

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: willhwongwork

PODEMO

Description

A simple podcast app, help user discover and subscribe to podcasts

Playlists management, collect whatever episodes into a group

Download the episodes and listen to them without network connection

Intended User

Podcast lovers

Features

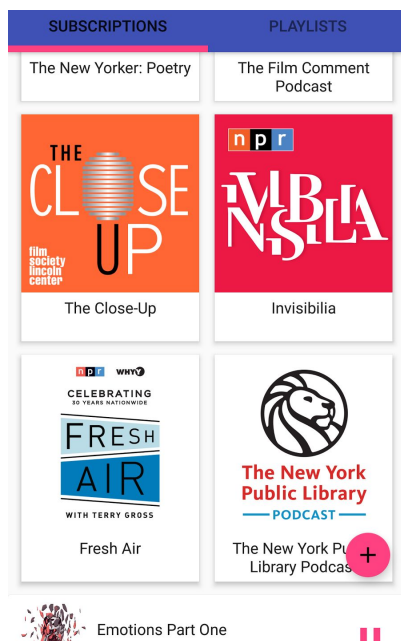
List the main features of your app.

- Podcast playback
- Search for podcasts with keywords
- Subscribe to a channel
- Create playlists and save episodes into playlists
- Download episodes for offline listening

User Interface Mocks

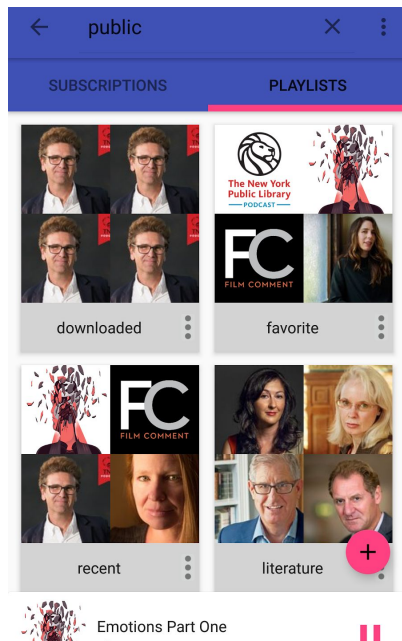
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



The main screen of the app, includes this subscription tab.

Screen 2



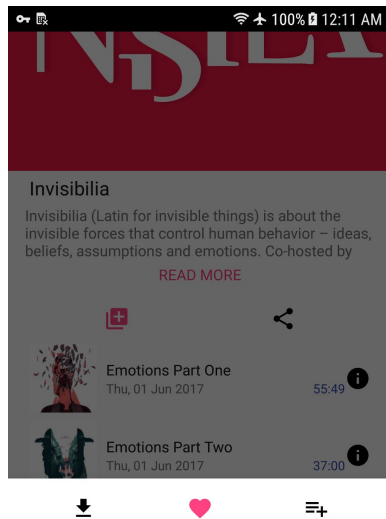
The playlists tab

Screen 3



The detailed view of the channel

Screen 4

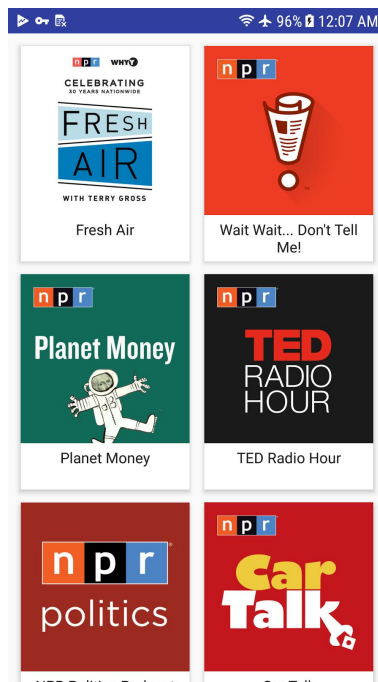


Emotions Part One

We offer you a truly mind-blowing alternative explanation for how an emotion gets made. And we do it through a bizarre lawsuit, in which a child dies in a car accident, and the child's parents get sued by the man driving the other car.

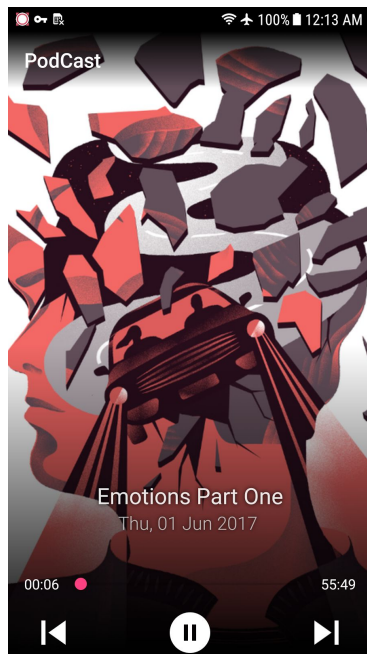
A bottom sheet for user to add episode to playlists

Screen 5



Search podcasts

Screen 6



Playback control

Key Considerations

How will your app handle data persistence?

The app will have a content provider of its own for storing data like channels, episodes and playlists

Describe any corner cases in the UX.

There is a small playback bar at the bottom when there is audio playing in the background, when the user taps on it, it brings back the main playback screen.

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

- `Com.squareup.picasso:picasso:2.5.2` for downloading images
- `Org.jsoup:jsoup:1.10.2` for getting plain text out of some xml elements values
- `Org.apache.commons:commons-lang3:3.5` use the `FilenameUtils` to get the name of the downloaded file out of the url.
- `Com.facebook.stetho:stetho:1.5.0` for debugging

Describe how you will implement Google Play Services.

Firestore Analytics to collect usage and behavior data for the app
Admob for displaying ads

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

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.

- Find a podcast search engine and examine its returned search data
- Examine the RSS feeds of the podcasts

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity, the subscription tab and playlist tab
- Build UI for search results activity
- Build UI for detailed channel activity and a bottomsheet for adding episodes to playlists and displaying information about the episodes
- Build the full screen playback control activity and bottom playback bar

Task 3: Implement playback logic using MediaBrowserService

Use MediaBrowserService to separate playback UI control and media playback.

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

Task 4: Implement data storage with content provider and file storage

- Create a data storage model that stores data in sqlite database and manipulate the database through a customized content provider
- Store downloaded audio file using external storage

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**”