INFO 263
Group J - Assignment

Members:
Felix - 36167301
Blake - 47787387
Jack - 31633497

Introduction

Description
How the website works

Client side:
Once the index page has loaded, a map object is constructed and the main map element is initialised. When a route is selected from the dropdown menu or an update is called specifically, a query is sent to the database and a call to the auckland transport API, comparing these results gives the currently active vehicles on the selected route. A marker on the map is set for each vehicle returned and the auto refresh timer is set to refresh the map every 30 seconds. There are buttons for re-centering the map, for forcing an update of vehicle positions, and to stop or start the auto refreshing timer.

Server side:
When a call to get vehicle information about a route from the client side occurs, the service side takes the following steps to return a list of currently operating vehicles:

1) Prepare a query to the database for all entries with a trip_id given that they are associated with the given route. Send and retrieve the query from the database.
2) Prepare and send a call to the Auckland Transport API with an empty list of parameters.**
3) Compare the results of the database query and the AT API call to get a list of currently operating vehicles (from API) that are on the currently selected route (database query).
4) Send the list of current vehicles, as a JSON object, back to the client side processing, which creates a marker for each vehicle and displays it on the map.

Process:
** We found that it was not possible to send an API request given all of the results from the database query. It would also be much slower if many API requests were made to accommodate the large amount of potential vehicles from the database query (one route had ~450 results). So in the end it was faster and easier to do two separate operations and compare the results from each.

Teamwork

Communication.

For us to communicate with each other we first sent out emails to our respective email addresses. From there we decided to use a cloud based messaging service called Slack to collaborate and communicate.

We used this to our advantage to assign tasks and organize group meetings. Since we all had differing schedules this proved very effective as we could relay important information about the task in a consistent location. However whilst there is 4 of us in this group we were only able to contact three members. This made it hard as we didn't want to go too far on ahead without all the members. For next time we would use Slack again as it was very easy to use and effective in communicating. In addition we would try to determine if our missing member is still taking the course sooner.

Share Resources.

To share resources one of our members set up a repository on github then sent out links to the other members through Slack. This meant, as long as we were connected to the internet, we could easily update to the latest working version of the assignment a without any major issues and having to message the group chat if they had done any more work. With each push to the repository we left a message in the log detailing what the update included (creating, editing and deleting).

This was very effective as in gave us the current progress of a certain feature another user was implementing. With each member working on the project changes were frequent; it allowed us fast prototyping and refinement with each review of our code. However there is some downsides with using github as sometimes we ended up with collisions . This is when one user edits code that another ones already editing, resulting in the repository not knowing which version to use.

Appointed tasks.

To appoint tasks we used Slack. We listed the requirements of the specification, detailing notable features. We then allocated ourselves to tasks notifying the other members. Users then would push their work to repository. If a task was unfinished another member then come along and continue their work.

We found this solution to be very fluid and allowed us to work on the assignment in short periods as we have other interests and not having to commit a large portion of a week to finish and test a certain feature of the specification. Sometimes this would cause confusion between members when the repository wasn't up to date, however these were fair and few. We felt this approach was satisfactory for the assignment but for next time we could consider a more rigid task appointment system.