Assignment 2: LocationPinned App

SOFE 4640U

Professor: Dr. Akramul Azim

Lab Instructor: Austin Page

Ontario Tech UNIVERSITY

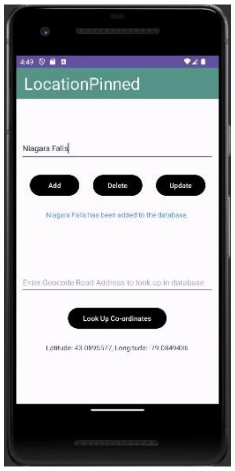
https://github.com/milansamuel10/LocationPinned

1. Introduction

The LocationPinned app utilizes Geocoding to convert a user input address to latitude and longitude coordinates. This process is aided using a SQLiteDatabase and InputStreams, as a text document with 50 predefined coordinates are read by the InputStream and are used to populate the database in its initial state.

2. Main Activity Screen UI Functionalities

The Main Activity Screen offers the user the ability to input an address in an EditText and query the coordinates via a button labeled "Look Up Co-ordinates" if the address already exists in the database. If the geocoded street address is not in the common location names database, a TextView displays the text "Road Address does not exist in the database" so the common location name and road address have to be an exact match. If they are a match, the co-ordinates pop on the screen e.g. the Lake Placid screenshot below. Furthermore, the screen allows the user to add and update location entries in the database, as well as delete entries in the database if they type in the common location name, not the geocoded street address. Please see all screenshots for full functionality.

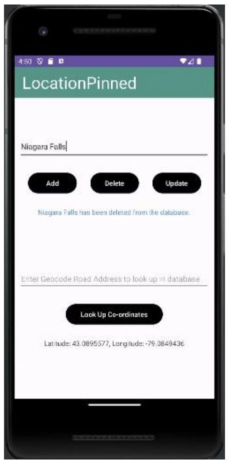


Niagara Falls

Niagara Falls has been added to the database.

Enter Geocode Road Address to look up in database

Latitude: 43.0895577, Longitude: -79.0849436

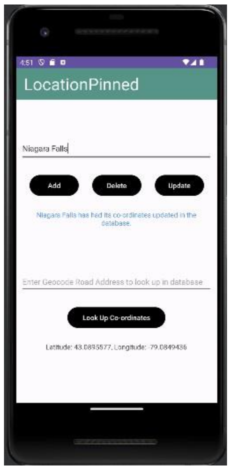


Niagara Falls

Niagara Falls has been deleted from the database.

Enter Geocode Road Address to look up in database

Latitude: 43.0895577, Longitude: -79.0849436

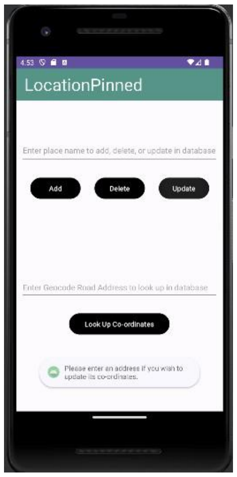


Niagara Falls

Niagara Falls has had its co-ordinates updated in the database.

Enter Geocode Road Address to look up in database

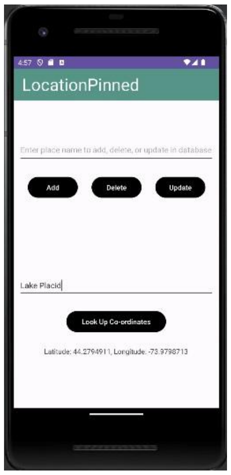
Latitude: 43.0895577, Longitude: -79.0849436



Enter place name to add, delete, or update in database

Enter Geocode Road Address to look up in database

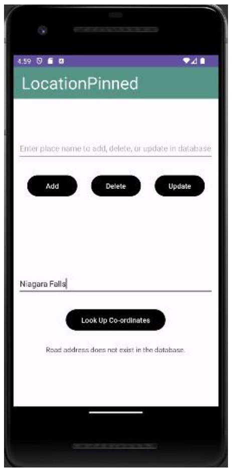
Please enter an address if you wish to update its co-ordinates.



Enter place name to add, delete, or update in database

Lake Placid

Latitude: 44.2794911, Longitude: -73.9798713



Enter place name to add, delete, or update in database

Niagara Falls

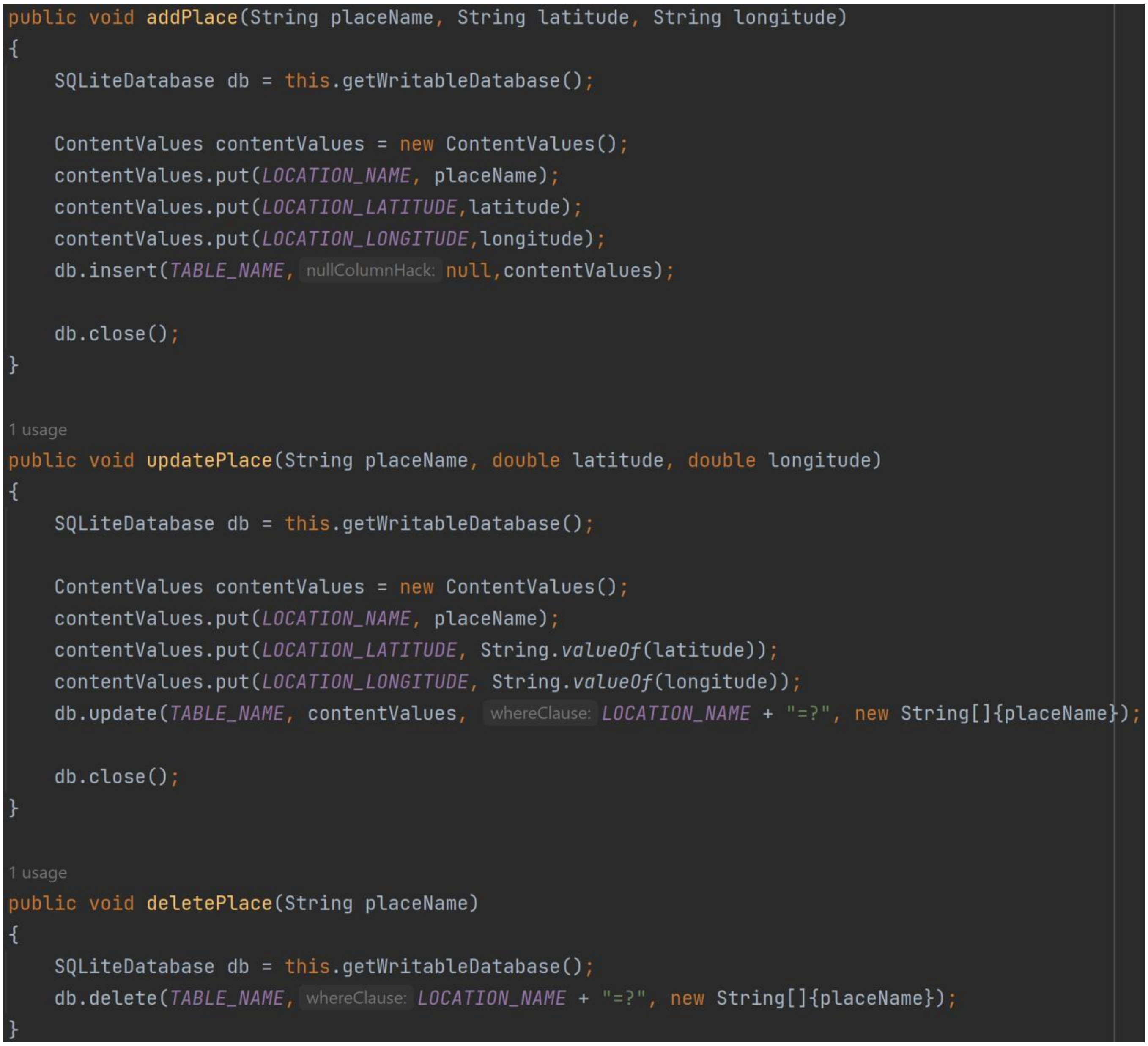
Road address does not exist in the database.

3. DBHelper Backend Functionalities

I established the database in the DBHelper.java class, and created a schema with the columns ID, the location's common name, and the corresponding latitude and longitude coordinates. Inside this class, the fillDatabaseFromInputFile method populates the initial database state with the 50 coordinates in the text file (input.txt which resides in the directory res/raw). This done with an InputStream and BufferedReader object to read the text file line by line.

4. Main Activity Screen Backend Functionalities

I modified the database to ensure that it reflects the changes made during CRUD operations on the locations database. The database now effectively stores and updates data, including add and remove functions based on the buttons that are available on the main screen. Below is a screenshot showcasing the database modifications.



```java
public void addPlace(String placeName, String latitude, String longitude)
{
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues contentValues = new ContentValues();
    contentValues.put(LOCATION_NAME, placeName);
    contentValues.put(LOCATION_LATITUDE,latitude);
    contentValues.put(LOCATION_LONGITUDE,longitude);
    db.insert(TABLE_NAME, nullColumnHack: null,contentValues);

    db.close();
}

1 usage
public void updatePlace(String placeName, double latitude, double longitude)
{
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues contentValues = new ContentValues();
    contentValues.put(LOCATION_NAME, placeName);
    contentValues.put(LOCATION_LATITUDE, String.valueOf(latitude));
    contentValues.put(LOCATION_LONGITUDE, String.valueOf(longitude));
    db.update(TABLE_NAME, contentValues, whereClause: LOCATION_NAME + "=?", new String[]{placeName});

    db.close();
}

1 usage
public void deletePlace(String placeName)
{
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_NAME, whereClause: LOCATION_NAME + "=?", new String[]{placeName});
}
```