Team Senioritis

Alex Collins

Bryce Brady

Connor Ivy

Kevin Valdez

Emmie Chng

Third part of project:

We chose to work on both Module 3, and enhance and refactor the work done in Module 1 and 2.

Module 1: Take user photo input and export it into KMZ file

Module 2: Visualize photos on dynamic map like google earth

Module 3: Allow users to send images in an email to their chosen recipients

User Stories:

1. Extract metadata from images
   1. From an uploaded image, access the metadata and extract latitude and longitude coordinates
2. Create KML file
   1. From uploaded images, generate a KML file
3. Display
   1. Create interface for users to navigate, upload images, and save KML file
4. Map with images
   1. Display images on a dynamic map for intuitive viewing without having to manually sort
5. Users save KML
   1. Users should be able to save the generated file to re-run on our dynamic map
6. Email image button
   1. Add a button to the vizualize pop up window that takes users to the email page with the image selected.
7. Get user input for email button
   1. Add a form that asks user for email address, subject, and message
8. Send Emails
   1. Given address, subject, message, and attachment send an email using our email address that we created.

Unit Testing:

We created test scripts in Django and made sure the buttons work as they should on the GUI.

Acceptance Testing:

A set of test images will be gathered to test the image to KML module. A test KML file will be created to test the dynamic map display module. For the user exposed elements, we will use the GUI and ensure all buttons work and that the website is fully functional.

Plan of work:

We completed the functionality of module 3. Users can send an email an image from the kml that they choose. Users are prompted for the address, subject, and message that they would like to send. In the map visualization a new button in the image popups allows users to navigate to this email page. We completed Module 1 and the web application is able to take the user’s photo input and export it as a KML file. Module 2 is also completed and the KML file data can be displayed on a map with the image as a pop up. It looks really good now. We would have to further refactor what we have as there are some areas that needs improvement.

General Description:

We will be creating an application that uses exif data to locate where a photo was taken and visualize it nicely on a dynamic map for later review. The main group of people we are hoping to help is those who go on site visits and take pictures for new projects, e.g. an Engineer assessing a treatment plant. The process for this now is multiple people take photos, dump them in a folder, and someone goes back in later to sort them by location/subject.

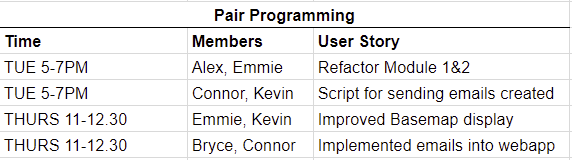
Details:

When the user sends their images through our software, they will get sorted by location and displayed on an interactive map for later viewing. We will be accessing the latitude and longitude from metadata of images taken. Then we will export it into a KML file and be able to display it on our own dynamic map like Google Earth. For the images of nameplate information, we will be able to extract the text for fast post processing. This view will be available after running the program and the user can then download the KML file to re-run it on our app, share it, etc.

Refactor:

1. Improved Aesthetic of the extract webpage.
2. Implemented buttons in the popups on the visualize webpage
3. Improved headers
4. Integrated the two web applications to work on one web application where we can access both the map and uploading
5. We did a code clean up and decreased the number of functions by using Django views
6. Styled the map to have markers for the images and it would zoom into the coordinates of the KML file when the file is uploaded

Pair Programming Chart:



Reflection:

The application is really coming together. Each part works on its own if the webpage was navigated to. And the application is capable of flowing between each module as needed. The aesthetics of the application make it look very professional, and we feel the application is also very useful. At the moment, the application is made in a way that would only work on local host, which for our use case is good enough. However, we believe it would not be hard to refactor this into an application that is deployable and useful to many people.