

GitHub Username: [miki995](#)

MiniMax

Description

MiniMax is an app where you can find some of the most popular podcasts from #1 podcasting network in America, The commentist.

Currently it has 6 active podcasts:

- **Roll to hit** - a 5th ed Dungeons and Dragons,
- **Epic nitpic** - Each episode they choose one piece of pop culture, ranging from movies to music videos to cultural phenomenons,
- **Sky on fire** - star wars topics,
- **SkyFire comics** - Most of comics take place in New VanCity,
- **Loose Endz** - a liveplay podcast using D&D 5e,
- **The bearded vegans** - all things vegan including news, reviews, interviews and issues within the vegan community,

Users will be up to date on latest shows with notifications and ability to play the episodes in app.

Intended User

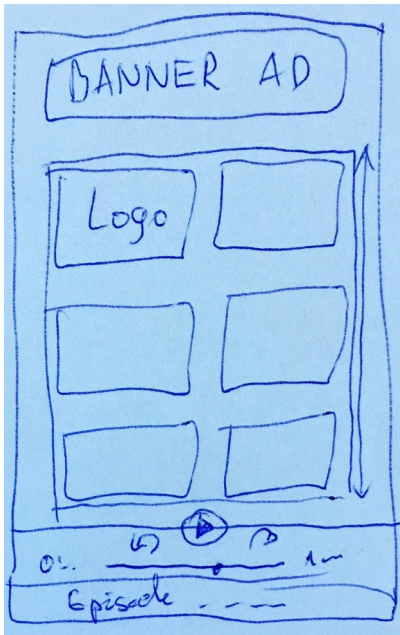
The app is for people looking for new set of podcasts to listen, and also fans of these shows.

Features

1. View current shows on MinMax.
2. View and listen (background supported) episodes of choosen show,
3. View description,hosts
4. Widget on main screen

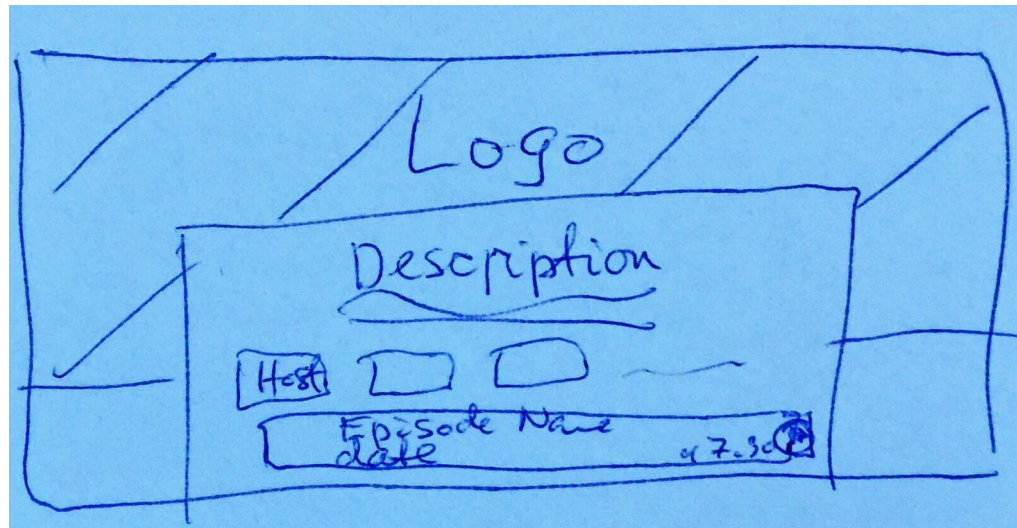
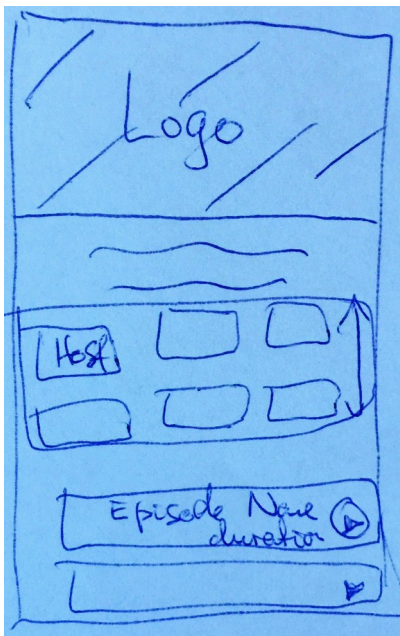
User Interface Mocks

Screen 1 Main screen show list



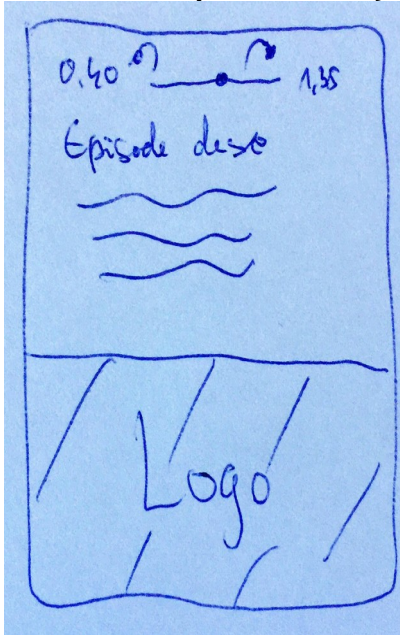
Main screen shows ad in free version,.
Logo images will be displayed in gridview.
Once started player will show up and persist on all screens.
It will have play, pause, forward/backward buttons.
It can slide up and down as well to display description.

Screen 2 Show Details



Details page will display logo, show description, host images and list of episodes.
Episode cardview consist of name, date, length and a button.
On tablet it will look like on mock.

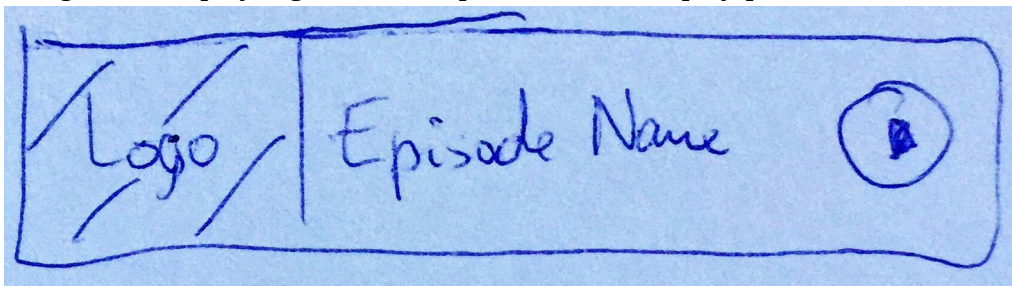
Screen 3 Expanded Player



Player page will show Logo, episode name, description, length.

Screen 4

Widget will display logo of show, episode name and play/pause button.



Key Considerations

How will your app handle data persistence?

On every launch the app will fetch new data via AsyncTask, which goes through Content Provider and finally storing in SQLite database.

Describe any edge or corner cases in the UX.

The media player will persist on all screens.

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

- Picasso to handle the loading and caching of images.
- Parallax Everywhere for different scrolling speed.

- Butterknife for binding views and callbacks to fields and methods.
- OkHttp as client for faster and secure network calls.

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

The app will use Admob for banner ads in free app version, and Notifications for sending alerts about new episodes.

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

Tasks:

- Async Task setup
- Content Provider
- SQLite database

Task 2: Implement UI for Each Activity and Fragment

Tasks:

- Build UI for MainActivity
- Build UI for Fragments (grid and page)

Task 3: Player Service Implementation

Tasks:

- Build Service for media player

Task 4: Flavors, Play Services

Tasks:

- Add free and paid version of app
- AdMob banner ad
- Notification about new episodes managed by admin

Task 5: Widget, Background play

Tasks:

- Background Play
- Widget Provider
- Widget UI