INFO263 Written Report

Date: 11/10/2017
Group G
Denisa Dumitrescu, 25202742
Seina Nagai, 66321463
David Nguyen, 51113878
Xiaoshi Xie, 22378644

How our website works

On the server side, we have seven files: Config.php, footer.php, header.php, index.php, requests.php, getTrip.php, requests.php. On the client side we have map.js.

The drop-down select box shows the list of all the routes and allows the user to select one route on which they want to receive information. In order to show this list of the routes, firstly the server checks the connection with the MySQL database. If the server can successfull connect to the database, the server sends a query to retrieve all the route_short_name, which will be printed on the list.
When the user selects a route, the map shows locations of active vehicles on the route. The server sends a query to the database asking for the route short name of the selected route.

getTrip.php receives the route short name ($routename = $_GET['q'];) and getActiveVehicle function uses the route short name in the query to retrieve the corresponding trip ID. Then we create $querystring array consists of trip IDs. We call the apiCall function using the $querystring as a parameter to get the response containing the information about the vehicles on the selected route.

The first function in the map.js **initMap** displays the map on the website and sets the latitude and the longitude to Auckland. The second function **setMarkers** sets the latitude and the longitude, customize the marker and also creates an information window. When the user places the cursor on a vehicle location, an information window appears and shows the bus ID and the timestamp of the vehicle. This information window disappears when the user moves the cursor away from the marker. The map automatically resizes according to the vehicle positions. We set the zoom level to where all the markers can just be seen on a map, rather than setting the level to a higher resolution and showing the exact street which the vehicle is on. This is because we wanted to show the relative location of the vehicles.

One of the first things that we changed in the CSS file was the website header background colour to #ADD8E6 Color Hex. On our first try in doing this, only the colour inside the textbox changed. However, after we inspected the page elements using the ‘developer tools’, we realized that in order for the #ADD8E6 Color to display on the entire header background, we needed to add the tag “!important”. This CSS rule tells the page to prioritize the #ADD8E6 Color Hex and use it as the main background colour for the header instead of the black default color.

The above change was followed by changing the header text font. This was done by editing the font-style property in the MasterCSS file.
Thirdly, we have also changed the size of the map using CSS. The map property we have changed is the height. This was done by manually inputting a number of pixels and using the trial-error method to decide which size was ideal.

A reflective summary of teamwork

We communicated with each other through our Facebook group and set weekly meetings. At the beginning, we planned what we wanted to get done in each meeting but we found some tasks difficult halfway and could no longer follow the plan. Although things did not go as smoothly as our initial plan, the process of planning definitely helped us to break down each task into smaller parts and see roughly how long each task would take. This encouraged us to start working on the project as early as possible and meet regularly. We tried to find the time where all of us could meet and work on the project but sometimes it was difficult as we all had quite a different timetable and other assignments due. However, most of us managed to meet in the weekly lab and stay updated with any changes or progress made. Some of us found that a smaller group would have been easier to work with as it would be less painful to schedule meetings. However, having four of us did help gain more ideas and inputs.

We shared our codes on GitHub, which we found was the most efficient way of sharing codes. On GitHub we could easily see who has updated which file and it was much more efficient than sharing the entire project file to all of us via email. We did not separate tasks clearly, but some of us were more comfortable with CSS than JavaScript or PHP so they focused more on the designing part. We were all at different stages of programming, so we thought assigning tasks to each member would not be a good idea.

At the beginning, we struggled with understanding how the files were interacting with each other. However, as we went through the weekly lab materials and read some articles on the internet, for example Google Map Tutorials, it started to make sense. Being able to work with the team members and listening to their ideas have given each of us additional knowledge about programming.