of correlation. So, we need to combine these outputs so that Our recommendations can be versatile and also helps to reduce cold start problem. Here once we have a list of similar user and the shows that they have watched liked we compare and find those TV shows that our user has not watched yet .Putting this list as an input to content filtering we would get TV shows similar to those content and all of these shows are put in and

array .This is our personalized recommendation array which is displayed to the user on his android application.

6.2 Working of Project:

The TV Series recommendation system uses hybrid approach

Following is the Workflow of the project

1. Scrap dataset from using TVdb api
2. Clean the data for better computation capability
3. Use Tf-if to provide content-based filtering for a given TV show as input
4. Use correlation for collaborative filtering using user's history
5. Combine the two-given output with hybrid approach
6. Display the output on android application

Chapter 7

Testing

The testing is performed based on the algorithms used in the system. The main flow of the system depends upon whether the used algorithms function properly or not. White Box testing is performed to test the functionality of the system. Functional testing is done on three main steps to guarantee proper working of the system.

- Security (Identification of the user)
- Data flow

7.1 Test Module 1: Security

The security of the database is maintained by avoiding unauthenticated access by any attacker. Whenever person wants to access the data or view history, he first needs to login which is the matched with the values stored in database. If the values match the person is authenticated and granted an access.

Table 7.1: Security Test Table

Validation Module	Security
Scenario	Securing User's history
Triggering Event	Attempt for login
Brief Description	To avoid any data theft via identity spoofing, the login details is considered.

Related Modules	User	
Flow of Activities	Actor	System
	1.User is assigned with user name and password .	1.1. System accepts the details
		1.2. The received details are verified
		1.3. If details match with stored details grant an access
Conclusion	If unauthorized user tries to access with user id, he cannot get access to users account for password which is secured by hiding it while entering. So unauthorized authentication is prevented.	

7.2 Test Module 2: Data Flow

The flow of data into the system is must to show proper information when asked. All modules are interlinked therefore changes made in one module must be reflected in the other without any inconsistency.

- User -> Shows -> Similar shows
- System -> Recommendation

One of the flow is tested below where the user clicks on a show and then view similar shows.

Table 7.2: Data Flow Test Table

Validation Module	Data Flow	
Scenario	Making proper data available to the user	
Triggering Event	User Clicks on a TV-Series	
Brief Description	When the Shows details are shown it also shows similar shows to the chosen shows. .	
Related Modules	User,	
Flow of Activities	Actor	System
	1. User enters the System.	1.1. System gets all the details of the shows using previous preferences, Genres, recommendation.
	2. Tiles of tv-shows are displayed.	2.1. Thelist of Tv-series is shown to the user.
	3. User clicks on the tile of their choice.	3.1. Displays Tv-show Details as well as similar shows to selected show.
Conclusion	With the help of similarity feature it becomes easy for user to know similar shows to the selected shows and also can see the details of the Tv-series.	

7.3 Miscellaneous Tests

Testing the working of some of the features provided by the system. To carry out this testing the test cases are developed for each of the modules in the system.

Table 7.3: Test Case Template for authorization procedure

Project Name: Rated Recommendation System	
Test Case Template	
Test Case ID:001	**Test Designed by:** AMBROSE T.
Test Priority (Low/Medium/High): High	**Test Designed date:** 3rd April, 2019
Module Name: User	**Test Executed by:** RASHLIN D..
Test Title: To check whether user is authorized.	**Test Execution date:** 3rd April, 2019
Description: Test the authentication procedure	
Pre-conditions: User must have internet, username and password	
Dependencies: Internet	

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Start	------	------	------	------
2	Is Internet on?	------	The Internet must be enabled	The Internet is enabled	Pass
3	Is user registered?	------	User must be registered	User is registered	Pass

4	User enters username and password	User name received (Rashlin) Password received (Rashlin)	Login details should be correct and password should be hidden with *	Login details are correct and password is hidden with *	Pass
5	Validation of details	------	Matching the credentials with the ones stored in database.	Matching the credentials with the ones stored in database.	Pass
6	Data retrieved		Login successful	Login Successful	Pass

Table 7.4: Test Case Template for data retrieval procedure

Project Name: Rated Recommendation System	
Test Case Template	
Test Case ID:002	**Test Designed by:** ELISON T.
Test Priority (Low/Medium/High): High	**Test Designed date:** 4th April, 2019
Module Name: User	**Test Executed by:** AMBROSE T.
Test Title: To check whether user can view the images.	**Test Execution date:** 5th April, 2019
Description: Test the Image Display	
Pre-conditions: User must have Internet.	
Dependencies: Internet.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Start	------	------	------	------
2	Is Internet on?	------	The Internet must be enabled	The Internet is enabled	Pass
3	Is user registered?	------	User must be registered	User is registered	Pass
4	User enters username and password	User name received (Ambrose) Password received (Ambrose)	Login details should be correct and password should be hidden with *	Login details are correct and password is hidden with *	Pass
5	Validation of details	------	Matching the credentials with the ones stored in database.	Matching the credentials with the ones stored in database.	Pass
6	Image viewing		Able to see Thumbnail images	Able to see Thumbnail images	Pass

Table 7.5: Recommendation test

Project Name: Rated Recommendation System	
Test Case Template	
Test Case ID:003	**Test Designed by:** ELISON T.
Test Priority (Low/Medium/High): High	**Test Designed date:** 4th April, 2019
Module Name: User	**Test Executed by:** ELISON T.
Test Title: To check whether user can view the recommended shows	**Test Execution date:** 5th April, 2019
Description: Test the Recommendation	
Pre-conditions: User must have Internet.	
Dependencies: Internet.	

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Start	------	------	------	------
2	Is Internet on?	------	The Internet must be enabled	The Internet is enabled	Pass
3	Is user registered?	------	User must be registered	User is registered	Pass
4	User logs in the system	User name received (Elison) Password	Login details should be correct and password should be	Login details are correct and password is hidden with *	Pass

		received (Elison)	hidden with *		
5	User clicks on suggested	------	List of Suggested Tv-shows should be shown with their details.	List of Suggested Tv-shows should be shown with their details.	Pass

Table 7.6: Test for similar shows

Project Name: Rated Recommendation System	
Test Case Template	
Test Case ID:004	**Test Designed by:** RASHLIN D.
Test Priority (Low/Medium/High): High	**Test Designed date:** 4th April, 2019
Module Name: User	**Test Executed by:** ELISON T.
Test Title: To check whether user can view similar shows	**Test Execution date:** 5th April, 2019
Description: Test the Similar Shows	
Pre-conditions: User must have Internet.	
Dependencies: Internet.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Start	------	------	------	------
2	Is Internet on?	------	The Internet must be enabled	The Internet is enabled	Pass
3	Is user registered?	------	User must be registered	User is registered	Pass
4	User logs in the system	User name received (Elison) Password received (Elison)	Login details should be correct and password should be hidden with *	Login details are correct and password is hidden with *	Pass
5	User clicks on a show	------	List of Similar Tv-shows should be shown with their details on the lower end of the page.	List of Similar Tv-shows should be shown with their details on the lower end of the page.	Pass

Chapter 8

Results and Discussions

8.1 Content Filtering:

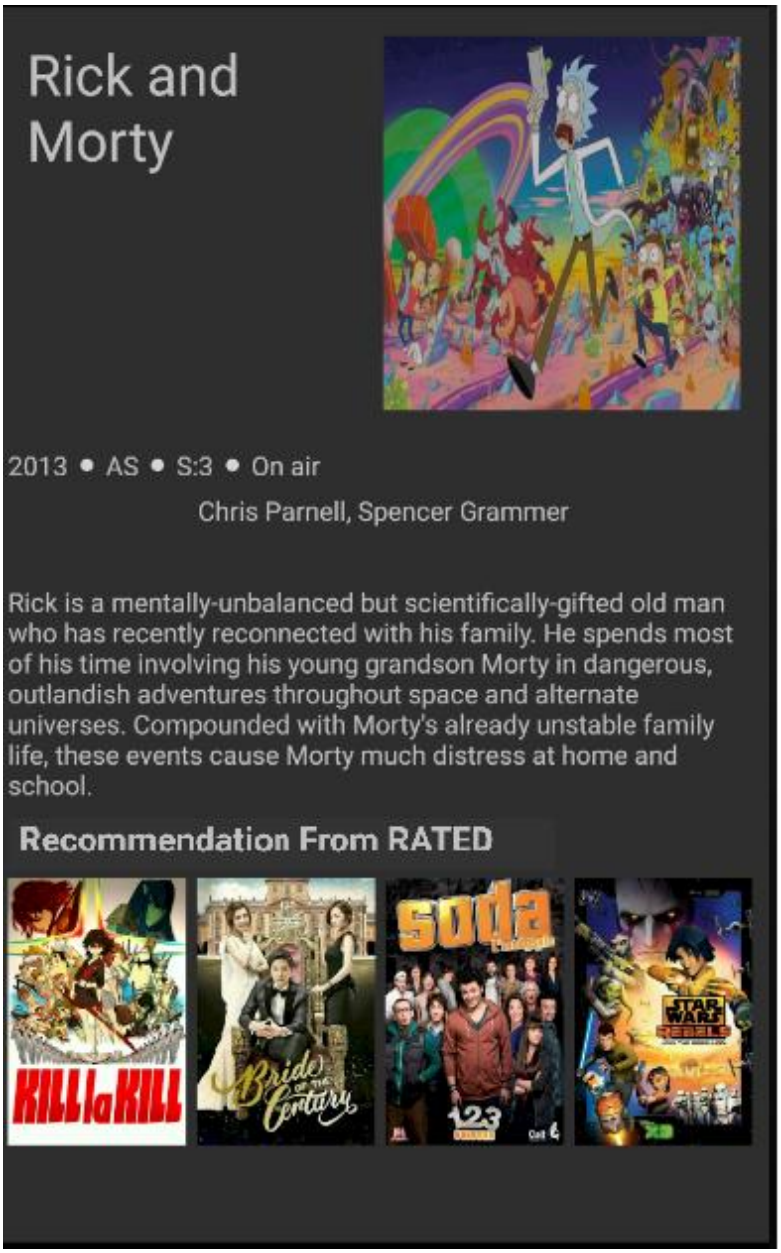


Fig 8.1: Content Filtering for Rick and Morty

The above figure 8.1 shows content filtering for Rick and Morty the user has selected the show Rick and Morty. Its image and name are displayed at top. Below it is the description

of the TV show selected. Below that is shows displayed with similar content. Here User selected show is Sci-Fi, animated and adventures. The recommendations provided are He gets kill-bill which is an Sci-Fi and animated show, soda which is Sci-fi and adventure series ad star wars which is sci-fi, animated and adventure TV show. Hence, we can say that content filtering is done properly here.

8.2 Suggestions:

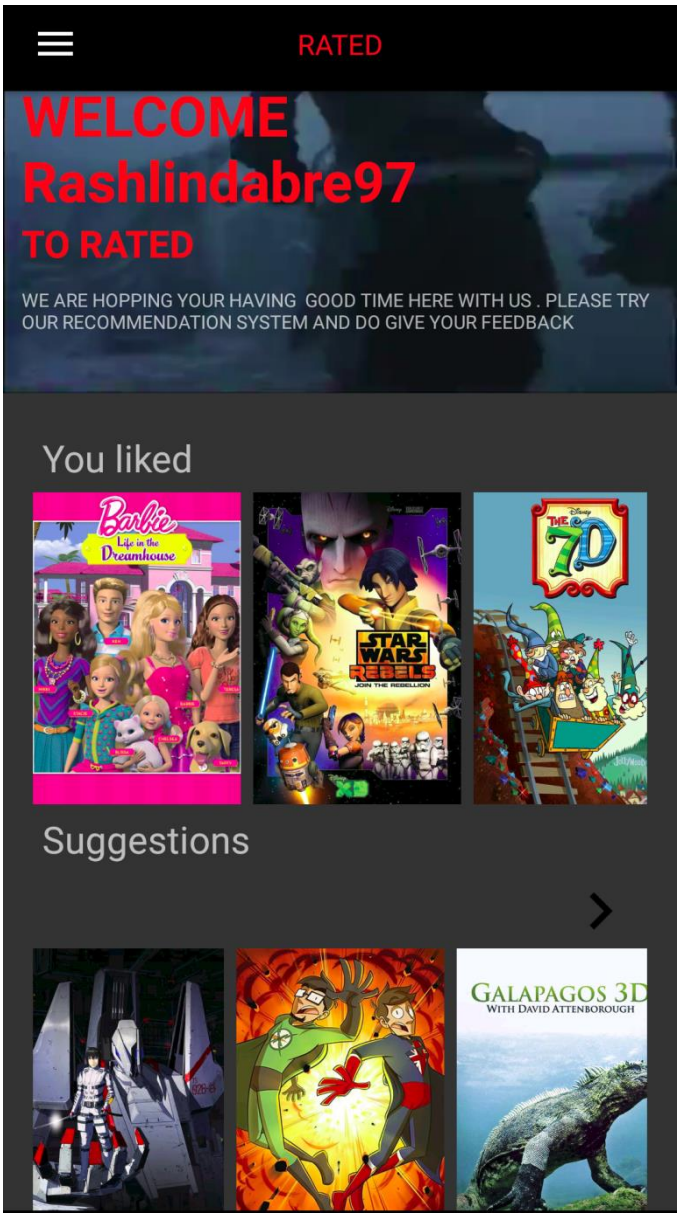


Fig 8.2: Suggestions

The above figure 8.2 shows suggestions to user Rashlindabre97. List of liked videos of Rashlindabre97- Barbie, an epic tale of a fantasy in a dollhouse, Star Wars Rebels a sci-fi animation and Disney's 7D which is a Animation of adventures of seven Dwarfs. All of them focus on Children related shows with genres in line of Animation, Sci-Fi and Fantasy. Our suggestions to Rashlindabre97 are Full metal which is an Anime based Animation, Lanterns 2 which is a Sci-Fi based animation show and Galapagos 3D an Animal based animation targeted towards children. Hence, we can say that recommendations on users past history is working fine.

Chapter 9

Conclusions

In this work we have presented concept of recommendation system for tv series. The goal of recommendation system is to provide the optimal list of tv shows which are close to his/her type of liking. We have used Tf-id vectorization to provide content-based filtering and correlation for collaborative based filtering. Finally using hybrid approach, we combine the benefits of both these filtering techniques and provide a personalized recommendation for the users. This recommendation can be further made dynamic with web crawling where new series data and rating can be retrieved with the help of web crawling tools. Also, we can add video links to their respective tv show so that suggested tv show can be watched from the application itself.

References

[1]Muyeed Ahmed, Anirudha Paul, Mir Tahsin Imtiaz, Md. Zahid Hassan, Shawon Ashraf,Rashedur M Rahman (2017) " TV Series Recommendation using Fuzzy Inference System ,K-means Clustering and adaptive Neuro Fuzzy Inference System" *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery 978-1-5386-2165-3/17/*

[2]Jenq-Haur Wang and Yi-Hao Chen (2015) "A Distributed Hybrid Recommendation Framework to Address the New-User Cold-Start Problem" *978-1-4673-7211-4/15 $31.00 © 2015 IEEE DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.307*

[3]Shreya Agrawal, Pooja Jain "A TV program recommender framework",*17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems -KES2013*

[4] https://tvdb/swagger

The movie database (2018, December 26) dataset of 10 thousand tv shows from

[5]Datacamp website (2018, October 10) reference from

https://www.datacamp.com/community/tutorials/recommender-systems-python

[6]Siddiquee, M. R., Haider, N., & Rahman, R. M. (2015)." Movie Recommendation System Based on Fuzzy Inference System and Adaptive Neuro Fuzzy Inference System." *International Journal of Fuzzy System Applications, 4(4), 31-69. doi:10.4018/ijfsa.2015100103*

Acknowledgements

A project is an activity that needs involvement of a lot of people from different cadres and calibres. Hence it would be highly unfair on our part to claim credit for the whole project. With due respect, we express our sincere gratitude to our esteemed Principal Dr. Sincy George for this teaming experience and for giving us an opportunity to work on this project.

We sincerely want to thank our project guide Ms. K. Priya Karunakaran who is Project Co-Ordinator as well for her valuable guidance and advice she has given us when faced difficulties while working on the project. We are also grateful for her constructive criticism and for helping us develop this idea to pursue the project.

We would also like to thank our HOD Dr. Kavita Sonawane and the Computer Department staff who have helped us throughout this project. We are fortunate enough to get encouragement and support from all teaching staff of computer department. We would also like to extend our sincere regards to all the non-teaching staff of computer department for their timely support.