INFO263

Auckland Transport Vehicle Tracking Assignment report

Produced by Dev Group B:

Aaron Bong

55623236

Ariel Corpuz Evangelista

62193622

Jinyoung Park

15648768

Juanpeng (David) Qiu

35635713

13/10/2017

- **Teamwork description**

Before we started handing on the project, we met up for the first time and decided to do our work together in LAW110 for every meetup since the machines in the lab has everything we need for the project. Since we already have database setup with MySQL workbench and Microsoft Expression Web from our lab session, it will be easier this way and also saving the time for setup a server at our home machine. Furthermore, we think its good to work on the project when our fellow classmates are around so we can ask each other questions as well as know what result they got.

- **Reflective summary**

In order to share our progress and keep them up to date, we were using Github that were thought in the lab session by our tutor on everyone's machine. Due to the convenience of Github, we are able to share by pull and push to online. However, we realised if we push or pull at the same time we might experience some collision between files. Although we were having Github setup in different machine, we still tried to do our tasks in the same machine as much as possible to reduce the chance of messing up our codes accidentally. Also we kept one machine that has a clean backup of our files until we finalised every new file and added it in.

Aaron, was our team leader and had great leading experience. He delegated tasks and kept track of progress. At first Ariel was the first one to get onto the fetching data from Auckland API to database, and Jinyoung has got great database management skill so they can work together for figuring out how Auckland API is working and fetching data out of the database provided. We decided that Ariel and Jinyoung would work on the database part and Aaron and David would be in charge to write the codes with php such as how to fetch data from database to the website, html and JavaScript such as how to display information or function needs to do with the map for our website. Eventually we worked on CSS all together to make our website look better and also change the code style and structure.

After we have decided each other's task, the first difficulty we met was the Auckland API. We stuck there for a very long time because we could not query out what we would like to display, and we found out the database was constantly updating. Our problem was fixed when our tutor fixed the API result. Second difficulty we had was creating the dropdown list in combo box to display

several options that can be clicked. Because in this part we had to also reference the data from database. Third difficulty was when we have to setup a timer for refreshing the location of markers routinely and map resizing. We met many more difficulties than just these, but we managed to solve it by asking tutor and lecturer and keep our progress ongoing. Pretty much everything went as expected especially once we solve one difficulties, everything just became easier and easier afterward. Eventually we think we made a very good decision on keeping minimal number of PC to use at the same time and without doing individual work at home for this project. This strategy is a good way to keep up for our future project but this also depends on the workloads and tasks of each team member.

- **Brief description of our website**

For server-side, we queried everything from Auckland API and the database to the website for client. It is significantly user friendly on the client-side, all they need to do is access the webpage, the webpage will show a map that will initially focus at Auckland and a dropdown list called “Select route” will appear. Then the user will select a route that he/she would like to view on the map by choosing one of the route in dropdown list. On server-side it will fetch the requested data from Auckland Transport Database and feed it to the Auckland Transport API. Then, the API send the information of each busses back to the client. After that, the Google Map API will process the information and place several markers on the map. The markers are where the busses are currently located. When clicking on the marker of each bus, it will pop up a window that shows the Vehicle ID, start time for this vehicle and the time stamp. The webpage will automatically refresh the map every 30 seconds in order to update the location for busses. On server-side this was achieved by removing all the markers and query new vehicles information again every 30 seconds.

- **Conclusion**

Our ideas came from every one of us and every decision was finalised by our leader when everyone agreed with it. From the great direction given by our leader, we managed to finish each other’s task on time and help out others if necessary. We had been doing great teamwork with good efficiency on progress for our project overall.