Checkpoint (2): App Requirements and Design

Dave Becker, John Choi, Jonathan Nutter

This checkpoint entails envisioning and designing a mobile app that meets the requirements specified in the "High-Level Requirements" section.

Due Date: Friday, February 12, 2021, 11:59 p.m.

Points: 5/70
Submit: A SINGLE design document that shows:

- Notes of your app conception, specifically:
    - Narratives (one or two paragraphs describing how someone uses your app and why ***your*** app is necessary to achieve certain goals);
    - A UML class diagram of your app's domain model designed following the object-oriented approach discussed in class;
    - A ***categorized*** list of use cases describing users' interactions with the app (example categories can include "Account Management," "Mapping," and so on);
    - Your app's ***relational model*** that describes tables and relations in your relational database schema (you need to include the relational model in your document, not *just* an ER diagram);
    - Sketches (drawings of people using the app as you envision, how the app connects to external services, and so on);
    - Screen layout mockups along with screen flows showing how clicking different UI elements navigates the user among mocked-up screen layouts;
    - The external (Internet) service your app will use; and
    - The device sensor your app will use.

Note: Your screen layout mockups, screen flows, and sketches may be drawn by hand, but they ***MUST be legible.***

Show to Grader: Your app design document with the information above. You should explain your app to the grader during your group's meeting with him/her.

Evaluation Criteria: Clear evidence of the object-oriented design process.

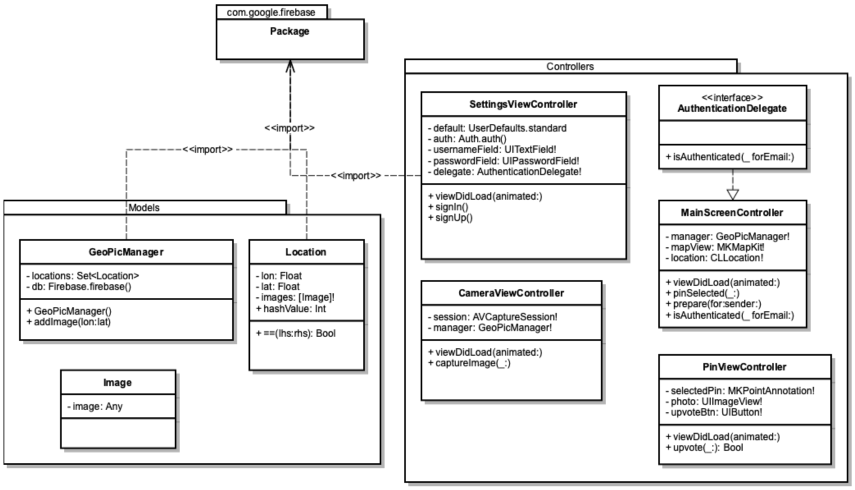
Narratives:

Our iOS app, “GeoPic”, will be a GPS-based scavenger hunt photography sharing app. During COVID-19 many people have started to learn photography. GeoPic will allow users to view beautiful photos that other people have taken. However, to view these photos, they will have to physically go to the place they were taken. This lets aspiring photographers gain insight into the techniques that the photographer used such as the lighting, angles, objects in the foreground, etc. Users can then practice these techniques by trying to recreate the photos or they can look at it as a scavenger hunt to find cool areas that people thought were photo-worthy. Our goal is to inspire beginner photographers and provide a fun way for people to get out of their homes to explore the areas around them.

The application will open to a full screen map with a large camera button and a settings button. The map will have pins that represent pictures. When the user clicks the camera button, they will be prompted to take/upload a picture. When they take the picture, the user will be prompted for confirmation that they would like to use the photo, then their GPS coordinates will be saved, and a new pin will be made for that picture. Pins are surrounded by a small circle representing the radius that it can be viewed. When a user enters the radius of a pin, it will change color and the user can select it to open the picture in a full screen modal. The modal will consist of the person who took the photo’s name, the date it was taken, the score, and an up/down arrow allowing you to like/dislike the photo. The settings page will have options to log out and change user information such as name, password, etc.

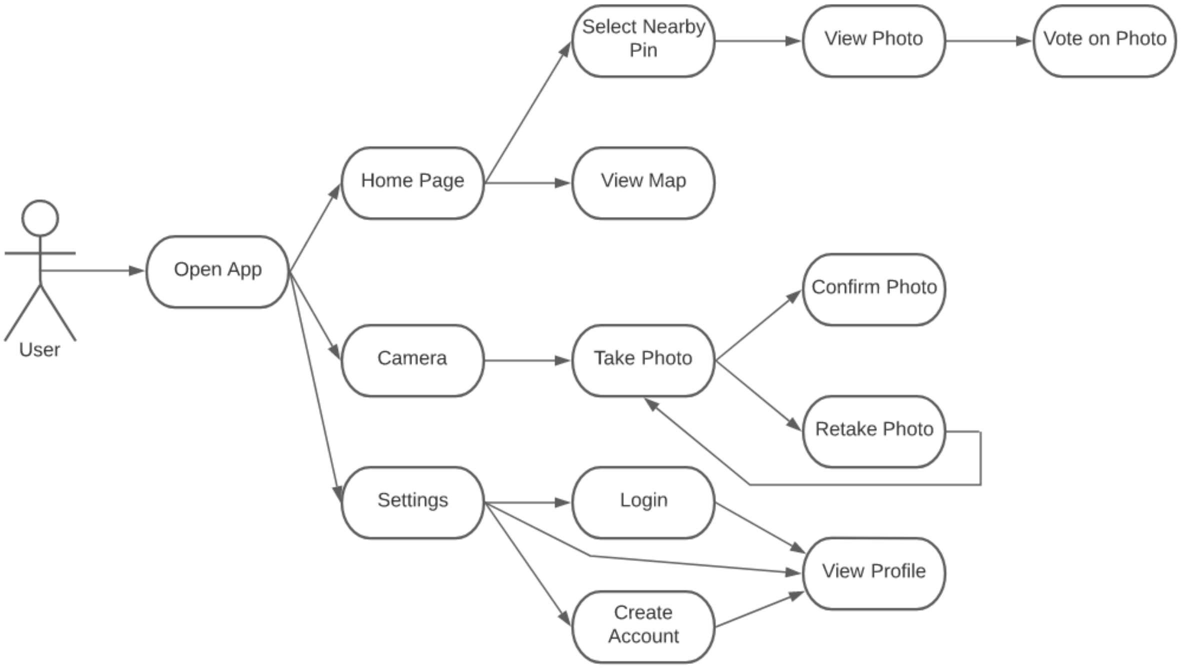
We plan to use Firebase to handle users, account authentication, and store photos/location data. The device sensors we will be using are the camera and GPS. If we have time, we would like to add augmented reality to our app so that users can see pins through their camera, walk up to it, press it, and the picture opens.

UML Class Diagram: (John)



List of Use Cases: (Jonathan)

- Main Screen
  - User views pins on map
  - User sees their current location
  - User selects pin to view corresponding photo
  - User is able to vote on the photo
  - User can select the settings page
  - User can open their camera to take a photo
- Camera
  - Users is able to take a photo
  - User will then see a preview of their photo
  - User then confirms photo submission or retakes photo
- Settings
  - User can enter login information and login to an existing account or choose to create a new account
  - When creating an account, user enters relevant information
  - After successful login, user is able to view profile information
  - If already logged in, profile information is shown



Relational Model: (Jonathan)

User

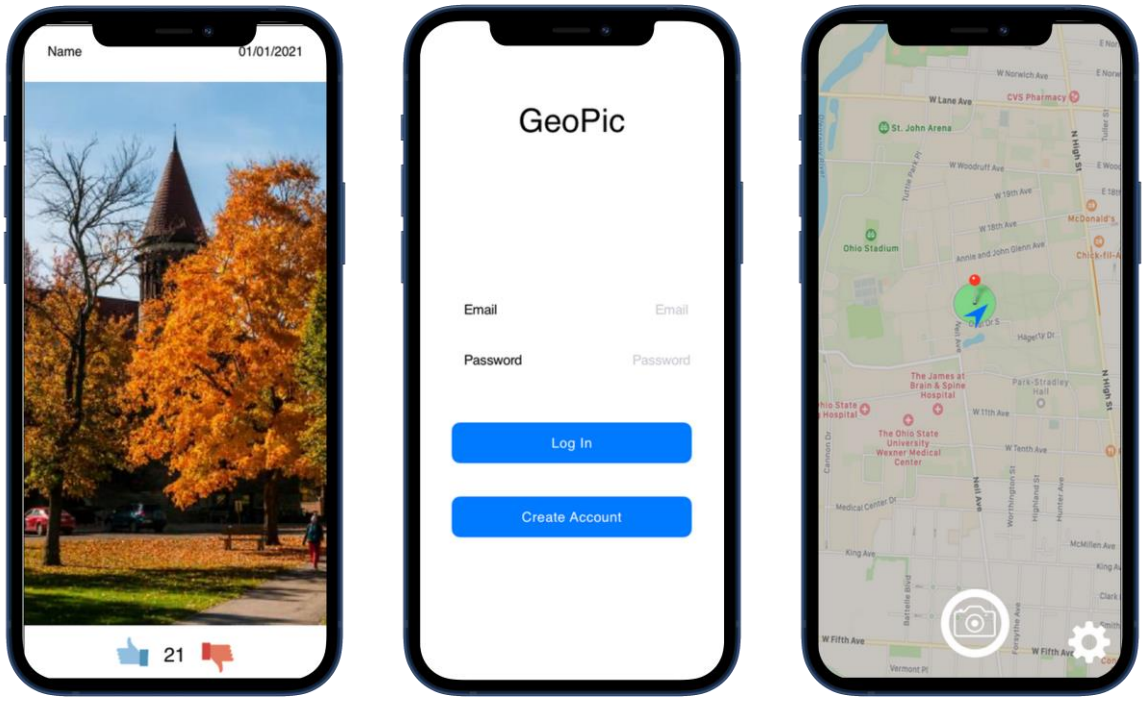
User_ID	Email	Name	Password

Photo

Photo_ID	User_ID	Latitude	Longitude	Score

Foreign key User_ID references User

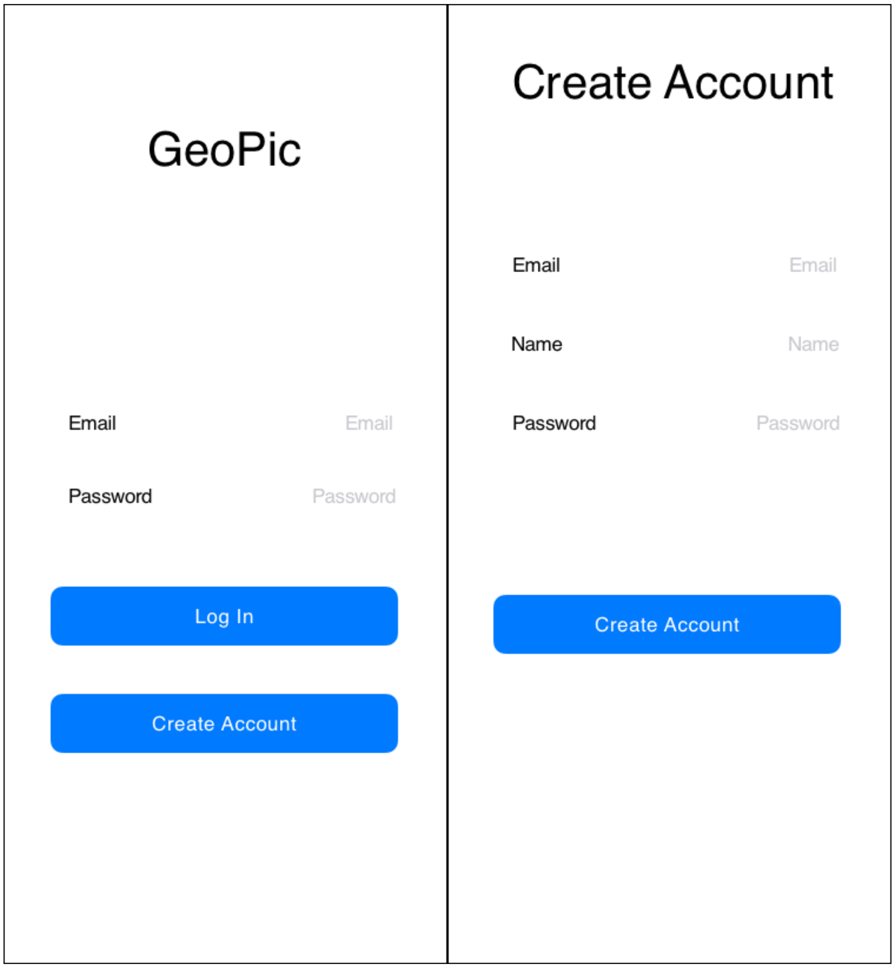
Sketches: (John)



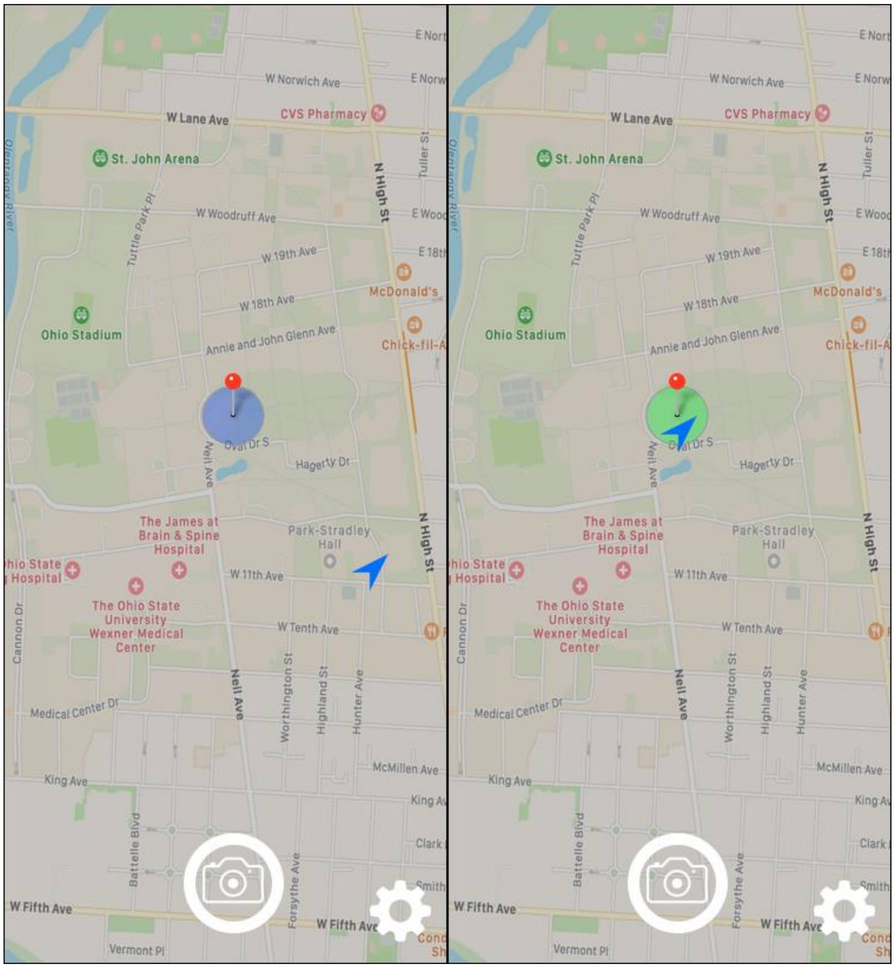
Data that persists throughout multiple user sessions: Images, authentication data, location data, image upvote data. External Service Used: Firebase

Screen Layout Mockups: (Dave)

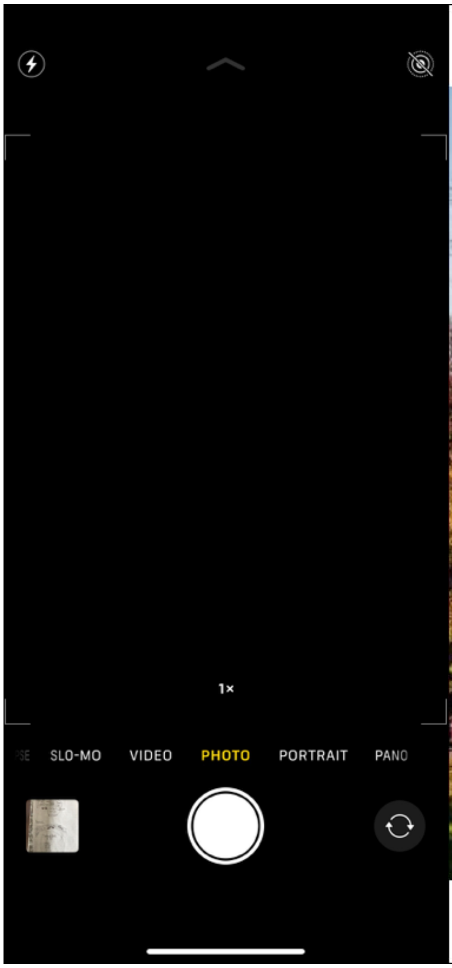
- Login screen and create account screen.

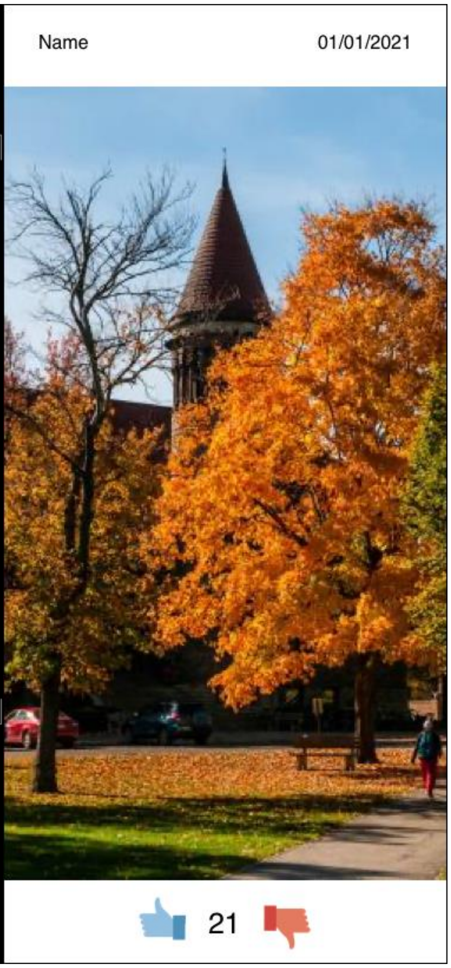


- Home map view when inside and outside radius of a pin

  - 

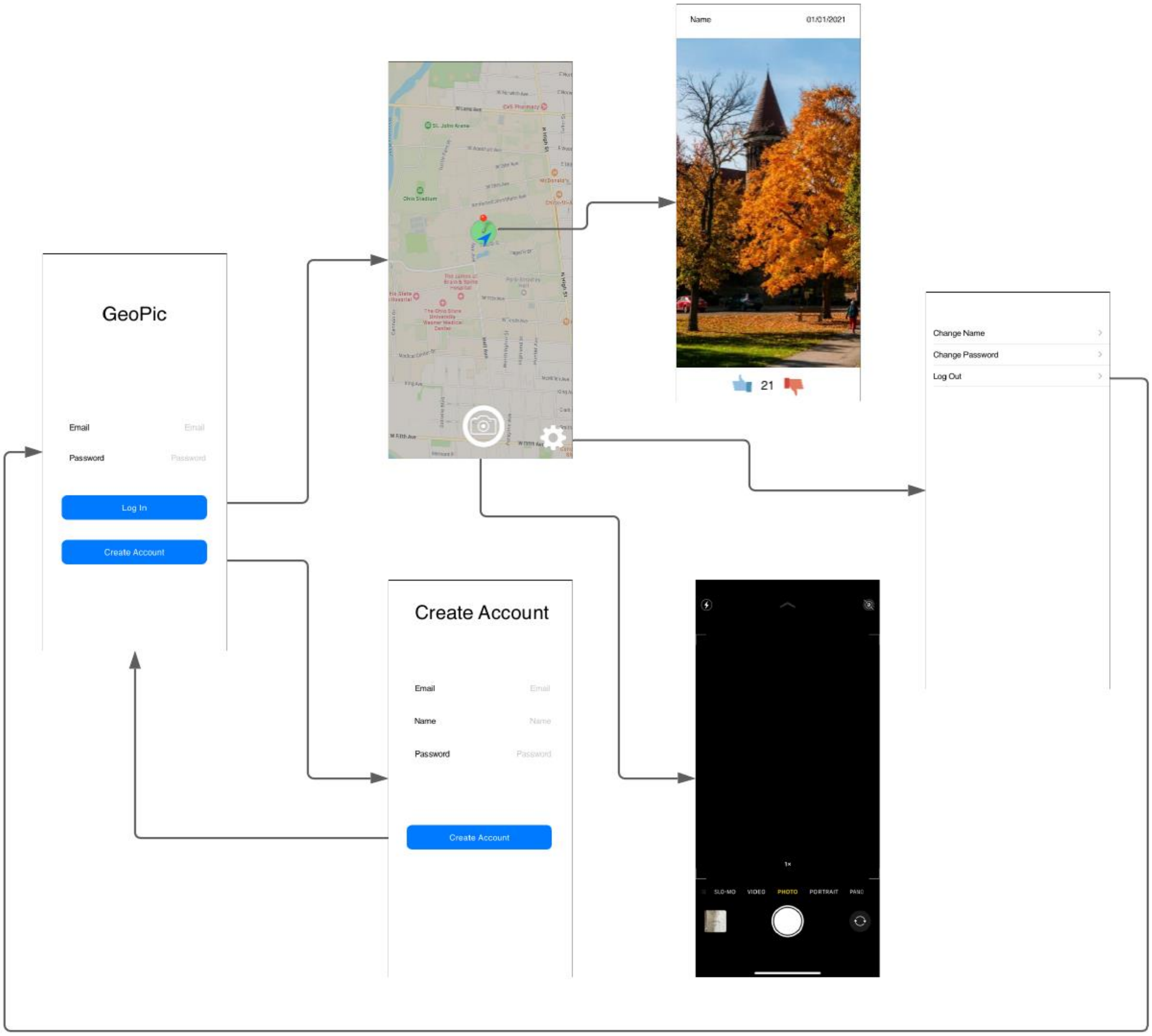
- Camera view and photo view


1×
SE
SLO-MO
VIDEO
PHOTO
PORTRAIT
PANO


Name
01/01/2021
21

○

Screen Flows: (Dave)



External Service Used:

- Firebase Authentication
- Firebase Storage

Device Sensor:

- Camera
- GPS