



SOFE 4640U

Mobile Application Development

Assignment 2

CRN 44434

Samuel Ijose – 100819367

Objective:

The purpose of this assignment is to practice Android application development with databases by building a simple LocationFinder app. The app includes a database that stores 100 predefined locations in the Greater Toronto Area (GTA), featuring functionalities to query, add, delete, and update locations within the database.

Overview of the LocationFinder App

The app is designed to allow users to manage a list of locations, each represented by an address, latitude, and longitude. Key functionalities include querying an address to retrieve its geographical coordinates and performing CRUD (Create, Read, Update, Delete) operations on the stored location data.

Database Structure

My app makes use of an SQLite local database to store location data. The database includes a single table, location, with the following columns:

1. **id**: INTEGER, Primary Key, Auto-incremented
2. **address**: TEXT, Stores the location address
3. **latitude**: REAL, Stores the latitude of the location
4. **longitude**: REAL, Stores the longitude of the location

I added a total of 100 unique locations into the table as outlined by the assignment requirements, mostly covering key areas within the GTA like Oshawa, Ajax, Pickering, Scarborough, Downtown Toronto, Mississauga, Brampton, and Markham).

```
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    5 usages
    private DatabaseHelper dbHelper;
    5 usages
    private EditText editTextAddress, editTextLatitude, editTextLongitude;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dbHelper = new DatabaseHelper(context, this);

        editTextAddress = findViewById(R.id.editTextAddress);
        editTextLatitude = findViewById(R.id.editTextLatitude);
        editTextLongitude = findViewById(R.id.editTextLongitude);

        Button buttonAdd = findViewById(R.id.buttonAdd);
        Button buttonDelete = findViewById(R.id.buttonDelete);
        Button buttonUpdate = findViewById(R.id.buttonUpdate);
        Button buttonQuery = findViewById(R.id.buttonQuery);
    }
}
```

```
private void preloadLocations(SQLiteDatabase db) {
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Oshawa', 43.8971, -78.8658)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Ajax', 43.8509, -79.0204)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Pickering', 43.8351, -79.0890)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Scarborough', 43.7764, -79.2318)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Downtown Toronto', 43.65107, -79.347015)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Mississauga', 43.5890, -79.6441)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Brampton', 43.7315, -79.7624)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Markham', 43.8561, -79.3370)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Richmond Hill', 43.8828, -79.4403)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Vaughan', 43.8372, -79.5083)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Etobicoke', 43.6205, -79.5132)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('North York', 43.7615, -79.4111)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('East York', 43.7010, -79.3216)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('York', 43.6896, -79.4795)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Bloor-Yorkville', 43.6708, -79.3948)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Harbourfront', 43.6383, -79.3807)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Distillery District', 43.6503, -79.3595)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('St. Lawrence', 43.6488, -79.3727)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Yonge-Dundas Square', 43.6562, -79.3808)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Rogers Centre', 43.6414, -79.3894)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('High Park', 43.6465, -79.4637)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Liberty Village', 43.6372, -79.4225)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Leslieville', 43.6661, -79.3272)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Riverdale', 43.6763, -79.3485)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('The Beaches', 43.6727, -79.2957)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Don Mills', 43.7335, -79.3310)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Forest Hill', 43.6973, -79.4158)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Rosedale', 43.6796, -79.3755)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('The Annex', 43.6703, -79.4069)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Bayview Village', 43.7705, -79.3828)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Yorkdale', 43.7254, -79.4520)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('King West Village', 43.6447, -79.4024)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Chinatown', 43.6535, -79.3982)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Kensington Market', 43.6544, -79.4000)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Financial District', 43.6486, -79.3815)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Entertainment District', 43.6465, -79.3903)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Queen West', 43.6476, -79.4024)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Little Italy', 43.6540, -79.4156)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Little Portugal', 43.6474, -79.4307)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Greektown', 43.6786, -79.3518)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Roncesvalles', 43.6488, -79.4488)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('East Chinatown', 43.6637, -79.3458)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (address, latitude, longitude) VALUES ('Koreatown', 43.6647, -79.4182)");
}
```

Main Features

1. Query Feature:

- Users can input an address to retrieve its corresponding latitude and longitude from the database.
- This functionality is implemented by querying the SQLite database for the address and displaying the results if found.

```
public Cursor getLocationByAddress(String address) {
    SQLiteDatabase db = this.getReadableDatabase();
    return db.query(TABLE_NAME, columns: null, selection: COL_ADDRESS + "=?", new String[]{address},
        groupBy: null, having: null, orderBy: null);
}
```

2.

Add Location:

- Users can add new locations by entering an address, latitude, and longitude.
- The app validates inputs and inserts the new location into the database.

```

public boolean addLocation(String address, double latitude, double longitude) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(COL_ADDRESS, address);
    values.put(COL_LATITUDE, latitude);
    values.put(COL_LONGITUDE, longitude);
    long result = db.insert(TABLE_NAME, nullColumnHack: null, values);
    return result != -1;
}

```

3. Delete Location:

- Users can delete an existing location by providing its address.
- The app finds and removes the record from the database if it exists.

```

1 usage
public boolean deleteLocation(String address) {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(TABLE_NAME, whereClause: COL_ADDRESS + "=?", new String[]{address}) > 0;
}

```

4. Update Location:

- Users can modify the latitude and longitude of an existing location.
- The app locates the specified address and updates its coordinates in the database.

```

1 usage
public boolean updateLocation(String address, double latitude, double longitude) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(COL_LATITUDE, latitude);
    values.put(COL_LONGITUDE, longitude);
    return db.update(TABLE_NAME, values, whereClause: COL_ADDRESS + "=?", new String[]{address}) > 0;
}
}

```

User Interface Design

To enhance usability, the app's interface was designed with simplicity and readability in mind. Key UI components include:

- **Text Fields:** Used for inputting the address, latitude, and longitude.
- **Buttons:** Clearly labeled buttons for each action (Add, Delete, Update, Query).
- **Color Scheme:** Different colors for each action button for easy identification (e.g., green for add, red for delete).

Location Finder

Oshawa

43.8971

-78.8658

Add Location

Delete Location

Update Location

Query Location



Oshawa



q¹

w²

e³

r⁴

t⁵

y⁶

u⁷

i⁸

o⁹

p⁰



a

s

d

f

g

h

j

k

l



z

x

c

v

b

n

m

,

.



?123



?123



GitHub Repository: <https://github.com/OluwaJomilojuljose/LocationFinder>