Table of Contents

Github Username: thisisaicee

POTTERHEADS

Harry Potter Book Data App for Fans

Description

Potterheads App is an Android App - solely written on java - that will enlist and allow users to search for information on all the Harry Potter Book series

Intended User

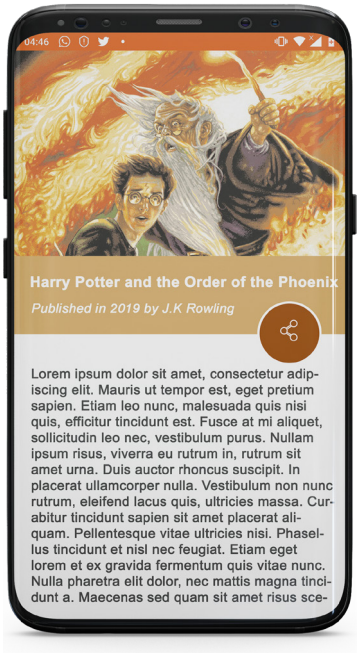
This app is intended to be used by Harry Potter Fans or anybody willing to explore any information on Harry Potter books

Features

- It's also required to mention that all the libraries, services, Gradle and AS will use only stable release versions
- Explore all Harry Potter book series
- Access extra details on any of the book on the series
- Share item details with others
- Ability to add any book to favorites list

MainActivity

DetailActivity



Widget

Key Considerations

How will your app handle data persistence?

Potterheads data are retrieved using PAPI (Potterheads API). The app will save data internally using SQLite and a Content Provider.

Describe any edge or corner cases in the UX.

In case of any failure, empty list content or lack of internet connection, the user is informed with an appropriate status message;

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

- Glide for image loading and caching.
- Butter Knife for view binding

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

• Firebase Storage to store image assets
• Google AdMod to monetize the app by displaying in-app ads

Next Steps: Required Tasks

Task 1: Project Set-up

- Create the project and add it to git repository
- Set up the libraries
- Structure the project packages: view, data, model, libs, etc...
- Add the necessary ProGuard rules
- Generate a KeyStore and add it in the project's root directory
- Add in Gradle the signing configuration with the KeyStore's location, password and the key alias name.

Task 2: Set-up API

- Create the project and add it to git repository
- Set up the libraries
- Structure the project packages: view, data, model, libs, etc...
- Add the necessary ProGuard rules
- Generate a KeyStore and add it in the project's root directory
- Add in Gradle the signing configuration with the KeyStore's location, password and the key alias name

Next Steps: Required Tasks - Continued

Task 3: Implement UI for MainActivity

- Build UI for MainActivity
- Implement the procedure of fetching the item details from the server and launching the DetailActivity to display them on fragment item click callback

Task 4: Implement Item Fragment

- Fetch the items from the server and display them on a recycler view
- On save and restore instance state, preserve and restore recycler's view data and current position
- Inform the parent Activity on item click or when the loading status change Something else

Task 5: Implement Item Fragment

- Build UI for Detail Activity
- For the selected item, create dynamic RecyclerViews regarding the item lists related to that item
- Implement sharing functionality, making use of intent extras to share the item details

Task 6: Implement Widget

- Create widget using GridView layout
- Load items to widget
- On the item click, the widget, must provide via an intent extra the selected item's id to the MainActivity