package com.example.containmentzone\_alert.roomdatabase;

import android.content.Context;

import android.location.Location;

import android.util.Log;

import com.example.containmentzone\_alert.extras.LogTags;

import com.example.containmentzone\_alert.roomdatabase.VisitedLocations;

import com.example.containmentzone\_alert.roomdatabase.VisitedLocationsDao;

import com.example.containmentzone\_alert.roomdatabase.VisitedLocationsDatabase;

import java.util.ArrayList;

import java.util.List;

public abstract class LocalDatabaseContainer {

private static VisitedLocationsDatabase database;

private static VisitedLocationsDao visitedLocationsDao;

private static List<String> diagonalRangePoint =new ArrayList<>();

public static void addToLocalDB(Location location, String dateTime, Context context) {

calculateContainer(location.getLatitude(), location.getLongitude(), "Bangladesh");

database = VisitedLocationsDatabase.getDatabase(context);

visitedLocationsDao = database.visitedLocationsDao();

final List<VisitedLocations> visitedLocationList = new ArrayList<>();

for (String drp: diagonalRangePoint) {

// format = "lat1,lon1,lat2,lon2\_dateTime"

String conatainerDateTimeComposite = drp+"\_"+dateTime;

visitedLocationList.add(

new VisitedLocations(conatainerDateTimeComposite, 1)

);

}

Log.d(LogTags.LocalDBContainer\_TAG, "addToLocalDB: db entry list size = "+visitedLocationList.size()+"\n\n");

database.databaseWriteExecutor.execute(new Runnable() {

@Override

public void run() {

for(VisitedLocations entry: visitedLocationList){

try {

visitedLocationsDao.insertLocations(entry);

Log.d(LogTags.LocalDBContainer\_TAG, "run: room entry created");

}catch (Exception e){

visitedLocationsDao.update(entry.getConatainerDateTimeComposite());

Log.d(LogTags.LocalDBContainer\_TAG, "run: room entry updated");

}

}

}

});

}

public static List<String> calculateContainer(Double lat, Double lon, String country)

{

Double latDevider=0.000000d, lonDevider=0.000000d, latX, lony;

diagonalRangePoint =new ArrayList<>();

if(country.equals("Gujarat")){

latDevider=.0002000d;

lonDevider=.0002000d;

}

latX=Math.floor(lat/latDevider)\*latDevider;

lony=Math.floor(lon/lonDevider)\*lonDevider;

//upper left upper right

Double boxA\_X,boxA\_Y,boxC\_X,boxC\_Y;

boxA\_X=latX;

boxA\_Y=lony;

boxC\_X=latX+latDevider;

boxC\_Y=lony+lonDevider;

// # # lower

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxA\_X))+","+checkLatLongLength(Double.toString(boxA\_Y))+","+checkLatLongLength(Double.toString(boxC\_X))+","+checkLatLongLength(Double.toString(boxC\_Y)));

if(lat- boxA\_X<latDevider/4){

//left box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxA\_X-latDevider))+","+checkLatLongLength(Double.toString(boxA\_Y))+","+checkLatLongLength(Double.toString(boxA\_X))+","+checkLatLongLength(Double.toString(boxC\_Y)));

}

else if(boxC\_X-lat<latDevider/4){

//right box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxC\_X))+","+checkLatLongLength(Double.toString(boxA\_Y))+","+checkLatLongLength(Double.toString(boxC\_X+latDevider))+","+checkLatLongLength(Double.toString(boxC\_Y)));

}

if(lon- boxA\_Y<latDevider/4){

//lower box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxA\_X))+","+checkLatLongLength(Double.toString(boxA\_Y-lonDevider))+","+checkLatLongLength(Double.toString(boxC\_X))+","+checkLatLongLength(Double.toString(boxA\_Y)));

}

else if(boxC\_Y-lon<lonDevider/4){

//Upper box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxA\_X))+","+checkLatLongLength(Double.toString(boxC\_Y))+","+checkLatLongLength(Double.toString(boxC\_X))+","+checkLatLongLength(Double.toString(boxC\_Y+lonDevider)));

}

if(boxC\_X-lat <latDevider/4 && boxC\_Y-lon<lonDevider/4){

//Upper Right box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxC\_X))+","+checkLatLongLength(Double.toString(boxC\_Y))+","+checkLatLongLength(Double.toString(boxC\_X+latDevider))+","+checkLatLongLength(Double.toString(boxC\_Y+lonDevider)));

}

else if(lat- boxA\_X <latDevider/4 && lon- boxA\_Y<lonDevider/4){

//Lower left box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxA\_X-latDevider))+","+checkLatLongLength(Double.toString(boxA\_Y-lonDevider))+","+checkLatLongLength(Double.toString(boxA\_X))+","+checkLatLongLength(Double.toString(boxA\_Y)));

}

else if(lat- boxA\_X <latDevider/4 && boxC\_Y-lon<lonDevider/4){

//Upper Left box's diagonal points are to be inserted

diagonalRangePoint.add((Double.toString(boxA\_X-latDevider))+","+checkLatLongLength(Double.toString(boxC\_Y))+","+checkLatLongLength(Double.toString(boxA\_X))+","+checkLatLongLength(Double.toString(boxC\_Y+latDevider)));

}

else if(boxC\_X-lat <latDevider/4 && lon- boxA\_Y<lonDevider/4){

//Lower Right box's diagonal points are to be inserted

diagonalRangePoint.add(checkLatLongLength(Double.toString(boxC\_X))+","+checkLatLongLength(Double.toString(boxA\_Y-lonDevider))+","+checkLatLongLength(Double.toString(boxC\_X+latDevider))+","+checkLatLongLength(Double.toString(boxA\_Y)));

}

Log.d(LogTags.LocalDBContainer\_TAG, "calculateContainer: diagonalPoints size = "+diagonalRangePoint.size());

return diagonalRangePoint;

}

//This method keeps the lenght of the String same all the time

private static String checkLatLongLength(String latLonDigits){

int index;

int len=latLonDigits.length();

int decimalPointIndex=latLonDigits.indexOf('.');

int checkRequiredDigits=len-decimalPointIndex-1;

if(checkRequiredDigits<6){

for(index=checkRequiredDigits;index<6;index++)

latLonDigits=latLonDigits+"0";

}

else if(checkRequiredDigits>6){

return latLonDigits.substring(0, len -checkRequiredDigits+6 );

}

return latLonDigits;

}

}