GitHub Username: aki25

Photo Editor

Description

Photo Editor is an easy to use image editor that lets you quickly transform the images that match your desire, even if you've never edited a photo before!
It comes with various filters and effects that can be applied to your images such as blur, mirror, grayscale, rotation etc.
It even gets you random images from the internet that you can download and set as wallpaper.
When you are done editing your image, you get the ability to share your masterpiece with the world!

Intended User

This app is for anyone who wishes to edit digital images.

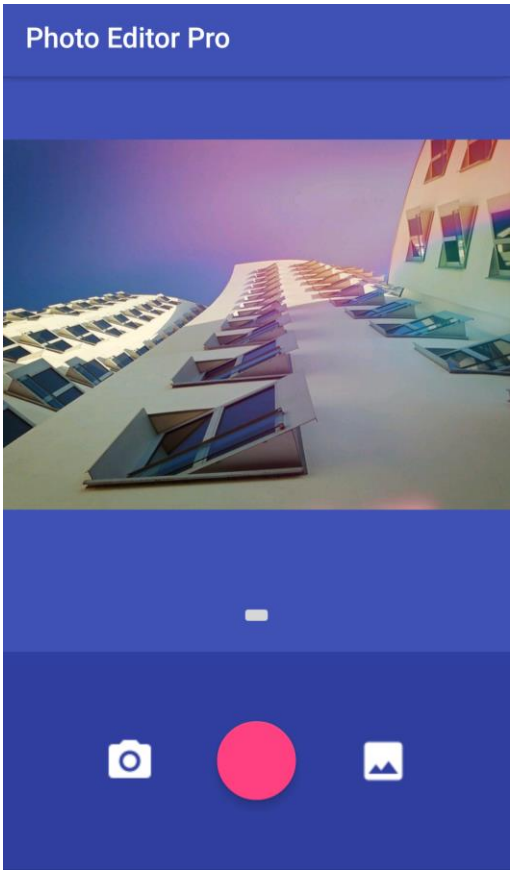
Features

- Takes pictures

- Fetches a random image from the internet on the home screen of the app that can be saved to disk and also set as wallpaper
- Has widget that displays that the downloaded wallpaper
- Allows overlapping of editing effects
- Contains various effects such as
    - image rotations
    - image blurring
    - grayscale filter
    - adjusting brightness
    - mirror flipping
    - various other filters

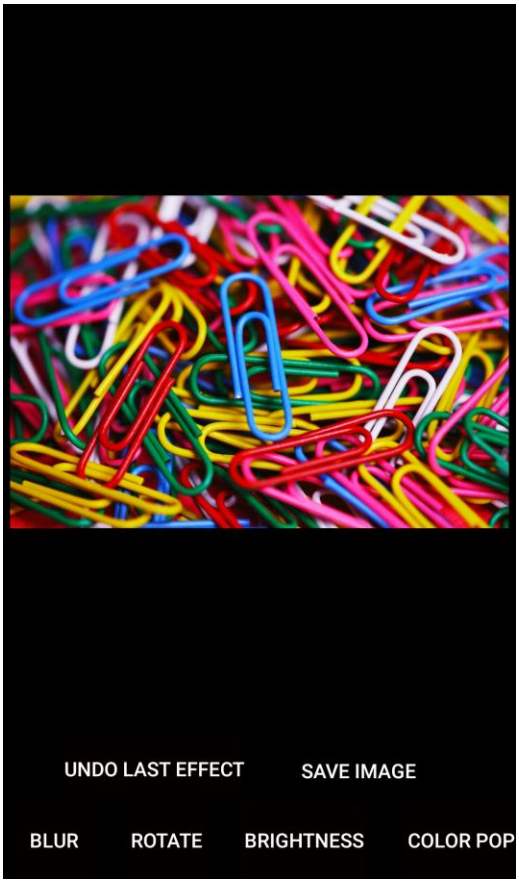
User Interface Mocks

Screen 1



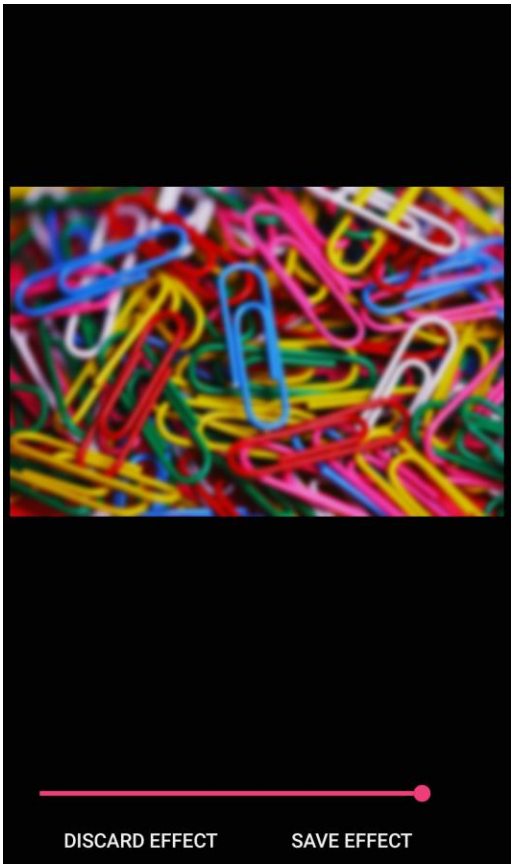
The initial home screen, with the name of the app, a randomly downloaded image from the internet that can be set as background and/or saved locally on the device.
Two buttons that lets user pick image from either the camera or the device.
Once the image is loaded it is taken to the next screen/activity for processing.

Screen 2



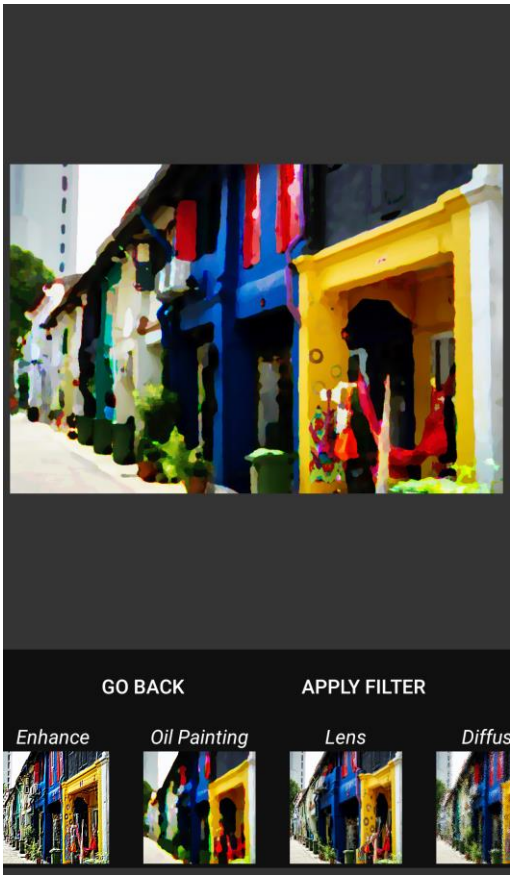
This screen is displayed once an image has been selected for editing. A huge portion of the screen is taken up by the image that is to be edited and the remained shows various editing options

Screen 3



If a selected option has adjustable strength options, such as the blur option, the UI changes to present user with more options such as in the above example.

Screen 4



Certain major filters and image effects are displayed with a preview of the effect.

Key Considerations

How will your app handle data persistence?

A content provider will be implemented that will allow users to view all the saved wallpapers and optionally delete them

Describe any corner cases in the UX.

The navigation in the app is fairly linear and the app will only be functional when it is in foreground. At the time any corner case is unknown, but if any were to surface during implementation, it shall be handled gracefully.

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

Picasso or Glide to handle the loading of images,
Android-jhlabs (android – arsenal) for some applying certain effects in the image,
View animator (android – arsenal) for smooth image loading animations to enhance UX,
Other libraries will be added when necessary.

Describe how you will implement Google Play Services.

1) Firebase crash reporting: This service will be used to get elaborate report if the app crashes. This will help in providing better and more stable updates for the app.
2) Google adMob: This service will be used to display banner/interstitial ads.

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

1) Get an API key from pixabay.com that will allow getting wallpapers
2) Add all the dependencies in the gradle file

Task 2: Implement UI for Each Activity and Fragment

1) Build UI for MainActivity
2) Form a request to pixabay to get images
3) Parse the incoming data and load the image on the main screen
4) Build UI for Image Processing activity
5) Keep the theme constant throughout the app
6) Add fragment that changes layout in image processing activity (for image editing buttons that provide options)

Task 3: Implement settings activity

1) Add a checkbox that gives the ability to enable/disable loading of wallpapers on main screen
2) Add option that lets user manage saved wallpapers

Task 4: Image processing

1) Thumbnails are shown as image previews for buttons
2) Image processing is done on background thread
3) Add various image editing effects including:
    a. Mirror flips (horizontal and vertical)
    b. Blurring
    c. Grayscale
    d. Brightness adjustment
    e. Other cool effects such as twirl, oil painting, etc.

Task 5: Implement Google Play Services

1) Connect the app to firebase for crash reporting
2) Implement banner ads in the app