GitHub Username: chandan276

Comics World

This app will be written solely in the Java Programming Language.

Description

This application help users to find comics of their choice or select from the listed comic collection. Comics World has a built-in search feature which allows users to load and browse some additional comic info like volume descriptions, character profiles and so on. It shows Volumes, Issues, Characters, Movies, vies and user's favorite list. All comic related data provided by Comics Vine API.

Intended User

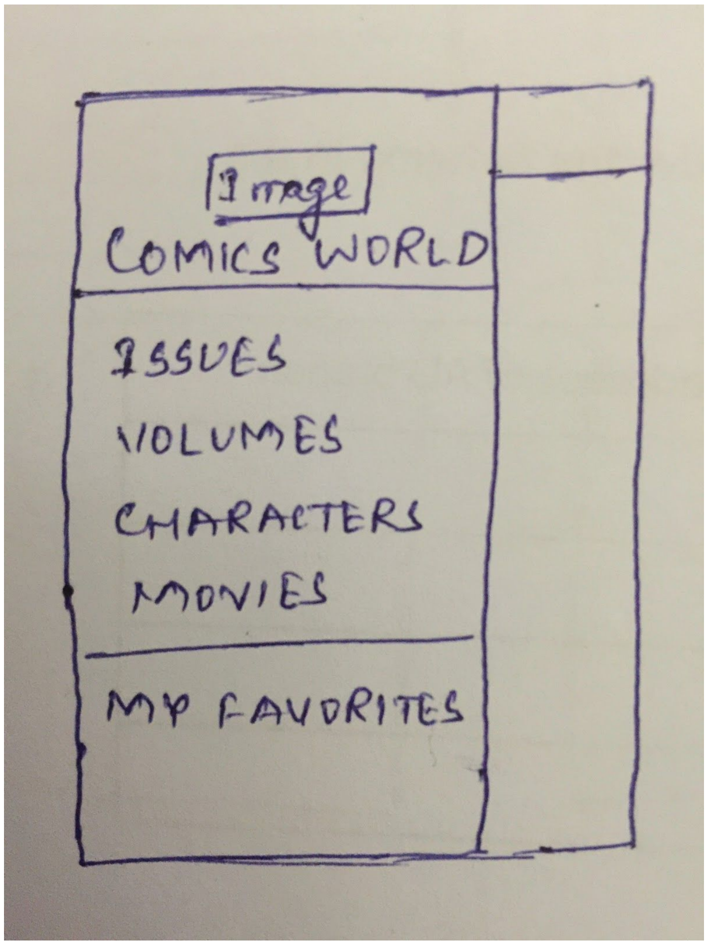
This application is aimed to comic fans who want to organize their own comic collection, track release dates for chosen volumes and get online access to Comic Vine search engine.

Features

- Browse Volumes, Issues, Characters in comics.
- Browse Movies.
- Add to Favorite list.
- Browse Favorite list.
- Search in Volumes, Issues, Characters.

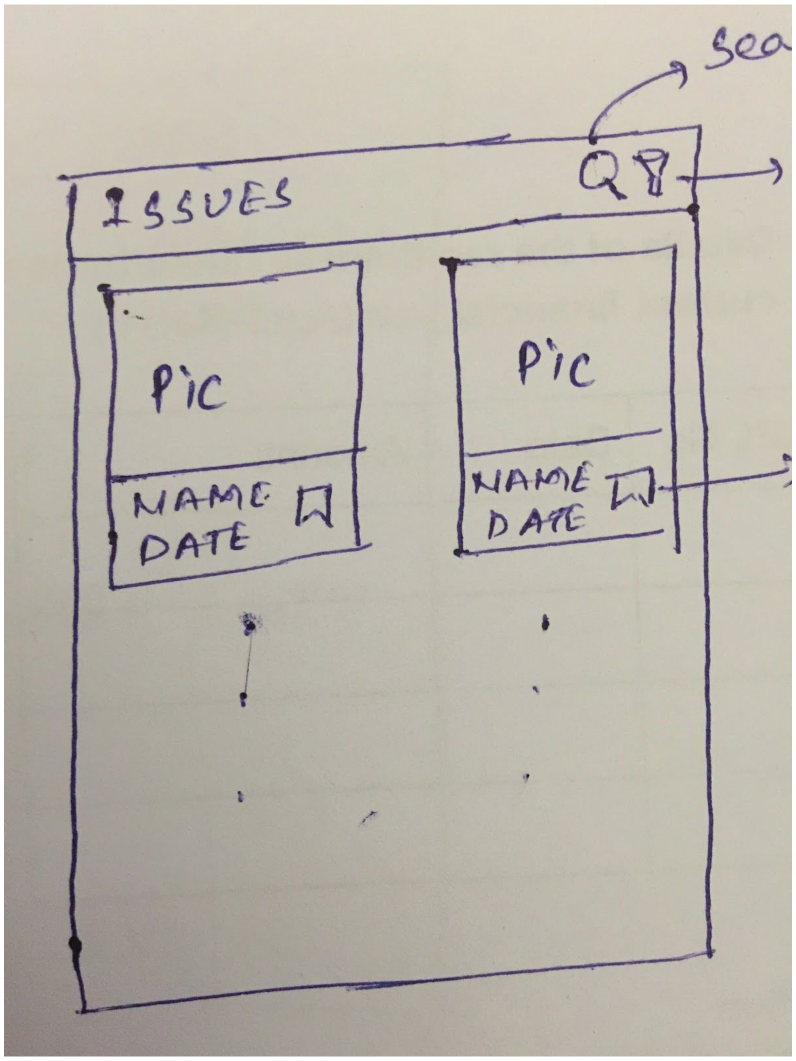
User Interface Mocks

Navigation Bar Drawer



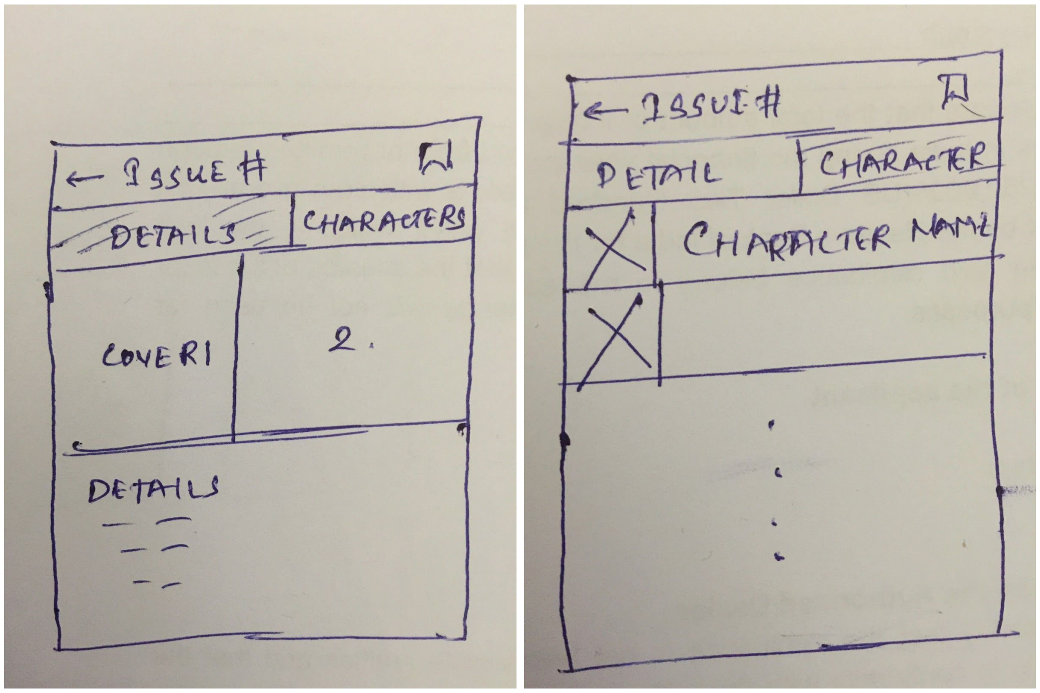
Displays main navigation bar drawer with all app features.

Issues



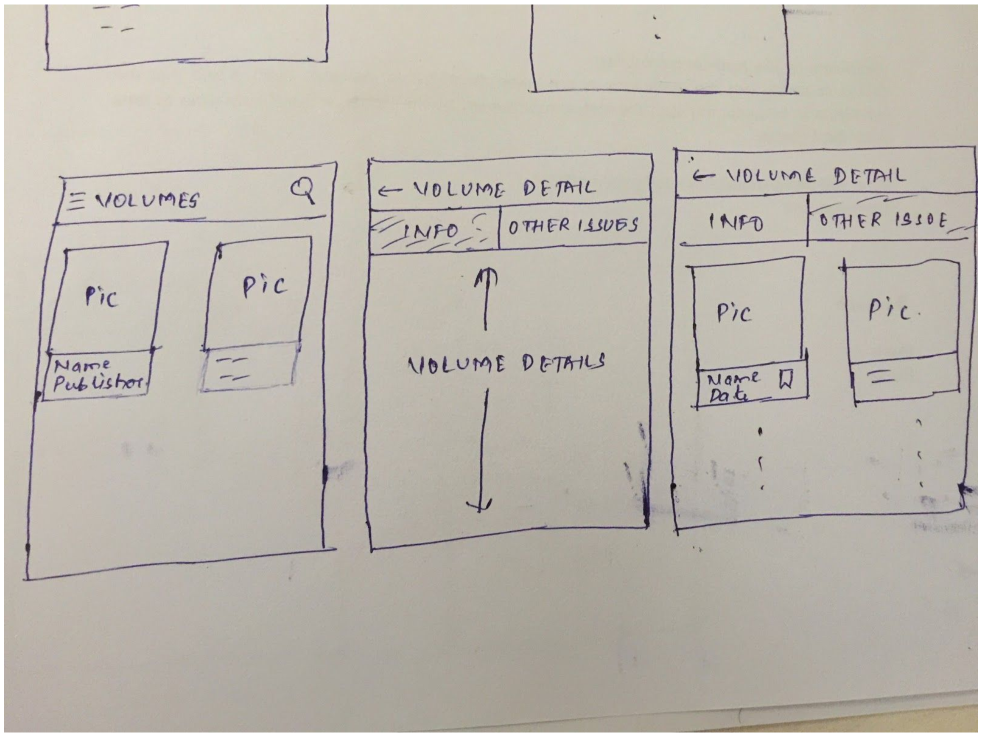
Main application page which shows today's issue releases. Contains filtering and search options, also each issue card has a bookmark option which allows user to mark selected issue as owned.

Issue Detail



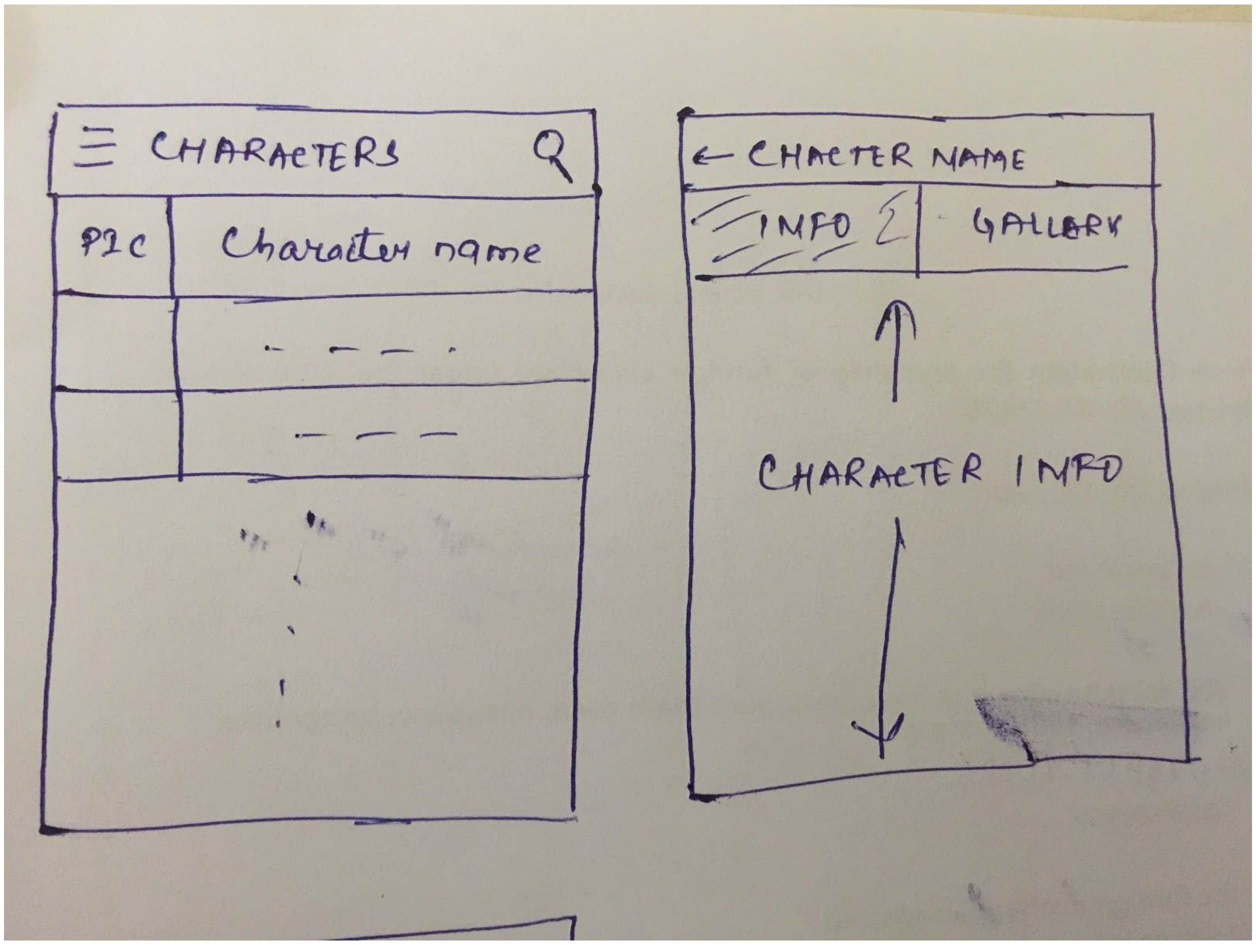
Issue details activity, allows to check the main issue info as well as browse issue characters list. Click on any character item launches a new activity which displays chosen character info.

Volumes

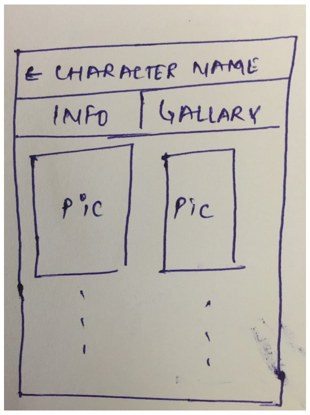


Volumes activity with a search option.
Volume details activity displays primary volume info and all issues related to this volume. Each issue card has bookmark option which allows user to mark selected issue as owned.

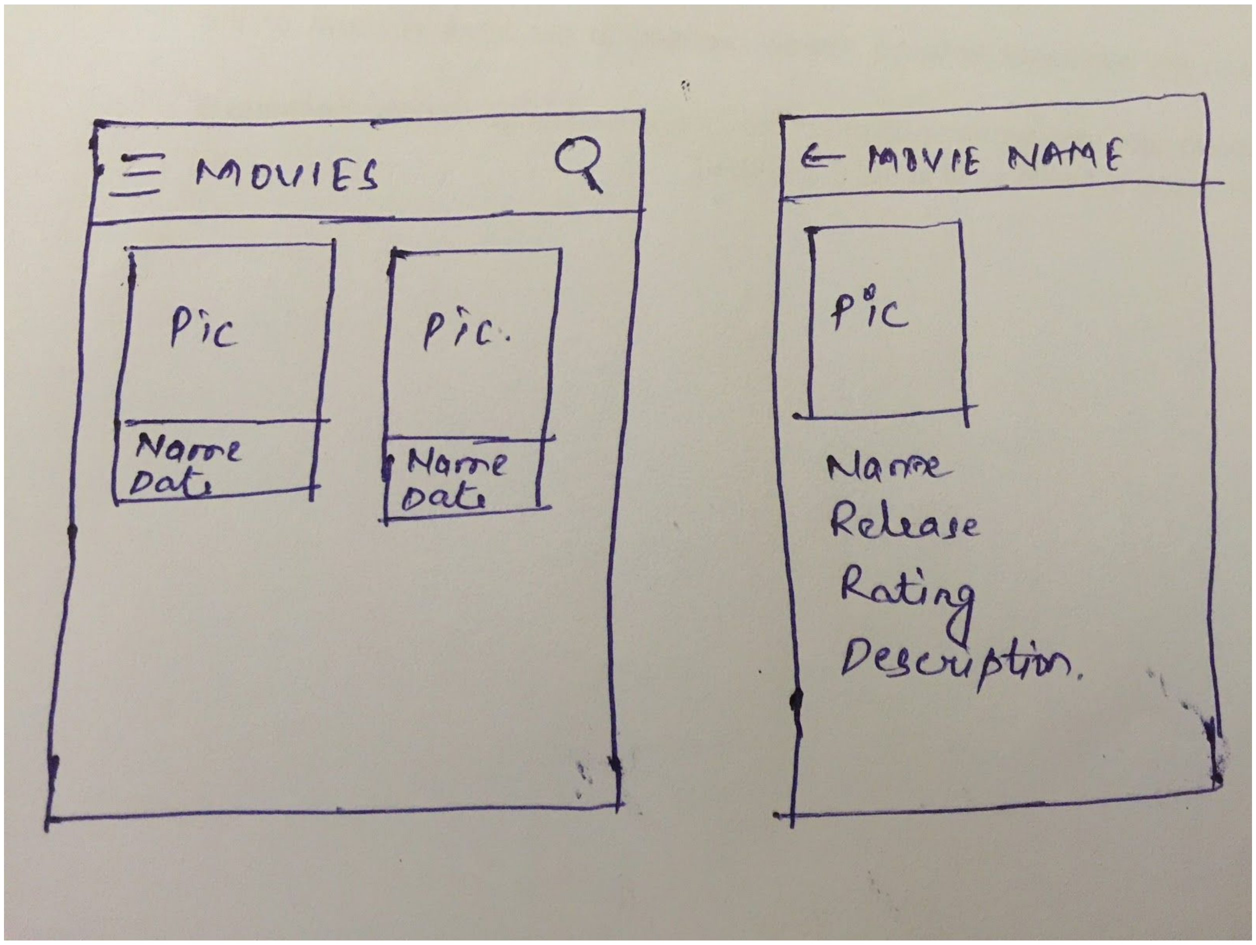
Characters



Characters activity with a search option.
Character details activity displays primary character info and character gallery as shown below.

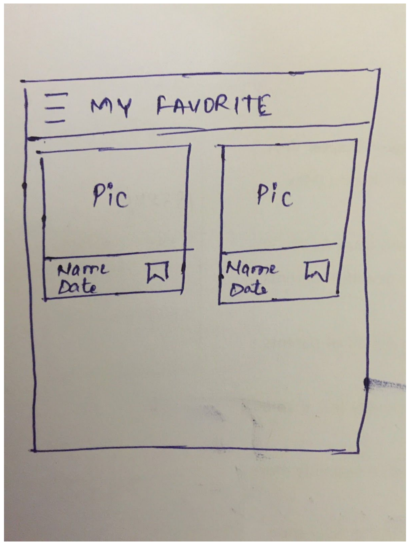


Movies



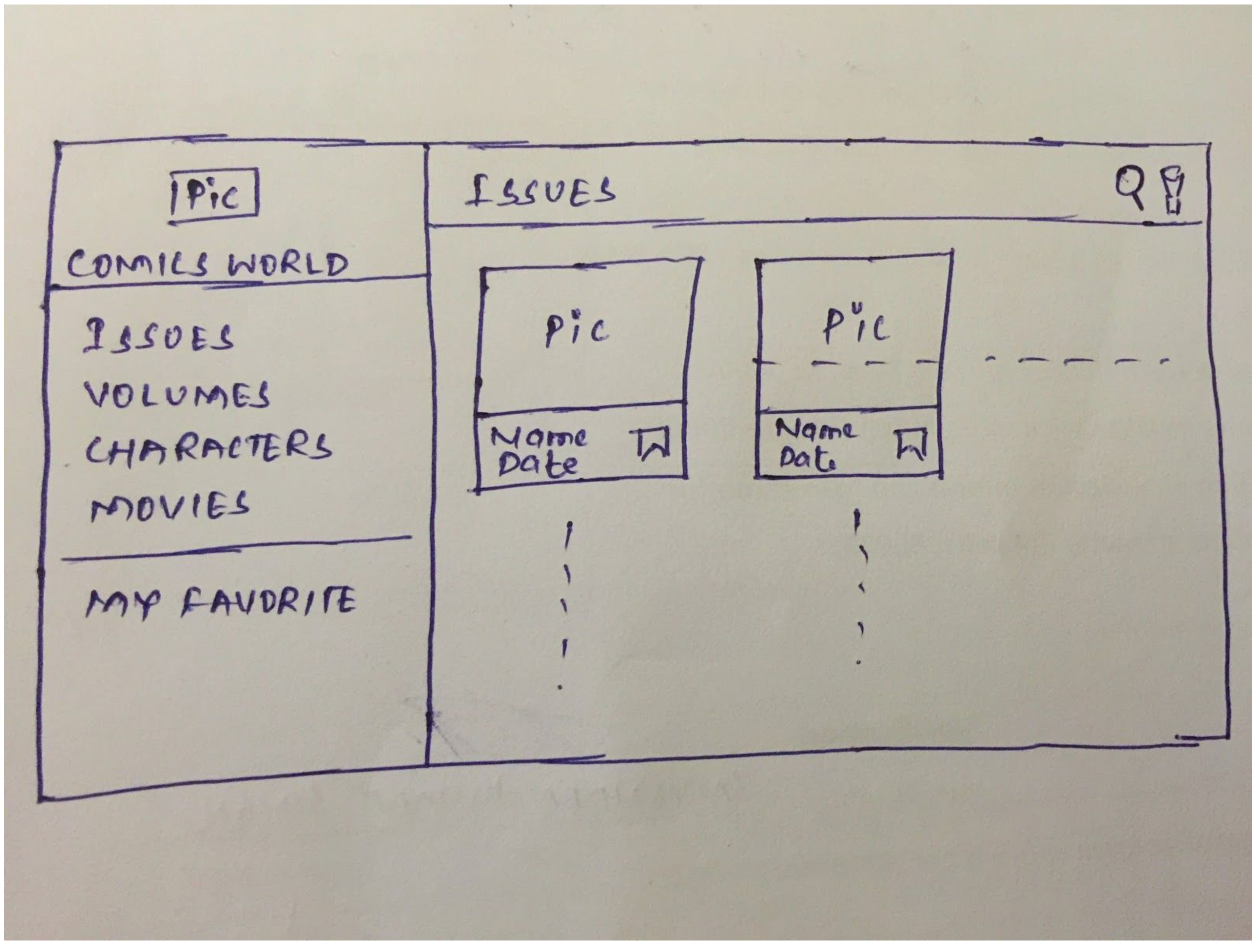
Movies activity shows movies list. Clicking on any movie shows movie details.

My Favorite



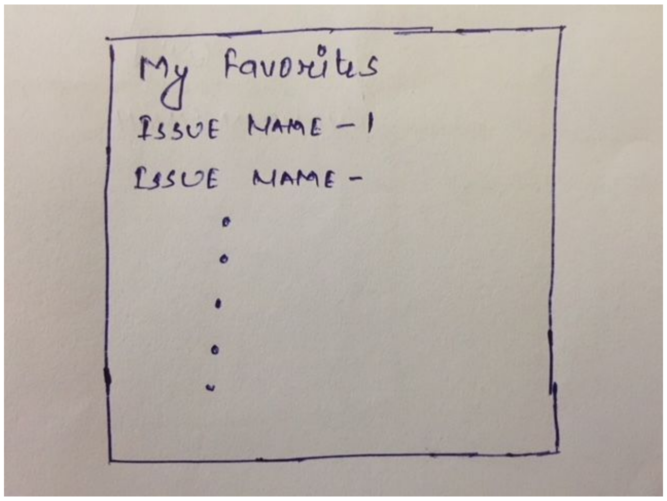
Favorite list activity allows user to manage owned comic collection. Click on any item opens chosen issue details activity.

Tablet Layout



Tablet layout variations will have constantly visible left navigation bar.

Widget



App Widget displays user’s favorite list. Click on any item list opens issue details activity.

Key Considerations

How will your app handle data persistence?
App will use content provider to maintain local data.

Describe any edge or corner cases in the UX.

Unstable or missed network connection: the application must not crash in that cases

Device orientation change: the application must handle all long- running operations correctly considering possible configuration changes

UI freezes: the application must not use the main thread for any resource consuming operations

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

IDE/Libraries	Purpose	Version
Android Studio	IDE used for development	3.1.4
Gradle	gradle	3.3.0
Glide	for image downloading and caching	4.6.1
picassopalette	for palatte	2.0.0
retrofit	for network calls	2.4.0
converter-gson	for parsing response data	2.4.0
firebase-core	for analytics	16.0.6
crashlytics	for crashlytics	2.9.8

App performs short duration, on-demand requests(such as search) and fetching comics data, for which app uses an AsyncTask.

Describe how you will implement Google Play Services or other external services.

The application will use Firebase Crash Reporting and Google Analytics for Firebase which both depend on Google Play Services.

Next Steps: Required Tasks

Task 1: Project Setup
Create and setup a new project. This task includes:

creating a new project in Android Studio
configuring libraries by adding all necessary dependencies

Task 2: Data model classes
Create data classes which help to handle all response data provided by Comic Vine API calls.
Required Classes:
- Response
- Issues
- Characters
- Volumes
- Movies

Task 3: API
Implement a service which provides all necessary network API requests. Service based on Retrofit.

Task 4: Data persistence
Add a content provider helper class to handle all locally stored data.
Required steps
- create a database helper
- create a content provider
- implement database access functions using Loaders and created content provider, Loader finishing callback should update the related views

Task 5: Create the UI
Implement all necessary fragments and activities.
Following UI parts will be created:
- left side navigation bar drawer
- issues list [activity]
    - Issuedetails[fragment]
    - Issuecharacterslist[fragment]
- volumes [activity]
    - volume details [fragment]
    - other volume related issues [fragment]
- characters [activity]
    - character info [fragment]
    - character story arcs [fragment]
    - character gallery [fragment]
- Movies [activity]
    - Movie List [fragment]
    - Movie details
- My Favorite [activity]

Task 6: Responsive design
Adapt the application layouts for tablets.

Task 7: Google Play Services
Implement chosen services (Firebase Crash Reports and Firebase Analytics).

Task 8: Make the App production ready
- Improve the app visuals using some Material Design techniques (shared element transitions, parallax scrolling, animations etc.)
- Provide RTL-layout support
- Provide accessibility support

- Provide content descriptions for meaningful UI elements
- Plan for localization (move all strings into strings.xml)

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”