

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[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.](#)

[Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Create SQLite Database structure](#)

[Task 3: Create Content Provider](#)

[Task 4: Create Main Activity and Fragment](#)

[Task 5: Create Detail Activity and Fragment](#)

[Task 6: Create New Item Entry Activity](#)

**GitHub Username:** johnshea

## Borrowed It

### Description

In a simple and efficient way, this app will help you keep track of the items you have borrowed and the items you have lent out.

Users can quickly add items and quit worrying that they will forget.

All items can have a due date assigned so you avoid keeping track of that information in your head.

For items you borrow, the app will keep track of the item borrowed, from whom it was borrowed, due date for returning the item (if there is one such as with a library item), an image of the item (if the user wishes), where the item is located (to help you easily find it to return it), and general notes (eg. why you borrowed it, did they lend you something in return, etc.)

For an item you lent out, it will keep track of the item lent, to whom it was lent, when they expect to return it, an image of the item, and general notes (eg. why they borrowed it, what they gave you in return, etc.)

## Intended User

The intended audience for this app is anyone who borrows and/or lends any items and would like a simple and efficient way to keep track of them.

They can keep track of items that are primarily meant for borrowing such as library books or rental DVDs.

They can also keep track of personal items that they have borrowed from family and friends or items they have lent out to others.

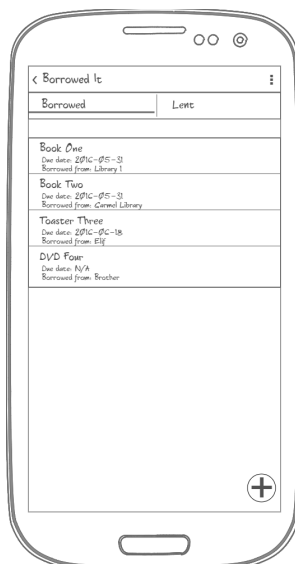
## Features

The main features of this app will be to:

- List the items borrowed
- List the items lent
- Collection widget to show items with upcoming due dates
- Take pictures of item
- Use of Google Play services API (Google Location and Google Map) to help keep track of the borrowed item location or the location it was borrowed from (eg. library location)

## User Interface Mocks

### Screen 1

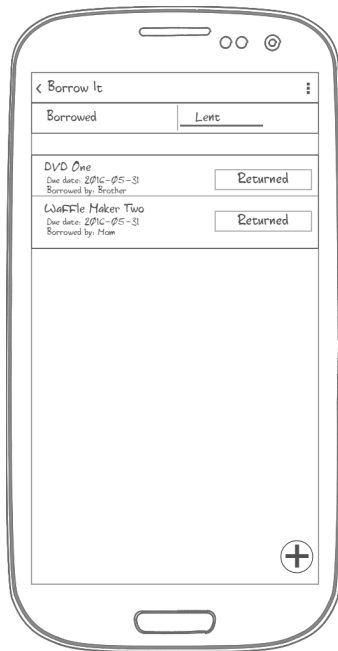


This is the main screen the user will see when the app opens. It contains two tabs (Borrowed, Lent) consisting of the items relevant to the selected tab.

From here the user can click on an item which will open a details screens.

Using the floating action button (FAB), they can add a new entry to the list for item they borrowed.

## Screen 2



This is another view of the main screen with the “Lent” item tab selected.

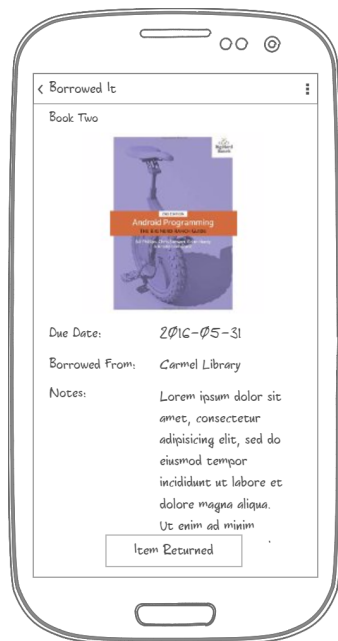
The FAB will allow the user to create a new entry for an item that they are lending out.

### Screen 3



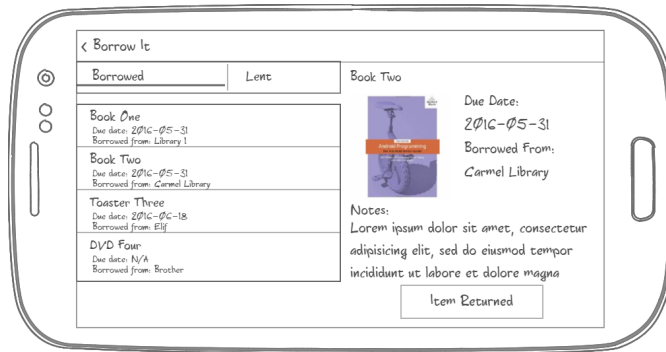
This is the item entry screen for adding a newly borrowed item.

### Screen 4



Detail screen - When a user selects an item from the main list, it will then open up a details screen which contains more information about the selected item.

## Screen 5



The screen depicts a landscape tablet view which shows the main fragment and the detail fragment positioned side-by-side.

## Key Considerations

### How will your app handle data persistence?

The app will store all item information in a local SQLite database. The app will use a Content Provider as an interface with the database.

### Describe any corner cases in the UX.

The collection widget will display a message that no items are currently due instead of displaying an empty view.

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

Picasso/Glide - loading of images, Reason: Using an image loading library simplifies the code required to perform this function. These libraries also use caching and have the built-in capability to load images off of the main thread.

Android Design Support Library - floating action button (FAB), tabs

Android v4 Support Library - fragment, ViewPager

Android v7 Support Library (appcompat) - ActionBar, AppCompatActivity

Android v7 Support Library (recyclerview) - RecyclerView

## Required Tasks

### Task 1: Project Setup

- Create initial project set-up.
- Add the Picasso/Glide library entry.
- Add Google Design and Support library entries
- Add Google Play services API for Google Location and Google Map.

### Task 2: Create SQLite Database structure

- Create database structure used to store items.

### Task 3: Create Content Provider

- Create and test Content Provider.
- Seed Content Provider with data. This will make working on the UI easier as it will populate the views.

### Task 4: Create Main Activity and Fragment

- Create main activity with fragment
- Implement item loader
- Wire up loader and main list

### Task 5: Create Detail Activity and Fragment

- Create detail activity with fragment
- Load UI with data based on passed in item ID

### Task 6: Create New Item Entry Activity

- Create UI
- Save new entry to database
- Access Location service to get lat/lon of user
- Access Map and place marker at the lat/lon