Hamza Damiel

REQUIREMENTS AND UML DOCUMENT

Contents

[Functional Requirements 1](#_Toc145079480)

[Non-Functional Requirements 2](#_Toc145079481)

[Sections 3](#_Toc145079482)

[Class Diagram 3](#_Toc145079483)

[Functional Diagram 5](#_Toc145079484)

[Sequence Diagram 6](#_Toc145079485)

[Package Diagram 7](#_Toc145079486)

[Deployment Diagram 7](#_Toc145079487)

[Use Case 8](#_Toc145079488)

# Functional Requirements

1. **Login:**
   * Registered Spotify users can log in securely using their credentials.
2. **Dashboard Access:**
   * Users must have access to the dashboard upon successful login.
3. **Terms and Conditions Consent:**
   * Users must provide consent to terms and conditions before accessing the dashboard.
4. **Statistics Display:**
   * The dashboard should display statistics, including top artists, tracks, and genres over a specified time.
5. **Music Discovery and Search:**
   * Users should be able to discover music recommendations based on their listening history.
   * Recommendations should be searchable and filterable.
6. **User Profile Management:**
   * Users must be able to view and edit their profiles.
   * Profile edits should include changing profile pictures and personal information.
7. **Data Visualization:**
   * The platform should present data insights through charts and graphs.
   * Visualisations should include listening trends and favourite genres.
8. **Music Recommendations:**
   * Users should receive personalized music recommendations based on their listening history.
9. **Playlist Management:**
   * Users should be able to create, edit, and manage playlists within the platform.
   * Changes made to playlists should sync with their Spotify accounts.
10. **Settings Customization:**
    * Users must have options to customize the platform's theme.
    * Users should be able to send feedback for improvements.
    * Access to terms of service and privacy policy should be available.
    * Users should have the ability to log out from their accounts.

# Non-Functional Requirements

1. **Performance:**
   * The platform should respond quickly to user interactions.
   * Data visualizations should load efficiently.
2. **Security:**
   * User data should be stored securely and protected from unauthorized access.
   * OAuth 2.0 should be used for secure user authentication.
3. **User Experience (UX):**
   * The platform should provide an intuitive and user-friendly interface.
   * Mobile responsiveness is required for seamless access on various devices.
4. **Scalability:**
   * The system should be designed to handle an increasing number of users and data over time.
5. **Data Privacy:**
   * The platform must comply with data privacy regulations and Spotify's terms of service.
6. **Accessibility:**
   * The platform should be accessible to users with disabilities, adhering to accessibility standards
7. **Reliability:**
   * The system should be always available and reliable for users.
8. **Testing:**
   * Rigorous testing should be conducted to ensure all features operate smoothly and meet user expectations.
9. **Documentation:**
   * Comprehensive user documentation should be provided, including guides on using platform features.
10. **Data Backup and Recovery:**

* Regular automated backups of user data should be performed.
* A disaster recovery plan should be in place to minimize data loss in case of system failures.

# Sections

**Homepage:** The main landing page that provides an overview of the platform's features and encourages users to use the service

**User Login:** The login page where registered users of Spotify can enter their credentials to access the dashboard.

**Terms and Conditions**: Page just before user logs in where they consent to sharing data

**Statistics:** Display the main statistics (eg top artists, tracks, genres over a time period)

**Discover/Search**: Find recommendations based on the music that the user listens to and allow these to be added to their playlists

**Settings**: Provides options for changing the theme, sending feedback, accessing terms and privacy policies, and logging out.

# Class Diagram

**A diagram of a website

Description automatically generated**

# Functional Diagram

**A diagram of a software

Description automatically generated**

# Sequence Diagram

**A diagram of a sequence diagram

Description automatically generated**

# Package Diagram

**A diagram of a application

Description automatically generated**

# Deployment Diagram

**A white cube with black text

Description automatically generated**

# Use Case

**A diagram of a computer program

Description automatically generated**