

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[Screen 7](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Implement task associated with the backend](#)

[Task 4: Handle Exceptions and improve UI](#)

[Task 5: Make Application compatible](#)

[Task 6: Get feedback from code reviewer and friends](#)

GitHub Username: [prithvibhola](#)

Allure

Description

Write a brief summary of what your app does. What problem does your app solve?

Allure provides beautiful and top notch work of the photographers from all over the world. It provides over 10000+ high quality photographs which can be downloaded, shared with friends and social websites and even set them as a device wallpaper.

You can like/love the images which can be stored on the cloud and will appear across all your device in which Allure is installed and logged in.

Search with the tags like Adventure, Nature, People etc. and get the results from over 10000+ photographs.

Optimized for tablets and many android devices Allure uses very clean and minimalistic design with Google material design specifications in mind. Included pretty animations to enhance user UI/UX.

Intended User

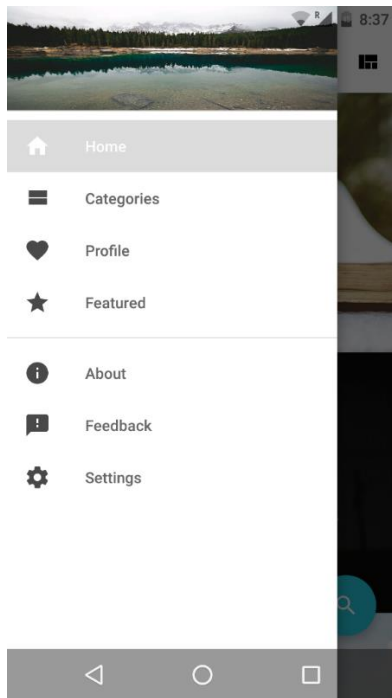
Allure is for everyone who likes/love photography or simply love photographs and/or want to set beautiful photographs as their device wallpapers.

Features

- Display beautiful photographs submitted by photographers on Unsplash.
- Saves information liked by user online to make it available over all devices.
- Allows to download or set any photograph as device wallpaper.

User Interface Mocks

Screen 1



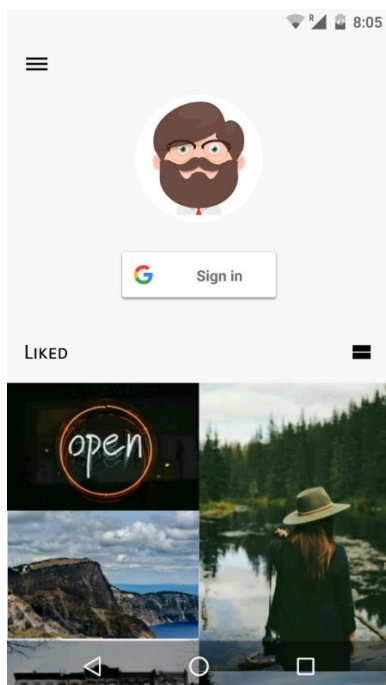
Navigation Drawer with navigation items for exploring the different features of the app.

Screen 2



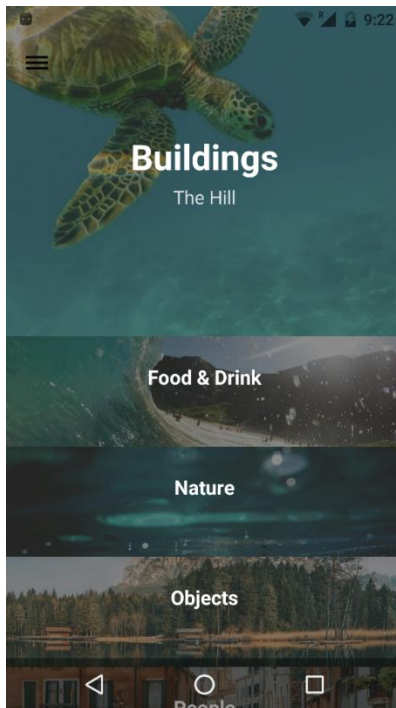
Main screen to display Unsplash photographs with shuffle button to show random photographs (classified as latest, oldest and popular). Layout button to change layout between one column and two columns.

Screen 3



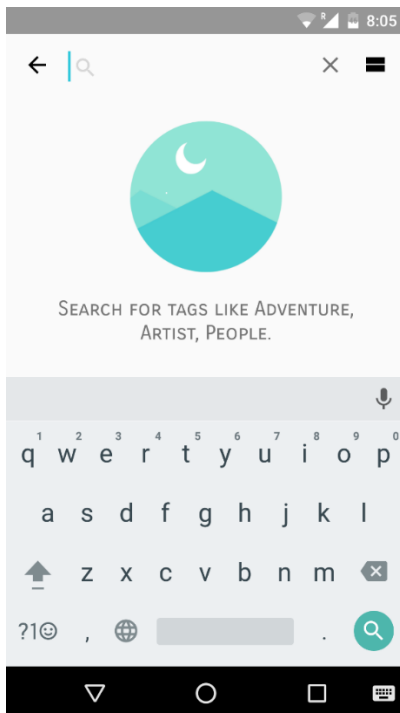
User Profile screen which includes default user image and login button if user is not logged in. If user is logged in user image along with user name and email id will be shown.

Screen 4



Category section which gives option to select images from different categories like Buildings, Nature, People etc.

Screen 5



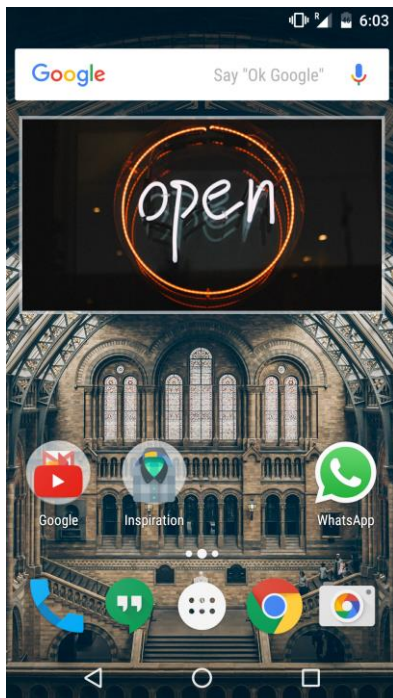
Search functionality to search over 10000+ Unsplash photographs.

Screen 6



Detail view of a photograph where beautiful photograph with parallax effect will be shown (Just need to move your device. :P). This layout includes main photograph, photographer's name, image added date, share functionality, download image functionality, set wallpaper functionality and like image functionality.

Screen 7



App widget would be simple, showing the liked images in a ListView.

Key Considerations

How will your app handle data persistence?

Allure mainly uses Firebase for data persistence. User login/information and photographs liked by the users will be store on Firebase. I will also user SQLite and Content Providers to store the liked images locally so that the users could see the photographs even if they are not connected to the internet for better UI/UX.

Describe any corner cases in the UX.

- Even if the user has not logged in then also images can be stored locally using SQLite database and when the user logs in the liked images automatically get backed up to firebase and are shown.
- If the user is logged in, then the liked images will be stored both online and locally. Disabling the click on the like/unlike button until information is not successfully stored or removed from online database (Case: if user tries to continuously press like/unlike button) to remove redundancy.

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

- Firebase for data persistence, user authentication and backend.
- Volley for network calls.
- Design support library for material design.
- Google Play services for Analytics etc.
- Glide for data loading images and image caching.
LINK: <https://github.com/bumptech/glide>
- Butterknife for binding views and callbacks to fields and methods.
LINK: <https://github.com/JakeWharton/butterknife>
- MaterialDrawer for navigating to different screens.
LINK: <https://github.com/mikepenz/MaterialDrawer>

Describe how you will implement Google Play Services.

Google Authentication:

In my application user can like photographs which will be stored both locally and online using Firebase. If user wants to save information online, he/she first needs to authenticate using Google Authentication which is implemented using Firebase.

Google Analytics:

Firebase provides very easy and effective way of integration with Google Analytics in application. In Allure when user sets a photograph as a wallpaper the information is stored using Google Analytics.

Next Steps: Required Tasks

Task 1: Project Setup

- Create new project.
- Configure libraries.
- Add models for data.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity.
- Navigation Drawer and add the items.
- Build UI for Categories.
- Build UI for Featured Photographs.
- Build UI for User Profile.
- Build UI for Detail View of Photograph.
- Build UI for Settings.
- Build UI for Search Section.
- Design graphics for search, no internet and likes.
- Design icon for the app.

Task 3: Implement task associated with the backend

- Implement Google Play Services for User Authentication.

- Use Unsplash API to fetch photographs in various sections like popular, oldest, latest, featured photographs.
- Provide mechanism to save image information liked by user using firebase and SQLite.
- Provide features for sharing image over other applications.
- Download and set photograph as wallpaper functionality.
- Checking network connection and displaying liked images from local database when no network access.

Task 4: Handle exceptions and improve UI

- Animations when opening profile fragment.
- Animation when user likes a photograph.
- Activity transition animation for lollipop and above.
- Runtime user permission for android M and above.
- Handle exceptions like when no internet access.
- Add progress bar wherever required.
- Add graphics wherever required.
- Add licenses and terms.

Task 5: Make Application compatible

- Implement Material Design elements.
- Check string resources for localization.
- Make app compatible with tablets.
- Check for right to left alignment.
- Check for landscape mode.
- Check for different android versions.

Task 6: Get feedback from code reviewer and friends

- Get feedback from code reviewer and friends.
- Implement suggested feedback.
- Publish on Google Play Store.