BOOKWORM: TECHNICAL DETAILS

AIM

The app aims to help people who want to find and share books with other people near them. It enables users to search for the books they want within a 5 KM radius of their location. The app lists all the other users who have the book and allows the user to contact them for exchanging the book.

As far as we have researched, there are no other apps which allow for book-sharing in a locality. Additionally, the app also allows users to add or search for books by directly scanning ISBN barcodes.

INTRODUCTION

The app enables users to find nearby users who have the book they are looking for. For this, the app stores the location of each user using the GeoFire library. When the user searches for a book, search results are fetched using the Google Play Books API. When one of these searched results is selected, a GeoFire query is made which fetches all users in a 5 KM radius. Once the users are fetched, the app checks which of these users have the requested book. These users are then listed and then the logged in user can contact them using the in-app chat feature where they can exchange pictures too.

The app also allows users to scan the ISBN barcodes of books to get more details about a book. For this, the app uses the BarcodeScanner library. The user has to point their camera towards the barcode and it gets scanned automatically, taking the users to a page where more details of the book can be found.

IMPLEMENTATION

According to the project objective, the following requirements were initially thought of:

- Integration with Firebase for user account management and data storage.
- Using Google Play Books API to get book details for use in the app.
- A user book inventory for storing the books that a user owns.
- A chat feature to enable communication between users.
- A way to find and fetch nearby users.
- An efficient database structure for storing user messages.

During the course of development, we realized some more requirements:

- The chat feature needed to have a way for users to exchange pictures. This is mainly for exchanging pictures of books.
- A revised database structure to enable sharing of these picture messages.
- Firebase storage was also required for storing the pictures.
- We also had to extend the Activity and Fragment classes to allow for new updateUI functions to be called when the API replies were received.
- An ISBN barcode scanner was needed to get details of books that the user owns, so that they can be easily added to inventory.

NOVELTY

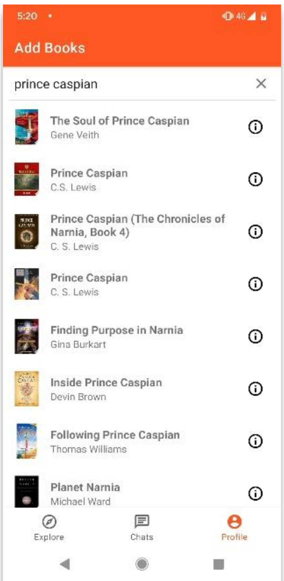
Some of the things we self-learned and implemented are as follows:

ANDROID VOLLEY LIBRARY

Volley is an HTTP library which allows automatic scheduling of network requests for making API calls.

We used this library for making our API calls to Google Play Books. We had to create request URLs according to the user's requirements and fetch the JSON results using this library.

PLAY BOOKS API



prince caspian

The Soul of Prince Caspian
Gene Veith

Prince Caspian
C.S. Lewis

Prince Caspian (The Chronicles of Narnia, Book 4)
C. S. Lewis

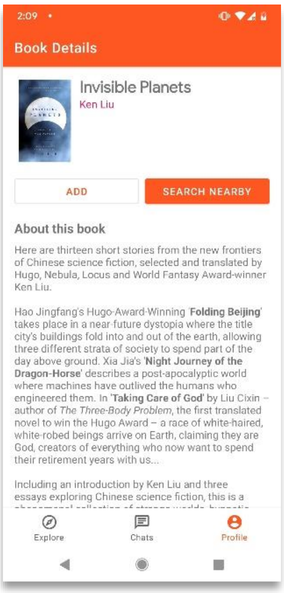
Prince Caspian
C. S. Lewis

Finding Purpose in Narnia
Gina Burkart

Inside Prince Caspian
Devin Brown

Following Prince Caspian
Thomas Williams

Planet Narnia
Michael Ward



Invisible Planets
Ken Liu

About this book

Here are thirteen short stories from the new frontiers of Chinese science fiction, selected and translated by Hugo, Nebula, Locus and World Fantasy Award-winner Ken Liu.

Hao Jingfang's Hugo-Award-Winning '**Folding Beijing**' takes place in a near-future dystopia where the title city's buildings fold into and out of the earth, allowing three different strata of society to spend part of the day above ground. Xia Jia's '**Night Journey of the Dragon-Horse**' describes a post-apocalyptic world where machines have outlived the humans who engineered them. In '**Taking Care of God**' by Liu Cixin – author of *The Three-Body Problem*, the first translated novel to win the Hugo Award – a race of white-haired, white-robed beings arrive on Earth, claiming they are God, creators of everything who now want to spend their retirement years with us...

Including an introduction by Ken Liu and three essays exploring Chinese science fiction, this is a

We used Google Play Books API for two purposes: Firstly, to fetch books according to the search term that the user entered in the Search Bar. Secondly, to get the book details of books using their ISBN. The ISBN was extracted by scanning ISBN barcodes.

We also used the API to fetch images of book covers for displaying on the UI.

GEOFIRE LIBRARY

GeoFire is an open-source library that allowed the app to store and query user locations. It stores locations with string keys and can query keys within a geographic area in real-time.

It uses the Firebase Realtime Database for data storage just like the rest of the app.

When the user creates their account, we store their location using GeoFire which allows to later fetch the user's location when needed. We mainly use this to find users within a 5 KM radius of the current user.

FIREBASE

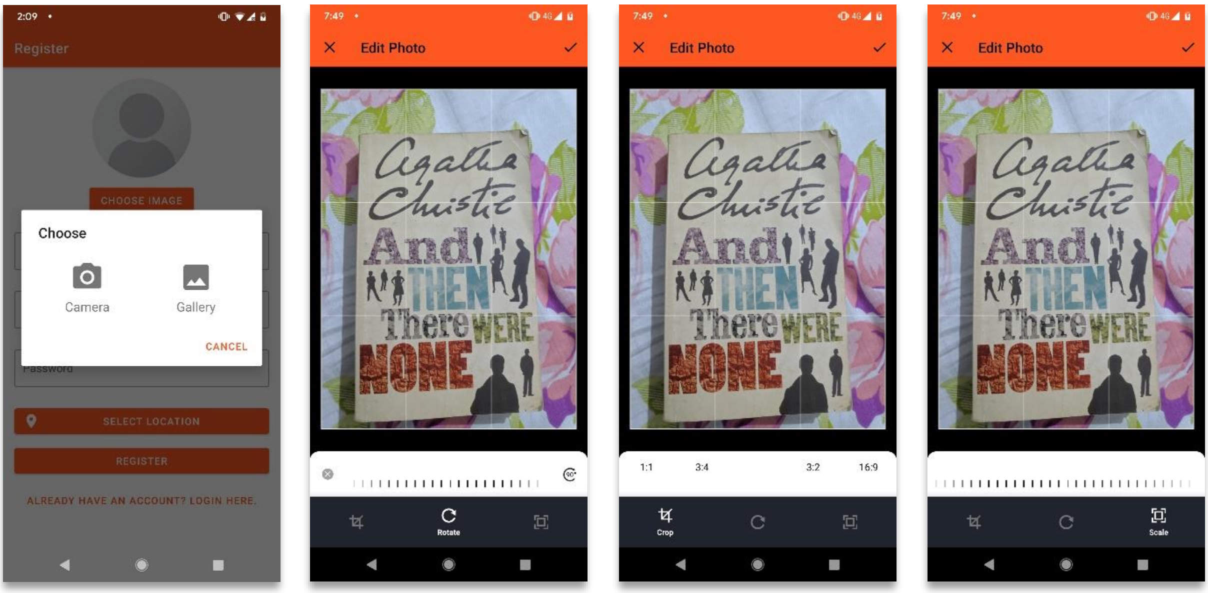
We are using Firebase for user account management and for storing user data. We are also using it for storing GeoFire locations.

Firebase Realtime Database allows us to store user data like their first name, last name, current location, the location of their profile photo in Firebase Storage and the list of books that they own.

Firebase Realtime Database also allows us to store the messages being exchanged by users in chats.

We are using Firebase Storage to store the user profile photos as well as to store the pictures exchanged by users in chats.

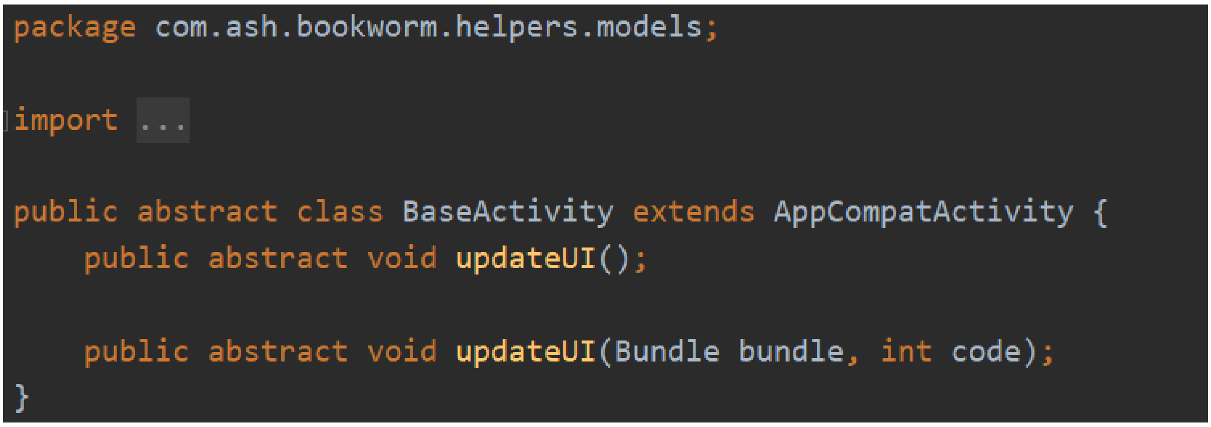
IMAGE PICKER LIBRARY



The Image Picker library allowed the app to pick an image from the Gallery or capture image using Camera. It also allows to crop and compress the image based on aspect ratio, resolution and image size.

Using this library made handling pictures easier.

EXTENDED ACTIVITY AND FRAGMENT

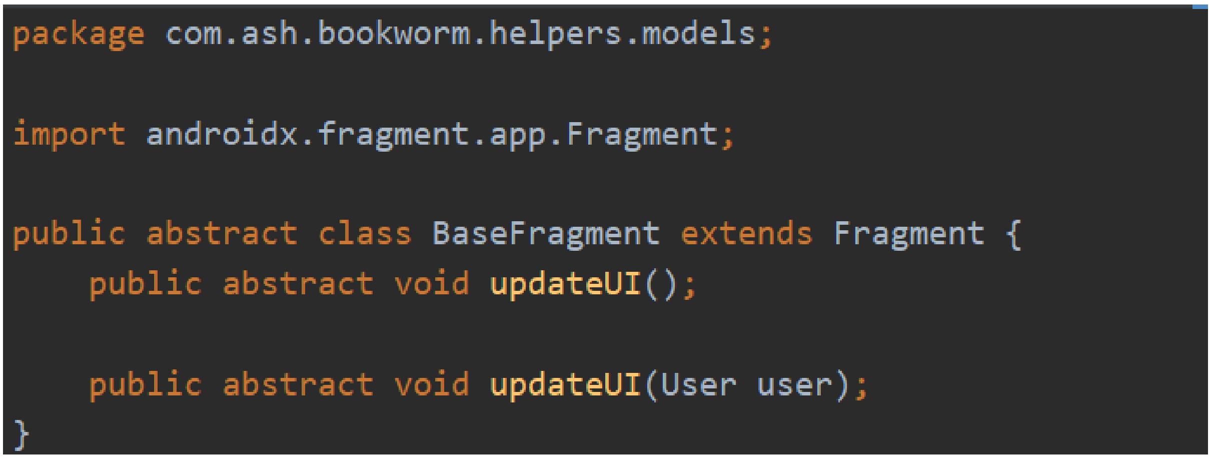


```java
package com.ash.bookworm.helpers.models;

import ...

public abstract class BaseActivity extends AppCompatActivity {
    public abstract void updateUI();

    public abstract void updateUI(Bundle bundle, int code);
}
```



```java
package com.ash.bookworm.helpers.models;

import androidx.fragment.app.Fragment;

public abstract class BaseFragment extends Fragment {
    public abstract void updateUI();

    public abstract void updateUI(User user);
}
```

We had to extend the Activity and Fragment classes so that we could create updateUI functions which could be used as a callback whenever API replies were received. All the Activities and Fragments in the app extend from these classes (BaseActivity and BaseFragment).

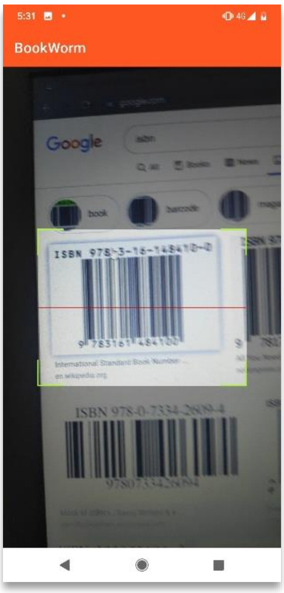
This allowed us to use the same functions for multiple activities and fragments.

Whenever an API response is received, the updateUI function of the activity / fragment that made the API request is called. The Activity / Fragment can override their updateUI function to utilize the data received from the API response to change their UI.

The updateUI function with the User parameter is for updating the UI with the details of the user.

Similarly, the updateUI function with the Bundle and code parameter are for updating the UI according to the data in the bundle. The code is used to identify which API is invoking the updateUI method.

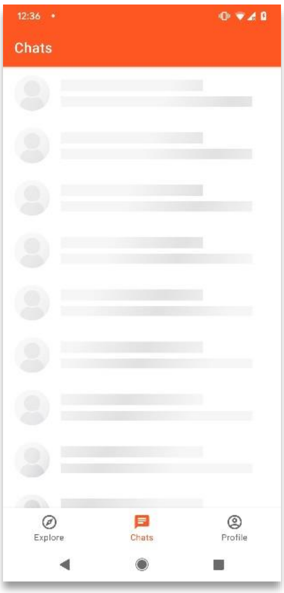
BARCODE SCANNER



Barcode Scanner is a library which uses the phone’s camera to scan barcodes. We use it to scan ISBN barcodes of books.

When an ISBN barcode is scanned, the ISBN string of the book is returned to the previous activity which uses the Google Play Books API to fetch the details of the book associated with the scanned ISBN.

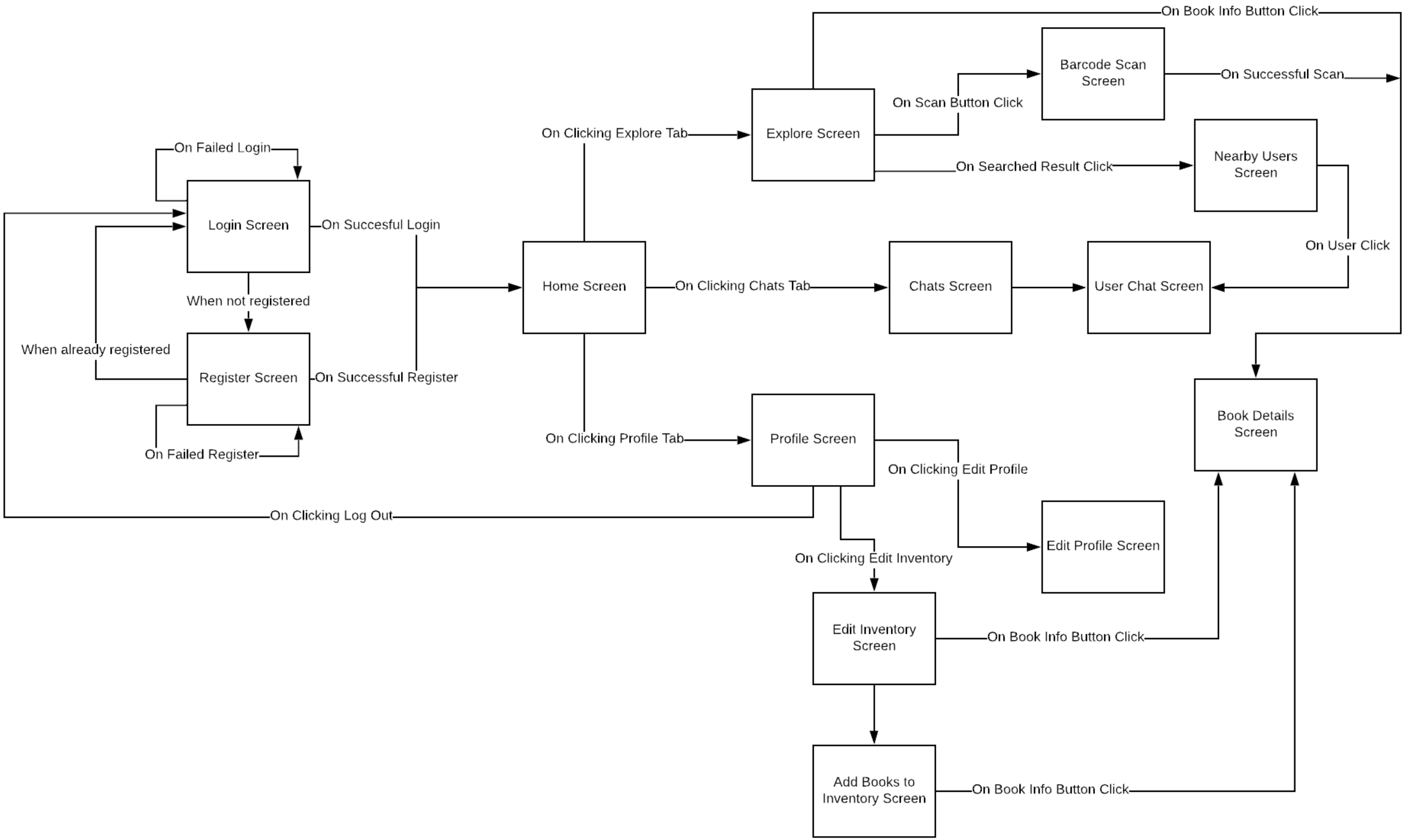
FACEBOOK SHIMMER LIBRARY



We are using Facebook’s Shimmer library to create attractive loading animations for the app’s components.

The library requires us to create a placeholder view of the components being loaded. The placeholder view is then placed inside a ShimmerLayout and a shimmer effect is applied on the placeholders by the library.

DESIGN



On Book Info Button Click
Barcode Scan Screen
On Successful Scan
On Scan Button Click
On Clicking Explore Tab
Explore Screen
Nearby Users Screen
On Searched Result Click
On Failed Login
Login Screen
On Succesful Login
On User Click
Home Screen
On Clicking Chats Tab
Chats Screen
User Chat Screen
When not registered
When already registered
Register Screen
On Successful Register
Book Details Screen
On Clicking Profile Tab
Profile Screen
On Failed Register
On Clicking Edit Profile
On Clicking Log Out
Edit Profile Screen
On Clicking Edit Inventory
Edit Inventory Screen
On Book Info Button Click
Add Books to Inventory Screen
On Book Info Button Click