

Johvonne Keane
100784274
Wednesday November 6th, 2024
Mobile Application Development,
SOFE 4640U
[github](#)

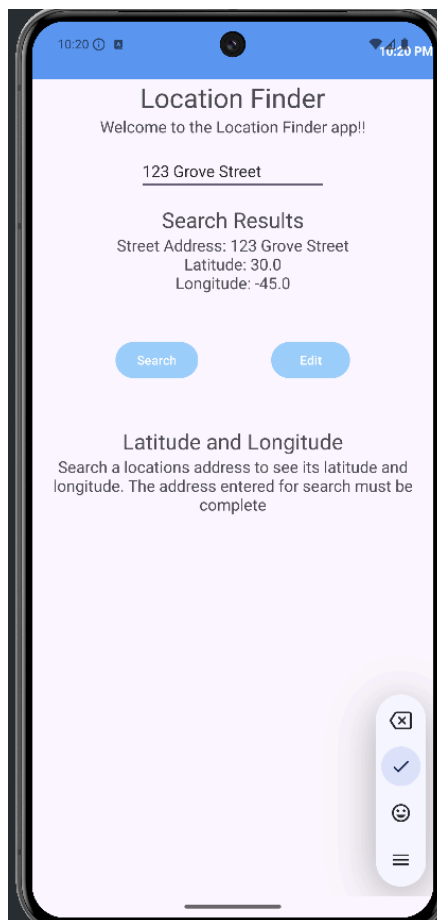
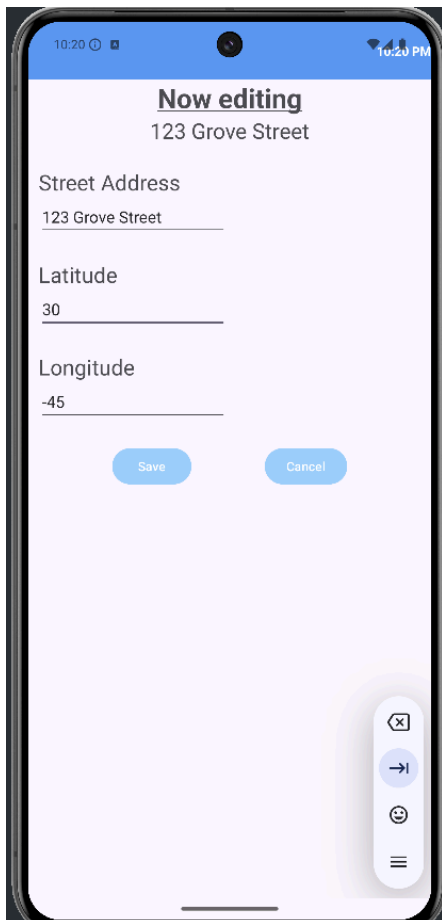
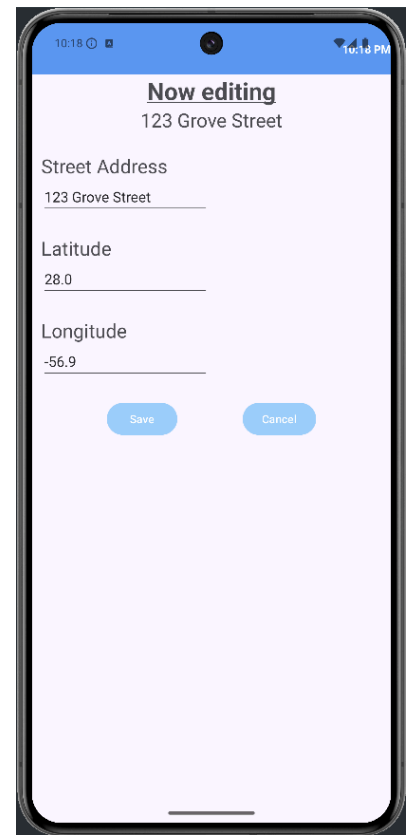
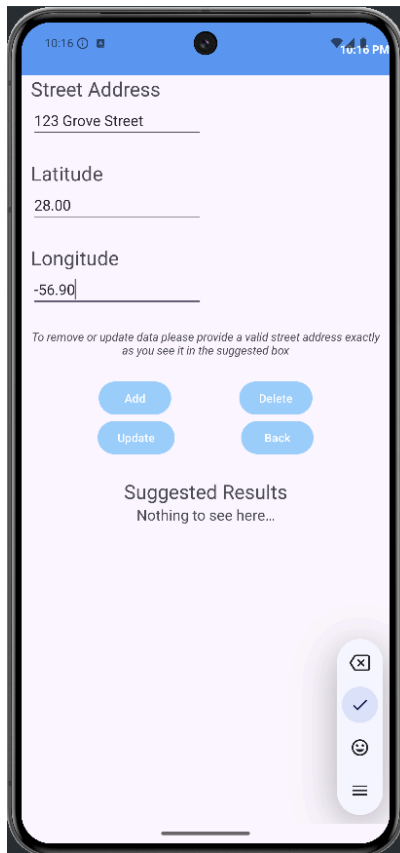
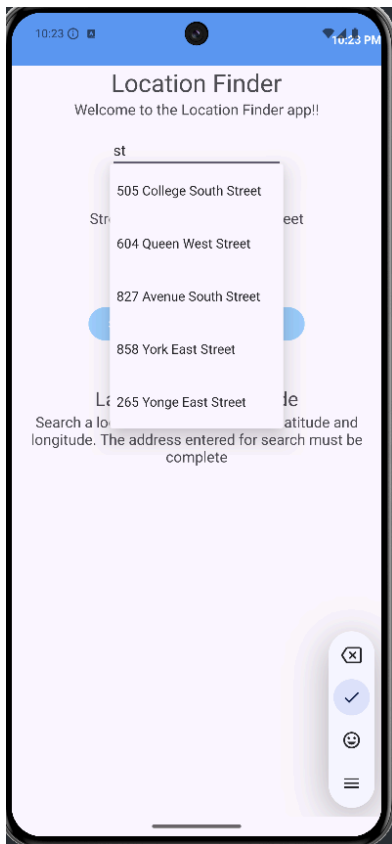
LocationFinder

Overview

In this assignment the task was to create an application that can find a location's latitude and longitude based on an entered street address. LocationFinder does just that as well as has the ability to add, update, and remove locations. This current version of the application will not allow duplicate addresses. That being said an address entered as "123 Grove Street" and "Grove Street 123" would *not* be considered a duplicate as the order in which the address is entered *matters* when adding, updating or removing an entry, however when searching the order doesn't matter as a suggested box will appear. The application does allow short forms of street names for example: 123 Grove Street = 123 Grove St, when queried will pull up the same entry (please note that if you tried to add 123 Grove St and an address 123 Grove Street was already in the database the operation would produce an error toast message). Please note that when adding, updating or deleting an entry while the order matters, it is case insensitive as all entries added to the database are standardized before they get added as well as all queries are standardized. Finally the application consists of three pages, one page where locations can be searched, another page where they can be added and or deleted and finally a page where the locations are edited/updated.

DatabaseHelper.kt

This file is where all the database logic happens, it is where the database schema is created. In the case of LocationFinder, the DatabaseHelper file provides the functions used to manipulated the database functions like: populateDatabase(), addLocation(), removeLocation(), suggestedAddress(), getLocation(), and updateLocation() are all function used to manipulate the database or retrieve data from it. These functions are extremely helpful when trying to see what is actually in the database and retrieving information to display to the user on the fly. The database helps hold information for all elements on the app in one place making information easy to manage and keep track of. It stores all the information in an SQLite database that is persistent among multiple sessions (when the app closes the information remains, until the app is uninstalled of course). In the case of this app the database is populated the first time the app is opened via the populateDatabase() function. It pulls values from a **.csv** file and parses the data and adds it to a LocationData object so the data within the object can be added to the appropriate columns in the database, this is how LocationFinder was able to add 100+ entries.



These Images show the search, add and update feature of LocationFinder. The first image depicts the suggested search feature. While the next four show an address of 123 Grove Street being added to the database, then modifying its latitude and longitude and then displaying the search results for "123 Grove Street".