

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: [acasadoquijada](#)

Podcastfy

Description TODO

[App aimed to the podcast lovers!](#)

[Write a brief summary of what your app does. What problem does your app solve?](#)

[Not sure how to write a good description? Search 5-star apps on the Play Store for inspiration.](#)

Intended User TODO

Who is your intended user? (For example, is this an app for dog owners? Families? Students? Travelers?)

Podcast lovers, people want to have fun, learn new things

Features TODO

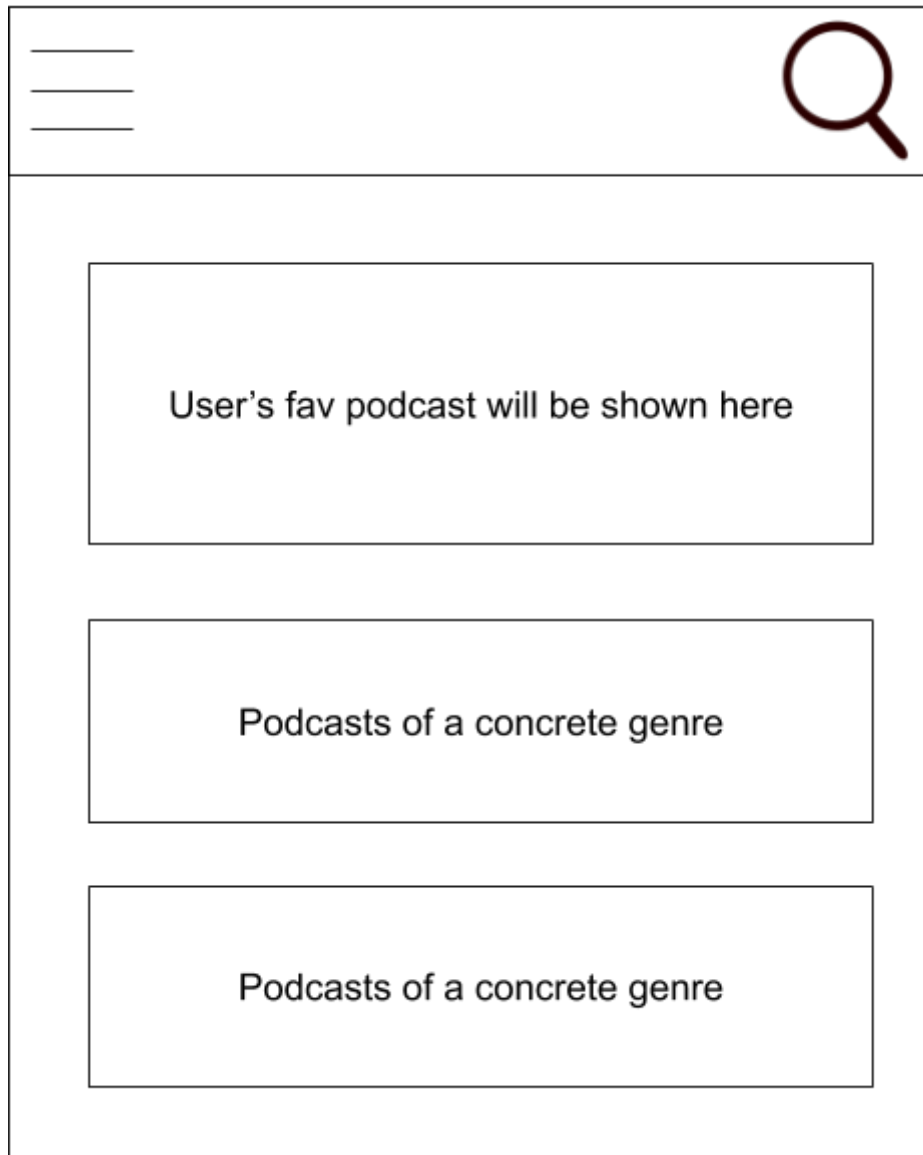
List the main features of your app. For example:

- Saves information
 - Takes pictures
 - Other features
-
- Listen a lot of podcast
 - Save your fav podcast to easily listen to them
 - Share your podcast with your friends
 - Play on foreground (notification)

User Interface Mocks (Work in progress)

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, Paper by 53, Photoshop or Balsamiq.

Screen 1: MainActivity



This activity will contain an AppBar with a search button, besides a navigation drawer icon will be shown in order to let the user know they can open it by sliding from left to right.

By clicking in the search button, a new Activity (searchActivity) will be launched. See more details about it below.

The navigation drawer will contain at least 3 elements:

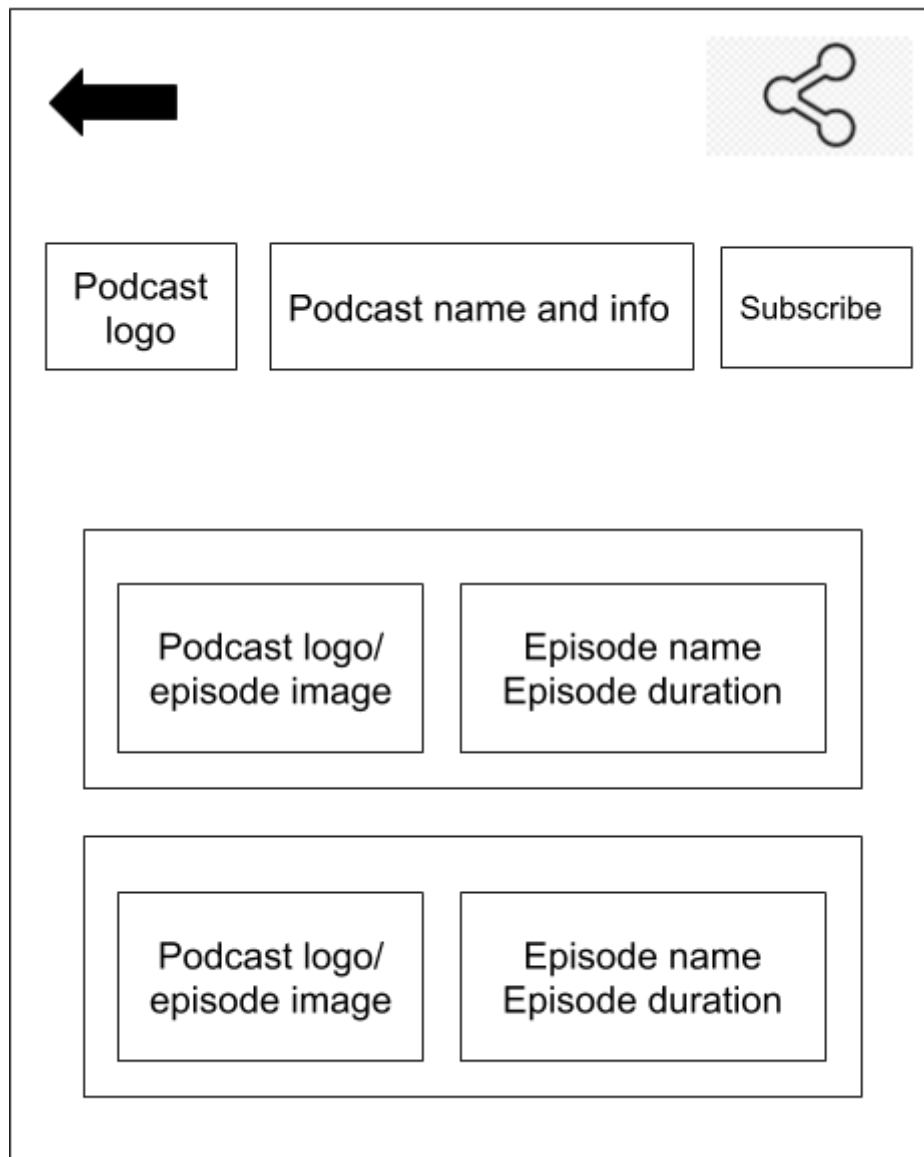
- How to use
- Manage your fav podcast
- About

Below the AppBar there will be shown the user's favorites podcast (Master Detail Flow) and relevant podcast sorted by genres (Master Detail Flow) which the user will be able to

navigate by scrolling horizontally (to see more fav podcast o genre-specific) and vertically to see more genres. Surfaces will be used following the Material Design specification.

When a podcast is clicked by the user, the PodcastActivity is launched.

Screen 2: PodcastActivity

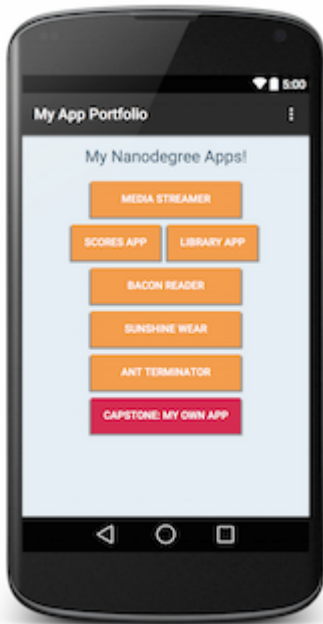


In an AppBar the image of the Podcast along with its name (and maybe a bit info) will be shown. Besides, there will be a share button to share the podcast with friends.

Below the AppBar, the episodes available for the podcast will be listed. When the user click on one episode, PodcastDetailActivity is launched.

There will be an option to set an unset the podcast as favorite

Screen 3: PodcastDetailActivity



In this activity the AppBar will have a back button to go to PodcastActivity and a share button to share the episode.

Besides, the podcast logo and episode description will be center in the screen. Below then an exoplayer will reproduce the episode. The user will be able to interact and perform the usual actions with it (pause, skip, go to next..).

In addition, a notification will be shown if the user leaves the application, allowing it to interact with it the same way as the exoplayer.

Screen 4: SearchActivity



This activity will have a search bar where the user can search Podcast. Once the user introduce the name and click on the search button. The query result will be shown below the search activity.

The user will be able to click on the result and launch PodcastActivity or PodcastDetailActivity depending on the result

Key Considerations

How will your app handle data persistence?

Describe how your app will handle data. (For example, will you build a Content Provider or use Firebase Realtime Database?)

The Data will be stored in the device using Room

Describe any edge or corner cases in the UX.

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

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

For example, Picasso or Glide to handle the loading and caching of images.

Picasso for the image handling

Spotify Web API for Android to interact with the Spotify web API.

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

Describe which Services you will use and how.

TODO

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

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Configure libraries
- Something else

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for something else

Task 3: Your Next Task

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

Task 4: Your Next Task

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

Task 5: Your Next Task

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

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**”