Design Document for Cyclone Sounds
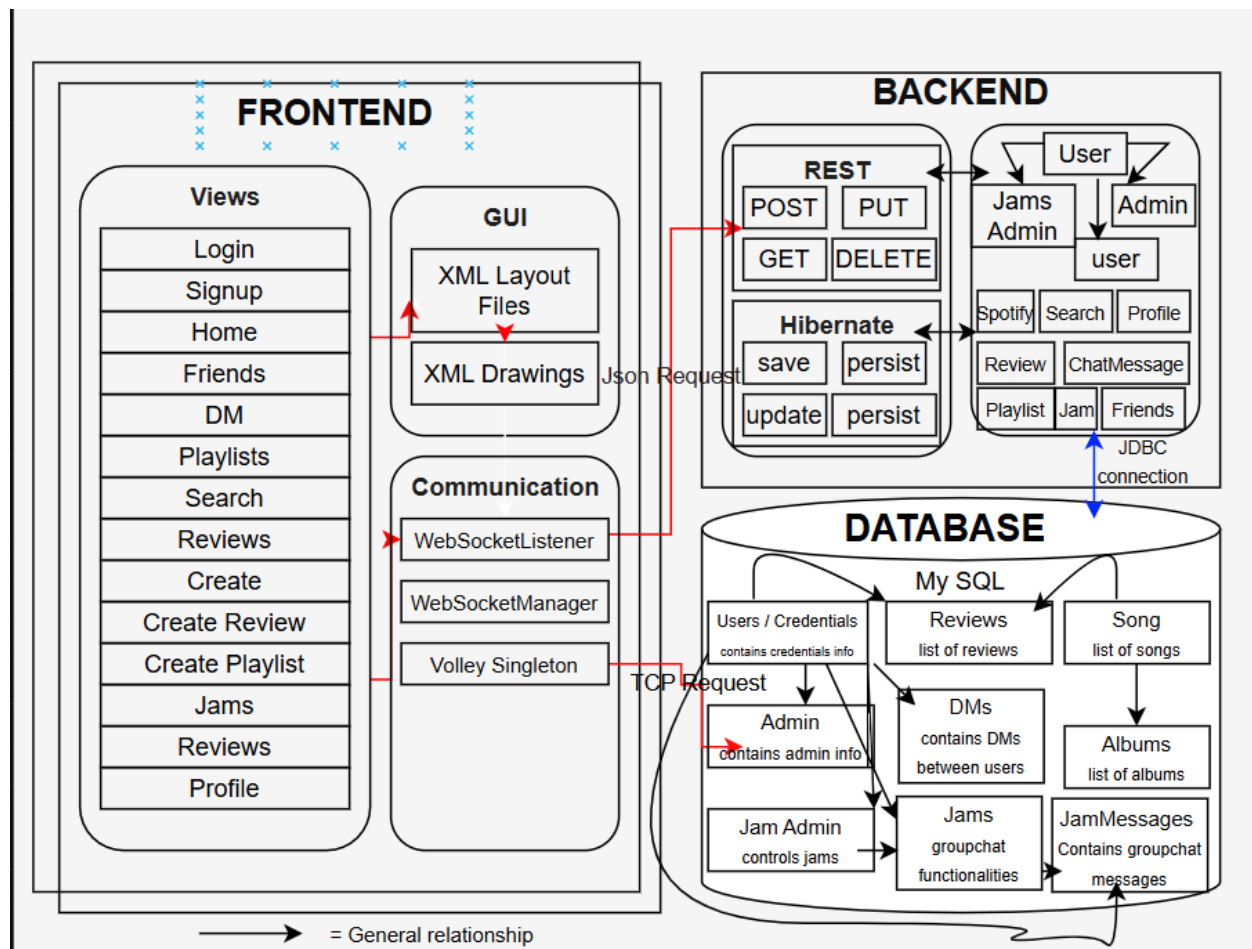
Group 1_srijita_7

Alex DeWitt: 25% contribution

Jack Danby: 25% contribution

Mark Seward: 25% contribution

Martin Kochadampally: 25% contribution

Block Diagram

FRONTEND

Views
- Login
- Signup
- Home
- Friends
- DM
- Playlists
- Search
- Reviews
- Create
- Create Review
- Create Playlist
- Jams
- Reviews
- Profile

GUI
- XML Layout Files
- XML Drawings

Communication
- WebSocketListener
- WebSocketManager
- Volley Singleton

Json Request
TCP Request

BACKEND

REST
- POST
- PUT
- GET
- DELETE

Hibernate
- save
- persist
- update
- persist

User
Jams Admin
Admin
user
Spotify | Search | Profile
Review | ChatMessage
Playlist | Jam | Friends

JDBC connection

DATABASE

My SQL

Users / Credentials
contains credentials info

Reviews
list of reviews

Song
list of songs

Admin
contains admin info

DMs
contains DMs between users

Albums
list of albums

Jam Admin
controls jams

Jams
groupchat functionalities

JamMessages
Contains groupchat messages

→ = General relationship

Frontend

- Authentication & User Management Users authenticate via Login (GET) or register via Signup (POST) with role selection (User, Admin, JamManager). The ProfileActivity displays user details and supports updates (PUT) or account deletion (DELETE). The HomeActivity acts as the central navigation hub connecting all features.
- Social Features FriendsActivity manages connections (GET) and handles friend requests (POST). DMActivity provides real-time bidirectional messaging between users using WebSockets and a RecyclerView interface.
- Music Discovery & Playlists SearchActivity allows querying for Songs or Profiles (GET). Users manage collections in MyPlaylistsActivity, using TableLayouts to create playlists and add/remove songs via Volley requests (POST/DELETE).
- Jams (Group Sessions) JamsActivity lists active sessions; creation is restricted to Admin or JamManager accounts. IndividualJamActivity connects users via WebSockets for live group chat and features a song suggestion system where users send requests for Admin approval.
- Reviews Users submit song ratings (0-5) and descriptions via CreateReviewActivity (POST). MusicActivity lists these reviews, allowing community interaction through upvotes/downvotes (PUT) and entry deletion.

Backend

Communication

- Get: Used mainly for retrieving specific objects using an identifier and to search for objects using a search key, but is also used to list all.
- Put: Used to update a specific object in the database using an identifier.
- Post: Used to create new objects into the corresponding database.
- Delete: Used to delete an object based on a specified identifier.

Controllers

- Credentials (Users) - For full CRUD operation user credentials and has one to one relation with Profiles.
- Profile - Stores some info about the User and is connected to Credentials by username variable and has a One to Many relationship with Reviews
- Reviews - Allows users to review songs and has a Many to One relation with Songs and is connected to songs by the songs id and is connected to Profiles by username
- Songs - Has a Many to Many relation with Playlists and it stores information about various songs, which it pulls from the Spotify API.
- Playlist - One to One relation with Jams, and allows users to store songs they like in groups for easy access.
- Jams - Allows user to chat in a groupchat and make playlists together in real time.
- DMs - Allows users to to talk with one another, and its links the message with the two people talking based on thier usernames.
- Search - Users can search Songs ans other Profiles and results are ordered by popularity