GitHub Username: anandwana001

ComiCoid

Description

ComiCoid android app is for those people who love marvel comics, characters and many more. This app provides the data from the Marvel API like characters, events and comics cover.

Intended User

People who love the marvel comics and children.

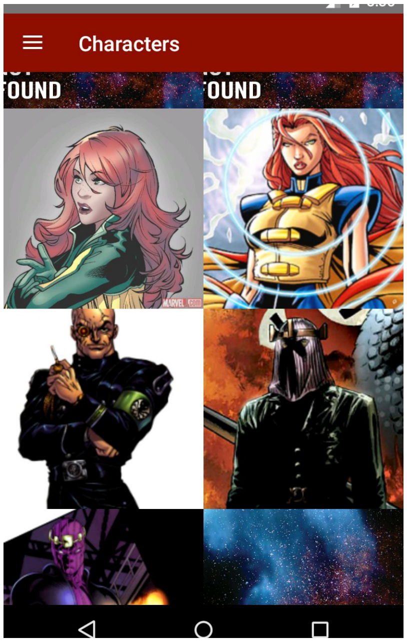
Features

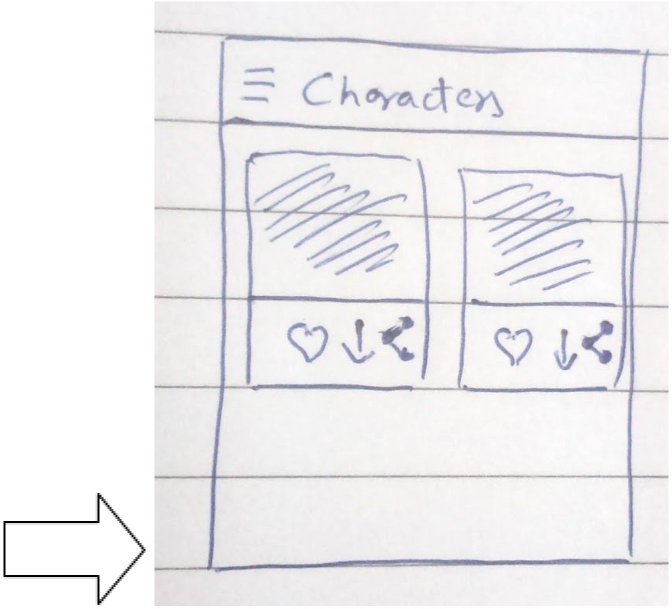
List the main features

- Favourite your characters and save them
- Favourite your comics cover and save them
- Share character and comic cover with others
- Download images of character and comic cover in your gallery

User Interface Mocks

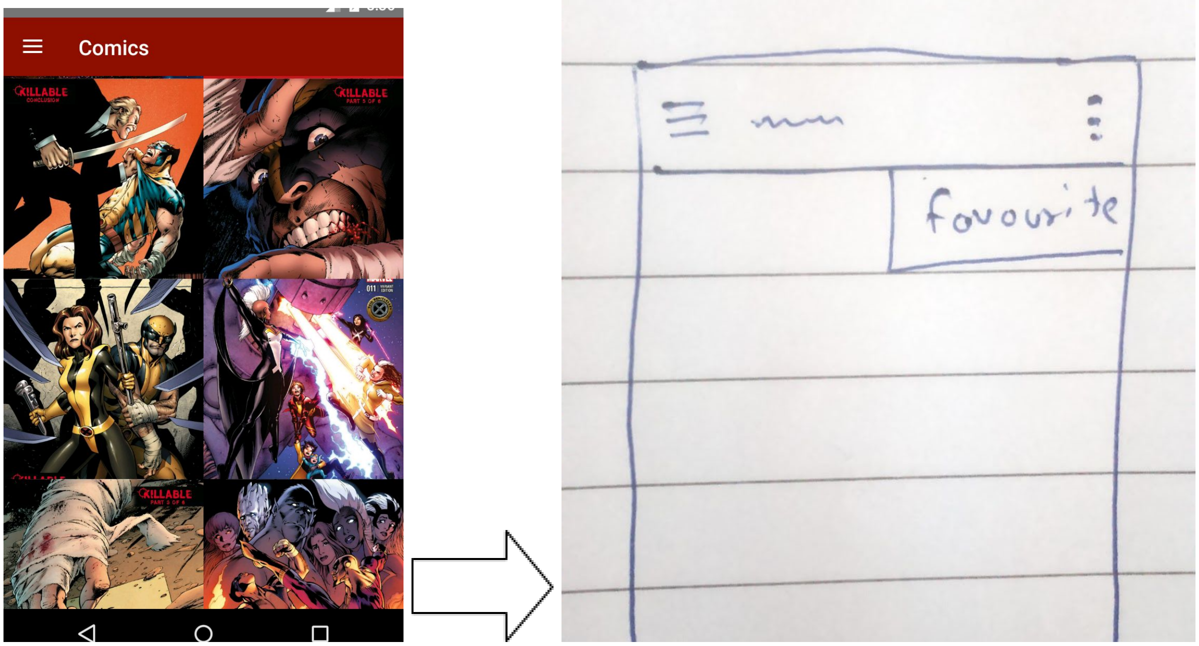
Screen 1





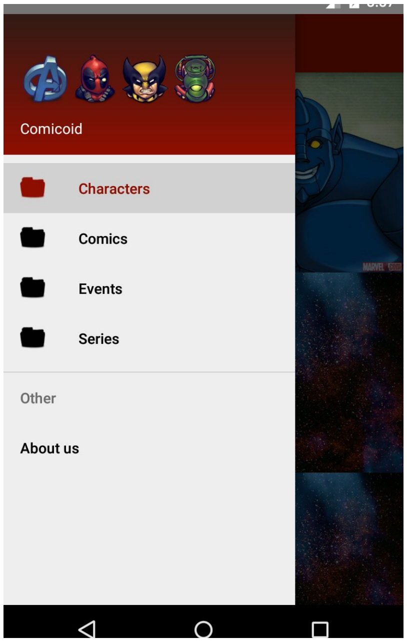
This is the Launcher screen where user get the list of characters and directly share, download and save them from here only

Screen 2



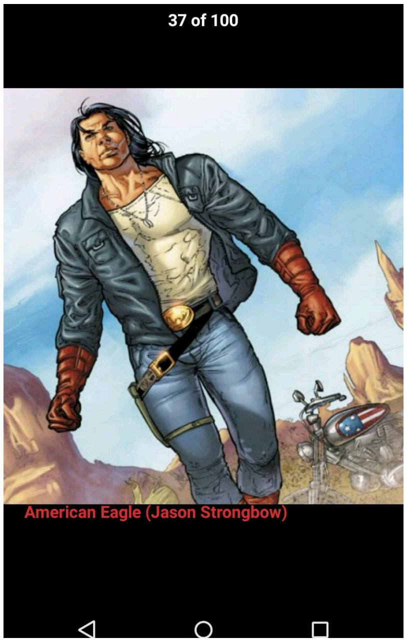
Same as the Launcher screen user can do all functionality here and here the get the list of comics cover

Screen 3



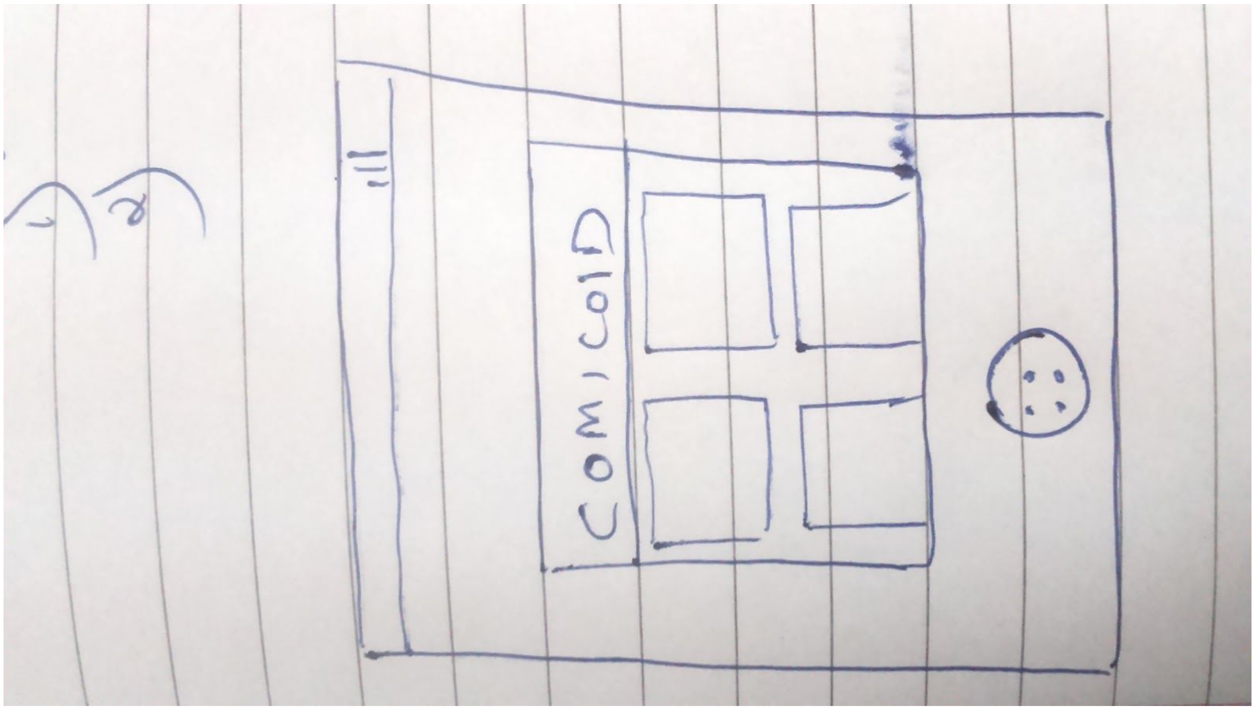
Navigation bar help user to switch from one content to another and in about us they get the detail of marvel

Screen 4



User can click on item and they get the feature of simply switching from item to another (ViewPager) horizontal scrolling of images

Screen 5



Widget on home screen will show the favourite data

Logo

User can click on item and they get the feature of simply switching from item to another (ViewPager) horizontal scrolling of images

Key Considerations

How will your app handle data persistence?

As user favourite their characters and comic cover they will store in content provider and backed by a SQLite database. Content Provider update whenever user favourite or unfavourite the item. For network, app use AsyncTask to get the data from Marvel API. Loader is used to load the data from API and also from the database.

Describe any corner cases in the UX.

Sharing the character and comic cover, app will handle the situation if no other application is there to perform the action by giving proper message to the user so user will not get confuse. Downloading the image if no space in phone will give user the information of insufficient memory.

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

Glide to handle the loading and caching of images. Butterknife for binding the UI. Marvel API is used for the data.

Describe how you will implement Google Play Services.

Google AdMob for showing the ads to the user on the Launcher screen.
Firebase Analytics for analyzing the application behaviour.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure Glide - Image Loading Library
- Getting Marvel API Key and the url to get the data from
- Choosing the endpoints for the application

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
  - Set RecyclerView for the image list
  - Build the layout for content of the list
- Build NavigationBarLayout and layout for different fragments
- Build Overflow menu and favourite option inside it

Task 3: Handling Data

- Call Marvel API for character and comic cover to show the relevant data in relevant fragment
- App use AsyncTask to get the data
- Build content provider and sqlite database and attach relevant layout and show data
- Loader is used to load the data for Launcher screen and other fragments and also to load the favourite data
- Widget will show the list of data which user make favourite

Task 4: Share and Download

- Perform share and download image to gallery

Task 5: AdMob and Analytics

- Show ads before the app start
- Firebase analytics for characters
  - Make firebase project
  - Set firebase dependencies
  - Create instance for the characters and integrate it