Mini Project 4: My Map

Due: May 5, 23:59 PDT

Overview

Build an app (Fig. 1) that 1) displays a Google map and allow users to switch between different views; 2) contains a floating action button (FAB) that will display the current location when the user clicks it. 3) has an SQLite database and a content provider so that:

- On clicking the Map, a marker will be drawn at the taped location, and the corresponding latitude longitude coordinates with map zoom level will be saved in the database in background.
- On restarting the app/changing orientation, the saved locations are retrieved from the database in background and redrawn on the Map.
- On long-clicking, all makers will be cleared from the map, and all data will be removed from the database in background.

When creating the project, select "Google Map Activity". Name the project as "Project4YourName". Feel free to use the code in exercise 6 & 7.

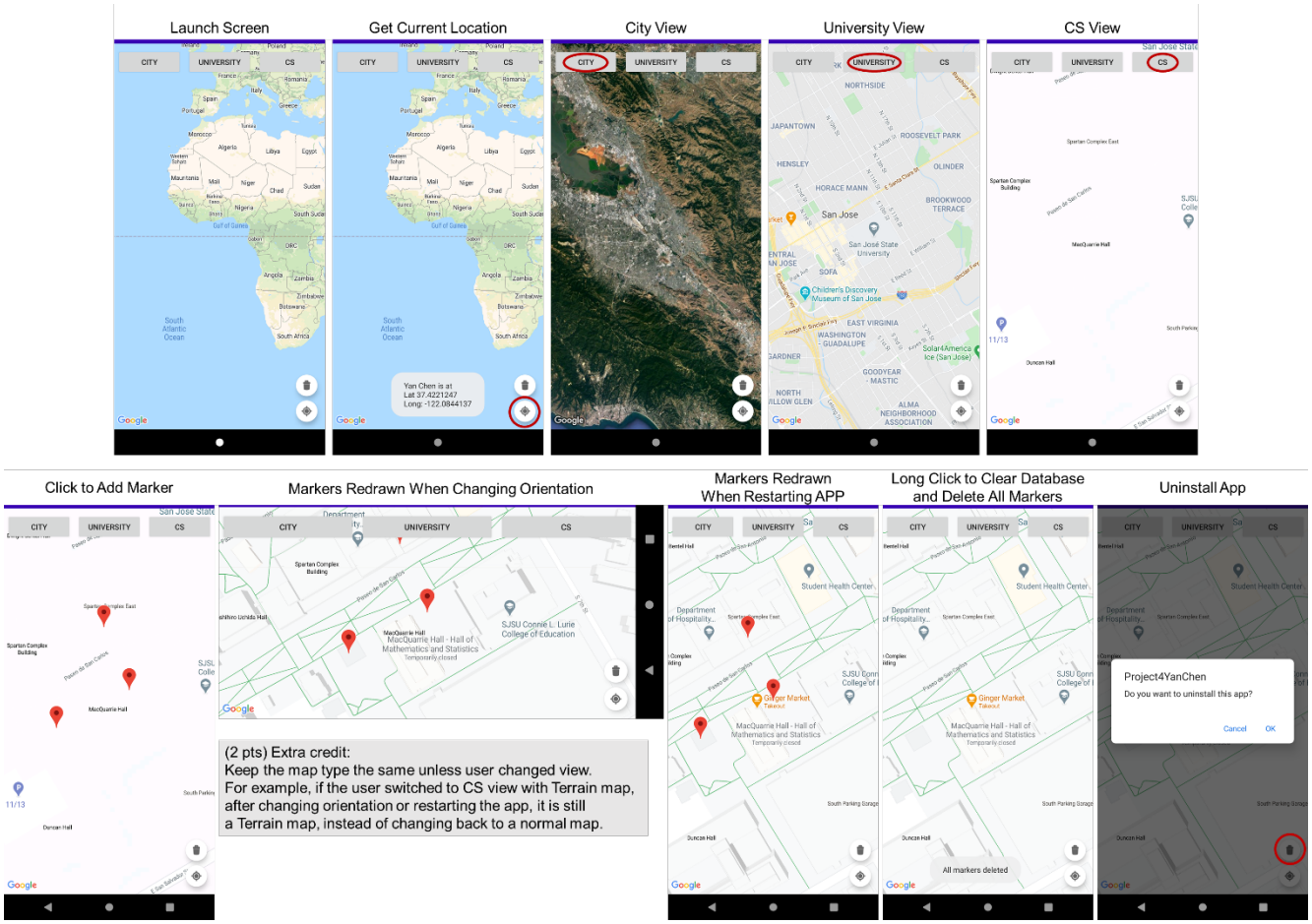

Fig. 1 Sample run of the app

Setup

Before you start coding, you need to setup following:

- Set Google map API. After creating the project with a Google map activity, you will be directed to google_maps_api.xml file. Follow the steps in the comment to add a Google API key.
- Request permissions (ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION, INTERNET, and REQUEST_DELETE_PACKAGES) in the manifest.
- If you want to use Google Location Service API for getting the location, add the following dependency in the module level build.gradle, You can use other ways to get location.

```
implementation 'com.google.android.gms:play-services-location:18.0.0'
```

Map Activity

There will only be one activity in this app.

User Interface (UI)

The positions can be different from Fig. 1, but you should include the following widgets and the UI should be reasonable. Most of the functionalities are the same as the app in Exercise 7.

Widget	Content	Usage	Details
Button	"City"	Switch to City View	Focus on the area of San Jose City with Satellite map
Button	"University"	Switch to University View	Focus on the area of SJSU with basic map
Button	"CS"	Switch to CS View	Focus on the area of CS department with Terrain map
FAB	Uninstall (delete) Icon	Uninstall App	Uninstall the app (send an implicit Intent)
FAB	Location Icon	Get Current Location	Toast a message with your name and the current location (lat. and long.)
Fragment	Google Map	Display Map	Display the map and change view based on which button being clicked

Interaction with Database

The insertion and deletion operations should be executed as AsyncTask and the data retrieval is done using CursorLoader through LoaderManager.LoaderCallbacks<Cursor> interface. Note that AsyncTask was deprecated from API 30, and the official documentation asks you to use java.util.concurrent instead. Try use that if you are familiar with threads and concurrency. Otherwise, just use AsyncTask.

For accessing the database, make sure the URI matches the URI authority of content provider.

Database & Content Provider

Similar to exercise 6, It may be easier to have a class for the SQLite database and a class for the content provider. You don't need to put them to another app though.

Database

The SQLite database has one table with 4 columns: _id (int, auto increment), latitude (double), longitude (double), and zoom level (float). Note that when naming the columns, don't put any space. And you can make the Strings for column names as protected, so you can use them in MapsActivity.

In the database class, the required constructor, onCreate, and onUpgrade can all be the same as StudentsDB in exercise 6. It should also have custom methods to insert data, delete all data, and get all data. Those methods are very similar to the custom methods in StudentsDB in exercise 6, except that when getting all data, you don't need to specify the order.

Content Provider

Create the class for content provider by right-clicking the source code package, choose New -> Other -> Content Provider. When setting the URI Authorities, make sure it's the same as the package name. In this way, a content provider will be automatically registered in manifest.

The structure of the content provider class is very similar to StudentsProvider in exercise 6. You don't need a constructor. You should have a database object and initialize it in onCreate. Implement insert, delete, and query by calling the corresponding custom method in the database class.

Additional Notes

- It may be easier to implement the database and the content provider first.
- If you need help, you can request a template (starter code) via email (include the email address you used for Bitbucket). But you will lose 2 points.
- And here are some potential errors/problems you may encounter and how you may solve them. Some details were given in exercise 7.
    - Map not showing. It's possible that the Google API key is not set correctly. Check if the certificate and package name of the app match the information registered in the key.
    - App crashes when accessing the database. Other than checking if the URI is correct, check if the string you used for creating the table is correct. It should be "create table table_name(column1 type1, column2 type2, column3 type3, column4 type4)", where column1 is the name of the first column and type1 is the data type of the first column, etc. You may need to uninstall and reinstall the app after fixing database/content provider related of problems.
    - Can't get current location. Try to set the location in the emulator, or/and turn off the location and turn it back on in the device's setting. Also, you should use a device that supports Google Play.
    - Markers not being redrawn when restart the app. Check database inspector to see if data are added successfully.

Functionality Requirements

Your project will be graded based on if your app meets all requirements.

Requirement	Points
The app can successfully run with the map drawn on the screen	1
Can switch to the required city view	0.5
Can switch to the required university view	0.5
Can switch to the required CS view	0.5
FAB for getting the current location works	1
App will not crash even if the location service is not enabled	1
App will not crash even if the permission to get current location is not given	1
FAB for uninstalling the app works	0.5
On click, a marker is drawn at the clicked location on the map	0.5
On click, required data are inserted to database in background (AsyncTask)	1
On long-click, all markers on the map are cleared	0.5
On long-click, all data are deleted to database in background (AsyncTask)	1
When the orientation changed, the marker added will not disappear	1
On restarting the app, the data are retrieved in background (CursorLoader)	1
On restarting the app, the saved makers locations are redrawn on the Map	1
Did not request for a template	2
All other requirements meet (package name, your name in message, etc.)	1
Total	**15**

Submission

- Push your project to the Bitbucket repository (Project4YourName) the due date.
- Invite and share your Bitbucket repository the grader (edmond.lin@sjsu.edu) and the instructor (yan.chen01@sjsu.edu).
- Submit "Mini Project 4 - My Map" assignment on Canvas using the template provided (see description of the assignment) and submit your APK (or you will get a zero!).
- Only your last submission before deadline will be graded based on the criteria in Functionality Requirement.
- (Optional) Post a discussion on Canvas to share any suggestions/tricks/hints, etc. for finishing this project.