Building a Real-Time Bus Tracking Data Display System

Jason Dudley
jaydud@gmail.com
Department of Computer Science

Ron Vetter
vetterr@uncw.edu
Department of Computer Science

Jeff Brown
brownj@uncw.edu
Department of Mathematics and Statistics

Tom Janicki
janickit@uncw.edu
Dept. of Information Systems & Operations Management

UNC Wilmington
Wilmington, North Carolina 28403 USA

Abstract

Mass transit competes with automobiles to move passengers. However, until recently, mass transit has not utilized technology to keep their potential passengers informed of estimated time of arrival (ETA) of buses or even mechanical breakdowns of buses on the passenger’s route. A local public transit authority desired to increase its level of technology and communication with passengers as it built a new transfer station between its routes. They felt this increased level of communication would have a positive effect on the perception of their service by potential riders. This research discusses the design and implementation of a system to provide real time tracking of buses for a public transit authority. Key requirements were the development and display of ETA for all buses via mobile, web, LCD and LED displays. The research builds on the work of other transit authorities and Google. Once implemented a study found that the ETA algorithm developed by the researchers was superior to the estimates provided by Google for identical routes. Results of the study are included.

Keywords: bus tracking, global positioning, mobile web development, estimating arrival times

1. INTRODUCTION

Real time tracking is becoming more and more popular as devices utilizing the Global Positioning System (GPS) become more readily available. There are a plethora of different uses for this technology from keeping track of loved ones to guiding cruise missiles to their target. Wave Transit, a local bus service provider for a small metropolitan area of about 160,000 citizens has a system in which the buses send their coordinates (latitude and longitude, among other data) to a central database every thirty seconds using AT&T's cellular data service. This data allows the dispatcher to know where all the buses are at any given time. The data has previously been used on the Wave Transit website to show the current location of specific buses. Because Wave Transit opened a new central bus ‘transfer’ station (Forden Station) they wanted to make this data available to the passengers who come to the station in a way that benefits them.

The primary goal of this research was to enable passengers to easily access data related to the bus route they had interest in (especially estimated arrival time). This goal was accomplished by creating three display monitors at Forden Station. Two of the displays show the estimated arrival times of the six different bus routes that come to the transfer station. These displays also serve as a means to provide current news and information about Wave Transit. For instance a message can be displayed to broadcast bus downtime and any emergencies that may arise on any of the routes. The news and information is updated by Wave Transit employees using a content management system. The third display shows a real time map marking each of the buses with a label that identifies the location of each bus.

Another goal of the research was to make all of this real time data available to the public on the Wave Transit website. The website displays a list of all available buses. Users can click on and view a real time map for each bus. An information window on the map shows users the route, next stop, estimated time of arrival (ETA), and the buses' direction. These pages had to be made in a way that is mobile friendly so passengers can pull up maps on any mobile device while they are at a bus stop and watch the bus make its way to the stop. Since many users have smart phones, the website detects which mobile browser is requesting the information and formats the information for the smaller screen size. In addition, the website centers the map on the bus’s location and follows it as it moves throughout the city.

The data presented on the displays inside the transfer station is invaluable to the passengers that it serves. They are able to quickly and easily see the amount of time until their bus will be at their particular stop. Having this information readily available should help to increase ridership within the Wave Transit service area. In addition, the perception of Wave Transit's bus service has improved due to the fact they have this technology available. It seems that many major cities have this technology and some have just been released as recently as 2010 (Melanson, 2011*)*. Because approximately 25% of Wave Transit's ridership is directly due to a local university (TJR Advisors, 2009), as the technology becomes more widespread the university ridership should also increase. The reason for this is a student can quickly see how soon a bus will be at a particular stop and he/she will be able to plan accordingly. The public benefits from this research because the project is showcased at the bus station and on the Internet for all the public to see.

2. BACKGROUND AND RELATED WORK

Currently several large cities have similar bus tracking applications in place. The New York Metropolitan Transportation Authority (MTA) recently released a pilot project on a single route in Brooklyn called MTA BusTime. They have a web page which displays this route with information on each stop showing how far each bus is from that stop, a bus icon is also displayed to show where each bus is on the map. This web page has been developed in a way that it is mobile friendly as well, allowing users to access it at their convenience. MTA updates the map with new data every 30 seconds, much like Wave Transit's system does. This pilot has not been live long enough for the MTA to determine if it is increasing ridership on that route (MTA Bus Time, 2011).

Regionally, both North Carolina State University (NCSU) and the University of North Carolina at Chapel Hill (UNC) have bus tracking applications. The NCSU application is provided by a company called TransLoc. The TransLoc application offers a real time map at the bus stop, on the web, and anywhere via mobile application. The map shows the relevant bus routes, stops, active route color key, and real time bus location. Each route is color coded and has a clickable marker at every stop. The information window that displays after clicking a marker shows what buses stop at that location and the estimated arrival time of that bus. The map updates every second so you can see the buses' actual location in smaller increments. This is a unique feature that this particular application has that is of great benefit to the user because the bus location is more dynamic. On the mobile website the available routes are shown first, once a route is selected the map is shown with only that route overlaid on the map. This application focuses solely on the map to provide information about the bus schedule (TransLoc, 2011).

UNC uses an application provided by NextBus Incorporated (NextBus, 2011). The application is not centered on the map, instead it uses a form with dropdowns that enables the user to choose a route, direction and stop. After entering this information the ETA is provided for the next bus in that route to reach that stop. On the final page an option for viewing a Google map is given. A Google map draws the route with all the stop locations, each stop location is clickable providing an information window on the route, direction, and next departure time. This application offers the ability to hide or show the vehicles and stops on the map using a checkbox. The other unique feature is the ability to select more routes to be shown on the map. There is not as much focus on mobile users in this application as it just offers an image of the map with the bus location noted with two blue lines (Town of Chapel Hill, 2011). The user interface is lacking as it is tedious to select the correct route, and the stop and destinations, especially if you are not familiar with the area.

Text Messaging Capabilities

Wave Transit quickly saw the value in texting and has already implemented short message service (SMS) capability to receive real time bus location information. SMS is available on over 98% of all cell phones which makes this service available to almost any passenger with a cell phone (Cell Signs, 2011). Passengers can simply text "Bus 101" to 90947 to receive that buses next stop. University routes are a little different where passengers can text "Bus R" and get the next stop of the bus on the university Red Route. When a passenger sends a text message to 90947 it is sent from their service provider to an SMS aggregator who then passes it on to a content provider (in this case Mobile Education LLC is the content provider for the short code 90947). Mobile Education has server software that allows it to make a database query against the Automatic Vehicle Location (AVL) database which holds the buses next stop. The content server then replies with the information from the AVL database to the SMS aggregator who sends it back to the passenger’s phone using their service provider. This system allows a passenger to quickly and easily find out where a bus is in any given route, but with low accuracy because it only shows the next stop. The new application, developed as part of this research, improves on that by showing a map of the bus with ETA to the next stop, however, it will only be available to smart phones.

Google Transit

Another service that Wave Transit offers to its passengers is Google Transit. Google Transit is a public transportation planning tool that combines public transit data such as stop, route, schedule, and fare information with the power of Google maps (Google Transit Partner Program, 2011). A passenger interacts with this service through the normal Google maps interface which is already available in 12 different languages and is mobile friendly. After going to Google maps passengers simply get directions from one place to another in the chosen city. Google offers a public transportation option as a mode of travel along with each step the passenger would need to take to get to their destination. This involves uploading very detailed route data to Google. With this data Google performs the necessary calculations to offer it as a transportation option in the host city. There are several files that need to be submitted to Google to make this work, but they are all publicly available at this URL http://www.wavetransit.com/gtransit/google_transit.zip.

3. IMPLEMENTATION

Application

The system required that many different parts work together to accomplish the specifications set forth by Wave Transit. The source of all the data that is presented originates in the AVL Database. This database is where the buses send data every 30 seconds. The buses send information such as longitude, latitude, heading, route, and speed. From there the data is accessed on two different web servers. The estimated arrival time server grabs the data and does calculations on each bus and outputs a comma separated string to a specific URL (http://mymobed.com/map/all.jsp). The second server uses the AVL data to map the buses on a real time Google map. This map is accessible on the web via the following URL -

http://www.wavetransit.com/RealTimeBusTracking.aspx. Figure 1 in Appendix A provides an overview of the servers and transfer station displays
The second server also communicates with the estimated arrival time server to request the ETA string and parse it to display in a table that is presented to the user via another URL. The server inside the Wave Transit station can access all the content from both of the servers via the Internet. There is a Java application running on the inside server that makes a request for the ETA string, parses it, and sends the times to each specific LED display, mounted above each buses' stop. The Java application opens sockets to each LED display individually and sends the appropriate arrival time. The communication uses the wireless local area network at the station. Two PCs control two of the liquid crystal displays (LCD) via High-Definition Multimedia Interface (HDMI) ports. The displays use a "spanning" configuration in Windows to show two different web pages displaying either ETA or real time map in full screen mode. The other PC controls the other display in the same fashion. Figure 2 in the Appendix displays the flow of data from the buses to the servers and finally to the LCD, LED and web page displays.

Design Issues

A design concern was the presentation of the data to the user. At the bus station the data on the page changes without refreshing the entire page. Previously a meta refresh was used to update the information. That wasn't acceptable because it would cause a noticeable blink of the user interface (UI) every 30 seconds. Since JSP is a server side language the only way around this was to change the data using a client side language like JavaScript. Also, the displays are all 1080p so all of the interfaces needed to be optimized for that resolution. This would have represented a challenge for the news page that was part of Wave Transits public site. The public site should be accessible to all resolutions therefore it cannot be optimized for one of the highest resolutions (1080p), instead it is optimized for the more common resolution of 720p. The news page was eventually merged with the arrivals display in prototype 3 so, this concern was eliminated.

Software and System Quality Metrics

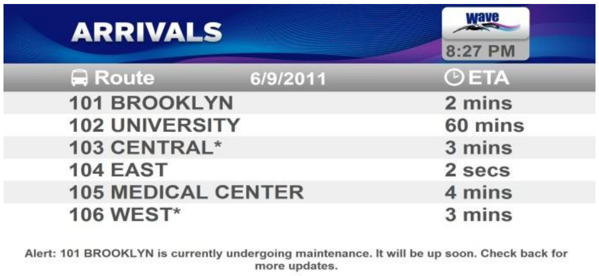
The International Organization for Standardization (ISO) defines quality as "the totality of characteristics of an entity to bear on its ability to satisfy stated or implied needs" (ISO, 1986). This definition itself is very vague, but in essence the quality will be measured by how well it conforms to the requirements and the extent to which it can be used as it was intended. In other words this project would be considered successful if it meets the scope in the allotted time, satisfies the customer (Wave Transit), and reaches the ultimate goal of providing a benefit to the passengers of Wave Transit. Meeting these broad goals ensured that the system met the requirements set forth by Wave Transit.

The software was developed based on best practices expressed for source code quality including readability, maintainability, low complexity and robust error handling. Because readability and maintainability go hand in hand, the software was written in a way that another individual can easily understand and update it if necessary. The software was broken down into individual components, each focused on a particular and singular task so as to have the least complexity possible. Finally, robust error handling was enforced to have a system that was capable of running 24/7 and provide the user with a system that provided them with the tools they needed to operate it successfully.

4. RESULTS

Design Methodology

The spiral model (Pressman, 2005) proved to be an ideal way to implement the Wave Transit real time bus tracking system. Each prototype allowed for enhancement visually and functionally. As soon as prototype one was released for the public several suggestions came up for improvement in prototype two. For example, on the arrivals display inside the bus station a clock, date, and message of the day were all added in the following prototypes to increase usability for Wave Transit passengers. In addition, the map display was improved with a better orientation on the screen to show more of the local bus service area and a "you are here" marker was added to show passengers where they were located while looking at the map. Along with these improvements we were also able to spot a few bugs in prototype one that were remedied in prototype two. One issue was a caching issue specific to Internet Explorer where it would cache the location of the buses which caused the map display to show the buses in the same location on each 30 second update. Figure 3 displays the ETA screen inside the transfer station while figure 4 is a screen shot of one of the 60" LCD's inside the transfer station.



ARRIVALS

wave

8:27 PM

Route 6/9/2011	ETA
101 BROOKLYN	2 mins
102 UNIVERSITY	60 mins
103 CENTRAL*	3 mins
104 EAST	2 secs
105 MEDICAL CENTER	4 mins
106 WEST*	3 mins

Alert: 101 BROOKLYN is currently undergoing maintenance. It will be up soon. Check back for more updates.

Figure 3 – Final Arrivals Display

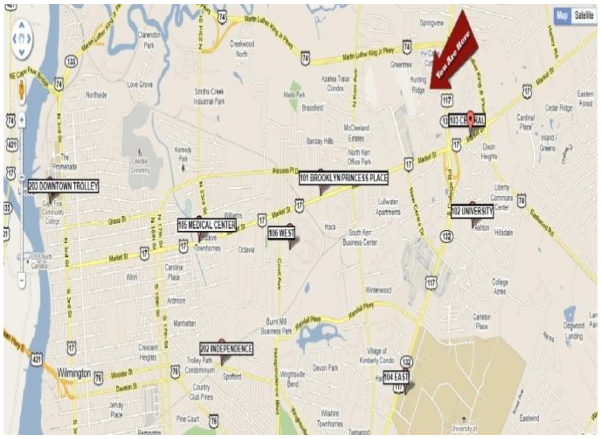
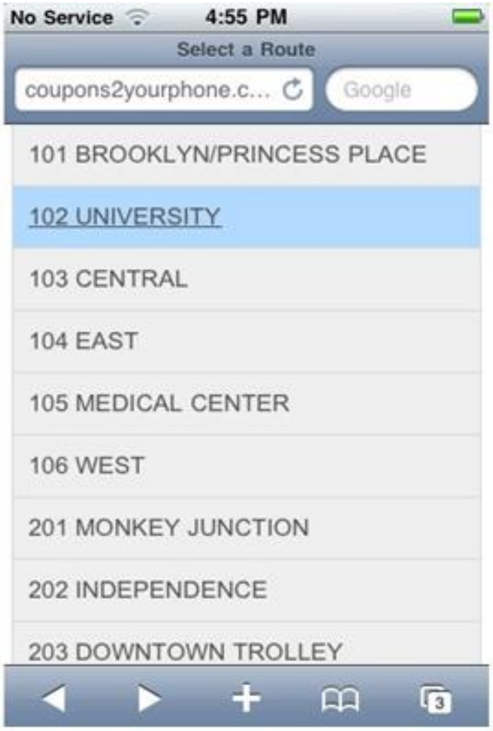


Figure 4 – Final Maps Display Inside Transfer Station

Website

The website implementation started in prototype two as a simple upgrade to the existing system that used Yahoo maps. The only option was to choose a single route and have those buses show up on a map. The newly developed system uses Google maps and allows passengers to interact with the map by clicking on the bus marker to see information on that specific bus. Previously this information was shown above the Yahoo map image. Another improvement was to show all buses in the same route. The old map would only show a single bus on each route and the user had no control over which bus was shown.

In prototype two, the user could see all the buses on any given route as well as the information specific to each bus. The text on the real time tracking page before was quite lengthy and may have distracted users from the pages main purpose. In prototype two, the page was made much shorter by moving the SMS texting instructions to its own page and providing a link for users to navigate to the page if they were looking for that information. The intent of the real time tracking page is much clearer now as the route selection interface is very prominent on the page and the use of bright hover states draws the user's attention.



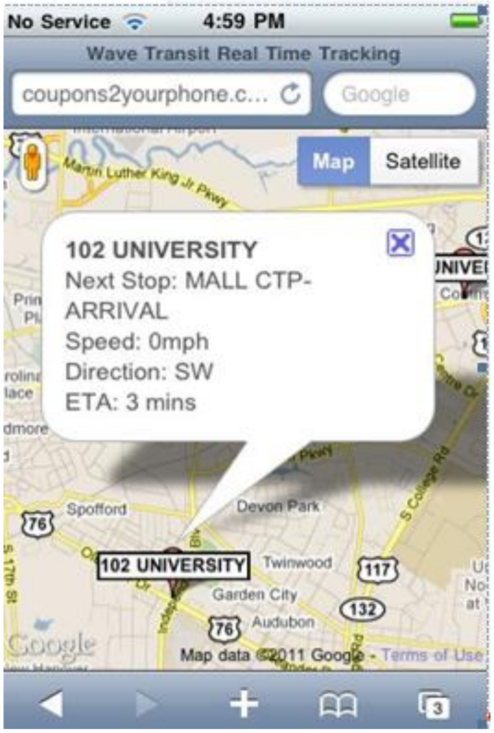
Select a Route

coupons2yourphone.c...

101 BROOKLYN/PRINCESS PLACE
102 UNIVERSITY
103 CENTRAL
104 EAST
105 MEDICAL CENTER
106 WEST
201 MONKEY JUNCTION
202 INDEPENDENCE
203 DOWNTOWN TROLLEY

Figure 5A: Selecting Information for a Particular Route

Prototype three was a huge improvement over prototype two with the implementation of a mobile friendly tracking page as well as the implementation of the other use cases. Now a user can select from 3 options on the website, a table display of a single route, a map display of a single route, or all of the routes on a single map. The mobile friendly version has a very simple user interface to cut down on load times. It has the option to view a table which is a text based display of a specific route or view a map of a specific route. Due to the large number of buses to display, a map with all buses was left out of the mobile version because it was unusable on a small screen. Along with all these improvements a simple implementation of Google's direction service was used to give the user an ETA for the bus to arrive at its next stop. This information is available along with route name, heading, speed, and next stop on all versions of the map and table display. Figures 5A and 5B illustrate a sample smart device screens from selecting one route to ETA information for that route. Figure 6 in the appendix shows a sample 'full web' page that a user might see on a tablet device.



102 UNIVERSITY
Next Stop: MALL CTP-ARRIVAL
Speed: 0mph
Direction: SW
ETA: 3 mins

Figure 5B: Smart Phone Display for a Bus on the Selected Route

LED Program

The final implementation of the LED Java code resulted in a shell program that calls a Java program twice every minute. Since CRON has a minimum scheduling resolution of 1 minute, it calls the Java program, sleeps for 30 seconds, and then calls the same program again. The Java program requests the ETA values from the server for each of the routes and then sends a command to the appropriate display to show the route name and ETA. Each bus has a specific LED sign that it stops under which has a static IP address. The Java program statically assigns IP addresses to display appropriate information on the correct LED sign. Outside of normal business hours, the program simply sends commands to clear the displays.

In addition to the Java that was written to display information on the LED displays, a UNIX shell program was written for a Wave Transit employee to execute a variety of commands on a single LED display through a command line interface using SSH. SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices (Networking Working Group of the IETF, 2006). The shell program has a number based menu that is very easy to use and explains each option that is available. When an employee logs in using SSH they are presented with a menu where they can stop or start the automated LED Display program as well as choose from more advanced options. Some of the advanced options include: set RTC time, set RTC date, set brightness, get display configuration, get error status, and execute an LED test. A user guide was provided to show Wave Transit how to access the program if they ever have a problem.

ETA Calculation

The ETA estimate algorithm calculates the ETA for the six bus routes that stop at the Forden Transfer Station. This estimate is shown in real time on LCD and LED displays to give passengers an idea of when their bus will arrive. Additional data from the AVL database was needed to calculate the ETA. Not only does the AVL database store information about the buses current statistics, it also has tables that contain information about each stop on a route and coordinates along a route called breadcrumbs. They contain the latitude, longitude, heading and timestamp information at different points along a given route. Each route has over 1000 breadcrumbs collected by driving the route and gathering data, but not stopping to pick up passengers. So, the total time to breadcrumb a route could be half the time it takes the bus to complete the route. To account for this time difference an error is added to the calculation after all the remaining breadcrumbs are added up. This error factor is only added if the estimate is more than five minutes. If the estimate is less than five minutes then there are not many stops left so the breadcrumb times should be accurate.

Data collection was performed on the six routes to determine how close to the actual time the estimates were. After collecting data for a full Saturday which included about 14 cycles for each route, the average of actual ETA minus estimated ETA was 465 seconds with a median of 188 seconds over all routes at any given time. It was expected the average to be quite a bit higher than the median because there are instances where the bus is at the station and the actual is at zero while the estimate has begun calculating the next cycle. A median of 188 seconds is a closer estimation of how accurate the estimate is because the outliers are not factored in. From the data, we see that the estimates are more accurate as the bus gets closer to the station. So, when the bus is an hour away it would be acceptable to be off by a few minutes. The ETA serves its purpose and Wave Transit considers it close enough to the actual for a passenger to plan accordingly. As figures 7 to 12 show, some routes are more accurate than others. They ranged from a median of 285 seconds on route 102 to a median of 120 seconds on route 101.

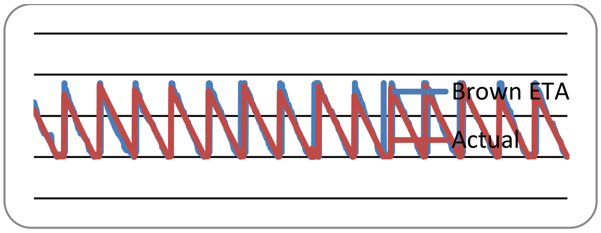


Figure 7 - 101 Brooklyn (Average 261 seconds, Median 120 seconds)

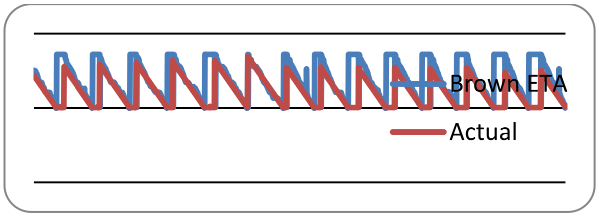


Figure 8 - 102 University (Average 969 seconds, Median 285 seconds)

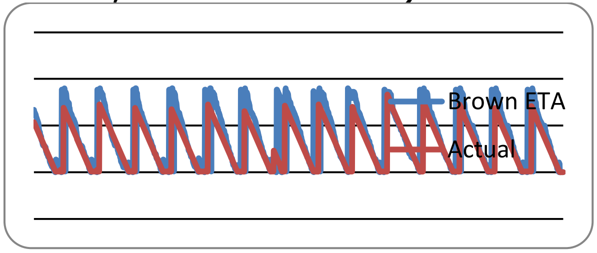


Figure 9 - 103 Central (Average 576 seconds, Median 271 seconds)

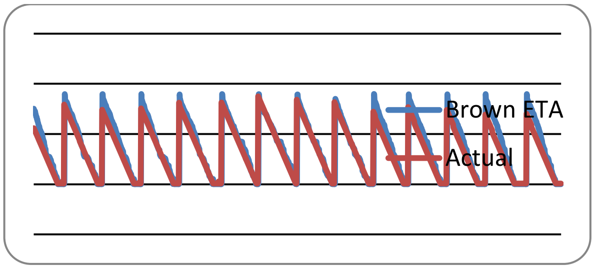


Figure 10 - 104 East (Average 258 seconds, Median 162 seconds)

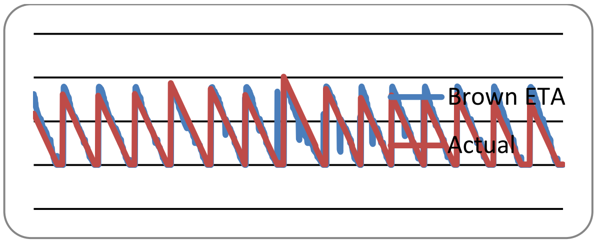


Figure 11 - 105 Medical Center (Average 278 seconds, Median 158 seconds)

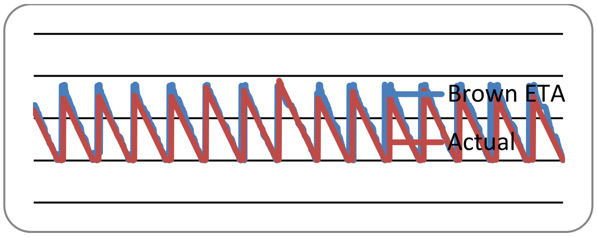


Figure 12 - 106 West (Average 436 seconds, Median 202 seconds)

One interesting question which arose was, "How do these results compare with the time estimates provided by Google's direction service?" Starting with a simple implementation of Google's direction service which determines the ETA to the next stop on Wave Transit's map interface, a more complex program was written to determine the ETA to Forden Station. Google's direction service allows for waypoints to be passed along with a directions request to keep the directions along a specified route. As the bus moves along its route waypoints are removed after the bus has passed them so they no longer affect the ETA. This would make sure that the ETA that Google returned would at the very least account for each stop before it would arrive at Forden Station.

After implementing the Google ETA for Forden station, data was collected to figure out how precise the ETAs values were. A simple program was written using JSP, JavaScript and HTML.

Every 30 seconds the JavaScript would get the researchers' ETA, latitude, longitude and make a call to Google directions for an estimated ETA. Once the data is collected it sends it to a JSP file which executes an insert into the "ETACompare" table in the database. The data was logged from 8am to 5pm so there would be several iterations of the bus leaving and arriving at the station. After the data was collected, it was updated to record when the bus had arrived at the station. The bus arrival at the station was determined by drawing a bounding box around the station using the maximum and minimum latitude and longitude the bus could be in and still be at the station. From there it was an easy update statement to set an "arrived" bit field in the table to one when the bus met the specifications. After the update statement had run, the actual time of arrival could be determined by subtracting two dates for each record. The data would have to be analyzed in subsets so we would only be dealing with one circuit at a time. The first date comes from when the bus arrived during the current circuit and the second date was the current record that is being analyzed within that subset. After the actual time was computed for each record we could then graph the results to see how close the Google ETA was to our estimate.

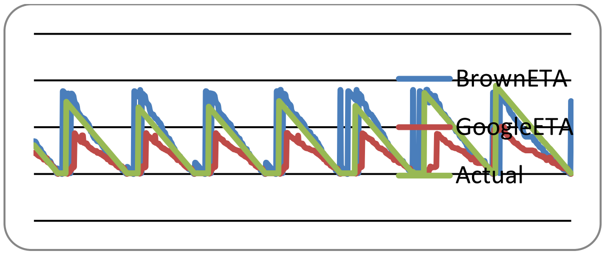


Figure 13 - 103 Central (Google vs. Researchers)

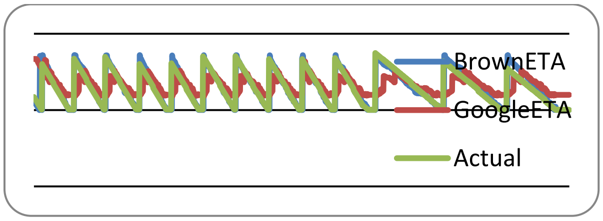


Figure 14 - 104 East (Google vs. Researchers)

As shown in figure 13 and 14, our ETA turned out to be more accurate than the implementation of Google's direction service. The problem with Google's ETA is that there is a limitation of eight waypoints per direction request. With only eight waypoints the best way to keep a bus on its correct route was to use the buses stops as waypoints. Google will calculate the most efficient route to get to the next stop which will not necessarily match the buses actual route. Also, Google will not account for the bus stopping at the waypoints to pick up passengers which is a variable amount of time. The third challenge was the data in the AVL database was not normalized which made it difficult to pull out the right stops in the correct order. With more normalized data, adding an adjustment for stop times, and upgrading to Google Premier which allows for 25 waypoints per request, Google's ETA could be made much more accurate.

5. SUMMARY

Many different skill sets were required to deliver a fully functional real time bus tracking data display system on time and on budget. This included several designs including an arrivals display, bus map display, and a mobile map interface. The design required focus on different aspects of each deliverable. For instance, the arrivals display was designed to be visually appealing to passengers at Forden Station while the mobile map interface was designed for simplicity and minimal load times. However these skills may be used by other smaller transit authorities to begin providing their customers more accurate ETA route information.

A key programming challenge was to communicate with the Google map API to bring Wave Transit's data to the map in an intuitive way. JavaScript played an important role not only in the Google Map API but to refresh elements on the screen without reloading the page. That combined with an XML document generated from a JSP file that communicated with the AVL database enabled the project requirements to be met. The programming deliverables included an interactive real time bus map and a real time arrival display that updates with current data every 30 seconds. The map also includes an information window that displays route, next stop, direction and ETA. A very simple administration page was provided for Wave Transit employees to quickly and easily update the news message on the arrivals display. Each of these deliverables is represented by several files including; HTML, CSS, JavaScript, JSP, Java and several images.

Finally, a simple analysis of Google ETA versus the researchers' ETA was included to verify the accuracy of the provided estimations at Forden Station as well as the viability of using such methods. Google ETA was added to two different routes, 103 Central and 104 East. Charts were created to see if there were any patterns in the data to work with. From the charts it was easy to see that the implementation of Google's direction service would not be adequate to provide these estimations. Data was also collected for all six routes at the same time for an entire day. The charts resulting from this data collection show how accurate the estimations are. From the data that was collected we may adjust the ETA algorithm to make it even more accurate in the future.

Wave Transit is very pleased with the new implementation of their real time bus tracking system. It exceeded their expectations in almost every way. The spiral model proved to be very beneficial throughout the project because the customer had suggestions for each prototype. It was very easy to add those suggestions in the requirements for the next prototype. Having the customer aware that updates could be accomplished so easily made the project go much more smoothly. With the system live and available to the public Wave Transit can now collect more suggestions through their online contact form and information desk at Forden Station for future enhancements.

6. REFERENCES

CellSigns (2011). "Mobile Statistics". CellSigns Last retrieved: 5/20/11. http://www.cellsigns.com/industry.shtml

Google Transit Partner Program (2011). Last retrieved: 5/26/11. http://maps.google.com/help/maps/transit/partners

ISO 8042 (1986). "Quality Vocabulary" International Organization for Standardization, Geneva, Switzerland.

Melanson, D. (2011). "Brooklyn bus riders get real time bus tracking via cellphone". Last retrieved: 5/2/11. http://www.engadget.com/2011/02/05/brooklyn-bus-riders-get-real-time-tracking-via-cellphone/

MTA Bus Time (2011), "Metropolitan Transportation Authority", Last retrieved 3/12/11. http://bustime.mta.info

Networking Working Group of the IETF (2006). "The Secure Shell (SSH) Authentication Protocol", January 2006, RFC 4252.

Nextbus (2011). "NextBus", Last retrieved: 3/26/11. http://www.nextbus.com

Paulk, M. (2001). "Extreme Programming from a CMM Perspective", *IEEE Software*, 18 (6) 19-26.

Pressman, R. (2005). "Software Engineering: A Practitioner's Approach". McGraw Hill, Boston, MA.

TJR Advisors (2009). "Organizational Structure and Analysis, Wave Transit – Wilmington NC". Last retrieved: 3/14/11. http://www.wavetransit.com/org_study_final_report.pdf"

Town of Chapel Hill (2011). "Town of Chapel Hill Transit." Last retrieved: 3/26/11. http://www.townofchapelhill.org/index.aspx?page=1175

TransLoc (2011). "The NSCU Wolfline – Transit Visualization System". Last retrieved: 3/26/11. http://ncsu.transloc.com

Appendix A (Figures)

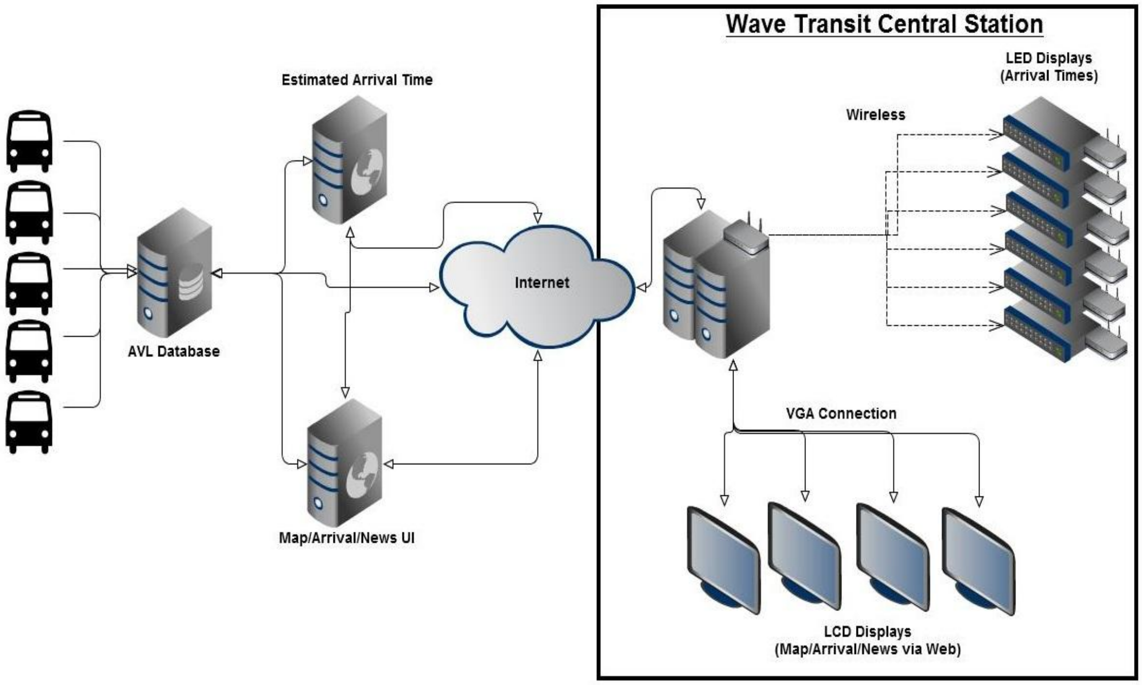


Figure 1: **Implementation Diagram**

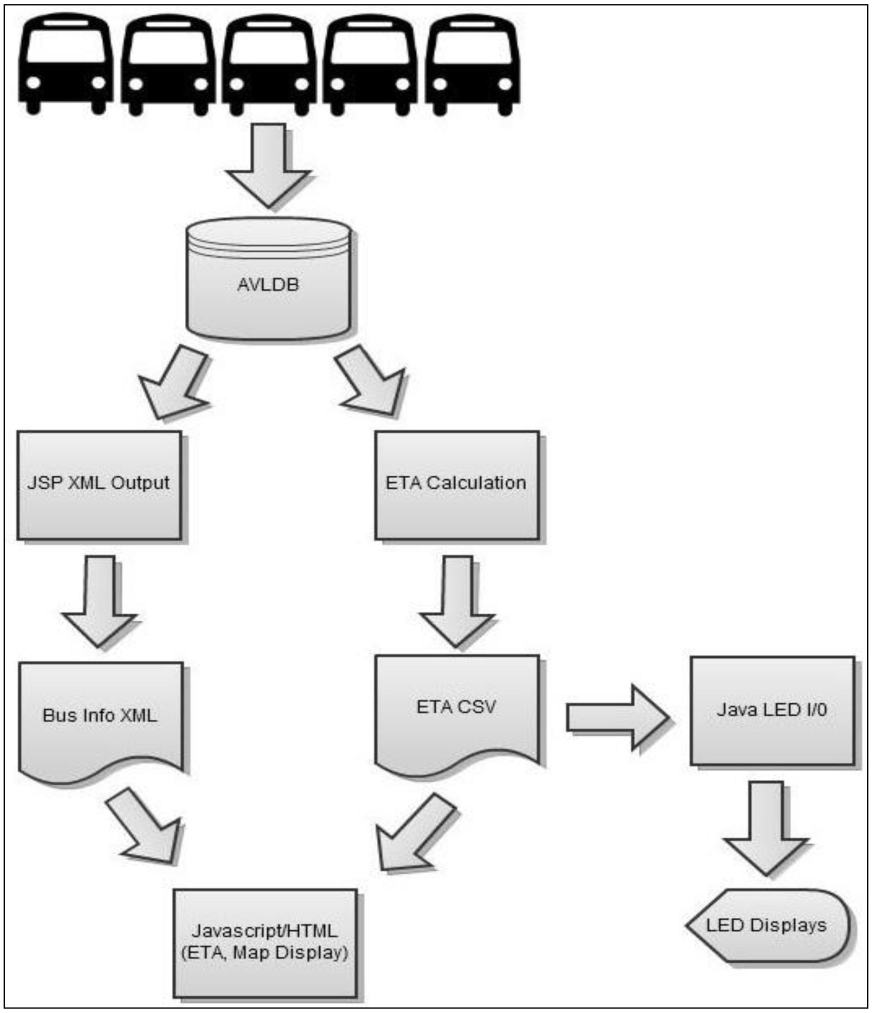
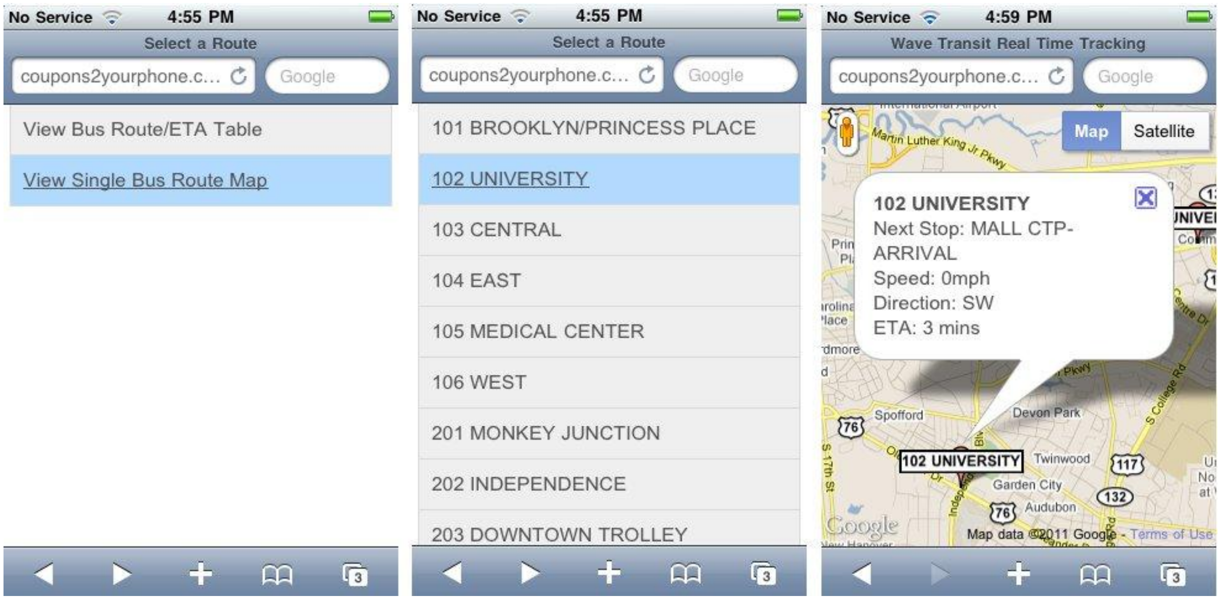


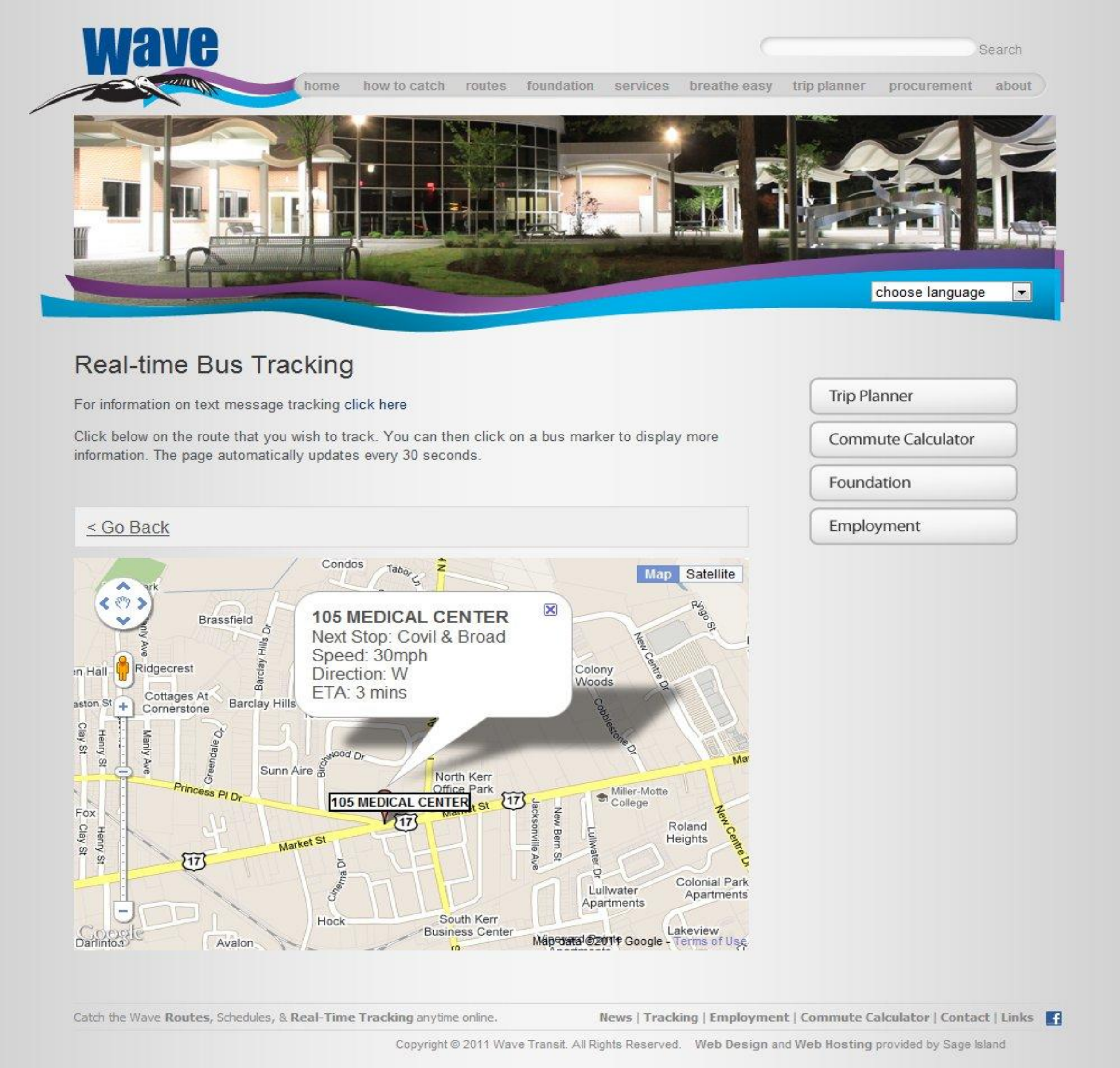
Figure 2 – Data Flow Diagram



101 BROOKLYN/PRINCESS PLACE
102 UNIVERSITY
103 CENTRAL
104 EAST
105 MEDICAL CENTER
106 WEST
201 MONKEY JUNCTION
202 INDEPENDENCE
203 DOWNTOWN TROLLEY

102 UNIVERSITY
Next Stop: MALL CTP-ARRIVAL
Speed: 0mph
Direction: SW
ETA: 3 mins

Figure 5 – Sample Screens from Selecting a Single Route to ETA Information for that Route



Real-time Bus Tracking

For information on text message tracking click here

Click below on the route that you wish to track. You can then click on a bus marker to display more information. The page automatically updates every 30 seconds.

< Go Back

105 MEDICAL CENTER
Next Stop: Covil & Broad
Speed: 30mph
Direction: W
ETA: 3 mins

Catch the Wave **Routes**, Schedules, & **Real-Time Tracking** anytime online.

News | Tracking | Employment | Commute Calculator | Contact | Links

Copyright © 2011 Wave Transit. All Rights Reserved. Web Design and Web Hosting provided by Sage Island

Figure 6 – Sample Full Screen Webpage