How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: nghianja

Find My Book

Description

My Udacity Android Developer Nanodegree's capstone app, Find My Book, is developed to help booklovers locate the nearest library that a book is available by enabling a user to scan a book's ISBN number and retrieve its information from the national library's database.

This app is currently available for users in Singapore only as it accesses the Open Web Service REST API of the Singapore's National Library Board at this time. Future versions will auto select API sources based on the user's country location.

Intended User

Book-loving library members

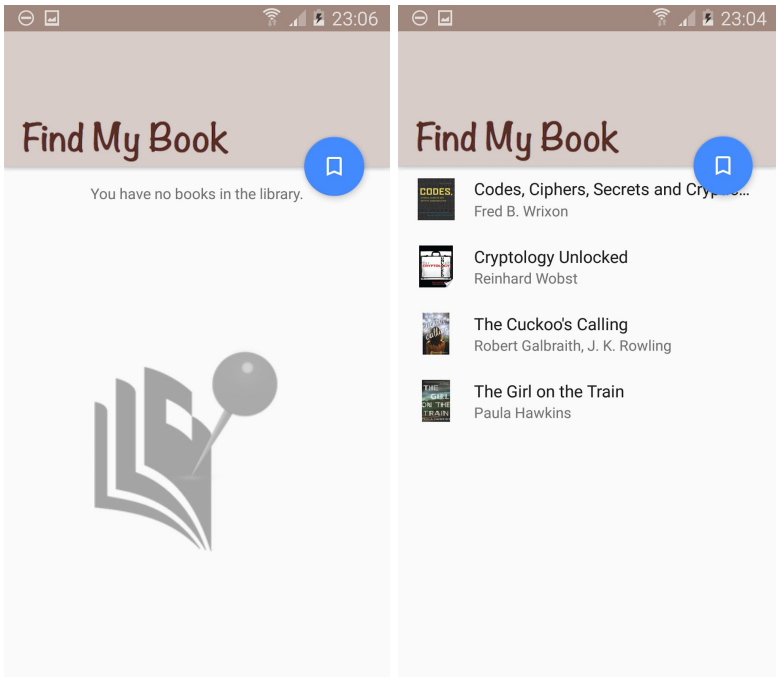
Features

- Scans ISBN barcode
- Queries library for book information
- Saves book information of scanned ISBN in list
- Shows nearest library retrieved book is available
- Displays map of library location

User Interface Mocks

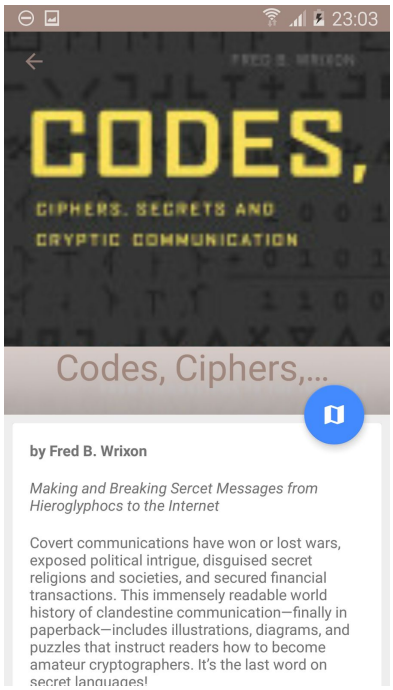
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

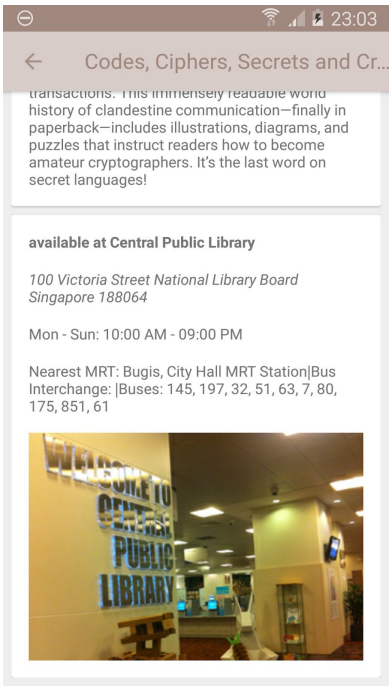
Screen 1



Main activity screen whereby user can see the list of books he had scanned or a message if the list is empty. Click on the “bookmark” floating action button to initiate barcode scanning of ISBN. Click on a list item to view a book’s details. Swipe a list item to delete a book.

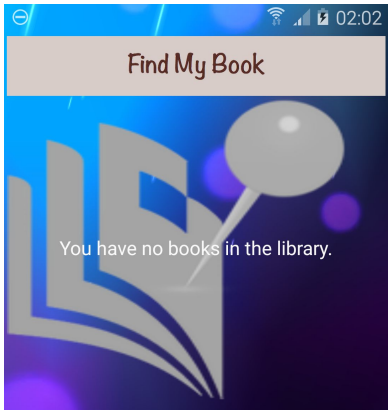
Screen 2

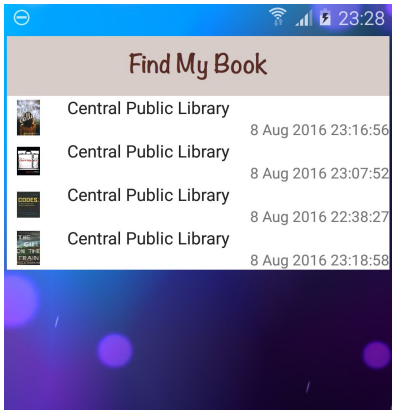


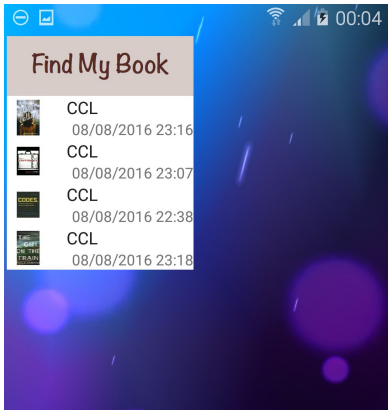


Detail activity screen whereby user can view information regarding selected book from list. The information will include the nearest national library branch’s name that the book is available. Click on the “map” floating action button (shown only if book is available) will open the library branch’s location in Google Maps.

Screen 3







Home screen AppWidget showing a summary list of the books scanned and the last library locations updated. The resized smaller widget will show the library codes in place of their full

names. Click on the widget title to open the app from start or click on the list item to open the detail activity to see the book's information.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

I will have a content provider for two database tables: books and libraries. The "books" table is for storing the books' information that the user retrieved via Google API. The "libraries" table is for storing the library branches' information retrieved from the National Library Board's web and data services.

Describe any corner cases in the UX.

No corner case is foreseen at the moment.

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

The set of Android support libraries for compatibility and material design I will be using are:

- com.android.support:appcompat-v7
- com.android.support:cardview-v7
- com.android.support:design
- com.android.support:percent
- com.android.support:support-v4

The third-party libraries I will be using are:

- com.google.android.gms:play-services
  - Google play services for getting current location
- com.google.zxing:core
  - ZXing library for scanning barcodes
- com.journeyapps:zxing-android-embedded
  - Library for embedding the barcode scanning functionality inside the app instead of having the need to call the ZXing app via intent
- com.github.jorgecastilloprz:fabprogresscircle
  - Wrapper library for having a progress circle around the floating action button
- com.github.bumptech.glide:glide
  - Library for downloading the book covers and library images

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

Subtask 1: Git clone and import into app project as an Android library module

1. Read reference https://developer.android.com/training/volley/index.html
2. Follow instructions in https://developer.android.com/studio/projects/android-library.html

Subtask 2: Configure classpaths and dependencies for libraries

1. Add the following line in the project's "build.gradle" file, inside dependencies:

   classpath 'com.google.gms:google-services:3.0.0'

2. Add the following line to the end of the app module's "build.gradle" file:

   apply plugin: 'com.google.gms.google-services'

3. Add the following lines in the app module's "build.gradle" file, inside dependencies:

   compile 'com.android.support:appcompat-v7:24.0.0'
   compile 'com.android.support:cardview-v7:24.0.0'
   compile 'com.android.support:design:24.0.0'
   compile 'com.android.support:percent:24.0.0'
   compile 'com.android.support:support-v4:24.0.0'
   compile 'com.github.bumptech.glide:glide:3.7.0'
   compile 'com.github.jorgecastilloprz:fabprogresscircle:1.01@aar'
   compile 'com.google.android.gms:play-services:9.2.1'
   compile 'com.google.firebase:firebase-ads:9.0.2'
   compile 'com.google.zxing:core:3.2.1'
   compile 'com.journeyapps:zxing-android-embedded:3.3.0@aar'

Subtask 3: Request API key from National Library Board

1. Go to http://www.nlb.gov.sg/labs/mash-create-collaborate/
2. Download and complete the request form

Task 2: Implement UI for Each Activity and Fragment

Subtask 1: Build UI for MainActivity and MainFragment

- Use Toolbar and RecyclerView to display a list of books or a message and logo if list is empty.

Subtask 2: Build UI for DetailActivity and DetailFragment

- Use CollapsingToolbarLayout, PercentLayoutHelper and CardView to build display for detailed information of selected book.
- Show library information in CardView.

Subtask 3: Build UI for SplashActivity and SplashTaskFragment

- Show a splash screen of app's logo while loading library branches' information into database.
- Use a UI-less fragment to load data from web service and save to database.

Task 3: Implement Database Classes

Build the set of classes for accessing the database and storing of data. The list of subtasks are:

1. Create a class that extends SQLiteOpenHelper.
2. Create a data contract class.
3. Create a content provider.
4. Create a CursorAdapter for the RecyclerView for the list of books.

Task 4: Implement Intent Service and Receiver

Build the set of classes to retrieve book information, save information in database, and notify the adapter of the update, all in the background. The list of subtasks are:

1. Create BookIntentService to retrieve information from Google Book API.
2. Create BookResultReceiver to notify adapter of database saves.

Task 5: Implement Location Services

Build the set of classes for getting user's current location. The list of subtasks are:

1. Specify app permissions to request location permissions.
2. Connect to Google Play Services.
3. Get the Last Known Location i.e. current location since last query, periodically.
4. Compute distance between current location against list of library branches as necessary (when a selected book is available in the returned list).

Task 6: Implement Utility Classes

Build the set of utility classes for processing RSS feed, validating ISBN, etc. The list of subtasks are:

1. Create a class to validate ISBN-13 and convert from ISBN-10 to ISBN-13.
2. Create a class to process RSS data on library branches' information.
3. Create a class to compute nearest library from current location based on GPS coordinates.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"