Auckland Transport Vehicle Tracking

**Dev Group N**

Chuan Law - 81677469  
Bryan Lim - 39351653  
Kyran Stagg - 78943881

INFO263

Web Design and Development

2017

University of Canterbury

# **Structure**

Our website, GoBus Transport App, is a website that gets the location of a bus via the use of the the Auckland Transport API and displays it on a live map with the Google Map API. With those API, we are able to search for buses on the Auckland transport database and display them dynamically on an interactive map. With the Auckland transport API, bus locations are also able to be tracked in real time upon selection of a bus on the map.

Languages used were HTML and PHP on the server side and JavaScript for client side.

When the site first launches, the header and database PHP files are included which displays the static header of the site which stays in a fixed position as the user scrolls. The database.php file enables the pages connection to the remote Auckland database.

A call to the JavaScript code map.js and the function initMap() is done next which initialises the google map using our google API key. This is where the jQuery and JavaScript code will take requests from the user and send them to the server to find the bus routes and display them on the map.

Upon selection of a route name, the apiQuery in map.js is called which uses an AJAX request to send the user input to the vehicle\_query.php page to be used in an SQL statement to locate all route\_ids. Route\_ids is used for the most important part of the app; the API call to find all the trips of busses on that particular route. JQuery is used to decode the returned JSON result from API call and create the array of each buses locations on that route. Finally, the array is returned to the map.js page and used to populate the map with map markers of the bus locations.

In addition to the skeleton files used, our group created and made use of 2 extra .php files: database.php and behicle\_query.php.

Database.php is the page we reference whenever a connection to the akl\_transport database needs to be made. Upon being called, it makes a remote connection to the databse with params stored in our config.php file. The variable $conn is returned. This page also contains two helper functions used to populate the dropdown menu on the index page with route\_short\_names and one to return an array of strings containing all the trip\_ids for a particular route selected.

Vehicle\_query.php is an essential page used to process the users route selection input and echo back to map.js the results containing the location and id of each bus on that route. It contains 1 main function processJSON takes in json formatted parameters and returns a decoded version of the bus information needed for the map markers. The getTripsId and apiCall functions are called from this page and are used to generate the required output for the map markers.

# **Teamwork**

Our group initially had 4 members but after contacting our 4th member and adding him to the team repository he failed to contact us in anyway and eventually dropped the course. This hindered our teamwork as a whole as we had an increase in workload to delegate to the remaining members.

The group project was stored online in a private repository on GitHub.com and this was the primary means of how code was shared between group members. GitKraken was also instrumental in ensuring merge errors were handled correctly and each member was kept with the most up to date version of the program. Most of the group work was done over a voice chatting service (Discord app) where the 3 of us were able to talk and update each other on the changes and discoveries we made. We had a few meetups at the university labs to work side-by side when our schedules lined up.

Tasks were delegated as follows:

Chuan: Calls to the Auckland transport API, SQL query statements and functions. Report writing.

Kyran: AJAX and XML requests, sending of data between the website (client and server side). Google map makers and general features of the map.

Bryan: CSS, documentation, comments and readability of code. Report writing.

# **The good and the bad**

Communication within the team was strong as we worked together well and methodically on most days. We were able to learn a lot from each other after each bug we encountered and fixed. We feel the CSS came out very nicely and were happy how the project came to be.

Things that we as a team had most difficulty with was getting the APIcall to function correctly using jQuery. We spent a lot of time trying to figure out the best way to approach this and used a lot of our limited time in our tight schedules which led to stress near the end of the project. Another difficult point during development was what direction to take after getting the route names from the drop down menu. We were very indecisive and uncertain as how to approach actually getting the map markers of the vehicles on the map and this is where were stuck for a few days.

# **Improvements**

A major cause of difficulty of this project for us the lack of knowledge of how the apiCall function worked. Near the end of the project we were rushed for time especially as it was the last week of UNI with everything else due at the same time. Starting earlier and getting it touch with everyone sooner would have been key in preventing this and provided us with more time to solve our issues and prevent the later frustration of the project.

In hindsight, working on the project earlier would have been more beneficial compared to late at night. This is due to the majority of busses not being active later at night thus making testing and debugging of the map marker placements a bit difficult and inconsistent.

# **Final reflections**

In the end, our GoBus transport app does not function correctly due to the difficulties we encountered with the apiCall. Instead of getting the results of all busses on a particular route instead we received all the current busses on the Auckland transport API.