CSC 714: Project Report

Karthikeyan Sivaraj
ksivara@ncsu.edu

Mansoor Aftab
maftab@ncsu.edu

Service Time Overlay for Google Maps
http://www4.ncsu.edu/~maftab/android

Introduction

The project undertaken develops an application for phones running the android platform to indicate estimated service times for various locations the user intends to visit in a Google maps overlay. The user can plan his schedule better using this information. Our project consists of tasks to periodically sense the GPS location and send messages to a centralized server, which then processes this data based on predefined rules and stores the average service time info for a particular location in a database. The user can then access this information using the maps overlay.

Tasks

In our proposal, we had split the project into six major tasks. They were

- Background reading and learning the framework and APIs required
- Coding the GPS sensing service
- Implementing the centralized server part
- Defining a Messaging Protocol between the server and the clients
- Creating the Google maps overlay
- Implementation of UI interface for the application

All activities have been completed successfully.

Application Usage

In order to use our service the user will need to access the UI Interface to start/stop the service, upon start of the service, the application will display the map of the current location. The user may wish to explore the current location or in some cases a location which he is planning to visit, for such cases the application provides zoom-in zoom-out controls and the user can navigate to the location in the map where he wishes to go, once the point of interest is visible on screen the user just needs to tap the screen on the location shown and the phone will display service time records for that location if available, by means of a pop-up message.

Design

The design consists of a GPS sensing application on the android platform which will monitor the GPS location of the phone and transmit the GPS co-ordinates to a centralized server, to achieve this the

application consists of a background service which is dormant most of the time and becomes active whenever the user moves about resulting in a change of position, when this change of position occurs the application on the phone evaluates the time spent at the previous location by comparing current time with time recorded during last execution, this time spent signifies the service time and it is reported to the server, the server processes the data and stores extracted information on a database. This database keeps historical averages of time spent at that location. Once a service area has sufficient amount of historical record, the server will send out a reports to queries the users send about that location, informing about the time the customer can expect to wait there. This information is displayed in conjunction with a map interface

Implementation

- GPS Sensing Service: The GPS location sensing code will run as a service in the background, it is mainly brought into play when ever there is a change of position, this change of position and distance at which a difference is reported is configurable, and the application uses the *Service Class*, to implement the service and the *LocationManager Classes* to sense the GPS location and the get notified about a change in position.
- UI Interface: The UI interface provides the user with the option to start/stop the service this is implemented using the *Activity Class*.
- Map View: The maps displaying the location is a direct instance of the *MapOverlay Class,* it also utilizes a *GeoPoint* object to determine the location the user touched, and uses it to extract the co-ordinates to be queried.
- Server: The server which listens to the report and query packets and responds with a result packets is designed as a simple UDP server, it listens for datagrams on a predefined port, this server, follows a simple structure of flag based code flow, where actions are defined similar to switch case structure for each kind of packet it receives and the action to be taken. Also for each query the server defines an area of consideration, this area is a square of 10x10 meters which is considered when calculating service times, this choice of area is due to the fact that most constructions are 4 sided in nature and a 10 meter square is a typical size for most establishments. Also recorded service times within this area of operation are considered when calculating the average service time.
- Messaging Protocol: When the location service is notified of a location change by the Location Manager class, it sends a message to the centralized server. We have decided to use simple datagram messaging (UDP) between the Phone, and the centralized server since updates regarding the position are just couple of bytes, and any other method such as HTTP Post would involve setting up of a TCP connection which will is a time consuming process for just transmitting 5-10 bytes of information. The message formats, which are as follows

Firstly, all messaging between the client and server follows a single uniform messaging packet shown below

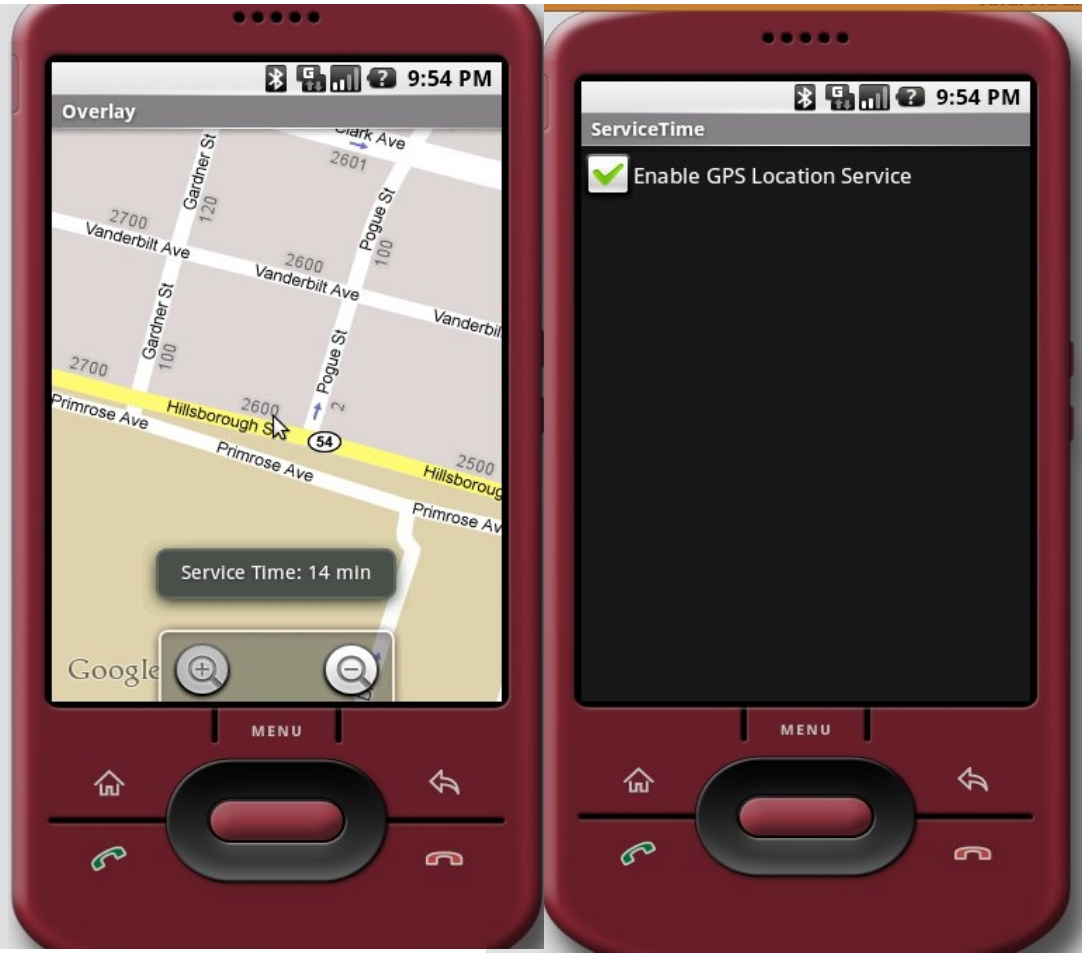
Longitude	Latitude	Time spent	Query Flag	Report Flag	Result Flag

The message for different cases are as follows

- **Report**: When the client wishes to report its last service location with time spent it places its longitude and latitude co-ordinates and the time spent, the report flag is set to 1 and the other to flags are set to zero.
- **Query:** When the user wants to know about a service location the message contains the co-ordinates, with the query flag set to 1, other values in the packets cease to have meaning in this case.
- **Result:** When the server responds back with the service time of a location queried using **"Query"** packets it responds with co-ordinates set along with historical record of time spend in the time spent field and the result flag set to 1.

Results

Below is a screen shot generated using the android emulator provided along with the SDK, we were able to get results on an actual android phone within a range of 20 meters with the area being restricted due to the availability of the WIFI connectivity, the area was partly indoor partly outdoor, with minimal radio interference, further results can be achieved using a 3G connection.



Test Cases

Unit testing for MapView	Tried to display location with Geo Points	Passed
Interaction test cases	Tested the zoom controls along with location selection	Passed
Communications testing	Tested with sample client program which generates various packets randomly and sends to server	Passed
Service testing	Tested the Wake-up events for change of GPS location	Passed
Functional testing	Tested the phone within a range of 20meters	Passed
Test for service time display	The service was able to display service times for a bank establishment with plugged-in values	Passed

Future Work

- Currently the user navigates using the touch sensor and zoom in/out buttons but in the future we would like to provide an interface to directly enter an address.
- We need to test how the communication between the client and server behaves in a 3G connection and see how much fault tolerant the UDP based communication exhibit.
- Another potential improvement is using multiple service times on a stretch of road to determine delays due to traffic.

References

Google App Engine
http://code.google.com/appengine

Android Developers Reference
http://developer.android.com/sdk/1.1_r1/index.html

Using Google Maps in Android
http://mobiforge.com/developing/story/using-google-maps-android

Google Maps API
http://code.google.com/apis/maps/