Mappics: Journeys through Photos Developer Guide

Braeden Neta

CS 460

2026

Table of Contents

Set up

This section talks about how to get my project set up outside of the one hosted on the compsci04 website.

After downloading the zipped folder here, and hosting the files on a server all that needs to be done is verifying that the functions work on the hosted server. All this means is to test the various functions of the project to make sure that the paths are working properly. This can be ensured after each of the different functions are used by manually checking the files and folders where data is stored. After confirming that the paths work, you're all good to go!

==========

Data Files

This section talks about the files that this project either creates or stores as part of the project.

My project stores JPG/jpegs in an uploads folder so that when a user looks at a pin's metadata they can see a higher quality image than what is displayed through the pin in the google map. Within that uploads folder, PNG's of each of the uploaded images will also be stored. These are created and used by the project in order to display a pin on the map for the user.

==========

Info Files

This section talks about the files that this project stores information in as part of the project.

There are two files that store information that my project uses:

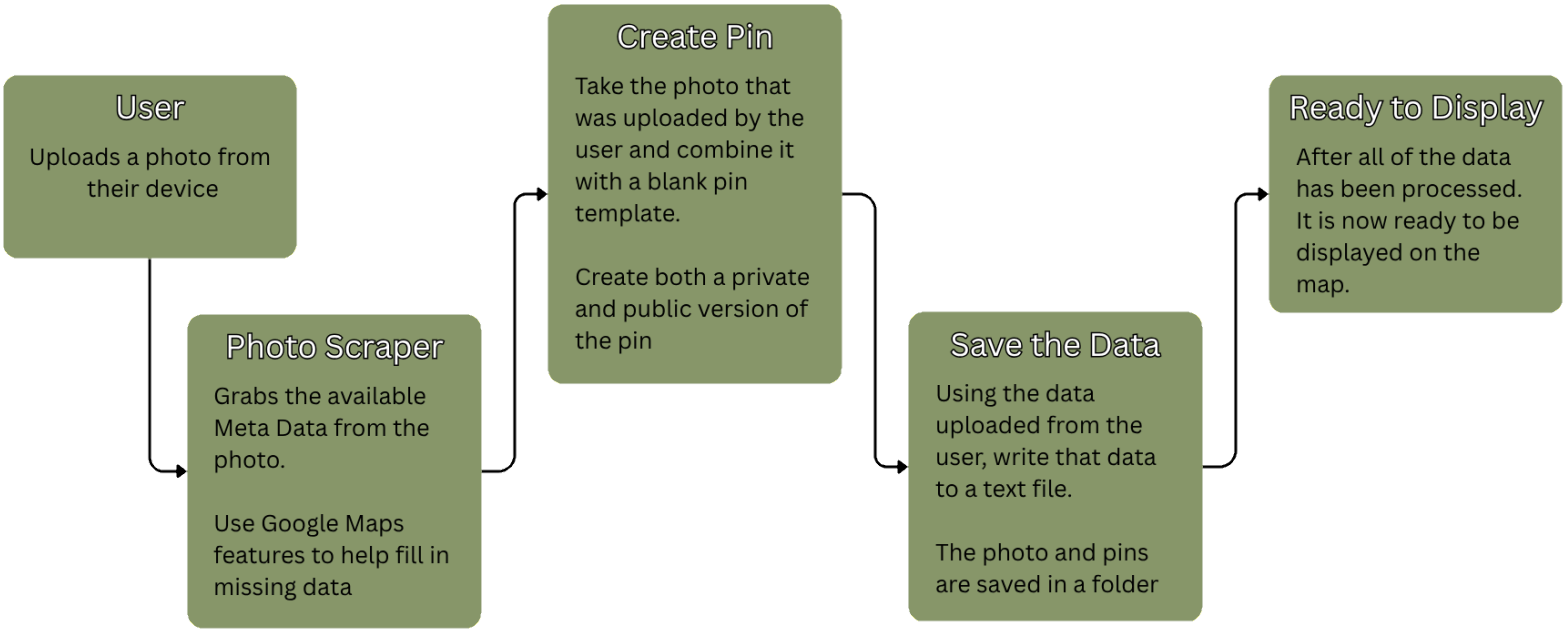
1. Accounts.txt:
    - This file contains the account information for users that have created accounts. All the user needs in order to create an account is a username and password. The username and a hash of the password are stored in this file which can be used to validate users that are attempting to login.

2. Photos.txt:
    - This file contains the meta associated with the photos that have been uploaded to my project. The metadata consists of the title of the image, the id of the image, the date that the photo was taken, the time that the photo was taken, the longitude and latitude that the photo was taken, and then the city/region that the photo was taken.
    - Each photos entry in this file is stored as a newline delimited json entry.

———
==========
———

Uploading a Photo

This section talks about the process of uploading a photo in my project.

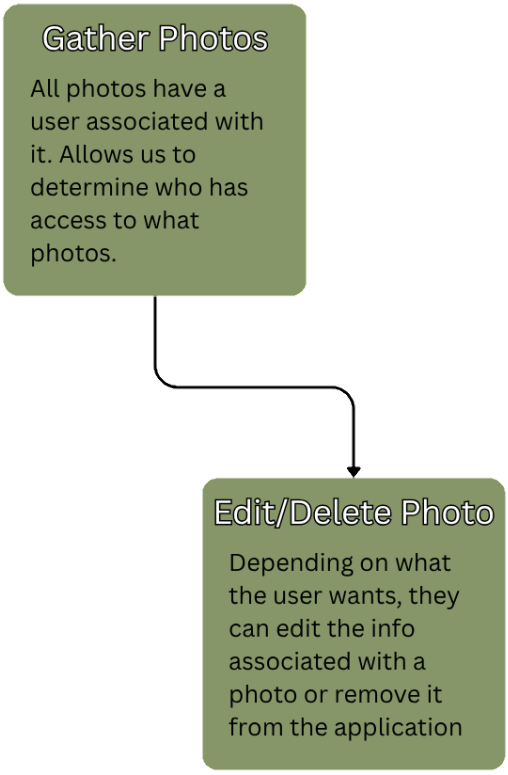


The process, as seen above, is as follows. A user picks a photo that they wish to upload and it is then passed to *photoScraper.php* which will extract as much of the relevant metadata as it can and display the information that it extracted back to the user for review. After the user confirms that information the image and its metadata are passed to the *combineImg.php* script which will store the metadata in a session variable and then create both a public and private version of the images pin. After this is complete it will run the *createImg.php* script which will create the JSON entry in the *Photos.txt* file.

==========

Editing a Photo

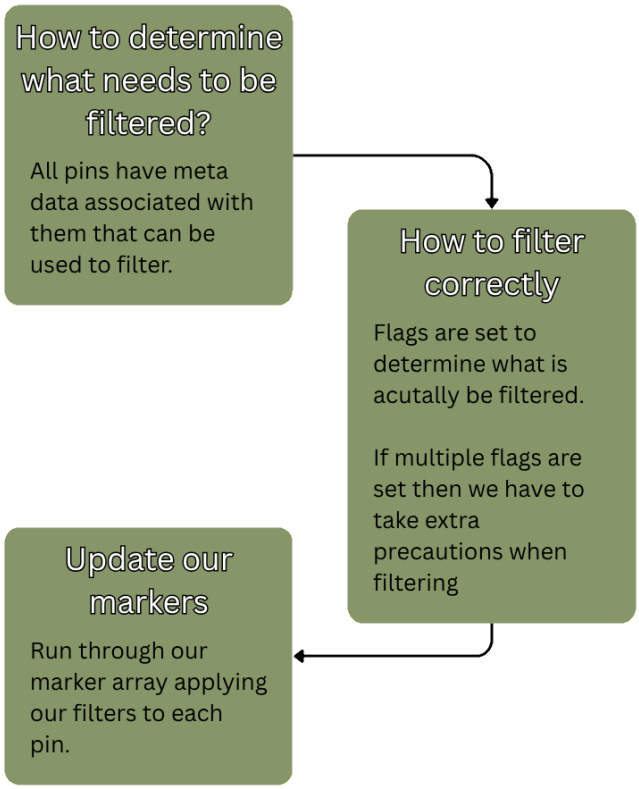
This section talks about the process of editing a photo in my project.



The process of editing a photo is as follows. When a logged in user wants to edit one of their photos and they have clicked on the button to edit their photos, the *Edit.php* script will be run which runs through the *Photos.txt* file and finds all of the photos associated with the currently logged in user. After that the user can edit the relevant metadata fields of the relevant photo and then click one of two buttons. The two buttons being Edit Me or Delete Me. The Delete button will remove the image file and pin PNGs as well as its json entry in *Photos.txt.* The Edit button will update the json entry in *Photos.txt* with the newly updated metadata.

Filtering Photos

This section talks about the process of filtering photos in my project.



The process of filtering for photos is as follows. All of the possible filters have flags associated with them that are either set to true or false when a user clicks on the filter button. If a filter's flag has been set to true then when we run through the array of markers we can safely ignore the filters with flags that were not set to true. Once we have checked all of the relevant filters, if we have determined that a marker does not meet the threshold for display then we hide it from view on the map.