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: Gh0stWires

Pulp

Description

This comic book reader lets users organize their comic books in a more tangle way. Ever search through the stacks at a comic book store? This app aims to recreate that digitally.

Intended User

Comic Book Lovers

Features

List the main features of your app. For example:

- Finds all .cbr and .cbz files stored on the device
- Displays them as movable cards on screen
- User can read through the comics

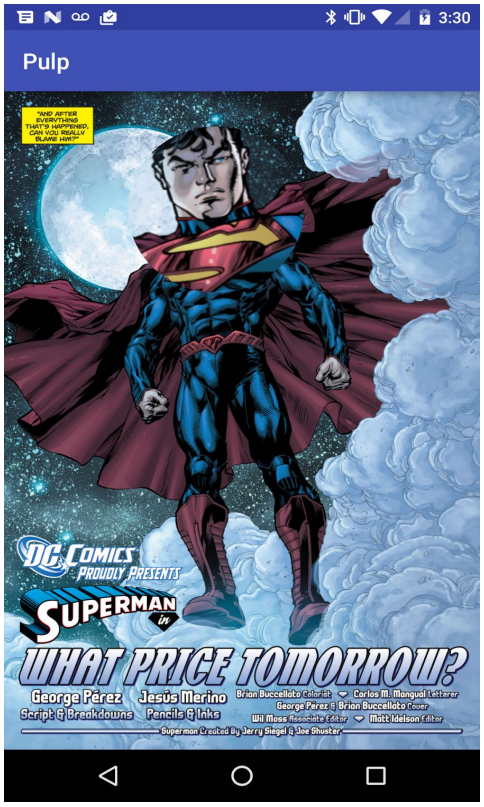
User Interface Mocks

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

This is a stack screen. The comics are stacked on top of each other and you swipe right to open the comic or left to go to the next one.

Screen 2



This is the reader activity. The images will be held in a ViewPager to scroll through the comic. It will use a modified version of ImageView that gives the user a magnifying glass for reading text in the comic.

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 use a content provider to store comic files names and their associate recent page number

Describe any corner cases in the UX.

If the user presses the back button it will always return the the previous activity and not the previous page in the viewer. This action is expected of the back button and I don't want to change it.

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

Swipe-Deck

This library adds a swipeable feature like you see in apps like tinder. It makes the objects feel more tangible.

junrar

This library will allow me to manipulate cbr files. This is a format a lot of digital comics use

Describe how you will implement Google Play Services.

I will use google play services to display comics that users might have on there google drive.

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

This step will make sure we have the libraries we need and the permissions we need for the app

- Add libraries to gradle build
- Add permissions for accessing the phone's storage to the manifest file

Task 2: Implement UI for Each Activity and Fragment

This sets up the basic foundation of what each activity does

- Build UI for Stack Activity
- Build UI for Reader Activity

Task 3: Decode Bitmaps from Archive

This will allow us to get the images needed for the UI in both activities

- Create class that manipulates archive file to retrieve and decode bitmaps
- Extract files for decoding for android version lower than 7

Task 4: Find Archives

This will Find all comic files on the users device

- Create function that will search the file system for any files ending in .cbr or.cbz
- Check if user has any comic files on there google drive

Task 5: Content Provider

This will create a db of all the archive files and there names so we can store a reference to the pages the user left off on

- Create a Database
- Create Provider for Database

Task 6: AsyncTask

I will use async tasks to get and decode bitmaps that need to be loaded into the UI thread

- Create AsyncTask for Stack Activity
- Create AsyncTask for ViewPager Activity

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**"