Movie Recommendation System Project Concept

Pennsylvania State University

SWENG 894 Capstone

Sean Xiao

Software Requirements

Product Backlog

1. The system shall provide user registration function, allowing users to create accounts with unique usernames and passwords.
2. This system shall implement a secure login mechanism for registered users to access their accounts.
3. The system shall allow users to set and update their movie preferences, including preferred genres, actors, and directors.
4. The system shall provide movie browsing functionality, allowing users to explore movies based on different categories such as genre, release year, popularity, etc.
5. The system shall implement a search function that allows users to find specific movies based on title, actor, director, or genre.
6. The system shall allow users to rate movies (e.g. 1-5 stars) and provide written reviews.
7. The system shall generate personalized movie recommendations for each user based on their preferences and rating history.
8. The system shall provide a similar movie suggestion list.
9. The system shall display a list of popular movies based on overall user ratings and viewing trends.
10. The system shall implement recommendation filtering options, allowing users to refine recommendations based on specific criteria such as type and publication year.
11. The system shall maintain a movie database, including detailed information such as titles, genres, directors, actors, release years, and descriptions.
12. The system shall encrypt sensitive user information and implement secure data transmission protocols to ensure data privacy and security.
13. The system shall provide a user configuration page where users can view and edit

their account information and preferences.

14. The system shall enable users to create and manage a list of movies they plan to watch in the future.

User Cases/ User Stories

1. As a new user, I would like to register an account so that I can access personalized movie recommendations.
2. As a registered user, I want to securely log in to my account so that I can access my personalized features.
3. As a user, I want to set and update my movie preferences so that I can receive more accurate recommendations.
4. As a movie enthusiast, I want to browse movies by different categories so that I can discover new movies that interest me.
5. As a user, I hope to search for specific movies by title, actor, director, or genre in order to quickly find movies that interest me.
6. As a user, I want to rate movies and write reviews so that I can share my opinions and help others make wise decisions.
7. As a user, I hope to receive personalized movie recommendations based on my preferences and rating history, so that I can discover new movies that I may like.
8. As a user, I hope to see similar movie suggestions when watching a specific movie, so that I can find more movies that I like.
9. As a user, I want to see a list of popular movies so that I can keep up with current trends and enjoy a wide range of movies.
10. As a user, I want to filter my recommendations based on specific criteria so that I can find movies that match my current mood or interests.
11. As a system administrator, I hope the system can maintain a up-to-date movie database so that users can access comprehensive and up-to-date information.
12. As a user, I hope that my personal information is securely stored and transmitted so that I can trust my data system.
13. As a user, I would like to view and edit my personal profile information and preferences so that I can keep my account up to date.
14. As a user, I want to create and manage a movie viewing list so that I can track the movies I plan to watch in the future.

Sprint Backlog

Prioritized User Stories (with time estimation in days and priority):

1. User registration (Priority: High, Estimate: 3 days)
   a) As a new user, I want to register for an account so that I can access personalized movie recommendations.
2. User login (Priority: High, Estimate: 2 days)
   a) As a registered user, I want to log in to my account securely so that I can access my personalized features.
3. Movie database setup (Priority: High, Estimate: 7 days)
   a) As a system administrator, I want the system to maintain an up-to-date database of movies so that users have access to comprehensive and current information.
4. Basic movie browsing (Priority: High, Estimate: 5 days)
   a) As a movie enthusiast, I want to browse movies from different categories so that I can discover new films that interest me.
5. Basic movie search (Priority: High, Estimate: 4 days)
   a) As a user, I want to search for specific movies by title, actor, director, or genre so that I can quickly find the films I'm interested in.
6. User preference setting (Priority: Medium, Estimate: 3 days)
   a) As a user, I want to set and update my movie preferences so that I can receive more accurate recommendations.
7. Basic recommendation algorithm (Priority: High, Estimate: 8 days)
   a) As a user, I want to receive personalized movie recommendations based on my preferences and rating history so that I can discover new movies I might enjoy.
8. Movie rating and review (Priority: Medium, Estimate: 5 days)
   a) As a user, I want to rate movies and write reviews so that I can share my opinions and help others make informed decisions.
9. Similar movie suggestions (Priority: Medium, Estimate: 5 days)
   a) As a user, I want to see similar movie suggestions when viewing a specific film so that I can find more movies like the ones I enjoy.
10. Popular movies list (Priority: Medium, Estimate: 3 days)
    a) As a user, I want to see a list of popular movies so that I can stay up to date with current trends and widely appreciated films.
11. Recommendation filtering (Priority: Low, Estimate: 5 days)
    a) As a user, I want to filter my recommendations based on specific criteria so that I can find movies that match my current mood or interests.

12. User profile management (Priority: Medium, Estimate: 3 days)
    a) As a user, I want to view and edit my profile information and preferences so that I can keep my account up to date.
13. Watchlist management (Priority: Low, Estimate: 3 days)
    a) As a user, I want to create and manage a watchlist of movies so that I can keep track of films I intend to watch in the future.
14. Data security implementation (Priority: High, Estimate: 7 days)
    a) As a user, I want my personal information to be securely stored and transmitted so that I can trust the system with my data.

Sprint Backlog (Selected high priority and critical user stories):

1. User registration (Priority: High, Estimate: 3 days)

2. User login (Priority: High, Estimate: 2 days)

3. Movie database setup (Priority: High, Estimate: 7 days)

4. Basic movie browsing (Priority: High, Estimate: 5 days)

5. Basic movie search (Priority: High, Estimate: 4 days)

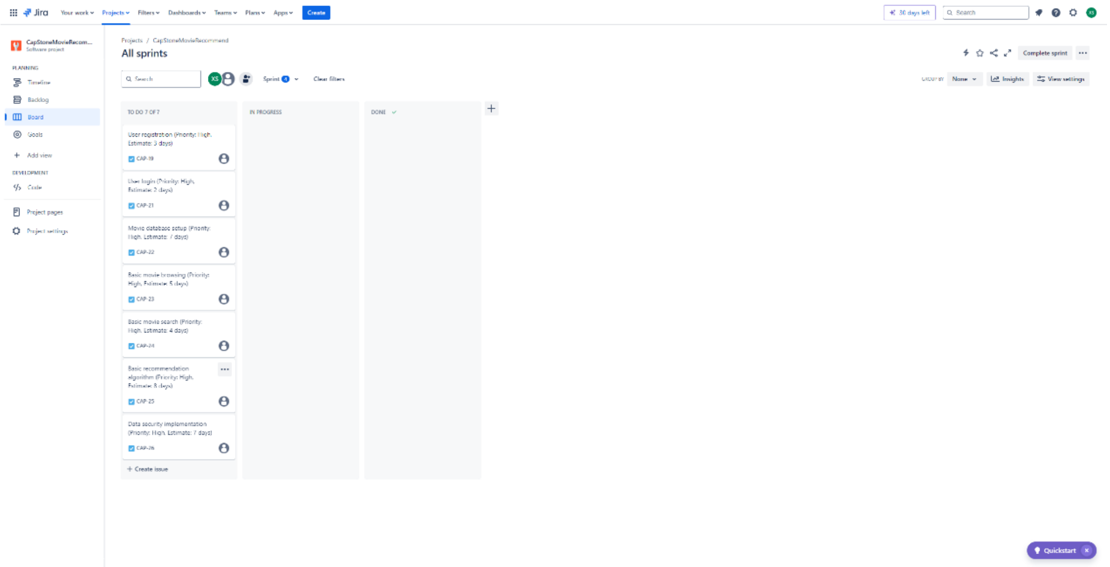
6. Basic recommendation algorithm (Priority: High, Estimate: 8 days)

7. Data security implementation (Priority: High, Estimate: 7 days)

Total Estimated Time for Sprint Backlog: 36 days

Agile Board

https://seanxiao923-1725992916126.atlassian.net/jira/software/projects/CAP/boards/1



Projects / CapStoneMovieRecommend

All sprints

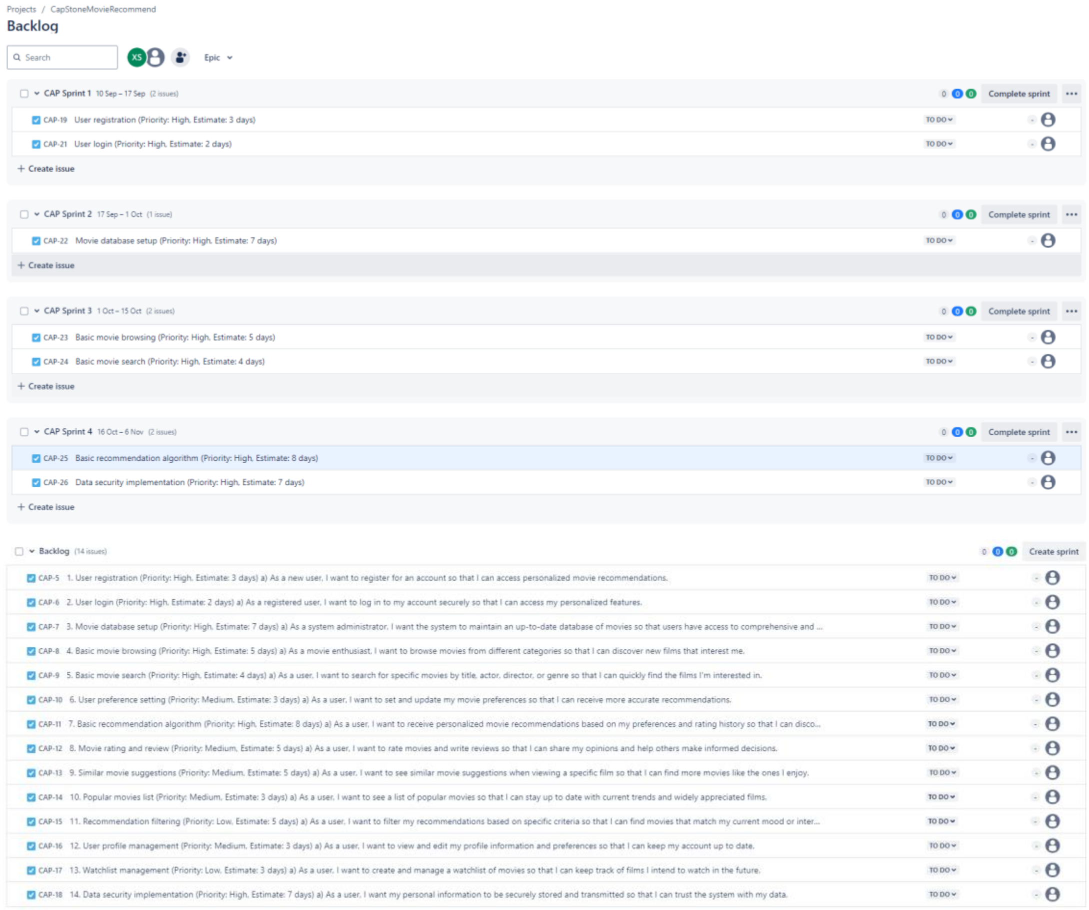
TO DO 7 OF 7

- User registration (Priority: High, Estimate: 3 days) — CAP-19
- User login (Priority: High, Estimate: 2 days) — CAP-21
- Movie database setup (Priority: High, Estimate: 7 days) — CAP-22
- Basic movie browsing (Priority: High, Estimate: 5 days) — CAP-23
- Basic movie search (Priority: High, Estimate: 4 days) — CAP-24
- Basic recommendation algorithm (Priority: High, Estimate: 8 days) — CAP-25
- Data security implementation (Priority: High, Estimate: 7 days) — CAP-26

+ Create issue

IN PROGRESS

DONE ✓



Projects / CapStoneMovieRecommend

Backlog

CAP Sprint 1 10 Sep – 17 Sep (2 issues) 0 0 0

- ☑ CAP-19 User registration (Priority: High, Estimate: 3 days) TO DO
- ☑ CAP-21 User login (Priority: High, Estimate: 2 days) TO DO

CAP Sprint 2 17 Sep – 1 Oct (1 issue) 0 0 0

- ☑ CAP-22 Movie database setup (Priority: High, Estimate: 7 days) TO DO

CAP Sprint 3 1 Oct – 15 Oct (2 issues) 0 0 0

- ☑ CAP-23 Basic movie browsing (Priority: High, Estimate: 5 days) TO DO
- ☑ CAP-24 Basic movie search (Priority: High, Estimate: 4 days) TO DO

CAP Sprint 4 16 Oct – 6 Nov (2 issues) 0 0 0

- ☑ CAP-25 Basic recommendation algorithm (Priority: High, Estimate: 8 days) TO DO
- ☑ CAP-26 Data security implementation (Priority: High, Estimate: 7 days) TO DO

Backlog (14 issues) 0 0 0

- ☑ CAP-5 1. User registration (Priority: High, Estimate: 3 days) a) As a new user, I want to register for an account so that I can access personalized movie recommendations. TO DO
- ☑ CAP-6 2. User login (Priority: High, Estimate: 2 days) a) As a registered user, I want to log in to my account securely so that I can access my personalized features. TO DO
- ☑ CAP-7 3. Movie database setup (Priority: High, Estimate: 7 days) a) As a system administrator, I want the system to maintain an up-to-date database of movies so that users have access to comprehensive and ... TO DO
- ☑ CAP-8 4. Basic movie browsing (Priority: High, Estimate: 5 days) a) As a movie enthusiast, I want to browse movies from different categories so that I can discover new films that interest me. TO DO
- ☑ CAP-9 5. Basic movie search (Priority: High, Estimate: 4 days) a) As a user, I want to search for specific movies by title, actor, director, or genre so that I can quickly find the films I'm interested in. TO DO
- ☑ CAP-10 6. User preference setting (Priority: Medium, Estimate: 3 days) a) As a user, I want to set and update my movie preferences so that I can receive more accurate recommendations. TO DO
- ☑ CAP-11 7. Basic recommendation algorithm (Priority: High, Estimate: 8 days) a) As a user, I want to receive personalized movie recommendations based on my preferences and rating history so that I can disco... TO DO
- ☑ CAP-12 8. Movie rating and review (Priority: Medium, Estimate: 5 days) a) As a user, I want to rate movies and write reviews so that I can share my opinions and help others make informed decisions. TO DO
- ☑ CAP-13 9. Similar movie suggestions (Priority: Medium, Estimate: 5 days) a) As a user, I want to see similar movie suggestions when viewing a specific film so that I can find more movies like the ones I enjoy. TO DO
- ☑ CAP-14 10. Popular movies list (Priority: Medium, Estimate: 3 days) a) As a user, I want to see a list of popular movies so that I can stay up to date with current trends and widely appreciated films. TO DO
- ☑ CAP-15 11. Recommendation filtering (Priority: Low, Estimate: 5 days) a) As a user, I want to filter my recommendations based on specific criteria so that I can find movies that match my current mood or inter... TO DO
- ☑ CAP-16 12. User profile management (Priority: Medium, Estimate: 3 days) a) As a user, I want to view and edit my profile information and preferences so that I can keep my account up to date. TO DO
- ☑ CAP-17 13. Watchlist management (Priority: Low, Estimate: 3 days) a) As a user, I want to create and manage a watchlist of movies so that I can keep track of films I intend to watch in the future. TO DO
- ☑ CAP-18 14. Data security implementation (Priority: High, Estimate: 7 days) a) As a user, I want my personal information to be securely stored and transmitted so that I can trust the system with my data. TO DO

Non-Functional Requirements

1. Performance

requirement:

The system should provide movie recommendations within 5 seconds of user requests and support up to 10 concurrent users without significant performance degradation.

Architecture support:

Efficient database queries: Optimize database queries to reduce response time.

Local cache: Implement memory caching for frequently accessed data.

Asynchronous processing: using backend jobs to perform time-consuming tasks, such as updating recommendations.

Responsibilities of Components:

Database Manager: Process optimized queries and maintain indexes for fast data retrieval.

Cache Manager: Manage local cache to store and retrieve frequently accessed movie and user data.

Task Scheduler: Manage backend jobs to update suggestions during low usage periods.

2. Reliability

requirement:

The system should maintain 99% of normal operating time during runtime and implement basic error handling and logging mechanisms.

Architecture support:

Error handling: Implement try catch blocks and appropriate exception handling throughout the entire application.

Logging system: Use a simple logging framework to track errors and system activity.

Auto restart: Set up a simple script to automatically restart the server application when it crashes.

Responsibilities of Components:

Error handler: Capture and handle exceptions to prevent system crashes.

Recorder: Record system activity and errors for troubleshooting and monitoring.

Application Monitor: Monitor the status of applications and initiate restarts if necessary.

3. Safety

requirement:

The system should implement basic authentication, secure password storage, and prevent common web vulnerabilities.

Architecture support:

Authentication system: Implement simple username/password authentication.

Password hashing: Use strong hashing algorithms (such as bcrypt) to store passwords.

Input verification: Implement server-side input verification to prevent SQL injection and XSS attacks.

Responsibilities of Components:

Authentication Manager: handles user login and session management.

Password Manager: Manage secure password hashing and verification.

Input validator: Disinfect and validate all user inputs before processing.

Architecture design

Overview of Advanced Architecture

Our movie recommendation system is designed as a standalone application that runs on a personal computer serving as a server. This design choice is based on the scale and resource limitations of the system, while still aiming to meet our non-functional requirements for performance, reliability, and security.

Key components

- Web server: processes incoming HTTP requests and provides static content.
- Principle: Provide users with a simple interface to interact with the system through a web browser.
- Application Server: The core of the system, managing application logic and coordinating between different components.
- Basic principle: Centralize application logic, simplify development and maintenance.
- Authentication Manager: Manage user authentication and session processing.
- Basic principle: Addressing security requirements by providing a centralized mechanism for user authentication.

- Movie Manager: handles operations related to movie data, including CRUD operations.
- Principle: Encapsulates all operations related to movies, improving modularity and maintainability.
- Recommendation engine: Generate movie recommendations based on user preferences and behavior.
- Basic principle: The core characteristics of the system are implemented as a separate component to achieve better performance optimization.
- User Profile Manager: Manage user profile information and preferences.
- Basic principle: Separate user specific operations from other components to enhance modularity.
- Search engine: handles movie search functionality.
- Principle: A specialized component used for search operations, allowing optimization of search algorithms.
- User database: stores user information and preferences.
- Reason: Separating user data storage allows for easier data management and potential future expansion.
- Movie database: stores movie information and metadata.
- Principle: Centralize the storage of movie data for convenient and efficient data retrieval and updating.

Design decisions and Rationale

- Monolithic Architecture:
  - Rationale: Given the constraint of running on a personal computer, a monolithic architecture simplifies deployment and reduces system complexity.
- Component-Based Design:
  - Rationale: Although monolithic, the internal structure is divided into components. This improves maintainability and allows for future modularization if needed.
- Separate Databases for Users and Movies:
  - Rationale: Improves data management and allows for independent scaling of user and movie data.
- Integrated Recommendation Engine:
  - Rationale: Keeps the recommendation logic close to the data sources, reducing latency in generating recommendations.
- Local Caching:
  - Rationale: Implement caching within the Application Server to improve

performance, especially for frequently accessed data.

Solutions for non-functional requirements

1. Performance:
   a) Component based design allows for optimization of key components such as recommendation engines and search engines.
   b) Local caching can be implemented in Application Server to reduce database load.
2. Reliability:
   a) Error handling and logging can be implemented across all components.
   b) If necessary, the single-chip design simplifies the restart process.
3. Safety:
   a) The authentication manager has centralized security issues.
   b) User data is separated in their own database, allowing for more centralized security measures.

This architectural design provides a balance between meeting the functional requirements of the system and adhering to the constraints of running on a personal computer. It allows for future expansion and optimization while maintaining the manageability and effectiveness of the current implementation.