

## Assignment 2 Report

Github

I have uploaded my code to the following repository:

[https://github.com/binaryeric/sofe4640\\_assignments](https://github.com/binaryeric/sofe4640_assignments)

Please see Assignment2 Folder for my application source code.

### Overview

In this report I will discuss my implementation of the assignment deliverables and how I utilize LocationServices on the Android device. Additionally, I will talk about how I use Geotagging/Geocoding with Latitude and Longitudes to display information to the user and how I implemented the Geocoding class with an online dataset to import some sample latitude and longitude locations of Canadian cities.

My sample data was taken from here:

<https://www.canadacitylist.com>

I developed an application that has two fragments. In the first fragment, I display all my information from my SQLite database in a RecyclerView on custom cards. These cards contain the “address” field, and then the corresponding “latitude” and “longitude” fields. I wrap all the data for these items in a custom class called “LocationWrapper” which also stores the ID such that when a user wishes to edit the data, the ID is passed from the search fragment, and the appropriate data can be loaded into the Edit fragment. The ID is then used to delete or update any information in the SQLite database. On the main page I have also implemented a FilterView which allows the users to sort through all data that is currently loaded in the recycler adapter view. The user types in the address, and the filter attempts to match the string patterns in hopes that the user can find the desired geolocation.

In the second fragment, I allow users to edit the latitude and longitude of a location. I also query the geocoding while the user is editing to ensure the desired address is obtained. I have also implemented location services with this application. I use a button with the text “Use Current” which will automatically request the appropriate permissions as specified in the android manifests file, prompt the user if they do not have permission, and then it will call on the devices LocationManager from getResources. If the user has granted permission and all has gone well, my implementation will automatically change the text input fields and update the address for the user. Should the user want to do this manually, they can continue updating the latitude and longitude values until the desired location is selected. It is important to note that Latitude and Longitude coordinates are constrained. To ensure we don’t cause unexpected behaviours with Geotagging, Latitude coordinates must be between -90 and 90. The longitude coordinates must be between -180 and 180.

To update or delete entries, users can simply select the card they want on the main page. By default the user will view all database entries unless a search filter is applied to the dataset. When a user clicks on the desired card, they will be taken to the Edit fragment. The edit fragment will adjust such that if no input is passed, it will know to create a new entry in the database. If an edit operation is desired, the Search fragment must pass the associated ID with the LocationWrapper object to properly make changes to the table stored in SQLite. Once the desired changes are made, the user is given two options; they can delete the table entry from SQLite which is implemented by a button which knows the ID. Otherwise, the user can save the changes which will create an ID for the new entry and update the address and get the geolocation for the entered values.

If any issues with SQLite or Geotagging are encountered, I have implemented some fault tolerances and debugging to let the user know there was an issue with the request. I have done this using toast just so the user is aware in case there is an unknown issue on the backend.

## Deliverables

### Database and 50 Locations

On app start up, the application will load in a static variable of location pairs using the Geotagging feature and save 50 locations to the database for the user to interact with. I created a DatasetLoader object which handles this by checking the latitude and longitudes and if they don't exist, it uses geocoding to get the address and then adds the entry to the SQLite database.

### View and Query for Locations

Implemented on the main page, the first component a user will see is the Search fragment. By default, all of the locations are visible and they can query to view only desired locations. From here the user can also create new geotags for latitude and longitude pairs or editing existing pairs by interacting with data in the RecyclerView.

### Create, Edit, or Delete Locations from SQLite

Implemented as a fragment, the Search fragment will pass information to the Editor fragment to either create or edit a location. The user can change the corresponding latitude and longitude and update or create table rows with the corresponding GeoTags. When the user is done, they will be taken to the search fragment where they can view all data or search for other locations.

Users editing location data may also choose to delete it which is given to them using a Button; this will use the ID to remove the location entry from the database. Then it will return them to a new instance of the search view.