

How to Use this Template

- Make a copy [File → Make a copy...]
- Rename this file: “**Capstone Stage1**”
- Replace the text [in green](#)

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
- 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**”

[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.](#)

[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: [georgi-neykov-hub](#)

PodcastPortal

Description

[Discover and subscribe to interesting podcasts, download them for offline playback and get notified for updates on your subscriptions.](#)

Intended User

Users who would like to listen/watch podcasts on an Android-based mobile device

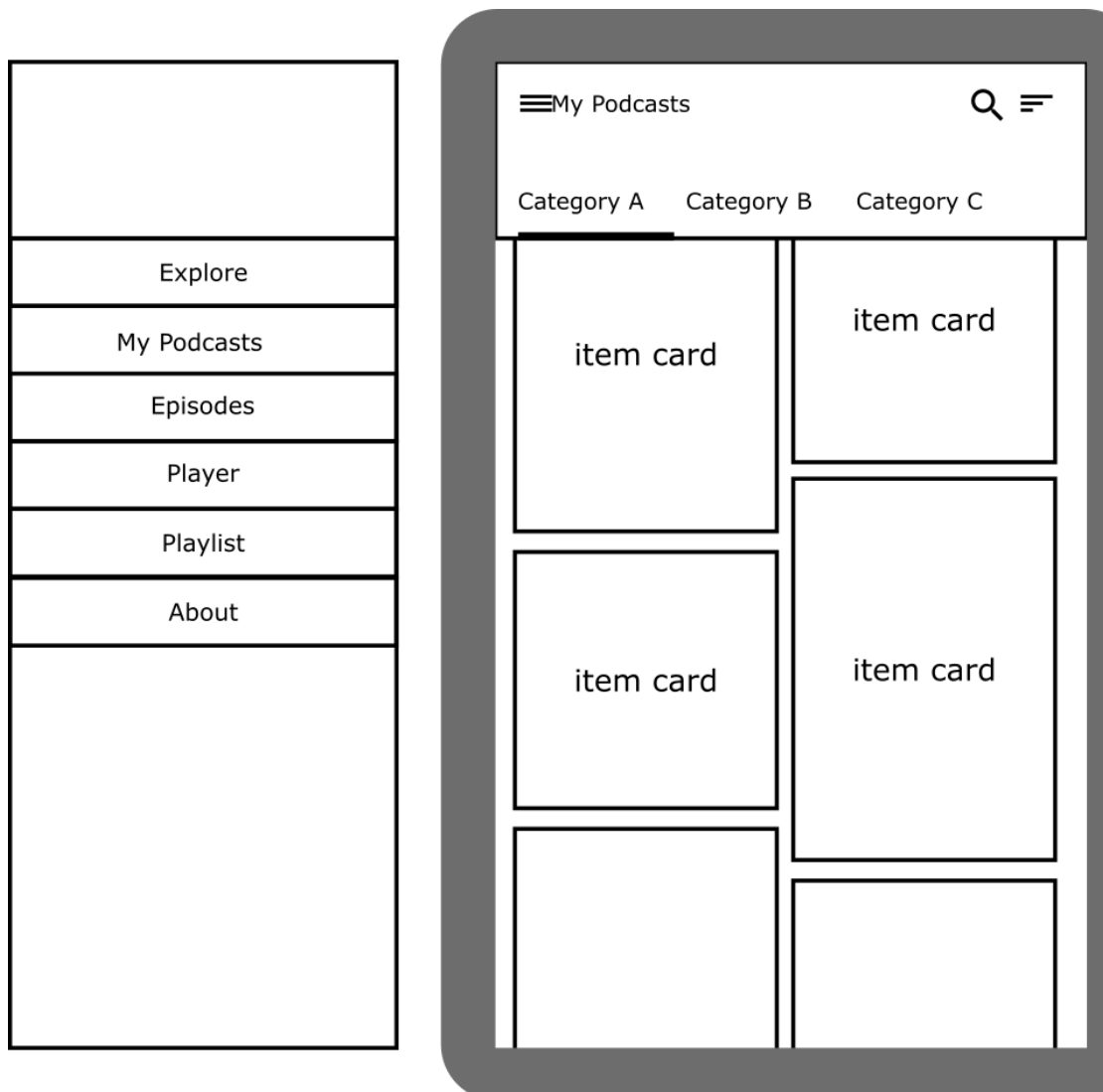
Features

- Discover or search for new podcasts.
- Subscribe/ Unsubscribe for podcast episodes and updates
- Download episodes for offline use
- Podcast player

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.

My Podcasts



Shows all subscribed podcasts, tabs show subscriptions by their categories. This is the launcher screen for the application.

Explore screen



This screen is all about adding more podcasts to your subscriptions. Displays a list of top-rated podcasts and allows for a search through GPodder's directory API.

Episodes screen



This screen displays a list of episodes. It will be re-used for the different contexts (displaying episodes for a certain podcast, downloaded episodes, the player playlist)

Podcast details



Display detailed information (artwork, description, feed urls, subscriber count, number of episodes, tags) about a podcast/subscription. The FAB will appear only for unsubscribed podcasts.

Podcast player



Provides playback controls for podcast episodes. Displays information about the currently played episode, has a shortcut to the playlist screen.

Key Considerations

How will your app handle data persistence?

Data persistence will be with a SQLite database

Describe any corner cases in the UX.

- All screens are accessible from a navigation drawer
- The GPodder API that will be used does not provide any notification mechanisms (WebSockets, push notifications, etc) so updates of pinned subreddits will happen on timed intervals or manually by the user through RSS feeds.

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

- OkHTTP/Retrofit/GSON
Simplify and speed-up the implementation of GPodder's REST API
- Picasso
Greatly simplified and optimized image loading and caching. Will be sharing a common cache with OkHTTP.
- EventBus
Reduces the boilerplate code and callback hell.
- RxJava+RxAndroid
Allow for easier management of asynchronous calls, chaining, filtering and modification of data. Allows for better observing of data.
- RetroLambda
Adds lambda support and other Java 8 features to Java 7 compatible projects.
- Nucleus
A tiny library that allows for a proper MVP architecture. Will aid in separating the view logic from the model and model-manipulation logic.
- Android Support Library v22+ & Design Support Library
Needless to say.

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:

- Register the app in GPodder. Get client id, secrets, etc

- Add & configure the dependencies to Gradle's build files.

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

- Build UI for MyPodcastsActivity (My Podcasts screen)
- Build UI for ExploreActivity (Explore screen)
- Build UI for PodcastDetailsFragment (Podcast Details Screen)
- Build UI for EpisodesFragment (Episodes screen)
- Build UI for PlayerActivity (Player screen)
- Others

Task 3: Implement model layer

- Implement the infrastructure around persistence.
- Implement a ContentProvider
- Create Models for the different types of entities
- Create Managers/Use cases that will handle the queries/fetching/persistence of entities, model change notifications and the submission to the UI
- Implement a podcast download service

Task 4: Implement Playback logic

- Implement a playback service.

Task 5: Updates Syncing

- Implement a periodic, battery-efficient way to fetch the updates for subreddits, maybe through a sync adapter.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
- 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**"