Podkis Functional Specification

GitHub Username: jasonscott5391

Podkis

Description

Browse and listen to the latest and greatest podcasts curated by Podkis editors. Don't miss out on the most recent episodes of your favorite podcasts. Whether you're commuting, working, or just relaxing, stream episodes directly from your phone to your ears with the simple, free, and no nonsense app Podkis!

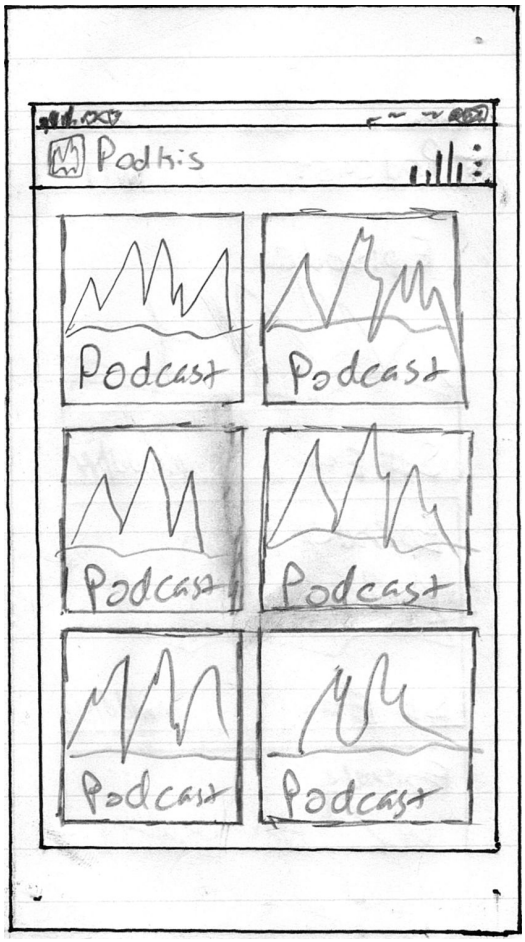
Intended User

Podkis is intended for any user, technologist or not, that wants a simple way to stream their favorite podcasts and discover new content.

Features

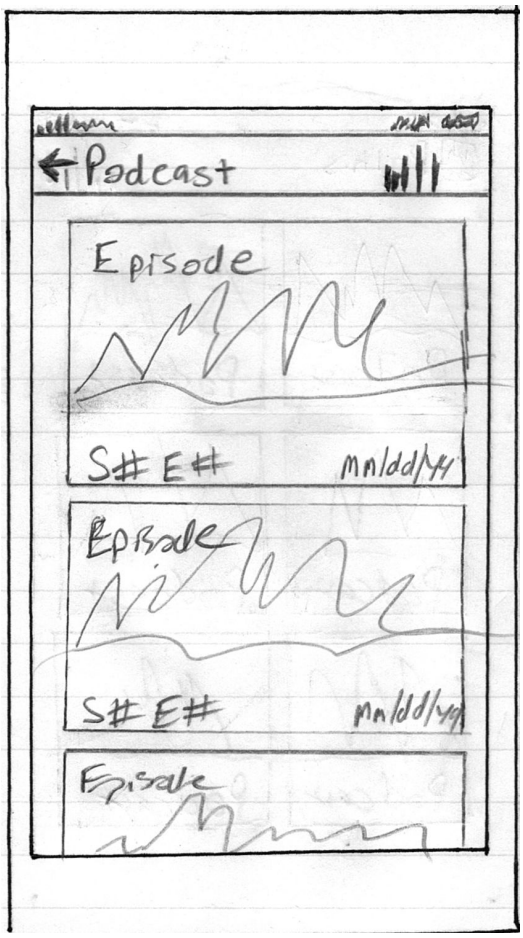
- Browse from a selection of podcasts curated by Podkis editors.
- Discover latest episodes at a glance for a particular podcast.
- Stream episodes of a chosen podcast.

User Interface Mocks



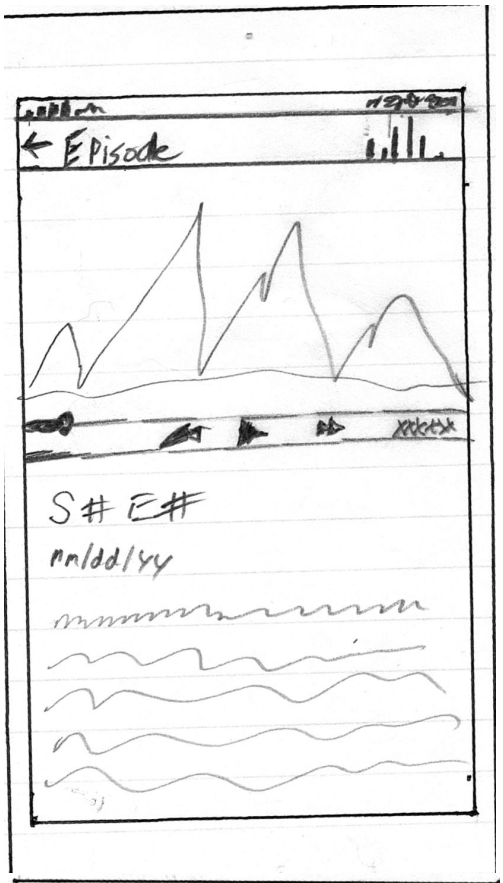
MainActivity Screen

- MainActivity screen with RecyclerView using GridLayoutManager.
- Each grid item contains podcast main image and title.



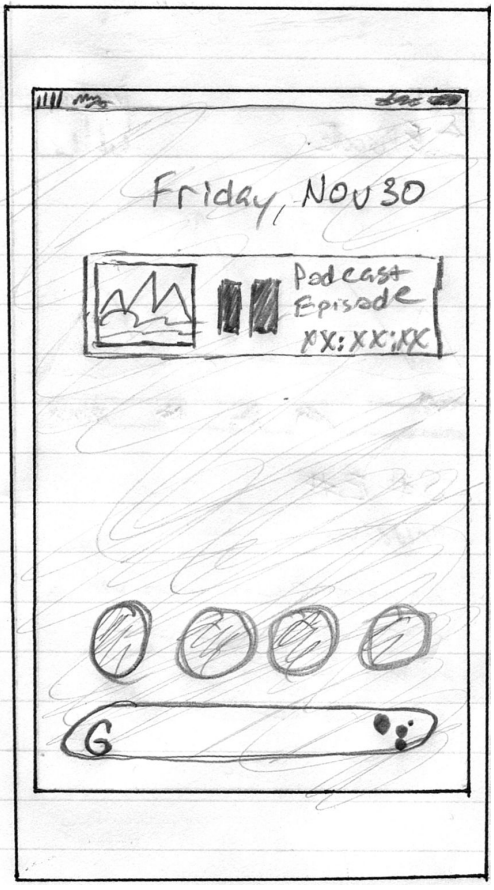
PodcastDetailActivity Screen

- PodcastDetailActivity screen with RecyclerView using vertical LinearLayouManager.
- Each list item contains episode image (podcast main image as fallback), title, season number, episode number, and published date.



EpisodeDetailActivity

- EpisodeDetailActivity screen with Exoplayer.
- Exoplayer image is episode image (podcast main image as fallback).
- Episode title is in the AppBar.
- Episode season number and episode number are followed by published then episode summary.



PodcastWidget

- ImageView with episode thumbnail (main thumbnail as failback).
- TextView with podcast name and episode title.
- Play/Pause button.

Key Considerations

How will your app handle data persistence?

Using configured podcast RSS URLs, Podkis will download new content, when available, persisting to a SQLite database using Room persistence library.

Describe any edge or corner cases in the UX.

No matter which screen the user has navigated to they will be able to return to now playing screen with the click of a button on the AppBar or notification when app is not in focus.

Podcast stream will continue to play when focus has left the app with media controller buttons as notification.

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

The Retrofit with simple XML converter will be used for deserializing XML from podcast RSS feed into entities, picasso will be used for images and Exoplayer for streaming audio.

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

Podkis is directly integrated with RSS feeds from the cloud controlled by podcast authors. It uses Google Analytics for Firebase to track app usage and user engagement as well as AdMob for earning revenue.

Required Tasks

Task 1: Project Setup & Best Practices

- Create new project in Android Studio (latest version 3.2.1) called Podkis.
- Set gradle distribution URL for latest version 4.6.
- Set compile, minimum, and targeted SDK versions…
    - *compileSdkVersion -> 28*
    - *minSdkVersion -> 19*
    - *targetSdkVersion -> 28*

- Add dependencies to app's gradle file…
    - *Support libraries...*
        - *AppCompat version 28.0.0*
        - *Constraint version 1.1.3*
        - *RecyclerView version 28.0.0*
        - *CardView version 28.0.0*
        - *Design version 28.0.0*
        - *MediaCompat version 28.0.0*
    - *Exoplayer version r2.2.0*
    - *Picasso version 2.71828*
    - *Retrofit with Simple XML converter version 2.4.0*
    - *Room persistence version 1.1.1*
- Use Java language for development.
- Keep all strings in a strings.xml file and enables RTL layout switching on all layouts.
- Include support for accessibility by using content descriptions and navigation using a directional pad.

Task 2: Implement UI for Each Activity and Fragment

- Implement main Activity UI using RecylerView for displaying podcasts in grid and associated adapter.
    - Grid item as CardView with ConstraintLayout should include podcast title and main image.
- Implement podcast detail Activity/Fragment UI using RecylerView for displaying episodes in list and associated adapter.
    - List item as CardView with ConstraintLayout should include episode title, episode image (main as fallback), season #, episode #, and published date .
- Implement episode detail Activity/Fragment UI using ConstraintLayout to include Exoplayer with image and episode title, summary, season #, episode #, duration, and published date.

Task 3: Create Entities, Database, and Dao using Room persistence Library

- Create entities for a podcast and episode. Each podcast has reference to its episodes and each episode a reference to its podcast.
- Create Podkis SQLite database with entities above.
- Create dao with CRUD operations.

Task 4: Implement Sync Service using Retrofit Library

- Create podcast service interface that uses Retrofit annotations for defining endpoints to consume.

- Create podcast synchronized task that uses Retrofit to build podcast service that consumes endpoints and write to database using podcast dao.
- Create intent service for syncing Podkis content and register in manifest.
- Create podcast sync utility class for starting sync intent service.

Task 5: Create and Integrate ViewModel using LiveData with Activity/Fragment RecylerViews adapters.

- Create ViewModel, and ViewModelFactory if necessary, for podcast list, episode list, and episode itself.
- Implement ViewModels within each associated Activity/Fragment.

Task 6: Create Podcast Widget Layout, Service, and Provider and integrate with Activities

- Create widget layout with ImageView for episode/podcast thumbnail, ImageButton for play/pause functionality, TextView for podcast and episode.
- Create PodcastWidgetProvider for handling updates and register in manifest.
- Create PodcastWidgetService for handling intents and register in manifest.
- Support play and pause of podcast episode on click.