

## 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:** <https://github.com/TyraCodePDST>

## Free AudioBooks

### Description

This is an app that allows anyone to listen to audiobooks for free! Not need to sign up and create an account. It is using a public domain audiobook api from <https://librivox.org/>. User can browse hundreds of audiobooks from around the world, in a category of children fictions, non-fictions, poetry's...ects. User can also download the entire audio book content into his phone.

## Intended User

This app is great for everybody who's looking for an easy-use audiobook apps fast, and efficient.

## Features

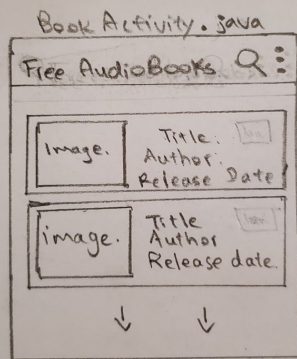
List the main features of your app. For example:

- Shows audiobooks in Recyclerview layout.
- Each audiobook includes (author, title, description, language, release date).
- Search button that allows user to search books (title, author, genre).
- Each book includes a download button.
- A audio player page ( play/pause, forward, and rewind button).
- User can share the entire book with friends.

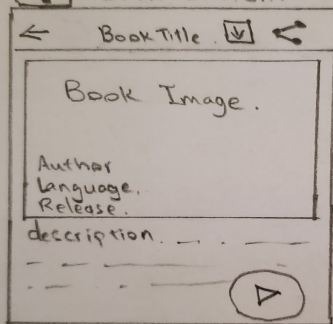
## User Interface Mocks

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.

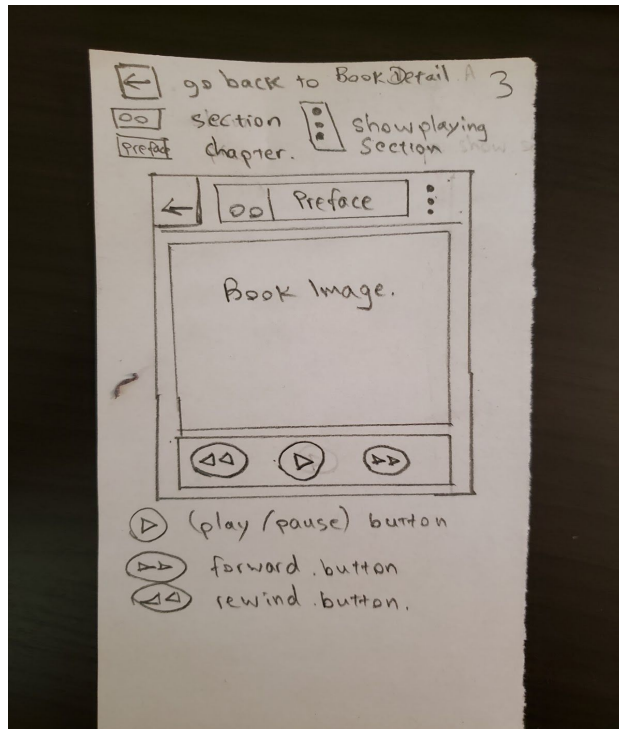
1



- ← go back to BookActivity. 2
- ↓ Download entire book.
- ↔ Share button.



⏮ play now button.



## Key Considerations

How will your app handle data persistence?

Using Room data persistence.

Describe any edge or corner cases in the UX.

User can navigate to a current playing state by press on the play button.

User can see what's being play (section and chapter).

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

Picasso is used to load the image of each book.

Retrofit is used for network calls (Http request).

Exoplayer is used for audio playbacks.

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

I will be using Google Admob service to populate ads for free version of the app.

## 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

Steps.

1. Create a simple BookActivity.class with its layout, populate a list of audiobooks.
2. Create a BookDetail.class that shows to user of selected audiobook with its info.
3. Create a PlayActivity.class that takes care of the playing features of a selected audiobook ( play/pause, forward, rewind, stop).
4. Create a widget , shows user of current playing audiobook or (shows a message that user has not selected any audiobook to play).

### Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for BookActivity, includes selected book , BookFragment (shows a list of General Fiction books)
- Build UI for BookDetailActivity, includes selected book , BookDetailFragment (shows user of the title, description, copyright\_year, language of the audiobook)
- Build UI for PlayActivity, includes selected book , PlayAudioFragment includes the playing features  
General Fiction books)

### Task 3: Implementing an Espresso test for Network calls, and data validation from the source.

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

- Create 3 or 4 espresso tests, making sure that the data is properly loaded,

### Task 4: Implementing a widget feature.

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

- Create a Widget.java class, and widget.xml file.

---

#### 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**"