

SOFE 4640U

Mobile Application Development

Fall 2021

Assignment 2: LocationFinder Application

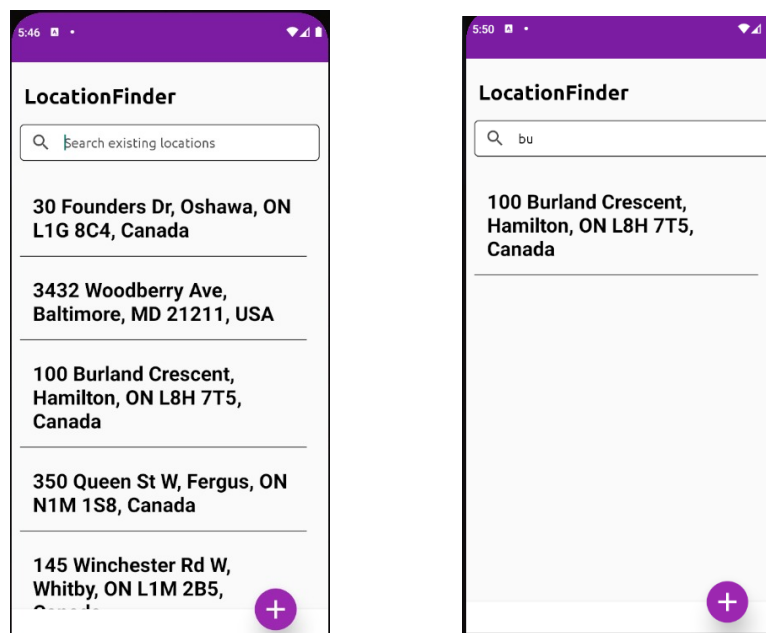
Xyron Brual - 100700268

Objective:

Practice Android application development with databases and location services

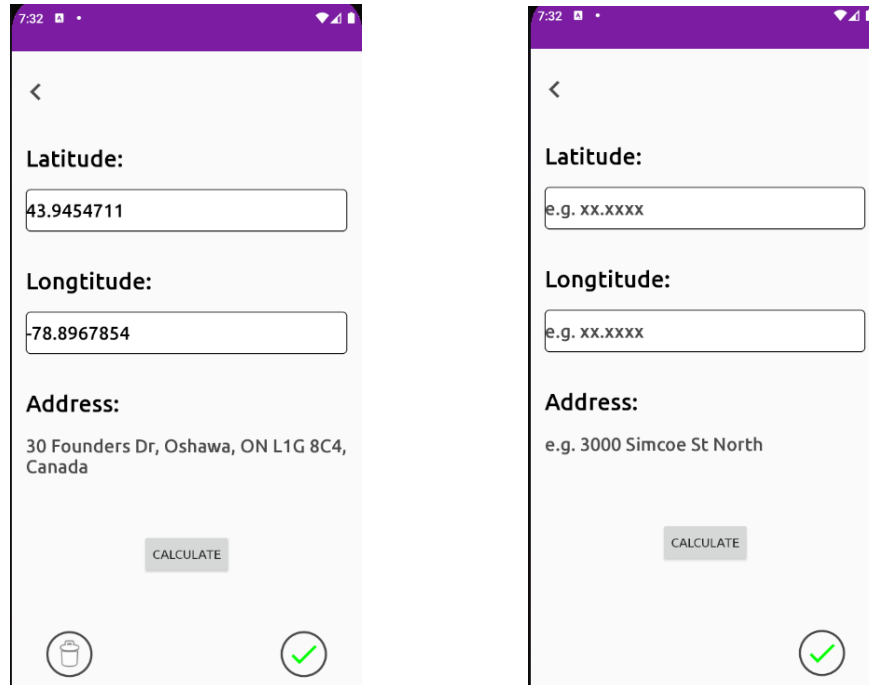
The task given to us was to develop a mobile application that made use of reverse geocoding and databases to obtain and save the addresses of locations given a particular set of latitude and longitude coordinates. To do this, I created two activities to handle user interactions as the view and made one locations table in my LocationFinder database to store locations. On top of this, I created three more classes to handle database interaction and act as the model for the application.

Within my LocationFinder database, there is one table called *locations*. Each entry of this table contains four attributes: an integer ID which acts as the primary key, a string for the address, and double values for latitude and longitude. In order to manipulate the entries in this table, I made two classes: LocDatabase which handles the add, delete, and edit functionalities, and the LocModel class which acts as a model for an individual entry. The LocModel class has the same attributes as entries in the locations table and because of that, an object of this class is created whenever an entry of the locations table will be added, deleted or edited. The functions in the LocDatabase class that handles adding, deleting and editing a location are addLoc(), deleteLoc() and updateLoc() respectively. The addLoc() function takes a LocModel object, retrieves its attributes, inserts them into the locations table with a new ID and returns true or false depending on the success of the operation. The deleteLoc() and updateLoc() functions also take a LocModel object and returns true or false based on success, but the deleteLoc() function specifically deletes the entry that has the same ID as the LocModel while the updateLoc() function replaces the values of the entry in the table with the same ID with the new values that came from the passed LocModel.



The above images are examples of the layout that I have for my first activity which is the HomeActivity. The purpose of this activity, as the name suggests, is to act as a home screen for the application and will display all of the existing entries in the locations table. As you can see on the right image, you can also use the search bar to find locations that have a certain substring in their address. The list of locations is displayed on the HomeActivity using a RecyclerView and my LocAdapter class. The LocAdapter class acts as an adapter to display the ArrayList of LocModel objects on the HomeActivity's RecyclerView and is also responsible for giving each existing location displayed its own on click function so that they can be viewed,

edited and/or deleted in the DisplayLocActivity. For this on click function, it will take the LocModel object from the adapter list, create a new intent for a result and send the object to the DisplayLocActivity with an action string “edit”. Along with clicking a location from the list, you can also click the add button on the lower right corner of the screen which will make a new intent for a result to the DisplayLocActivity with the action string “add”.



The next activity is the DisplayLocActivity that can be used to display an existing location in the database (the left image) or create a new location to be saved (the right image). When a location from the list is selected in the HomeActivity, the LocModel object is passed through allowing for the information on latitude, longitude and address to be displayed as well as the delete button. When the add button is clicked, no LocModel object is passed so the entries for latitude, longitude and address remain empty. Whenever you enter a new set of coordinates in the text boxes and then click the calculate button, the calculate() function will be called which will perform reverse geocoding using the entered values. This is done through the geocoder’s getFromLocation() function that returns a list of locations based on the given latitude and longitude. If a location is found, a new LocModel object will be returned with the respective address and latitude and longitude coordinates, and the address will be displayed on the screen. When the delete button is clicked, an alert will appear asking to confirm the decision. If confirmed, then a new intent will be made to the HomeActivity and it will pass the existing LocModel along with the action string “delete”. When clicking the done button, the circle with the green tick, then an intent will be made to the HomeActivity. If the current activity is from clicking the add button, then the intent will pass a new LocModel object with the values taken from the inputs and an action string of “add”. However, if the current activity is from clicking an existing entry, then the new LocModel to be passed will take the new values from the inputs but take the same ID as the existing entry and the passed action string will be “update”. Any of these actions will bring the user back to the HomeActivity and based on the resulting intents passed from the DisplayLocActivity, the HomeActivity will use the appropriate LocDatabase function to add, delete or update an entry in the locations table.