**Experiment No – 2**

**Aim :** Develop an application that makes use of database.

**CODE:**

import android.content.ContentValues; import android.content.Context; import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase; import android.database.sqlite.SQLiteOpenHelper; import java.util.concurrent.atomic.AtomicInteger;

public class Database extends SQLiteOpenHelper {

private final static String DB\_NAME = "steps"; private final static int DB\_VERSION = 1; private Database(final Context context) {

super(context, DB\_NAME, null, DB\_VERSION);

}

@Override

public void onCreate(final SQLiteDatabase db) {

db.execSQL("CREATE TABLE " + DB\_NAME + " (date INTEGER, steps INTEGER)");

}

@Override

public void onUpgrade(final SQLiteDatabase db, int oldVersion, int newVersion)

{

}

public Cursor query(final String[] columns, final String selection,

final String[] selectionArgs, final String groupBy, final

String having,

final String orderBy, final String limit) {

return getReadableDatabase()

.query(DB\_NAME, columns, selection, selectionArgs, groupBy, having, orderBy, limit);

}

/\*\*

* Inserts a new entry in the database, if there is no entry for the given
* date yet. Steps should be the current number of steps and it's negative
* value will be used as offset for the new date.

\*/

public void insertNewDay(long date, int steps) { getWritableDatabase().beginTransaction();

try {

Cursor c = getReadableDatabase().query(DB\_NAME, new String[]{"date"}, "date = ?", new String[]{String.valueOf(date)}, null, null, null);

if (c.getCount() == 0 && steps >= 0) {

// add today

ContentValues values = new ContentValues();

values.put("date", date);

// use the negative steps as offset

values.put("steps", -steps);

getWritableDatabase().insert(DB\_NAME, null, values);

}

c.close();

getWritableDatabase().setTransactionSuccessful();

}

finally {

getWritableDatabase().endTransaction();

}

}

public int getSteps(final long date) {

Cursor c = getReadableDatabase().query(DB\_NAME, new String[]{"steps"}, "date = ?",

new String[]{String.valueOf(date)}, null, null, null); c.moveToFirst();

int re;

if (c.getCount() == 0) re = Integer.MIN\_VALUE; else re = c.getInt(0);

c.close(); return re;

}

?",

public int getSteps(final long start, final long end) { Cursor c = getReadableDatabase()

.query(DB\_NAME, new String[]{"SUM(steps)"}, "date >= ? AND date <=

new String[]{String.valueOf(start), String.valueOf(end)},

null, null, null);

int re;

if (c.getCount() == 0) { re = 0;

} else {

c.moveToFirst(); re = c.getInt(0);

}

c.close();

return re + 7257;

}

public void saveCurrentSteps(int steps) { ContentValues values = new ContentValues(); values.put("steps", steps);

if (getWritableDatabase().update(DB\_NAME, values, "date = -1", null) == 0)

{

values.put("date", -1); getWritableDatabase().insert(DB\_NAME, null, values);

}

}

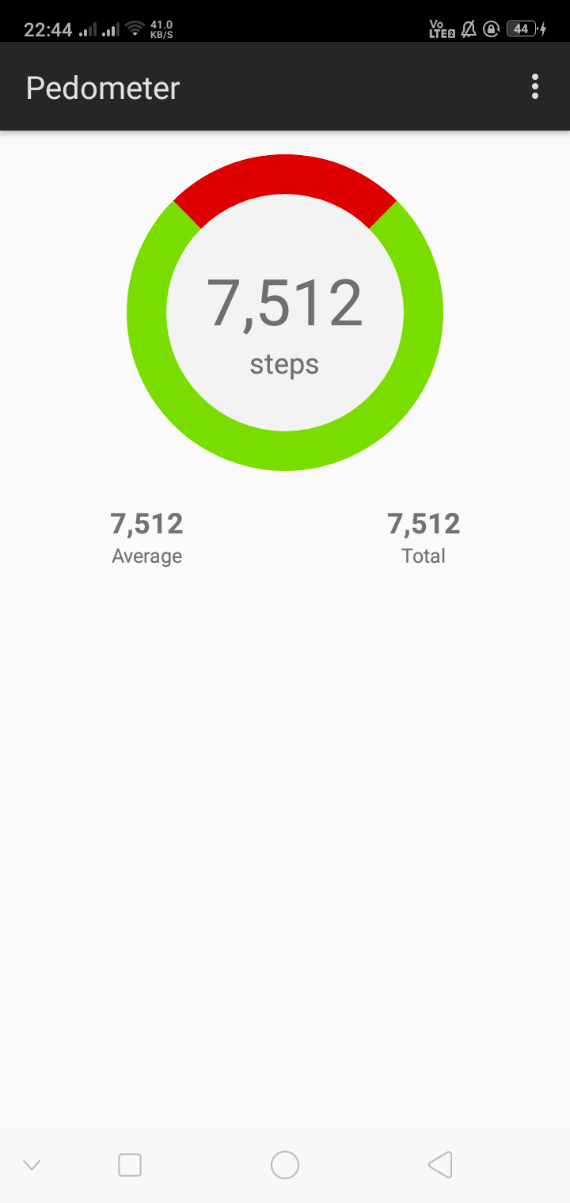
public int getCurrentSteps() { int re = getSteps(-1);

return re == Integer.MIN\_VALUE ? 0 : re;

}

}

**OUTPUT :**



**CONCLUSION :**

In this experiment we learnt how to access database in android and what are the classes and methods required for it, we also verified the concept by making use of the database in our problem statement to track the count of steps walked by the user successfully.