

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Home Screen](#)

[Book Detail Screen](#)

[Authors Screen](#)

[Author Detail Screen](#)

[Navigation Drawer](#)

[Settings Screen](#)

[Tablet Screens](#)

[Wishlist Widget](#)

[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: Project Setup](#)

[Task 2: Create a Content Provider](#)

[Task 3: Implement UI for Home Screen](#)

[Task 4: Implement the UI for Book details Screen](#)

[Task 5: Implement the UI for the Authors Screen](#)

[Task 6: Implement the UI for the Author Detail Screen](#)

[Task 7: Implement the UI for the Wishlist screen](#)

[Task 8: Implement the Navigation Drawer](#)

[Task 9: Implement the Settings Screen](#)

[Task 10: Setup Google Analytics for Wishlist and Authors Screen](#)

[Task 11: Setup Google Identity for User](#)

**GitHub Username: sengopal**

# Author Follow

## Description

An avid book reader is always on the lookout for new releases especially from the authors that they love. However, there is no such tool today which informs you about upcoming releases. "Author Follow" allows you to keep tabs on upcoming and recent releases of your favorite book authors, search for new authors and more. This app uses Amazon's product release API to scour for their favorite authors' publications.

Some of the highlights of this app are:

- Offline wishlist to save books
- Goodreads integration
- and many more..

## Intended User

This app is intended for book readers who wish to follow new and upcoming releases of their favorite authors.

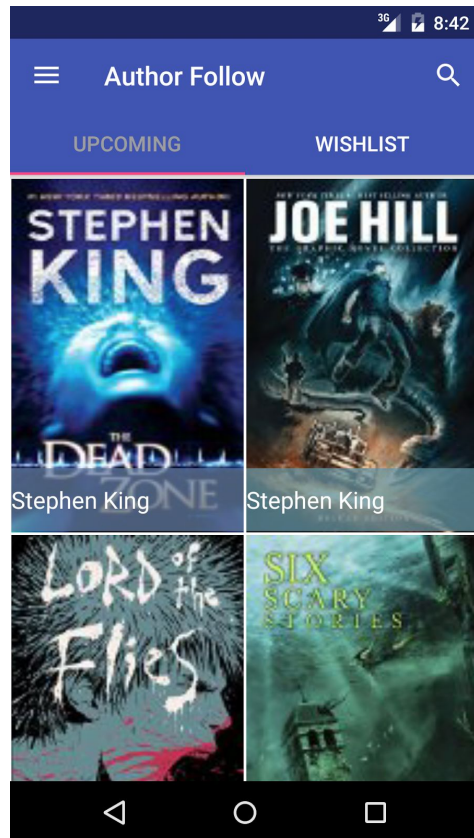
## Features

This app comes with a tons of features such as:

- Search and save authors to follow
- Shows recent and upcoming releases
- View information about authors and books such as reviews, ratings etc,,
- Allows books to be stored to a wishlist
- Goodreads integration
- Google Identity API integration
- Google Analytics API integration
- Widget to display wishlist

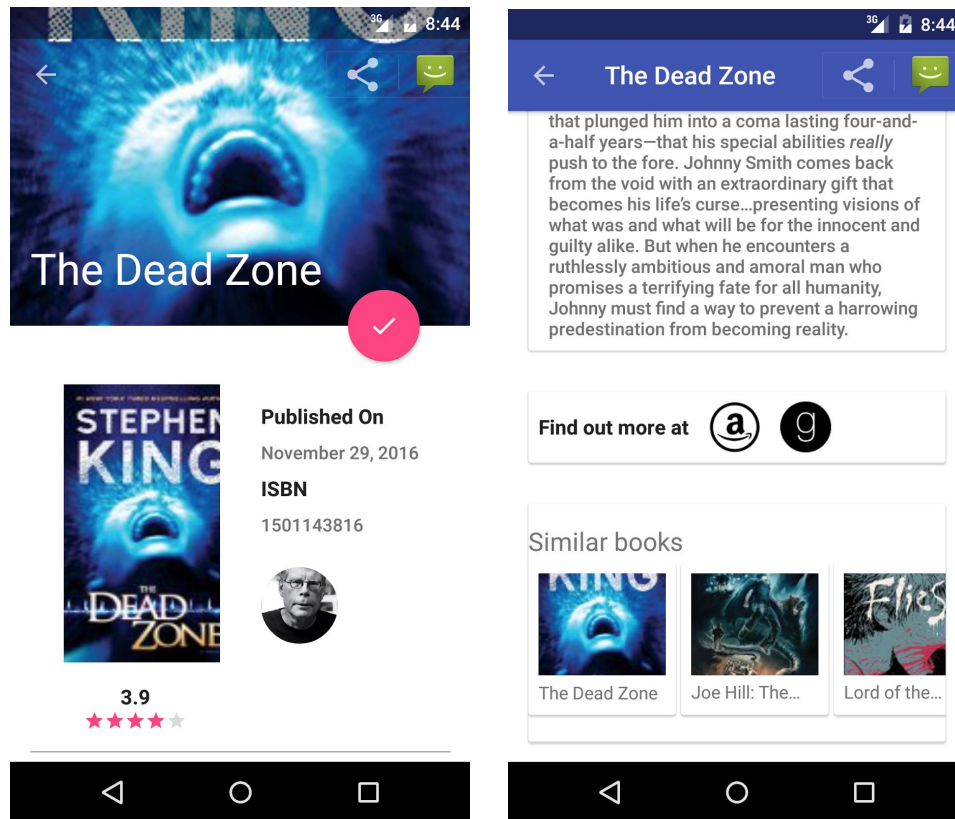
## User Interface Mocks

### Home Screen



The home screen displays the Upcoming releases and Wishlist for all the followed authors. In case there are no results, there will be a message to add more authors to follow. The user can also use the search from the toggle bar to search for books or authors from the list. The Wishlist screen displays the books and their details that have been added as a wish list item.

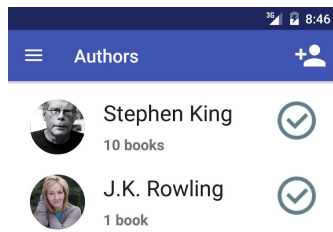
## Book Detail Screen



The book details screen can be triggered from multiple places such as the Home, Wishlist, Goodreads shelf etc., This screen offers rich detail including the book cover, Goodreads rating, release information, description etc.,

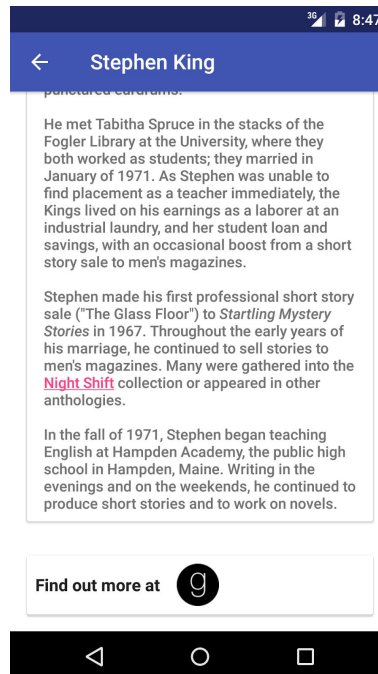
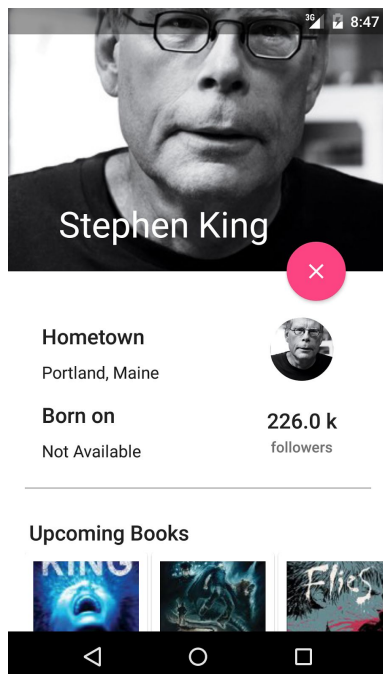
The Fav button helps add/remove the book to the Wishlist.

## Authors Screen



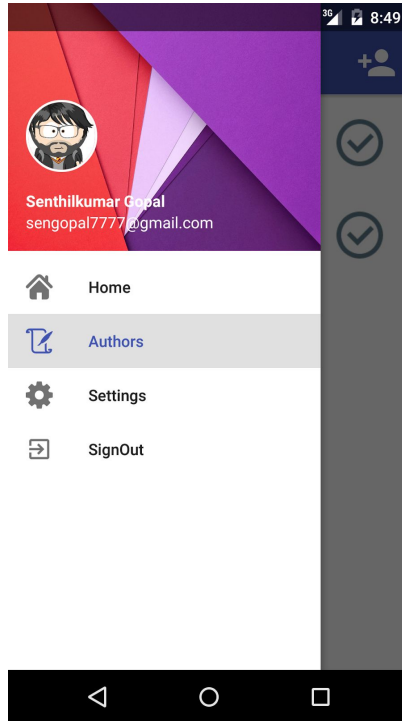
The Authors screen allows the user to maintain the list of followed authors, search for new authors and add to the followed list.

## Author Detail Screen



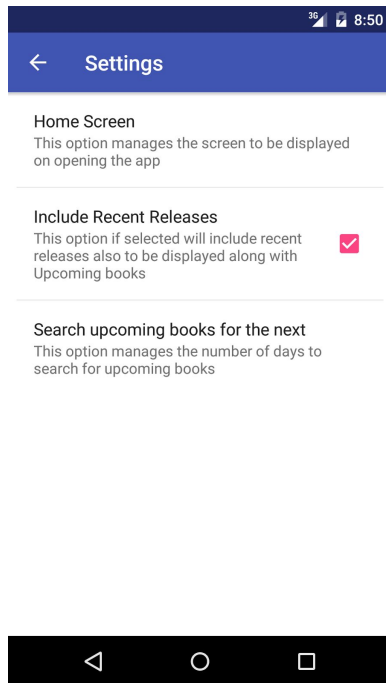
The Author details screen provides important information about the author, cover picture, their Goodreads rating, some of their upcoming books etc., The fav button helps add the author to the follow list.

## Navigation Drawer



The navigation drawer helps identify the different pages/screens within the app and also gives access to the settings screen.

## Settings Screen

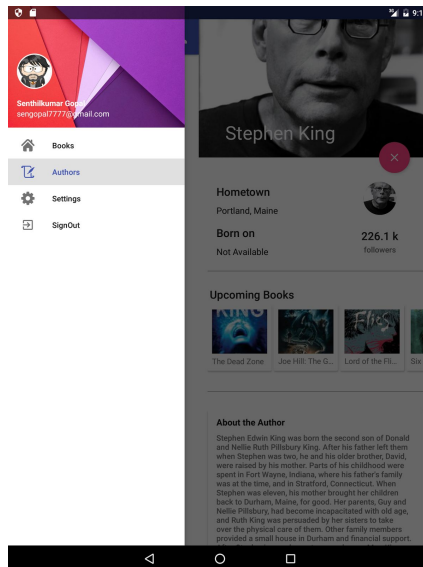
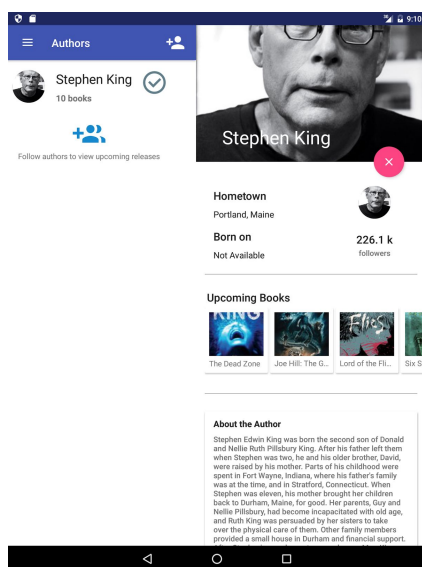
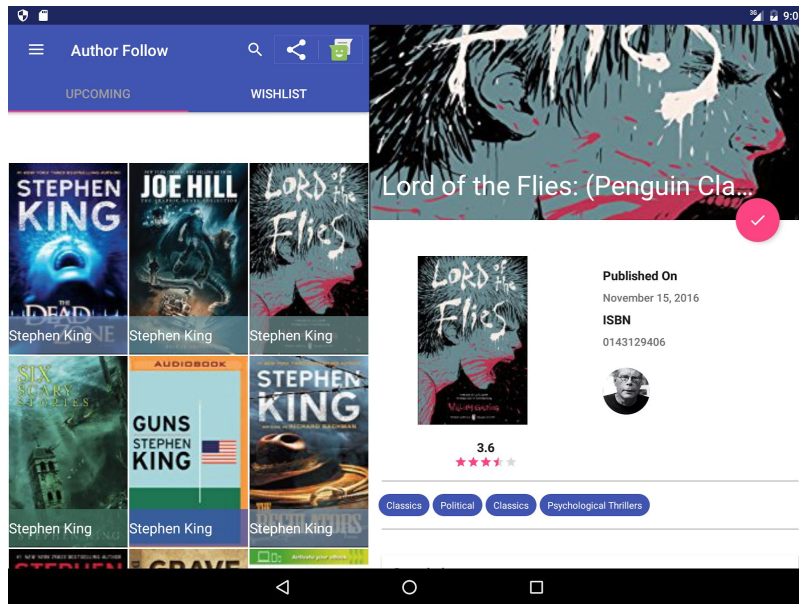


The Settings screen helps in configuring some of the user preferences such as:

- Include recent releases as part of results.
- Configuration for how many days to look forward for upcoming releases
- Choose initial screen as either Books/Authors

## Tablet Screens

All the above screens will be implemented with its equivalent Tablet screens. Some of the screen mocks are provided below as examples.





## Wishlist Widget



The app also provides a widget which helps the user a quick peek into the books added to the user's wishlist.

## Key Considerations

### How will your app handle data persistence?

Most of the data will be persisted using a Content Provider and using a Sync Adapter the data will be updated whenever the network is available. This allows the user to use the app even in offline mode.

The Content provider will also help for Notifications and Widget.

### Describe any corner cases in the UX.

The corner cases for this app are:

1. Empty data screen where no authors have been added to follow
2. No recent or upcoming releases
3. No network available for viewing/adding to Goodreads shelf.

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

The following libraries will be used:

1. **Retrofit** for making network calls
2. **Picasso** for image loading
3. Android support design library
4. **ButterKnife** for injection
5. **CircleImageView**
6. **SimpleXML** with Retrofit for reading XML APIs
7. **Google Analytics and Auth**
8. **ActiveAndroid** for content provider

## Next Steps: Required Tasks

### **Task 1: Project Setup**

The first setup is to get Android studio up-to-date and create a master detail setup with all the libraries configured with build.grade

### **Task 2: Create a Content Provider**

Create a Content provider with ActiveAndroid which provides Books and Authors. The Java model will have identifiers such as cover image, release date, wishlist-added etc.,

### **Task 3: Implement UI for Home Screen**

Implement the UI screens for the Home Activity/Fragment and ensure that the data from the Books ContentProvider is displayed. Also implement the search within in the App toolbar.

### **Task 4: Implement the UI for Book details Screen**

Implement the UI screens for the Book details screen and setup such a way that the intent uses a common interface or uses the Content provider URL scheme to load the book details. Use Picasso for Image load of the book cover and also Async response for the books rating.

### **Task 5: Implement the UI for the Authors Screen**

Implement the UI for the Authors screen where the user can view the Authors added and their book release count (*total of recent and upcoming releases*). The users can search for new authors by name and add to the list

### **Task 6: Implement the UI for the Author Detail Screen**

Implement the UI screens for the Author details screen and setup such a way that the intent uses a common interface or uses the Content provider URL scheme to load the Author details. Use Picasso for Image load of the released books and also Async response for the author rating.

### **Task 7: Implement the UI for the Wishlist screen**

Implement the UI screens for the Wishlist screen which will display the Books which are marked with "wishlist-add". Each item has a context menu to get removed from the Wishlist. Clicking on the book initializes the Book details screen.

### **Task 8: Implement the Navigation Drawer**

Implement the Navigation drawer for showing the User cover picture and navigation links to other screens in the app.

### **Task 9: Implement the Settings Screen**

Implement the Settings screen to setup preferences to show notifications, days to check for release etc.,

### **Task 10: Setup Google Analytics for Wishlist and Authors Screen**

implement Google Analytics to learn how many users add a specific Author to follow and also which books are added most to Wishlist.

### **Task 11: Setup Google Identity for User**

Implement Google Identity to login the user and associate the data to a specific user of the device. If the device is multi user, then this helps in associating the data with each specific account.

### **RUBRIC COMPLIANCE**

1. Third-party libraries integrated
2. Widget available
3. Google Services- Identity and Analytics
4. ContentProvider via ActiveAndroid
5. Loaders and AsyncTasks used

### **Optional Components**

1. Material design patterns such as shared element transitions across activities and parallax scrolling.
2. Implement sharing functionality, making use of ShareActionProvider
3. Custom views for widget.