Abbottabad University of Science & Technology

SOFTWARE REQUIREMENTS SPECIFICATION

(SRS DOCUMENT)

For

<Movie Recommendation System>

Version 1.0

By

Submitted By	**Maryum Qaiser**
Roll no	**14660**
Subject	**Data Structure and Algorithm**
Date	**4/1/2025**

GitHub

https://github.com/MARYUMQAISER/Movie-Recommendation-System

Supervisor

(Sir Jamal Abdul Ahad)

Table of Contents

1. Introduction

1.1 Purpose

This system helps users find movies they might enjoy by recommending options based on their ratings and preferences. It uses smart algorithms to understand what users like and suggests the best movies for them.

1.2 Document Conventions

- Code names like find_similar_users() use snake case.
- Comments explain how each part of the code works.
- Variables are written in clear and descriptive names.

1.3 Project Scope

The system provides:

1. Personalized movie suggestions based on user behavior.
2. Filters for specific movie genres (e.g., horror, comedy).
3. Smart recommendations using a graph-based method.

2. Overall Description

2.1 Product Perspective

The recommendation system builds a network (graph) where users and movies are connected. Movies are recommended based on how users rate them and their similarity to other users.

2.2 User Types

- **Movie Fans**: People who want good movie suggestions quickly.
- **Tech-Savvy Users**: Users who might explore advanced features like genre selection.

2.3 Operating Environment

The system works on any platform with Python installed. It uses the NetworkX library to create and manage graphs.

2.4 Design and Constraints

- **Data Handling**: User ratings and movie genres are stored as simple Python dictionaries.

- **Third-Party Libraries**: Uses NetworkX for graph-based recommendations.
- **Limitations**: The system assumes that all users provide ratings for the movies they've seen.

2.5 Assumptions and Dependencies

- Users are honest in their ratings.
- Python and required libraries like NetworkX are installed.

3. System Features

1. **User Ratings**: The system records the movies users rate and how much they liked them.
2. **Genre Preferences**: Users can select their favorite genres (e.g., sci-fi or comedy) to filter recommendations.
3. **Recommendations**: Suggests movies based on what similar users liked or what fits the user’s preferred genre.

4. Interface Requirements

4.1 User Interface

The system runs on a command-line interface (CLI). It:

1. Asks users about their favorite genre.
2. Recommends movies based on their choice or other user ratings.
3. Shows ratings for specific movies if requested.

5. Quality Features

5.1 Performance

The recommendation process is fast because it uses efficient algorithms to calculate similarity and find top movies.

5.2 Usability

The system is simple to use with clear prompts for users to follow.

5.3 Security

The user data stays local and is not shared or stored online.

6. Appendix

6.1 How It Works

The code connects users and movies in a network (graph). It looks for users with similar tastes and recommends movies they liked. If a genre is selected, it focuses only on movies in that category.

6.2 Example

- A user who likes sci-fi movies and rates "Inception" highly might get recommendations like "The Matrix" or "Interstellar."

This project provides a simple yet powerful way to find great movies quickly.