**ENGINEERING METHOD INTEGRATIVE TASK 2 CYED1**

**JUAN FELIPE JOJOA CRESPO A00382042**

**FELIPE ROJAS PRADO A00393918**

**JOSEPH VERDESOTO**

**Problem statement:**

The well-known company MEGA-MOVIES needs you and your group of programmers to develop a movie recommendation system that uses graph algorithms to offer personalized suggestions to users.

The system should allow users to enter their movie preferences and, using graph algorithms, find similar movies and recommend them based on other users' preferences and interactions. Two versions of the Graph will be used to represent the relationships between the films:

* Implement a version using an adjacency matrix-based representation.
* Deploy another version using an adjacency list-based representation.

Each version of the graph must include the operations necessary to find similar movies and generate recommendations based on the information stored.

The system should allow users to search for movies by genre, director, actors, etc. Also, you should provide an automatic recommendation feature that suggests movies based on user preferences and similarities between movies.

The system should also offer an intuitive graphical user interface that allows users to browse and discover movies, view details of each recommended movie, and add movies to their favorites list.

In short, a movie recommendation system should be developed that uses graph algorithms and offers personalized suggestions to users. The system must have two versions of the graph, allow searching by different criteria and provide a friendly graphical user interface.

**Phase 1: Problem Understanding**

**Problem Description:**

The goal is to develop a movie recommendation system that utilizes graph algorithms to provide personalized suggestions to users. The system should allow users to input their movie preferences and, using graph algorithms, find similar movies and recommend them based on user preferences and interactions with other users. Two versions of the graph will be implemented:

* Version 1: Matrix-based adjacency representation.
* Version 2: List-based adjacency representation.

Each graph version should include the necessary operations to find similar movies and generate recommendations based on stored information. The system should also allow users to search for movies by genre, director, actors, etc. Additionally, an automatic recommendation feature should suggest movies based on user preferences and similarities between movies. The system should provide an intuitive graphical user interface that enables users to explore and discover movies, view details of recommended movies, and add movies to their favorites list.

**Objectives:**

* Develop a movie recommendation system using graph algorithms.
* Provide personalized movie suggestions based on user preferences and similarities between movies.
* Implement two versions of the graph: matrix-based adjacency representation and list-based adjacency representation.
* Enable users to search for movies based on various criteria such as genre, director, actors, etc.
* Include an automatic recommendation feature that suggests movies based on user preferences and movie similarities.
* Create a user-friendly graphical user interface for easy movie exploration, detailed movie viewing, and adding movies to favorites.

**Scope:**

The movie recommendation system will focus on providing personalized movie suggestions to users based on their preferences. The system will utilize graph algorithms and implement two versions of the graph representation. It will allow users to search for movies based on different criteria and include an automatic recommendation feature. The system will also provide a user-friendly graphical interface for seamless movie exploration and interaction.

**Stakeholders:**

* **Users:** Individuals interested in discovering and receiving movie recommendations.
* **System Administrators:** Responsible for system maintenance, updates, and ensuring smooth operation.
* **Movie Database Providers:** The system may require access to a comprehensive movie database for accurate recommendations.

**Constraints:**

* **Performance:** The system should handle a large number of users and movies efficiently to provide real-time recommendations.
* **Privacy:** User preferences and personal information should be securely stored and protected.
* **Availability:** The system should be available and accessible to users without significant downtime.
* **Compatibility:** The graphical user interface should be compatible with various platforms and devices.
* **Maintainability:** The system code should be well-structured, documented, and easily maintainable for future updates and enhancements.

**Functional Requirements:**

**RF1: User registration:** Allow users to create an account in the system.

**RF2 Login:** Allow users to access the system with their credentials.

**RF3 Movie Preferences Entry:** Allow users to enter their movie preferences, such as genre, director, actors, etc.

**RF4 Movie Search:** Allow users to search for movies by genre, director, actors, etc.

**RF5 Personalized Recommendations:** Use graph algorithms to find similar movies and generate recommendations based on user preferences and similarities between movies.

**RF6 Auto recommendation feature:** Provide a feature that automatically suggests movies based on user preferences and similarities between movies.

**RF7 View movie details:** Allow users to view detailed information for each recommended movie, such as synopsis, cast, duration, etc.

**RF8 Add movies to favorites:** Allow users to add movies to their favorites list for easy access in the future.

**RF9 Graphical User Interface (Optional):** Provide an intuitive graphical interface that allows users to browse and discover movies, interact with system functions, and perform actions such as search, view details, and add to favorites.

**Non-functional Requirements:**

**RNF1 Usability:** The system must be easy to use and understandable for users.

**RNF2 Efficiency:** The graph algorithms used to find similarities and generate recommendations must be efficient at runtime.

**RNF3 Scalability:** The system must be able to handle a large number of users and movies without degrading its performance.

**RNF4 Security:** The system must guarantee the confidentiality of user information and protect against possible vulnerabilities.

**RNF5 Availability:** The system must be available and accessible to users at all times, with minimal downtime.

**RNF6 Adaptability:** The system must be adaptable to different platforms and devices, such as computers, tablets, and mobile devices.

**RNF7 Maintainability:** System code must be well structured, documented, and easy to maintain and update in the future.

**REQUIREMENTS TABLE INTEGRATIVE TASK 2 CYED 1**

| Client | MEGA-MOVIES |
| --- | --- |
| User | Administrator, Movie watcher |
| Functional requirements | **RF1 User registration:** Allow users to create an account in the system.  **RF2 Login:** Allow users to access the system with their credentials.  **RF3 Movie Preferences Entry:** Allow users to enter their movie preferences, such as genre, director, actors, etc.  **RF4 Movie Search:** Allow users to search for movies by genre, director, actors, etc.  **RF5 Personalized Recommendations:** Use graph algorithms to find similar movies and generate recommendations based on user preferences and similarities between movies.  **RF6 Auto recommendation feature:** Provide a feature that automatically suggests movies based on user preferences and similarities between movies.  **RF7 View movie details:** Allow users to view detailed information for each recommended movie, such as synopsis, cast, duration, etc.  **RF8 Add movies to favorites:** Allow users to add movies to their favorites list for easy access in the future.  **RF9 Graphical User Interface (Optional):** Provide an intuitive graphical interface that allows users to browse and discover movies, interact with system functions, and perform actions such as search, view details, and add to favorites. |
| Problem context | The MEGA-MOVIES company needs you to develop a movie recommendation system that uses graph algorithms to offer personalized suggestions to users, with the aim of facilitating the search and discovery of movies based on their preferences and similarities with other movies. |
| Non-functional requirements | **RNF1 Usability:** The system must be easy to use and understandable for users.  **RNF2 Efficiency:** The graph algorithms used to find similarities and generate recommendations must be efficient at runtime.  **RNF3 Scalability:** The system must be able to handle a large number of users and movies without degrading its performance.  **RNF4 Security:** The system must guarantee the confidentiality of user information and protect against possible vulnerabilities.  **RNF5 Availability:** The system must be available and accessible to users at all times, with minimal downtime.  **RNF6 Adaptability:** The system must be adaptable to different platforms and devices, such as computers, tablets, and mobile devices.  **RNF7 Maintainability:** System code must be well structured, documented, and easy to maintain and update in the future. |

| Name or identifier | **RF1 User Registration** | | |
| --- | --- | --- | --- |
| Abstract | Allow users to create an account in the system. | | |
| Inputs |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| General activities needed to obtain the results |  | | |
| Result or postcondition |  | | |
| Outputs |  |  |  |
|  |  |  |