

Low Level Design Document

for Movie Recommendation Website

Title: MovieBash

Created By: Anushtha Bageria

1 CONTENTS

1	Contents.....	2
2	Introduction.....	3
2.1	Scope of the Document.....	3
2.2	Intended Audience.....	3
2.3	System Overview.....	3
3	Low Level System Design.....	3
3.1	Sequence Diagram.....	3
3.2	Navigation Flow.....	3
3.3	Screen Validations, Defaults and Attributes.....	3
3.4	Components Design Implementation.....	4
3.4.1	Class Diagram.....	4
3.4.2	Pseudo Code	4
3.5	Configurations/Settings.....	4
4	Data Design.....	4
4.1	List of Key Schemas/Tables in Database.....	4
4.2	Details of Access Levels on Key Tables in Scope.....	4
5	Unit Testing.....	4

2 INTRODUCTION

2.1 SCOPE OF THE DOCUMENT

This document contains the low-level functional design for movie recommendation system. It highlights the low-level use cases in recommendation and filtration process. The website also displays movie on the basis of genre.

2.2 INTENDED AUDIENCE

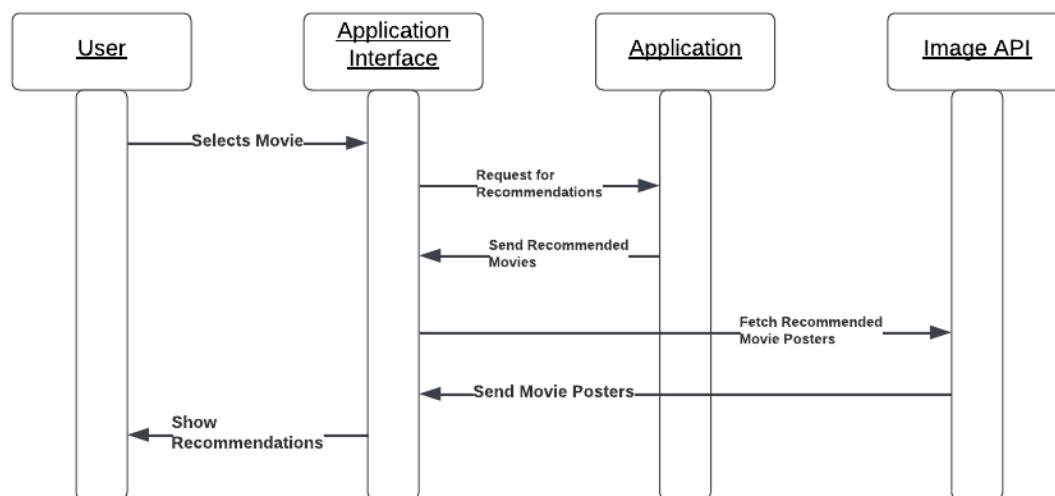
- Engage Mentors
- Judging Team

2.3 SYSTEM OVERVIEW

This system is designed for recommendation of movies to the customers on the basis of the genre of movie they watched in the past days. Also, they can search for a movie in particular genre.

3 LOW LEVEL SYSTEM DESIGN

3.1 SEQUENCE DIAGRAM



3.2 NAVIGATION FLOW

There is only one page in the website which contains a select option for movies and a button to display recommended movies.

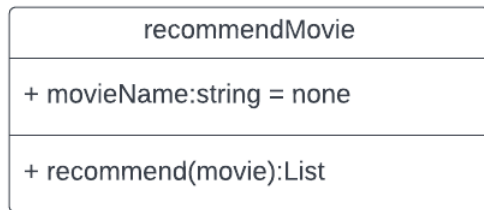
3.3 SCREEN VALIDATIONS, DEFAULTS AND ATTRIBUTES

There are no validations required for accessing the page.

Default value of movie selector is the first movie in the list.

3.4 COMPONENTS DESIGN IMPLEMENTATION

3.4.1 Class Diagram



3.4.2 Pseudo Code

1. First, we load the similarity matrix from the pickle file.
2. Then, we create a function that takes in a movie title as an input and returns a list of recommended movies.
3. Next, we create a DataFrame of movie titles and their corresponding indices.
4. Then, we create a Streamlit app with a title and a selectbox that allows the user to select a movie.
5. If the user clicks the recommend button, we get the index of the movie that the user has selected.
6. We then use the similarity matrix to get the indices of the 10 most similar movies.
7. Finally, we display the titles of the 10 most similar movies.

3.5 CONFIGURATIONS/SETTINGS

1. web: sh setup.sh && streamlit run app.py
2. enableCORS = false
3. headless = true

4 DATA DESIGN

4.1 LIST OF KEY SCHEMAS/TABLES IN DATABASE

There is only one Table DataFrame in database for movie recommendation which contains movie ID, name, cast, description, genre, etc.

4.2 DETAILS OF ACCESS LEVELS ON KEY TABLES IN SCOPE

The table is read only. User cannot modify anything in the table.

5 UNIT TESTING

S. No.	Input/ Test Case	Output/ Recommended Movies
1	Avatar	Mad Max Beyond Thunderdome Iron Man The 13th Warrior The Empire Strikes Back

		<p>Suicide Squad</p> <p>The Wolverine</p> <p>Street Fighter: The Legend of Chun-Li</p> <p>Teenage Mutant Ninja Turtles III</p> <p>Prince of Persia: The Sands of Time</p> <p>Jupiter Ascending</p>
2	The Dark Knight Rises	<p>Michael Clayton</p> <p>Harper</p> <p>Dead Man Down</p> <p>Bless the Child</p> <p>Non-Stop</p> <p>The Client</p> <p>Midnight in the Garden of Good and Evil</p> <p>Mystic River</p> <p>Nighthawks</p> <p>Broken Horses</p>
3	Tangled	<p>Hoodwinked Too! Hood VS. Evil</p> <p>Fantasia</p> <p>The Princess and the Frog</p> <p>Rodeo Girl</p> <p>Fantasia 2000</p> <p>The Simpsons Movie</p> <p>Glee: The Concert Movie</p> <p>The Angry Birds Movie</p> <p>Pooh's Heffalump Movie</p> <p>Hoodwinked!</p>
4	The Avengers	<p>Journey 2: The Mysterious Island</p> <p>Iron Man 3</p> <p>AVP: Alien vs. Predator</p> <p>Raiders of the Lost Ark</p> <p>Independence Day: Resurgence</p>

		<p>Guardians of the Galaxy</p> <p>Serenity</p> <p>The Lone Ranger</p> <p>Jurassic World</p> <p>Shanghai Noon</p>
--	--	--