GitHub Username: javierarboleda

Super Comic Reader!

Description

Problem:

There are some comic reader apps that provide an immersive panel-to-panel reading experience. However, most of these apps do not support DRM-free comics, i.e., you cannot use this app to read comics you purchase outside of that app's store. Furthermore, none of these apps let the user create their own panel-to-panel experiences for their favorite comics. And what are you to do if your favorite comics are not supported?

Proposed Solution:

Provide an app to users that allows them to create the panel-to-panel experience of any comic out there! The user can then read their comic using the panel-to-panel experience they created, and they can save the data for their unique panel-to-panel reading experience and share it with the community.

Intended User

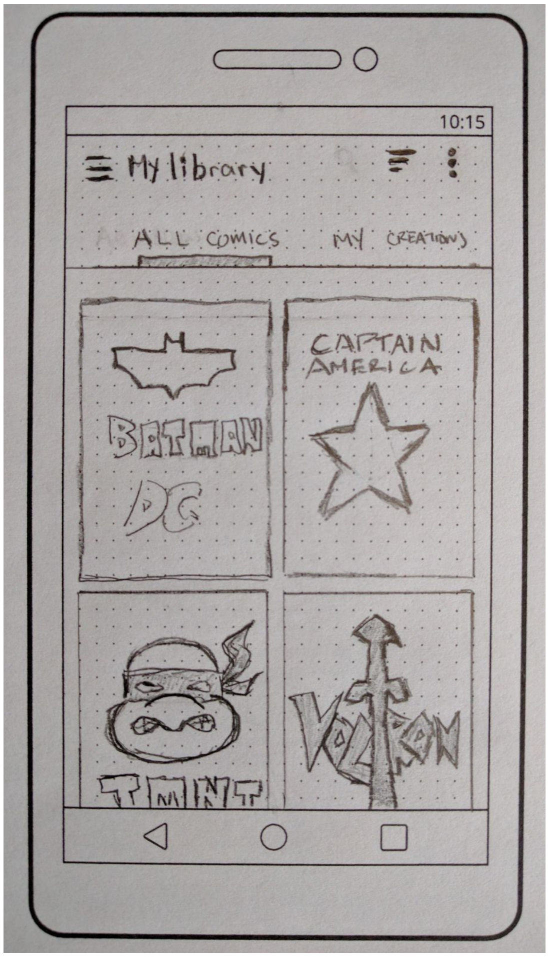
Comic book readers. Comic book enthusiasts. Users that enjoy creating and sharing with communities. Users that don’t want the rich reading experience of major comic book apps but with freedom to have that feature for comics that are DRM free or obscure, less popular comics that aren’t supported in major comic apps.

Features

- Load comic file to library to read and/or create a panel-to-panel experience
- Create a panel-to-panel experience for a comic issue
    - Save each panel along with optional border
    - Save file that contains all of the saved panels
- Load a panel-to-panel file to a matching comic in library
- Read comics with a panel-to-panel experience

User Interface Mocks

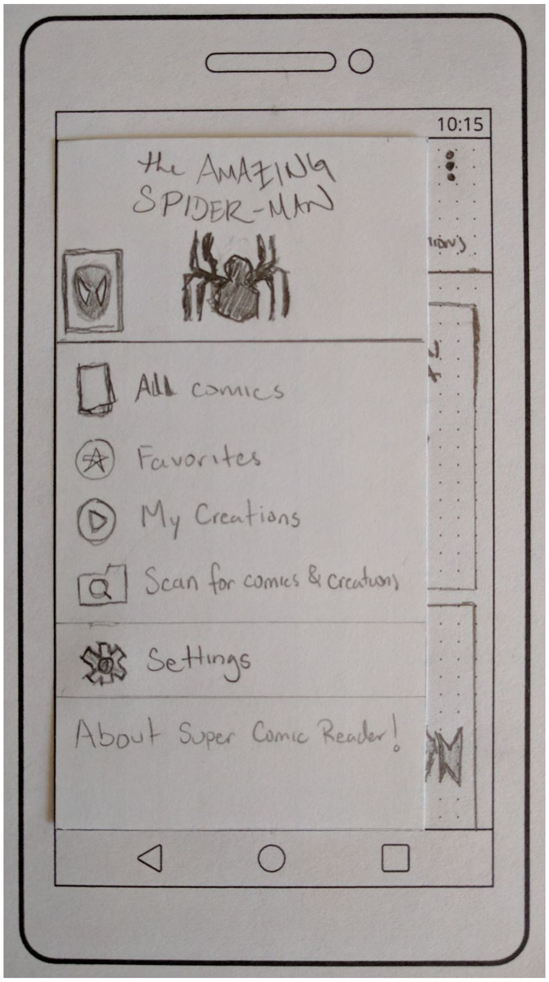
Screen 1: Home Screen



This is the home screen which will have a scrollable grid view of user's comics. There will be tabs, one that shows all comics, and one that will show only those comics that you have created panel-to-panel experiences.

Clicking on any of the comic covers will take you to the details page of that comic.

Screen 2: Navigation Drawer

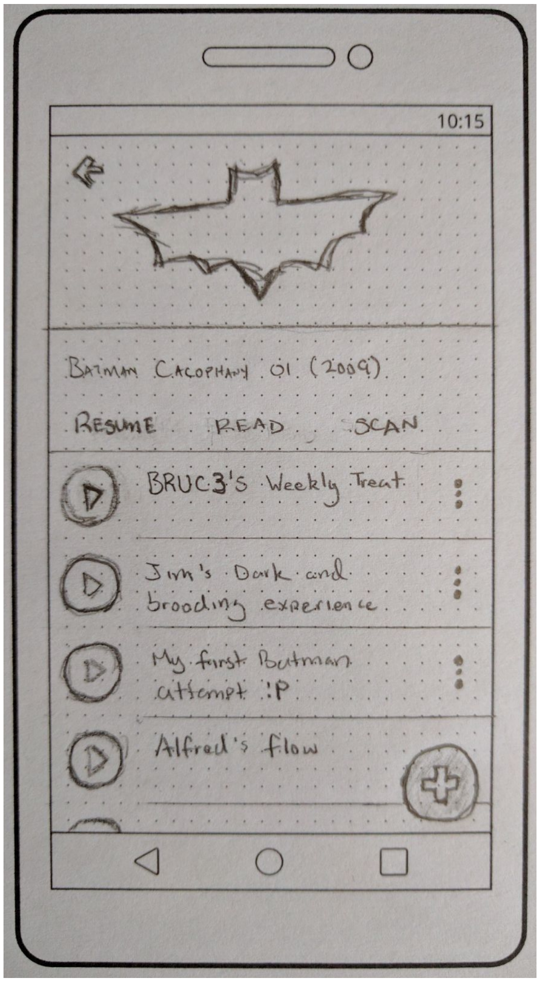


Screen of what the navigation drawer will contain. Top banner has a customizable background and avatar that the user can set from images on their device.

Next you have navigation to **All Comics**, **Favorites**, and **My Creations** sections of app.

Scan for comics & creations option will scan device for new comics and creations that have not been added yet to the library.

Screen 3: Comic Detail


10:15
BATMAN CACOPHANY 01 (2009)
RESUME
READ
SCAN
BRUC3'S Weekly Treat
Jim's Dark and brooding experience
My first Batman attempt :P
Alfred's flow

The comic detail screen has a top banner with a scaled/cropped portion of comic cover. Under it there is the comic title, as well as three buttons:

Resume
This will resume where ever the user left off when last reading this comic. Will be grayed out if user hasn’t started a readthrough of this comic.

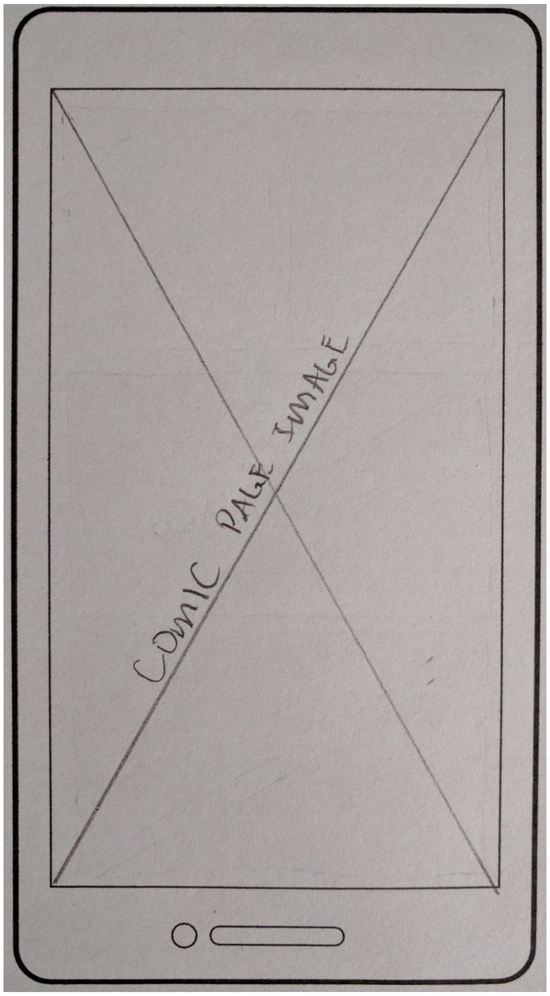
Read
This will begin a comic read with no panel-to-panel play, so like free read option, user zooms and swipes to new pages as they please.

Scan
This will scan the device for panel-to-panel experiences for this comic.

There is then a list of panel-to-panel creations for this comic that have been added.

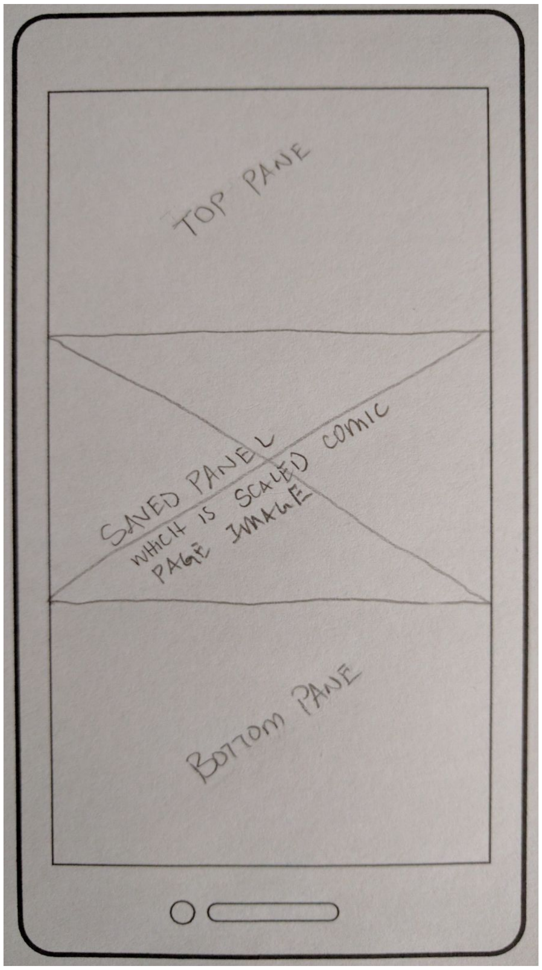
Finally, there is a FAB which will be used for user to begin process of creating a panel-to-panel creation.

Screen 4: Comic Free Read



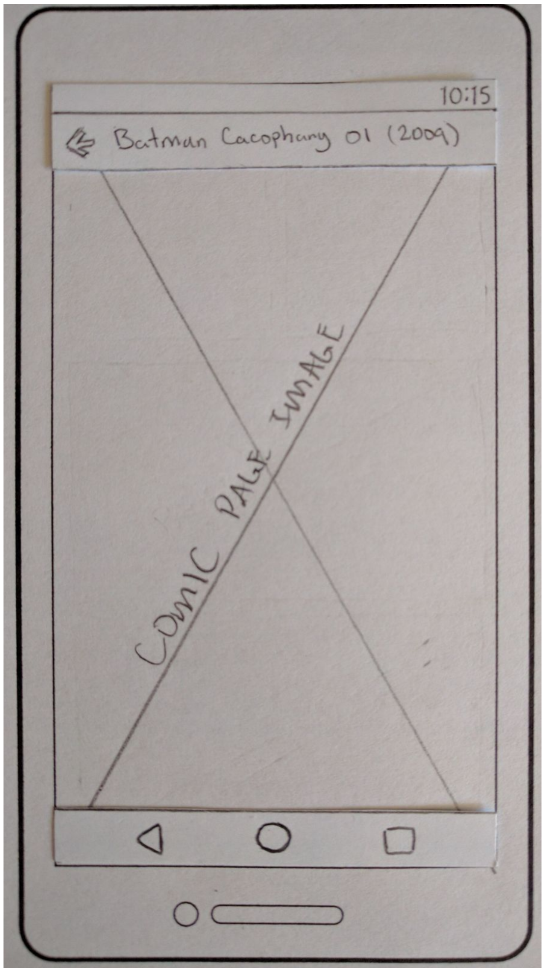
This is a screen showing that when the user is reading the comic, it takes up the entire screen. The user can swipe left or right or tap on left or right of screen to change pages (images) and pinch to zoom in/out.

Screen 5: Comic Panel-to-Panel Experience Read



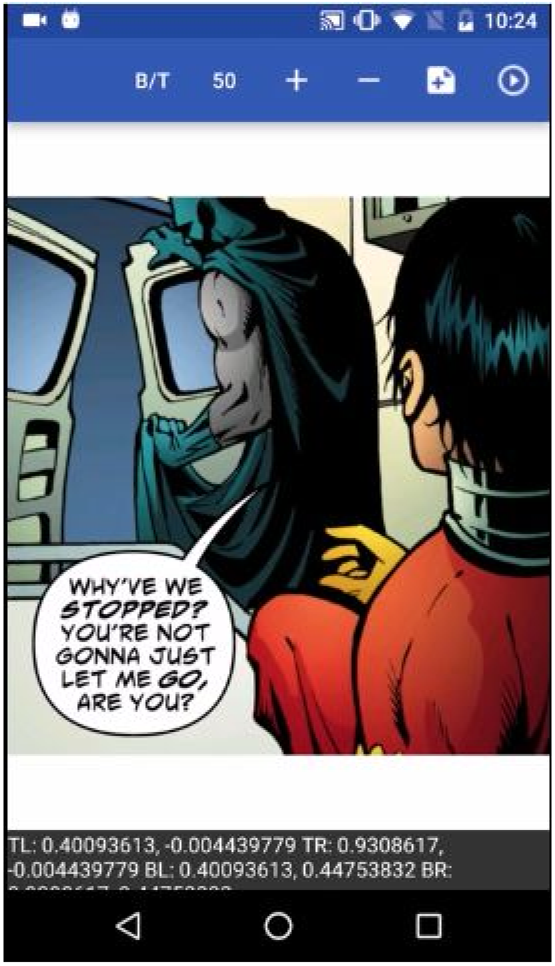
This screen shows that panel-to-panel reading also takes up the entire screen and shows the panes (letterboxes) from that saved panel.

Screen 6: Show Status Bar, Toolbar, and Navigation Bar



When in reading a comic in free read or panel-to-panel mode, the user can tap around center of screen to bring up the status bar, toolbar, and navigation bar.

Screen 7: Panel-to-Panel Creator Mode


10:24
B/T
50
WHY'VE WE STOPPED? YOU'RE NOT GONNA JUST LET ME GO, ARE YOU?

```
TL: 0.40093613, -0.004439779 TR: 0.9308617,
-0.004439779 BL: 0.40093613, 0.44753832 BR:
```

This screen shows a POC I worked on which shows the creator mode for creating panels with the top/bottom panes (letterboxes). You can see this in action, and also the panel-to-panel animations, from a screen video capture I uploaded: https://www.youtube.com/watch?v=Vtithth-r2w

Key Considerations

How will your app handle data persistence?

I'll be building a content provider, using SQLite, and will save preferences with SharedPreferences.

Describe any corner cases in the UX.

- User hits back button while in a transition
    - In onPause, interrupt animation, save state and user's position in the comic in DB in case Activity is killed by user or system
- While creating a panel-to-panel creation, user hits home or back button
    - Save state and user's position
    - Creations will have data saved automatically to DB as user works on it

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

- subsampling-scale-image-view https://github.com/davemorrissey/subsampling-scale-image-view
    - View for images that supports pan, zoom, animations, tiling, memory handling in regards of large image files, etc
    - Picking this for the view when viewing a comic in read and creation modes
- Picasso
    - Will use this for simple and efficient image loading on gridview, banners, avatar, etc
- Junrar
    - Will need to use this Java library in order to be able to unpack rar files, which is the file type most DRM free comics are released in
- UI & Material libraries
    - MaterialDrawer by mikepenz
- Google Analytics
    - I would like to track reports on exceptions, crashes, and other user activities
- Google AdMob
    - I'm going to add advertisements to satisfy capstone requirements and get more experience with adding ads to app, and who knows, maybe I'll release a version to play store with ads and finally make my millions!

Next Steps: Required Tasks

Task 1: Project Setup

- Create project structures
    - Packages, resource directories, etc
- Create shared resource files and begin populating with shared values
    - string, color, style, etc
- Add libraries and dependencies

Task 2: Implement database and content provider

- Design the database tables
- Create contract, helper, and provider files

Task 3: Implement UI for Each Activity and Fragment

- Build UI for activities:
    - Library activity which has tab views
    - Detail activity for when user selects comic
    - Full view comic reading activity
    - Full view comic panel-to-panel creation activity
    - Settings activity
    - About activity

- Design/implement other UI items:
    - Navigation drawer
    - File select dialog
    - Paging on the read and create activities

Task 4: Implement decode and load functionality for comic files

This app initially will support 3 ways to load a comic, so we will need a service for decoding and/or loading images of the comic. Once decoded, images will be loaded into memory (Comic POJO)

- Decode .cbr files (RAR) and load images
- Decode .cbz files (ZIP) and load images
- Load directory of images into memory

Task 5: Implement create and save functionality for panel-to-panel experience

This feature is the most time and code intensive, so it gets its own task.

- This will be on the Full view comic panel-to-panel creation activity previously mentioned
- Reuse and refactor code from POC
- Create classes for capturing saved panel info and saving to a file

Task 6: Create open and load panel-to-panel saved file

- Create a class for opening and loading a saved panel-to-panel file and loading into the DB as saved panels for a comic

Task 7: Implement Google Analytics and AdMob

- Analytics
    - Track exceptions and crashes
- AdMob
    - Add ad banners to Library and Details activities
    - Add paid and free flavors

Task 8: Create a widget

- Widget can display list of comics in library
- Clicking on comic will call the resume function on that comic
    - If the comic has not been read yet, then clicking on comic will take user to the comic’s detail activity