INTERNATIONAL HELLENIC UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATICS, COMPUTER AND TELECOMMUNICATIONS ENGINEERING

GEOSNAP
FINAL REPORT

Project Team:

Anastasiades Alkinoos (20003)

Zina Eleni (20046)

Parasxos Stergios (20045)

Tsonidis Konstantinos (20023)

Tzegkas Konstantinos (20106)

Tziouvakas Stylianos (20066)

Supervisors:

Koureas Argyrios

Lantzos Theodoros

SERRES 2023-2024

Contents

Summary.....3

Introduction.....4

1. Methodology.....6

2. Implementation.....9

3. Timetable.....13

4. Results-Conclusion.....15

Summary

The "GeoSnap" application was developed in order to provide users with the ability to post and share photos using a geographical map, Google Maps, while post data is securely stored in the database of the Firebase API. By integrating the function of “pins”, the app offers a new dimension to social connectivity, allowing users to explore the world around them through photos that other users have shared. For the best user experience, "pins" are presented in a "cluster" format to optimize the visibility of posts in areas with a high number of photos. Also, the application offers flexibility in editing photo data and description before posting the final post. The final post, like all other users, is accessible for browsing its data in search which supports quick display of post photos with the help of the Glide library.

Introduction

The application implements four main use cases, namely posting, searching, editing photo data and viewing posts.

Users can:

- Upload one or multiple photos to the same post with a single description and location. Each post can be categorized using a tag to facilitate its search.
- Edit the photos of their post before posting it with their own data, such as changing the location and date if they do not exist.
- View all posts in the database, which are placed on the homepage map using pins.
- Search for posts according to the tag of each post.

The procedure followed by the team was as follows:
Week 1: The “base” of the application was implemented, i.e. the creation of the basic UI, the addition of the Google Maps API service to integrate the map into the application and the Firebase API to provide a cloud database. Also, extracting the information of the photos (metadata) and finding the user's location on the map was accomplished.

Week 2: The ability to upload a photo (initially with no tags) to the database was developed. Later, an attempt was made to upload photos from the app's camera, but due to some drawbacks of this feature in terms of displaying photos, it was never added. Finally, the function of tags was implemented, where the user selects one of the available tags and his selection is stored in the database.

Week 3: Displaying posts on map was implemented. In addition, searching posts based on the tags chosen by the user was added. Also, the validity checks of the photos data entered by the user for the existence of the location were implemented, as this is an important condition. Finally, the user can now select multiple photos.

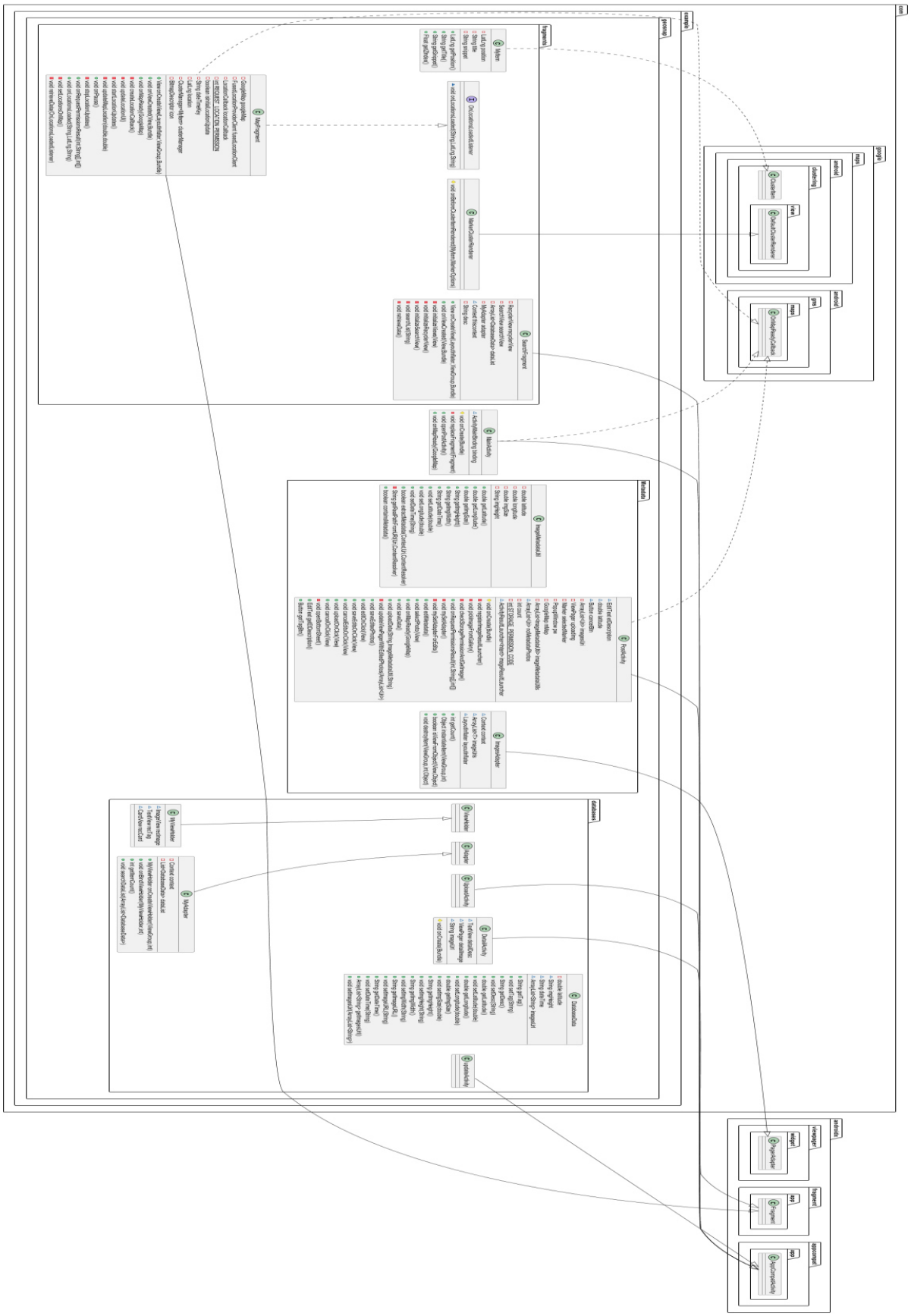
Week 4: The structure of the database was changed so that it can now store data from multiple photos in each post. Also, by interacting with a pin on the map, the user can see the date and tag of the post.

Week 5: Evolution of the app by adding increased user authorization checks of app permissions for the app to access the user's device photo folder for various Android versions.

Week 6: Code improvements and now the user has the ability to edit the data of photos that do not have location and date information.

1. Methodology

Application structure and design (UML):



Breakdown of the main classes of the application:

The MainActivity class represents the home screen of the Android application and includes the map from the Google Map API integration and a navigation menu in other parts of the application.

The DatabaseData class represents a data structure used to store information about data stored in a database, in this case Firebase.

The DetailActivity class was developed to display details for a specific element in the application. It uses TextViews to view description, date, and tag, as well as a ViewPager for images. The code allows you to load and view images from URLs using the Glide library.

The MyAdapter class represents an adapter for a RecyclerView. This adapter is used to display and manage a list of items (DatabaseData) in a RecyclerView.

The MyViewHolder class is a subclass of RecyclerView.ViewHolder and is used to assemble the elements of a RecyclerView into it.

The MapFragment class represents a Fragment that displays a Google Maps map and manages the display of markers/pins on the map that correspond to geographic locations stored in a Firebase database.

The MarkerClusterRenderer class is a subclass of the DefaultClusterRenderer provided by the Google Maps API. This class is used for the renderer of clusters and unique items on a Google Map.

The MyItem class implements ClusterItem from the Google Maps API. The use of the class is to represent a specific item to be used in clusters on a Google Maps.

The SearchFragment class is a Fragment, designed to display a search list. When creating the view, it initializes the necessary Views, including recyclerView and searchView. RecyclerView is used to display the data, while searchView provides the ability to search the list. The data is retrieved from the database in Firebase using retrieveData() and the list is dynamically updated during the search. The class utilizes an adapter (MyAdapter) to test and update the recyclerView with search results.

The ImageMetadataUtil class is responsible for extracting metadata from an image. Provides methods for reading GPS coordinates, dimensions, and date/time from an image. In addition, it contains a method (containsMetadata) that checks if valid metadata exists, i.e. if GPS coordinates, dimensions and date/time have been successfully extracted from the image.

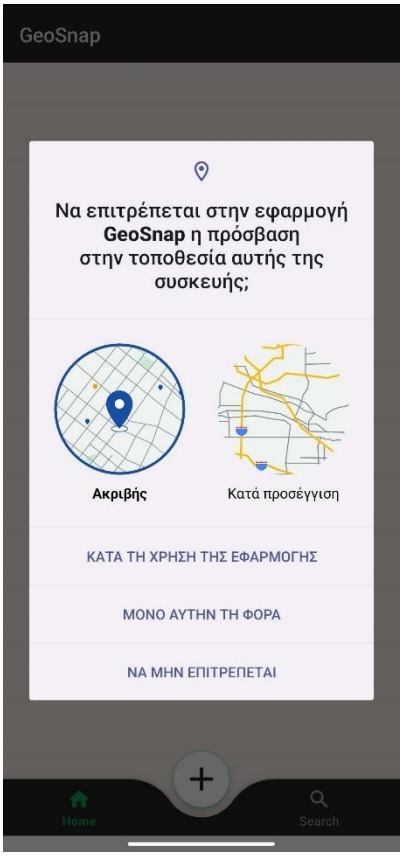
The ImagesAdapter class is an adapter for managing images in a ViewPager. It is used to link images to the user interface when using ViewPager.

The PostActivity class represents an activity that allows users to upload photos to Firebase, including associated metadata such as location and date. This activity provides functions such as selecting and previewing photos, editing metadata, selecting tags, and storing the data in the Firebase Realtime Database. The activity also provides metadata editing features before saving, as well as editing options for photos that don't contain original metadata. The activity manages sending photo data to Firebase Storage and storing it in the database.

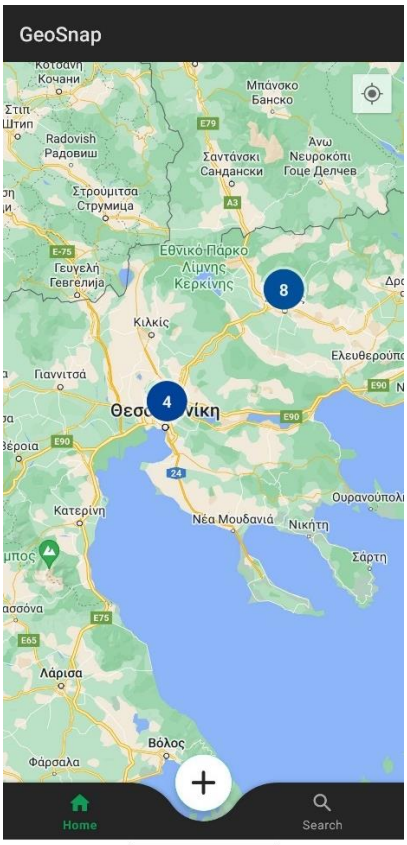
2. Implementation

Application icon:

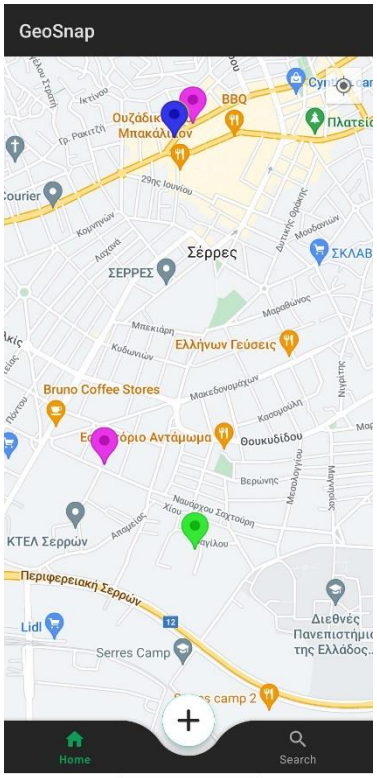
Application running:



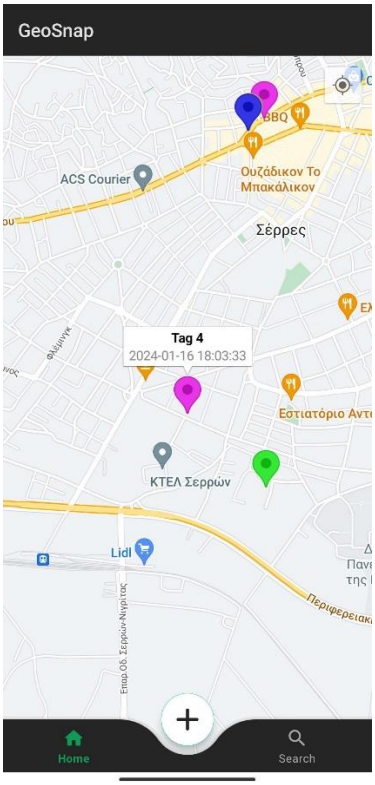
Permission to use location



Home screen



Home screen zoomed in



Displaying post on map

Post creation (by pressing the button in the middle):

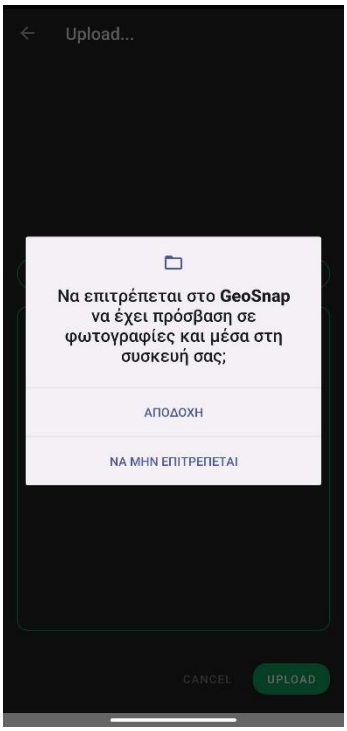
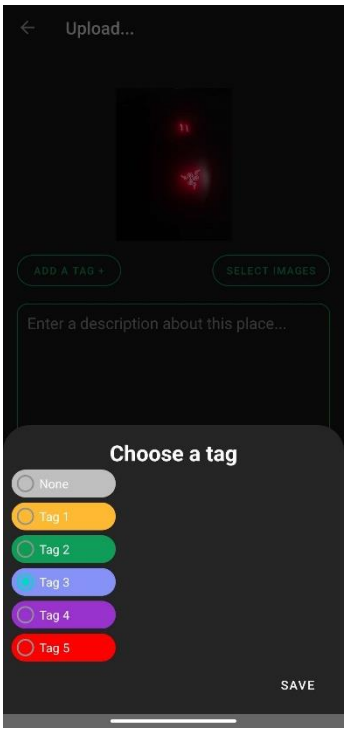
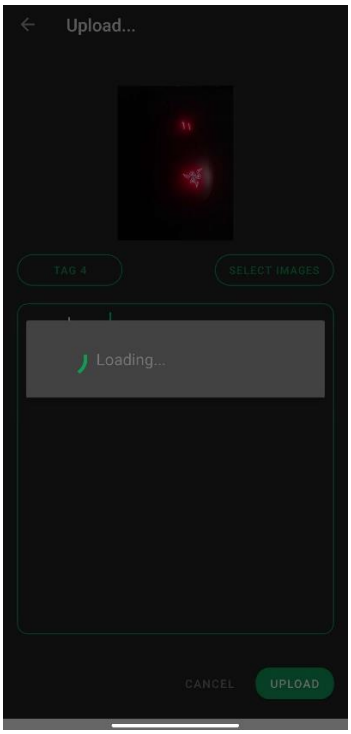


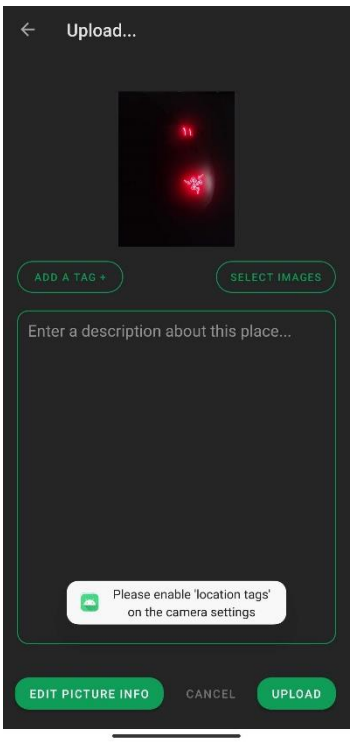
Photo access permission



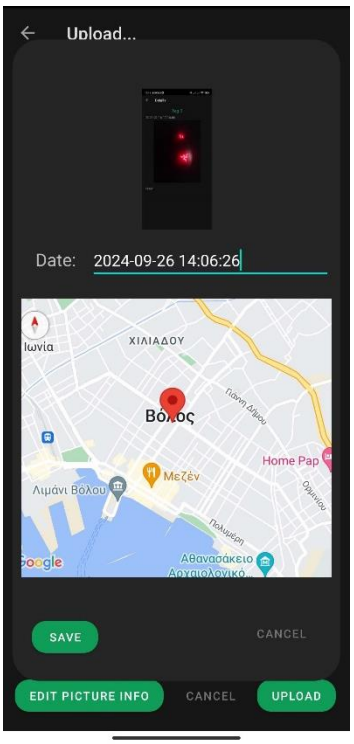
Choosing a tag



Loading tag on post

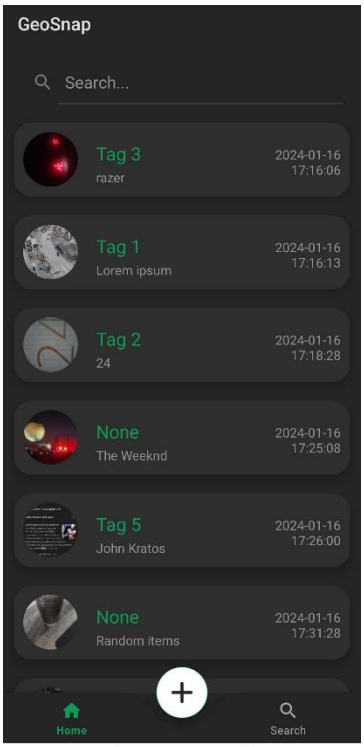


Location error message

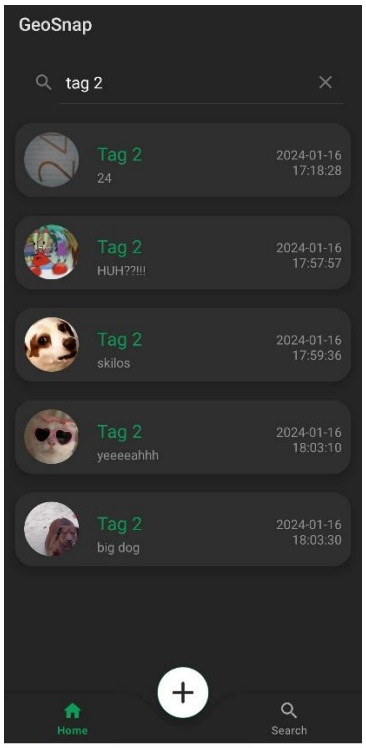


Editing date (EDIT PICTURE INFO)

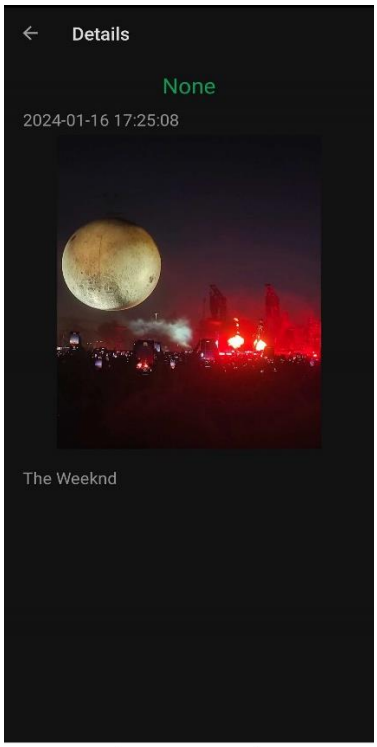
Posts search (by pressing the “Search” button):



Displaying posts search

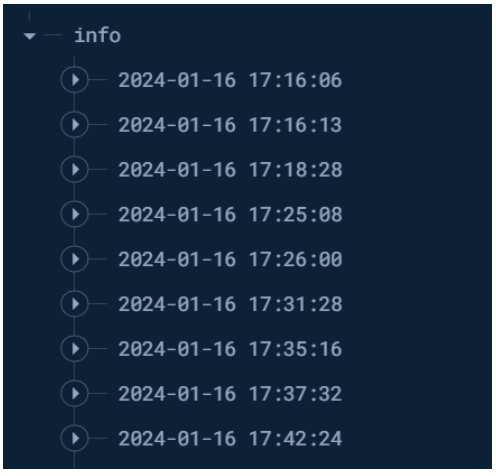


Search based on a tag



Displaying post data

Displaying stored posts in the database:



3. Timetable

Implemented tasks:

	Title	Assignees	Status	Starting Date	Deadline
1	Base UI creation #4	AriZume and Tziouve	Done	Nov 3, 2023	Nov 11, 2023
2	Integrate Google Maps #3	Tziouve	Done	Nov 3, 2023	Nov 5, 2023
3	Initialize DB for pictures #2	Helen1Z and kapad...	Done	Nov 4, 2023	Nov 9, 2023
4	Manage pictures info for post (Metadata) #5	d00om	Done	Nov 5, 2023	Nov 9, 2023
5	Get users location and place on map #6	alk-an and AriZume	Done	Nov 3, 2023	Nov 12, 2023
6	Post function w/o tags #10	AriZume, d00om, ...	Done	Nov 6, 2023	Nov 10, 2023
7	Use camera button to upload to DB and posts #28	d00om	Trashed	Nov 12, 2023	Nov 13, 2023
8	Work on the "add tag" in post UI + save tag on DB #25	Tziouve	Done	Nov 11, 2023	Nov 12, 2023
9	Add default tags/filters #24	AriZume and Tziouve	Done	Nov 22, 2023	Nov 27, 2023
10	Get image location from base and place pin on map #31	alk-an and AriZume	Done	Nov 13, 2023	Nov 25, 2023
11	Allow users to search photos/posts by tag #29	kapadokos and Tzio...	Done	Nov 23, 2023	Dec 16, 2023
12	Ensure that the user has "Save location info" enabled before upl... #33	d00om	Done	Nov 24, 2023	Nov 25, 2023
13	Allow user to select multiple images from gallery #34	Helen1Z	Done	Nov 24, 2023	Nov 25, 2023
14	Retrieve image data from multiple images to store #35	AriZume	Done	Dec 2, 2023	Dec 4, 2023
15	View posts by clicking the marker on the map (Week 4 - 5) #41	alk-an	In Progress	Dec 1, 2023	
16	Upload data from multiple images in one child in database #42	Helen1Z	Done	Nov 27, 2023	Nov 30, 2023
17	Permission to access gallery #50	d00om	Done	Dec 4, 2023	Dec 5, 2023
18	Allow users to set custom Metadata to a photo #51	d00om	Done	Dec 9, 2023	Dec 19, 2023
19	Add settings				

Stats:

Summarized report of each team member:
Anastasiades Alkinoos: Worked on the part of code that involved accurately finding the user's location on the map from the application, placing his post as a pin on the map, displaying pins as clusters in distant view and displaying pins based on the color of the label.

Zina Eleni: Worked on the part of code that involved integrating the database into the application (Firebase API), displaying the posts data in the search section, selecting multiple photos in a post and aggregating these photos into the same post with their own data in the database. She also drafted the final report and creation of the UML.

Parasxos Stergios: Worked on the part of code concerning the metadata management of the photos, i.e. their processing by the user before final post, the validity check that there is metadata in the photos selected, as well as the permission of access to the photo gallery.

Tsonidis Konstantinos: Worked on the part of code that involved the integration of the database into the application (Firebase API) and the search of posts.

Tzegkas Konstantinos: Worked on the part of code related to the graphical user interface (UI), finding the exact location of the user on the map, adding the tags to the creation of posts, placing their post as a pin on the map and retrieving the data of multiple images. He also managed the GitHub environment.

Tziouvakas Stylianos: He worked on the part of code that involved the integration of the map in the application (Google Maps API), the addition of tags in the creation of posts and the search of posts. He also drafted the final presentation.

4. Results-Conclusion

In conclusion, with the GeoSnap app, the team has successfully developed an app to create and display posts according to user need. The application demonstrates the ability of the program to be easy to use by any user, as well as the need to configure the data of a post. While the program was evolving rapidly during the project weeks, the team was able to collaborate efficiently and complete the project, as well as the members to collect useful knowledge and experience in a practical way.