

CSC 714 Project Report

Karthikeyan Sivaraj
ksivara@ncsu.edu

Mansoor Aftab
maftab@ncsu.edu

Service Time Overlay for Google Maps

URL: <http://www4.ncsu.edu/~maftab/android/index.htm>

Introduction

We are currently developing an application 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.

In the period since we submitted the project proposal, we have completed various tasks and come up with a detailed design which we have described below.

Tasks

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

- background reading and learning the framework and APIs required
- coding the GPS sensing service
- Implementing the centralized server part and
- creating the Google maps overlay

Of these, we have completed the first two tasks – background reading and coding the GPS location sensing and transmitting part. We have tested the network packets sent with a sample program on our PC. We have also come up with an overall detailed design for the project.

Design/Implementation Details

The GPS location sensing code will run as a service in the background. The application will have an activity to start/stop this service. We also have another activity which displays the embedded MapView for overlay. When the location service is notified of a location change by the LocationManager 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 be a time consuming process for just transmitting 5-10 bytes of information. There are two types of message formats, which are as follows:

Location Update from the phone

Android ID	GPS Co-ordinate X	GPS Co-ordinate Y
------------	-------------------	-------------------

Location info from the server (for overlay)

Location Name	Average Service Time at location
---------------	----------------------------------

For the device ID, we plan to use the unique 64-bit Android ID obtained from `android.provider.Settings.System.ANDROID_ID`.

Server Rules

Every time the server receives a message from a mobile device it timestamps the message and runs a set of rules on the message. These rules are as below:

- Was the device at the same location previously ?
- If yes accrue the time spent at that location
- If not and the device has moved to a new location check the accrued time spent at the previous location if it the time spent is above a certain (configurable) threshold, locate the service record for that particular location, and average this new time spent against the historical records this can be done by taking a weighted average as follows.

$$\text{New Average} = \text{Historical Average}(\text{Alpha}) + \text{New observation}(1-\text{Alpha})$$

This parameter Alpha acts as a knob to control how much relevance the new readings are given compared to the recorded averages.

- Also if customer has entered new location check if the new location is within a point of interest, if yes fetch service record for this new location.

Server Records Storage

The server uses the Google App Engine Data base to store these data values and they are accessed using the Google Query Language(GQL). We maintain two tables – one to record the previous position of the device and one to store the average service time for a location. The record formats are as indicated below.

Android ID	Location	Time @ Location
------------	----------	-----------------

Location	Average @ Location
----------	--------------------

Individual Contributions

Karthikeyan Sivaraj - GPS Sensing Application, Background Reading, UI design

Mansoor Aftab - Maps API, Database API, Table structure, Background Reading

The system design was done by both of us together.

References

Google App Engine

<http://code.google.com/appengine>

Android Developers Reference

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

Google Maps API

<http://code.google.com/apis/maps/>

Google Base Data API

<http://code.google.com/apis/maps/>