GitHub Username: aadilahmed

Rainy Day Podcasts

Description

This podcast app allows users to search for and listen to podcasts, and add them to a favorites list.

Intended User

The intended user is anyone who wants to keep a list of favorite podcasts and easily listen to them from their smart device.

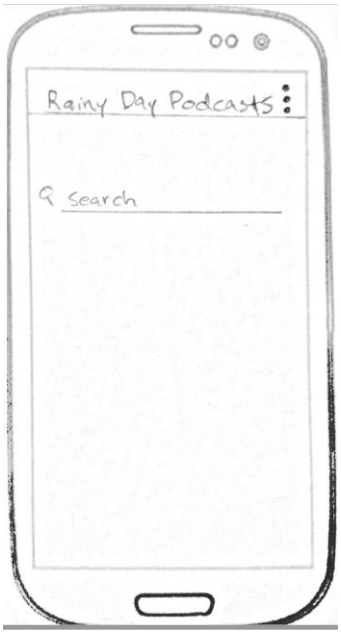
Features

- User can search for podcasts
- Listen to podcasts through a media player
- Add podcasts to a favorite list
- Access favorites list from companion home screen widget

User Interface Mocks

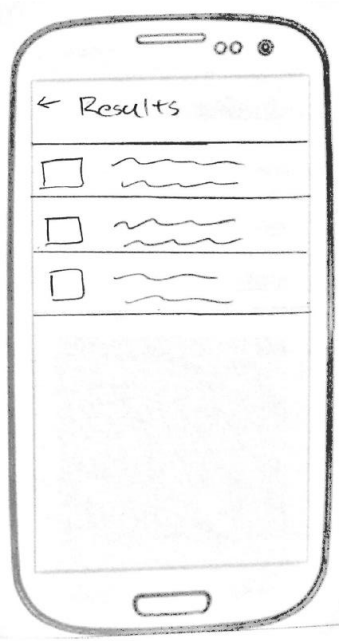
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



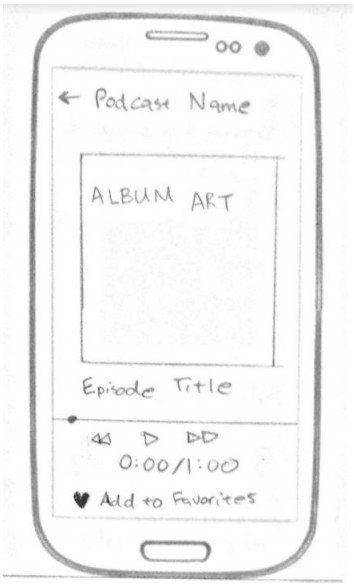
This is the main activity which has a search bar to search for podcasts, and a menu button in the top right corner to access the user's favorite list.

Screen 2



This activity shows the podcast search results. By clicking one of the results the user goes to the podcast player screen. Clicking the favorites button on the first activity will launch a results screen with only favorites listed.

Screen 3



This is the podcast player activity. The player controls are at the bottom and you can add the podcast to the favorites list by clicking on the heart image.

Screen 4

This is the favorites podcast list where the user can directly enter the podcast player activity for each favorite. This list will also be displayed in the companion home screen widget.

Key Considerations

How will your app handle data persistence?

The app will use a local Room database with LiveData to hold the list of favorite podcasts data. There will also be a companion home screen widget that will use a content provider to display the list of favorite podcasts.

Describe any edge or corner cases in the UX.

The ExoPlayer state will be saved on rotation or moving to another activity using shared preferences.

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

Glide will be used to handle image loading as it is generally faster than Picasso. ExoPlayer will be used to play podcast audio. Room will be used as the database backend for holding favorite podcasts.

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

This app will be backed by the Listen API which is a podcast search API. I will pull the JSON data from the API using an asynctask to populate search results. The app will also use Google AdMob to display ads in the free build variant. The paid build variant will not display ads.

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

- Add required libraries to the app level build gradle.
- Create activities for each of the screens

Task 2: Implement UI for Each Activity and Fragment

- Build UI for the main search activity
- Build UI for the search results activity
- Build UI for the podcast player activity

Task 3: Connect to the internet

- Connect to the API url using HTTPUrlConnection
- Create a method to parse JSON result
- Create asynctask to handle pulling search results

Task 4: Build Favorites Database

- Create entity for favorite podcast entries
- Create DAO for favorite entries
- Create AppDatabase class that implements Room
- Use LiveData and ViewModel to view favorites list

Task 5: Build widget

- Create layout for the favorites widget
- Create widget provider and RemoteViewsServices classes
- Build content provider to supply the list of favorite podcasts to the widget

Task 6: Implement Google Play Services

- Create free and paid build variants of app
- Use Google AdMob to add advertisements to the free variant

Task 7: Handle Error Cases

- Create tests using Espresso to make sure all UI connections are working correctly.
- Test AdMob ads