CSCI 4237 Project Writeup

ParkDC

Dec. 10, 2012

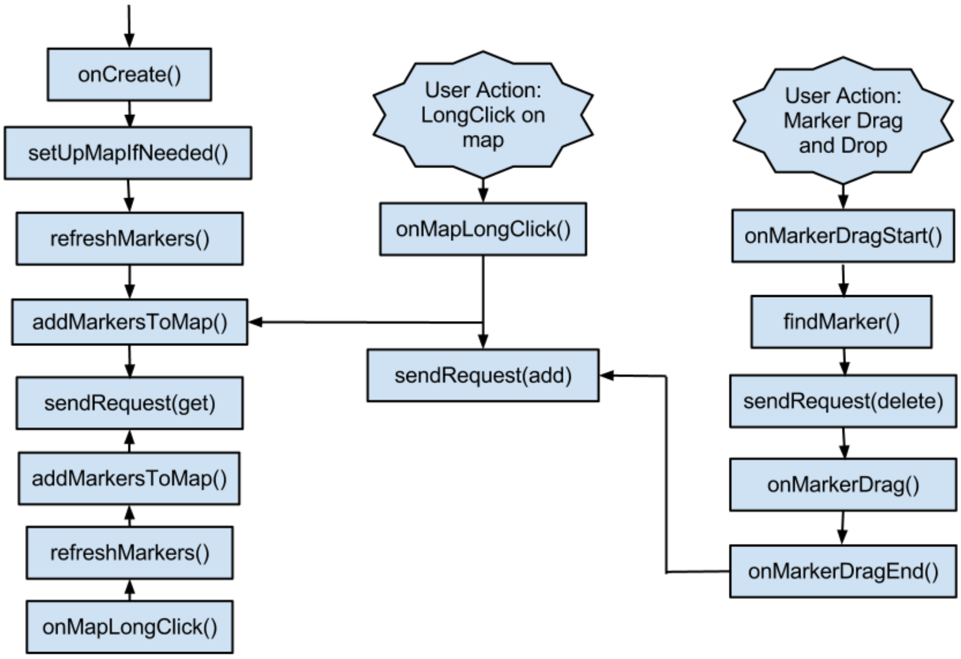
Tianyi Song

Introduction

This app helps people find a street parking place. Users could just load the app and check where they could park and take off to the destination. This could save some time and trouble of the exhausting street parking. The main elements in this app are google map api and HttpRequest. And demo video is available on youTube (http://youtu.be/jXRXKmvWG2M).

Implementation

There are only two activities in this project, WelcomeActivity and ParkingMapView. WelcomeActivity handles user signing in and the ParkingMapView activity is the major activity of this project. Following is the workflow of the program.



As soon as the user signs in the app, the program calles the onCreate function to create a instance of ParkingMapView activity. This function sets the layout of the view and setup the Map view for the app.

When setting up the Map view, the program adds some Listeners, such as onMapLongClickListener, onMarkerDragStart, onMarkerDrag, onMarkerDragEnd, the first one is for responding to users' long click on the map, the last three listeners are called when the user performs a drag-and drop on a marker.

After the Map view is created, the program sends a HttpRequest to the web server to query all the marker data in database. Server returns the output to the program as a JSON string. This part is implemented using python. In order to "understand" the JSON string, program parses the string into some values of properties of the markers and create new markers using those values and add the markers to the map.

When a user performs long click on the map view, which will trigger the onMapLongClickListener to react, he is asked to choose the status of the new marker. Accoring to the user's choice, I set the marker's color using HUE. Meanwhile, a HttpRequest is sent to the server that a new marker is added to the map and insertion of this marker is needed.

The program also "listens" to users' drag and drop operation. The program first requests to delete the old data and add a new data item to the database during this progress.

Details of Major Methods

Since sendRequest is the called most of the time, I will start with this method. This method takes 5 parameters, they are *method, id, type, lat, lng*. Parameter *method* denotes the types of operation the program wants the database to take; *id* is the unique identification of the marker which is generated using java.util.UUID; *type* is the status of the stree, such that 0 (full), 1(Only one available), 2(2+ available); *lat* and lng are the geo location of the marker. Different methods have different inputs, when the method field is "get" then all the other parameters can be set as "". When method is "del" (short for delete), then rest of the parameters except *id* to be "". And the program forms a new URL and send it to server and request.

The request procedure takes place in another class MyHttpRequest that extends AsyncTask<URL, Integer, String>. It implements the inherited function doInBackGround to issue the HttpRequest and then get the response from server.

After the request is sent, the server will parse this request into valid a query statement and issue a query. The it wraps query result with a JSON outfit and responds to the program.

Method addMarkerToMap is the only method that could add a marker to the map. I utilize the status parameter to set the color of the marker using HUE.

Method onMapLongClickListener offers a way for the user to add a marker to the map. There is a alertDialog used in this method allowing the user to choose the desired status of the street.

Problems and Solutions

(1) Problem: When I tried instantiate a AlertDialog object, the program threw a NullPointerException error and stopped.

Solution: The way of getting context to create the AlertDialog.Builder is wrong. The correct way is to use MyClass.this as the context.
(2) Problem: I have had trouble with all kinds of listeners. There was a time when the program stopped suddenly as long as I added a few lines of listener.
Solution: There is not actually a solution to this. One just needs to be careful and understand how the listerens works.
(3)Problem: When I tried to issue a http request in the ParkingMapView.java, the program just stopped working.
Solution: Turns out we can issue a http request in the main thread. This request has to be done through a class that extends the AsyncTask<URL,Integer,String> interface.

Problems

I still didn't manage the program to get the current location of the user.