

[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:** [Armi23](#)

# UpAudio Podcast App

## Description

UpAudio is a podcast listening app that allows you to take notes for your podcasts as you listen to them. You can create a highlighted section (or UpAudio) of a podcast episode which will be very easy to do and see visually on your screen. You'll then be able to take notes for that section as well and share both the UpAudio and notes with your friends.

## Intended User

The intended users are podcast listeners in general.

## Features

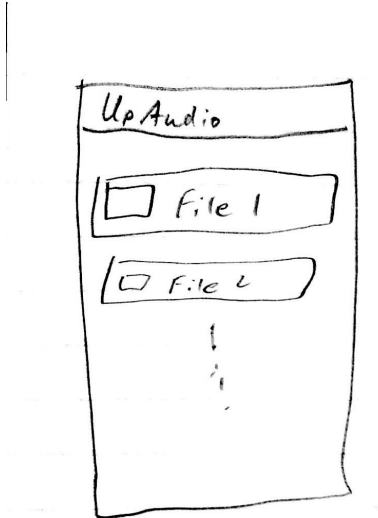
- Allows you to play podcasts (or any audio file)
- Allows you to mark sections of those podcasts as highlighted

- Allows you to take notes for those sections / podcast in general
- Allows you to share them

## User Interface Mocks

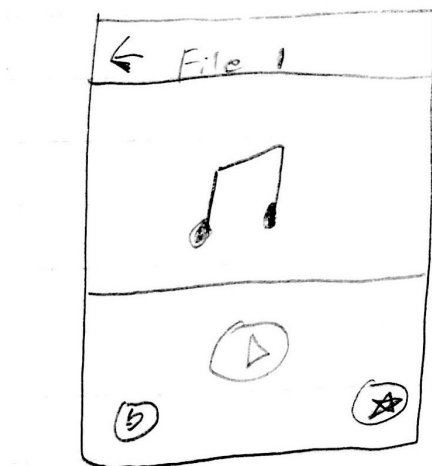
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

### Screen 1



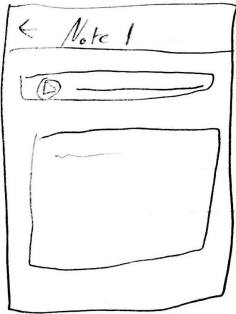
The opening screen of the app will display all files found that can be played. If none are available it will display a message letting the user know where they can upload their files.

### Screen 2



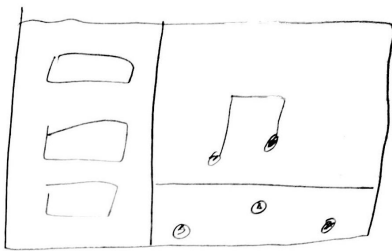
Tapping on a file will take you to a player screen where you can control it. If there is cover art for the file then it will be displayed. Under it will be the seekbar and below that playback controls. Supported controls include jumping back a few seconds (default 30, but should be configurable), play/pause, and the UpAudio button to make a new high light. You can tap to start a section and tap again to create a new note.

### Screen 3



Creating a new highlighted section will open the notes view where you can input notes if you want.

### Screen 4



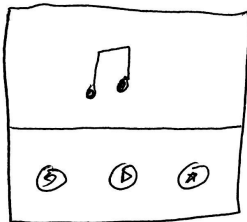
There will also be a tablet view of the player layout which combines the file view and the player view.

## Small Widget



A small widget will be available to the user that will provide the same basic playback controls available in the player screen.

## Large Widget



A larger version of the small widget will display the cover art of the currently playing file as well.

## Key Considerations

**How will your app handle data persistence?**

The app will have its own Content Provider to store, display, and share the UpAudio and affiliated notes.

**Describe any corner cases in the UX.**

Audio files will be retrieved from the device. There will be a “No files found” page if no audio files are found in the user’s storage.

If the user has previously made notes for an audio file then they’ll still have access to the notes but play options will be disabled to indicate the file cannot be played. This can be generalized to

other file error states that we may encounter (file corrupted, downloading but not yet playable, etc...).

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

Picasso will be used for displaying images. This is to keep the image loading code simple and easily extendable if I choose to allow podcasts that are streamed.

Exoplayer will be used for media playback since it is also extendable and would make streaming more easy if I choose to do that.

**Describe how you will implement Google Play Services.**

Firebase Realtime Database will be used for syncing notes across multiple devices

Firebase Authentication will be used to establish a user's identity for syncing as well.

## 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

- Start new project in Android studio
- Setup tasks to retrieve files from disk
- Setup unit tests for basic functionality

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

- Build UI for Fragment that will display available / playable files
- Build UI for Fragment that will serve as the player screen

### Task 3: Implement Playback functionality

- Implement full playback functionality (play/pause, jumpback, etc..)
- Implement cover art loading and display for player

#### Task 4: Implement notes

- In the player screen, implement functionality to make highlights and notes
- Build UI for screen that will handle editing notes and displaying them
- Use Loader to bring data to views
- Enable Sharing

#### Task 5: Implement Firebase Integration

- Implement support for Firebase Realtime Database and Firebase Auth
- Use Firebase JobDispatcher for sending/pulling information to and from database

Add as many tasks as you need to complete your app.

---

#### 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**"