

## 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:** [GilbertoHdz](#)

# GCatCast: The most place for podcast

## Description

Do you like the podcast? Here you can find all list of popular and best podcast, Free!

## Intended User

For all user who like listen podcast and discover new channel of interest.

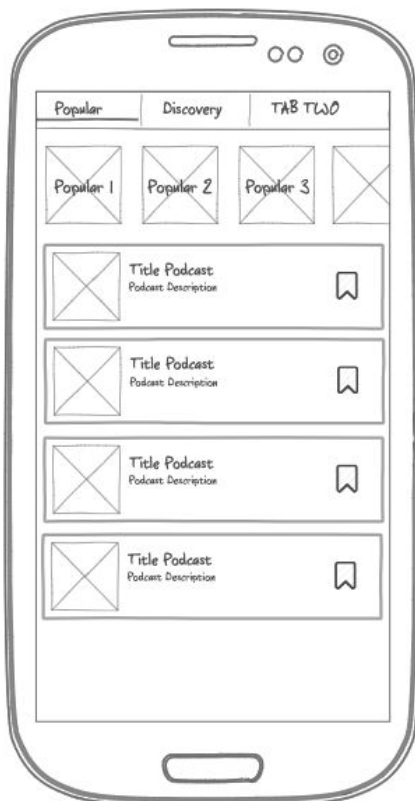
## Features

- Saves user information
- Full network access
- Receive data from Internet
- Storage Space
- Location
- Write External Storage
- Read External Storage

## User Interface Mocks

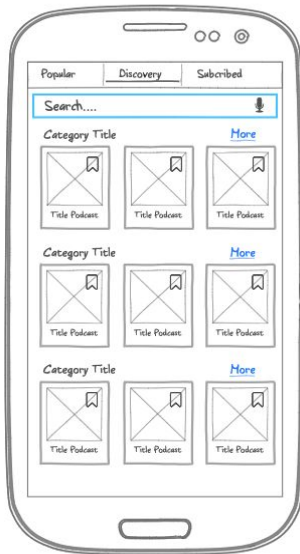
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, [www.ninjamock.com](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

### Screen 1: Popular



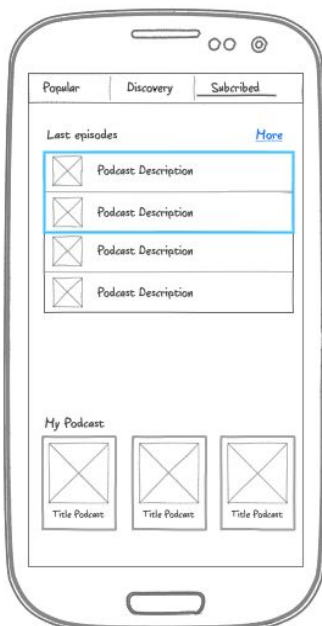
Description: Popular tab is where show the most podcast the user can listen one of there

## Screen 2: Discovery



Description: Discovery is where group the podcast by category and the user can search a specific category.

## Screen 3: Subscribed



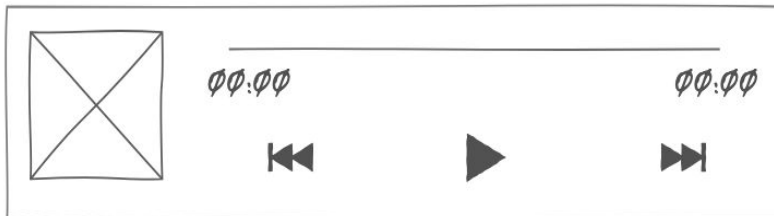
Description: Subscribed is where have the podcast saved by the user and show last podcast updated by the podcast.

## Screen 4: Playlist



Description: Player is where have the podcast is reproduced in full screen, we can save the current podcast or we can share it, also we can see all list of reproduction of the podcast.

## Screen 4: Widget



Description: Widget here we will provide a last podcast listened or proposal podcast to listen.

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

## Key Considerations

### How will your app handle data persistence?

All podcast will be consumed from free API services about of Podcast.

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

When the user interacts with the list of podcast, can select an item where the Full screen player will be displayed (Screen 4) and the user can reproduce the podcast.

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

- Gradle build 3.5.3
- Android Studio 3.6.1
- ExoPlayer 2.11.3
- Glide 4.11.0
- AndroidX 1.1.0
- AsyncTask for pull data from web services
- Services for background task
- Widget with the latest popular podcast
- RoomDB 1.1.1
- LiveData 1.1.1
- Lifecycle 1.1.1
- App is written solely in the Java Programming Language
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- App support English and Spanish language
- App keeps on background the current playlist listened with service job.
- The app navigation should with tap navs and one full screen with player reproducor with native android back to app navigation.

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

- Ads Services
- Location
- Analytics

## 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

- Configure libraries
- Configure Dependency Injection
- Configure Styles, Fonts and Colors
- Configure API services
- Configure Build Variant

We need to include all libraries for the project also we need to configure the flavors dimension to avoid this configuration after, the design configuration will be with a separate module..

### Task 2: Implement UI for Each Screen

- Build ViewPager for each screen
- Build UI for Popular
- Build UI for Discovery
- Build UI for Subscribed
- Build UI for Playlist
- Build UI Components

### Task 3: Implement Logic for request data

- Create RecyclerView for Popular categories
- Create RecyclerView for Discovery
- Implement MVP pattern for request to services
- Implement MVP for Popular and Discovery
- Create a Handler errors for common use case.

### Task 4: Implement logic for reproduce playlist item

- Create layout for Playlist screen
- Implement ExoPlayer for reproduce Podcast
- Implement MVP for Playlist
- Implement Service for reproduce playlist in background

### Task 5: Implement logic for Subscribed

- Create layout for subscribe screen
- Implement MVP for save data
- Implement Database logic for persist data

- Implement logic for download podcast

### Task 6: Implement Widget

- Create layouts for widget to provide relevant information to the user
- Implement logic for load widget with existing data persisted from DB

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**"