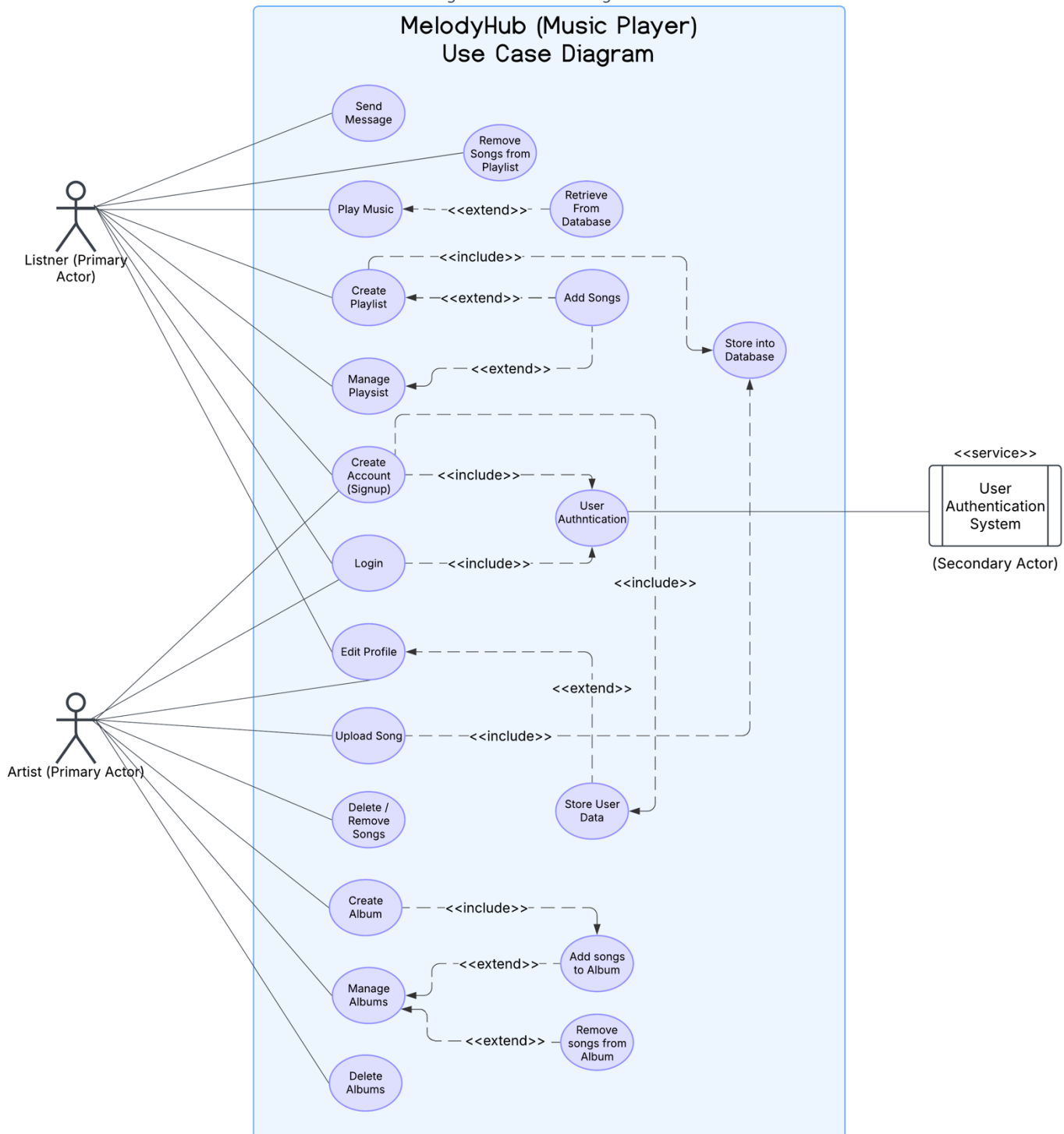


Group: Spring-2025-v2 5

Use Case Diagram

Figure 1: Use Case Diagram



Note: This Use Case Diagram represents the MelodyHub Music Player, an online platform where users can stream, upload, and manage music content. The system has two primary actors (Listeners and Artists) and one secondary actor (User Authentication System). The system is designed with Object-Oriented Design (OOD) principles to ensure scalability and maintainability.

Actors & Their Descriptions

1. **Listener (Primary Actor):** A general user who can stream music, create and manage playlists, and edit their profile. The Listener interacts primarily with the music library, creating personalized experiences.
2. **Artist (Primary Actor):** A user with the ability to upload songs, create albums, and manage their music collection. Artists interact with both the system and listeners, providing new content for the platform.
3. **User Authentication System (Secondary Actor):** This system ensures secure access to user accounts by handling user login, account creation, and authentication processes. It validates users before allowing access to features like uploading songs or managing playlists.

Use Case Descriptions

1. **Play Music:** A listener selects a song from the platform and plays it. (Extends: Retrieve from Database)
2. **Remove Songs from Playlist:** A listener removes unwanted songs from a playlist they created. (Extends: Manage Playlist)
3. **Retrieve from Database:** The system fetches the requested song data from the backend database when a user wants to play music.
4. **Create Playlist:** A listener creates a new custom playlist by naming it and selecting songs. (Extends: Add Songs)
5. **Add Songs:** Adds selected songs to a playlist or album. (Used by: Create Playlist, Manage Playlist, and Create Album)
6. **Store into Database:** Saves newly added songs, playlists, and albums into the backend storage. (Used by: Upload Song and Add Songs)
7. **Manage Playlist:** Allows listeners to rename, reorder, or remove songs from their playlist. (Extends: Remove Songs from Playlist)
8. **Create Account (Signup):** A new user registers on the platform by providing details like username, email, and password. (Includes: User Authentication)
9. **Login:** Users log into the system using their credentials. (Includes: User Authentication)
10. **User Authentication:** A service use case that verifies user credentials to allow secure access. (Used by: Login, Create Account (Signup))
11. **Edit Profile:** Users can modify their personal information, such as their profile picture, bio, or password. (Extends: Store User Data)
12. **Store User Data:** Saves updated user information (such as profile updates) into the database.
13. **Upload Song:** Artists can upload new songs, providing metadata such as title, genre, and album details. (Includes: Store into Database)

- 14. Delete/Remove Songs:** Artists can delete songs they previously uploaded, removing them from the platform.
- 15. Create Album:** Artists can compile multiple songs into an album with a title and cover image. (Includes: Add Songs to Album)
- 16. Modify Albums (Manage Albums):** Artists can update their albums by adding or removing songs. (Extends: Add Songs to Album, Remove Songs from Album)
- 17. Delete Albums:** Artists can permanently delete an album from the platform.
- 18. Add Songs to Album:** Allows artists to include additional songs in an existing album. (Used by: Modify Albums)
- 19. Remove Songs from Album:** Artists can remove specific songs from an album. (Used by: Modify Albums)
- 20. Send Messages:** The users can send messages and receive messages from other users using MelodyHub.

Robustness Diagrams:

Figure 2: Login Setup

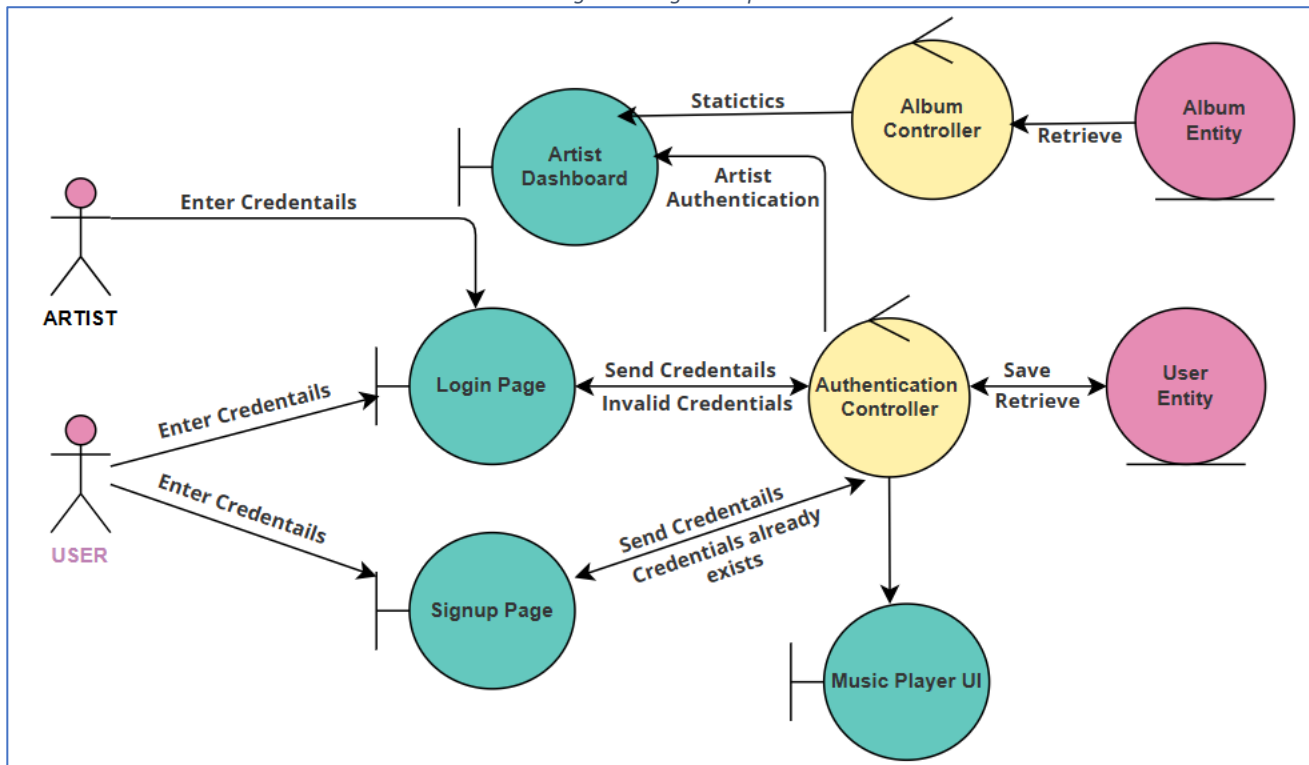


Figure 3: Add Song

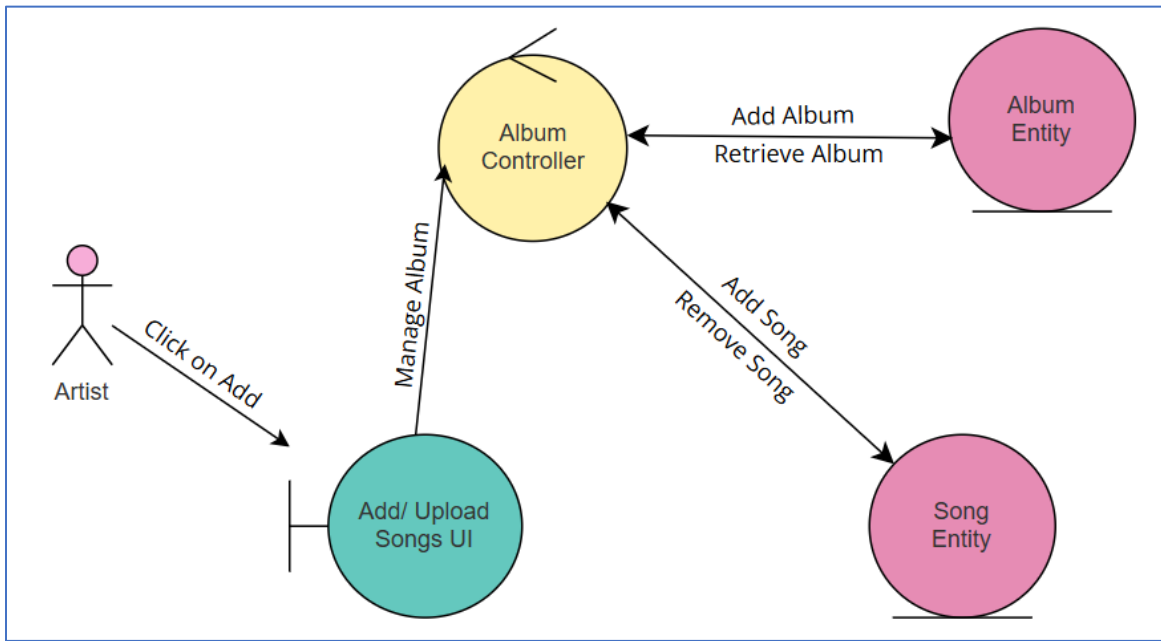


Figure 4: Play Song

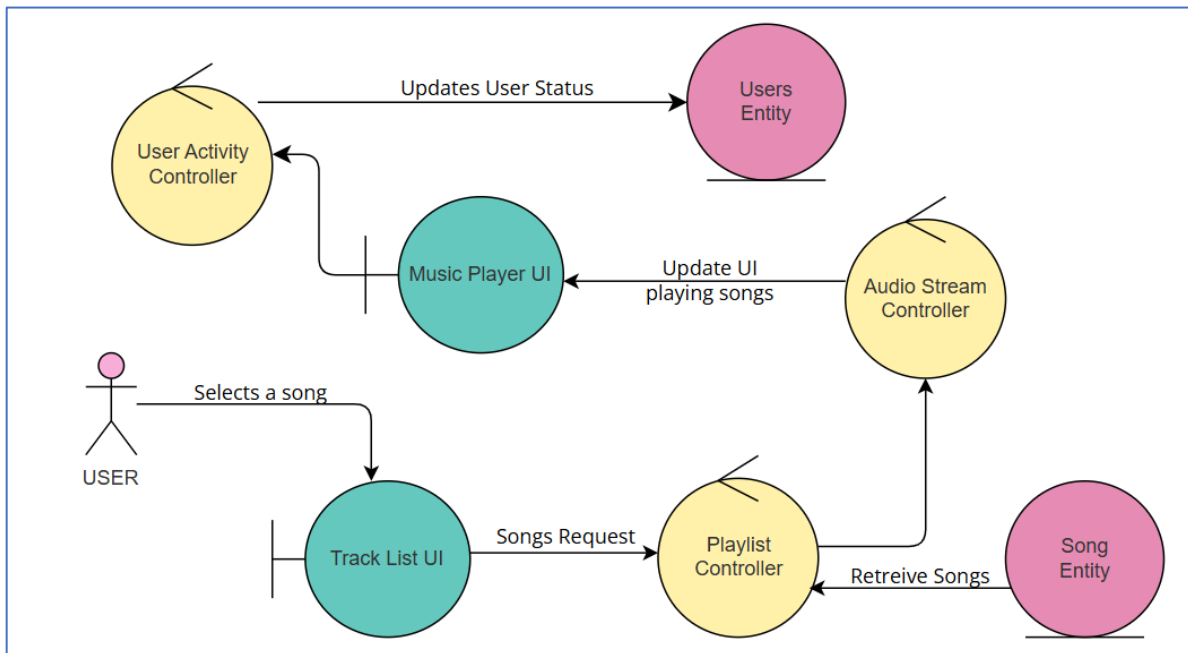


Figure 5: Manage Playlist

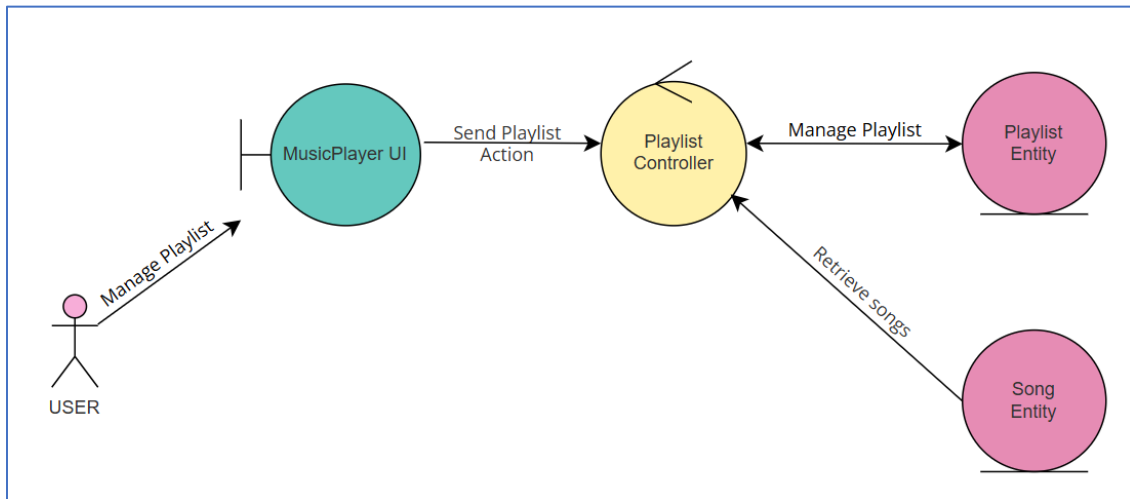


Figure 6: Message Exchange

