

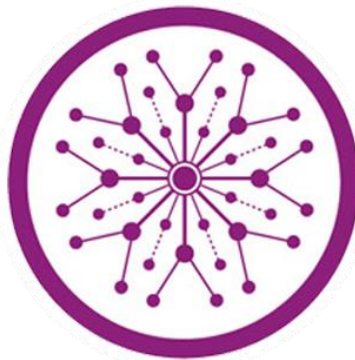
Movie Recommender System

Final Year Project

Session 2019-2023

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	[☒] Development [☒] Research [☒] R&D			
Area of specialization	Web Development			
FYP ID	FYP-BCSM-F22-022			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	Bcsm-f19-414	Bilal Shahid	bcsm-f19-414@superior.edu.pk	
(ii)	Bcsm-f19-476	Atif Mushtaq	bcsm-f19-476@superior.edu.pk	
(iii)	Bcsm-f19-233	Usama Iftikhar	bcsm-f19-233@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I Bilal Shahid S/D of Khushamad Ali Shahid, group leader of FYP under registration no FYP-BCSM-F22-022 at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: 13 July 2023

Name of Group Leader: Bilal Shahid

Signature: _____

Name of Supervisor: Ms Arshia Naeem

Co-Supervisor: Mr. Talha Amjad

Designation: Lecturer

Designation: Lecture

Signature: _____

Signature: _____

HoD: Dr. Irfan-u-din

Signature: _____

Movie Recommender System

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

I would like to dedicate this project to my family, whose unwavering support and encouragement have been my driving force throughout this journey. Their belief in me has been a constant source of motivation.

I would also like to express my heartfelt gratitude to my supervisor Ms. Arshia Naeem, whose guidance, expertise, and valuable insights have shaped this project and helped me navigate through challenges.

Furthermore, I extend my appreciation to my friends and classmates who have provided me with their assistance, constructive feedback, and moral support along the way.

Lastly, I am grateful to the Superior University for providing me with the platform and resources to pursue my passion for Movie Recommendation System. This project is a culmination of the knowledge and skills I have acquired during my time here.

To all those who have contributed to the development and success of this project, I am truly thankful. Your involvement has made this journey meaningful and memorable.

Acknowledgements

I am really thankful to my supervisor and the team of Innovation who helped me achieve competition for this project.

Executive Summary

The Movie Recommender System is a web platform on which user can watch online movies and series with hybrid recommendation technique and recommendation system for the user by using user experience. We are also giving our project the advance machine learning recommendation techniques which will help user to experience recommendations without any long tail problem.

Table of Contents

Plagiarism Free Certificate	ii
Dedication	v
Acknowledgements	vi
Executive Summary	vii
Table of Contents	viii
List of Figures	xi
List of Tables	xii
Chapter 1	1
Introduction.....	1
1.1. Background	2
1.2. Motivations and Challenges	2
1.3. Goals and Objectives.....	2
1.4. Literature Review/Existing Solutions	2
1.5. Gap Analysis	3
1.6. Proposed Solution	3
1.7. Project Plan	3
1.7.1. Work Breakdown Structure	4
1.7.2. Roles & Responsibility Matrix	4
1.7.3. Gantt Chart	5
1.8. Report Outline	5
1.9. Empathy Map	6
Chapter 2.....	7
Software Requirement Specifications	7
2.1. Introduction	8
2.1.1. Purpose	8
2.1.2. Document Conventions	8
2.1.3. Intended Audience and Reading Suggestions	8
2.1.4. Product Scope	9
2.1.5. References	10
2.2. Overall Description	10
2.2.1. Product Perspective	10
2.2.1. Product Functions	11
2.2.2. User Classes and Characteristics	11
3.1.1. Operating Environment	12
3.1.2. Design and Implementation Constraints.....	12
3.1.3. User Documentation	12
3.1.4. Assumptions and Dependencies	13
3.2. External Interface Requirements	13
3.2.1. User Interfaces	13
3.2.2. Hardware Interfaces.....	14
3.2.3. Software Interfaces	14
3.2.4. Communications Interfaces	15
3.3. System Features.....	15
3.3.1. System Feature 1	16

3.3.1.1.	Description and Priority	16
3.3.1.2.	Stimulus/Response Sequences	16
3.3.1.3.	Functional Requirements	16
3.3.2.	System Feature 2	17
3.3.2.1.	Description and Priority	17
3.3.2.2.	Stimulus/Response Sequences	17
3.3.2.3.	Functional Requirements	17
3.3.3.	System Feature 3 (and so on)	18
3.4.	Nonfunctional Requirements.....	18
3.4.1.	Performance Requirements.....	18
3.4.2.	Safety Requirements.....	18
3.4.3.	Security Requirements.....	18
3.4.4.	Software Quality Attributes.....	19
3.4.5.	Business Rules.....	19
3.5.	Other Requirements.....	19
Chapter 3.....		21
Use Case Analysis.....		21
3.1.	Use Case Model	23
3.2.	Use Cases Description.....	27
Chapter 4.....		30
System Design		30
4.1.	Architecture Diagram	31
4.2.	Domain Model.....	32
4.3.	Entity Relationship Diagram.....	33
4.4.	Class Diagram	34
4.5.	Sequence / Collaboration Diagram	35
4.6.	Activity Diagram.....	36
4.7.	State Transition Diagram	38
4.8.	Component Diagram	39
4.9.	Deployment Diagram	40
4.10.	Data Flow diagram	41
Chapter 5.....		43
Implementation		43
5.1.	Components, Libraries, Web Services and stubs	44
5.2.	Deployment Environment	44
5.1.	Tools and Techniques.....	44
Back End:.....		45
5.2.	Best Practices / Coding Standards.....	45
5.1.	Version Control	45
Chapter 6.....		46
Testing and Evaluation		46
6.1.	Use Case Testing.....	47
6.2.	Data flow testing	47
6.3.	Performance testing.....	48
6.4.	Stress Testing	48
Chapter 7.....		49

Summary, Conclusion and Future Enhancements	49
7.1. Project Summary	50
7.2. Achievements and Improvements	50
7.3. Critical Review.....	52
7.4. Lessons Learnt.....	52
7.5. Future Enhancements/Recommendations	53
Appendices.....	54
Reference and Bibliography	55
Index	57

List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

Chapter 1

Introduction

Chapter 1: Introduction

This chapter includes all the basic information of project that we are going to develop. This includes the complete information of project feasibility statement problem statement Better enhancement comparative of this application in market new features to be included. What the user got benefits from our platform scope of this project. Goals and objective Gantt chart analysis project plan work breakdown challenging faces.

1.1. Background

An online movie website is the platform where user can comprehensive resource for information related to movies, TV shows, actors, and other entertainment apart from that a user can also get the movie recommendations. A normal user visits the website, login first and search any movie. At the same time user can also give rating to the specific content and give post content analysis in the review section.

1.2. Motivations and Challenges

When I was young I used to watch movies and series with great interest but in the same time I realize that the good quality content is either paid or hard to find. So I want to build a website on which user can search movie or series and can get recommendations, resources, reviews and some d/f prospective.

1.3. Goals and Objectives

Our major goal is to provide a platform on which user can search any movie and series they want to get downloading resources and some prospective and give them the best recommendation system experience without any flaws.

1.4. Literature Review/Existing Solutions

Right now, most online platform like Imdb or Netflix and amazon which is also include in this prospective are either paid or less effective. User just has to register and login to account and can access to any type of content with good quality and they can also download content from third party website if they want. User can also enjoy the hybrid and recommendation

experience on our website unlike other platforms which only have availability of movies and series with poor quality and no recommendation system experience, which end up having the result of poor user experience.

1.5. Gap Analysis

Whenever user want to search online movies and series for a comprehensive resource for information and get recommendations they don't have any free platform to access the content. Moreover, if they find any platform which is free of cost which don't step into meeting user needs or enhanced user experience or recommendation system or continuous improvement plan.

1.6. Proposed Solution

We have provided the platform where a user can search online movies and series for a comprehensive resource for information and get recommendations. And on the second hand we are giving not only a traditional recommendation system but the advanced machine learning recommender system.

We are also providing the different user experience qualities like giving comprehensive feedback and free of cost user experience to all sectors.

1.7. Project Plan

The work breakdown structure (WBS) and Gantt Chart are listed below according to the activities we are going to perform with relevant dates and responsibilities assigned to the team members. All the activities are specifically assigned by the Team Leader with respect to their relative abilities.

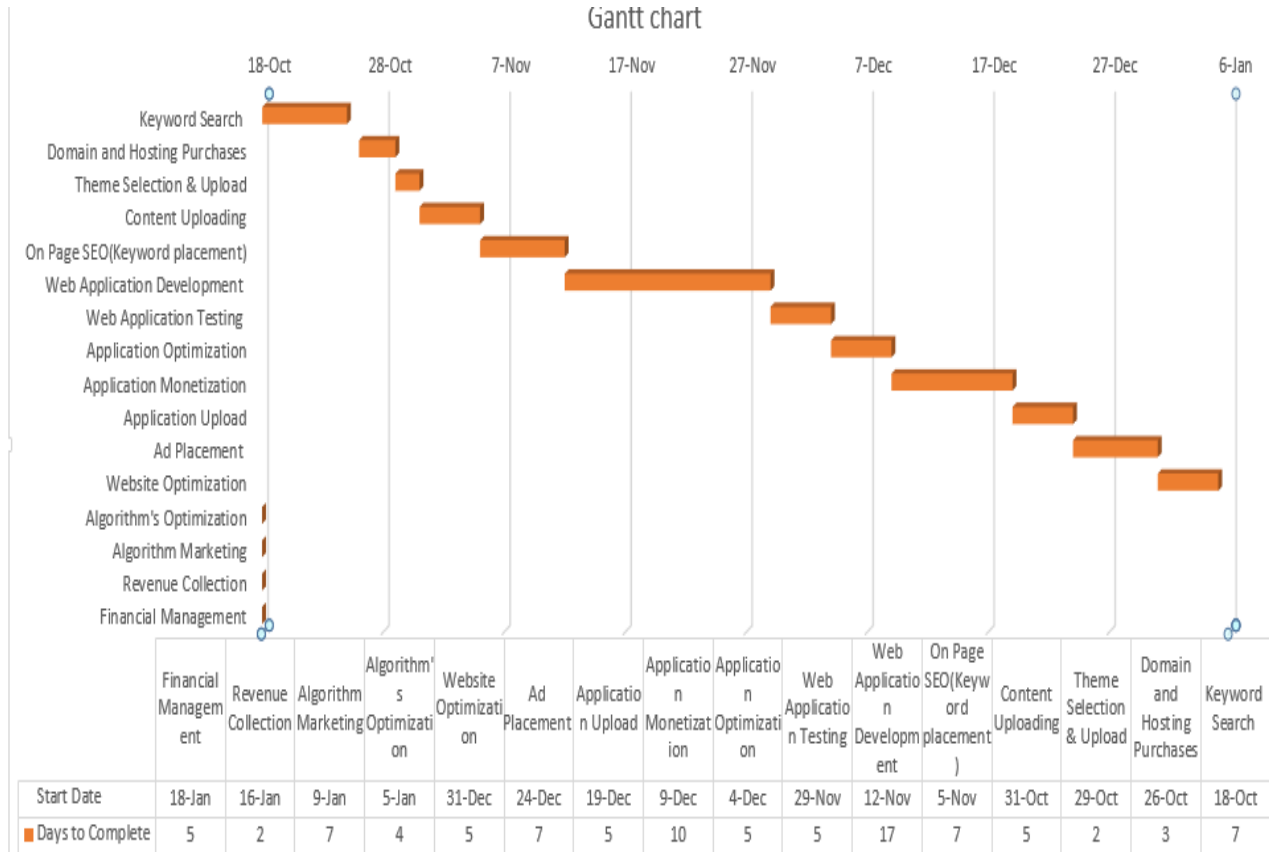
1.7.1. Work Breakdown Structure

Tasks	Start Date	Days to Complete	Roles and Responsibilities
Keyword Search	18-Oct	7	Usama Iftikhar
Domain and Hosting Purchases	26-Oct	3	Atif Mushtaq
Theme Selection & Upload	29-Oct	2	Bilal Shahid
Content Uploading	31-Oct	5	Atif Mushtaq
On Page SEO(Keyword placement)	5-Nov	7	Usama Iftikhar
Web Application Development	12-Nov	17	Bilal Shahid,Atif Mushtaq,Usama
Web Application Testing	29-Nov	5	Bilal Shahid
Application Optimization	4-Dec	5	Usama Iftikhar
Application Monetization	9-Dec	10	Usama Iftikhar
Application Upload	19-Dec	5	Atif Mushtaq
Ad Placement	24-Dec	7	Usama Iftikhar
Website Optimization	31-Dec	5	Bilal Shahid
Algorithm's Optimization	5-Jan	4	Bilal Shahid
Algorithm Marketing	9-Jan	7	Atif Mushtaq
Revenue Collection	16-Jan	2	Usama Iftikhar
Financial Management	18-Jan	5	Bilal Shahid,Atif Mushtaq,Usama

1.7.2. Roles & Responsibility Matrix

Project Deliverables	CEO/CTO(Bilal)	BDM/MM(Atif)	FH/Designeer(Usama)
Wesbite Technicalities+Supervision	Accountable	Consult	Informed
Marketing	Consult	Informed	Accountable
Finances	Informed	Consult	Accountable
UI/UX +Ad-placement	Accountable	Consult	Informed
Buisness development	Accountable	Informed	Consult
Global Reachout	Consult	Accountable	Informed

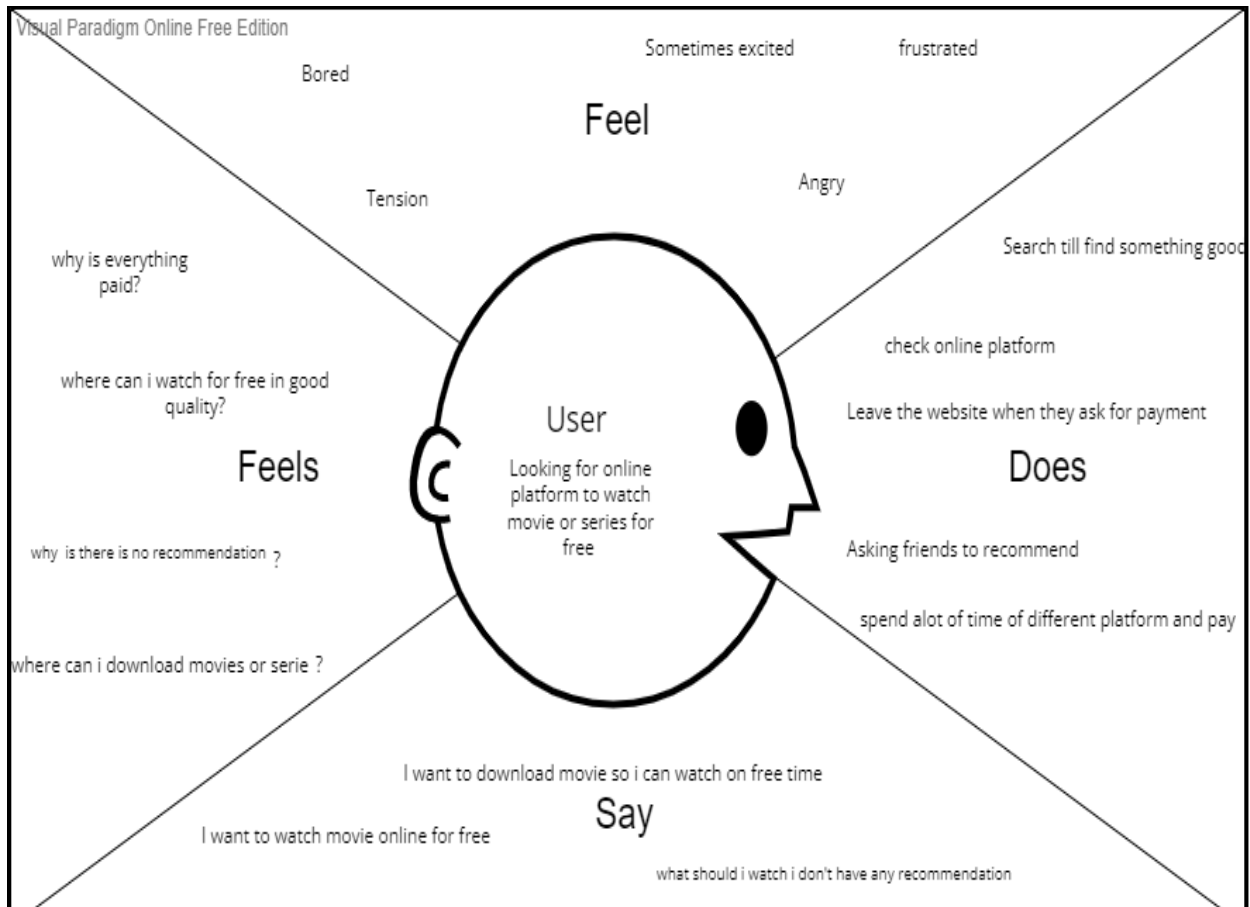
1.7.3. Gantt Chart



1.8. Report Outline

MRS is a website on which user can search online movies and series for a comprehensive resource for information and get recommendations experience for free.

1.9. Empathy Map



Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

The purpose of this software requirement specification document is to provide a detailed description of the functional and working of the website. This document will cover each and every feature of the application including movies/series creating, uploading, navigation and all the interfaces. This document will cover all intended feature and technical dependencies.

2.1.2. Document Conventions

Standard rules of documentation have been followed in order to standardize the work. This software requirement document uses the following conventions.

Description	Appearance
Website	MRS
Server	firebase Server
Database	DB Sqlite3
SRS	Software Requirements Specification

2.1.3. Intended Audience and Reading Suggestions

This document is intended for developers, project managers and users of this website who wish to read about what the website can do. Rest of SRS contains work flow of the system, how the system works, flow of data at the hosting and other technical functionalities of the software.

- For developers there is a need to start from introduction and understand workflow diagrams.
- The project manager sequence of reading is from introduction to class diagram and use cases.

- For users of a website there is a need to read from introduction and read all descriptions including scope and functionalities.

2.1.4. Product Scope

This website will be used as a platform where user want to search online movies and series for a comprehensive resource for information and get recommendations

The website will be globally used.

Module of this website includes all the navigation menu, a simple interface that covers the whole user experience for the users.

For getting started, we need to get some resources to take our first step. Those resources are listed below

1. Domain
2. Hosting
3. License and Copyrights
4. AdSense Approval & Monetization
5. Proper marketing Strategy (We've made our own)

There are steps which will ensure our success and proper structure that we work towards consists of some things that are important for us to move forward with respect to revenue and traffic gathering.

Step 1: We need to have a proper website that meets AdSense policies to get it approved for revenue generation (It takes 10 days to approve a domain)

Step 2: We need to publish the resources which we have backed up with our test running we did.

Step 3: We need to have proper navigation for the stuff/resources we upload at our server.

Step 4: We've to do proper keyword search with competitor analysis for our categories and the content we want to rank at Search Engines. – That's where Search Engine Optimization Kicks in

Step 5: We'll do complete optimization of our website and application and We'll develop a environment that makes our website

- Crawlable

- Indexable
- Rank able

Step 6: We will develop landing pages with Google sites for each category that'll convert our social traffic into organic traffic – It affects the CPC that is the major component of your revenue generation.

Step7: We'll do algorithm marketing (developed by our team) to get traffic to our website and relevant people who want to watch movies/series. This will help us with some points let me list them for you

- Ranking at Google at relative keyword which will increase because high traffic sources + google sites links will increase our DA – Domain Authority
- People who wants to watch movies/series (our customers)
- Revenue

2.1.5. References

Article Title: Best AdSense Approval Practices

Website Title: [Google Blog](https://www.blog.google/)

URL: <https://www.blog.google/>

2.2. Overall Description

2.2.1. Product Perspective

The website is designed and developed to provide the best services to the users who surf the website search movies/series for free on the specific category and get recommendations and get resources on various factors. A user can use the navigation menu to go to a specific category according to his/her choice. The user's analytics and all the other factors will be managed in the admin panel. The uploading of content, optimization and Ad-placement will be done at admin or developer end.

This website is the replacement of all the websites that provide movies content at specific cost. All these services will be provided on a single platform to facilitate its users. We've recommender system.

2.2.1. Product Functions

The project function depended on three types of users that impacted on the website depending on their skills and needs. All the users are listed below.

General User.

1. Provides resourceful and filtered Content.
2. Allows to comment at any content to give review.
3. Chatbot to contact and get your queries resolved.

Administrator.

1. Admin can upload the content.
2. Setup Navigation

Developer.

1. Website Structure Setup.
2. Optimization of website
3. Ad-Placement.
4. Linkage (Internal/External).

2.2.2. User Classes and Characteristics

3. Classes	Characteristics
Admin	Audit all the activities using analytics with respect to user behavior, uploading of latest content and website performance.
User	Users will go through a certain content to search or comment the content. User can also share the resource at social media or access the contact button or chatbot for counselling.
Developer	Responsible for optimization, Inbound & outbound linking

Classes are made for describing the relation between user characteristics and others with operation functions. User classes help in this project to relate or use the parent functionality and perform more operations. There are following types of users on the website.

3.1.1. Operating Environment

1. The website can be accessed by using any type of device.
2. Its design is responsive with every screen. It's dimensions and its content are responsive along every device.
3. In addition, for the user who visits the website from Google search.
4. No user's personal information will be collected apart from its website surfing behavior and content analysis/comments.
5. User can check the websites navigation using menu.

3.1.2. Design and Implementation Constraints

1. Managerial: As our main revenue stream is AdSense, Popup ads, Banner ads and direct ads. So, we've to manage the invalid activities like invalid clicking and invalid traffic. For that we've to manage the scripts that help us combat the invalid traffic. No third-party technologies or plugins are present that we can implement at our website through third party APIs that we can implement.

2. Resources: The courses and other learning resources that we present on our website can cause an issue due to low availability.

3. Performance: REST API in a our project provides a flexible and standardized way to expose your application's functionality and data to clients, making it easier to build scalable and interconnected systems.

3.1.3. User Documentation

A complete documentation guide will be present in the website name as 'website documentation' Moreover, if a user wants to learn about website usage. There'll be a manual attached in the navigation menu which is a perfect navigation practice recommended by Google. These are user documentation lists. As follow:

- Minimum browser requirements
- Search guide.
- How to get to a specific content?
- How to use different features of the application.
- Example inputs and outputs.
- Explanations of error messages and troubleshooting guides.
- Information to contact the developer of the system if an undocumented question arises.

The quality of user documentation can affect the rate of implementation of updated applications. Examples should include methods such as: help content & online support.

3.1.4. Assumptions and Dependencies

All the dependencies and assumptions that this project will have are listed below.

1. Google AdSense Policies for Ad-Placement at units present in the website.
2. The third-party plugins that we use can cause performance issues related to the website's design.
3. The website will be customized and manual designed according to the CMS uploading and streaming.
4. The root and sub folders for website code will be done according to CMS's policies so we can have a dynamic use.
5. Hosting server's bandwidth will be a challenge but it will be analyzed and combated between stress testing of websites.
6. As the major source of our traffic is Search Engine. I.e. Google. As per experience and previous stats it changes its algorithm with notifying. So, we've to follow the KGR (**Keyword Golden Ratio**) structure for website's search engine optimization.

3.2. External Interface Requirements

3.2.1. User Interfaces

The website is developed for making the user navigation through pages and posts as simple as possible. The font family which we used in our website is **Poppins**. Geometric sans serif typefaces have been a popular design tool ever since these actors took to the world's stage. The

UI of the website also contains special ad-units for displaying advertisements. Let me mention some of the units in the list that represent advertisements.

1. Header
2. Footer
3. In Feed
4. In-Article
5. Sidebar

The user will have a navigation menu consisting of categories and search bar. Through which they can navigate through websites and get to their desired course.

Responsiveness: The website is responsive across all the devices and screens that are manufactured across the world. I use Google recommended policy guidelines for navigation along with responsiveness recommendations.

3.2.2. Hardware Interfaces

A user can access this website by using the following devices.

- Mobile Phone
- Tablet
- Computer
- Laptop

Any other device which supports browser installation. Users must be connected to the internet to surf the website.

3.2.3. Software Interfaces

Website is an easy to use platform designed and developed for native usage. It is developed in HTML CSS and JavaScript bootstrap for designing interfaces. Design is managed by CSS and bootstrap.

. The components we are using in firebase are:

1. Fire store
2. SQLITE3
3. Fire storage
4. Push Messaging

5. Analytics.

Security: We use Let's Encrypt Authority X3 Secure Sockets Layer (**SSL**) securing the hypertext protocol. That helps us secure the website's domain from different attacks including SQL injection which can cause a very bad effect on website's content and stored databases.

Data management: CDN is used in backend (Django framework) statically for performance management for our website. That includes the following functions:

- Lazy Load
- Cache Management
- Content Loading
- Font Minification
- JavaScript & CSS minification.

3.2.4. Communications Interfaces

The website is hosted on a server that uses a Django embedded Sqlite3 database to host the website's database. It uses a HTTPS connection. There's no use of cookies, the user behavior and other analytics will be controlled and analyzed by a third party application provided by Google. I.e. **Google Analytics**. The connection of the internet is must for using the website otherwise you'll see a cached version and can't avail the proper usage that is promised by team ALPHA.

3.3. System Features

Movie recommender system is a user-friendly website that provides all the features to its users at a single place with. The website's features are as follow:

- Navigation through header menu
- View content.
- Search Content.
- Third Party content.
- Rate content

- Get Recommendation of content
- Contact Us (Email, WhatsApp)

3.3.1. System Feature 1

The system feature is named as the Top most important feature of our website. I.e.

“Recommendation Panel”

3.3.1.1. Description and Priority

The major stakeholder of our project is the user with a high-level priority rated as 10/10. We've the website provides movies and series to search with recommendation system. So, we've to maintain security and make sure it doesn't get affected with any type of malware or ransomware. When a large flow of traffic will be at our domain, we've to manage the backdoors and other accessible pages like the login of our Move website to make sure that no one gets access to the data flow and behavior stats of users surfing at our website in real time as well as old users.

3.3.1.2. Stimulus/Response Sequences

- At first, when the user will come to our website's landing page. The website will load the fonts, images with Lazy Load and already store cache files for faster loading. After that the user will go to a specific movie/series by scrolling to landing page or search bar. If the user uses the search bar the search will be transferred to the hosted server and database will respond with a relative query according to the user's input and the content with that keywords will appear at the user's screen. In case of no content at those keywords the user will get a 404 page with a message of no not found.
- When a user clicks the download button a third party website will be open which will point the user to get an access to either watch or download movie.. All of this process will be done using a custom script made by the developer.

3.3.1.3. Functional Requirements

REQ-SF1-1: Menu Navigation

REQ-SF1-2: Search Content

REQ-SF1-3: Recommend Content

REQ-SF1-4: Rate Content

REQ-SF1-5: Review content

3.3.2. System Feature 2

This system feature describes the developer and admin side that how a developer can develop, optimize and do ad-placement at the website as well as how an admin can audit the website's condition and optimization needs. It's named as **"Website Optimization Panel- CMS"**

3.3.2.1. Description and Priority

As this feature directly links to our revenue generation and our policies which can indirectly get our website penalized. So, it holds a high priority at scale of 9 out 10.

3.3.2.2. Stimulus/Response Sequences

At first, the developer and admin have to login into the CMS. It requires him to have a username and email that is provided by the host manager or admin. When the developer will enter his credentials an automated system of authentication will authenticate the user from the host and in response provide the access to the developer. The core responsibilities of a developer are to check the user behavior from analytics and provide the best solution for optimization with respect to that. In case of forgetting the password, the developer can request to forget password by pressing the forgot password function but the admin needs to reset his/her password through the python in-build **"createsuperser"** function.

3.3.2.3. Functional Requirements

REQ-SF2-1: Set Menu

REQ-SF2-2: Add Content at website.

REQ-SF2-3: Add Pages and Contents.

REQ-SF2-4: Place Ads at Ad units.

REQ-SF2-5: Add Meta title for Movies and Series.

REQ-SF2-6: Add Meta description for pages and posts.

REQ-SF2-7: Delete a Content.

REQ-SF2-8: Update a content.

REQ-SF2-9: This feature will have a forget password button that'll navigate to the forgot password page.

3.3.3. System Feature 3 (and so on)

All the system features are discussed at System feature 1 and 2.

3.4. Nonfunctional Requirements

3.4.1. Performance Requirements

Performance of this site is very efficient and responsive immediately. It takes 1 sec on average in response to the backend server response quickly and finds objects very fast. It loads in just 1.5s that is less than the google recommended loading time. All functions are responsive and take less time than a traditional website because the backend is supported by Django which have in-build Sqlite3 database and is powerful and flexible.

3.4.2. Safety Requirements

Our safety depends on some factors, all of them are listed below with proper explanation.

- **Use of cookies** is the most important data that we need to manage because of this we'll get user analytics and behavioral data about the users.
- **Malware** - The main focus will be preventing malware from getting into our core files that can harm our indexes as well the usage of users. So, we'll use a third party word fence to combat that and manage all our scanning.

3.4.3. Security Requirements

- As such there's no risk of security breach at user end but still. We'll use an SSL certificate. The main purpose of SSL is to protect user data. It reduces the risk of phishing and it also increases your search engine ranking. It's a lock of trust for people visiting your website. It develops trust of users for your website.

- At the admin and developer side we'll use a custom login panel. The URL of the login panel will be dynamic so that it will generate only one key button that will be the property of the host manager or admin. In case some access that we'll use a third-party notifier that will notify us about the login attempts at our CMS's login panel.
- Application control admin should use firebase email authentication for verification of the user login.
- Each stakeholder from the admin side will have a unique username dependent and the role will be specified accordingly by the host manager or admin.

3.4.4. Software Quality Attributes

- The website and application will be available at any browser.
- The website and application will be available using any type of internet connection
- The user interface will be responsive to enhance user experience.
- Stress testing of websites is done and it can manage up to 10000+ users at a time.
- The application and website will run at any device supporting browsers (In case of website).
- Sitemap is generated for each page, category and posts for pointing it into the robots.txt file.

3.4.5. Business Rules

- At the admin panel, he will be handling only to add content update or delete content and same functionalities on users too. Other than that there is no other work or responsibility given to admin.
- At the developer panel, hosting responsibility and algorithm optimization etc.

3.5. Other Requirements

The requirements that are must to fulfill AdSense approval factors are most important for us and I've listed them as follows:

1. Proper Navigation to the pages.
2. Sitemap and Technical Page should be in the menu.
3. Proper indexing should be done.
4. There should be almost 30 posts on the website for approval.

5. All the posts and pages should be SEO optimized and in perfect with respect to readability.
6. There should be a proper structure data for your website's search engine presence.

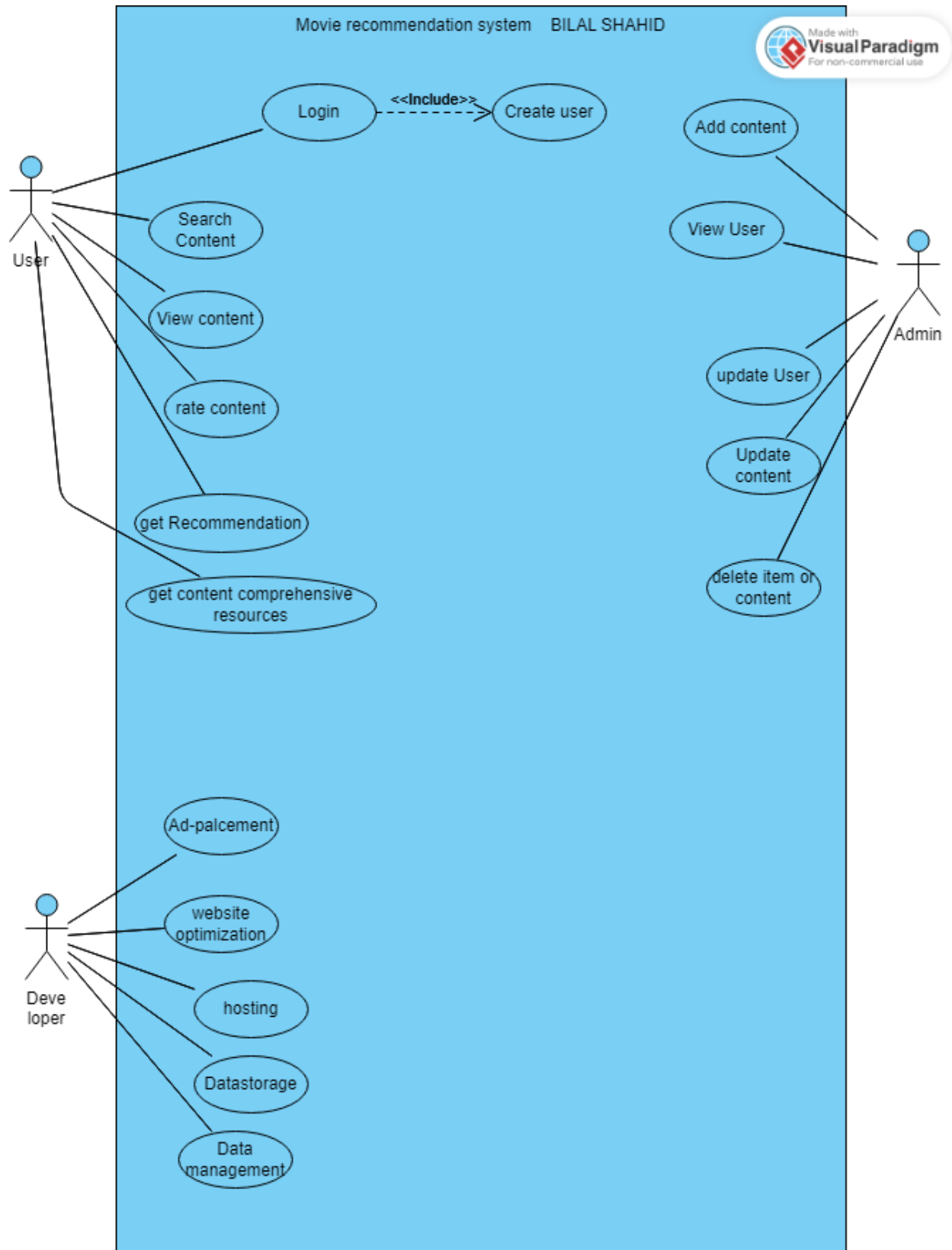
Chapter 3

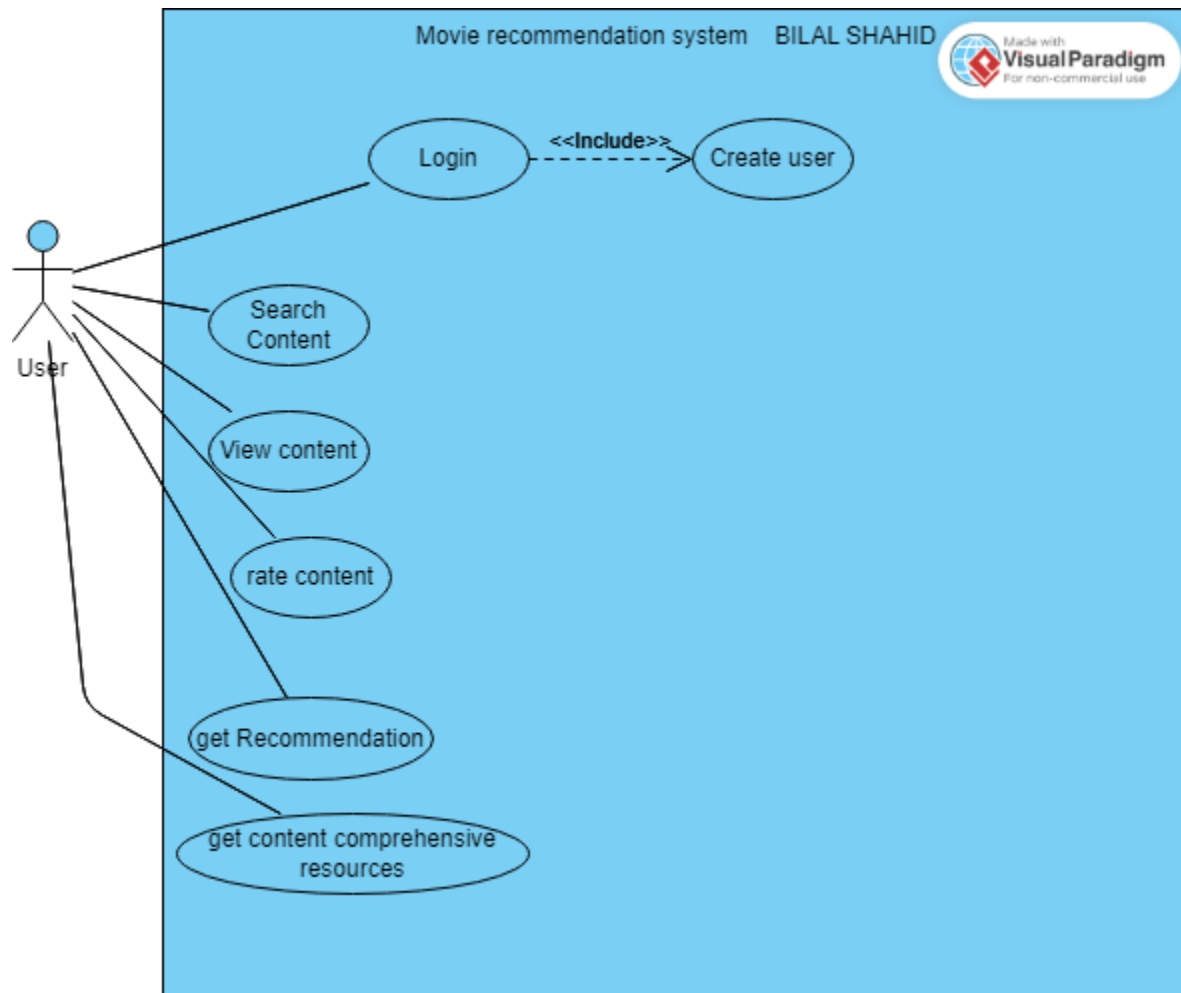
Use Case Analysis

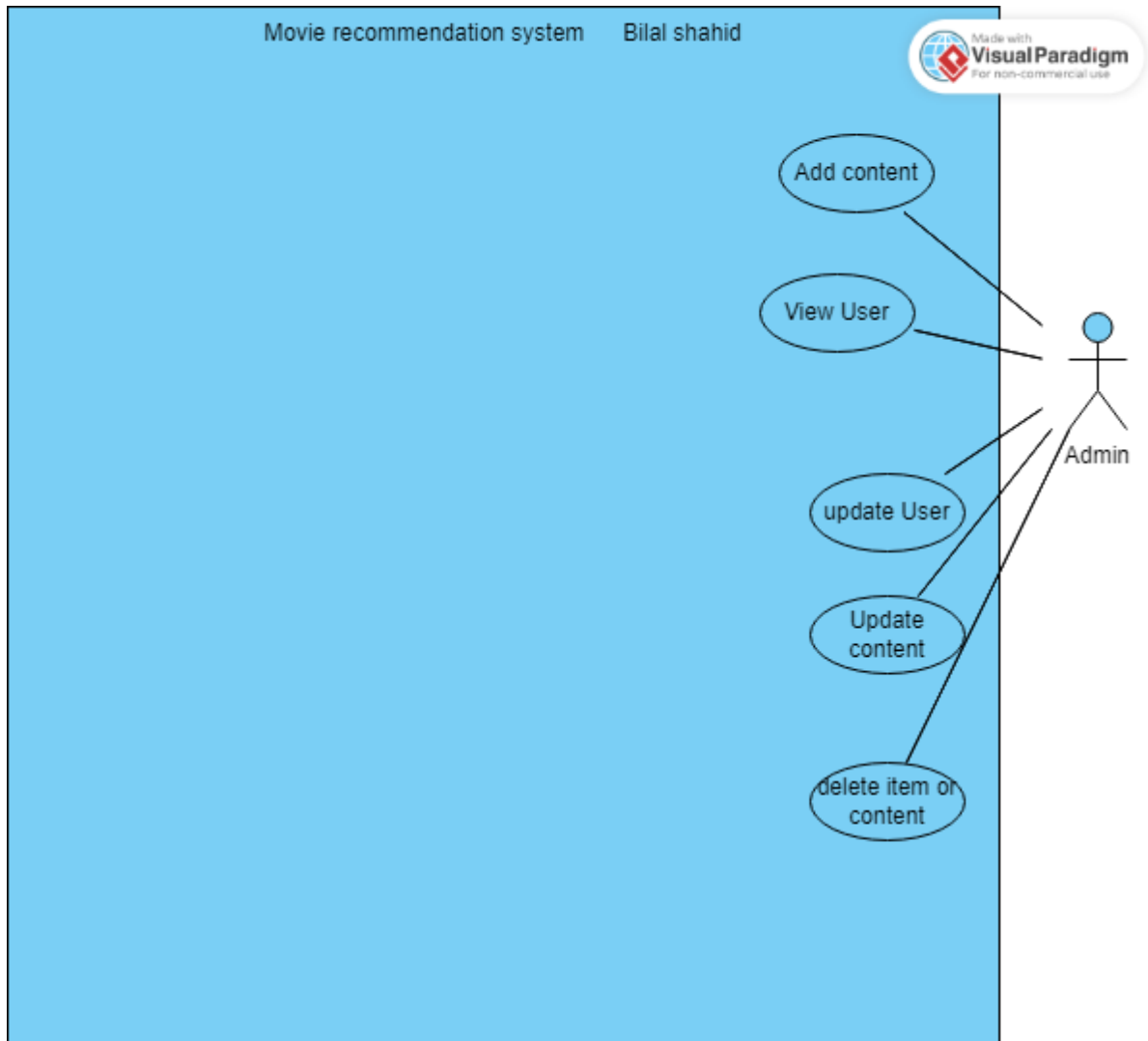
Chapter 3: Use Case Analysis

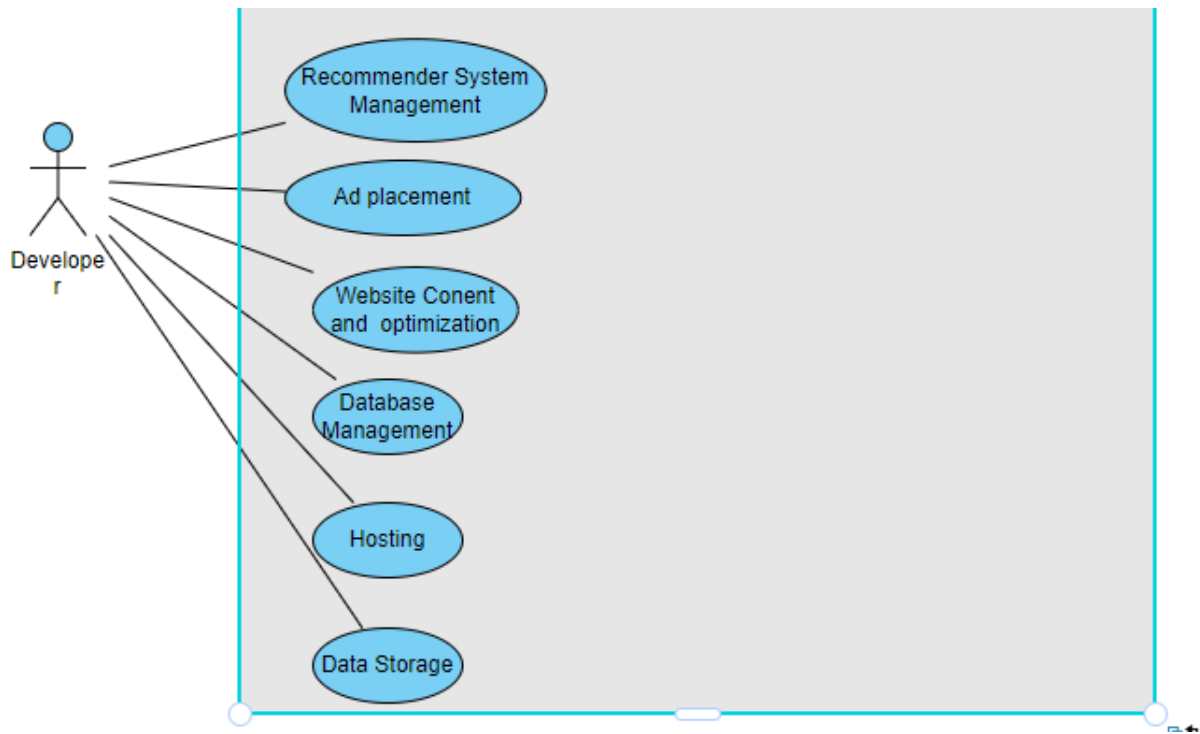
This chapter is all about the roles and functionalities of the system and its actors in the use case form. All the primary and secondary actors with respect to their responsibilities. We have made a use case diagram in which we've properly defined each and every functional and nonfunctional requirement according to the respected id or levels. In the use case given below explains the relation between actors and their functionalities.

3.1. Use Case Model









3.2. Use Cases Description

3. USE CASE NAME	Movie Recommender System
MADE/CREATED BY	Bilal Shahid Usama Iftikhar Atif Mushtaq
MADE/CREATION DATE	4-12-2022
ACTORS	Admin, Developers, Users
DESCRIPTION	The website is designed and developed to provide the best services to the users who surf the website to search movies or series to get comprehensive detail and third party links or rate and give review to the content and get recommendation. A user can use the navigation menu to go to a specific category according to his/her choice. The uploading of content, optimization and Ad-placement will be done at admin or developer end. This website is the replacement of all the websites that provide searching of movies/series to get comprehensive details and get content base recommendations platform to watch movies and series and experience recommendation to specific content. All these services will be provided on a single platform to facilitate its users.
DATA	Data is gathered according to the marketing strategy like information of Facebook groups which are related to learning and tech based. User data at the website is also gathered for analytics purposes.
PRE-CONDITION	Users can visit the website with no login or sign-up just a connection to the internet.
POST-CONDITION	Viewing, Search, Recommend, rating, review the content from our website.
PRIMARY ACTOR	User
SECONDARY ACTOR	Admin, Developer

Use Case Name	Search/Rate/Recommendation
Priority level	5
Primary actor	Users
Pre-Condition	User just needs to visit the website movie recommendation system
Post-Condition	He can then search, get recommendation, rate the content
Main success Scenario	The User just needs to visit our website and can search the content of his/her own choices and also can get recommendation or rate.
Extensions	If he cannot find the content, he/she can contact our customer support and can get help or he/she can use the Chat Bot for his queries.

Use Case Name	View Content
Priority level	2
Primary actor	Customer
Pre-Condition	Needs to visit the website
Post-Condition	They can see the content and all other resources on our website and can also get help from our automated chat bot.
Main success Scenario	Users need to visit the website and can see the content and different resources of his/her own choices and can also contact our 24/7 customer support for further information.
Extensions	Successfully got the proper information if not or not able to reach there can be a problem with his network or if any other issue can be resolved by our customer support team.

Use Case Name	Developing
Priority level	4
Primary actor	Developer
Pre-Condition	Needs full website access
Post-Condition	Handles the dull website designing and developing handling databases, ad placements, and all other
Main success Scenario	Can log in to the website and can make necessary changes if they are critical and solve all technical issues if they get in, place ads, hosting renewal and all developing related work
Extensions	Successfully resolving and managing the website if facing some problems can ask the admin before taking any critical step

Use Case Name	Administration
Priority level	4
Primary actor	Admin
Pre-Condition	Whole control on the website with full access of website and hosting
Post-Condition	Manages all the operations and directs other members to do certain tasks. Perform check-ins on the website, always looking for getting more traffic and growth of the website, full control over all the resources, perform all kinds of crude operations related to resources.
Main success Scenario	Manages all the work related to our website and its resources successfully
Extensions	All things done accurately if face some issues can the meetings and consult with members.

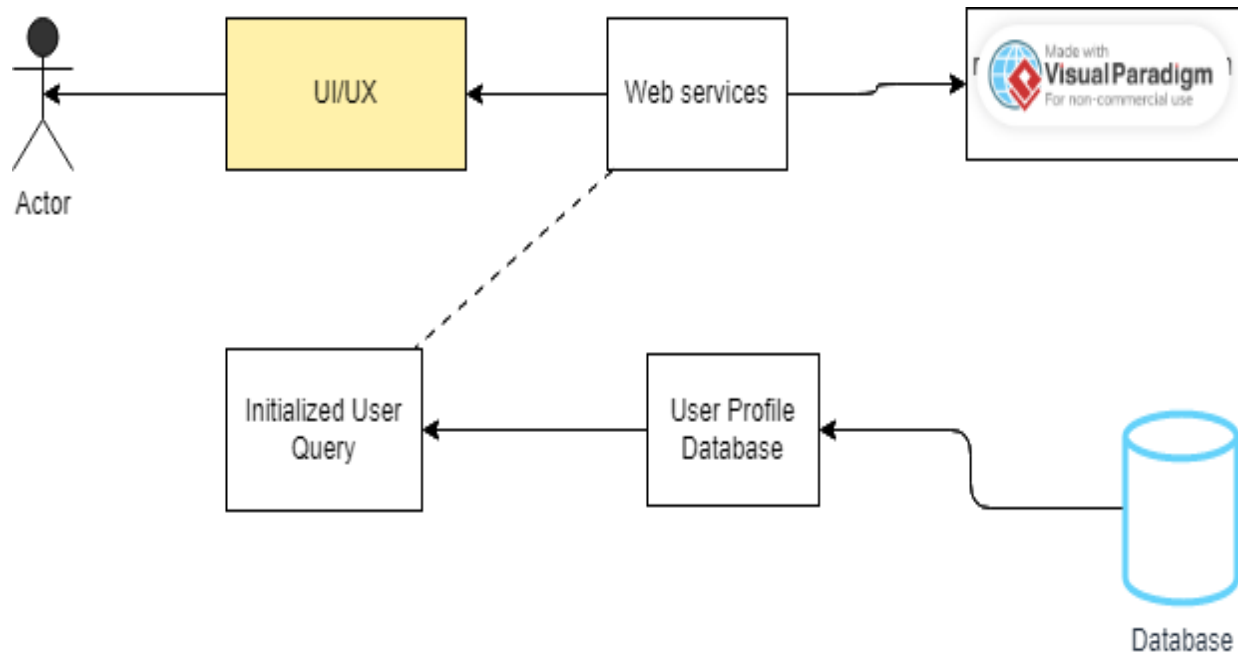
Chapter 4

System Design

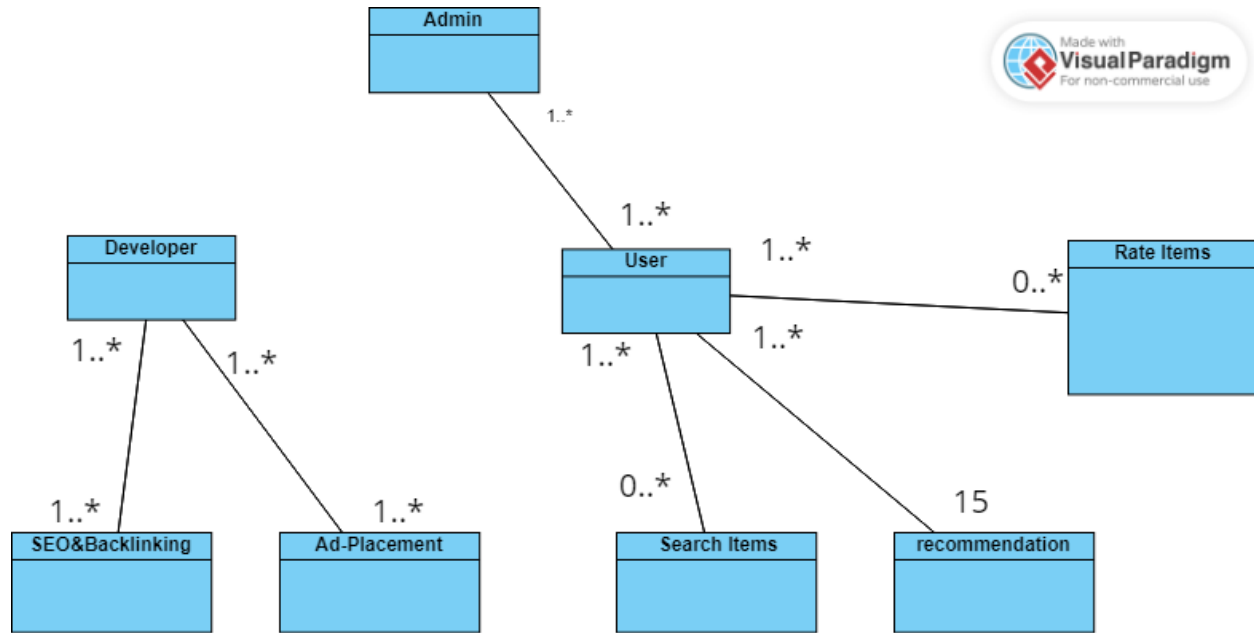
Chapter 4: System Design

In this chapter all the diagrams are given so you can clearly check out the overall working graphically.

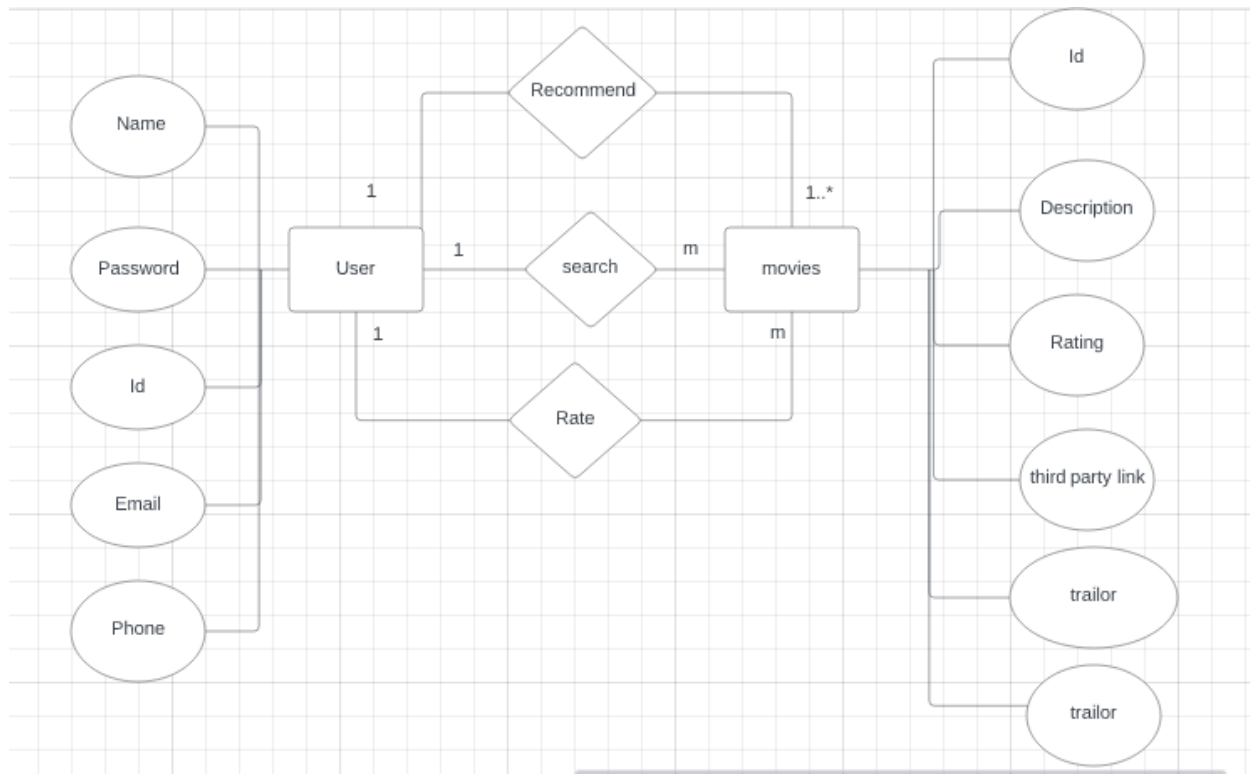
4.1. Architecture Diagram



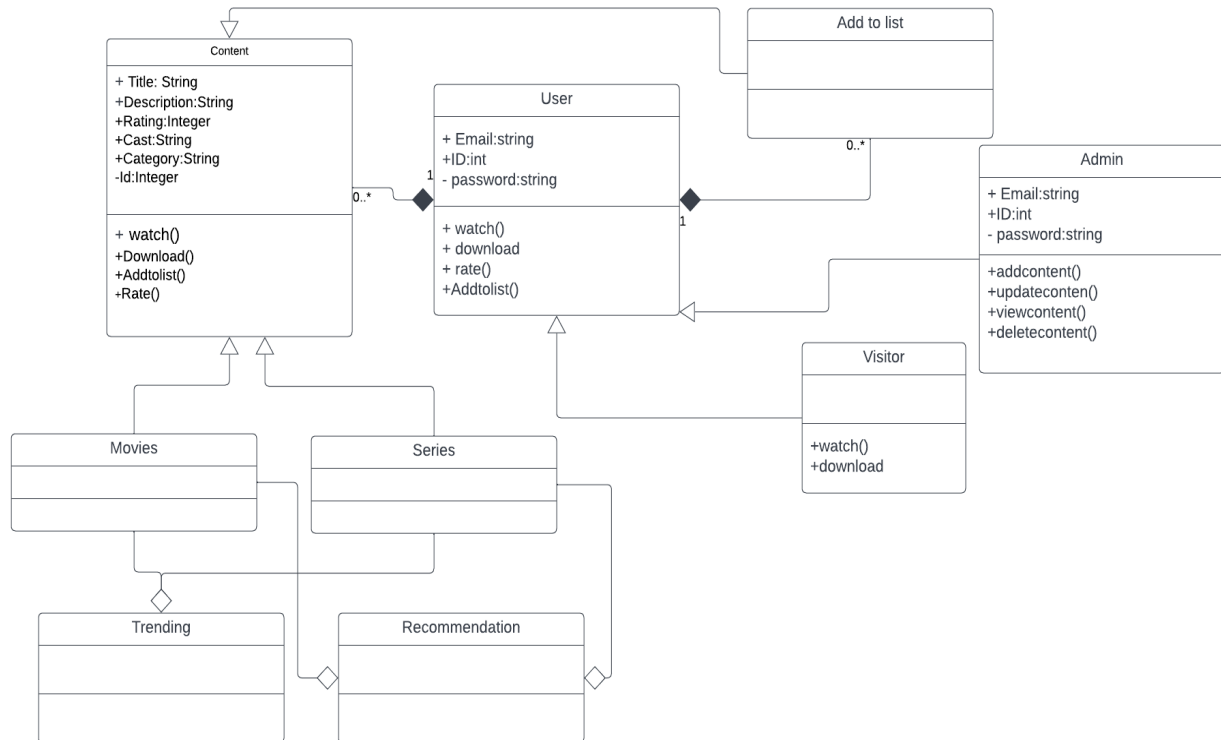
4.2. Domain Model



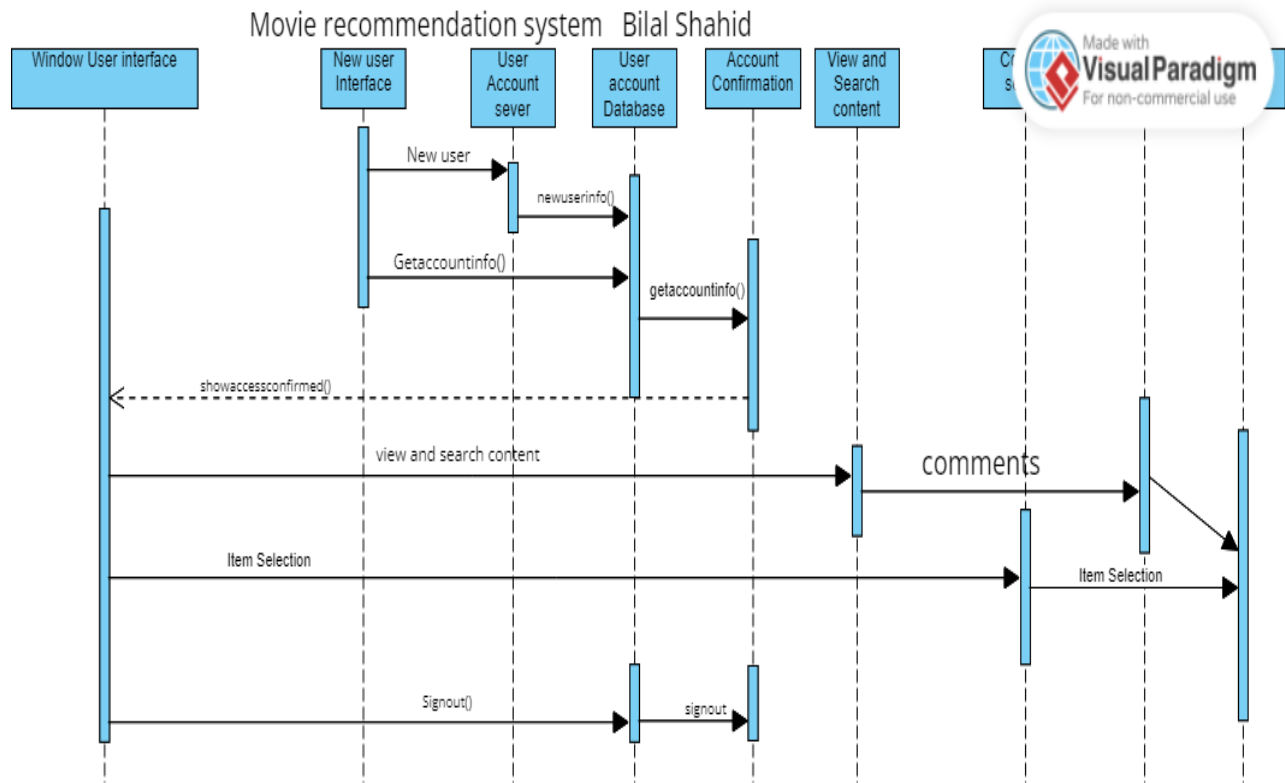
4.3. Entity Relationship Diagram



4.4. Class Diagram

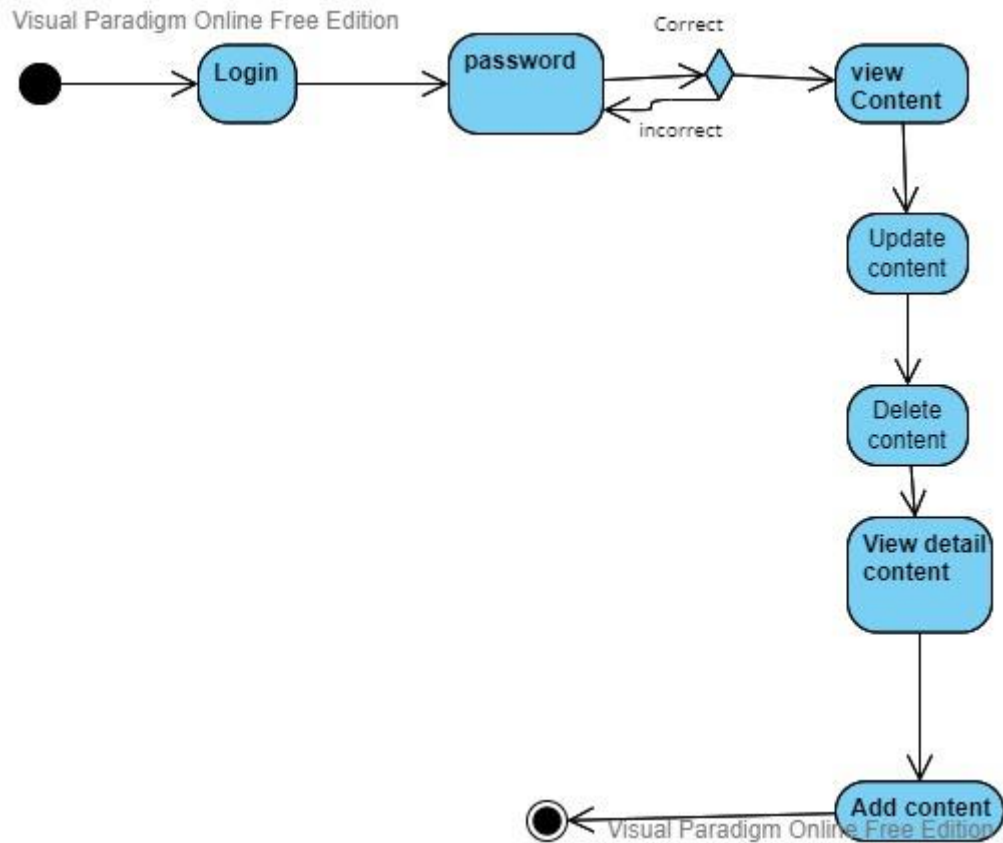


4.5. Sequence / Collaboration Diagram

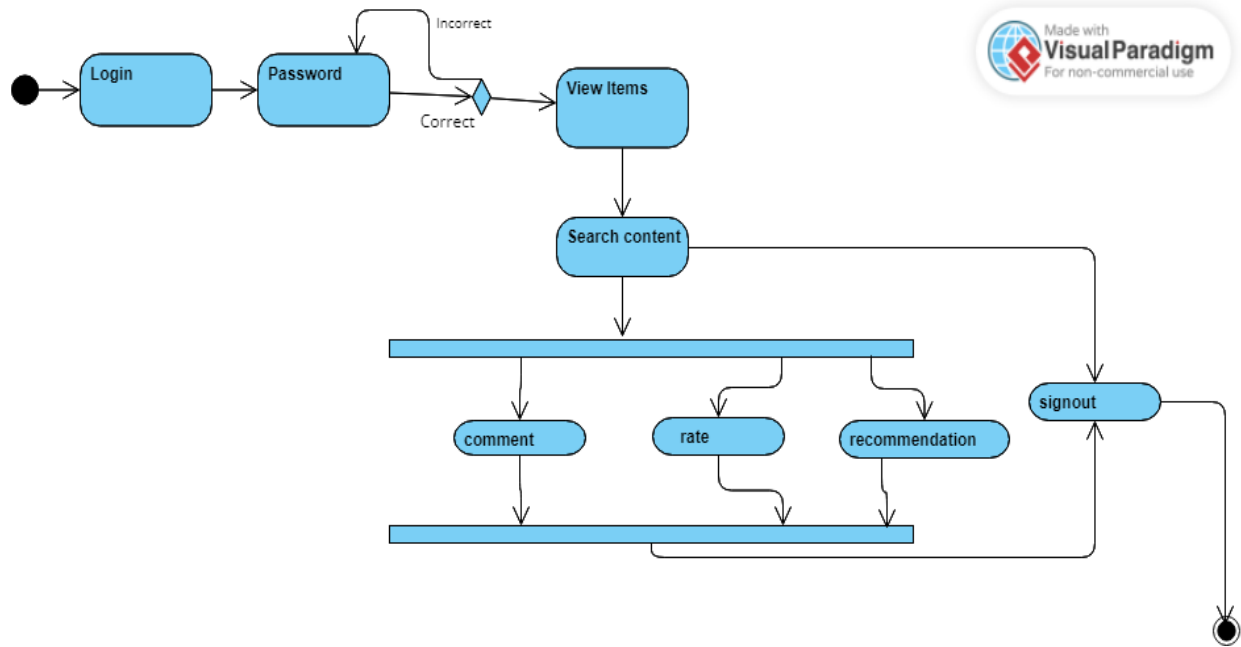


4.6. Activity Diagram

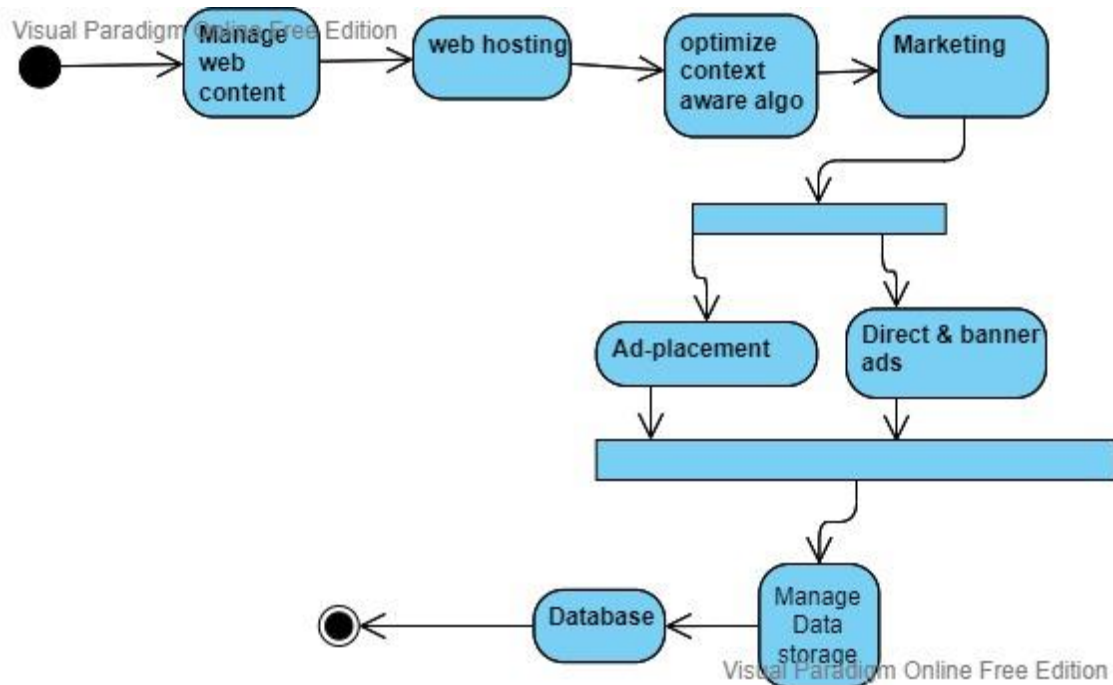
Admin activity diagram



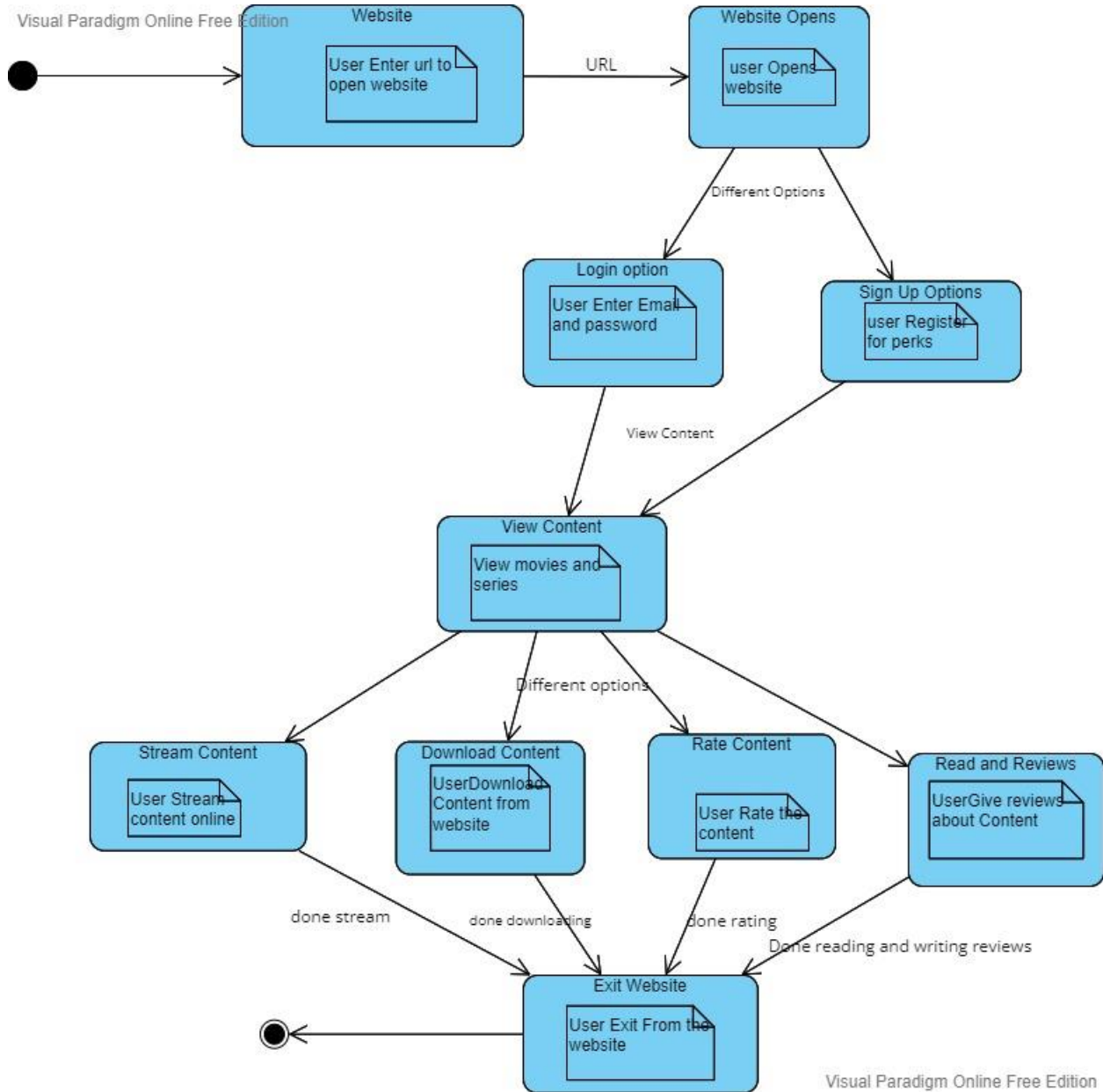
User activity Diagram



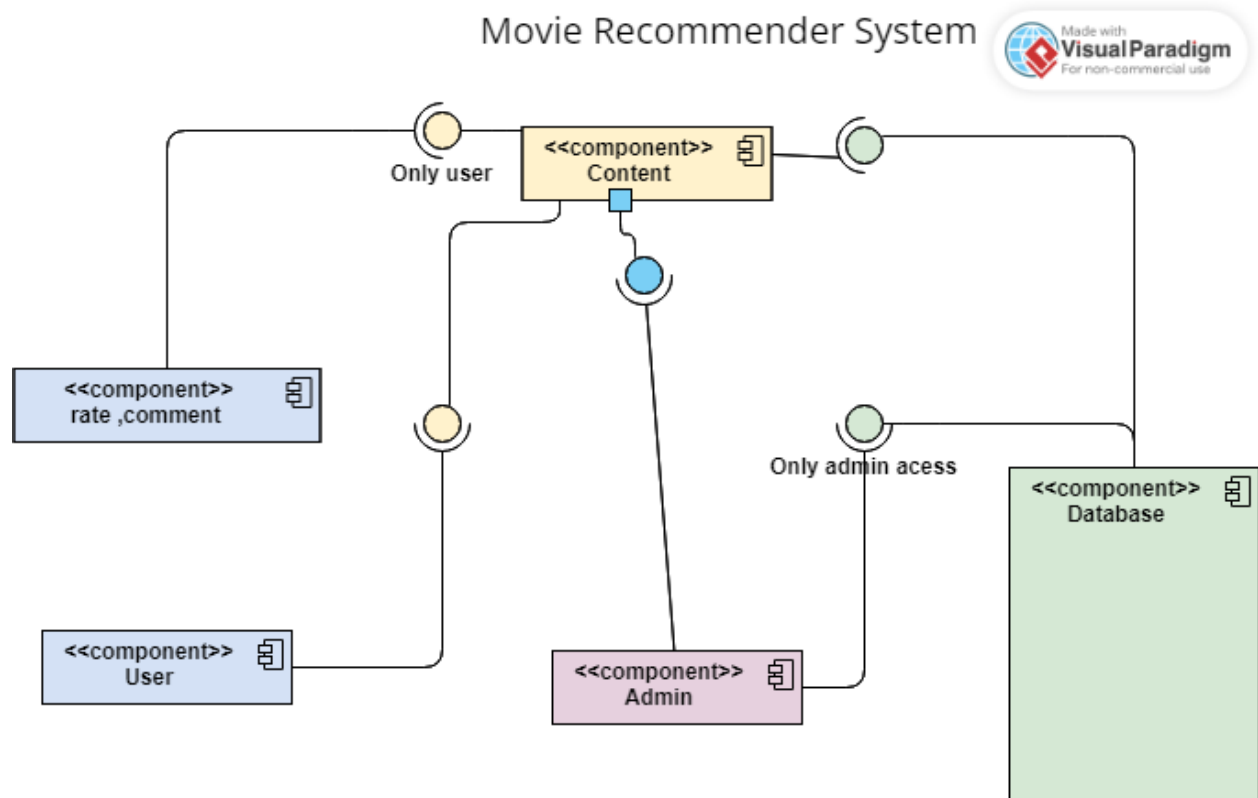
Host Manager Developer Activity Diagram



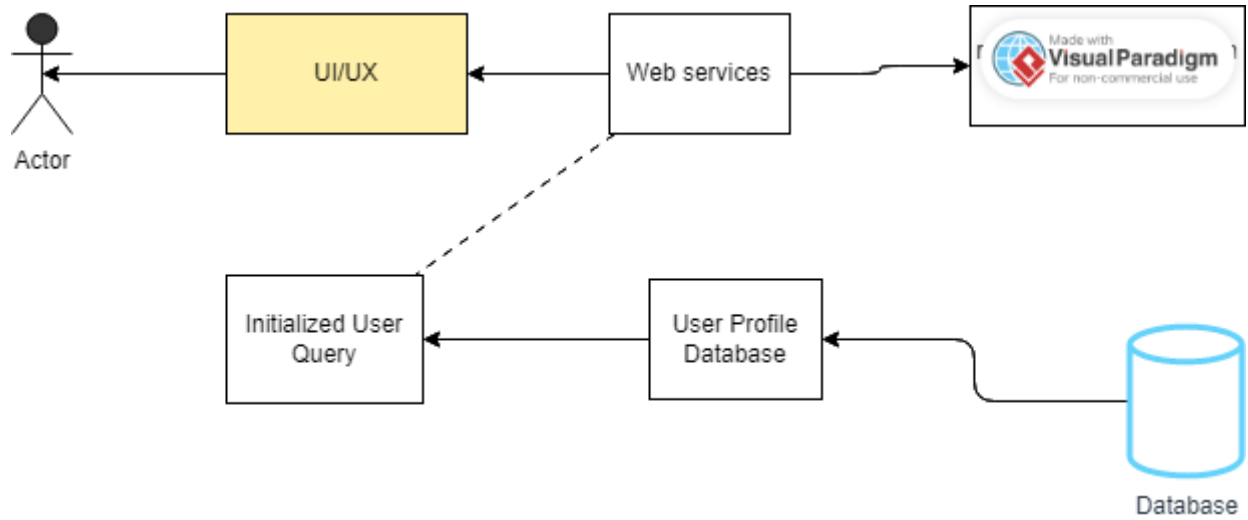
4.7. State Transition Diagram



4.8. Component Diagram

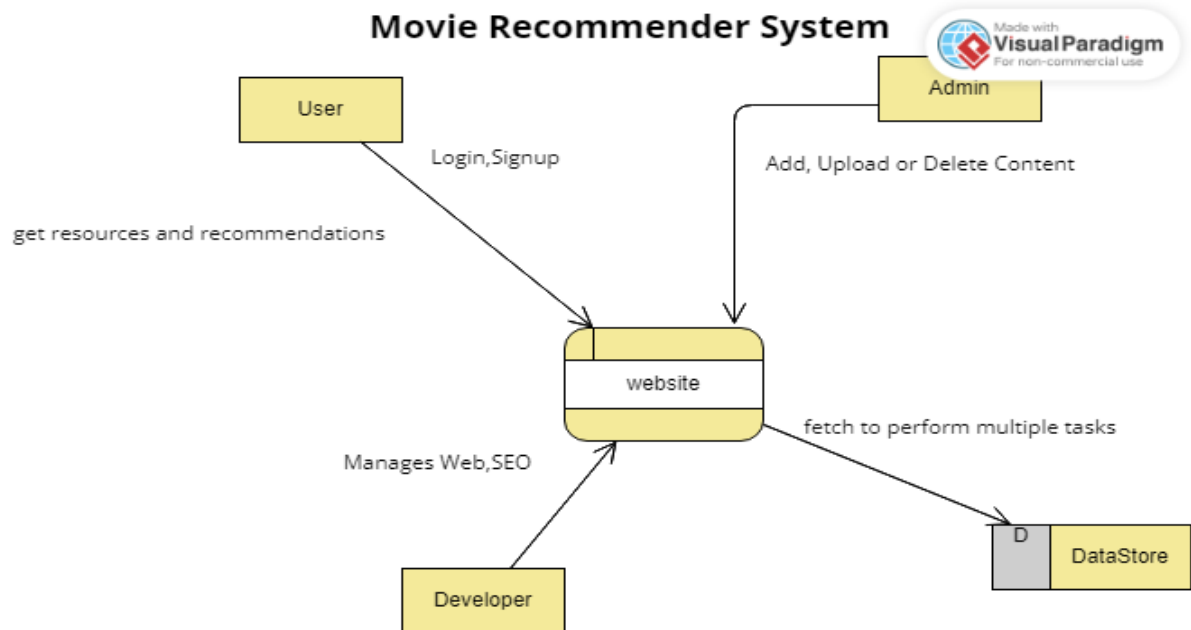


4.9. Deployment Diagram

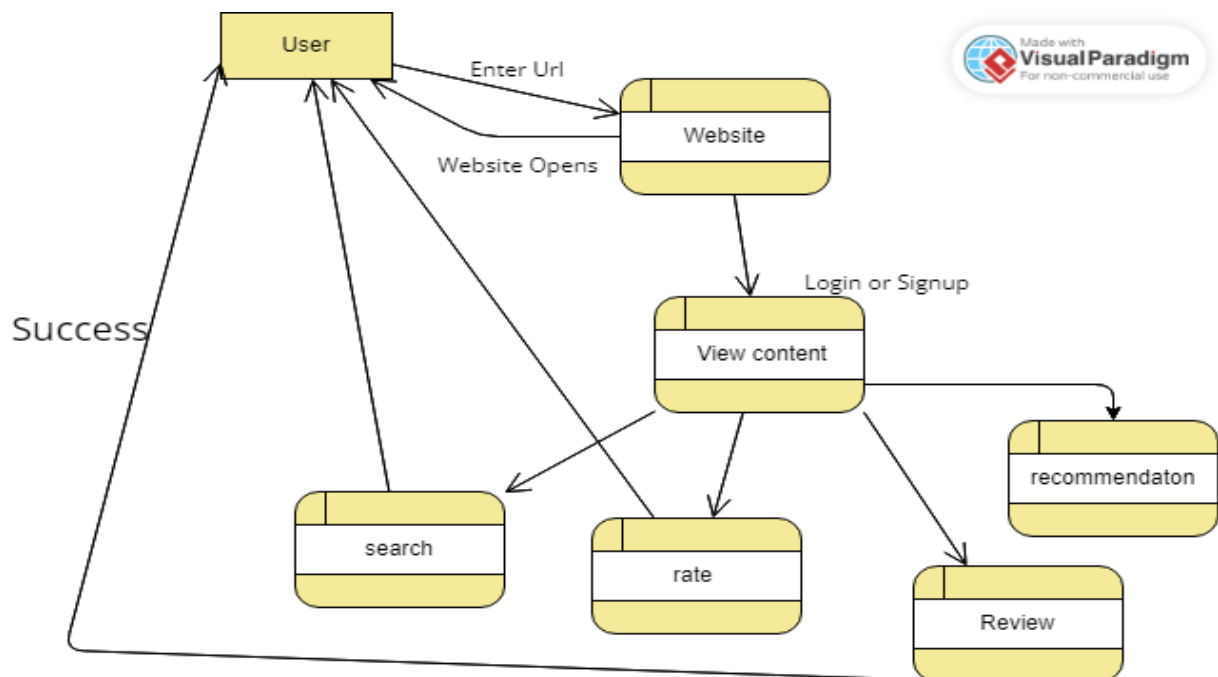


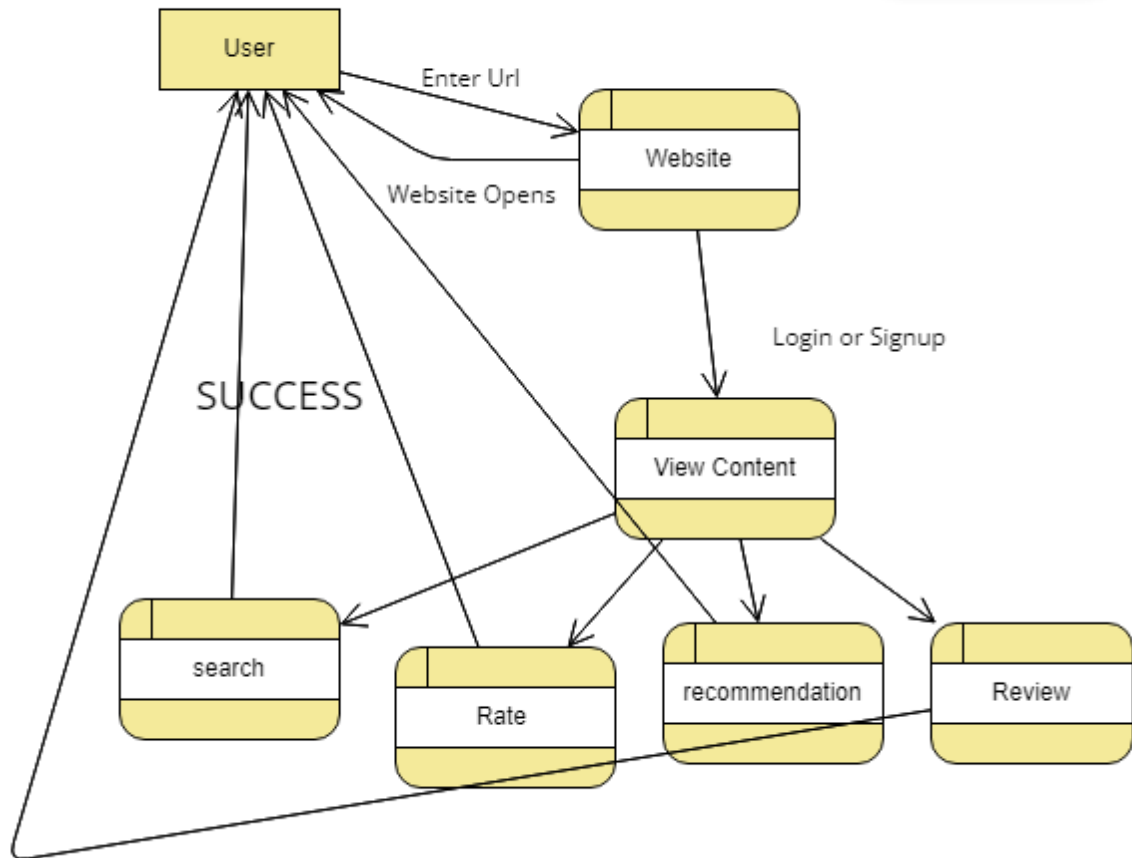
4.10. Data Flow diagram

Level 0:



Level 1:



Level 2:**movie Recommender System**

Chapter 5

Implementation

Chapter 5: Implementation

This document describes the project implementation for developing our project Movie website. The project implements, HTML, CSS, JavaScript and Bootstrap with backend of python framework Django which powered by Sqlite3. The project will be capable of running on phones, laptops and devices. The interface for the project will provide a single platform where you can get streaming content for free that are paid across the internet.

5.1. Components, Libraries, Web Services and stubs

In the recommendation system there are libraries like wheel pandas Gunicorn White noise fastparquet pyarrow and in backend of the website we used rest framework in Django.

5.2. Deployment Environment

4. We deployed this project in Windows 10 and We've connected an Open-Source Dashboard for setting up our website's theme.

5.1. Tools and Techniques

The tools that are required are Listed below

Development Tools

1. VScode
2. Sqlite3
3. Firebase

Auditing and Analytics Tools

4. SEOToolKit
5. Moz
6. SEMRUSH
7. AdSense Dashboard
8. Google Analytics
9. Search Console

Front End:

For our website we've used comprehensive languages like HTML5, CSS, JavaScript, Bootstrap and Python for Little alteration in third party plugins. With increasing end user expectations in relation to the quality, expressiveness and performance of web-based applications, we have given more attention to satisfy the customer. Major Ad-placement is done using JavaScript that is responsive for every device as well.

Back End:

For our website We've used Shared hosting with static CDN. We've made our database with Sqlite3 that is secured with Let's Encrypt X3. The CDN serves the purpose of loading media files from the closest data network.

5.2. Best Practices / Coding Standards

5. Defined Life Cycle and Milestones
6. Stable Requirements and Scope
7. Defined Organization, Systems, and Roles
8. Quality Assurance
9. Planned Commitments
10. Scope and mission
11. Search Engine Optimization – On-Page
12. Search Engine Optimization- Off-Page
13. Ad-Placement
14. Structure Data Setup

5.1. Version Control

We have this complete documentation that completely represents all the factors that are related to our project. In addition, we've used technical pages in our website and the application we've developed consists of web links to them too. That will consist of all the privacy policies, sitemap and about us pages. A video with complete functionality usage of website and application is also uploaded at YouTube for a naïve user. The name of our document that we are currently writing is **Project Report Documentation**.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

In this section we will be discussing testing and evaluation which is the process of assessing the quality, functionality, and performance of the project. It involves systematically checking and verifying the project's components, features, and requirements to ensure that they meet the intended objectives.

6.1. Use Case Testing

Both use case which is admin and user is interacting properly with the reliable manner with no typical error.

6.2. Data flow testing

Data flow is working properly in all the panels



6.3. Performance testing

The data get by the system is working properly

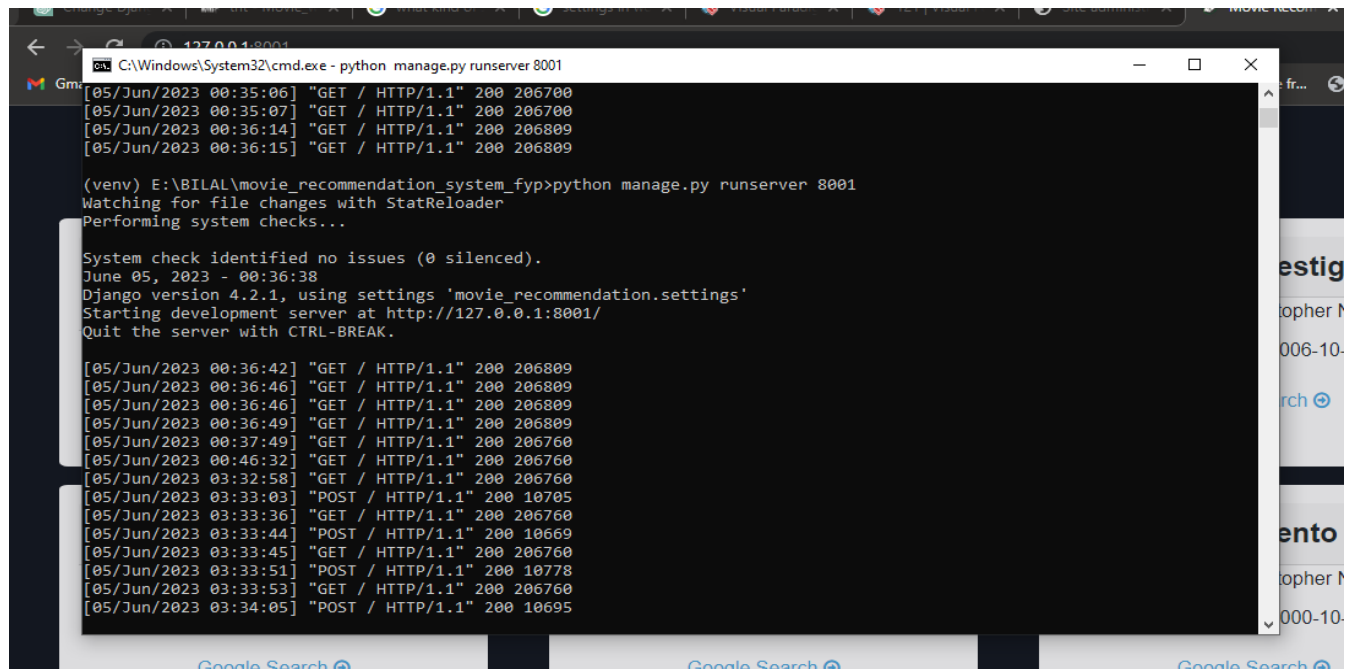
```
[05/Jun/2023 00:30:35] "GET / HTTP/1.1" 200 206389
[05/Jun/2023 00:31:41] "GET / HTTP/1.1" 200 206447
[05/Jun/2023 00:32:19] "GET / HTTP/1.1" 200 206745
[05/Jun/2023 00:32:54] "GET / HTTP/1.1" 200 206767
[05/Jun/2023 00:34:51] "GET / HTTP/1.1" 200 206682
[05/Jun/2023 00:34:51] "GET / HTTP/1.1" 200 206682
[05/Jun/2023 00:34:51] "GET / HTTP/1.1" 200 206682
[05/Jun/2023 00:35:06] "GET / HTTP/1.1" 200 206700
[05/Jun/2023 00:35:06] "GET / HTTP/1.1" 200 206700
[05/Jun/2023 00:35:07] "GET / HTTP/1.1" 200 206700
[05/Jun/2023 00:36:14] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:15] "GET / HTTP/1.1" 200 206809

(venv) E:\BILAL\movie_recommendation_system_fyp>python manage.py runserver 8001
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 05, 2023 - 00:36:38
Django version 4.2.1, using settings 'movie_recommendation.settings'
Starting development server at http://127.0.0.1:8001/
Quit the server with CTRL-BREAK.

[05/Jun/2023 00:36:42] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:46] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:46] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:49] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:37:49] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 00:46:32] "GET / HTTP/1.1" 200 206760
```

6.4. Stress Testing



```
C:\Windows\System32\cmd.exe - python manage.py runserver 8001
[05/Jun/2023 00:35:06] "GET / HTTP/1.1" 200 206700
[05/Jun/2023 00:35:07] "GET / HTTP/1.1" 200 206700
[05/Jun/2023 00:36:14] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:15] "GET / HTTP/1.1" 200 206809

(venv) E:\BILAL\movie_recommendation_system_fyp>python manage.py runserver 8001
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 05, 2023 - 00:36:38
Django version 4.2.1, using settings 'movie_recommendation.settings'
Starting development server at http://127.0.0.1:8001/
Quit the server with CTRL-BREAK.

[05/Jun/2023 00:36:42] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:46] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:46] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:36:49] "GET / HTTP/1.1" 200 206809
[05/Jun/2023 00:37:49] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 00:46:32] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 03:32:58] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 03:33:03] "POST / HTTP/1.1" 200 10705
[05/Jun/2023 03:33:36] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 03:33:44] "POST / HTTP/1.1" 200 10669
[05/Jun/2023 03:33:45] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 03:33:51] "POST / HTTP/1.1" 200 10778
[05/Jun/2023 03:33:53] "GET / HTTP/1.1" 200 206760
[05/Jun/2023 03:34:05] "POST / HTTP/1.1" 200 10695
```

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

7.2. Achievements and Improvements

The goal of this final year project is to develop a comprehensive movie recommendation system with a user-friendly interface. The project utilizes HTML, CSS, JavaScript, and Bootstrap for the frontend development, while Django is used as the backend framework. The main highlight of the project is the implementation of a content-based recommendation system.

The project includes the following key features:

- **User Interface:** The frontend of the application is developed using HTML, CSS, JavaScript, and Bootstrap. The interface is designed to be visually appealing, intuitive, and user-friendly, providing a seamless experience for the users.
- **Movie Database:** A database is set up to store movie-related information, such as movie titles, genres, actors, ratings, and user reviews. This database is accessed and managed using Django's built-in database capabilities.
- **User Authentication:** The application includes a user authentication system, allowing users to register, log in, and manage their profiles. This enables personalized movie recommendations and facilitates user engagement.
- **Content-Based Recommendation System:** The recommendation system is built on the concept of content-based filtering. It analyzes the attributes of movies, such as genres, actors, directors, and ratings, to generate personalized recommendations for users based on their preferences and viewing history. The recommendation algorithm is implemented using Django and leverages machine learning techniques.

- **Search Functionality:** Users can search for movies based on different criteria, such as title, genre, or actor. The application provides a search feature that allows users to easily find specific movies or explore movies within their preferred genres.
- **Rating and Review System:** Users can rate movies and provide reviews, allowing them to share their opinions and experiences with other users. These ratings and reviews contribute to the overall movie ratings and aid in refining the recommendation algorithm.
- **Responsive Design:** The project incorporates responsive design principles to ensure that the application is accessible and optimized for various devices, including desktops, laptops, tablets, and mobile phones.
- **Testing and Evaluation:** The project includes a comprehensive testing phase to ensure the functionality, usability, and reliability of the application. Various testing techniques, such as unit testing, integration testing, and user acceptance testing, are employed to identify and rectify any bugs or issues.

The project aims to provide an interactive and engaging movie recommendation platform for users, enhancing their movie-watching experience and helping them discover new movies based on their preferences. The combination of frontend technologies (HTML, CSS, JavaScript, and Bootstrap) with the backend Django framework, along with the incorporation of a content-based recommendation system, makes this project a valuable contribution to the field of movie recommendation systems.

7.3. Critical Review

- The project demonstrates a good integration of frontend technologies with the Django backend, allowing for a seamless user experience.
- The implementation of a content-based recommendation system is commendable, as it enhances the personalization and relevance of movie recommendations.
- However, the project could benefit from incorporating additional recommendation algorithms, such as collaborative filtering, to provide a more diverse set of recommendations.
- The user interface design could be further improved to enhance the visual appeal and usability of the application.
- While the project includes a user authentication system, it lacks social media integration, which could limit user engagement and sharing functionalities.

7.4. Lessons Learnt

- Integration of frontend and backend technologies: The project highlights the importance of effectively integrating frontend technologies (HTML, CSS, JavaScript, Bootstrap) with a backend framework like Django to create a cohesive and functional web application.
- Implementation of recommendation systems: Developing a content-based recommendation system provides valuable insights into understanding user preferences and enhancing personalized user experiences.
- User authentication and security: Incorporating user authentication and authorization features ensures the security and privacy of user data, promoting trust and confidence in the application.

7.5. Future Enhancements/Recommendations

- Integration of collaborative filtering: Consider implementing collaborative filtering techniques, in addition to content-based filtering, to enhance the recommendation system. Collaborative filtering takes into account user behavior and preferences, providing a more comprehensive and diverse set of recommendations.
- Social media integration: Incorporate social media features, such as sharing movie recommendations, connecting with friends, and accessing user-generated content, to enhance user engagement and expand the reach of the application.
- Advanced search functionality: Enhance the search feature by incorporating advanced filtering options, allowing users to search for movies based on specific criteria like genre, release year, actors, or ratings.

Appendices

Reference and Bibliography

Reference and Bibliography

- [1] <https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system>
- [2] <https://docs.djangoproject.com/en/4.2/>

Index

Index

[A]

[B]

[C]