

A Bus Tracking and Planning System for CTfastrak

Dr. Shuju Wu, Central Connecticut State University

Dr. Shuju Wu is currently an Associate Professor at the Computer Electronics and Graphics Technology Department at Central Connecticut State University. She holds a Ph.D. degree in Information Science from the University of Pittsburgh. Dr. Wu's teaching and research interests include computer communications and networks, multimedia systems, performance modeling and evaluation, and network applications. She is a member of IEEE and ASEE.

Mr. Shane Michael Carroll

Keith Boyd-Carter

James Krostoski

A Bus Tracking and Planning System for CTfastrak

Abstract

This paper presents the application design and development of a bus tracking and planning system that involves mobile application development, database and network communications. This system is specifically designed for a public transportation bus system similar to the CTfastrak. The developed system includes three modules, namely the bus-side application, server-side database and client-side mobile application. Network communication enables the interactions between the bus-side, server-side and client-side modules. The objective is two-fold: to provide the bus clients more convenience with bus schedules, real-time bus location and delay information so they can plan trips accordingly, and to provide useful information to the transportation center for management purpose. The developed system has been tested and demonstrated successfully.

Introduction

The presented work is the outcome of an undergraduate capstone project. Students in the Computer Engineering Technology (CET) major at Central Connecticut State University are required to finish a capstone project in their senior year study. The capstone project is accomplished by successful taking two consecutive courses, Capstone Project I and Capstone Project II. The Capstone Project I course is 1-credit in which students research, propose and finalize project ideas. The Capstone Project II course is 2-credit in which students finalize the design, implement and test the project ideas. This paper presents the application design and development of a bus tracking and planning system that involves mobile application development, database and network communications. This system specifically uses a public transportation bus system similar to the CTfastrak^[1] project in State Connecticut as a background and designs applications for it.

The university and the school where the CET program belongs to have been collaborating with the State Department of Transportation (DOT) in projects, teaching, student internship and co-op, as well as employment. The CTfastrak project from the State DOT aims to provide a new bus transit system interconnecting over 60 routes providing travellers convenient access to work, shopping, universities, downtown and other entertainment destinations. Systems on the scale of the CTfastrak don't have the demand as much as big-city metro systems, and thus a tracking and planning system with minimal cost and equipment (although there is no limit on the size of the transportation system that our project can be implemented on) is viable. The CTfastrak project has called for innovative applications and technologies to provide passenger-friendly service.

For any transit system, it will be extremely valuable for passengers to be able to access the accurate and real-time bus location, arriving time, and delay information (if any) anytime and anywhere so to decrease the possibility of waiting too long at the bus station or missing the bus for just being a few seconds late. It is especially beneficial to the passengers waiting at outdoor bus stations during summer and winter time when weather conditions are not ideal. Specifically, the CTfastrak project has bus stations at shopping centers, universities, and other entertainment locations, and with the real-time bus location and accurate arriving time information, passengers

could plan their activities better (e.g., shopping one more store, grabbing a cup of coffee or studying a few more minutes) without feeling over-stressed.

Many transit systems have online and print schedules. However the scheduled times are not updated based on the real-time traffic conditions. A passenger has no idea where the bus currently is and whether he or she has missed the bus if not arriving five or ten minutes earlier than the scheduled time. Earlier bus-tracking^[2] has been used for management and dispatching purpose but not for passengers. With the availability of mobile data communication, it is possible for the passenger to get more accurate bus location information. This senior capstone project develops a bus tracking and trip planning system that provides easy, reliable and mobile access to the real-time bus location and arrival/delay information for both passengers and the transportation management.

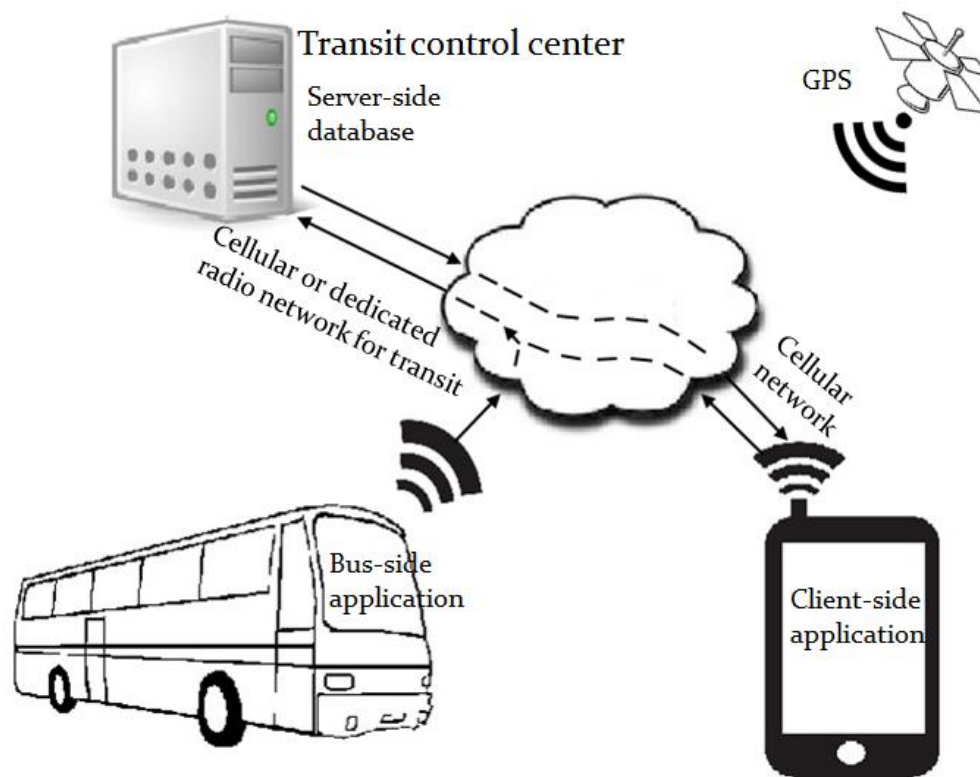


Figure 1. Bus Tracking and Trip Planning System Architecture

The prevalent use of mobile smart phones and Global Positioning System (GPS) technology has made the proposed system possible. The entire system consists of three modules: bus-side application, server-side database and client-side (passenger-side) application, as shown in Figure 1. The current bus-side application is developed on mobile phone with GPS receiver capability to acquire location information, calculate delay, and report to the transit center (server-side). If adopted in the real transit system such as CTfastrak, the application software can be integrated into a small customized hardware (with GPS receiver) that is attached to the bus. The server-side database holds the updated location and delay information for all the buses on the routes and provides response to passenger queries. The user-side application enables the passengers to

access the bus information from the server and plan their trips. Wherever users are, they should be able to quickly and conveniently find out their local bus arrivals, departures, and delays, etc. In addition to this information, users should also have a convenient way to find out more information about the bus vicinity, i.e., points of interest, and plan trips. If a user wants to travel from an area of one city to another, the ability to find out when and where the user should be in order to accomplish this trip via the bus system should be provided.

In summary, this senior project is motivated by multiple factors. First, the State DOT is located closely to the university and there has been communications on the *CTfastrak* project. Second, although there are user-side mobile applications that are open-source and free, there is no free and open-source bus-side and server side control application. It is costly to purchase or obtain license from commercial vendors. There could also be long-term continuous management cost. Third, free user-side mobile applications are normally tied to a specific transit system and cannot be used directly. In addition, a user-side application that has tools and information customized to local transit system and community will be highly desirable. Finally, as an undergraduate senior project, students practice and apply their learned knowledge to a real-life project. Overall, the system resulted from the current project incorporates many features from multiple transit applications which otherwise are not available in a single application.

Related Work

Normally, in addition to printed schedules, bus information in a transit system can be accessed by the users/clients through one of two ways: by visiting the system website directly, such as Where's My Bus (WMB) ^[3], Chicago Transit Authority (CTA) Bus Tracker ^[4], and NextBus ^[5], or by running a mobile application on the local smartphone, such as My Nextbus ^[6] and Centre Area Transportation Authority (CATA) ^[7], which in turn accesses information from the system and present the result to users. Bus scheduling and information systems ^[2, 3] can be very system-specific and targets to a specific city, therefore, most of the applications cannot be easily implanted and they are generally not open-source. NextBus provides service to multiple transportation systems therefore lacking customization. Commercially available systems are sold or licensed to transportation systems and continuous costs may be incurred. Mobile applications are often free to end users; however, they are generally designed for specific transit authorities and are tied to server-side applications which in turn are not open-source and free.

In addition to the above issues, bus tracking systems such as WMB and NextBus do not show real-time bus locations on routes, and they also do not provide points of interest service to the end users. CTA Bus Tracker shows the real-time bus location but there is no mobile application associated with it. Most client-side mobile applications are similar with some providing text-based route information ^[6] and others providing Google Maps graphic-based route and location information ^[7].

The senior project introduced in this paper designs a complete transit bus tracking and trip planning system for the *CTfastrak* project. It not only includes a mobile application, but also the necessary function on the buses as well as the transit control center server application. In addition, this project will incorporate many good features that are found in different previous applications. Unlike most of the other systems that use Google Map, our project adopts

OpenStreetMap^[8], which is open-source and free to use. It also allows us to generate updated and customized maps (e.g., community and local attractions) to reflect map change immediately. Overall, our project objective is to design a customized, cost-effective, efficient and user-friendly bus location and tracking system specifically for the CTfastrak project.

The proposed work involves mobile phone application development, database and network communication between the mobile application and the database server. Multiple software developing platforms and tools are used. The mobile phone application in the proposed work is specifically developed for Android mobile operating system. Similar applications can be developed on Apple iOS too. We chose Android mobile operating system for several reasons. First, it is the most common mobile operating system. Every project group member has an Android mobile device that can be used for debugging and testing the project. In addition, the project is meant to be available to users of a public transportation system via personal devices, so it is the most logical to develop the system in a form that can be used by the most people. Second, Android mobile applications are written in Java, a language that each capstone project member has learned in at least one course in the CET curriculum. Therefore, developing the program in Java satisfies one of the capstone course objectives: to consolidate and integrate concepts, skills and techniques acquired from the curriculum. Third, Android phones provide all the hardware and connection features needed for the project. Specifically, GPS positioning will be used for determining the bus location and for trip planning. Android phones have built-in GPS receiver thus reducing the need for relying on cellular networks for location information, which is much less accurate than GPS. In addition, Java classes for location services via GPS signal are available through the Android API.

The mobile application is developed in Eclipse^[9], an Integrated Development Environment (IDE) that includes a base workspace and an extensible plug-in system for environment customization. Eclipse is a Java-based environment and among its rich extensions are the plugin for Android development and the web tools platform. Most of the system-user interfaces are designed, implemented and debugged in Eclipse. Android Software Development Kit (SDK)^[10] is used and provides all the essential application programming interface libraries necessary for Android application development. Together with Eclipse, they are used as a baseline in creating the unique bus tracking and planning application.

For network communications between the bus-side application and the database server as well as between the client-side (passenger-side) application and the database server, HTTP web-based query and response are used. HTTP is widely used to connect to the internet and Android SDK offers HTTP connection services through Java API classes. Since HTTP is compatibility-friendly and available through the API, it was the best choice for a connection protocol between the mobile applications and the server. At the server-side, Windows Apache MySQL and PHP (WAMP) server platform is used for database management, location updates, queries and responses. The data format in the network communication adopts JavaScript Object Notation (JSON)^[11], which is a language-independent data format that uses human-readable text to transmit data objects in attribute-value pairs. Java also allows for easy parsing of JSON objects and using JSON is simpler and quicker than transferring the alternative XML files. This is especially beneficial to the proposed system as it enables low bandwidth consumption and easy encoding/decoding of the information on the Android devices.

System Design

This section first overviews the entire system, then describes the design details of each module of the system. The bus tracking and trip planning system consists of three modules: user-side, server-side, and bus-side applications. Java is used to program for Android devices along with the Eclipse and Android SDK to help build the applications. The bus-side application tracks the bus location via GPS and periodically sends update about its current location and delay (if any) in JSON format to the server via HTTP connection. The location information is in form of geographic coordinates with latitude and longitude and is obtained using the phone's built-in GPS receiver. Based on the real-time location, the bus-side program calculates the delay and sends both the delay and current location to the server. The server provides the HTTP service and manages the location and delay database for all the buses and responds to the user-side application request. When the user-side application sends a request for the bus location and delay, the server will push a JSON object to the device. The user-side application provides user interface for location and delay query, bus station information, schedules, and trip planners. It will also analyze the response from the server and display the information to the clients.

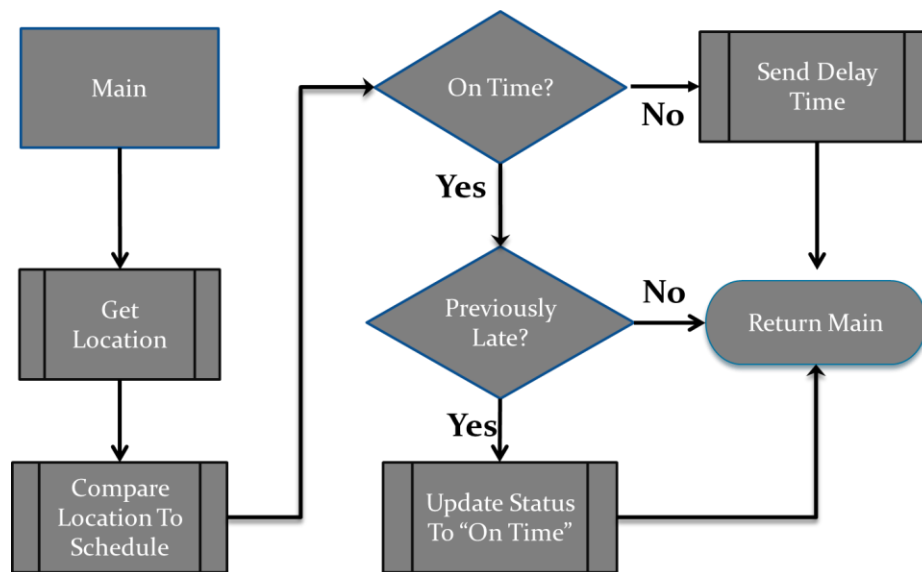


Figure 2. Bus-side application flowchart

Bus-side application - The bus-side application is responsible for reporting location and delay information to the server for the individual bus that it is assigned to. Figure 2 shows the flowchart process. This program module does not have a user interface, although an interface can be designed to show the bus driver current location and delay information. The main function of the program is to periodically collect the geographic coordination from the GPS receiver device and compare to the scheduled time on route. Depending on the result (delayed or on-time), the delay information is sent to the server. It is also possible that a previous delayed status is updated with on-time status. To report the information, a HTTP connection is established between the

server and the bus-side application. Once a good connection has been established, the bus-side application will upload its location information to the server encoded in a JSON object.

bus_location

Column	Type	Null
Bus_Num	int(1)	No
Latitude	text	No
Longitude	text	No

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	Bus_Num	7	A	No	

Figure 3. Example of the Server-side database table

Server-side application - The server interacts with both the client-side application and the bus-side application, with both using HTTP connections and JSON encoding. The server maintains databases for the buses and records their current location and delay information. Database is created and managed through PHPMysqlAdmin. PHP scripts are created and executed when application needs them to perform MySQL queries. When a connection to a bus-side application is open, the bus sends location and delay information to the server. The server then updates its database with the information from the individual bus. When a connection to a client-side application is open, the server will be queried for the location and delay information. Figure 3 is an example of the database tables that defines the bus location.

Client-side application – The client-side application accomplishes the following tasks:

- Provides detailed bus route information, including route overview and information about each stop. Client users can download the client-side application and have the route and stop information updated.
- Provides interactive map-based trip planning service. Client users can input the desired destination, and based on the current location of the user (through the phone's GPS module similar to the bus-side application), the system will provide a calculated route and display the route on the map. Different indicators on the map will show the client's current location, nearest bus stop and the destination bus stop.
- Provides a user with turn-by-turn navigation to any desired bus stop. This helps a client to arrive the nearest bus stop from the current location.
- Displays points of interest near the bus route.
- Shows clients the real-time locations of buses. Client users can view interactively on the map the real-time location of the buses as well as their delay information.

Figure 4 shows the trip planning process. It includes trip planner initialization, bus stop change and turn-by-turn route update.

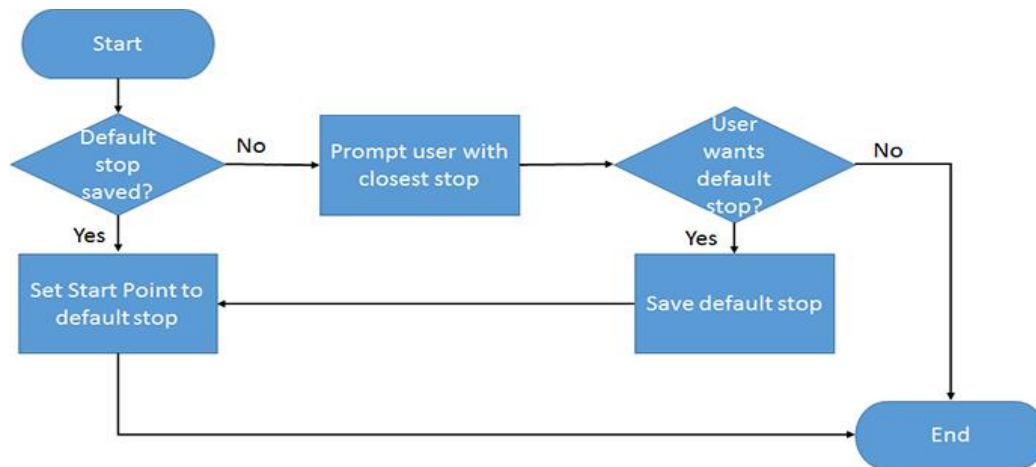


Figure 4 (1). Client-side application: trip planner initialization

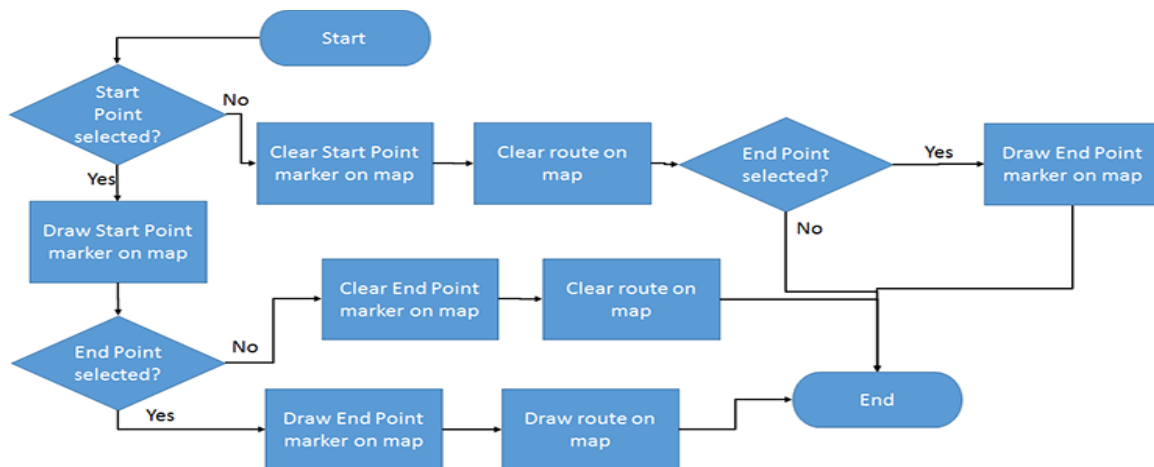


Figure 4 (2). Client-side application: trip planner bus stop change

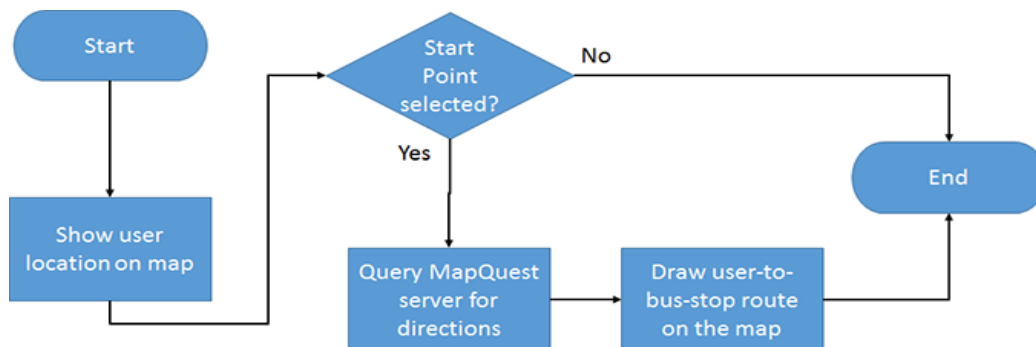


Figure 4 (3). Client-side application: trip planner turn-by-turn

Implementation and testing

The bus tracking and trip planning system is implemented using the software platforms and tools introduced in the previous sections. Both the bus-side and client-side applications have been successfully installed on Android mobile phones. A testing WAMP server is setup on the Internet to simulate the transit control center where the database is hosted.

Pre-testing - Once the individual applications work properly on their own, network communication is tested between the bus-side, server-side and client-side applications. First, hard coded location values are input in the bus-side application and periodically sent to the remote server to test if the data was correctly received, parsed and stored in the database. Then the application is changed to post the real location data to the server. The server and the client-side applications are tested by having a location already in the server database to be retrieved and displayed on a map. The bus-side application must enable the GPS receiver on the phone to retrieve the current location.

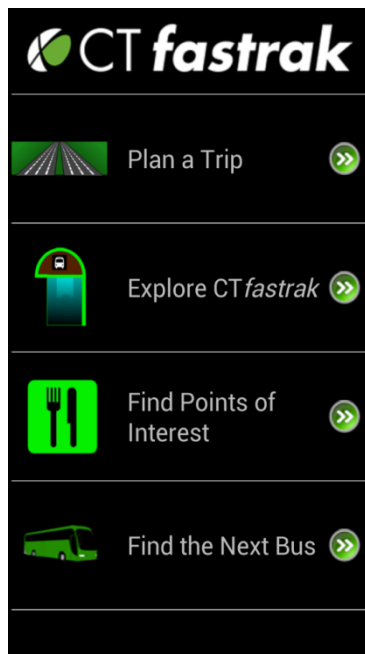


Figure 5. Home screen



Figure 6. Bus tracking

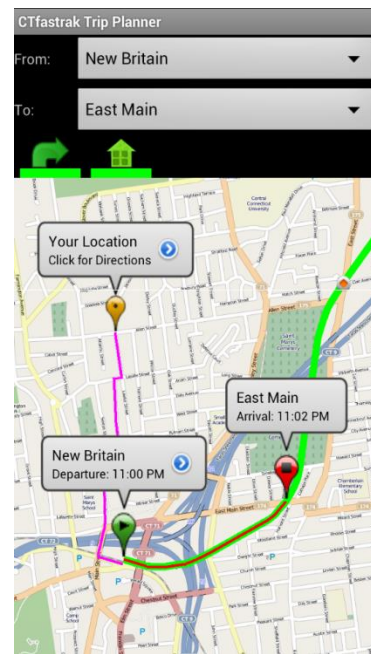


Figure 7. Trip planner

Client-side home menu - Figure 5 shows the home screen of the client-side applications. By extending the menu on the home screen, clients can track the buses on route, plan a bus trip, browse the available stops and find points of interest near the bus stops.

Bus tracking - When the pre-tests are successful, the bus-side location can be displayed consistently on both the bus-side and the user-side applications. Figure 6 shows the two buses that are currently on the queried bus route and the related information such as stop name and arriving time.

Trip planner – Figure 7 shows the trip planning function of the system. Clients can select a destination and based on the current location, the system will be able to find a route with departure and destination stops as well as the path to get to the departure stop from the client's current location. It also shows the departure time and arriving time. Because the CTfastrak project is in on-going phase, there is no set-up transit database and route information. The project team created a small database simulating the main route. Routing between bus stops on different routes will be presented in near future work.

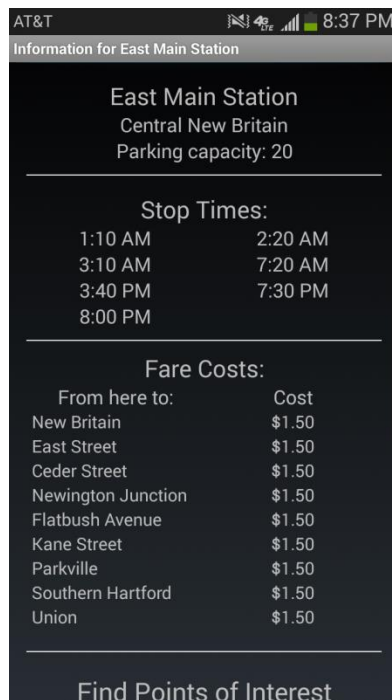


Figure 8. Station information

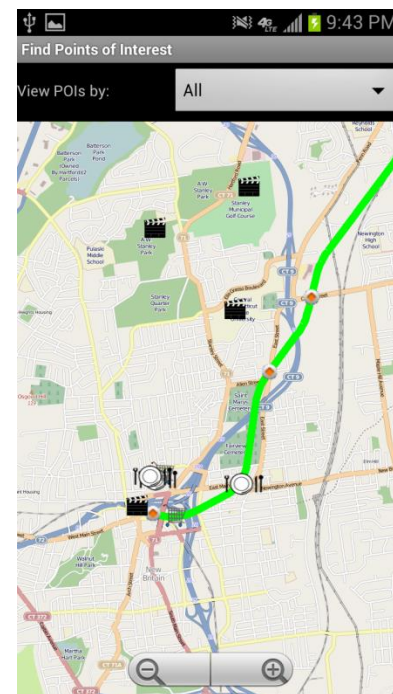


Figure 9. Points of interest

Bus station and points of interest – As shown in Figure 8, the system also allows clients to check the bus station information including the parking lot size, departure times and fare costs. Client can also find the points of interest of each station, as shown in Figure 9. The points of interest were intended to supply a user with helpful and interesting information about the route and passengers can become more familiar with the CTfastrak system and the nearby locations. The system allows the customization of the station and points of interest information, which can provide more convenient service to the bus passengers in the area.

Conclusion and Future Work

In summary, this paper presents the application design and development of a bus tracking and planning system using the CTfastrak project in Stat Z as a background. Overall, the system is well designed and implemented. The bus-side, server-side and client-side applications provide all the expected functions and seamlessly communicate with each other. The developed system has been tested and demonstrated to the Office of the Commissioner, Department of Transportation (DOT) of State Connecticut. The collaboration between the university faculty, students and the

State DOT department has been well recognized by the Commissioner through public media. Further meetings have been arranged with multiple DOT departments to bring the project closer to the CT*fastrak* implementation.

For future work, the system can be improved and extended in a few aspects. Currently, the bus-side application is implemented using the same software development platform as the user-side application on a mobile phone. If adopted in the transit system such as CT*fastrak*, the application software can be integrated into a small customized hardware (with GPS receiver) that is attached to the bus to reduce cost and enhance performance.

For the purpose of this project, a working and functional framework is the most important. In the future, the interface design can be further improved to be more user-friendly. The server-side architecture can be improved as well. For the project, a server running on a home computer was used. In a real-life bus transit system, however, it's expected that many clients will use the service. Therefore, a dedicated server platform or a third party hosting service is necessary to provide fault tolerance and redundancy.

Lastly, the bus-side application can be adjusted to work in the radio system of the bus line, if one is already available. This allows the bus line to work with and improve a system that is already in place rather than purchasing mobile cellular service for each bus.

References

- [1] CT*fastrak* Project, <http://www.ctfastrak.com>. Last retrieved on 2014-3-8.
- [2] M. Dessouky, R. Hall, A. Nowroozi, and K. Mourikas. "Bus dispatching at timed transfer transit stations using bus tracking technology", *Transportation Research Part C: Emerging Technologies*, 7(4): 187-208, 1999.
- [3] WMB: Where's My Bus, <http://wmb.unm.edu/>. Last retrieved on 2014-2-11.
- [4] CTA Bus Tracker, <http://www.ctabustracker.com/bustime/home.jsp>. Last retrieved on 2014-2-11.
- [5] Real-time Passenger Information, <http://cts.cubic.com/solutions/realtimepassengerinformation/nextbus,inc.aspx>. Last retrieved on 2014-2-12.
- [6] My Nextbus, <https://play.google.com/store/apps/details?id=com.blogspot.msandroid.mynextbus&hl=en>. Last retrieved on 2014-2-12.
- [7] Centre Area Transportation Authority (CATA), <http://www.catabus.com/Announcements/Mobileapps.html>. Last retrieved on 2014-2-13.
- [8] OpenStreetMap, <http://www.openstreetmap.org>. Last retrieved on 2014-2-12
- [9] Eclipse Foundation, <http://www.eclipse.org/>. Last retrieved on 2013-10-16.
- [10] Android SDK, <http://developer.android.com/sdk/index.html>. Last retrieved on 2013-11-1.
- [11] JSON, RFC 4267, <http://www.ietf.org/rfc/rfc4267.txt> . Last retrieved on 2013-11-14.