package com.example.ArchievedLocationDB.roomdatabase;

import android.content.Context;

import androidx.room.Database;

import androidx.room.Room;

import androidx.room.RoomDatabase;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

@Database(entities = {ArchievedLocationDB.class}, version = 1, exportSchema = false)

public abstract class ArchievedLocationDB extends RoomDatabase {

public abstract ArchievedLocationDay archievedlocationday();

private static volatile ArchievedLocationDB INSTANCE;

// room queries need to be done on separate threads

private static final int NUMBER\_OF\_THREADS = 3;

public static final ExecutorService databaseWriteExecutor =

Executors.newFixedThreadPool(NUMBER\_OF\_THREADS);

public static ArchievedLocationDB getDatabase(final Context context) {

/\*

thread safe singleton design to get database INSTANCE

\*/

if (INSTANCE == null) {

synchronized (ArchievedLocationDB.class) {

if (INSTANCE == null) {

INSTANCE = Room.databaseBuilder(context.getApplicationContext(),

ArchievedLocationDB.class, "word\_database")

.build();

}

}

}

return INSTANCE;

}

}