Updated Bus Routes and Text Messaging System for San Luis Obispo Public Transit System Web Application

Matthew Bojorquez

Table of Contents

Table of Figures

Abstract

As gas prices continue to rise and people become more aware of their carbon footprint, public transportation becomes more economically, environmentally, and sustainably viable. Currently, bus route timetables are available inside buses and at the transit center, but many users have little knowledge of how to read them. Therefore, to entice more people into using public transportation, we must make riding the buses less of a hassle for users. The Efficient Deployment of Advanced Public Transportation Systems (EDAPTS) system attempts to accomplish this. Building upon a web application designed by Jeff Tikkanen, a former Cal Poly student, that places bus information online, the bus routes were updated to incorporate new stops and to incorporate changes to the paths the buses travel. The current system proves useful for users with internet enabled smart phones, but for the rest of the users, the system does not meet their needs while they wait at bus stops for the arrival of a particular bus. The addition of a text messaging system was incorporated into the system to allow any user with a text messaging capable phone to retrieve a bus' current status information, thereby ensuring easier access to the information. This enables the majority of Cal Poly students to have easy access to the status of buses without the need for calling the transit center to enquire of a bus' location.

1: Introduction

In 2009, Jeff Tikkanen, a student at California Polytechnic State University San Luis Obispo, developed the Bus Location Prediction Web Application[1]. This system shows users the location of San Luis Obispo Transit buses in the city.

The system consists of two parts: A text-based version and a graphical version. The text-based version is intended for users with internet enabled smart phones, as the graphical version may be too intensive to load on the phones. The graphical version is intended for home computer use.

2: Project Overview and Goals

The goal of my project was to update the bus routes on the EDAPTS project and to provide a way for users to check the location of buses when they do not have access to an internet-enabled smart phone.

Currently the only way of knowing the bus' location is to call the transit office and inquire the status of each individual bus, unless you happen to find yourself at either the Downtown Transit Center or Kennedy library, which feature real time bus status signage. However, if you need to find a bus route to get to a certain location and want to know when the bus will arrive, this method proves inefficient. With riders better able to access information on the bus routes, I hope to make the buses more user-friendly.

2.1: Design Requirements

Marketing Reqs	Engineering Requirement	Justification
1, 3	The graphical web page must load completely in 20 seconds.	This was an engineering requirement of the previous EDAPTS project. My additions to the project should have no effect on this time.
1, 3	The text-based version must completely load in 5 seconds.	Even on a slow 14.4 kbs modem, a full page of text can load in 6 seconds. The EDAPTS mobile page has considerably less text.
4	The total cost of additions to the project (not including labor estimates) should not cost more than $150.	As of right now, there is no budget needed for the project. However, if time permits, I will begin work on the GPS receiver.
1, 2	The system will be operational from 6 AM until 11 PM, at least.	The system must be operational during the time the buses are in service. This is approximately during the times listed at left.

2.2: Design Specifications

1. The design must use as much of the existing framework as possible.
2. The system must be reliable and have little downtime.
3. The webpage must load in a reasonable amount of time.
4. The project uses a minimum budget.

2.3 Project Specifications

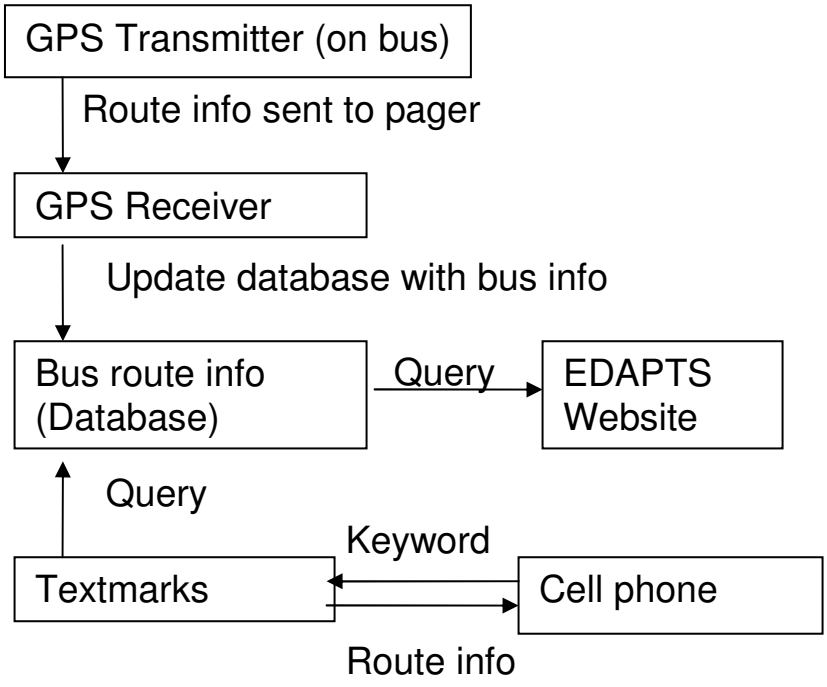

Figure 1: Block diagram of the system

2.4: Project Planning

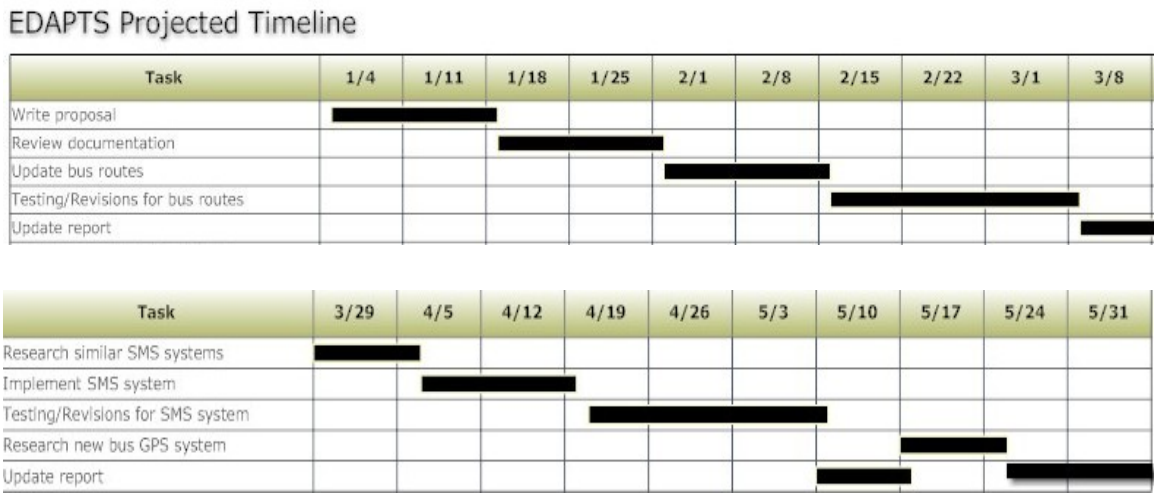

Figure 2: Gantt charts showing projected progress on the EDAPTS project over Winter and Spring 2010.

2.5: The EDAPTS System

The EDAPTS system shows the location of the 7 different buses that SLO Transit operates (bus routes 1-5, 6A and 6B). These bus routes are labeled Trip1, Trip2, Trip3, Trip41, Trip42, Trip51, Trip52, Trip61, and Trip62. The last two correspond to route 6A and route 6B. Trip41 and Trip42 correspond to route 4 bus 1 and route 4 bus 2. The same goes for Trip51 and Trip52.

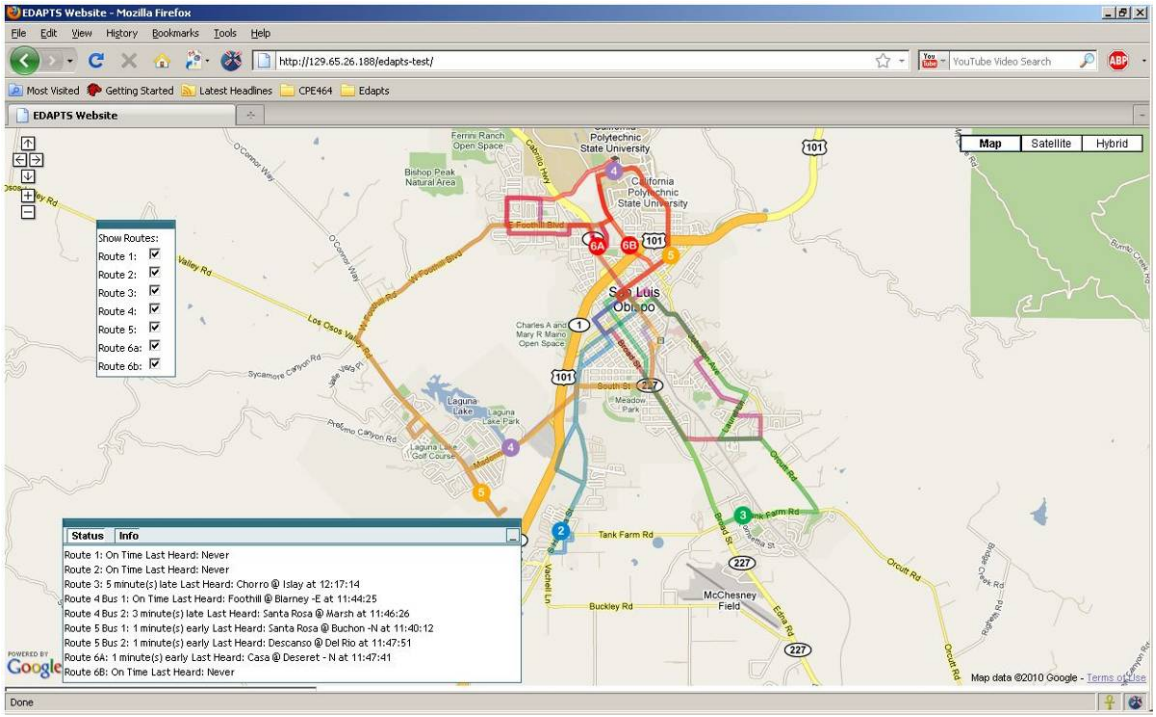

Figure 3: Screenshot showing the location of buses on the EDAPTS-test page.

A few points worth mentioning about the screenshot above:

- Each of the routes has a unique color (aside from the 6A and 6B, both shown in a pinkish color).

- Routes can be shown or hidden by clicking on the checkbox of the corresponding route in the Show Routes window (seen near upper left corner).

- Detailed information about the status of the routes can be seen in the box at bottom left. This table currently displays incorrect information due to changes in the routes, schedules, and GPS system of the buses made after the completion of Jeff Tikkanen's project.

The EDAPTS system also has a text based version of the application as shown below.

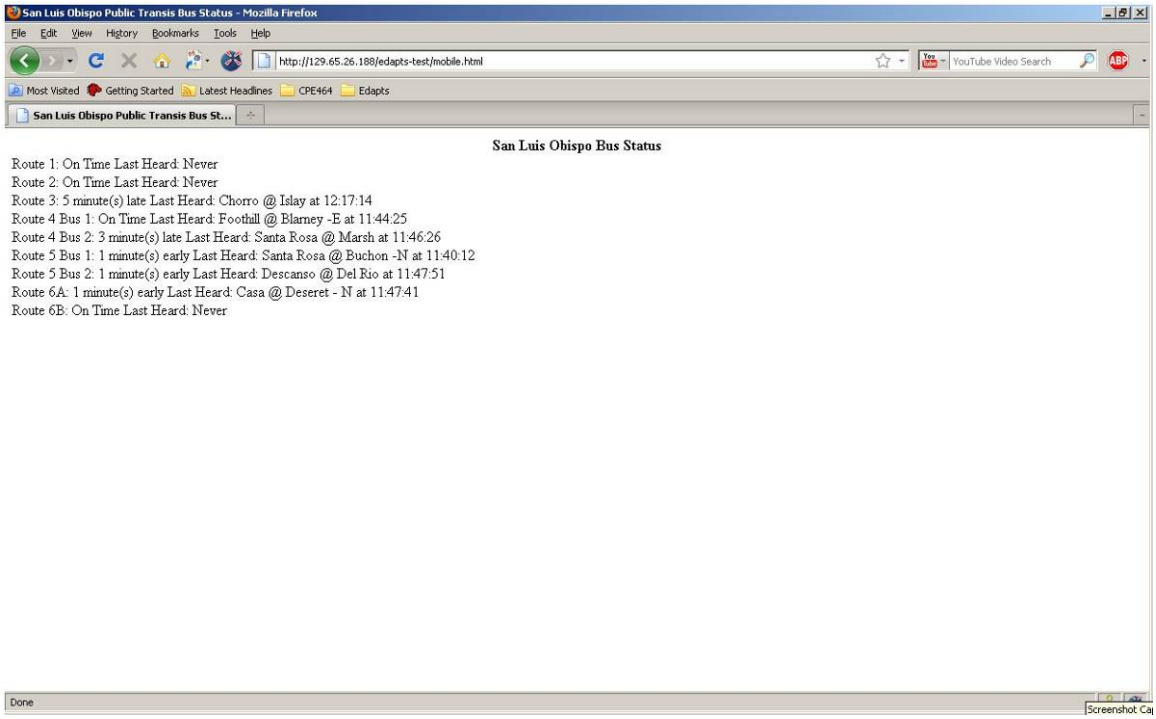

Figure 4: The EDAPTS mobile page that displays current route information.

The EDAPTS mobile page displays the same information as the status box on the bottom left corner of the main application (see figure 3) but does so in a simplistic easy-to-read manner that eliminates all of the graphical information.

3: Project Implementation

3.1: Remapping of bus routes

3.1.1: Polyline generator tool

The encoded polyline generator tool was the basis for the remapping of bus routes. Jeff Tikkanen designed the application and used it when he originally mapped the bus routes on the EDAPTS project. Figures 5 and 6 show screenshots of the polyline generator tool.

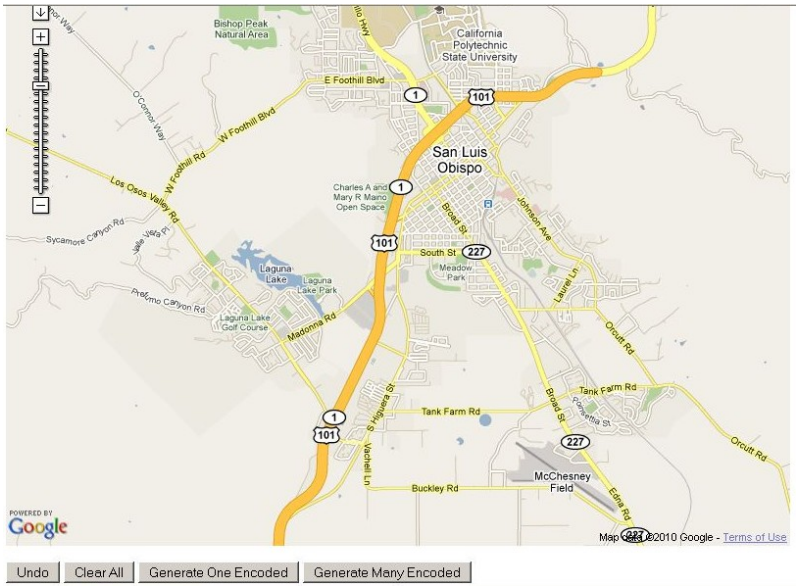

Figure 5: Top half of the polyline generator tool. The user can click on the map to generate the bus routes.

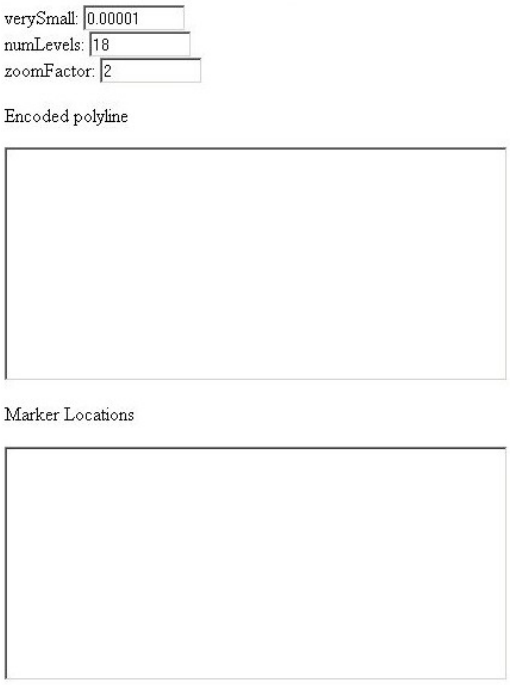

Figure 6: Polyline Generator application output textboxes

The polyline encoding tool works by taking the GPS coordinates of every point the user clicks on the map and converting them into "a series of points… using a compressed format of ASCII [text] characters."[2]

As seen by the screenshot, there are 3 main sections that compose this application: the map and associated input buttons and text boxes, Encoded Polyline text box, and marker locations box.

The map and associated input buttons and text boxes part consists of the map from Google Maps, 4 buttons under the map, and the 3 text boxes under it. On the map, the user can click along streets to generate markers or waypoints (I will use these interchangeably) to create a route for the buses. The 4 buttons below the map are used to manipulate the markers the user clicked as well as generate the string of text characters that represent the line. I will explain more about these buttons below:

- Undo: The undo button removes the last marker the user added to the map, as opposed to the undo function in Microsoft Word and other text editors. If the user accidentally removes a marker by clicking the undo button, clicking the undo button again will not add the marker back onto the map. Instead, the user will have lost the last 2 markers added to the map.

- Clear All: The clear all button will erase the whole route drawn on the map. This button is the equivalent of clicking the undo button n times, where n is the number of markers on the map.

- Generate One Encoded: Once the route has been mapped, the user clicks this button to generate a string of text characters that represent the line as seen on the map. This will generate text in the Encoded Polyline text box and Marker Location text box. I will elaborate more on these later.

- Generate Many Encoded: Do not use this button to generate the bus routes. Clicking this button will essentially treat each individual line segment as its own polyline. Every line segment will get a separate encoded polyline in the Encoded Polyline text box and the distance of each stop in the Marker Location text box (both Encoded Polyline text box and Marker Location text box are describe in more detail below) will be undefined.

The input textboxes verySmall, numLevels, and zoomFactor specify parameters for how to draw the polyline representing the bus route. Using small values for verySmall can help fix the problem of disappearing polylines. NumLevels indicates how many different zoom levels associate with the polyline. For example, if a map has 18 zoom levels and the polyline has 9 zoom levels, then the level of detail of the line should change every other zoom level. The routes I drew use the default values for these parameters.

The Encoded Polyline text box is the most important for EDAPTS. Upon clicking either of the generate encoded polyline buttons, the application will generate text in this box to represent the route or line segments along the route (depending on which button is pressed). The EDAPTS project uses the "points" and "levels" to correctly render the bus routes.

When the user clicks a button to generate a polyline, information shows up in the Marker Location text box. This information tells the user the distances along a route for each stop, in meters (starting from the first distance). If a user chooses to generate many encoded polylines, the distances in this box will all be undefined.

The EDAPTS project uses the distances in the Marker Locations box to tell the buses where to stop along the route. For the motion algorithm to work correctly, these values must be accurate. There exists a bug with the polyline generator application which requires that all stops be non-transparent otherwise the distance will not correctly compute for all of the stops (the application reports the distance as undefined). I recommend plotting the route up until a certain stop, clicking the generate one encoded button, and then manually recording the distance.

3.1.2: How to create bus routes

The user can create the bus routes by simply clicking once on the map for each waypoint (along the bus route (for example, at intersections and at stops). When a user clicks the "Generate One Encoded" button on the application's web page, an encoded polyline is created based on the waypoints the user clicked.

As stated above, the user should copy down the strings of text labeled points and levels in the Encoded Polyline text box as well as the distances for each stop in the Marker Locations text box. Using these, the user can construct SQL statements to update the routelines, stops, or trip tables in the EDAPTS database.

3.1.3: Updating bus routes

Updates to bus routes take a few steps to accomplish. First, the route has to be redrawn using the Polyline Generator application (described earlier in section 3.1.1). Next, for any new stops not currently present in the database (i.e. stops not used by any of the other bus routes), the user must add them into the stops table. Then, the user must change the corresponding Trip table to include information about the new route.

To have the route shown on the EDAPTS page, the developer must write a query to update the corresponding entry for the route in the Routelines table, using the points and levels copied from the Polyline Generator application as the GPolyEncoded and Levels columns, respectively.

A new stop can be added into the Stops table by supplying a name, latitude and longitude, and the corresponding stop number for the stop on each of the routes (i.e. for the 4^{th} stop on the 6A bus, the "Trip61" column should have a 4 in it). For all stops not on a particular route (i.e. 6A bus does not stop at the Downtown Transit center), the developer should put a 0 (the number zero) in the corresponding column. Any non-zero values result in the stop drawn on the map at the specified longitude and latitude in the color that identifies the route.

The next step involves modifying the route information in the corresponding trip table. In order to write a correct query for this table, the user must know the time after the hour that the bus arrives at the stop, the distance of the stop along the line (taken from the Marker Location text box when the line is generated), global stop identification number (the id column from the Stops table), stop number (like the 4^{th} stop example in the previous paragraph), and the time in seconds the bus waits at the stop before departing. Note that for the time the bus arrives after the hour, the code will automatically adjust the times for buses that run on a 30 minutes schedule (as in the route 6A and 6B buses) or for buses that are offset by 30 minutes (as in the route 4 and 5 buses).

3.2: Text messaging system

3.2.1 Textmarks

"TextMarks is an early stage company focused on making text messaging more powerful and more accessible. We empower individuals and organizations to 'text message enable' their information and their web pages -- and to do so in a matter of minutes."[3]

I needed a system that would allow the automatic sending of alerts to users on the fly. Textmarks offered an easy to use solution that enabled the sending of dynamic text message alerts to end users.

In order to sign up for a free keyword, the developer needs a text-messaging capable phone. Once the number has been verified, users may begin setting up their alert system.

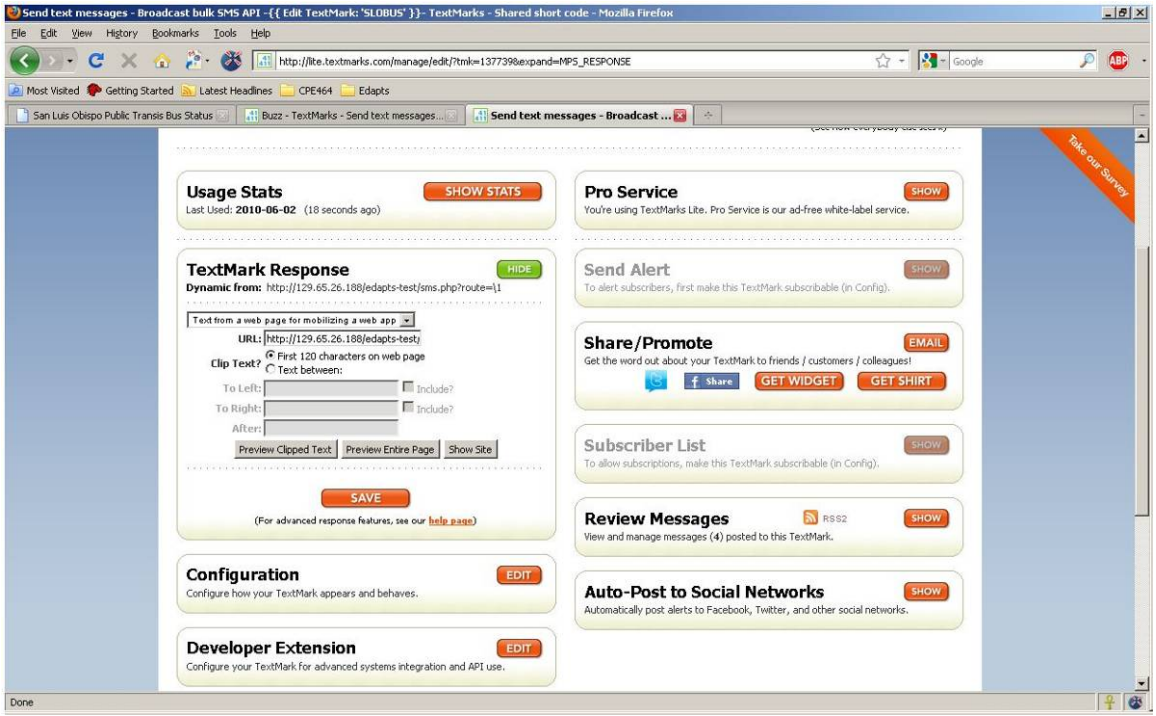

Figure 7: The Textmarks configuration page.

The next step requires the developer to select a keyword so Textmarks can handle the text messages appropriately. For this project, I chose the keyword "slobus."

Textmarks allows users to set up a subscriber based system (similar to the Cal Poly Emergency Text Message System) or set up a response based system (i.e. users get information when they text a keyword to 41411). A response based system provides the perfect set up for this project.

On the Textmarks configuration page, I set up the system to call a PHP script I wrote whenever a user texts "slobus" to 41411. The link has the format: http://129.65.26.188/edapts-test/sms.php?route=\1. The PHP parameter at the end of the link ("route=\1") requires some explanation. The Textmarks system parses additional arguments sent along with "slobus" and the developer can access these arguments using \n where n is a 1 digit number from 1-9. The script file uses this parameter to determine which route's information to sen back to the user.

For example, when a user wants information regarding route 6A, they can send "slobus 6A" to 41411. The Textmarks system knows to send "6A" to a PHP script titled sms.php on the EDAPTS server, which then processes the "6A" and sends back information related to the route 6A bus. The information the user receives in the text message mirrors that on the EDAPTS mobile website. The text message response system does not work correctly because it relies on the pager receiving GPS position updates from the buses. The text messaging system should work correctly once a future student integrates a new pager into the system.

3.2.2: Sms.php

The script file that parses the text message is really simplistic, yet powerful. First, the route parameter gets retrieved from the dynamic link and the script does some basic error checking. If the script deems the parameter valid, the script calls the processRoute() function from stats.php to print the route information. The EDAPTS mobile webpage uses this same function to print a given route's status. If the script deems the parameter invalid, the script sends a helpful error message back to the user. Also note that any function call that prints to the screen (i.e. print and echo) send data to the Textmarks system to send via text message to the user.

Note that sms.php also includes an API key, similar to what Google Maps requires. The text messaging system did not require any of the advanced functionality provided by the developer API, but I included the information should future developers decide to incorporate additional functionality. For more information on the API, developers can go to http://lite.textmarks.com/dev/docs/api2/.

4: Project Results and Problems Encountered

4.1: Bus Routes

Figure 8 below shows the new routes on Cal Poly's campus. Before the University Union construction, routes would travel along South Perimeter and stop in front of Robert E. Mott Gym. However, routes now travel along North Perimeter and stop at the Kennedy Library.

Figure 8: Screenshot of EDAPTS site showing Cal Poly campus

However, some issues exist in the final design.

- Even though correctly mapped and copied over to the EDAPTS database, part of the 6A route does not render. This part occurs between O'Neill Green and the Kennedy Library stop on the Cal Poly campus. After experimenting with many different solutions, I could not find one that would correctly draw in the missing line segment. However, the 4, 5, and 6B routes which all use California Boulevard to either enter or leave the Cal Poly campus render perfectly on the EDAPTS site. Even though that part of the route did not render, the bus has no problem completing the route and seems to follow an invisible line along the streets until it reaches Kennedy Library.

Figure 9: Route 6A not correctly rendering

- At the Promenade (Madonna Shopping Center) stop, Google Maps utilizes an incomplete map which required a workaround. The Mercado and Dalidio streets show up on the map, yet the street that intersects these two is absent. The Promenade stop is shown along Mercado Street, however the bus route does not go through it due to the absence of the cross street. Currently, the bus appears to stop along the Madonna Road. This was the final design choice that Jeff Tikkanen adopted for the EDAPTS system so I have kept it in my final design. The bus appears to move slower due to the timing of the stops, but the arrival times at the stops is far more important for the motion algorithm than the approximated travel between stops. There is no best choice for this, as any solution would provide false information to the end user.

Figure 10: Madonna Shopping Center – Buses stop at intersection of Mercado and Madonna

- Google Maps uses an incomplete map at Auto Park Way and does not show the loop at the end of the street that the buses use to essentially complete a u-turn and backtrack along part of Los Osos Valley Road. This causes only a slight inaccuracy to the correctness of the system and originally I thought this would cause a problem for the timing of the bus stops. However, the motion algorithm used by the EDAPTS system takes into account the distance along the line segment to travel and the time to reach the next stop. The bus appears to move slightly slower (as it does between the Madonna Road at Madonna Plaza and Promenade stops – see the above bullet) but the bus gets to the LOVR at Auto Park Way at the correct time.

Figure 11: Auto Park Way – the 4 and 5 complete a 180° turn at the end of the street instead of driving around a loop (not shown on Google Maps

4.2: Text messaging system

The text messaging system consists of one PHP script that I wrote: sms.php (described in detail in section 3.2.2 and included in entirety in section C.2). Upon visual inspection, the script should work exactly as intended but I have not had success in attempting to test it. The Textmarks site does not have the correct permissions to to access the file (only administrators have privileges to access the file on the server). Working with the Textmarks API, I found no way for the site to provide authentication to access the file but this would not be ideal anyways, as access to the server could be compromised. I attempted to change the permissions of sms.php to allow all users read access to the file, but I did not have sufficient privileges myself to accomplish this. I sent an email to the EE Server Administrator to have the permissions changed but never heard back from him.

However, with the above problem fixed, I presume the system would work exactly as below as seen by the text messaging simulator found on the Textmarks site (http://lite.textmarks.com/info/overview/).

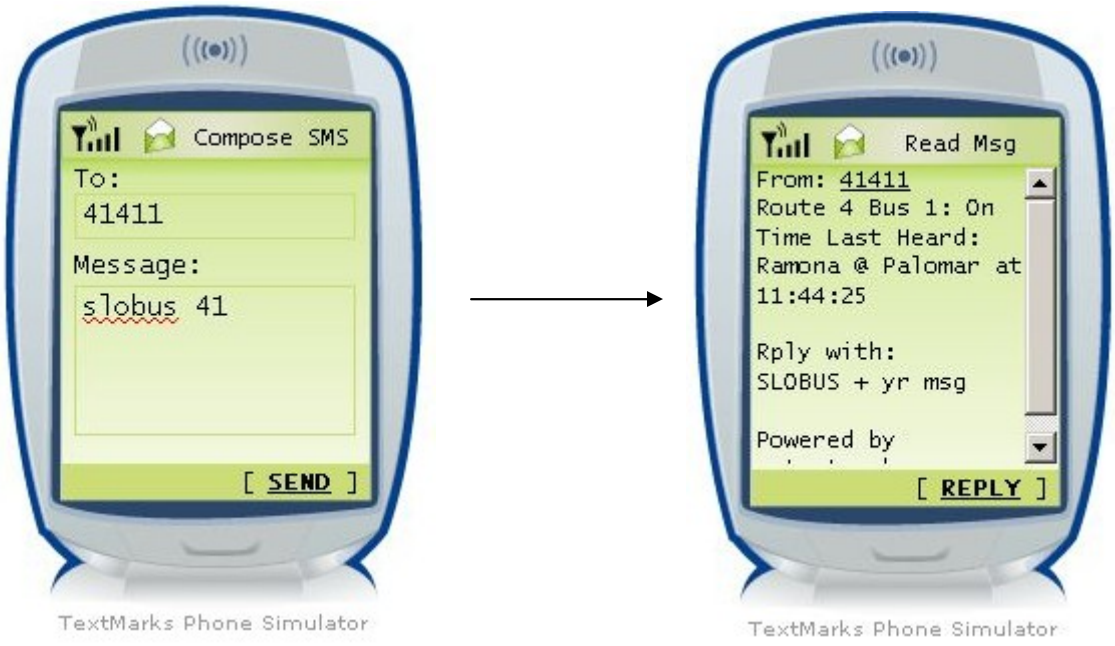

Figure 12: Theoretical demonstration of text messaging system alerting a rider about the status of bus 1 along route 4.

5: Conclusion and Recommendations

This project successfully implemented changes to the EDAPTS system to account for changes in the 4, 5, 6A, and 6B bus routes and with the addition of the text messaging system, more users can have near instant access to the bus statuses, specifically those users without internet enabled smartphones.

Sometime after Jeff completed the initial design of the EDAPTS site, SLO Transit purchased new buses which make use of a different GPS system. The current pager in use by the EDAPTS system does not work with the new GPS systems on the buses so a future project would require funding a new pager and integrating it into the system. Jeff's code for the current pager, with some tweaks, should fix this issue.

Currently, the EDAPTS website shows the buses running all day and night, which could potentially give riders incorrect information. Each bus starts at a specific time, at a specific stop every morning and ends at a specific time and stop every night. The system should not show buses running at 2 AM in the morning when all buses cease operation by 11 PM every night. Another, more difficult update, would change the routes on the EDAPTS site to accommodate sporting events (i.e. buses that have stops on the Cal Poly campus do not travel down the length of California Boulevard during a home football game).

Another update involves fixing the bug in the Polyline Generator application I used to redraw the bus routes. As noted in section 3.1.1, the application can make waypoints transparent with the end goal of hiding them from the distance calculations shown in the Marker Locations textbox. The application hides the transparent markers but instead fails to calculate the correct distance for non-transparent markers, usually defaulting to a distance of "undefined." To assist in future route edits, one can also add an encoded polyline decoder. This would essentially use Google's published algorithm[4] to decrypt the string of characters and place markers at each GPS coordinate. The developer could then take advantage of the points already plotted and change them with little hassle.

Since Google Maps does not show the cross street between Dalidio and El Mercado, another possible change to the system would incorporate a popup or information box to alert users about the Promenade stop. It should concisely tell users that while the buses appear to stop on Madonna Road, the buses stop at the location indicated on the map.

Appendix A: ABET Senior Project Analysis

Economic

California Polytechnic State University, San Luis Obispo subsidizes San Luis Obispo (SLO) Transit, the bus service for the city of San Luis Obispo, to enable its students and faculty free transit throughout the city by simply showing their identification card. The EDAPTS project aims to make SLO Transit easier to use, which would hopefully cause more students to ride the bus instead of drive to school. The students would save money on gasoline and maintenance of their cars as well as money on the $230 per year parking permit.

Environmental

Mass transit exists to reduce the carbon footprint caused by private cars, SUVs, and light trucks. According to a 2002 study done by the Brookings Institution and the American Enterprise Institute, private vehicles emit about 95 percent more carbon monoxide, 92 percent more volatile organic compounds and about twice as much carbon dioxide and nitrogen oxide than public vehicles for every passenger mile traveled[5]. Clearly, encouraging the use of SLO Transit will have a great positive impact on the environment.

Sustainability

Sustainability will have a big role in my project. The server code runs on an already in use server in the Electrical Engineering department. This reduces power use by having the code run on an existing server instead of running on a dedicated server that could possibly sit idle for a vast majority of time. Also, by reducing carbon dioxide emissions (as stated in the environmental section above), we are better able to meet our current needs without compromising the ability of future generations to meet their own needs[6].

Manufacturability

This project required very little manufacturing in the traditional sense and does not require manufacturing in physical quantity for individual users. The text messaging alert system code is written in PHP, a very common scripting language. PHP comprises a significant portion of the EDAPTS website code so it seemed a natural choice to use.

Ethical

This project intends to help users by making bus riding easier for everyone, specifically people who can not decipher the time tables for the buses or who may not have access to a time table. This project, however, neglects two specific types of users: those with no internet connection and those who do not speak English. Although Cal Poly students (who generally do not fall into either category) comprise the target demographic of the project, this may pose a problem for other potential users. Elderly and low-income users mostly comprise those with no internet connection. California contains a significant Spanish speaking population. The graphical design attempts to accommodate non English speaking users by visually displaying information. The text based version is designed to require little working knowledge of the English language.

Health and safety

Encouraging more students to ride the buses will result in fewer cars on the road and a reduced chance of accidents. Fewer cars on the road also leads to less air pollution thereby benefiting the air quality of San Luis Obispo.

Social

The project is mainly intended for use by the students of Cal Poly, although other demographics use the bus system. As stated before, the web application may not serve all bus riders (specifically those with no internet access and those who do not speak English), but most Cal Poly students do not fall into either of these categories.

Political

This project organizes data from a local government agency into an easily readable format. The maintenance of the website should require minimal resources, especially considering the budget cuts affecting the state of California.

Appendix B: Miscellaneous

B.1: Gantt Chart

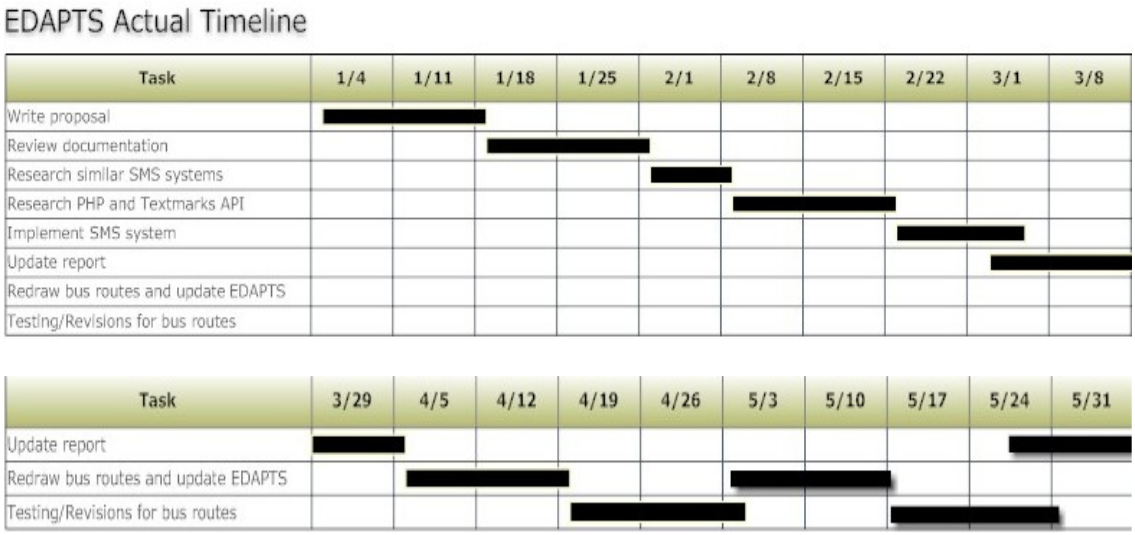

Figure 13: Actual timeline of progress on EDAPTS project.

The timeline for the second quarter of the project seems to be redundant, however the routes proved more troublesome than initially assumed. From April 5 until May 3, I spent a lot of time carefully drawing the routes but could not get the distances worked out correctly. Coincidentally, a lot of time during this period dealt with trial and error of distances but this technique ultimately proved fruitless.

Around May 3, I discovered the error in the polyline generator application (more information regarding this in section 3.1.1) which enabled me to use exact distances. This required redrawing each of the routes but made it immensely easier to write SQL queries to update the EDAPTS database. During this time I also added new stops (such as Kennedy Library) and changed the GPS coordinates of others (such as the Ramona at Tassajara stop).

B.2: Estimate of Parts and Costs

I spent approximately 8 hours a week working on this project over the course of 20 weeks which amounts to roughly 160 hours of work on this project. Assuming an unburdened engineering rate of $55/hour, my work on this project cost approximately $8,800.

My additions to the project did not require any additional funding beyond the $195.62 that was spent on the pager.

Appendix C: Bibliography

[1] J. Tikkanen. "Bus Location Prediction Web Application for San Luis Obispo Public Transit." San Luis Obispo CA: Kennedy Library, Published: 2009. [Microfiche] Available: Kennedy Library, California Polytechnic State University, San Luis Obispo. [Accessed January 13, 2010]

[2] "Google Maps JavaScript API V2 Overlays - Google Maps JavaScript API V2 - Google Code." *Google Code,* Published: 2010. [Online] Available: http://code.google.com/apis/maps/documentation/javascript/v2/overlays.html#Encoded_Polylines. [Accessed May 28, 2010]

[3] "About Us - TextMarks - Send text messages - Broadcast bulk SMS API," *TextMarks - Send text messages - Broadcast bulk SMS API*, Published: 2010. [Online] Available: http://www.textmarks.com/front/about/. [Accessed: June 9 2010]

[4] "Encoded Polyline Algorithm Format - Google Maps API Family - Google Code." Google Code, Published: 2010. [Online] Available: http://code.google.com/apis/maps/documentation/utilities/polylinealgorithm.html. [Accessed May 28, 2010]

[5] L. Layton, "Study Lists Mass Transit Benefits," *Washington Post*, Published: July 16, 2002. [Online] Available: http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=&contentId=A16228-2002Jul16&notFound=true. [Accessed: January 5 2010]

[6] "A/RES/42/187 Report of the World Commission on Environment and Development," *Welcome to the United Nations: It's Your World*, Published: December 11, 1987. [Online] Available: http://www.un.org/documents/ga/res/42/ares42-187.htm. [Accessed: January 15, 2010]

Appendix D: Code

D.1 SQL Scripts

D.1.1: Route 4

```
# Redraw route 4
update routelines set
gpolyencoded="kejvEzzm_Vr@o@{CqGlNyLfL}J`DlGzCkCxCwB^Ot@Q~ImAjHC|B
F~Id@|ANh@RT`@Pl@DrD@~dACfG|DDhB`BLD~GzK~HrL~@lBhS~Z`@ZnBzEzE
xNjQpj@fMqIfJeHdKiJXi@rBiB{CqGzCpGmB~Aq@VaKvIgOjLqDjCkBjAURCPiGr
EaBlBiRdViDhEyDhEOLtJdNaAfAs@j@{BlA}@~@iBmBaEaGi@s@sMpPSZG`@a
DtDyRlPsGxFgBvAoE|DwGjFoAnAoAxAY^wZqOwAw@{@y@q@cAc@kAcBaFsGc
Su@{A_FsGyCmEa@}@qEoOqAeEmHkS}A}DcC}DgDcGcBsEy@cBuCkDgDoDk@
{@W}@MmAAqGB{FtB?bCB?iRFqFEcDAyHD?E?BqINoAD}AAqLiGBFkJDoQEeA
QiAmD{M{DbDyAt@yCbA}Bf@mMbCaEd@wBH}BmRkAaLa@}EDkAN{@Zu@`
@i@tBgBj@a@j@Y~@On@Ab@Fr@?Fe@j@eC|CsBVYpIgNzAyBr@a@`@I|@GRJ
hKDtFErKLhCGz@[lByAb_@bu@nDtHpBgB",
levels="PCF?GFA@D?E@C?AI@BC?GC@E@AAAFA?HAC@AJFFAABA@A@AE
@?A?GDG@AFA?GB@AD???AB@?KCBAF??DAA@D?B@E?B@C@BAHA?E?E
@@?A@@A@AFF?@D@GC@D?B@H@BAI@C@@DAB@?E?C@C?BG@BA@@
@DAH@EP" where name="trip42";

# Delete all data from trip42 table
delete from trip42;

# Route 4
insert into trip42 values (41, 528.2, 438, 2, 10);
insert into trip42 values (41.5, 759.3, 435, 3, 10);
insert into trip42 values (43, 1331.7, 431, 4, 10);
insert into trip42 values (44, 1620.6, 433, 5, 10);
insert into trip42 values (45, 2068.6, 441, 6, 10);
insert into trip42 values (46, 2456.7, 444, 7, 10);
insert into trip42 values (47, 3087.4, 447, 8, 10);
insert into trip42 values (50.75, 4032.3, 395, 9, 10);
insert into trip42 values (52, 4314.1, 427, 10, 10);
insert into trip42 values (54, 4914.8, 397, 11, 10);
insert into trip42 values (56, 5684.0, 389, 12, 10);
insert into trip42 values (58, 6233.8, 384, 13, 10);
insert into trip42 values (0.5, 6950.8, 381, 14, 10);
insert into trip42 values (4, 7925.3, 387, 15, 10);
insert into trip42 values (4.5, 8263.4, 390, 16, 10);
insert into trip42 values (5, 8640.3, 385, 17, 10);
insert into trip42 values (5.3, 9090.4, 423, 18, 10);
insert into trip42 values (5.6, 9284.7, 335, 19, 10);
```

```
insert into trip42 values (6, 9538.4, 336, 20, 10);
insert into trip42 values (7, 9968.0, 382, 21, 10);
insert into trip42 values (7.5, 10931.2, 392, 22, 10);
insert into trip42 values (9, 12105.4, 337, 23, 10);
insert into trip42 values (11, 14711.6, 376, 24, 10);
insert into trip42 values (13, 15279.6, 429, 25, 10);
insert into trip42 values (14, 15643.6, 428, 26, 10);
insert into trip42 values (16, 16144.3, 341, 27, 10);
insert into trip42 values (18, 16480.3, 339, 28, 10);
insert into trip42 values (23, 17797.6, 455, 29, 120);
insert into trip42 values (25.5, 18481.5, 457, 30, 10);
insert into trip42 values (28, 19352.9, 351, 31, 10);
insert into trip42 values (29.5, 19714.8, 352, 32, 10);
insert into trip42 values (31, 19944.4, 408, 33, 10);
insert into trip42 values (32, 20294.8, 404, 34, 10);
insert into trip42 values (33, 20598.6, 407, 35, 10);
insert into trip42 values (34, 20875.0, 410, 36, 10);
insert into trip42 values (35, 21124.5, 418, 1, 300);

# Set all values in stops table to be 0
update stops set trip42=0 where trip42>0;

# Update all stops in stop table
update stops set trip42=2 where id=438;
update stops set trip42=3 where id=435;
update stops set trip42=4 where id=431;
update stops set trip42=5 where id=433;
update stops set trip42=6 where id=441;
update stops set trip42=7 where id=444;
update stops set trip42=8 where id=447;
update stops set trip42=9 where id=395;
update stops set trip42=10 where id=427;
update stops set trip42=11 where id=397;
update stops set trip42=12 where id=389;
update stops set trip42=13 where id=384;
update stops set trip42=14 where id=381;
update stops set trip42=15 where id=387;
update stops set trip42=16 where id=390;
update stops set trip42=17 where id=385;
update stops set trip42=18 where id=423;
update stops set trip42=19 where id=335;
update stops set trip42=20 where id=336;
update stops set trip42=21 where id=382;
update stops set trip42=22 where id=392;
update stops set trip42=23 where id=337;
update stops set trip42=24 where id=376;
```

```
update stops set trip42=25 where id=429;
update stops set trip42=26 where id=428;
update stops set trip42=27 where id=341;
update stops set trip42=28 where id=339;
update stops set trip42=29 where id=455;
update stops set trip42=30 where id=457;
update stops set trip42=31 where id=351;
update stops set trip42=32 where id=352;
update stops set trip42=33 where id=408;
update stops set trip42=34 where id=404;
update stops set trip42=35 where id=407;
update stops set trip42=36 where id=410;
update stops set trip42=1 where id=418;

# Set all values in stops table to be 0
update stops set trip41=0 where trip41>0;

# Update all stops in stop table
update stops set trip41=2 where id=438;
update stops set trip41=3 where id=435;
update stops set trip41=4 where id=431;
update stops set trip41=5 where id=433;
update stops set trip41=6 where id=441;
update stops set trip41=7 where id=444;
update stops set trip41=8 where id=447;
update stops set trip41=9 where id=395;
update stops set trip41=10 where id=427;
update stops set trip41=11 where id=397;
update stops set trip41=12 where id=389;
update stops set trip41=13 where id=384;
update stops set trip41=14 where id=381;
update stops set trip41=15 where id=387;
update stops set trip41=16 where id=390;
update stops set trip41=17 where id=385;
update stops set trip41=18 where id=423;
update stops set trip41=19 where id=335;
update stops set trip41=20 where id=336;
update stops set trip41=21 where id=382;
update stops set trip41=22 where id=392;
update stops set trip41=23 where id=337;
update stops set trip41=24 where id=376;
update stops set trip41=25 where id=429;
update stops set trip41=26 where id=428;
update stops set trip41=27 where id=341;
update stops set trip41=28 where id=339;
update stops set trip41=29 where id=455;
```

```
update stops set trip41=30 where id=457;
update stops set trip41=31 where id=351;
update stops set trip41=32 where id=352;
update stops set trip41=33 where id=408;
update stops set trip41=34 where id=404;
update stops set trip41=35 where id=407;
update stops set trip41=36 where id=410;
update stops set trip41=1 where id=418;
```

D.1.2: Route 5

```
# Redraw route 5
update routelines set
gpolyencoded="ygjvE~|m_V`CsB_DmGaDdCuHgOsEkJuNiYa@{@kCnBo@HoCCyE
FsWMSNoAJu@\\m@h@wKlQgD|B?Ag@hBQnAgAMeBD}At@sCbCi@`AU~@E`Bt
BdTrB~PzCQbCYvRuDfC}@rBgA`DqChDfMXtBMpi@CvTDdEKxf@?vFPrBp@xAlI
nJbAvBxA~DjA|BbFjI~KtZtGxTzAdCbIrKn@zAdG`RdClHb@v@|A~Ab]~PxDiElGcF
rd@k`@vEeFn@w@dUqYrItLPZaB`BmBfAiA|@UXeBsB}E{GlGcIFSTg@bIuJ`IeKrJ
wLzBiBdDcCJDXQvLaInJmHjKqJn@}@vAmA{CuGzCtGcBtAs@ZqKdJuFhE_MhJq
BlA}CgKyKc]oFePgBkEG]yE}HmHsKgC{Dg@m@WIuO}UsCiFyFIBgG@u[CwN@g
OG{MKk@S]q@[iN{@}J@_JlA?w@|BsB\\GDSGOO?IHOG}E{CgDvC_GzEuQ|O}N
xL~CzGZW",
levels="PEEE???HBD@@CA@GAC?C@EACDCAIBAH@BDABGAE@@@BAHBC
@B@EBDAD@?AFBKBAD?AG?EGA?E@GA?A@?E?B@AAC@AJFFAAAAAH@
A@@F@?A@BBEG???CB@IBDCFCAAEA@AAG??@FCP" where name="trip5";

# Delete all data from trip52 table
delete from trip52;

# Route 5
insert into trip52 values (21.5, 418.8, 409, 2, 10);
insert into trip52 values (22, 643.7, 406, 3, 10);
insert into trip52 values (22.5, 846.9, 403, 4, 10);
insert into trip52 values (23.5, 1320.6, 405, 5, 10);
insert into trip52 values (24.5, 1555.4, 349, 6, 10);
insert into trip52 values (26, 1867.2, 350, 7, 10);
insert into trip52 values (28, 2752.4, 458, 8, 360);
insert into trip52 values (35, 3475.9, 456, 9, 10);
insert into trip52 values (39.5, 4692.5, 340, 10, 10);
insert into trip52 values (41, 5031.4, 348, 11, 10);
insert into trip52 values (42, 5436.7, 345, 12, 10);
insert into trip52 values (42.5, 5752.9, 342, 13, 10);
insert into trip52 values (43.5, 6421.7, 347, 14, 10);
insert into trip52 values (45.5, 8840.4, 338, 15, 10);
insert into trip52 values (47.5, 10014.8, 393, 16, 10);
insert into trip52 values (48.5, 11038.4, 383, 17, 10);
insert into trip52 values (49, 11867.0, 423, 18, 10);
insert into trip52 values (49.5, 12073.0, 335, 19, 10);
insert into trip52 values (50, 12325.0, 336, 20, 10);
insert into trip52 values (50.3, 12837.0, 386, 21, 10);
insert into trip52 values (50.6, 13089.0, 391, 22, 10);
insert into trip52 values (52, 13461.1, 388, 23, 10);
insert into trip52 values (52.75, 13645.6, 389, 24, 10);
insert into trip52 values (54, 14195.3, 384, 25, 10);
insert into trip52 values (56.5, 14908.3, 381, 26, 10);
```

```
insert into trip52 values (0, 16413.1, 396, 27, 10);
insert into trip52 values (2, 17009.1, 427, 28, 10);
insert into trip52 values (2.6, 17258.1, 394, 29, 10);
insert into trip52 values (5, 18256.8, 446, 30, 10);
insert into trip52 values (6, 18674.9, 442, 31, 10);
insert into trip52 values (6.5, 18903.1, 443, 32, 10);
insert into trip52 values (7.25, 19139.4, 445, 33, 10);
insert into trip52 values (9.0, 19699.7, 432, 34, 10);
insert into trip52 values (10, 20054.9, 430, 35, 10);
insert into trip52 values (11, 20378.7, 304, 36, 10);
insert into trip52 values (11.5, 20546.2, 437, 37, 10);
insert into trip52 values (12.5, 20720.8, 434, 38, 10);
insert into trip52 values (15, 21134.3, 436, 39, 10);
insert into trip52 values (17, 21657.7, 418, 1, 180);

# Delete all data from trip51 table
delete from trip51;

# Route 5
insert into trip51 values (21.5, 418.8, 409, 2, 10);
insert into trip51 values (22, 643.7, 406, 3, 10);
insert into trip51 values (22.5, 846.9, 403, 4, 10);
insert into trip51 values (23.5, 1320.6, 405, 5, 10);
insert into trip51 values (24.5, 1555.4, 349, 6, 10);
insert into trip51 values (26, 1867.2, 350, 7, 10);
insert into trip51 values (28, 2752.4, 458, 8, 360);
insert into trip51 values (35, 3475.9, 456, 9, 10);
insert into trip51 values (39.5, 4692.5, 340, 10, 10);
insert into trip51 values (41, 5031.4, 348, 11, 10);
insert into trip51 values (42, 5436.7, 345, 12, 10);
insert into trip51 values (42.5, 5752.9, 342, 13, 10);
insert into trip51 values (43.5, 6421.7, 347, 14, 10);
insert into trip51 values (45.5, 8840.4, 338, 15, 10);
insert into trip51 values (47.5, 10014.8, 393, 16, 10);
insert into trip51 values (48.5, 11038.4, 383, 17, 10);
insert into trip51 values (49, 11867.0, 423, 18, 10);
insert into trip51 values (49.5, 12073.0, 335, 19, 10);
insert into trip51 values (50, 12325.0, 336, 20, 10);
insert into trip51 values (50.3, 12837.0, 386, 21, 10);
insert into trip51 values (50.6, 13089.0, 391, 22, 10);
insert into trip51 values (52, 13461.1, 388, 23, 10);
insert into trip51 values (52.75, 13645.6, 389, 24, 10);
insert into trip51 values (54, 14195.3, 384, 25, 10);
insert into trip51 values (56.5, 14908.3, 381, 26, 10);
insert into trip51 values (0, 16413.1, 396, 27, 10);
insert into trip51 values (2, 17009.1, 427, 28, 10);
```

```
insert into trip51 values (2.6, 17258.1, 394, 29, 10);
insert into trip51 values (5, 18256.8, 446, 30, 10);
insert into trip51 values (6, 18674.9, 442, 31, 10);
insert into trip51 values (6.5, 18903.1, 443, 32, 10);
insert into trip51 values (7.25, 19139.4, 445, 33, 10);
insert into trip51 values (9.0, 19699.7, 432, 34, 10);
insert into trip51 values (10, 20054.9, 430, 35, 10);
insert into trip51 values (11, 20378.7, 304, 36, 10);
insert into trip51 values (11.5, 20546.2, 437, 37, 10);
insert into trip51 values (12.5, 20720.8, 434, 38, 10);
insert into trip51 values (15, 21134.3, 436, 39, 10);
insert into trip51 values (17, 21657.7, 418, 1, 180);

# VG's stop is at distance: 2516.4

# Set all values in stops table to be 0
update stops set trip52=0 where trip52>0;

# Update all stops in stop table
update stops set trip52=2 where id=409;
update stops set trip52=3 where id=406;
update stops set trip52=4 where id=403;
update stops set trip52=5 where id=405;
update stops set trip52=6 where id=349;
update stops set trip52=7 where id=350;
update stops set trip52=8 where id=458;
update stops set trip52=9 where id=456;
update stops set trip52=10 where id=340;
update stops set trip52=11 where id=348;
update stops set trip52=12 where id=345;
update stops set trip52=13 where id=342;
update stops set trip52=14 where id=347;
update stops set trip52=15 where id=338;
update stops set trip52=16 where id=393;
update stops set trip52=17 where id=383;
update stops set trip52=18 where id=423;
update stops set trip52=19 where id=335;
update stops set trip52=20 where id=336;
update stops set trip52=21 where id=386;
update stops set trip52=22 where id=391;
update stops set trip52=23 where id=388;
update stops set trip52=24 where id=389;
update stops set trip52=25 where id=384;
update stops set trip52=26 where id=381;
update stops set trip52=27 where id=396;
update stops set trip52=28 where id=427;
```

```
update stops set trip52=29 where id=394;
update stops set trip52=30 where id=446;
update stops set trip52=31 where id=442;
update stops set trip52=32 where id=443;
update stops set trip52=33 where id=445;
update stops set trip52=34 where id=432;
update stops set trip52=35 where id=430;
update stops set trip52=36 where id=304;
update stops set trip52=37 where id=437;
update stops set trip52=38 where id=434;
update stops set trip52=39 where id=436;
update stops set trip52=1 where id=418;

# Set all values in stops table to be 0
update stops set trip51=0 where trip51>0;

# Update all stops in stop table
update stops set trip51=2 where id=409;
update stops set trip51=3 where id=406;
update stops set trip51=4 where id=403;
update stops set trip51=5 where id=405;
update stops set trip51=6 where id=349;
update stops set trip51=7 where id=350;
update stops set trip51=8 where id=458;
update stops set trip51=9 where id=456;
update stops set trip51=10 where id=340;
update stops set trip51=11 where id=348;
update stops set trip51=12 where id=345;
update stops set trip51=13 where id=342;
update stops set trip51=14 where id=347;
update stops set trip51=15 where id=338;
update stops set trip51=16 where id=393;
update stops set trip51=17 where id=383;
update stops set trip51=18 where id=423;
update stops set trip51=19 where id=335;
update stops set trip51=20 where id=336;
update stops set trip51=21 where id=386;
update stops set trip51=22 where id=391;
update stops set trip51=23 where id=388;
update stops set trip51=24 where id=389;
update stops set trip51=25 where id=384;
update stops set trip51=26 where id=381;
update stops set trip51=27 where id=396;
update stops set trip51=28 where id=427;
update stops set trip51=29 where id=394;
update stops set trip51=30 where id=446;
```

```
update stops set trip51=31 where id=442;
update stops set trip51=32 where id=443;
update stops set trip51=33 where id=445;
update stops set trip51=34 where id=432;
update stops set trip51=35 where id=430;
update stops set trip51=36 where id=304;
update stops set trip51=37 where id=437;
update stops set trip51=38 where id=434;
update stops set trip51=39 where id=436;
update stops set trip51=1 where id=418;
```

D.1.3: Route 6A

```
# Redraw route 6A
update routelines set
gpolyencoded="e~mvEfbn_VQ{A_IxA^zBXz@T^pA~APf@^lCb@jBX|@^|D`@nAf@
~@hKfK`DxA^XVb@VnADzD\\rB`BxFXhAFt@AvA]tCMjY@~QzVBv@C@oAvB?`
CBBks@PyABeAA_MiGBFoJ@kJ|EwAnL{B]sA@mJeI?{@ZiC`BsCtAsCiKmEnDgA
h@}CdA{NtCiFp@{CPqBsP",
levels="PDF@C?ACA?AB@EBB@EAAC?F@BB@GCCD?JA@BEE?GBFBHD@@F
FC@DBAGP" where name="trip61";

# Delete all data from Trip61 table
delete from trip61;

# Route 6A
insert into trip61 values (11, 636.9, 355, 2, 10);
insert into trip61 values (13, 1782.5, 353, 3, 10);
insert into trip61 values (14, 2022.4, 354, 4, 10);
insert into trip61 values (15, 2328.9, 422, 5, 10);
insert into trip61 values (16, 2723.0, 421, 6, 10);
insert into trip61 values (17, 2857.4, 376, 7, 10);
insert into trip61 values (18, 3426.8, 429, 8, 10);
insert into trip61 values (19, 3765.2, 428, 9, 10);
insert into trip61 values (21, 4284.0, 341, 10, 10);
insert into trip61 values (24, 5049.7, 331, 11, 10);
insert into trip61 values (25, 5251.7, 329, 12, 10);
insert into trip61 values (29, 6680.3, 455, 1, 660);

# Set all values in stops table to be 0
update stops set trip61=0 where trip61>0;

# Update all stops in stop table
update stops set trip61=2 where id=355;
update stops set trip61=3 where id=353;
update stops set trip61=4 where id=354;
update stops set trip61=5 where id=422;
update stops set trip61=6 where id=421;
update stops set trip61=7 where id=376;
update stops set trip61=8 where id=429;
update stops set trip61=9 where id=428;
update stops set trip61=10 where id=341;
update stops set trip61=11 where id=331;
update stops set trip61=12 where id=329;
update stops set trip61=1 where id=455;
```

D.1.4: Route 6B

```
# Redraw route 6B
update routelines set
gpolyencoded="q~mvEf`n_VkAyKa@wEDwANw@^{@`@c@f@e@p@i@\\[pAu@j
@K^Cp@?fALViB\\gA|CyBRSxKiQb@c@v@]^Kr@@PHfKDtFEtKL|CGp@YpB{Ah
@hAv]vr@nDpHf@[`CuBEB_DmGaDdCoSma@mErD_@`@a@t@e@`Be@jBg@rAq
@tAo@|@o@l@aLbJ?CwOvMsCfCgAr@aAb@gC|@eOxCqBZmBVmDR}BsR",
levels="PDF@C?ACA?AB@EBB@EAAC?F@BB@GCCD?JA@BEE?GBFBHD@@F
FC@DB@GP" where name="trip62";

# Delete all data from trip62 table
delete from trip62;

# Route 6B
insert into trip62 values (33, 681.2, 457, 2, 10);
insert into trip62 values (34.5, 1553.4, 351, 3, 10);
insert into trip62 values (35.5, 1916.4, 352, 4, 10);
insert into trip62 values (36, 2143.9, 408, 5, 10);
insert into trip62 values (37, 2490.7, 404, 6, 10);
insert into trip62 values (38, 2798.9, 407, 7, 10);
insert into trip62 values (38.5, 3073.8, 410, 8, 10);
insert into trip62 values (39, 3270.7, 418, 9, 360);
insert into trip62 values (46, 3690.9, 409, 10, 10);
insert into trip62 values (47, 3915.1, 406, 11, 10);
insert into trip62 values (48, 4229.5, 403, 12, 10);
insert into trip62 values (49, 4386.7, 327, 13, 10);
insert into trip62 values (50, 4823.0, 328, 14, 10);
insert into trip62 values (56, 6446.4, 455, 1, 360);

# Set all values in stops table to be 0
update stops set trip62=0 where trip62>0;

# Update all stops in stop table
update stops set trip62=2 where id=457;
update stops set trip62=3 where id=351;
update stops set trip62=4 where id=352;
update stops set trip62=5 where id=408;
update stops set trip62=6 where id=404;
update stops set trip62=7 where id=407;
update stops set trip62=8 where id=410;
update stops set trip62=9 where id=418;
update stops set trip62=10 where id=409;
update stops set trip62=11 where id=406;
update stops set trip62=12 where id=403;
update stops set trip62=13 where id=327;
```

```
update stops set trip62=14 where id=328;
update stops set trip62=1 where id=455;
```

D.1.5: Stops

```
## Insert new stops ##

# Routes 4, 6a, 6b stop
insert into stops (name, GLatLng, Trip1, Trip21, Trip22, Trip31, Trip32, Trip41, Trip42,
Trip43, Trip51, Trip52, Trip61, Trip62, Trip63, Trip64) values ('Kennedy Library - E',
'35.30229,-120.66318', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

# Route 5 stop
insert into stops (name, GLatLng, Trip1, Trip21, Trip22, Trip31, Trip32, Trip41, Trip42,
Trip43, Trip51, Trip52, Trip61, Trip62, Trip63, Trip64) values ('Kennedy Library - W',
'35.30230,-120.66346', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

# Routes 4, 6a, 6b stop
insert into stops (name, GLatLng, Trip1, Trip21, Trip22, Trip31, Trip32, Trip41, Trip42,
Trip43, Trip51, Trip52, Trip61, Trip62, Trip63, Trip64) values ('Performing Arts Center -
E', '35.30037,-120.65786', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

# Route 5 stop
insert into stops (name, GLatLng, Trip1, Trip21, Trip22, Trip31, Trip32, Trip41, Trip42,
Trip43, Trip51, Trip52, Trip61, Trip62, Trip63, Trip64) values ('Performing Arts Center -
W', '35.30039,-120.65776', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

## Update information of old stops ##

update stops set glatlng="35.2928656783017,-120.67769587039948" where id=429;
update stops set glatlng="35.29102669196606,-120.66474616527557" where id=331;
update stops set name="Foothill @ Cuesta - W" where id=342;
```

D.2: Sms.php

```
<?php
require "TextMarksV2APIClient.php";
require "stats.php"; // processRoute(SQLname, routeName) is used from stats.php

   /* The following lines of code are used only if making calls to the Textmarks API
    * This code, as written, does not use any API calls.
    */

   //$sMyApiKey        = API_KEY_HERE;
   //$sMyTextMarksUser = USER_PH_NUM_HERE
   //$sMyTextMarksPass = USER_PASS_HERE
   //$sKeyword         = 'slobus';
   //$tmapi = new TextMarksV2APIClient($sMyApiKey, $sMyTextMarksUser,
$sMyTextMarksPass);

   $route = $_GET['route'];

   if(isset($_GET['route']))
   {
       $route = $_GET['route'];
       $route = strtoupper($route);

       if($route == "4" || $route == "5")
       {
           echo "$route is not a valid route.  Please enter either $route" . "1 or " . "$route" .
"2 to specify the bus."
       }

       /* For routes 1, 2, and 3, pass $route as both params */
       if($route == "1")
       {
           processRoute($route, $route);
       }
       else if($route == "2")
       {
           processRoute($route, $route);
       }
       else if($route == "3")
       {
           processRoute($route, $route);
       }
```

```
        else if($route == "41")
        {
            processRoute($route, "4 Bus 1");
        }
        else if($route == "42")
        {
            processRoute($route, "4 Bus 2");
        }
        else if($route == "51")
        {
            processRoute($route, "5 Bus 1");
        }
        else if($route == "52")
        {
        processRoute($route, "5 Bus 2");
        }
        else if($route == "6A")
        {
            processRoute("61", $route);
        }
        else if($route == "6B")
        {
            processRoute("62", $route);
        }
        else // If we get here, the first word after "slobus" is not a valid choice
        {
        echo "$route is not a valid option. Valid options include: 1, 2, 3, 41, 42, 51, 52, 6a,
6b.";
        }
    }
    else // User did not enter an arg after "slobus" (aka $route == NULL)
    {
        echo "Please text slobus \"route\" where route is one of the following options: 1, 2,
3, 41, 42, 51, 52, 6a, 6b."
    }
?>
```