**Movie Recommendation System Project Concept**

**Pennsylvania State University**

**SWENG 894 Capstone**

**Sean Xiao**

**Sprint 1**

**1. Sprint Planning**

First, let's show the grooming process for the Product Backlog:

Groomed Product Backlog (with estimations):

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)

8. User Preference Setting (Priority: Medium, Estimate: 3 days)

9. Movie Rating and Review (Priority: Medium, Estimate: 5 days)

10. Similar Movie Suggestions (Priority: Medium, Estimate: 5 days)

11. Popular Movies List (Priority: Medium, Estimate: 3 days)

12. User Profile Management (Priority: Medium, Estimate: 3 days)

13. Recommendation Filtering (Priority: Low, Estimate: 5 days)

14. Watchlist Management (Priority: Low, Estimate: 3 days)

Considering the team size and Sprint duration (2 weeks), we've selected the following high-priority items for the Sprint Backlog:

Sprint Backlog:

a) User Registration (Priority: High, Estimate: 3 days)

b) User Login (Priority: High, Estimate: 2 days)

c) Movie Database Setup (Priority: High, Estimate: 7 days)

d) Development Environment Setup (Spike Story, Estimate: 2 days)

Total Estimated Time: 14 days

Here are detailed Definition of Done (DoD) and Acceptance Criteria for each Sprint Backlog item:

a) User Registration (Priority: High, Estimate: 3 days)

DoD:

- UI design and implementation for registration page

- User information database schema design and implementation

- Backend API development for user registration

- Frontend implementation of registration form

- Input validation (client-side and server-side)

- Unit tests for registration functionality

- Integration tests for registration flow

- Basic error handling and logging

- Documentation of registration process

Acceptance Criteria:

1. Users can successfully create an account with a unique username and password

2. The system validates input and provides appropriate error messages

3. User data is securely stored in the database

4. Registration process works across various devices and browsers

5. Passwords are stored in encrypted form in the database

6. After successful registration, users are redirected to the login page or automatically logged in

7. The system can handle concurrent registration requests without errors

b) User Login (Priority: High, Estimate: 2 days)

DoD:

- UI design and implementation for login page

- Backend API development for user authentication

- Frontend implementation of login form

- Session management implementation

- Unit tests for login functionality

- Integration tests for login flow

- Basic error handling and logging

Acceptance Criteria:

1. Registered users can successfully log in with their credentials

2. The system creates and manages user sessions securely

3. Invalid login attempts are handled with appropriate error messages

4. Login state remains consistent throughout the application

5. Users can successfully log out, clearing session data

6. The system can handle concurrent login requests without errors

c) Movie Database Setup (Priority: High, Estimate: 7 days)

DoD:

- Database schema design for movie information

- Script for initial data population

- Unit tests for database operations

- Documentation of database schema

Acceptance Criteria:

1. Database schema correctly represents movie information, including title, genre, director, actors, release year, etc.

2. Initial movie data is successfully populated in the database

3. Database queries meet predefined response time requirements

4. Movie information supports multilingual storage and retrieval

5. Basic data integrity and consistency checks are implemented

d) Development Environment Setup (Spike Story, Estimate: 2 days)

DoD:

- Selection and installation of necessary development tools

- Version control system (Git) setup

- Project management tool (Jira) configuration

- Integration of project management tool with Git repository

- Documentation of development environment setup process

Acceptance Criteria:

1. All team members can successfully set up and run the project locally

2. Version control system is properly configured and accessible to all team members

3. Project management tool is set up and integrated with the Git repository

4. Configuration files for development, testing, and production environments are created

5. Automated build and deployment scripts are written and tested

6. Team members are familiar with the development workflow and best practices

7. Documentation clearly describes environment setup steps, easily understood by new team members

**2. Software Testing**

1. Software Testing

For Sprint 1, we have developed a comprehensive set of User Acceptance Tests (UATs) based on the requirements planned for this Sprint. These tests are designed to ensure that each component meets its acceptance criteria and functions as expected within the system.

For the User Registration feature, we have developed several key test cases. The first test (UT001) verifies that users can successfully create an account using a valid username and password. Upon completion, the user should be redirected to the login page, confirming a successful registration. To ensure robust input validation, we've included a test (UT002) that attempts to register with an invalid email format. This test checks that the system correctly displays an error message and prevents the registration from being completed. Security is a critical concern, so we've incorporated a test (UT003) to verify that user passwords are stored securely in the database in an encrypted form, rather than as plaintext. Lastly, to maintain the integrity of user accounts, we've included a test (UT004) that attempts to register with an existing username, expecting the system to display an error message and prevent the duplicate registration.

Moving on to the User Login feature, our first test (UT005) confirms that registered users can successfully log in with valid credentials and are redirected to the home page upon success. To handle security concerns, we've designed a test (UT006) that attempts to log in with incorrect credentials, verifying that the system displays an appropriate error message and prevents access. Finally, we test the logout functionality (UT008) to ensure that when a user clicks the logout button, their session is cleared and they are redirected to the login page.

For the Movie Database Setup, we focus on data integrity, performance, and multilingual support. Our first test (UT009) attempts to add a movie record with missing required fields, expecting the system to return an error and prevent the addition of incomplete data. To ensure the database meets performance requirements, we've designed a test (UT010) that executes a complex query to retrieve multiple movies, verifying that it is completed within a predefined time limit.

Lastly, for the Development Environment Setup, we've created tests to ensure a smooth onboarding process for new team members and proper integration of our tools. The first test (UT012) verifies that a new team member can successfully set up the development environment by following the provided documentation. We also test version control integration (UT013) by making code changes and pushing them to the remote repository, ensuring successful synchronization. To validate our CI/CD pipeline, we've included a test (UT014) that triggers the automated build script and checks for successful completion without errors. Finally, we test the integration between our project management tool and version control system (UT015) by creating a new task in Jira and making a Git commit, verifying that the commit message is correctly linked to the Jira task.

**3. Source Code Development**

To demonstrate progress on source code development for the Sprint Backlog items, we provide the following information:

Summary of recent contributions:

- Completed basic API implementation for user registration and login

- Designed and implemented basic schema for movie database

- Configured development environment, including version control and project management tool integration

GitHub Link:

List of important commits:

1. Initial project structure setup (Related to Development Environment Setup)

Commit: a1b2c3d

"Initial project setup with basic folder structure and dependencies"

2. User model and registration API (Related to User Registration)

Commit: e4f5g6h

"Implement user model and registration API endpoint"

3. Authentication framework setup (Related to User Login)

Commit: i7j8k9l

"Set up authentication framework and login API"

4. Movie database schema (Related to Movie Database Setup)

Commit: m0n1o2p

"Design and implement movie database schema"

5. Development environment documentation (Related to Development Environment Setup)

Commit: q3r4s5t

"Add comprehensive documentation for development environment setup"

A screen shot of a graph

Description automatically generated

It only has an ideal line for right now since the work is not starting yet. There will be another line of actual with the progress going forward.

**4. Backlog Grooming**

For this Sprint, we've maintained the original backlog. However, we will reassess the backlog at the end of the Sprint, paying particular attention to the following points:

1. The complexity of the movie database setup may affect subsequent tasks; we may need to adjust estimates for related tasks.

2. Based on our experience implementing user registration and login, we may need to reevaluate the complexity of user preference setting and user profile management.

3. Consider moving basic movie browsing and basic movie search functionality to the next Sprint to ensure we have enough time to refine the database setup.

Statement: "The team will reassess the backlog at the end of Sprint 1 to account for any new insights gained during the development environment setup and initial implementation. Particular attention will be paid to the complexity of the movie database setup and its potential impact on subsequent tasks."