Custom Application Design Document

- App Name : **DanMack Music App**
- This Document gives an overview of all DanMack Music App functionalities that meet key areas defined in Udacity Nanodegree Android Program rubric.

Objective

- The objective of this project is to develop a music app called “DanMack” that will enable its user to view, play and search songs.

User Flow

- Sign Up Screen : This is a screen for user to get authorized into the app using email and password
- Login Screen : Screen for existing user to login
- Explore Screen : Home screen that consists of the list of top artist’s songs and recommended songs.
- Detail Screen : User can click of a song to view song details
    - In the detail screen user can do the following
        - Play the song
        - Add song to playlist
        - Share song with friends
        - Copy song url

- Playlist screen : User can see a list of added playlist songs and can click on each song to view song details.
- Search Screen : Users can search for songs based on song title or artist name. The search result brings list of songs that matches the inputted text and on click of search button, the song result shows
- User can click on song and then directed to a web view to play song
- Profile screen : User can view registered email and can choose profile image from phone gallery or take a picture with the phone camera when permission is granted

DanMack Music App that meet the rubric

- **Android UI/UX**
    - DanMack app consist of three or more screens
    - The Navigation Controller is used for Fragment-based navigation and intents are utilized for Activity-based navigation
    - An application bundle is built to store data passed between Fragments and Activities.
    - Constraints layout was used appropriately with ids

    - Resources are stored appropriately
    - Data collection was loaded using recyclerview
    - Motion layout was used for animating views

- **Local and Network data**
    - The Application connects to at least 1 external data source using Retrofit
    - Data retrieved from the remote source is held in local models with appropriate data types
    - The application performs work and handles network requests on the appropriate threads to avoid stalling the UI
    - The Application loads remote resources asynchronously using an appropriate library such as **Glide** or other library/component when needed.
    - All requests are performed asynchronously and handled on the appropriate threads
    - Room was used to perform hold remote data locally and was displayed to user even without internet connection
    - SharedPreference is used to hold user settings while they are on session or within the app
    - Data storage operations are performed on the appropriate threads as to not stall the UI thread
    - Data is structured with appropriate data types and scope as required by application functionality

- Android System and hardware requirement
    - Application separates responsibilities amongst classes and structures using the MVVM Pattern
    - Storing and restoring state and information
    - Properly handling lifecycle events in regards to behavior and functionality
    - Handling interaction to and from the application via *Intents*
    - Handling Android Permissions
    - Application utilizes at least 1 hardware component to provide meaningful functionality to the application as a whole (Camera)
    - Permissions to access hardware features are requested at the time of use for the feature
    - Behaviors are accessed only after permissions are granted