
GitHub Username: paulnunezm

Bookito

Description

A books collection manager with a wishlist. Giving you the control of your personal library that you always desired.

Intended User

For everyone that enjoy reading and buying books and keep things organized.

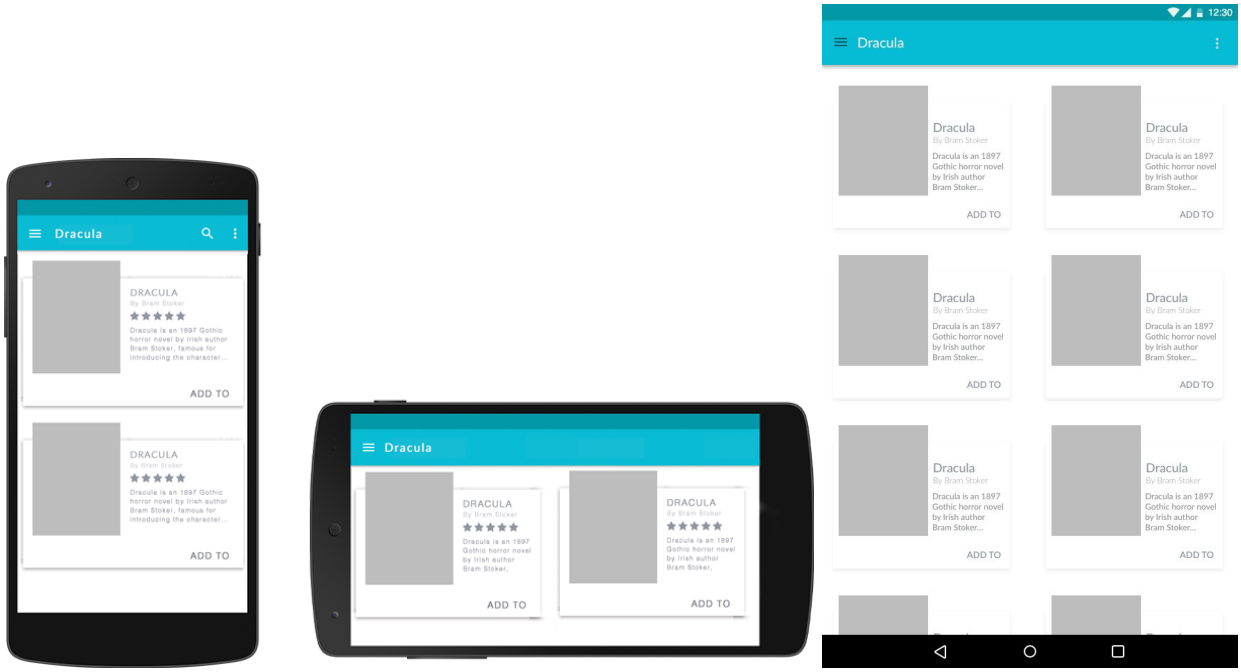
Features

- Search books from Goodreads
- Add/remove books from "owned books" list
- Add/remove books from to "wishlist" list

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 Photoshop or Balsamiq.

Screen 1

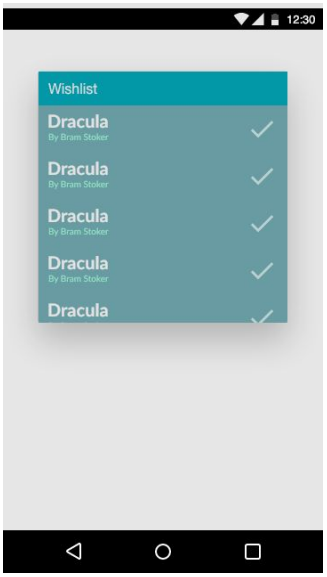


Search screen. The user will be able to search books from the Goodreads book database and add it to it's wishlist or owned book shelves.

Screen 2

Shelve. The list of books that the user has saved in its owned/wishlist shelve.

Screen 3

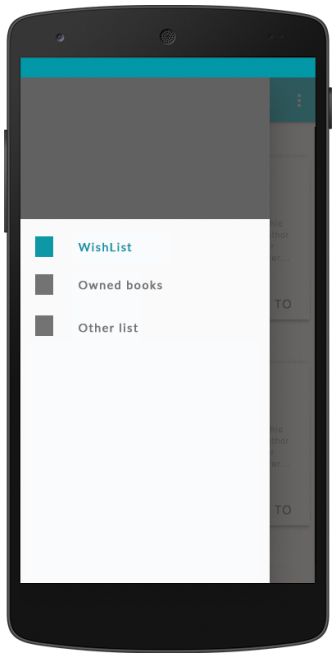


Homescreen widget. Giving the user a quick way to move books from the wishlist to the owned list.

Screen 4

Details. To provide book details to the user.

Screen 5



Navigation drawer. Enabling users quick switch from shelves.

Key Considerations

How will your app handle data persistence?

The wishlist and owned books information will be saved using Firebase Realtime Database.

Describe any corner cases in the UX.

The app will initiate in the “Search books” screen. Where the user will be able to search books and click on them to open the details screen. In this screen the information cards have an “Add to” button that let the user add the book to one of the book shelves.

The shelve screen. Can be the owned or wishlist shelve. It will show the list of books that the user has saved on it. This cards haves an option that will let the user change the book from the current shelve to other or to delete it from it.

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

Picasso to handle the loading and caching of images.
Retrofit to handle html request.
SimpleXml for parsing the Goodreads response.
Mockito to performing tests.
Butterknife for easier view to object binding.
Support library to ensure compatibility with older Android versions.
Firebase real time Database to give the users.
Firebase Authentication to authenticate users.
Firebase AdMob to present ads.

Describe how you will implement Google Play Services.

I will use Firebase services instead to perform authentication and maintain data synced between devices and to present ads.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

- Configure libraries
- Configure Firebase data structure
- Set API key in settings.gradle
- Create different packages to maintain the code organized
- Setup app colors theme

Task 2: Implement the components behavior

All screens:

- For each screen create the contract interface with presenter,view, and interactor methods to specify the behavior of each one.
- Start writing the presenter while testing to assure correct behavior between getting the data from the interactors and displaying it within the view.
- Write the interactor while testing to assure that always send correct data to the presenter.
- After the view has it's behavior defined, start writing the implementation, creating the layout and adding custom interactions (view clicks, on text changed for the search screen and bottom sheets for the shelve screens)

Task 3: Writing the model implementation

- After creating an interface that will model the repository behavior write the implementation of getting data from the servers and store it locally.
- Create methods to request books searched from the Goodreads API.
- Create methods to save/request wishlist or owned books from Firebase.

Task 4: Implement Admob

- Configure admob in the project.
- Add the admob view to the layouts.

Task 5: Implement Firebase Authentication

- Add a login screen.
- Manage Firebase Authentication.

Task 6: Homescreen widget

- Create homescreen widget layout.
- Manage Firebase Authentication.

Technical tasks:

- For loading books from the Goodreads API to the view, the app will use an AsyncTaskLoader as the request will be a short operation and to provide a data loader to the view.