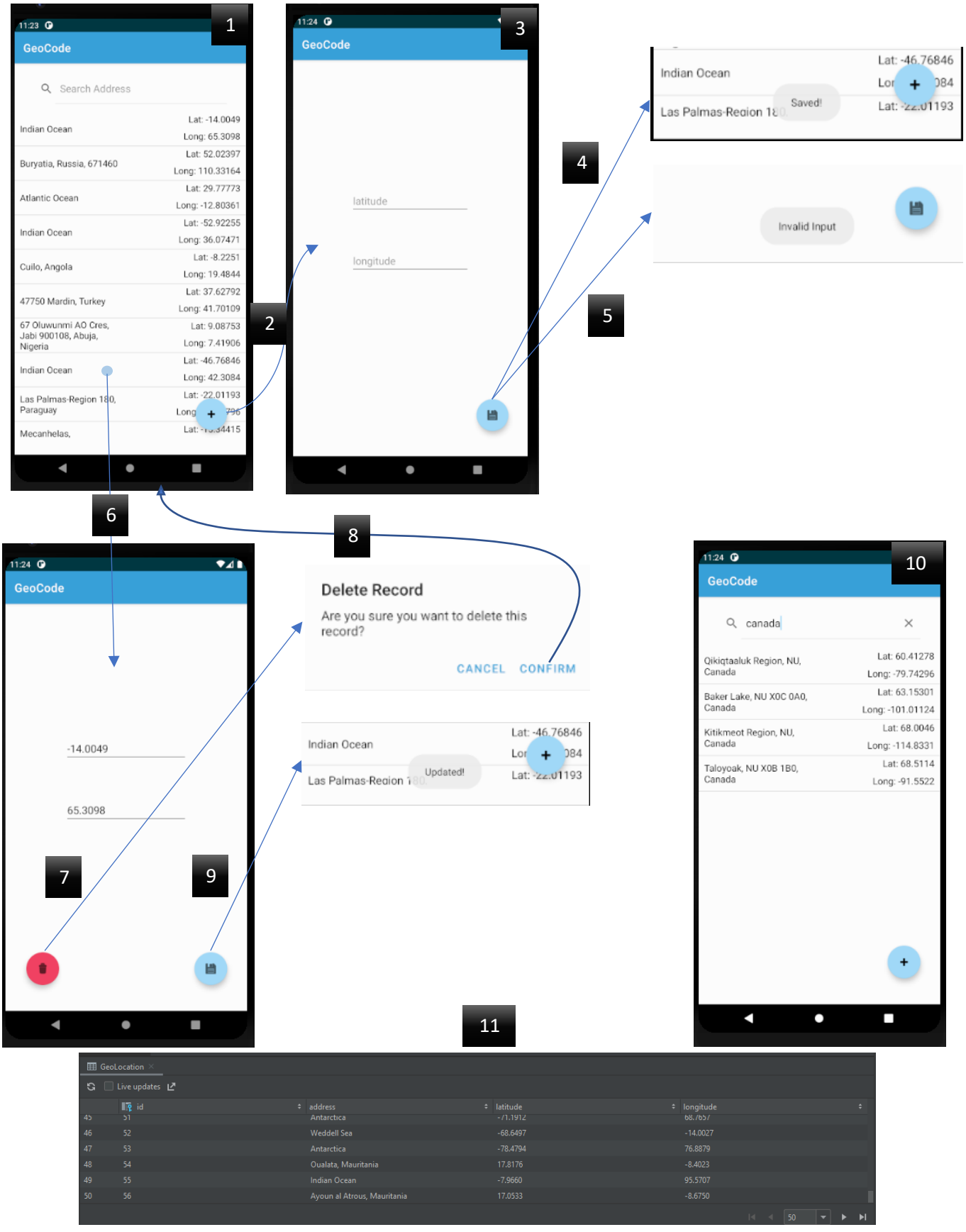
Yusuf Shaik
100 655 451



Yusuf Shaik
100 655 451

GITHUB LINK This application uses sqlite database to store information within the device. I created a database that contains one table. This table (figure 11) contains rowid, address, latitude, and longitude. For location services, I used the GeoCoder class to determine the address based on the user input of latitude and longitude.

1. This view contains the list view of all items that have been entered into the database. In this view, users can scroll to view entries, search, and click the add button in the bottom right corner to add a new record to the database.
2. When a user clicks the "add" button, they will be navigated to a the "add record" activity.
3. In this view, users will be prompted to enter a latitude and longitude.
4. If both inputs are valid, the geocoder library will be used to determine the address of the given coordinates. These values (address, latitude, longitude) will be entered into the database with an auto incrementing id. The user will be notified with a toast message that their information was saved successfully, and they will be navigated to the main activity page (1).
5. If the input is empty, or the inputted information is not a coordinate, the user's input will be rejected.
6. Within the list view, if a user clicks on an element in the list, they will be navigated to an update record view. In this view, users will have the ability to update the latitude and longitude for the record that they clicked on. The coordinates were passed to the next activity with an intent, and displayed in the given edit text fields so the user can actually edit the current information.
7. If the user clicks on the delete button, a confirmation will be requested. If they cancel the delete, nothing will happen.
8. If the user confirms the delete, the record that was clicked on would be deleted, and the user will be navigated to the main activity where they can browse available records.
9. If the user instead clicks the save button, the new address will be determined based on their coordinate input using the geocoder class. The database will then replace the old information with this new information. The user will then be navigated back to the main activity where they will be able to scroll through all active entries.
10. The user can search for all entries using any keyword. This is implemented in the database class using the "LIKE '%<input>%'". Therefore, any part of the address can be inputted, and the information will be displayed as long as there is an address that contains that keyword within itself. Furthermore, the search functionality is applied dynamically as the user types, so they do not need to click enter in order to view results.
11. This image contains the database schema. As we can see, per the requirements there are 50 entries within the database. I used the add record view within my app to input coordinates. The application then determined the address and added all the information to the database. As required, I have the correct schema (rowid, address, latitude, longitude)