

Project 4 Task 1 – Book Master App

by Xiaotong Luo (xluo2)

Description:

My application will take a search string from the user and then uses it to fetch information from Google Books and display a list (only include five books) of relevant books' information, including a thumbnail picture, book title, author, publication time, and a hyperlink that can direct the user to the home page of the book for more information.

Here is how my application meets the task requirements.

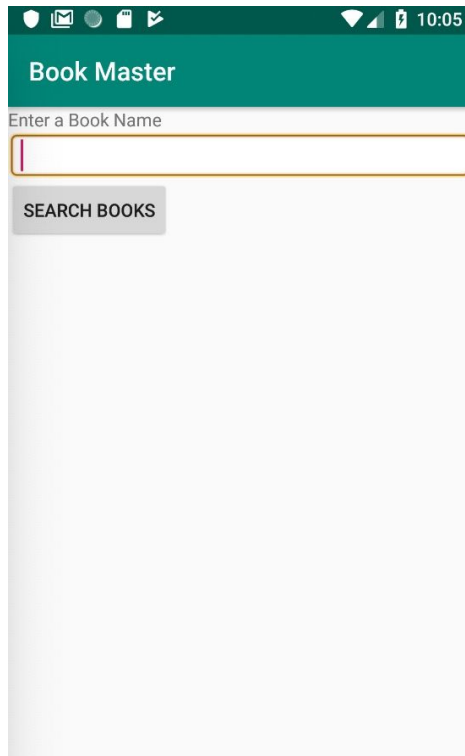
1. Implement a native Android application

The name of my native Android application project in Android Studio is: Project4Android

1.1. Has at least two different kinds of views in your Layout (TextView, EditText, ImageView, etc.)

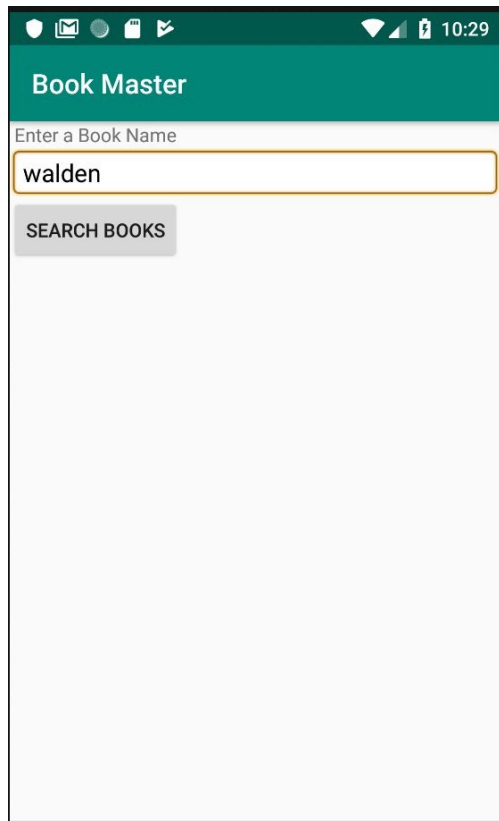
My application uses TextView, EditText, Button, and ImageView. See activity_main.xml for details of how they are incorporated into the LinearLayout and ScrollView.

Here is a screenshot of the layout before the user searches for books.



1.2. Requires input from the user

Here is a screenshot of the user searching for information of "walden".



1.3. Makes an HTTP request (using an appropriate HTTP method) to your web service

My application does an HTTP GET request in BookSearchModel.java. The HTTP request is: "https://stormy-hamlet-89103.herokuapp.com/BookServlet/" + searchTerm

where searchTerm is the user's search term.

The search method will make this request of my web application, parses the returned JSON to get 5 books' information (including a thumbnail picture url, book title, author, publication time, and an information link) and returns the information in a String.

In the callback method (dataReady) in BookSearch.java, the information of title, author and publication time will be displayed in TextView. The thumbnail pictures will be fetched in AsyncLoadImage.java and displayed in imageView in the callback method (picReady) in BookSearch.java. The information link will be attached to a button and the button will have an onclick event listener to direct the user to the homepage of the book when the user clicks the button.

1.4. Receives and parses an XML or JSON formatted reply from the web service

An example of the JSON reply from the web service is:

```
{
  "size": 5,
  "bookList": [{
    "title": "Walden",
    "author": "Henry David Thoreau",
    "imgURL":
"http://books.google.com/books/content?id=yiQ3AAAAIAAJ&printsec=frontcover&img=1&zoom=1&edge=
curl&source=gbs_api",
    "publishedTime": "1910",
    "infoURL":
"http://play.google.com/store/books/details?id=yiQ3AAAAIAAJ&source=gbs_api"
  }, {
    "title": "Walden",
    "author": "Henry David Thoreau",
    "imgURL":
"http://books.google.com/books/content?id=aCQ6AQAAIAAJ&printsec=frontcover&img=1&zoom=1&edge
=curl&source=gbs_api",
    "publishedTime": "1899",
```

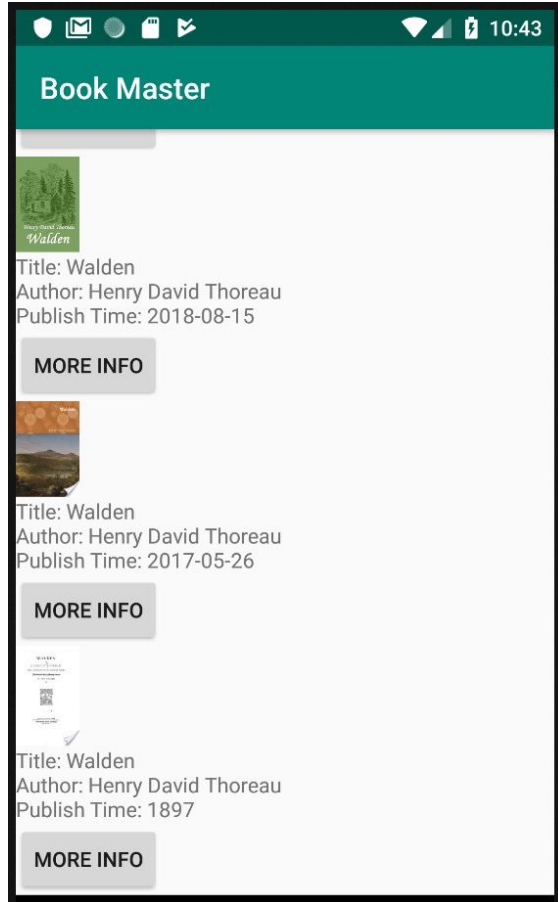
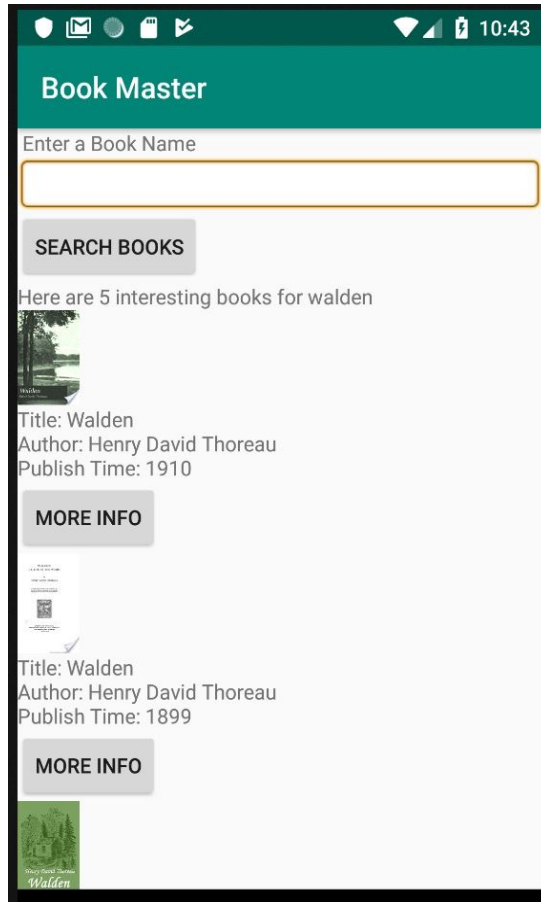
```

        "infoURL":
"https://play.google.com/store/books/details?id=aCQ6AQAAIAAJ&source=gbs_api"
    }, {
        "title": "Walden",
        "author": "Henry David Thoreau",
        "imgURL":
"http://books.google.com/books/content?id=98i3ugEACAAJ&printsec=frontcover&img=1&zoom=1&source=gbs_api",
        "publishedTime": "2018-08-15",
        "infoURL":
"http://books.google.com/books?id=98i3ugEACAAJ&dq=walden&hl=&source=gbs_api"
    }, {
        "title": "Walden",
        "author": "Henry David Thoreau",
        "imgURL":
"http://books.google.com/books/content?id=xnInDwAAQBAJ&printsec=frontcover&img=1&zoom=1&edge=cur&source=gbs_api",
        "publishedTime": "2017-05-26",
        "infoURL":
"http://books.google.com/books?id=xnInDwAAQBAJ&dq=walden&hl=&source=gbs_api"
    }, {
        "title": "Walden",
        "author": "Henry David Thoreau",
        "imgURL":
"http://books.google.com/books/content?id=hnJbAAAAMAAJ&printsec=frontcover&img=1&zoom=1&edge=cur&source=gbs_api",
        "publishedTime": "1897",
        "infoURL":
"https://play.google.com/store/books/details?id=hnJbAAAAMAAJ&source=gbs_api"
    }
}

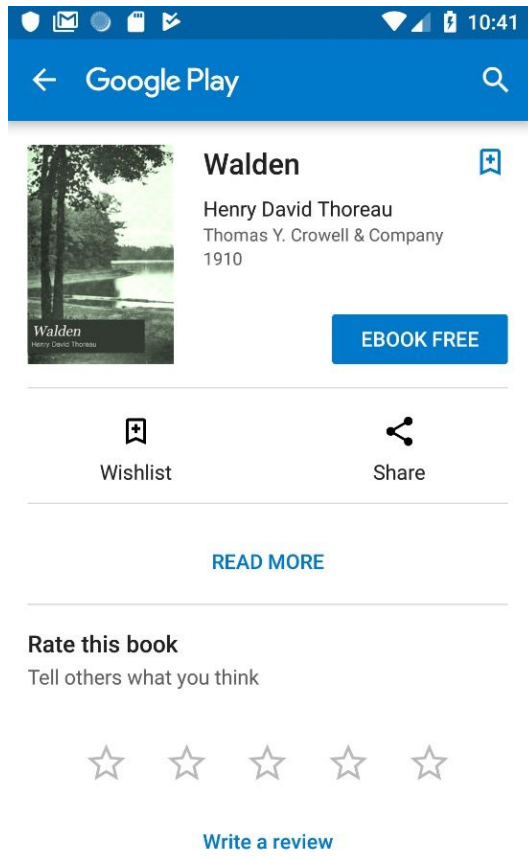
```

1.5. Displays new information to the user

Here is the screen shots after all the information has been displayed.

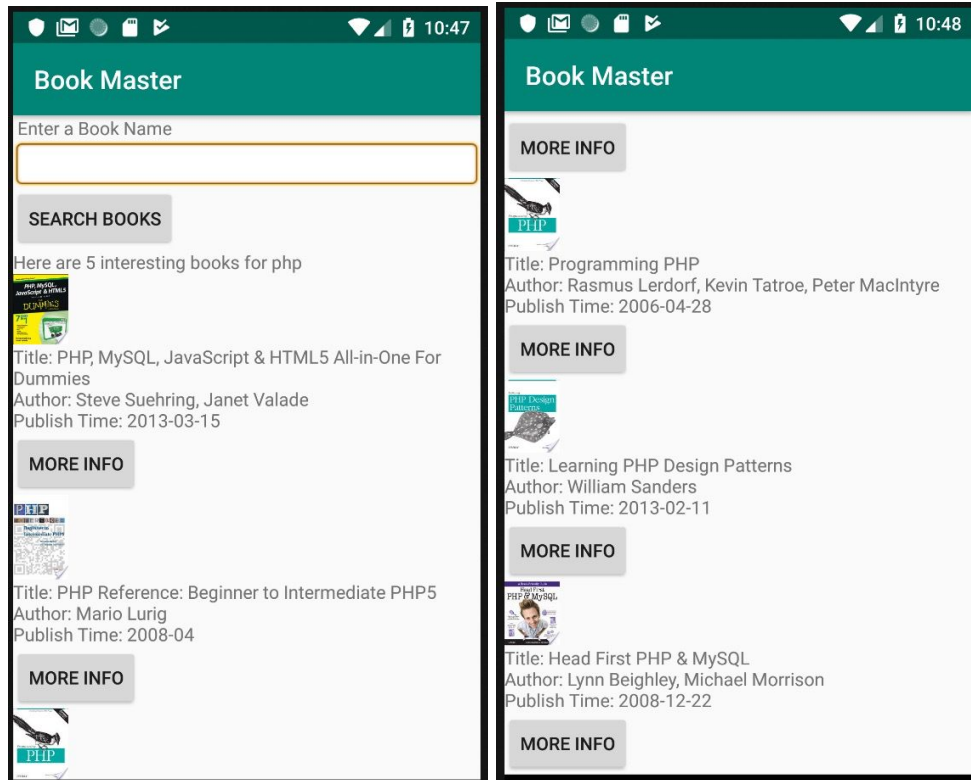


When the user clicks on the first **MORE INFO** button, the user will be redirected to the homepage of the first book:



1.6. Is repeatable (i.e. the user can repeatedly reuse the application without restarting it.)

The user can type in another search term and hit SEARCH BOOKS. Here is an example of having typed in "php".



2. Implement a web application, deployed to Heroku

The URL of my web service deployed to Heroku is: stormy-hamlet-89103

2.1. Using an HttpServlet to implement a simple (can be a single path) API

In my web app project:

Model: BookModel.java

Controller: BookServlet.java

2.2. Receives an HTTP request from the native Android application

BookServlet.java receives the HTTP GET request with the search term. It passes this search term on to the model.

2.3. Executes business logic appropriate to your application

BookServlet.java makes an HTTP request to:

<https://www.googleapis.com/books/v1/volumes?q=search&maxResults=5>, where “search” is the search term. It then parses the JSON response and extracts the parts it needs to respond to the Android application.

2.4. Replies to the Android application with an XML or JSON formatted response

BookModel.java extracts the information and formats the response to the mobile application in a simple JSON format of my own design.

Example:

```
{
  "size": 5,
  "bookList": [{
    "title": "Walden",
    "author": "Henry David Thoreau",
    "imgURL":
"http://books.google.com/books/content?id=yiQ3AAAAIAAJ&printsec=frontcover&img=1&zoom=1&edge=
curl&source=gbs_api",
    "publishedTime": "1910",
    "infoURL":
"http://play.google.com/store/books/details?id=yiQ3AAAAIAAJ&source=gbs_api"
  }, {
    "title": "Walden",
    "author": "Henry David Thoreau",
    "imgURL":
"http://books.google.com/books/content?id=aCQ6AQAAIAAJ&printsec=frontcover&img=1&zoom=1&edge=
curl&source=gbs_api",
    "publishedTime": "1899",
    "infoURL":
"http://play.google.com/store/books/details?id=aCQ6AQAAIAAJ&source=gbs_api"
  }, {
    "title": "Walden",
    "author": "Henry David Thoreau",
    "imgURL":
"http://books.google.com/books/content?id=98i3ugEACAAJ&printsec=frontcover&img=1&zoom=1&source=
gbs_api",
    "publishedTime": "2018-08-15",
    "infoURL":
"http://books.google.com/books?id=98i3ugEACAAJ&dq=walden&hl=&source=gbs_api"
  }, {
    "title": "Walden",
    "author": "Henry David Thoreau",
```



```

        "imgURL":
"http://books.google.com/books/content?id=xnInDwAAQBAJ&printsec=frontcover&img=1&zoom=1&edge
=curl&source=gbs_api",
        "publishedTime": "2017-05-26",
        "infoURL":
"http://books.google.com/books?id=xnInDwAAQBAJ&dq=walden&hl=&source=gbs_api"
    }, {
        "title": "Walden",
        "author": "Henry David Thoreau",
        "imgURL":
"http://books.google.com/books/content?id=hnJbAAAAMAAJ&printsec=frontcover&img=1&zoom=1&edge
=curl&source=gbs_api",
        "publishedTime": "1897",
        "infoURL":
"https://play.google.com/store/books/details?id=hnJbAAAAMAAJ&source=gbs_api"
    }
}

```

Reference:

[1] Interesting Picture Android Application

[2] Load image asynchronously in Android:

<https://www.concretepage.com/android/android-load-image-from-url-with-internet-using-bitmapfactory-and-imageview-setimagebitmap-example>

[3] Attach a url link to a button in Android:

<https://stackoverflow.com/questions/4930228/open-a-url-on-click-of-ok-button-in-android>

[4] Get resource ID from string:

<https://stackoverflow.com/questions/4427608/android-getting-resource-id-from-string>