
GitHub Username: AMoharib

BooketList

Description

Like a to-do list for your books, To Read provides a quick and simple way to catalogue and inventory the books you want to read later. Easily add new books with a simple search, once your books are in, you can update your read progress and get a reminders notification.

Intended User

Users who interested in reading books.

Features

- Search Books
- Sign in feature using firebase
- Sync data by uploading it to firebase real time database and save it offline.
- Make Reminders

User Interface Mocks

Screen 1 - Login Screen

Login page with email and password

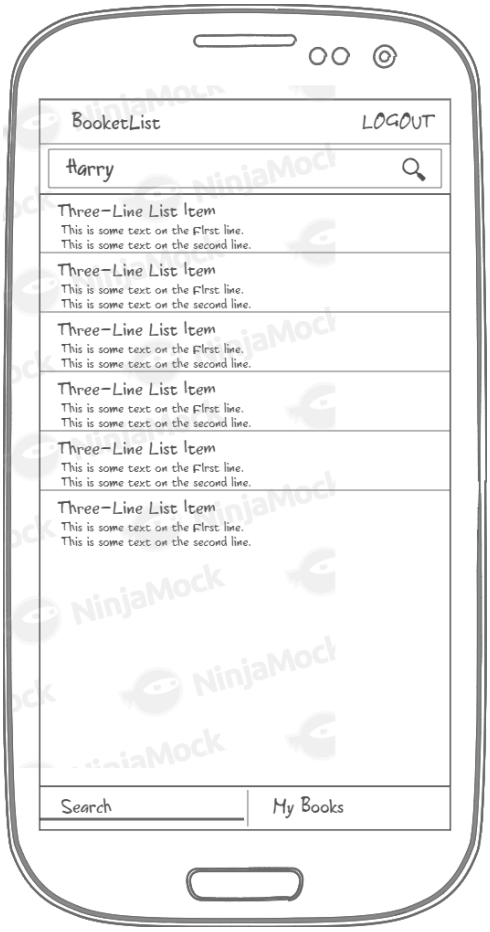
Screen 2 - Register Screen

Register page for making new account with email and password.

Screen 3 - Search Screen

Home page where users can search for their books.

Screen 4 - Search Result Screen

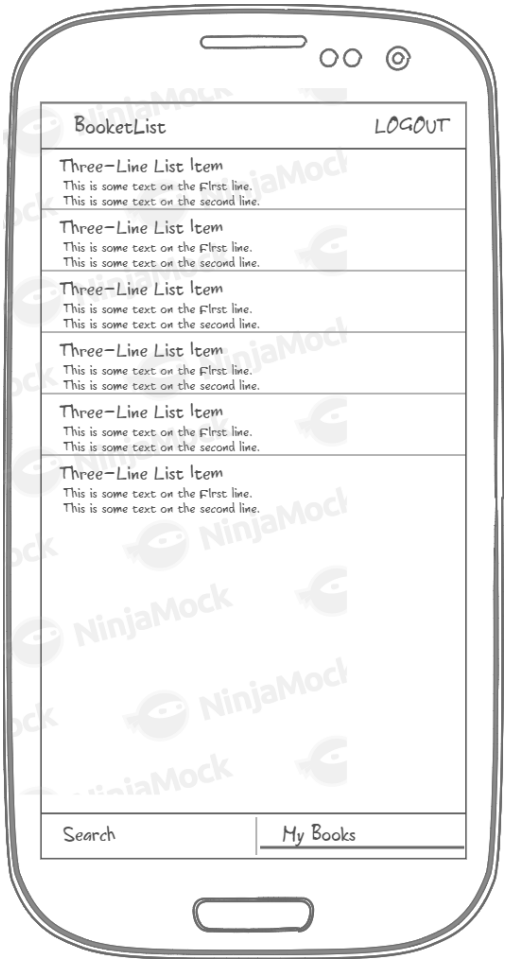


Search result page

Screen 5 – Book Details Screen

Book details page started when users needs to see the book details and to add it to his list.

Screen 6 - My Books Screen

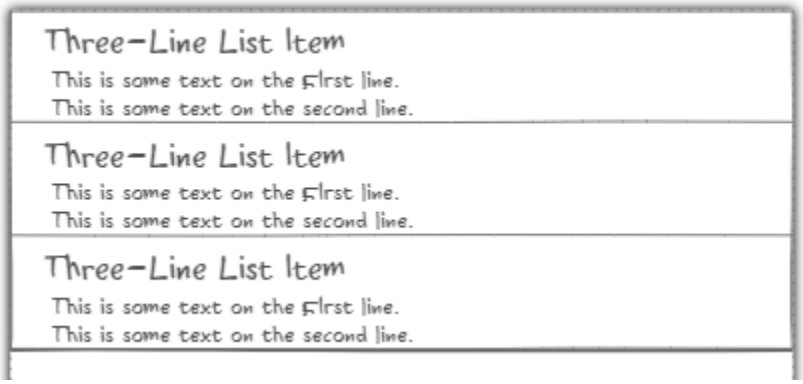


Users books list page updated when he adds books in the search page

Screen 7 – My Books Details Screen

Book details page where users can update their reading progress or remove books from his list.

Screen 8 - App Widget



A widget for viewing the current interested books and open it on click for updating the reading progress.

Key Considerations

How will your app handle data persistence?

Room database is used to store data locally and firebase real time data base is used to sore data online to be retrieved when the app is removed from the device.

Describe any edge or corner cases in the UX.

- The app will be written solely in the Java Programming Language
- The app keeps all strings in a strings.xml file and enables RTL layout switching on all layouts
- The app will not use maps or navigation or any media players
- The process of saving data to the application is done by after the user search for his book online he can save it locally and remotely with firebase then the app will sync between them with a service, and if the application is removed the data is saved before online so he can download it again.
- The app checks validity of user inputs before login or register
- The application must not perform any demanding operation on the main thread which might cause the app to feel buggy or freeze.

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

Picasso for loading image
Dagger 2 for dependency injection
Retrofit as a REST client
Rxjava for writing clean and simpler code and dealing with threads

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

Google Play Services –

- Firebase Sign In
- Firebase Realtime Database

Next Steps: Required Tasks

Task 1: Project Setup

Creating a setting up a new project from scratch. This will include –

- Creating a new project in Android Studio.
- Configuring the required libraries.
- Initializing the Github project repository.

Task 2: Implement UI for Each Activity and Fragment

For this task, I will –

- Creating a new Firebase project.
- Using SHA keys and google.json files to sync the project with Firebase.

Task 3: Creating the UI

This will include –

- Starting development the UI mocks.
- Focusing on responsive design.

Task 4: Building the app logic

For this task, I will –

- Setting up Firebase Authentication within the app.
- Setting up the Firebase Realtime Database within the app.
- Test run the services above to make sure they are working fine.
- Binding the view containers with the incoming data.

Task 5: Implementing data persistence

In this stage, I will explore different ways to make sure that data available for offline use too.

Task 6: Notification

Implementing Notifications which will notify users when a book is not read for a while.

Task 7: Developing the App widget

Creating the application widget, which will show the user's books and open it on click to update.

Task 8: Developing the App widget

This task will include the following steps –

- Providing a finish to the app design.
- Providing accessibility support.
- Creating and using app icons.
- Removing debug logs and messages.
- Generating the required keys and files (JKS).
