GitHub Username: Punpuf

Podceiver

Description

Listening to podcasts has never been easier, with Podceiver can easily listen to your favorite podcasts, download them for when you're offline. All with a very simple and easy to use design.

Intended User

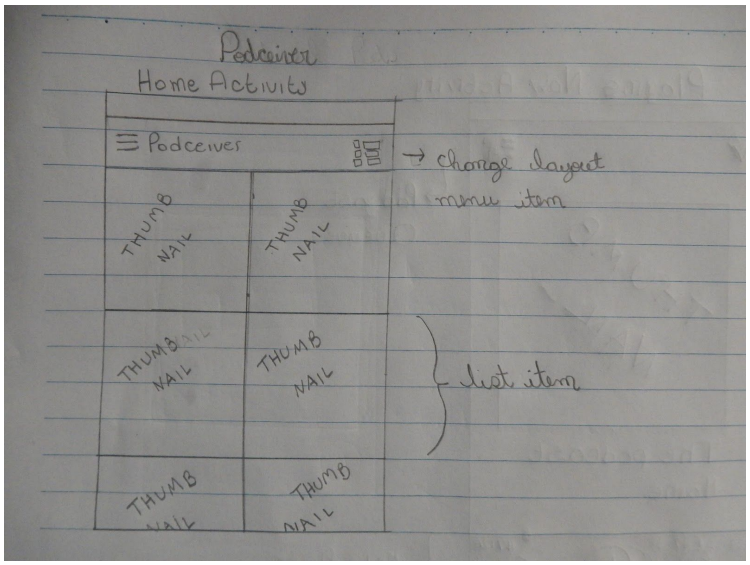
The intended user is anyone who likes and would like to listen to podcasts.

Features

- Search for podcasts.
- Subscribe to podcasts.
- Play podcasts.

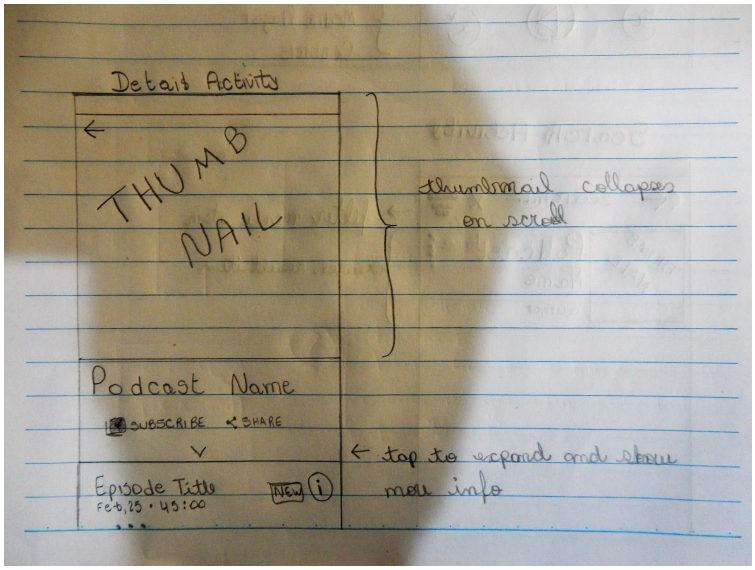
User Interface Mocks

Screen 1



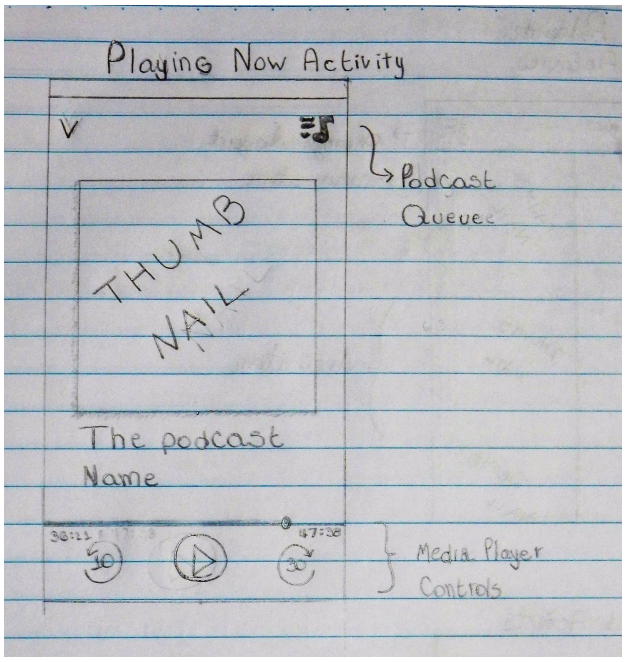
Home Activity - List of podcast items.

Screen 2



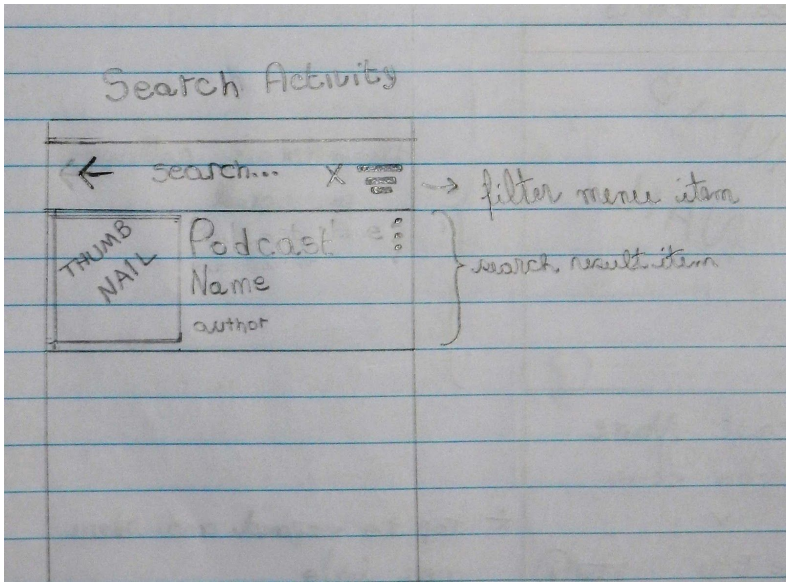
Detail Activity - shown when user clicks on a podcast item.

Screen 3



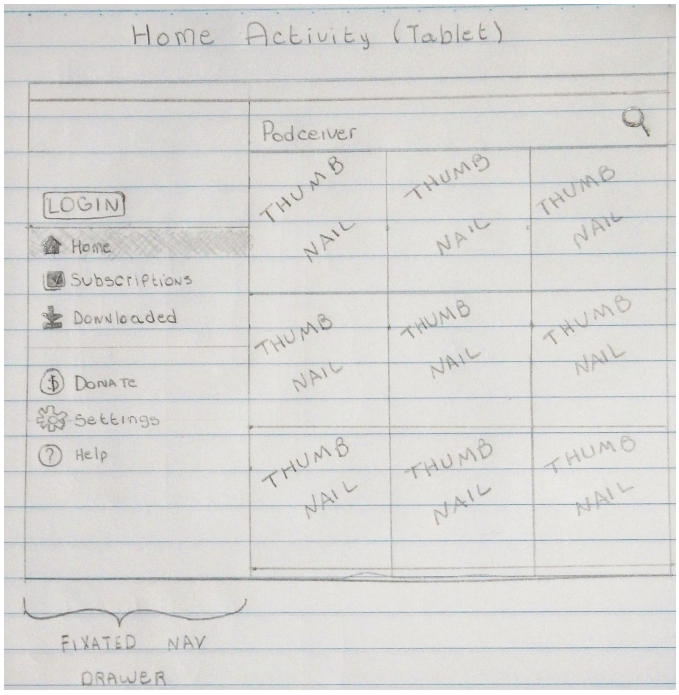
Now Playing Activity - Shown when user clicks on media player controls bottom sheet.

Screen 4



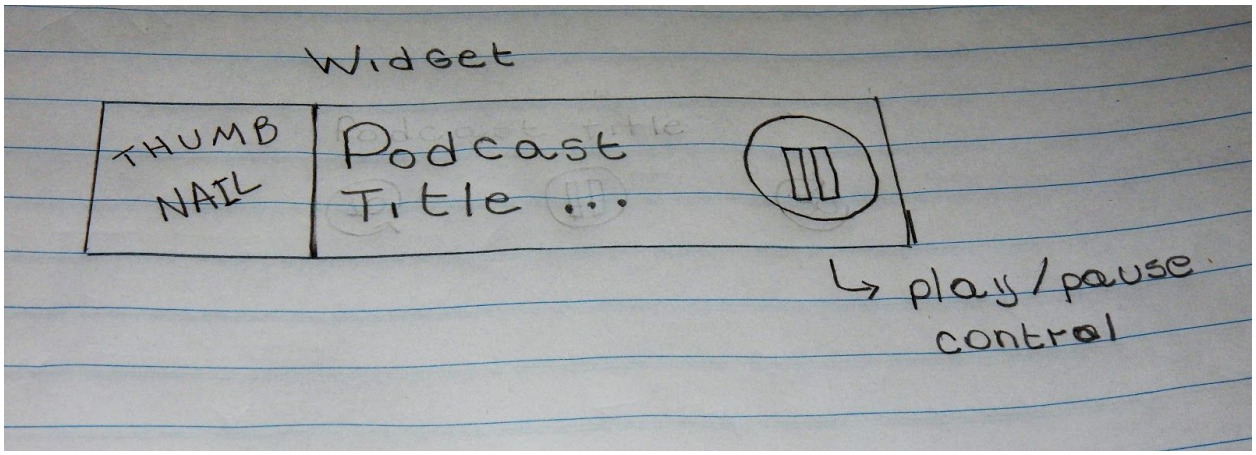
Search Activity - shows search result items.

Screen 5



Home Activity (Tablet) - the nav drawer is fixed in a open state.

Screen 6



Widget - visualize current playing podcast, and control it’s playing state.

Key Considerations

How will your app handle data persistence?

I will handle data persistence using an SQLite database and access it through a Content Provider.

Describe any corner cases in the UX.

When in landscape orientation on a phone there won't be a lot of vertical space so maybe it would be better to not show the podcast thumbnail.
On tablets in landscape orientation since there's a lot of horizontal space it would be better to use a 2 pane layout.

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

Timber - for logging.
Butter Knife - for view binding.
Firebase Analytics - for analytics
Volley - for fetching simple data like json and images.
Exoplayer - streaming the podcasts.

Describe how you will implement Google Play Services.

Firebase Analytics: I will add a call on the start of the app to initialize analytics, and later on call it to log more events.
Firebase Notifications: I will add a service so I can receive notifications even when my app isn't on the background.

Next Steps: Required Tasks

Task 1: Project Setup

- Add and configure required dependencies to project.
- Create and add icon.
- Connect with firebase.

Task 2: Implement UI for Each Activity and Fragment

- Build Splash Screen.
- Build UI for Home Activity.
- Build UI for My Subscription Activity.
- Build UI for Search Activity.
- Build UI for Detail Activity.
- Build UI for Now Playing Activity.

Task 3: Networking

- Fetch network data (e.g. displaying home screen's top charts).
- Parse it.

Task 4: Data Binding

- Populate views with data fetched from network.
- Handle item clicks (open detail activity).
- Implement Activity Transitions (shared element).

Task 5: Storing Data

- Implement SQLite database.
- Implement Content Providers.
- Add "subscribe" functionality.

Task 6: Add Playback Functionality

- Implement Exoplayer.
- Implement Media Player.

Task 7: Add Download Feature

- Store files to local storage.
- Retrieve files.