Software Requirements Specification: Letterboxd

Introduction

1.1 Purpose

To develop a user-friendly, interactive platform where users can discover, rate, and review tasks or items, similar to Letterboxd's functionality for films. The platform will offer features like user authentication, search, task addition, reviews, and social interaction.

1.2 Intended Audience

People who want to log, track, and review their activities or tasks in a structured and engaging manner, fostering community interaction.

1.4 Product Scope

This system serves as a social task management platform with features such as task addition, reviews, ratings, and user interaction. It bridges personal productivity and community engagement by allowing users to log tasks/items, discover recommendations, and connect with others.

1.5 Definitions and Acronyms

- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **UX:** User Experience
- **JWT:** JSON Web Token (used for secure authentication)
- **OAuth:** Open Authorization (protocol for social logins)
- **CRUD:** Create, Read, Update, Delete

2. Overall Description

2.1 User Needs

- Users need a seamless way to log, rate, and review tasks/items.
- Users require the ability to search for tasks/items by title, genre, or category.
- Users need recommendations based on their activity and preferences.
- Users want to engage with others by following, commenting, and sharing curated lists.

2.2 Assumptions and Dependencies

- Users have basic knowledge of online platforms and authentication processes.
- A stable internet connection is assumed for all users.
- The system relies on third-party services for social logins (e.g., Google, Facebook).

- The system depends on backend hosting services (e.g., AWS, Heroku) for deployment.

3. System Features and Requirements

3.1 Functional Requirements

1. **User Authentication:**
    - Allow users to sign up, sign in, and log out.
    - Enable social media login using OAuth (e.g., Google, Facebook).
    - Support password reset functionality.
2. **Task/Item Management:**
    - Users can add tasks/items with attributes like title, description, category, and images.
    - Users can edit and delete tasks/items they've created.
3. **Search Functionality:**
    - Provide search capabilities based on task/item title, genre, or date.
4. **Review and Rating System:**
    - Users can rate tasks/items on a 5-star scale.
    - Users can write, edit, and delete reviews for tasks/items.
5. **Social Features:**
    - Allow users to follow others and view their activity (e.g., reviews, ratings).
    - Enable users to comment on reviews.
6. **Recommendations:**
    - Suggest tasks/items based on user preferences and activity.
7. **Curated Lists:**
    - Display featured lists (e.g., "Top Rated Tasks") and allow users to create and share their own lists.

3.2 External Interface Requirements

1. **User Interfaces (UI):**
    - A clean and intuitive UI, responsive across devices (desktop, tablet, mobile).
    - Features like forms, buttons, search bar, and task/item cards.
2. **API Interfaces:**
    - RESTful APIs for backend communication with the database.
    - Integration with third-party authentication APIs (e.g., Google OAuth).
3. **Hardware Interfaces:**
    - Compatible with modern browsers and devices.

3.3 System Features

Feature 1: User Authentication

- Users must be able to securely log in, sign up, and reset passwords.

Feature 2: Task/Item Addition

- Users can create tasks/items with relevant metadata like title, category, description, and images.

Feature 3: Reviews and Ratings

- Each task/item page displays average ratings and user reviews.

Feature 4: Watchlist Management

- Users can add tasks/items to a personal watchlist for future reference.

Feature 5: Recommendations

- The system generates recommendations for users based on their interactions and ratings.

3.4 Nonfunctional Requirements

1. **Performance:**
    - The platform should load within 3 seconds under normal traffic.
2. **Scalability:**
    - The system should handle 10,000 users with minimal performance degradation.
3. **Security:**
    - Use HTTPS for secure data transmission.
    - Encrypt user credentials with industry-standard practices.
4. **Usability:**
    - The interface must be easy to navigate and visually appealing for all users.
5. **Reliability:**
    - The platform should ensure 99.9% uptime.