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**”

GitHub Username: jgodort

GentleCast

Description

Are you tired to manage your podcast subscriptions on a poor style interface?? Tired to search a handsome interface through a sea of differents apps?? Then, you need to try GentleCast, a beautiful way to enjoy your favourites podcasts. Search and manage new podcasts through a Material Design UI that allows you to stay update of new episodes.

Intended User

For those who love to listen the high quality content provided by the podcast community and want to stay update with the latest releases in their subscriptions.

Features

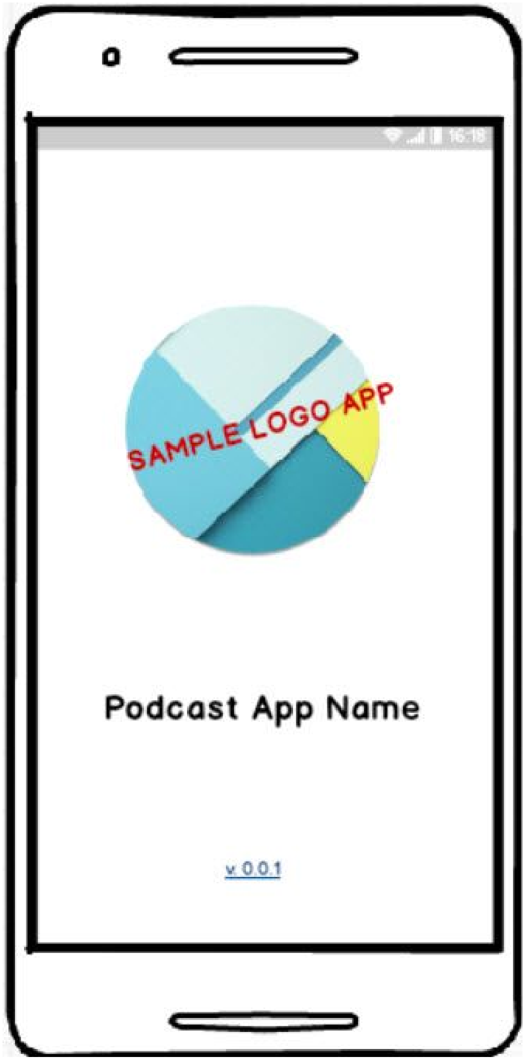
List the main features:

- Subscribe to your favorites podcasts.
- Check the most popular podcast, trending content, etc...
- Discover new podcasts and check the related ones based on your preferences.
- Do not you know where to start?? Try a random podcast episode, maybe you'll fall in love with the great content you will hear...

User Interface Mocks

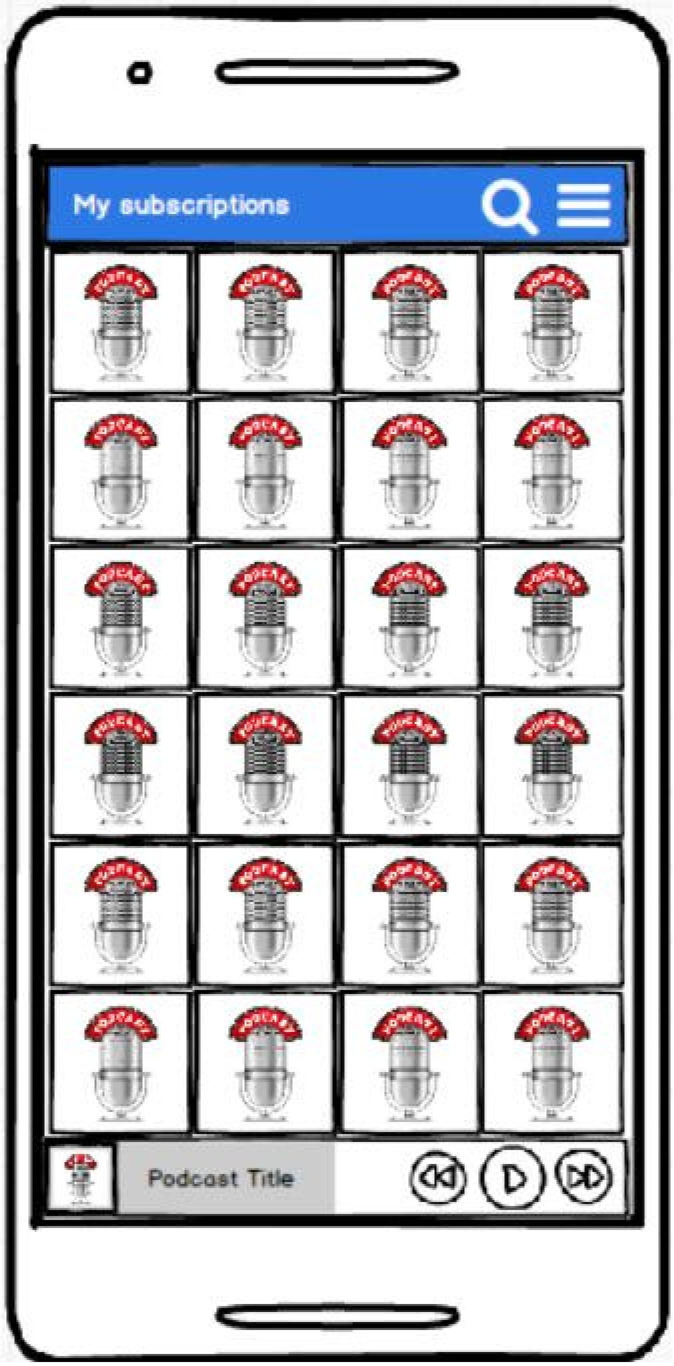
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Branded Launch Screen



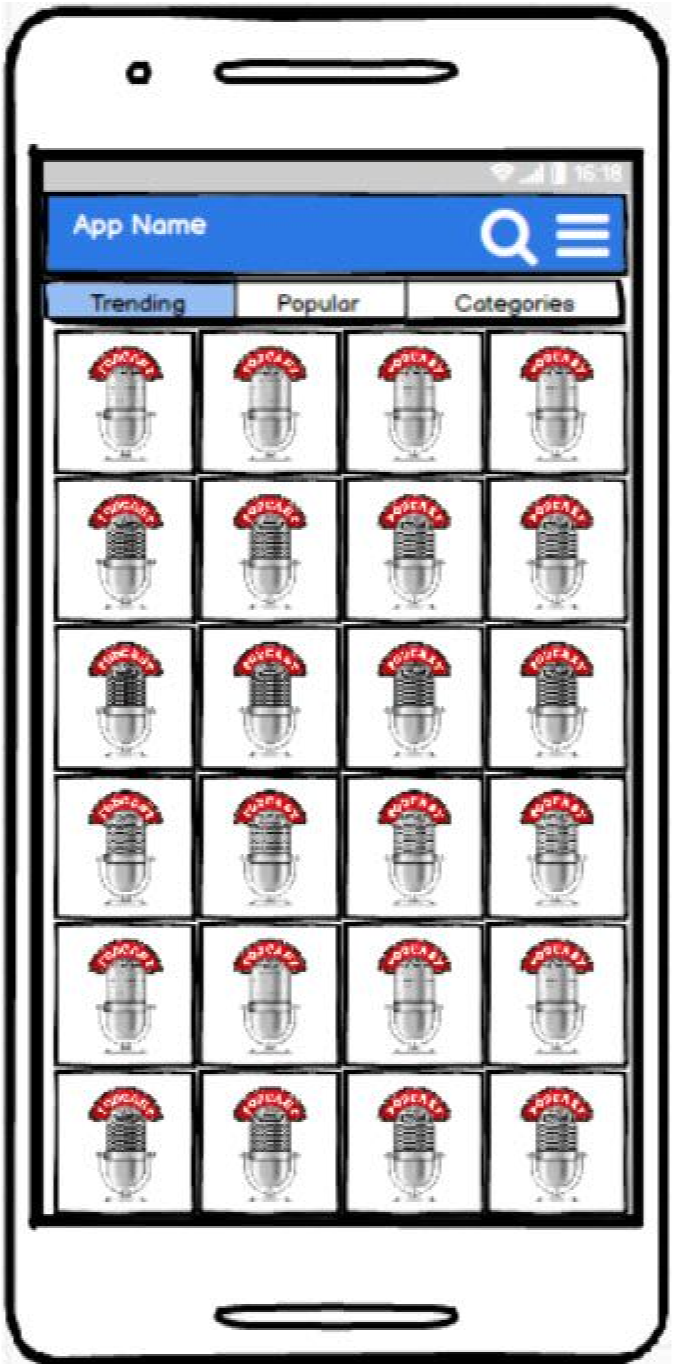
A Branded Screen of the App.

My Subscriptions



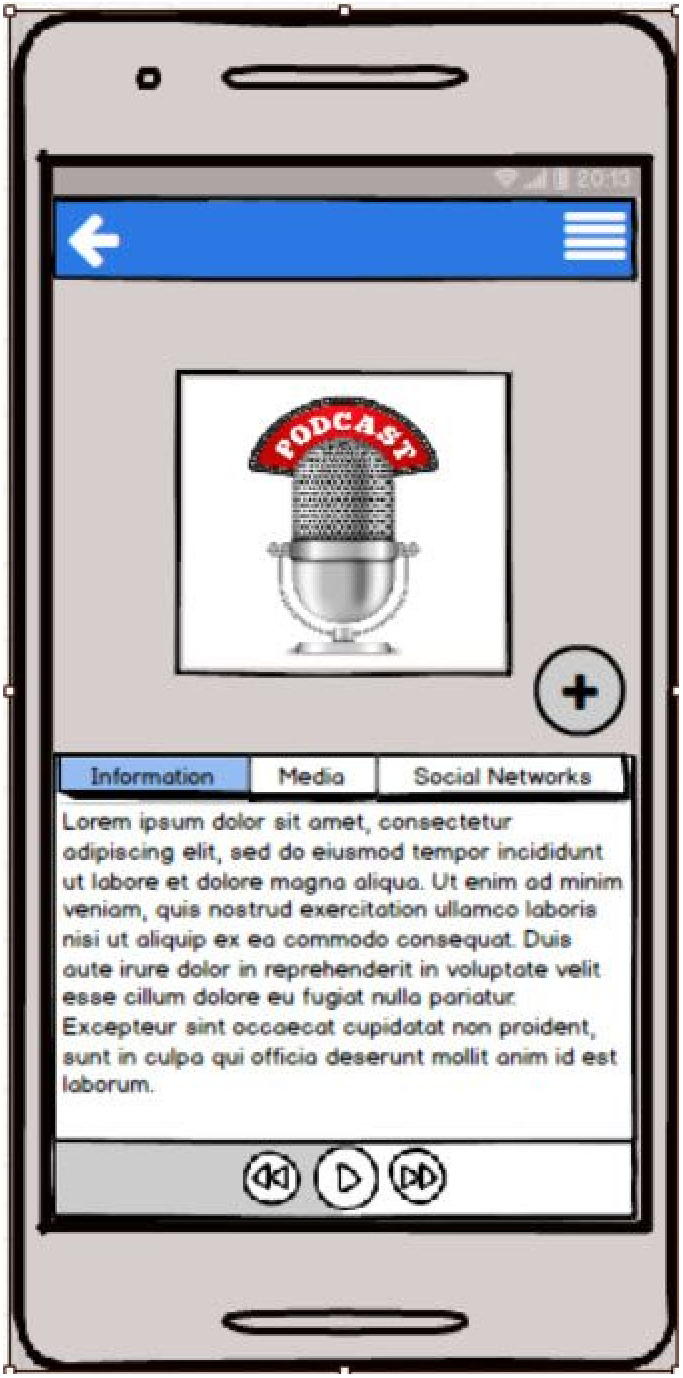
The main Activity shows your favourites podcasts subscription and gives you the possibility to manage them.

Explore



On this Activity you can navigate between tabs and explore the most trending podcast, the new launch ones and also you can find your favourite topics by categories.

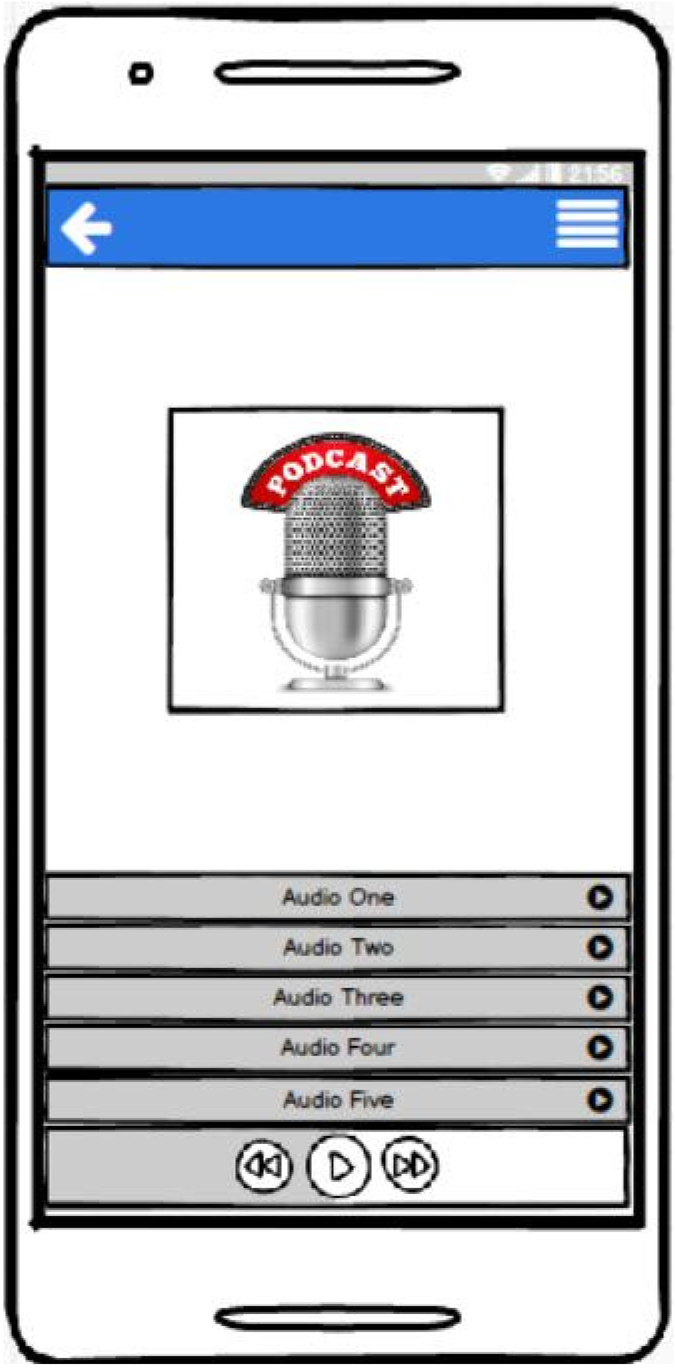
Podcast Detail Subscribe



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 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.

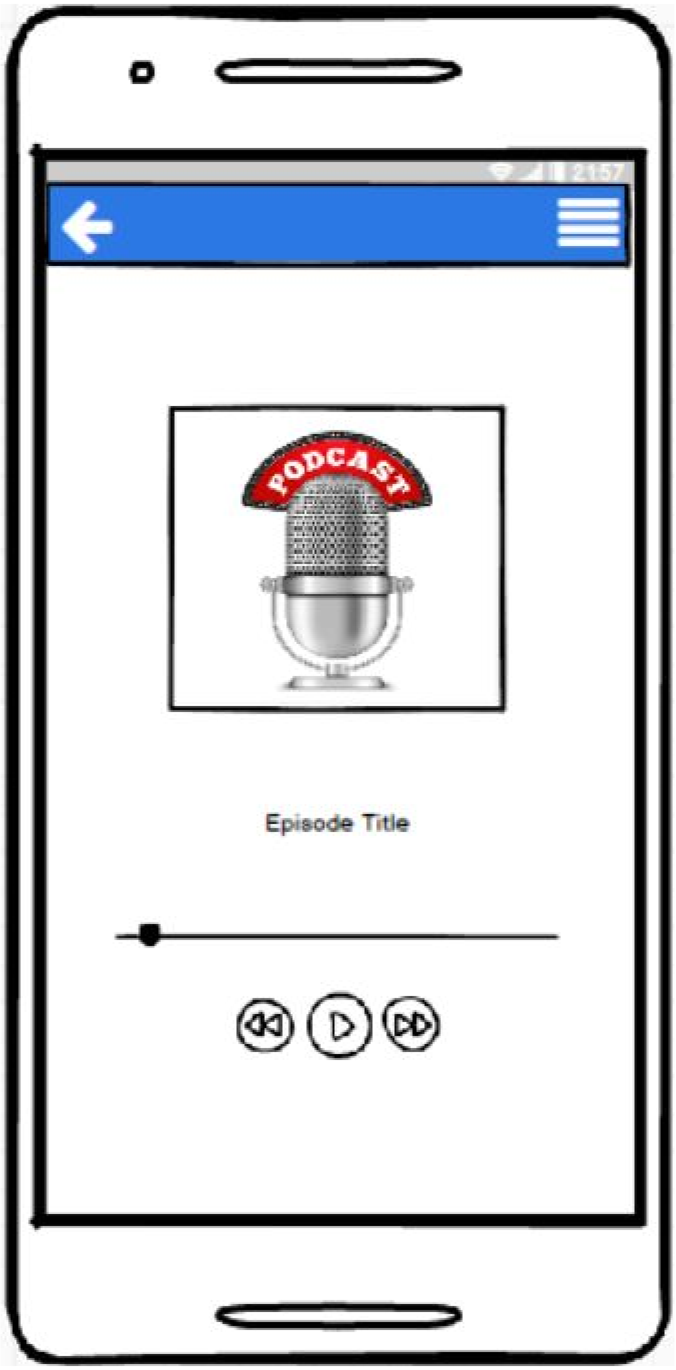
A detailed view of the podcast and it's information, you can read a brief introduction of the topics discussed on it. Also you can check publications on social networks by the hosts of the shows.

Activity Podcast Detail



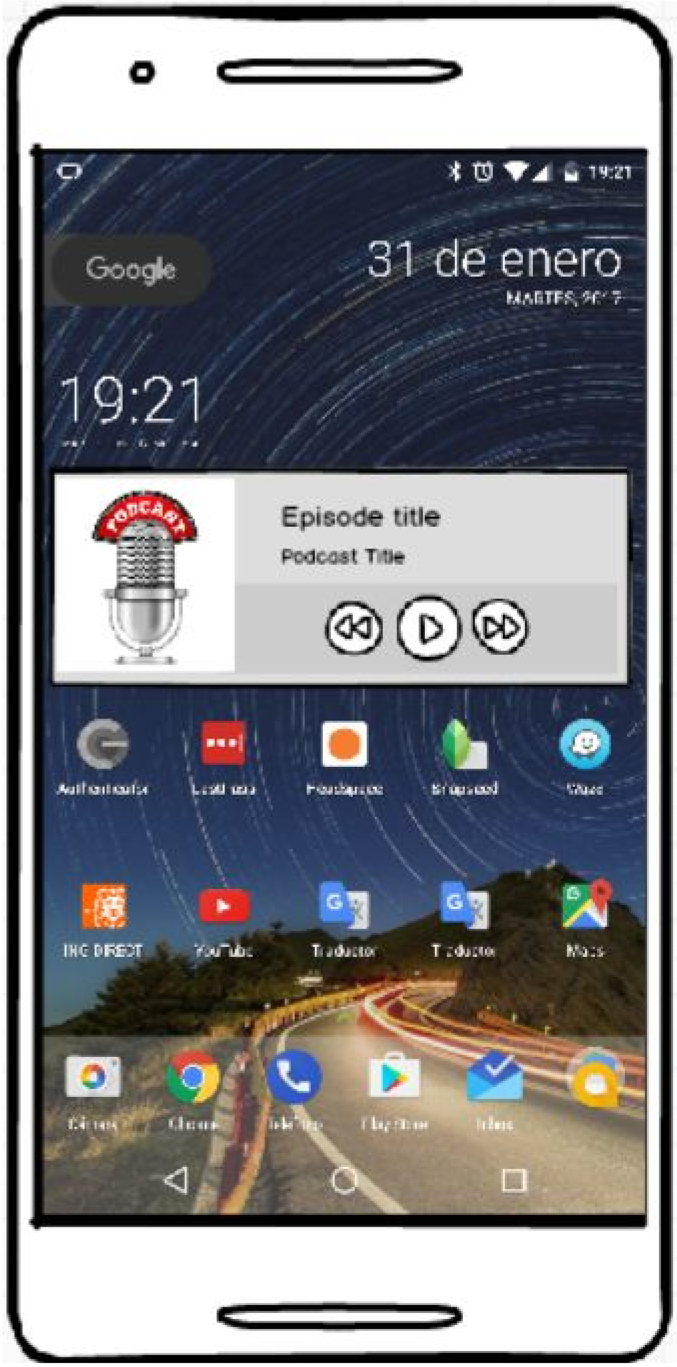
A detail view of the podcast with the latest episodes and recents comments on the social networks.

Now Playing Activity.



A view of the now playing podcast with a set of controls to manage the playing and the progress of the audio.

Widget



Key Considerations

How will your app handle data persistence?

I´ll use a mySqlite device database to storage data information about subscription podcast of the user and their preferences. The database will be accessible through a custom ContentProvider to ensure a good performance.

Describe any corner cases in the UX.

A corner case will be how does the user return to a Now Playing screen in a media player if they hit the back button.

Another corner case will be when the user lost the internet connection and tries to download a new episode.

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

- Glide: For image download and caching.
- Butterknife: For view binding.
- Retrofit: For HTTP connections.
- Dagger: For dependencies Injection.
- Mockito: For testing mocks.
- Stetho: For Development purpose (database view, http request efficiency, etc...)
- Firebase UI Auth: For login and signup process.

Describe how you will implement Google Play Services.

Firebase Auth

I'll use Firebase Auth module to manage the user Login and signup process. Also for future release this allow me to manage the device synchronization.

Firebase Realtime Database

I'll use Firebase Realtime Database to manage the user subscriptions with the target of storage the minimal data about subscriptions and their preferences about the podcast. For future releases, this information will help me to manage the device data synchronization between devices.

Firebase Crash Report

I'll use FIrebase Crash Report to manage the information about unhandled exceptions.

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: Project Setup

- Create a new empty Android Studio Project for Gentlecast
- Create a new CVS repository for the project and configure Readme and .gitignore files.
- Configure Audiosear.ch API: get an api_key from the service.
- Analyze the API Documentation to determine which methods fits with the app requirements.
- Configure the api_key in the gradle files.

Task 2: Implement Model API Client for Audiosear.ch

- Create the model entities to get information from the API service.
- Create the API client Interface.

Task 3: Implement Audiosear.ch Data API Integration.

- Implement the catalog of services to retrieving information from the API.

Task 4: Implement Getting Data from the API.

- The app will implement search by the user through AsyncTask.
- The app will implement a JobDispacher to stay update from new episodes and notify the user.

Task 5: Implement UI for Each Activity and Fragment

- Create UI for the Branded Launch Screen.
- Create UI for the MySubscriptions Activity.
- Create UI for the Explore Activity.
- Create UI for the Podcast Detail Subscribe Activity.
- Create UI for the Podcast Detail Activity.

- Create UI for the Now Playing Activity.

Task 6: Build data persistence support

- Implement all classes needed to handle data persistence: Content Provider, Cursor Adapters, etc.

Task 7: Implement Firebase Integration.

- Implement Firebase Auth integration to manage the user accounts.
- Implements Firebase Realtime Database integration to storage the user subscriptions.
- Implements Firebase Crash Report integration to obtain information about unhandled errors.

Task 8: Implement Accessibility.

- Configure support to RTL users.
- Configure multi-language support (English and Spanish on the early stages).
- Configure disabilities aids to the users (Content Descriptions).

Task 9: Implement Gentlecast Now Playing widget.

- A handsome widget to manage the Now Playing Podcast.

Task 10: Signing and Building.

- Configure app signing process.
- Generate the apk through the “Install Release” task.

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**”