
Pixtop.

Description

This app will be a great place to allow users to enjoy photographer's work. It doesn't matter if you like cats, birds, or buildings, this app will let you access the latest pictures taken from your favorite photographer. For photographers they will get a place to expose their work even in offline mode.

Intended User

This app is mainly for people who is interested in photography. They can show case their gallery of pcitures in this app. And can search for any picture and get the details of that picture(like photographer's name and blog).

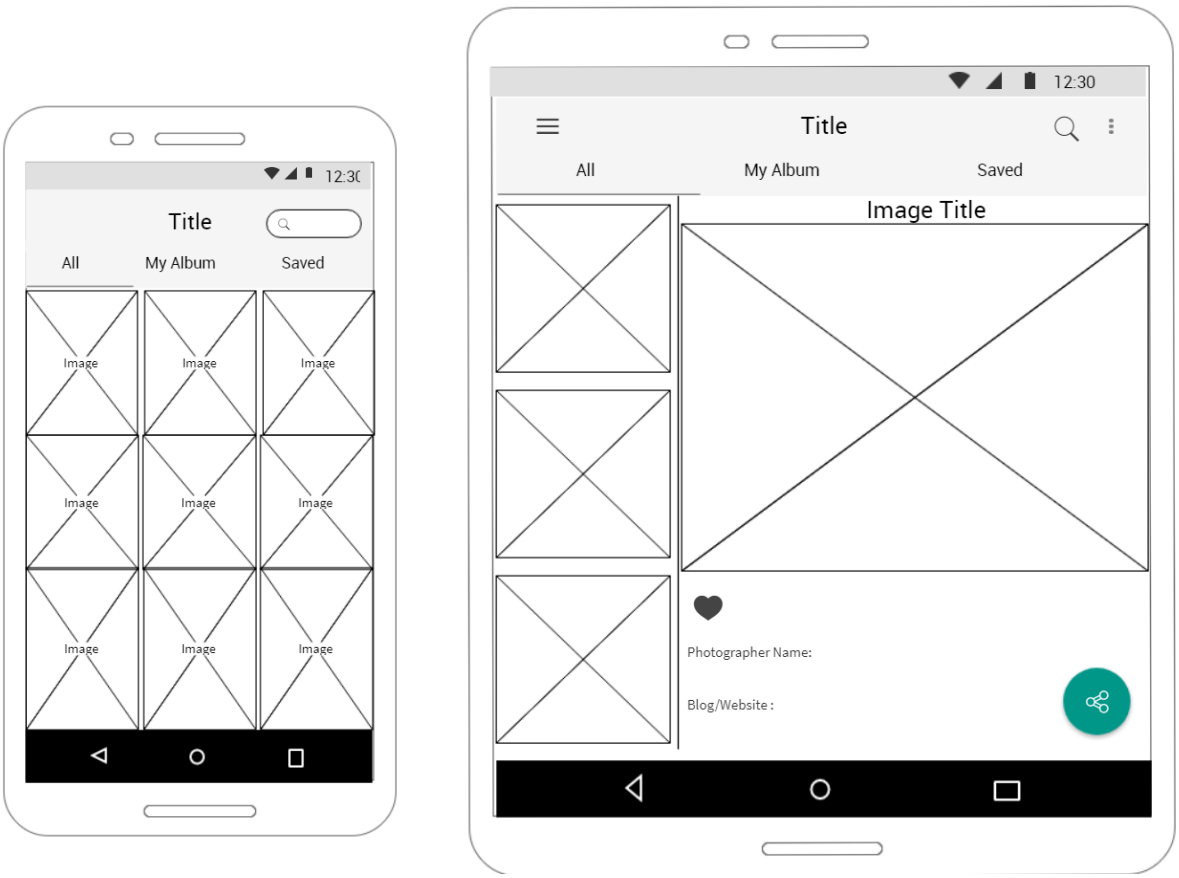
Features

Main Features of this app:

- Show case your gallery of pictures.
- Search for pictures.
- Save the picture.
- Share the picture.

User Interface Mocks

Screen 1

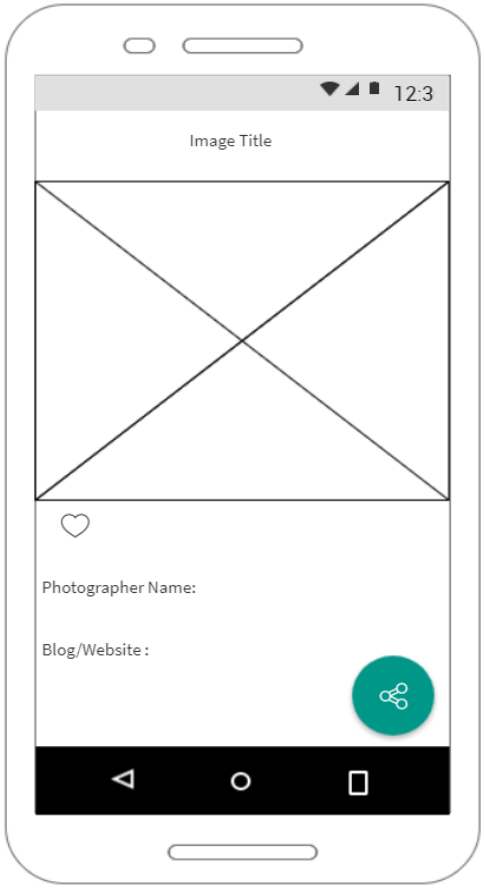


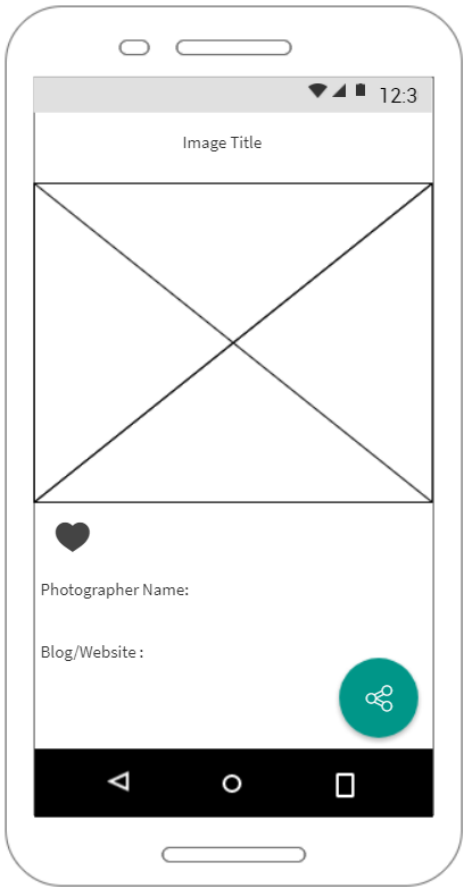
This is the landing screen the users lands on to, where it shows list of images in a grid view. It has three tabs in it All, My Album, Saved.

All tab shows images of birds by default. This screen also have search bar to search the images they want and also the images based on location.

My Album tab shows the images taken by the photographer(All these images are already saved and have to save using the services that is implemented)
Any image which is liked or favourited are shown in the saved tab.

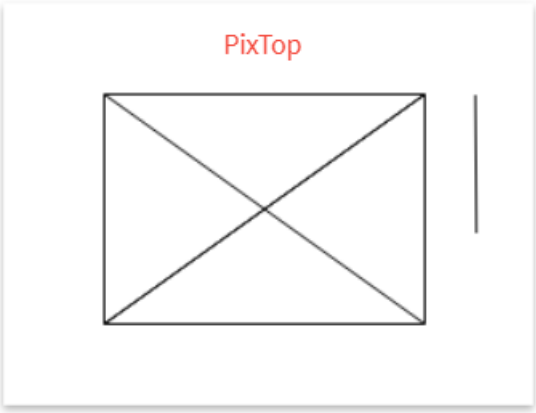
Screen 2





This is the detailed screen where it shows the details of the picture clicked on the landing screen. There is like button below the image to save that image to DB. User can share the image using the FAB. The image title will be shown on the top of the image and remaining details are shown in bottom if they are available.

Widget



This is the sample widget where it shows the list of saved images. User can scroll through and look at the images.

Key Considerations

How will your app handle data persistence?

App uses content providers to store the data in db.

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

Glide is used for image processing.

Describe any corner cases in UX.

At any point of time if application fails to retrive the data from server , application gracefully displays a error message.

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

Firebase cloud messaging to notify the users when any photographer uploads a new picture.
Firebase Analytics to check the analytics of the app.
Firebase carsh reporting to check the crash reports of the app.

Next Steps: Required Tasks

Task 1: Backend Service Setup

Create a service which accepts a picture and all information of the picture including the photographers name, Image title and where it is taken.

Task 2: Client project setup

- Configure dependencies
- Configure Release Signing
- Configure external libraries
- Add Crash reporting support
- Add application theme, and base styles for toolbar, buttons, text, titles, colors

Task 3: Implement UI for Each Activity and Fragment

- Create a viewpager in Main activity to show three tabs.
- Implement the grid view to show the gallery of images.
- Make the UI compatible for larger devices also.
- Implement the FAB in details activity, which allows the user to share the image.
- Implement the like button in the details activity when clicked on it allows the user to save the imge to DB.

Task 4: Integrate the app with services

- Upload the images to the backend service to make it available for the app to consume.
- Use the backend service to get the images from the service.
- Integrate the pixabay api to get the other images.
- Get the data from pixabay api using asyncTask to display the images.

Task 5: Content providers and Loaders

- Create content provider to store data into db.
- Use loaders to display the data in the screen.

Task 6: Save Favourites

- When user clicks on the like button in details screen, save the purticular image details into db.
- When user selects on saved tab on main activity, all the saved images in the content provider will be shown

Task 7: Integrate Google services

- Integrate Firebase cloud messaging system to receive the notification from the photographer when the new picture is uploaded.
- Integrate Firebase analytics and Firebase crash reports to the app.
