

## How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
  2. Name your document file: “**Capstone\_Stage1**”
  3. Replace the text in green
- 

[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:** ghilbing

# Band A

## Description

With Music Now you can get an amazing experience with your group, this app allows you the following:

- Coordinate and schedule rehearsals.
- Record your part to allow the rest of the group study based on that.
- Define and vote repertoire and make and share your own playlist.
- Differentiate between original versions and covers.

## Intended User

The intended user is anyone who likes music and belong to a group of people, like jazz band, cord band, choir, rock band, etc. to facilitate their work.

## Features


The main features of the app are the following:

- It will be written in java
- Multi Language, for now English and Spanish
- Manages authenticated login
- Saves users' and the band's information and which instruments they play.
- Uses Calendar to arrange rehearsals.
- Keep track of the possible places to rehearse and perform with Google Maps.
- Saves playlists that can be shared to study.
- Notificates in case something changes, rehearsal, concert, playlists, etc.

## 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 Google Drawings, [www.ninjamock.com](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

## Screen 1



**Band A**

Sign in with:



Or

[Forgot?](#)


**Sign In**

Do not have an account?

[Sign Up Now](#)


This screen is to Sign In or Sign Up in case the user does not exist yet. In case Sign in the app goes directly to Screen 3, otherwise goes to Screen 2.


## Screen 2





### Create Account

It is free and hardly takes more than 30 seconds.







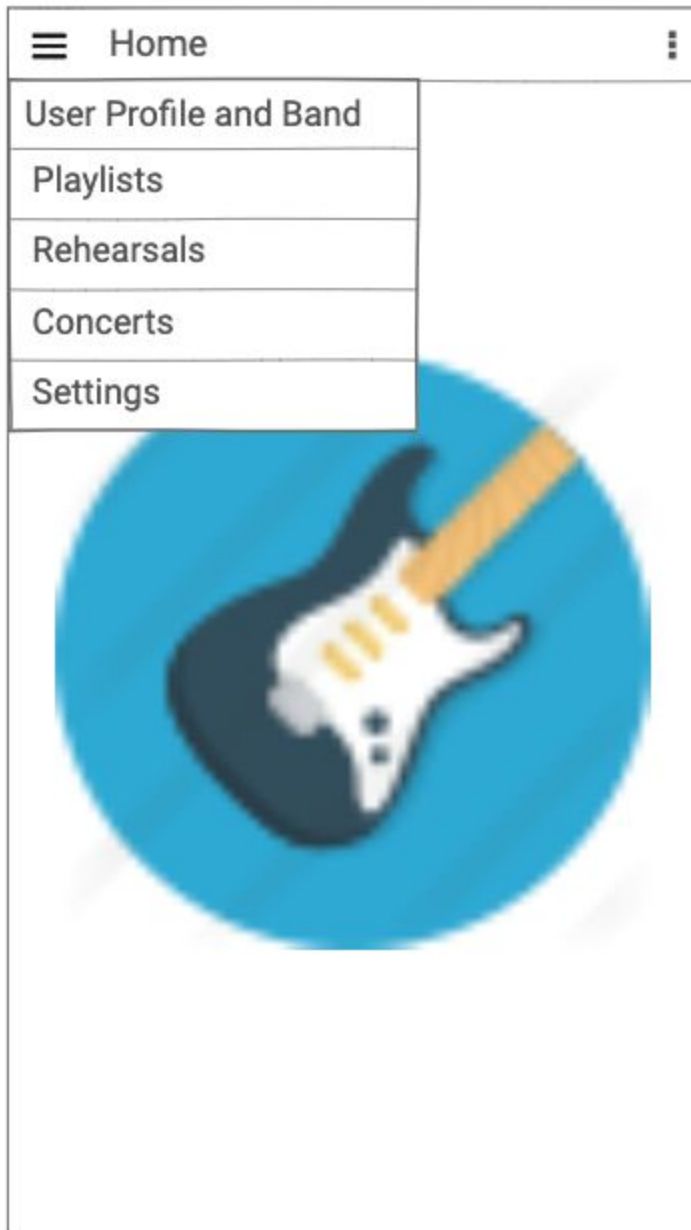


Sign Up

By clicking the Sign Up button, you agree to our  
[Terms & Conditions, and Privacy Policy](#)

Screen for a basic Sign Up.

### Screen 3



Home page where you can access through a Navigation Drawer to different options.

## Screen 4

<

User Profile

Personal Data

Band

Click to attach a picture

User Profile

Name and Lastname

Nickname

Phone number

Address

☒ Do you sing?

☒ Do you compose?

Guitar  
Advanced


Drums  
Beginner

Save

The User and Band profile have a ViewPager with 2 tabs. The first one is for Personal Data, like phone number, nickname, and the instruments that the person plays and the level of experience when clicks the floating button as shown in Screen 5.


## Screen 5

<

 User Profile

Personal Data

Band



Click to attach a picture

User Profile

Name and Lastname

Add new instrument

Instrument

Bass

Cello

Drums

Guitar

Experience

Beginner

Intermediate


Advanced

Cancel

OK

Drums




Beginner



Save

Screen for adding instruments.

## Screen 6

 User Profile

Personal Data

Band

Subscriptions

Music in Spirit  
Band created by @suerichardson

Far Far Away  
Band created by @me

Search Band

Beach Boys

Send Request

Created bands



Far Far Away  
Band created by @me

Create Band


The idea for this screen is to subscribe to an already existing band as a new member or to create a new band. You can delete your subscriptions or the bands you have created, but not the bands created by others.



## Screen 7




User Profile




Personal Data

Band




Click to attach a logo


Create a Band



Band's name

About Us 


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.




Location


Musicians

Ringo Star  
Drums



Paul McCartney  
Bass, Piano, Voice








Save


When you create a band you can put a logo, a name, a short story, and the location. Then you can add musicians. When you click the floating button, it gives you to Screen 8.

## Screen 8

  User Profile 


Personal Data

Band




Click to attach a logo

Create a Band



Band's name

About Us 

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex


Invite or select new Musician

Or:

Cancel

OK




bass, Piano, voice











Save

You can send an invitation to register at the app, or if the musician is already registered, you can search by name and add it.

## Screen 9

  **Playlists** 


<b>Poll</b> Playlist to choose songs	 
<b>For rehearsals in June</b> Songs to be played in June	 
<b>Ready for Concert 1</b> Concert in SFJAZZ	 
<b>For rehearsals in August</b> Songs to be played in August	 

**Playlistname**

Playlist1

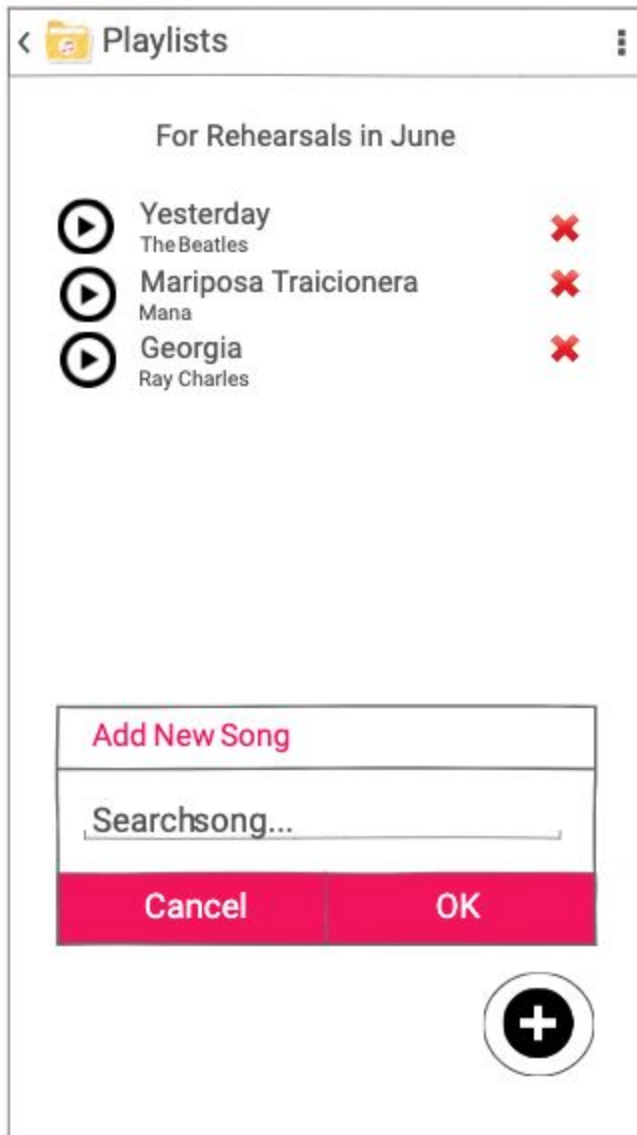
Cancel

OK



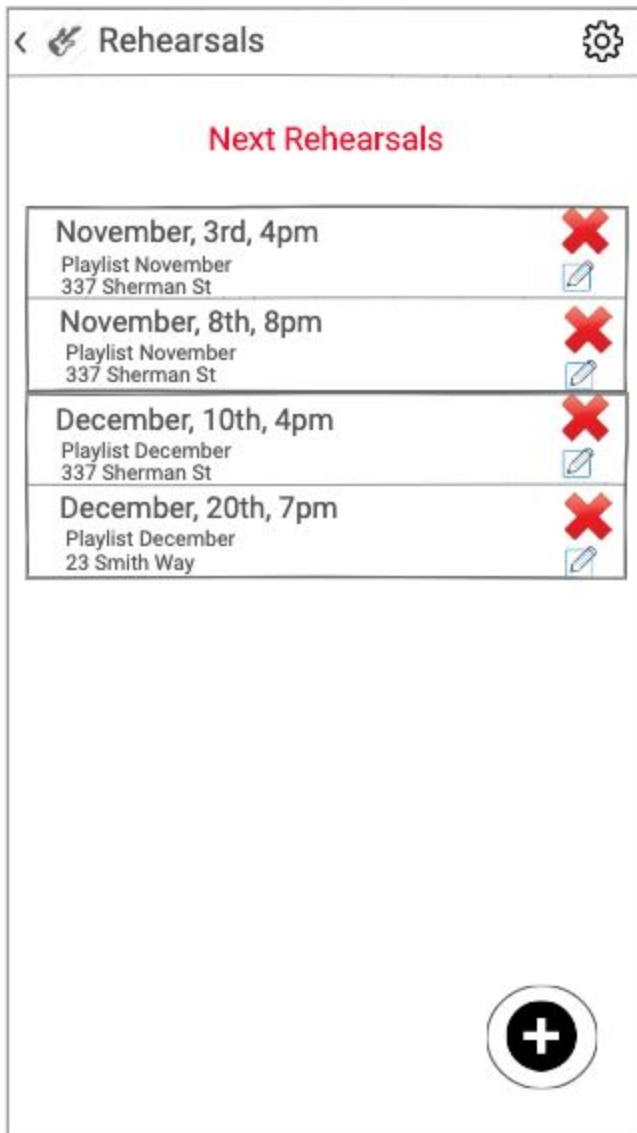
With this screen you can create different playlists. You can have one where you can suggest new songs, another one with the specific songs for a concert, another to let all now which songs are ready, etc.

## Screen 10




Once you choose a playlist, you can start adding songs from Spotify, for example.

## Screen 11




This screen will show the next rehearsals already created and the possibility to generate new ones or update the existing ones. Once you click the Add button, it will go to Screen 12.

## Screen 12


<  Rehearsals 

### New Rehearsal


☒ Open rehearsal?




Date




Time




Playlist



Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna  
aliqua. Ut enim ad minim veniam, quis  
nostrud exercitation ullamco laboris nisi ut



Location



Save

This screen will allow the user to create a new rehearsal, the rehearsal can be opened to the public or closed. Then you can select Date and Time through pickers as shown in Screens 13 and 14 respectively. You also can add some notes, the playlist to rehearse and location with Google Maps API.

## Screen 13

<

Rehearsals

New Rehearsal

☒ Open rehearsal?

Set date

▲

SEP

▼

▲

06

▼

▲

2011

▼

▲

OCT

▼

:

▲

07

▼

▲

2012

▼

▲

NOV

▼

▲

08

▼

▲

2013

▼


Cancel


OK

Save

Screen 14

<


 Rehearsals




New Rehearsal

☒

Open rehearsal?



Date



Time

Set time

▲

7

▼

▲

29

▼

▲

▼

8

:

30

AM

9

:

31

PM

▼

▼

▼

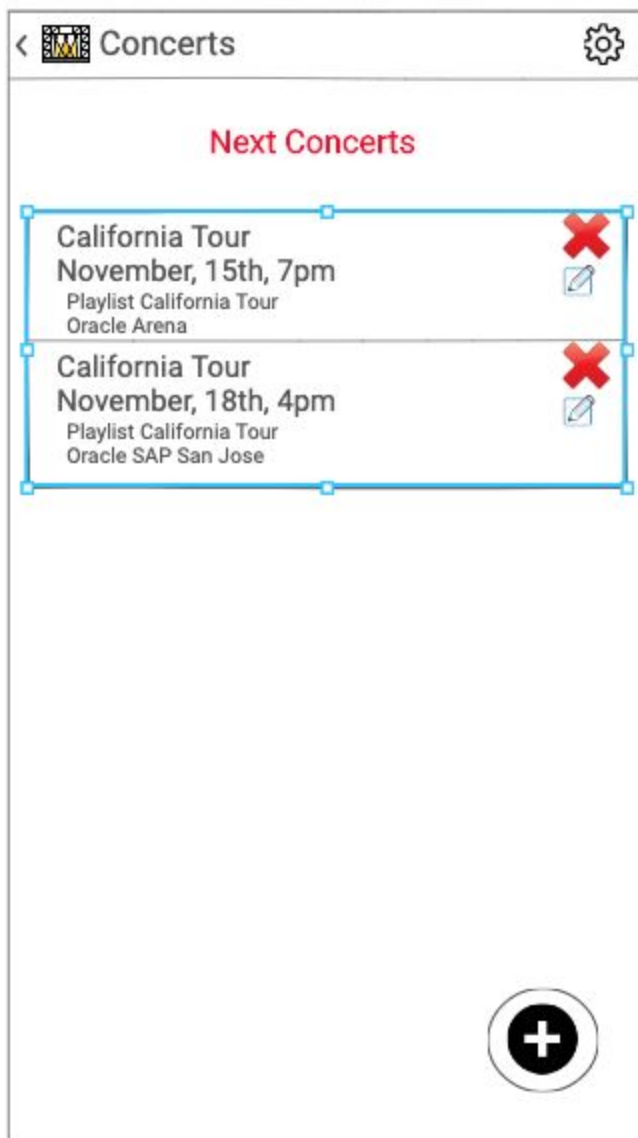
Cancel

OK

Save






## Screen 15




This screen will show the next concerts already created and the possibility to generate new ones or update the existing ones. Once you click the Add button, it will go to Screen 16.


## Screen 16

  Concerts 


### New Concert




Concert Name




Date




Time




Playlist



>Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna  
aliqua. Ut enim ad minim veniam, quis  
nostrud exercitation ullamco laboris nisi ut



Location



Save

This screen will allow the user to create a new concert. Then you can select Date and Time through pickers will be the same as the ones shown in Screens 13 and 14 respectively for Rehearsals. You also can add some notes, the playlist to play and location with Google Maps API.

## Screen 17



The widget of this app, will be very simple. It will show the next events coming and you should select your availability.

## Screen 18



The app will send notifications to users related to CRUD changes in the following aspects:

- Band data (adding or deletion of musicians, change of name, description, etc.)
- Rehearsals or Concert CRUD operations
- Playlists CRUD operations

Every time a user changes something related to the above, all the users subscribed to the band will receive the respective notification. And also the widget info will be updated.

Add as many screens as you need to portray your app's UI flow.

## Key Considerations

### How will your app handle data persistence?

The app will use Firebase Realtime Database (is a cloud-hosted NoSQL database that lets you store and sync data between your users in realtime. When the disk persistence is enabled, the app writes the data locally to the device so the app can maintain state while offline, even if the user or operating system restarts the app.

### Describe any edge or corner cases in the UX.

#### Corner Case 1

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

In this case, to manage MediaPlayer from Activity which is running in background using service, we have to create commands to broadcast when next, previous, and seekbar action occur in Activity. If service is running in the same process then we will use LocalBroadcastManager otherwise BroadcastReceiver for getting events from Activity.

#### Corner Case 2

If the user do not have Spotify installed, then the playlists won't show the play button enabled, and also the user won't be able to add songs to playlists. These users will only see the names of the songs and the author/s. It will be a message suggesting that for better functionality, please install Spotify.

#### Corner Case 3

To avoid create rehearsals and/or concerts with the same Date and Time or they also could be overlapped, before saving the event we will check if the Time and Date is greater than a previous event already saved. Also the user cannot create events with passed dates.

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

I will use the following libraries.

- Picasso, for image handling
- Butterknife, for data binding
- Retrofit, for API connection

**Describe how you will implement Google Play Services or other external services.**

I will use the following services:

- Firebase, for database, storage and notifications
- Google Maps, to set locations for rehearsals and concerts

To implement Firebase in Android, open the Android Studio Tools menu, and then click on the Firebase option in the drop down menu.

A menu appears. It will show all the services that Firebase offers. Choose the desired service and then click on Connect to Firebase. Then add the dependencies.

After that, create a Firebase Project in the Firebase Console. Fill the data. And then add this project to the android app. Also you need to download google-services.json file and place it in the root directory of the android app.

For Google Maps, install the Google Play Services SDK, Create a Google Map project, Get a Google Maps API key and save it into the <string> element in the google\_maps\_api.xml file, and then connect an Android device.

## Next Steps: Required Tasks

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

### Task 1: Project Setup

To start, a new project in java will be created. After this, it will be shared on GitHub to start committing changes. Next steps are:

- Configure libraries and necessary dependencies
- Set the required <uses permission> in Android Manifest
- Get API credentials
- Configure string.xml for multilanguage

### Task 2: Implement UI for Login and Sign Up

Create the logging screen and configure Firebase in the project with OAuth adding the corresponding library in the build.gradle file:

- Build UI for LoginActivity
- Build UI for SetupActivity
- Create the logout option in every Toolbar
- Set also Facebook and Google logging
- Implement Google Play Services

### Task 3: Implement UI for Main Screen and NavigationDrawer

For this screen find a background image that identifies the main function of the app and do the following tasks:

- Create a layout for MainActivity
- Create a Navigation Drawer
- Create Fragments for every option in the NavigationDrawer

## Task 4: Implement UI for User Profile and Band

Create Fragment to update the user profile data

- Create layout
- Create a ViewPager to separate Personal Data from Band
- Create a list of instruments to choose
- Create Fragments for ViewPager
- Connect everything with Firebase database and Firebase storage to save all the data
- The user can subscribe to an existing band or create a new one
- Once the band is created, the creator can invite new musicians to join the band

## Task 5: Implement UI for Playlists

Register the application with Spotify API, get credentials to the get the song's URI to be able to listen the song to be played:

- Create layout
- Connect with Spotify API with Retrofit
- Create a playlist and when the Add button is clicked a search dialog will appear to find the song to be added
- The playlists can be updated, deleted and created
- Save everything in Firebase

## Task 6: Implement UI for Rehearsals

The creation of rehearsals will utilize DatePicker and TimePicker to create a new Rehearsal, Set Google Map into Gradle.

- Create layout
- Add Pickers
- Set place with Google Maps
- The rehearsals can be updated, deleted and created
- Set everything in Firebase



## Task 7: Implement UI for Concerts

It will be the same structure from Rehearsals and the same functionalities.

## Task 8: Create a Widget to see Rehearsals and Concerts

Create a Widget that will be able to show what are the next Rehearsals and Concerts and the possibility to set if the musician will make it or not or maybe.

## Task 9: Create UI for Settings

The idea with this screen is the following:

- activate accessibility options
- Set if the profile will be public or not
- Set if the user like to receive notifications

## Task 10: Configure notifications

The notifications will be a useful tool that will let the user know the following:

- Any changes in rehearsals
- Any changes in concerts
- Any changes in playlists
- Any changes in band configuration

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

---

### Submission Instructions

- After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
  - Make sure the PDF is named **"Capstone\_Stage1.pdf"**
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it's named “**Capstone\_Stage1.pdf**”