

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Login](#)

[Landing Screen](#)

[Home Fragment](#)

[My Books Fragment](#)

[Favorites Fragment](#)

[Scan Screen](#)

[Search Screen](#)

[Detail Screen](#)

[Trade Flow](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Environment Setup](#)

[Task 2: Project Setup](#)

[Task 3: Implement Services and Backend](#)

[Task 4: Implement Landing screen](#)

[Task 5: Implement UI for each Activity and Fragment](#)

[Task 6: Create Widget](#)

GitHub Username: pablo-johnson

# ShareBook

## Description

Do you have a lot of books that you already read and want more? Share them and receive new ones! It's easy and safe. Let us find you people who has the books that you are searching for.

## Intended User

This app is for everyone, no matter the age or genre. It is for everyone who likes reading and sharing.

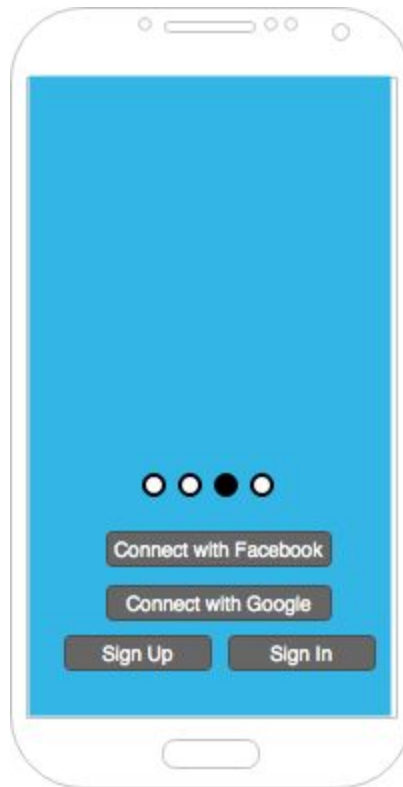
## Features

The main features of this app are the following ones:

- Share books with your friends.
- 
- Suggest a good book to your friends.
- Know your friends favorites books and read them

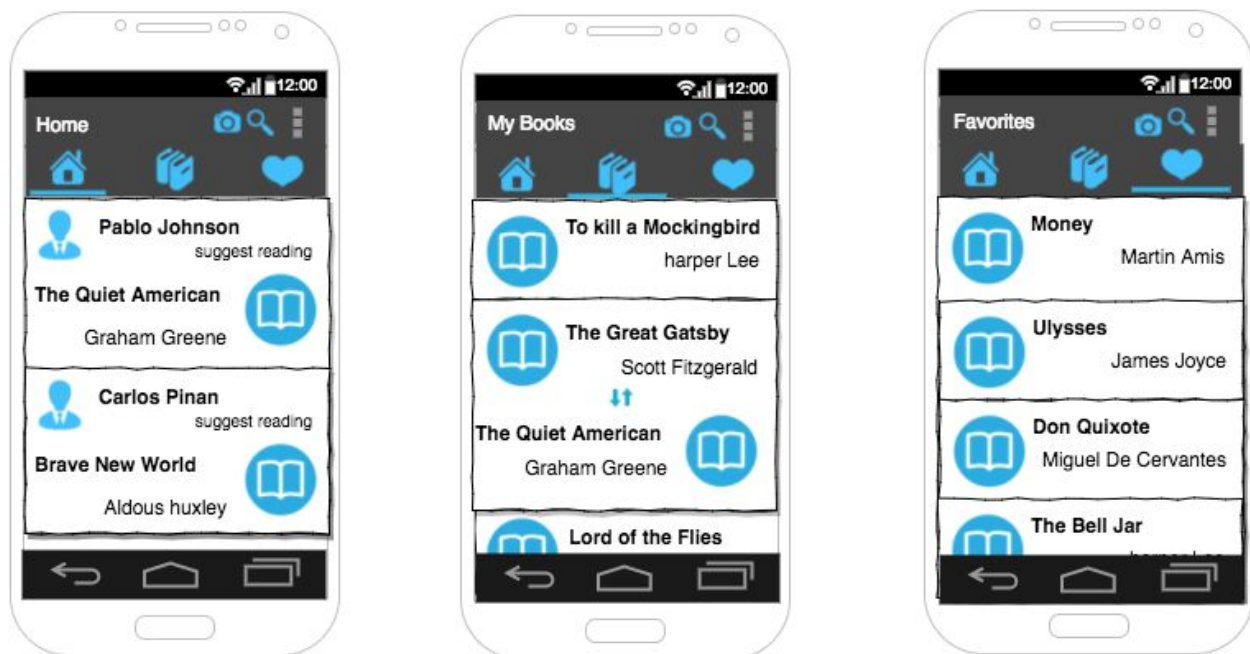
## User Interface Mocks

### Login



An image gallery (ViewPager) showing basic use of the app with some buttons in the bottom part of the app. This are the Facebook and Google login options. The user can select one of them or login/register using its own mail and password.

## Landing Screen



The main activity shows a tabhost with 3 fragments. This is the first screen the user will see when he gets into the app. Additionally, it shows two actions in the toolbar: Scan code and Search.

## Home Fragment



This screen shows the user's friends activity, suggestion and comments about several books. Here can also be seen trade books requests from our friends or questions about some book in specific.

## My Books Fragment



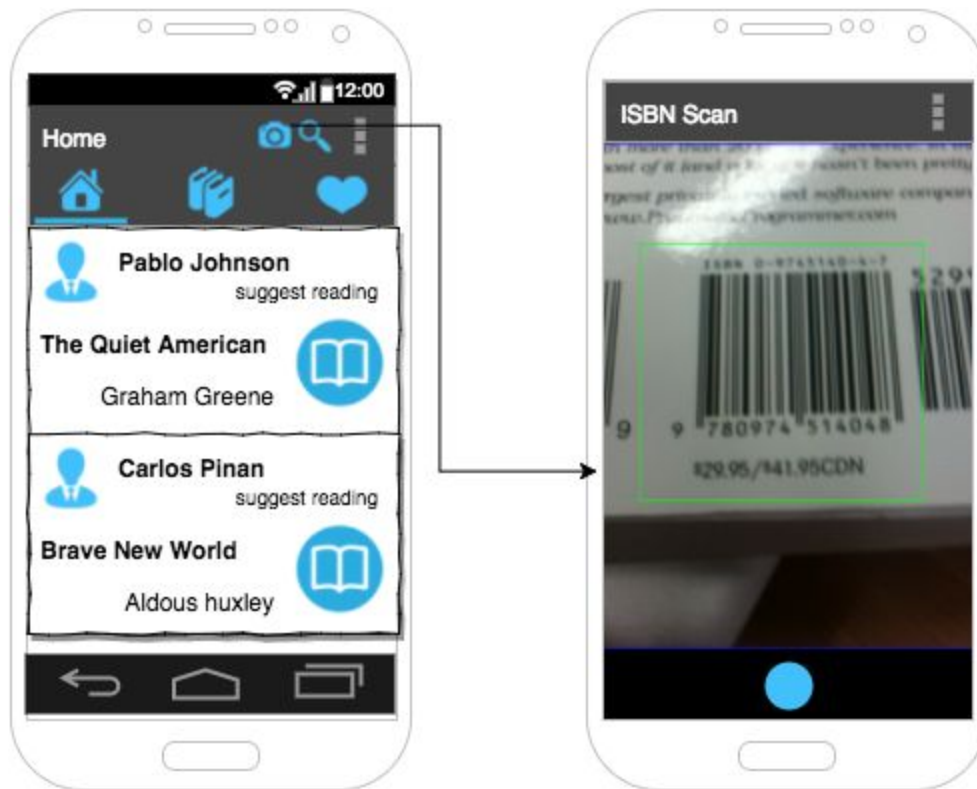
This screen displays the user's library. User can tap any item to see more details about the selection.

## Favorites Fragment



This screen displays a list of books that the user has mark as favorites. This books may be owned by the user or not.

## Scan Screen



The Scan screen is accessed via the toolbar action button in the main screen. This screen uses the device camera to scan the ISBN book code and adds it to the user library.

## Search Screen



In this screen the user can search for a book he is interested on. The search can be done by title or by author.

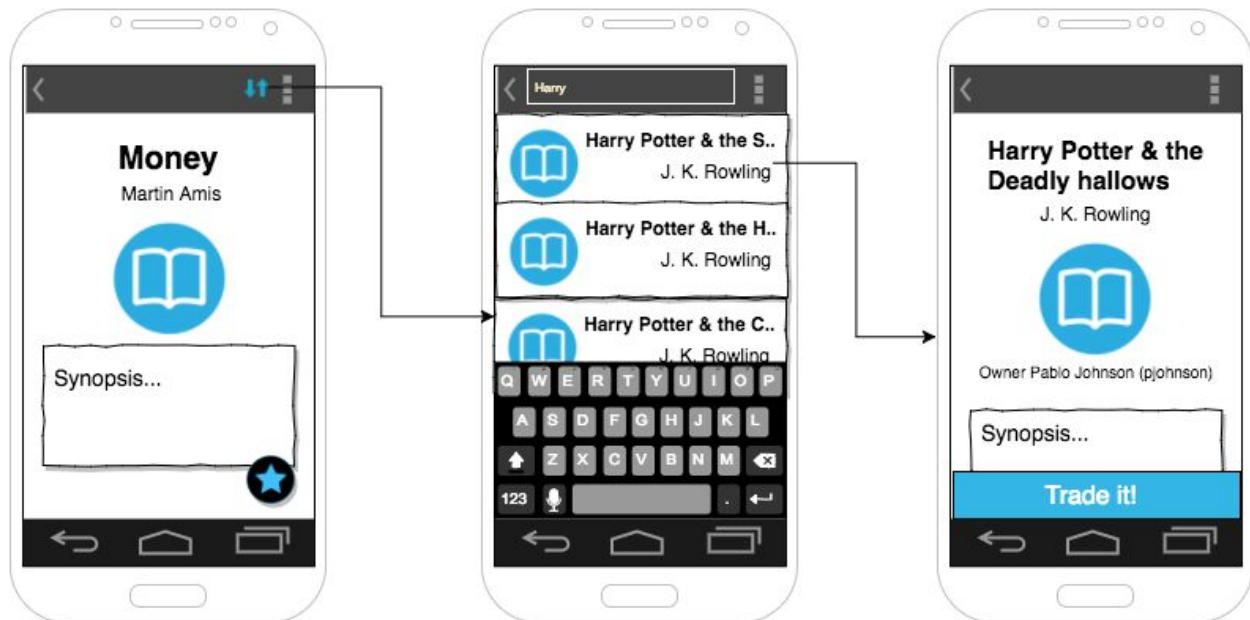


## Detail Screen



This screen displays details of the selected book like title, author, date, synopsis, genre, etc. It has a button (floating button) to mark it as a favorite. The toolbar has an action button to start the trade flow.

## Trade Flow



The trade screen is very similar to the detail screen, it has a bottom button to confirm the trade and shows additional data about the owner.

## Key Considerations

### How will your app handle data persistence?

The app will obtain user data, the activity feed and its book matches through a background task that will connect with the REST services. The user can save its favorite books and this will be stored in a SQLite Database that will be accessed by a private content provider.

### Describe any corner cases in the UX.

The app will have a tab widget in the main screen where the user can interact with the main features: Home, My Books and Favorite.

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

- Butterknife library to reduce boilerplate by UI dependencies injection.

- Glide library to obtain images and caching them locally.
- Retrofit library and possibly usage of Google Endpoints API and Firebase API to handle data request, storage and authentication.
- Android Design Support library to use components such as Navigation View, FAB, Coordinator Layout and so on.
- Crashlytics

## Next Steps: Required Tasks

### Task 1: Environment Setup

The following tasks are needed to set up the environment:

- Download and install Java 7 SDK
- Download and install latest version of Android Studio
- Download/Update Android SDK and support libraries.
- Download and install gradle 2.10.
- Download and install latest version of GitHub UI (including git).

### Task 2: Project Setup

These are the steps to get the project ready to run:

- Checkout the repo from this url <https://github.com/pablo-johnson/Capstone-Project>
- Import as a gradle project in Android Studio.
- Make a Gradle Sync.
- Run it!

### Task 3: Implement Services and Backend

I will Implement the backend services and the DB using Firebase.

### Task 4: Implement Landing screen

In this task, I will create the main screen structure where the main flow of the app relies on. This will be a tabhost with 3 fragments: Home, My Books and Favorites. Those will be the point of entry for the others features.

## **Task 5: Implement UI for each Activity and Fragment**

The order of the views will be the following:

- Main Activity that will display the navigation structure (tabhost) with 3 tabs and two main actions in the toolbar: scan book and search book.
- Home fragment that will display all the activity of your friends.
- My Books fragment where the user can see a list of all of his books.
- Favorites fragment that will display the books that the user has marked as favorites.
- ISBN Scan screen that will display a camera view where the user will scan the ISBN code of the book he wants to register.
- Book Detail screen that will display details of the selected book like title, author, year, synopsis, owner, etc.
- Search screen where the user can search for a specific book by its title or author.
- Settings screen to let the user customize some app behaviours.
- Login screen using Firebase Authentication.

## **Task 6: Create Widget**

I will create a collection widget to display the activity of some selected friends and another widget to display the user book collection.