# **SDLC**

**1. Database Planning:**

**Goal:** The goal of database planning is to organize the information that the movie recommendation system will use and store. The system needs a well-structured database to handle and manage all the data related to users, movies, ratings, and recommendations in an efficient way.

**Key Parts:**

* **User Information:** This includes users’ basic information, such as their email, preferences, and usernames, which will be saved in the Users table.
* **Movies:** The Movies table will store movie details like title, release year, description, cast, and runtime.
* **Ratings:** The Ratings table will store how users rate movies, along with the time they rated the movie and any reviews they provide.
* **Genres:** The Genres table will store different movie genres and their popularity.
* **Viewing History:** The Viewing History table will keep track of what movies users have watched, including when they watched them, for how long, and on which device.
* **Recommendations:** The Recommendations table will store the movie suggestions generated for each user.

**Features for Planning:**

* **Tracking User Preferences:** The system will save what movies users like and their ratings, so it can give better recommendations.
* **Personalized Movie Suggestions:** It will recommend movies based on the movies that users have watched and rated highly.
* **Genre Browsing:** Users can explore movies by genre to find more options.
* **Auto Spelling Correction:** The system will fix common spelling errors when users enter data, making it more accurate.
* **Sharing Recommendations:** Users can share their favorite movies with friends.
* **Real-Time Updates:** The system will update recommendations and data based on what the user does, keeping everything current.

**2. System Definition:**

**Purpose of the System:** The movie recommendation system’s goal is to help users quickly find movies they will enjoy. It suggests movies based on each user's preferences, viewing history, and ratings. It also allows users to share movie suggestions with others and browse movies by genre.

**System Features:**

* **User Profiles:** The system will store each user’s preferences, movie ratings, and what movies they’ve watched.
* **Movie Database:** A large collection of movie information, including details like title, genre, and user ratings.
* **Recommendation Engine:** The system will suggest movies based on what users have rated and watched, using methods that take into account user preferences.
* **Genre Browsing:** Users can filter and explore movies by genres like action, comedy, etc.
* **Real-Time Updates:** The recommendations will change based on what a user watches or rates recently.
* **Spelling Correction:** Automatically fix typing errors when users search for movies or enter ratings.
* **Social Features:** Users can share movie recommendations with friends or others, creating a community.

**3. Requirement Collection and Analysis:**

**Methods for Gathering Information:**

1. **Interviews:**
   * Interviews were conducted with potential users, both movie lovers and casual viewers.
   * **Findings:** Users wanted personalized recommendations based on their preferences, easy browsing, and a way to rate movies.
2. **Questionnaires:**
   * A survey was shared with users to understand their movie-watching habits and preferences.
   * **Findings:** Most respondents enjoy watching action and animated movies. They also wanted a system that could suggest movies based on ratings from other users.
3. **User Observation:**
   * Observing how people interact with current movie platforms helped to spot frustrations and difficulties in finding movies.
   * **Findings:** Users spend a lot of time scrolling through long lists without finding what they like and often rely on external reviews.

**Key Requirements:**

* **User Needs:** Users want personalized movie suggestions, an easy way to search for movies, and a way to share recommendations.
* **System Features:** The system should have user profiles, a recommendation engine, real-time updates, and auto-correction for misspelled inputs.
* **Data Management:** A solid database design is needed to store and manage all the information about users, movies, ratings, and movie-watching history.

**4. Database Design:**

**Entities and Attributes:**

1. **User:** This stores user-related data like user-id, email, preferences, username, and profile picture.
2. **Movie:** This stores movie details, including movie-id, title, release year, description, and cast.
3. **Rating:** This keeps track of ratings, linking users and movies. It includes rating-id, user-id, movie-id, rating value, and timestamp.
4. **Genres:** This tracks genres and includes genre-id, genre name, description, and popularity score.
5. **Viewing History:** This table stores information on what movies users have watched, when, on what device, and for how long.
6. **Recommendations:** This table stores the movies suggested to each user, along with the recommendation score and timestamp.

**Relationships:**

* **User and Ratings:** One user can give ratings for many movies, and each rating is linked to a single user.
* **Movie and Ratings:** Many users can rate one movie, and each rating is tied to a single movie.
* **User and Viewing History:** A user can have many records in the viewing history, and each record is connected to a single user.
* **Movie and Viewing History:** A movie can be viewed by many users, and each viewing is logged separately.
* **User and Recommendations:** A user can get many movie recommendations.
* **Movie and Recommendations:** A movie can be recommended to multiple users.
* **User and Movie:** Many users can watch the same movie, and each user can watch many movies.
* **Movie and Genre:** Movies can belong to several genres, and genres can include many movies.
* **User and Genre:** A user can have preferences for multiple genres, and genres can be liked by many users.

**5. Application Design:**

**Features of the Application:**

* **User Interface (UI):** The application will have an easy-to-use interface where users can manage their profiles, browse movies by genre, rate films, and view recommendations.
* **Recommendation System:** The system will use algorithms to suggest movies based on the user’s history, ratings, and preferences.
* **Database Interaction:** The application will pull data from the database to show personalized recommendations, movie details, and ratings.

**Design Considerations:**

* The interface will be simple and intuitive, so users can easily navigate through the system.
* Recommendations will be updated automatically as the user watches and rates movies.
* Users will be able to share their movie suggestions on social platforms or within the system.

**6. DBMS Selection:**

**Criteria for Choosing the Right DBMS:**

* **Scalability:** The database system should be able to handle an increasing amount of data as the user base grows.
* **Data Integrity:** The system must ensure that the data stored (such as ratings and user information) is consistent and accurate.
* **Performance:** The DBMS should provide fast access to data, especially when retrieving personalized recommendations.
* **Compatibility:** It should work well with the application to ensure smooth integration.

**Recommended DBMS:**

* **MySQL:** A relational database management system, perfect for handling structured data with many relationships, which is ideal for this project. It’s fast, scalable, and supports complex queries.

**7. Prototyping:**

**Goal of Prototyping:** The prototyping phase involves building an early version of the system to demonstrate its functionality and gather feedback for improvements.

**Prototype Features:**

* **User Profile Management:** Users can create and manage their profiles, updating preferences and viewing history.
* **Movie Recommendations:** A system that suggests movies based on the user’s previous activity and ratings.
* **Rating System:** Users can rate movies they’ve watched and see other users' ratings.
* **Genre Browsing:** Users can explore movies by their preferred genres.
* **Social Sharing:** Users can share their favorite movie recommendations with friends.

**Prototype Goals:**

* **Test the User Interface:** Ensure the interface is easy to use and navigable.
* **Validate Recommendations:** Make sure the recommendations are relevant to users based on their viewing history and ratings.
* **Gather Feedback:** Collect user feedback on the prototype to refine the system before fully developing it.