

## How to Use this Template

1. Make a copy [ File → Make a copy... ]
2. Rename this file: “**Capstone\_Stage1**”
3. Replace the text in green

## Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone\_Stage1.pdf**”

---

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you’ll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** AungPyae

# Chill with MM Music

## Description

- a Music Player
  - | that stream the songs from SoundCloud
  - | that organize Myanmar Songs for easier access

| that provide music-related user actions such as Making Playlists, Marking Songs as Favorite, Sharing the Song with Friends.

- listen all Myanmar Songs from SoundCloud with Mobile App Experience designed specifically for Myanmar People.
- organize all Myanmar Songs from SoundCloud for easier access to music lovers from Myanmar.
- stream the songs from SoundCloud music database with implementation that compensate the spotty network condition in Myanmar.
- provide user features such as
  - | Sharing the Song in Social Network with Lyrics.
  - | Sending the Song to (specific) special person.
- will include Remote Playback Control Access that user can do the playback controls (play, pause, skip, previous, etc) from Notification & from LockScreen.

## Intended User

- for all Myanmar Song lovers from Myanmar & all over the world.

## Features

### Core App Features

- organize all the Myanmar Songs in SoundCloud with easily-accessible manner.
- provide localized (Myanmar) user experience.
- stream the Song with implementation that compensate the spotty network connectivity of Myanmar.

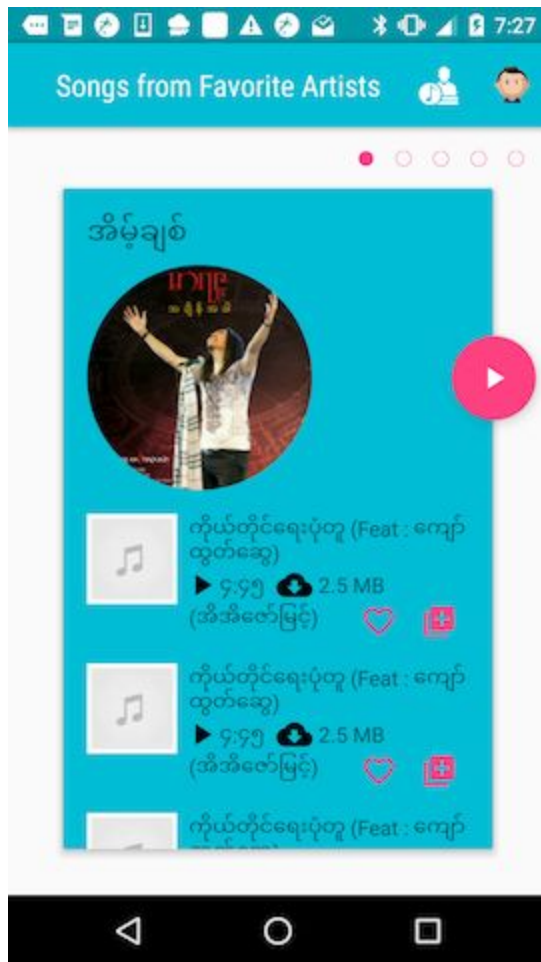
### User Features

- favorite the Song
- share the Song with friends
- share the Song (with Lyric) on social media.

## User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Screen 1



Home screen of the App. Will display a limited amount of songs from each artists who is being picked as favorite. User can swipe each Favorite Artist card to browse the list of favorite artists.

In each song item under each favorite artist, user can play the song by tapping the item. User can put the song into “Favorite Songs” section by tapping heart icon. User can put the song into “Song in Library” section by tapping music library icon.

In each song item under each favorite artist, the song name, the artist(s), the duration of the song, the size to the song & the name of the uploader will be displayed.

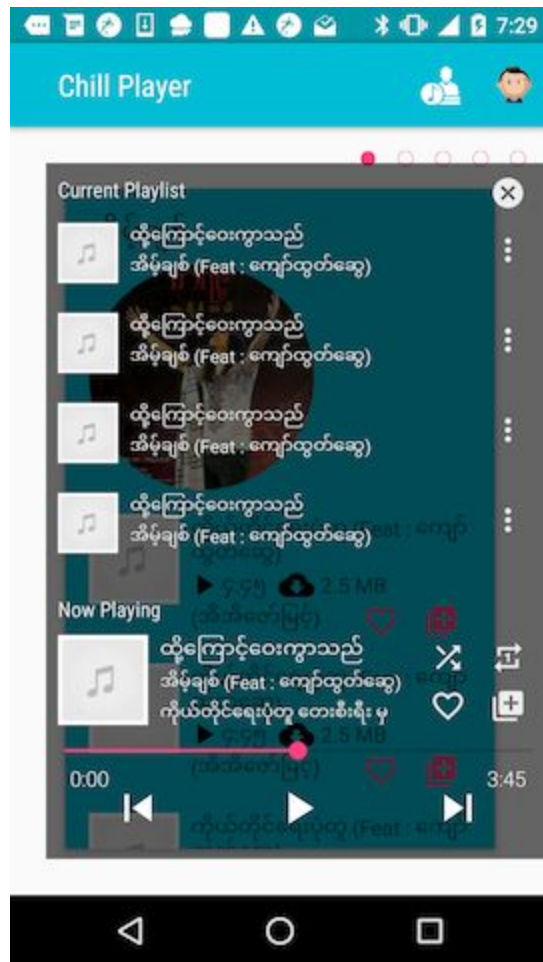
User can add more artists to here by tapping “artist icon” in AppBar. Please see Screen 2 for adding more artists into favorite.

## Screen 2



When the user tap on the “artist icon” of AppBar, this screen dialog for picking favorite artists will be displayed. Artists will be categorized into different groups based on their range & demographics of their fans. Users can select their favorite artists from the any category. After the user has selected any new artists, the list of favorite artists from Home Screen will be refreshed.

### Screen 3

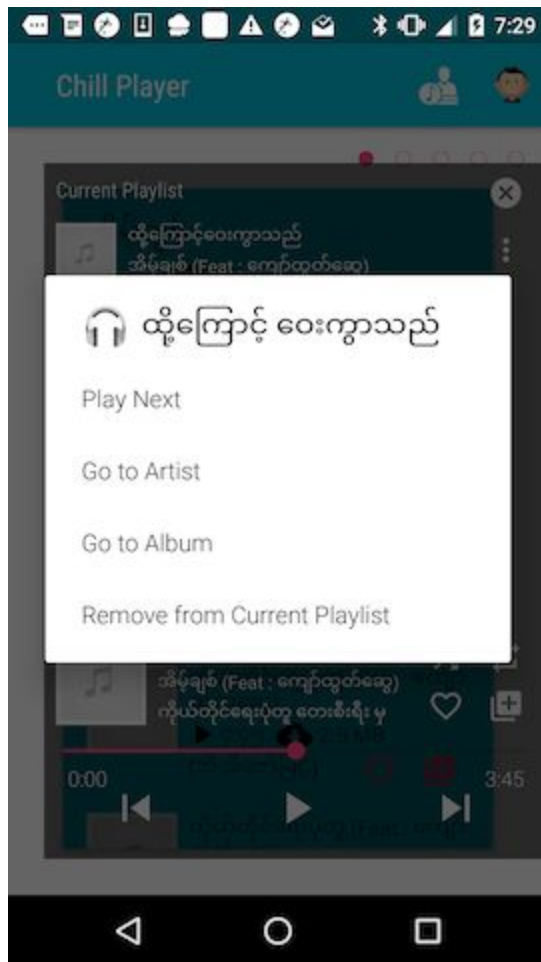


If the user tap any song from the song list of favorite artist or anywhere in the App, the song will be played. If the user tap on “Playing FAB”, the playback controller screen will be displayed.

In the playback controller screen, the currently playing song as well as the list of upcoming songs will be displaying. User can do regular song playback such as play, pause, skip to next, skip to previous and pick the certain point of song with seekbar. The additional playback controls such as “shuffle the current playlist” & “making repeat of current song” are also supported.

User can also add the current playing song into Favorite list or / and Song in Library list.

## Screen 4



In the upcoming songs, there are features such as putting a certain song next to the current playing song, navigating to the certain artist screen, navigating to the certain album screen & removing the song from current playing list - will also be implemented.

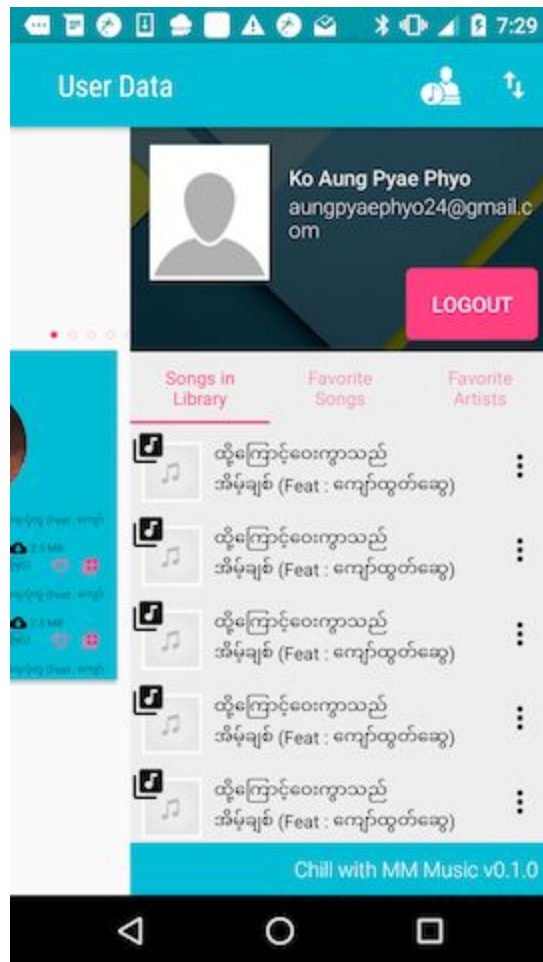
## Screen 5



When the user tap on the dummy user icon at the AppBar, the user data section will be displayed. If the user is not being login, the message encouraging the user to login with Facebook will be displayed. There won't be any registration. If the user connect their facebook account with "MM Music" App, the App will use the user info retrieved from their facebook account.

If the user isn't being login, there won't be any data in "Songs in Library" section & "Favorite Songs" section.

## Screen 6

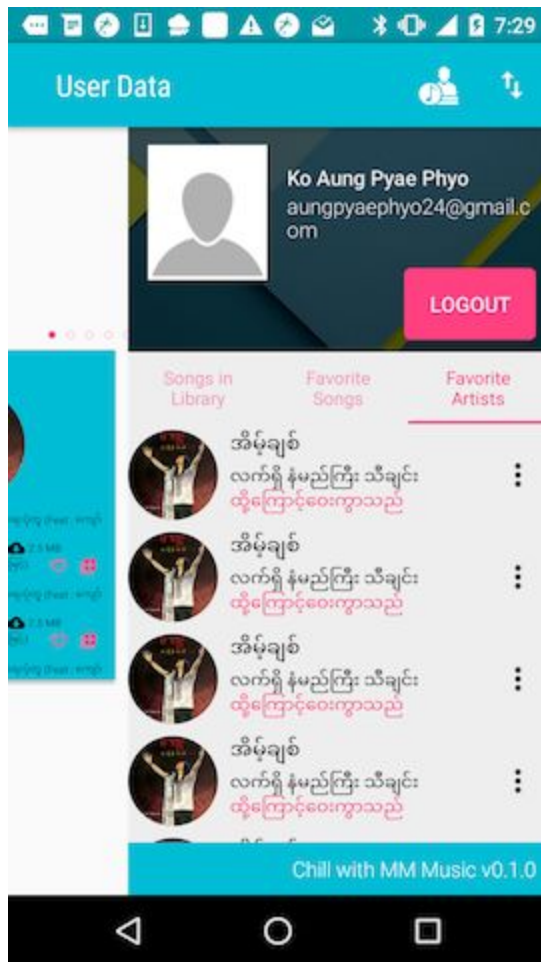


After the user has login the App with their Facebook Account, their user information will be displayed in user info section. User will be able to add the songs into their library & mark the song as favorite. Those songs in library & favorite songs will be displayed user the user info section so that user can easily replay those songs back.

The list of favorite artists will also be displayed as one section under user info section.

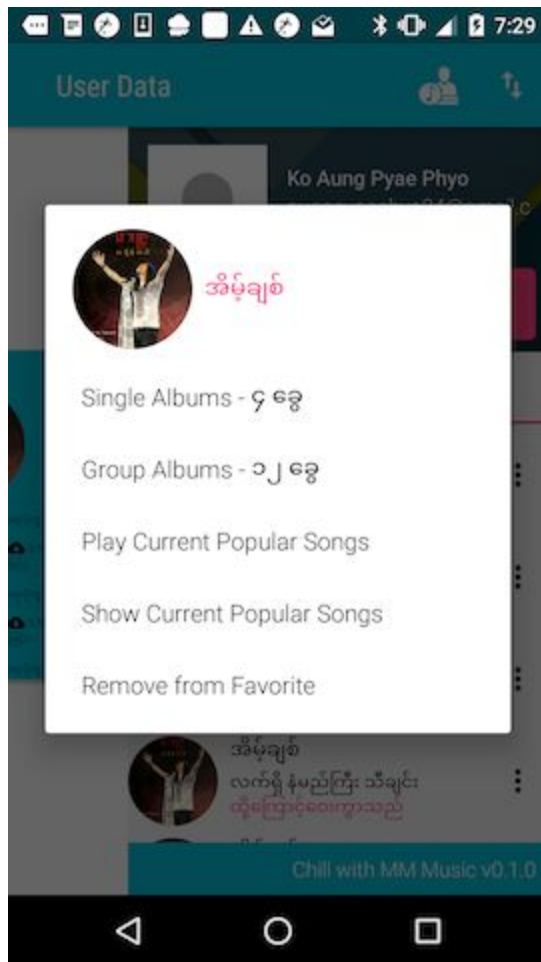


## Screen 7



In the favorite artists section below user info section, the currently most popular song of each artist will also be displayed.

## Screen 8



For each favorite artist, user can navigate to their single albums & group albums, playing the currently most popular songs & navigating to the list of those popular songs. User can also remove the artist from Favorite.

## Key Considerations

### How will your app handle data persistence?

- all the data used in the App (except the actual song file that will be streamed from network) will be available offline.
- all the data used in the App will be store in SQLite with the help of ContentProvider on top.
- the data synchronization between cloud API & local device storage with the help of SyncAdapter to minimize the battery impact.
- all data required for user facing screens will be retrieved from Persistent Layer (ContentProvider) with the help of Loader Pattern.

## Describe any corner cases in the UX.

- if the user is playing a song, the small media player will be showing up at the bottom even if user makes actions that would navigate the screen.

## Describe any libraries you'll be using and share your reasoning for including them.

### - MediaRouter APIs

| <https://developer.android.com/guide/topics/media/mediarouter.html>  
| to allow user playbacks from different interfaces.

### - EventBus from GreenRobot

| <https://github.com/greenrobot/EventBus>  
| to communicate the internal actions inside the App (Persistent Layer -> Model -> UI Layer)

### - ButterKnife from JakeWharton

| <http://jakewharton.github.io/butterknife/>  
| to bind the view component from XML into Java Object.

### - Retrofit 2 from Square

| <https://square.github.io/retrofit/>  
| to make HTTP request to Rest APIs.

### - Glide

| <https://github.com/bumptech/glide>  
| to load images from Network with efficient manner.

### - Crashlytics

| <https://fabric.io/kits/android/crashlytics>  
| to track the App crashes in beta & final releases.

### - Facebook SDK

| <https://developers.facebook.com/docs/android/>  
| Login with Facebook Account & other Facebook related social actions.

### - Google Play Services

| <https://developers.google.com/android/guides/overview>  
| Login with Google Account & other Play Services-related stuffs.

### - Gradle Advanced Build Version

| <https://github.com/moallemi/gradle-advanced-build-version>  
| to automate the build number incrementation & build name generation.

**Describe how you will implement Google Play Services.**

- MediaRouter APIs

| <https://developer.android.com/guide/topics/media/mediarouter.html>  
| to allow user playbacks from different interfaces.

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

### Task 1: Setup Github Repo

- Add Project Descriptions & Mock Screens to ReadMe.
- Clone the Repo & Setup Official GitFlow (master, develop - features, releases, hotfixes) branches.

### Task 2: Setup Android Project

- Refactor the generated class & xml layout files to our own packaging & naming structure.
- Add all the listed dependencies & sync the project.
- Select initial app theme using Material Design Color Palette Generator (<https://www.materialpalette.com/>) and put the theme color in colors.xml.

### Task 3: Implement all the screens & user action flows with mock data

- Create each screen based on the design mock using mock data.
- Setup user action flow between screens just to illustrate the user journey on each user actions.

### Task 4: Hook up the required APIs from SoundCloud to get the necessary data

- Create Data Layer (java object format) based on the json response from SoundCloud API.
- Create Network Layer using Retrofit as Network Client.

- Send the network response from Network Layer to UI Layer using EventBus ~ just for testing screen implementation & user action flow.
- Create Persistent Layer and save the data from network response into Persistent Layer.
- Load the data from Persistent Layer using Loader Pattern from UI Layer.

## Task 5: Implement User Actions in the App

- Implement user account-related actions (Register / Login / Forget-Password / Change-Password)
- Implement "Login-with-Social-Media-Accounts" (Facebook / Google).
- Implement user-related actions (Favorite Song, Share the Song, Create Playlist, Add / Remove Songs from Playlist)

## Task 6: Implement Music Player with Services & Android Media Framework

- Implement Music Service (control the streaming) & Playback Controls using Android Media Framework.
- Implement Small Media Player & Full-Screen Media Player - Hook up with Music Service & Playback Controls.

## Task 7: Integrate Google Analytics

- Create Google Analytics Account in Google Analytics Console & get Tracking ID.
- Set up the App with Tracking ID in the App.
- Implement the logic for sending Screen Names & User Actions from the App to Google Analytics.

## Task 8: Setup Gradle Advanced Build Version

- Set up Gradle Plugin in project-level build.gradle & app-level build.gradle.
- Get the dynamically generated build name & show up somewhere in the App.

## Task 9: Integrate Fabric

- Setup Fabric Account & Add the Project.
- Setup the Project for reporting the crashes.

Add as many tasks as you need to complete your app.

---

**Submission Instructions**

1. After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"