Music Circle

Block Diagram

Group CS_4

Aaron Mueller:	25% contribution
Chance Kupersmith:	25% contribution
Priyanka Kadaganchi:	25% contribution
Indrajeet Aditya Roy:	25% contribution

BLOCK DIAGRAM PICTURE- Created the picture using drawIO

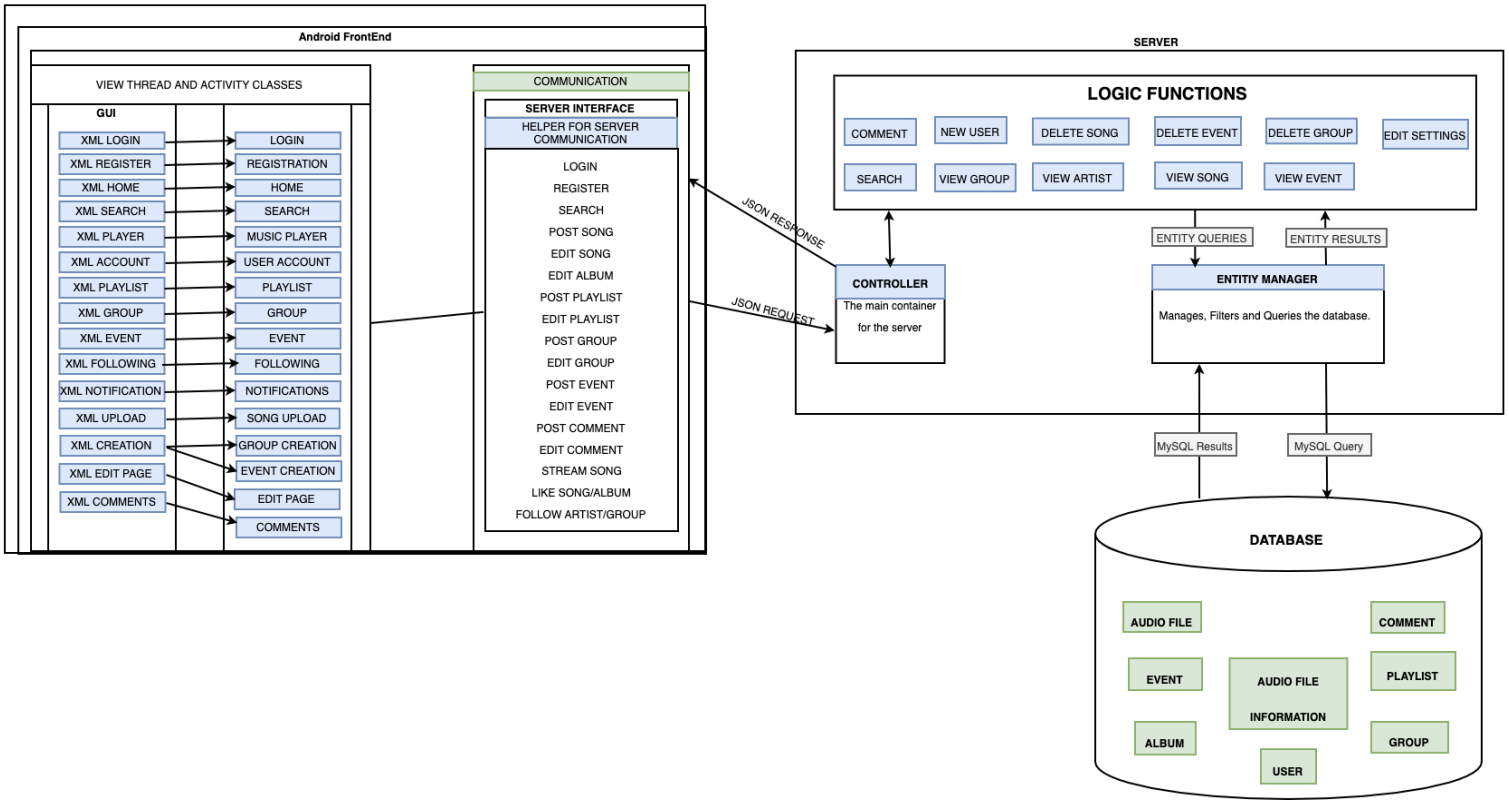

Fig 1:Block diagram.

Main idea:

Our app connects people through the music they listen to, bridging the gap between creators and listeners. Creators will be able to upload songs and interact with their followers through community posts. Users can browse and listen to songs and form groups with people that have similar interests, creating a strong and engaged music community.

Features:

1. **Creating an user account:** An individual can use the app via registration and login by inputting user parameters.
2. **Upload music files:** A registered and logged in user can upload music files which can be seen and played by other users.
3. **View user and creator accounts:** An user can view other user accounts and creator accounts.
4. **Liking a song:** An user can express their opinion about a song by downvote or upvote.
5. **Commenting on a song:** An user can express their opinion about a song by commenting on a song.
6. **Making groups of users:** A user can form a group with other users.
7. **Listening to creator song uploads:** Users can go to a creator's profile and listen to tracks.
8. **Streaming/listening to music:** A user can stream uploaded music to their phone.
9. Uploading a single song as a group of users: Verification system for groups of musicians to post created music.
10. **Post info about live music events (geographic location, type of music being played at an event):** Users can view local live music events happening in their area posted by other users or creators.

Design Description: This part describes the complex parts of your design.

1. MUSIC PLAYER PAGE

The purpose of the page is to successfully fulfil one of the features of the application, being that users will be able to download and listen to songs.The page's functionality also enables users to view the song artist's profile page and see what other albums or songs the artist has uploaded.

The user can use the navigation menu to navigate easily between pages in fewer steps in comparison to continuously pressing the back button multiple times on the android device. The User can view notifications relevant to the user on the Notifications page via the Notifications Button. The user can view his/her profile at any place in the app via the User Profile Button. The User can view the Album art via the image box. The User can play and pause the song via the music player's play/pause button, increase the playing rate of the song via the fast forward button and decrease the playing rate of the song via the reverse button. Additionally, the user can visit the songs' creator via clicking the Artist's name button.

2. SONG PAGE

The purpose of the page is to successfully fulfil one of the features of the application, being that users will be able to express their opinions on songs uploaded by other users via likes and comments. The page's functionality also allows the user to play the song in the music player page and add the song to their user playlist.

The user can use the navigation menu to navigate easily between pages in fewer steps in comparison to continuously pressing the back button multiple times on the android device. The User can view notifications relevant to the user on the Notifications page via the Notifications Button. The user can view his/her profile at any place in the app via the User Profile Button. The Song Name text box displays the name of the song, whose page is being viewed by the user. The Add to Playlist button, allows the user to add the song to his/her personal playlist. The Add Comment Text Box, allows the user to express their opinion of the song via a comment. The Comments List enables the user to get to know other users opinions of the song via reading their comments. The Play Counter/button enables the user to open/play the song in the music player page via a onClick() button, and additionally lets the user know the number of times the song has been played. The Like Counter/Button allows the user to express their opinion of the song via a onPress like button, and additionally displays the number of likes the song has received from other users.

3. EVENT PAGE

The purpose of the page is to successfully fulfil one of the features of the application, being that users will be able to gain information about live music events which are in close proximity to the location provided in the user profile. User's will also be able to post or advertise live music events.

The user can use the navigation menu to navigate easily between pages in fewer steps in comparison to continuously pressing the back button multiple times on the android device.

The User can view notifications relevant to the user on the Notifications page via the Notifications Button. The user can view his/her profile at any place in the app via the User Profile Button. The user can view information about a music event via the events page. The Map using Google Map API shows the user the venue location. The Event Address textbox allows the user to view the address of the event location. The Musical Act textbox allows the user to get information about the artist playing at the event and the artist's music. The Ticket website button redirects the user to a ticket vendor website, where tickets for the event are being sold. The Cost text box provides information to the user about event ticket prices. The Event time text box provides information about event time to the user.

4. Groups/Following Page

The purpose of this screen is to allow the user to view and navigate through a list of all the user's groups and artists he/she is following. Because the functionality is the same we decided to combine the groups and following list into 1 screen.

The user can navigate between the 2 different categories by pressing the following button or groups button. The page will display the number of artists user is following or the number of groups the user is part of. The page will also show the individual artist's name or the individual group's name, and will allow the user to navigate to the individual pages by pressing the artist profile button or group page button. User can also navigate to other main screens by clicking on navigation menu button, notification button, or user profile button.

5. Search Page

The purpose of the results page is to allow users to search for objects in app by text using the search bar and search button. They can filter the results from their search using the categories button and filter search button. The layout of the results and options in the drop down filter menu will change according to which category button is selected.

The 'All' categorey will show the most information showing different info based on the type of object. The 'All' category does the following: shows the thumbnail of the given object (example: album's artwork), shows the objects unique name (example: song's name), shows 2 different types of unique attributes describing the object (example: number of tracks posted by artist and number of users following artist), and clicking anywhere on the object bar will open given object's page. If the user chooses to click a different category than 'All" the results might show less object information for asthetic appeal. For example on the 'Events' results the object bar is replaced with a cirlce button and only shows the event's name. User can also navigate to other main screens by clicking on navigation menu button, notification button, or user profile button.

NON FUNCTIONAL REQUIREMENTS

1. **Scalability**
   Scalability is a system property which describes the ability of the system to appropriately handle increasing and decreasing workloads.

   Scalability is the non functional requirement of the highest or primary priority. As the application is data driven, being that one of the main factors which powers the application is users and user uploaded data. The development team does not provide any significant data to the users. The system is almost a closed loop, wherein the application is sustained by users for users, as data is both uploaded and utilized by users.

   Application scalability will ensure user retention and growth via increasing the amount of data available to the users. More data is available as a result of more users. Therefore, the system should be efficiently scalable in order to better handle large numbers of users and subsequently large amounts of user uploaded data.

2. **Reliability**
   Reliability is a system property which describes the extent to which the software system consistently performs the specified functions without failure.

   Reliability is the non-functional requirement of the second highest or secondary priority. As the system is a closed loop system, wherein users both supply and demand data without much interference from the development team, the flow of data from the backend to the frontend has to be efficient and consistent without disruptions.

   Application reliability will ensure user retention and growth, as users will constantly be supplied with data in the form of songs, albums, user profiles, comments, events etc. Additionally, application reliability will not only ensure consistent and efficient data download and stream to user accounts , but also ensure consistent upload from user accounts to the server.

3. **Performance**
   Performance is a system property which defines how fast a software system or a particular component of the system responds to users' actions under certain workloads.

Performance is the non-functional requirement of the third highest or tertiary priority. As the users both supply and demand data to and from the server, performance of the application is essential, as the application will be consistently pinging the server via http requests. Frequent issuing of multiple http requests in combination with large size of transmitted data may potentially impact the performance of the application during receiving, transmitting and utilizing the data. Thus, the application has to ensure performance in order to prevent high rates of data transmission lag, data loss and crashes.

TABLE RELATIONSHIPS DIAGRAM -Create the picture using MySQLWorkbench)