CSC 714: Project Proposal

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 plan to develop a service time reporting application for the android platform which will report to the phone user, the estimated service time for various places the user intends to visit. The user can then plan his schedule better using this information. As an example, a person wants to pick up some breakfast at the drive through on the way before attending a meeting, but is not sure how long it is going to take. This application could be very useful in such a case. This application can also be extended as an overlay layer on Google maps. As an example, bank branch icons are shown on Google maps and every time the user hovers the mouse pointer on such an icon it will display the service time at that branch.

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, which processes the data and stores extracted information on a Google Apps database. Once the server senses that the person has stopped at a particular location for more than a certain threshold period of time, it is understood that (s)he has entered a service area. Once the person starts moving away from that GPS co-ordinate the server translates that as a service completion and records the time spent. This is entered into a database which 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 the users when they are about to enter that location, informing about the time the customer can expect to wait at a particular location.

As mentioned another access method of this application is on Google maps, where points of interest are paired with the historical data of average time spent by people at that location, using the GPS co-ordinates as a key.

Tasks

We are estimating the following major tasks in the implementation phase.

1. Getting familiar with development environment

During this phase, we plan to familiarize ourselves with the development methodologies and the various APIs we will be using, such as the Android application framework, Google Maps API and the Google Apps API.

2. Coding the GPS sensing application

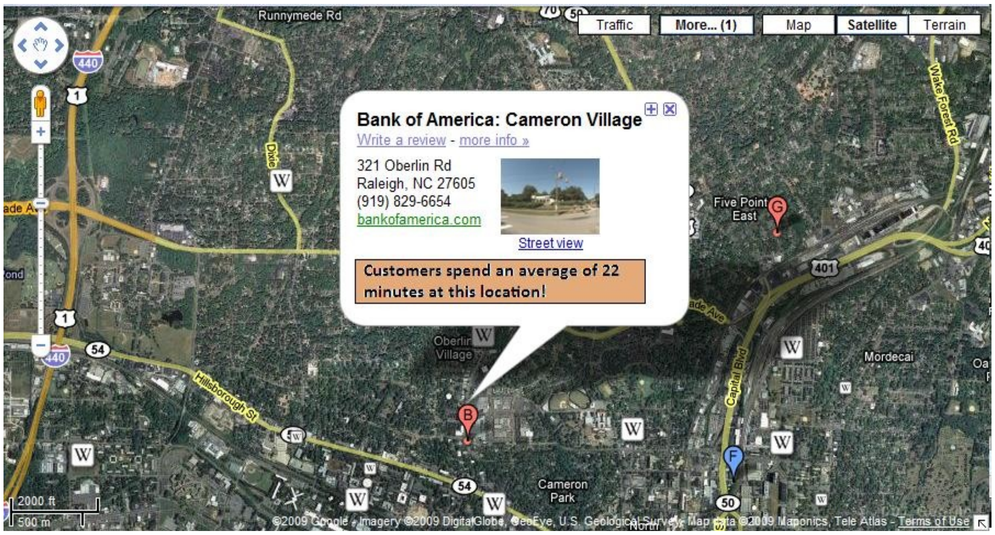
This phase consists of coding the GPS sensing application on the android platform. This application periodically sends out GPS location packets to the server. We will also design the protocol for communication between the user phone and the server.

3. Server side design and implementation

In this phase, we design and implement the server side application using Java. The server application receives the location packets from the users, and extracts information from the raw data. It then communicates with the Google Apps database and updates the average of the data for a particular location. The server can also implement various rules in order to define what range of coordinates comprises a location or when the user enters and leaves a location etc.

4. UI display on Android and Google maps overlay

We plan to create a custom Google maps overlay of service time linked with points of interest as shown in the picture. Also, we plan to implement a UI for Android to display the service times, based on user location.



Bank of America: Cameron Village

Write a review - more info »

321 Oberlin Rd
Raleigh, NC 27605
(919) 829-6654
bankofamerica.com

Street view

Customers spend an average of 22 minutes at this location!

Possible Extensions

This application can also be possibly extended in many ways. For example, it could be extended to include waiting time prediction for highways (ie taking traffic into account). Also, service times could be based on time of day.

References

1. Android developer reference (http://developer.android.com/guide/topics/fundamentals.html)
2. Goole Maps API (http://code.google.com/apis/maps/)
3. http://developer.android.com/reference/android/package-summary.html
4. http://code.google.com/apis/base/