GitHub Username: 3Heads6Arms

Zoom Point

Description

Zoom Point is a photo/photography application for photographers & people who love beautiful photos. Zoom Point gathers photos from popular sources to many photographers like **Unsplash**, **Flickr**, **500px**, **SmugMug** and **Instagram** (As for Capstone project, only Unsplash will be implemented). Zoom Point allows users to *subscribe to favorite collection**, to find their inspiration for photography or maybe just some new *wallpaper for their devices*** all in one place.

* May vary between photo sources, some doesn't have collections (like instagram for example)
** May vary between photo sources, due to legal term some might not allow their photos to be taken.

Intended User

Zoom Point targets photographers, inspiration seekers or just people who would appreciate beautiful photos/drawings

Features

List the main features of Zoom Point:
- Gallery of new pictures
- Gallery by Collection
- Featured collections
- Curated collections
- Create collections & add pictures to selected collection
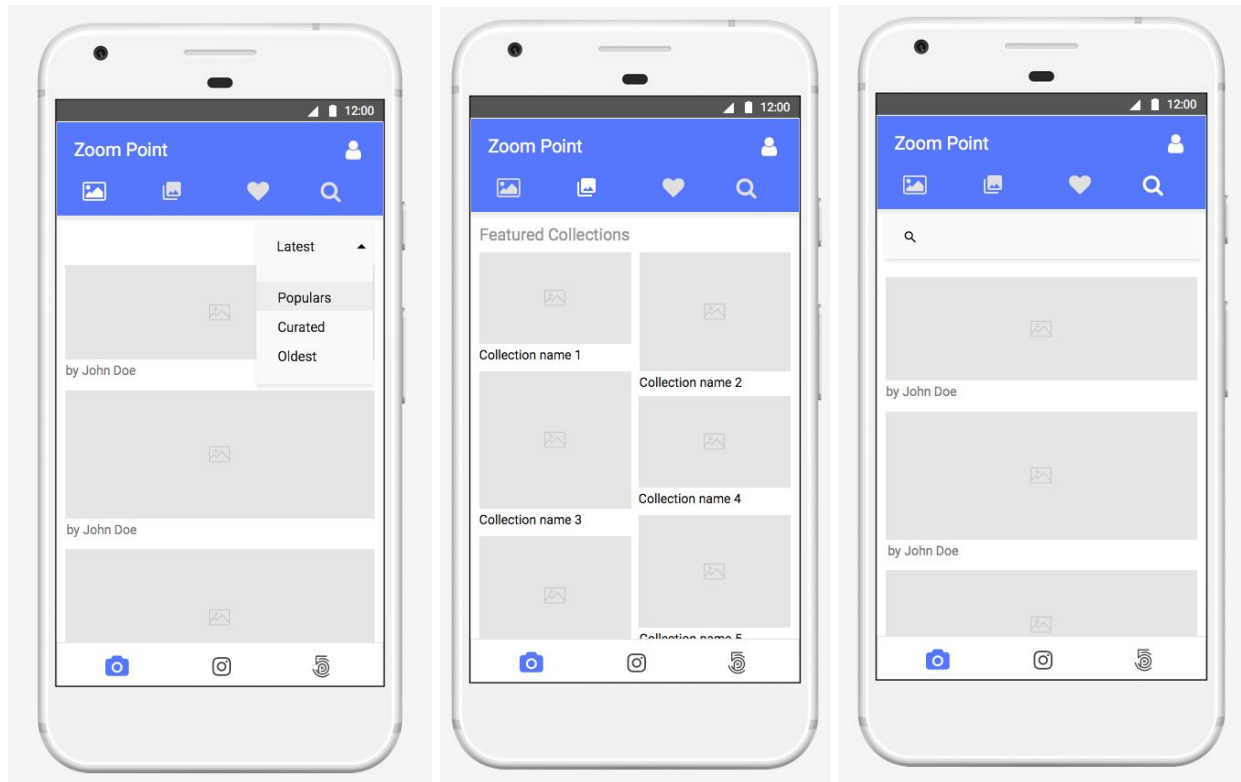- Like/favorite photos
- Set wallpaper
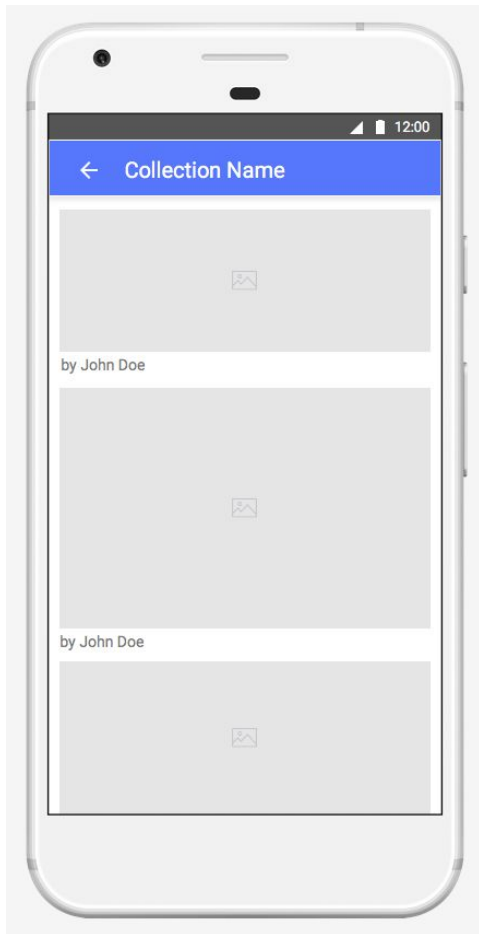
User Interface Mocks

Screen 1
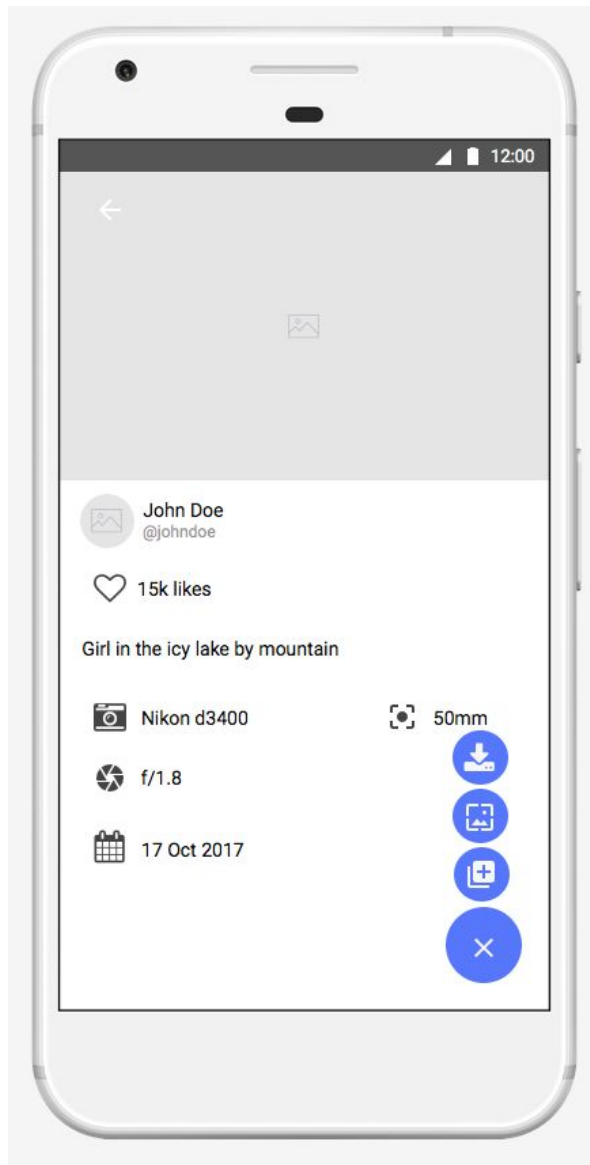

Login Screen

Screen 2

Main Screen. Displays pictures, collections, search results & liked pictures by current user

Screen 3

Displays pictures from Collection

Screen 3

John Doe
@johndoe

15k likes

Girl in the icy lake by mountain

Nikon d3400   50mm

f/1.8

17 Oct 2017

Shows picture with its information

Screen 4
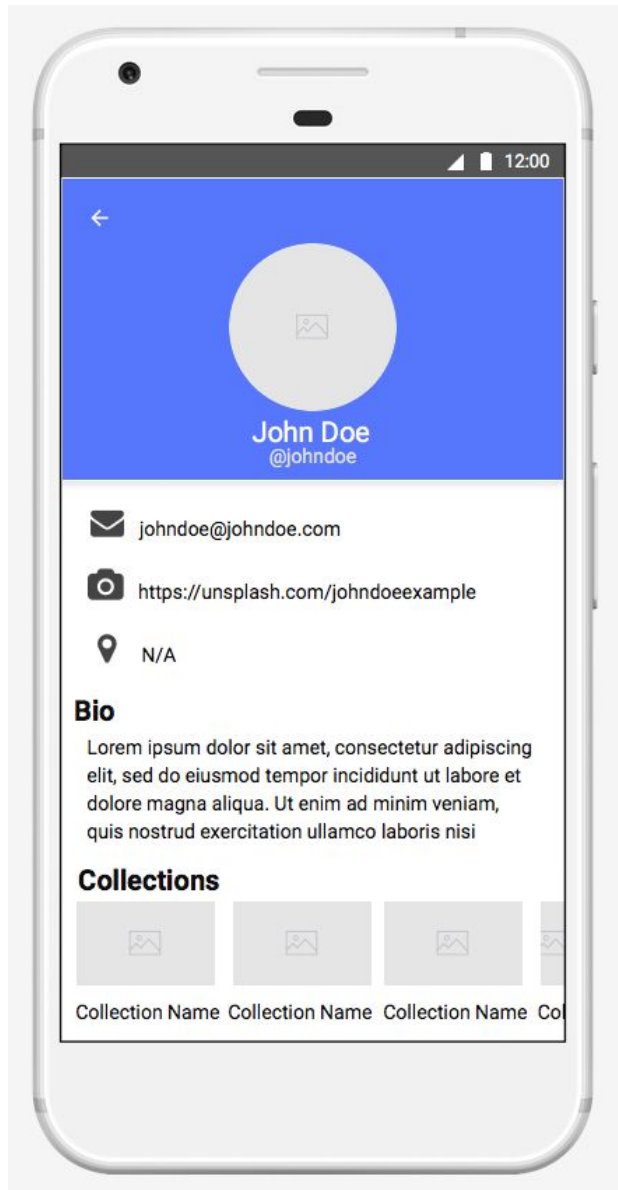
UserProfile Screen, shows user's information & user's collections. Allow updating profile if user is looking at their own profile.

Key Considerations

How will your app handle data persistence?

The app will persist all data loaded from server with SQLite via ContentProvider, which will ensure data when the app is required to work offline.
NOTE: Images might not be loaded without internet connection.

Describe any edge or corner cases in the UX.

The app allows preview pictures and few similar features, it will ensure any changes like rotation or returning from homescreen to be recovered just as it was.
Quick scrolling might cause lag on lower-end devices

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

- Glide for loading & caching images
- Retrofit for handling API communication to fetch data
- ButterKnife - Views binding & events binding

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

- Google Mobile Ads for monetization
- Google Maps - Some of the photos have location included, this will allow user to have vague idea of where the picture is taken

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup
- Configure gradle with libraries that will be used
- Set up POJOs to match json model from Unsplash's API.

Task 2: Implement UI for Each Activity and Fragment
- Build UI for LoginActivity
  - Implement UI views
  - Handle registration button
  - Handle login button
  - Setup default values when login
- Build UI for MainActivity
  - Implement Tab Navigation UI
  - Implement ViewPager
  - Implement Fragment for Daily Pictures
  - Implement Fragment for Featured Collections
  - Implement Fragment for Searching Photos
  - Implement Fragment for Favorited pictures
- Build UI for PhotosActivity
  - Reuse Fragment for Daily Pictures
- Build UI for PhotoDetailActivity
  - Implement required UI to show Photo and its data
  - Implement Actions on the picture
- Build UI for UserProfileActivity
  - Implement UI to show user data
  - Implement Edit action if the profile belongs to current User
- Build UI for EditUserProfileActivity

Task 3: Implement Google Play Service
- Implement Google Play Admob service
- Implement Google Map Service, to show location on map where the picture is taken if available

Task 4: Handle Edge Cases

- Handle when the list is empty
- Handle when internet connection is not available

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**"