VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Wellington Traffic Visualisation

Geoffrey Longuet

Supervisors: Kris Bubendorfer, Stuart Marshall

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering with Honours.

Abstract

Victoria University of Wellington has obtained traffic data from the Wellington City Council. As part of this ENGR489 project, a system consisting of both a server and a web based visualisation tool has been produced. This system streams the data in simulated real time from the server to the clients connected to it. The purpose is to help members of the public to identify traffic congestion easily and assist in making planning and travel decisions.

Contents

Chapter 1

Introduction

1.1 Problem

A business specialising in traffic consultancy by the name of AraFlow has been installing vehicle traffic measuring devices around Wellington City. These installations have been done in collaboration with the Wellington City Council (WCC). The WCC are trying to improve traffic congestion around Wellington, this data and systems which use it will assist them in making informed decisions about how to plan and manage Wellington roads. The tool built as part of this ENGR489 project aims to assist road planners, emergency services, and potentially regular members of the public - if the WCC approve. It does this by creating useful real-time information from the data provided by AraFlow. The tool works with a copy of a dataset which has been captured by AraFlow, the data within are spread out over the span of three months. The data will be played back from the server to clients in real-time, resulting in an accurate simulation as if the data were being reported live. Unlike the previous project, data must constantly be sent to the clients as it arrives.

The raw data are gathered by identifying unique Bluetooth devices such as mobile phones as they pass between two scanning devices placed along road segments. This data is sent in real-time to AraFlow, where it is cleaned of outliers and empty data. As this is handled by AraFlow, cleaning the data is outside the scope of this project.

These raw data need to be converted into human-understandable information such as graphs and visualisations on a map. These visualisations will help to identify current traffic conditions, how they are trending, and the rate at which they are worsening or improving. To achieve this, a server has been written to perform processing on the data to get it into a format which can be rendered on web maps and charting tools. A web-based front end has also been written to render the content in an easy to understand format.

1.2 Requirements

A system which is suitable for use by road planners, emergency services, and members of the public requires a an investigation of their individual needs. In this section, their needs are identified as a basis for the requirements. Areas of common ground are identified and used for the requirements to ensure that the they are suitable for all users.

As the tool needs to work for members of the public, the visualisations should be built to run on a convenient platform, a simple solution for this is to build it to run in a web browser. This results in a platform agnostic tool, and is still suitable for emergency services and city planners to use.

Users will most likely want to see information as graphs rather than raw data tables.

This applies to the public and emergency services as they will often lack domain specific knowledge to easily interpret raw data. This does not mean that graphs will unhelpful to city planners, as they help to summarise data for a quick overview, large raw data tables are generally difficult to summarise without some form of visualisation.

The tool will need some form of map which represents the traffic as if looking down from a birds eye view, this would help in visualising an overview of all traffic. For city planners it would help to identify routes of congestion and bottlenecks which may need improvement. For members of the public and emergency services, this would help to avoid areas of congestion.

Section 2.1 takes this analysis and uses it to produce a more formal set of requirements.

Chapter 2

Related Work and Background Information

2.1 Requirements

After further analysis of the problem it was broken down into specific, measurable, achievable, and realistic goals. The following requirements have been set as a target for this project.

- **R1** The system must be web-browser-based and made with two components, a client and a server.
- **R2** The client must visually represent Wellington traffic volume and trends on a map in real-time.
- **R3** The map must be interactive with support for zooming, panning, and selecting streets.
- **R4** The client must represent **current traffic volume** on a per street basis as recorded/observed by the server on a graph.
- **R5** The client must represent **current traffic trends** on a per street basis as recorded/observed by the server on a graph.
- **R6** The client must represent **historical traffic volume** by time interval on a per street basis as recorded/observed by the server on a graph.
- **R7** The client must support searching for specific streets available in the dataset.
- **R8** The server must handle multiple client connections concurrently.
- **R9** The server must update all clients with real-time updates as they arrive.

2.2 Traffic Visualisations

Real time vehicle traffic visualisation has been fairly well researched in the past. These research papers frequently discuss the difficulty of displaying dynamic data in comparison to static data when working with Geographical Information Systems (GISs). As stated by C. Claramunt et al, "current GISs are still not adapted to the management of very dynamic geographical phenomena due to the lack of interoperability with real-time computing facilities." It should be noted that this is a paper from 2000, since then with the advancement of web technologies there has been an emphasis on improving real time reactivity and web viewable maps. There are now many front end frameworks designed to handle reactivity such as React, AngularJS, and Blaze.

One possible method for overlaying data on to street maps is as a heat map. Heat maps are generally used to represent data from a matrix, where the colour of a point is based off of its value, and its position is proportional to its location in the matrix. This makes them ideally suited for data consisting of many points such as clicks on a web page or rain intensity in a region. Their suitability for visualising traffic data has been explored in tools such as The Strava Labs cycling heatmap [12], or the Floating Car Data Information System of Berlin [16].

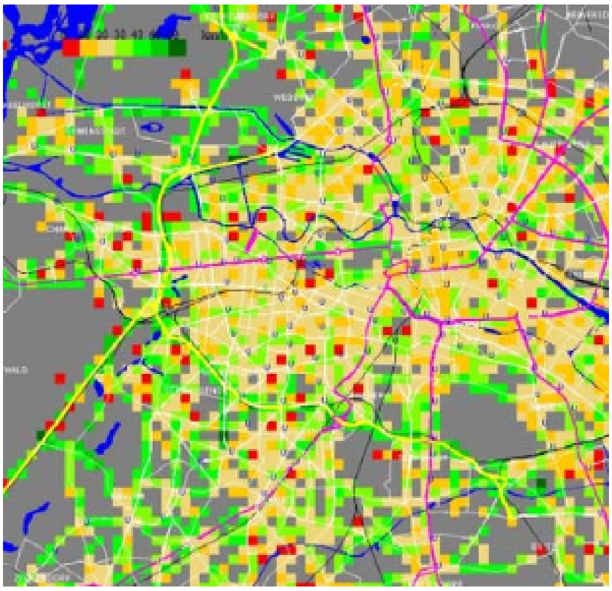


Figure 2.1: Mean Velocities in Berlin as a heat map.

The Floating Car Data (FCD) Information System was developed to gather information from devices placed on Berlin taxis. They used two methods for data collection, one was active GPS devices which communicated with the taxi headquarters, and the other was a passive system, which placed a unique transponder on taxis for beacons on the road to detect. This second method is very similar to the system being used by the WCC for this project, where the only major difference is that the unique transponder is replaced by a road user's Bluetooth device. The paper points out a drawback to this passive approach, which is the lack of necessary information when the beacons may be too far apart [16]. The system can be seen in Fig. 2.1, this is messy and is difficult to use as a means of identifying routes of congestion. When put in the context of this project, this makes it meet requirement R2, but not to a particularly high standard, the use of colour with a transition between red to green explored further in the Design chapter.

Bluetooth devices are suitable to use for unique vehicle detection as they have a unique MAC address associated with them. If a car passes through the detectors at the start and then the end of a segment, it is possible to measure the time taken to travel the segment by that car. This brings up the problem of outliers such as cars which leave the segment, or stop in the middle. In the case of Barceló et al. their system set an upper and lower bound of travel time based on the expected speed limits of the road. The aggregated average time taken was calculated at the end of each minute, and if the trend was worsening or improving, the bounds were recalculated [5]. Any data points outside the bounds were discarded. The findings of their report show high accuracy in terms of predicted travel time.

The website "Bostonography" is run by two cartographers who produced a visualisation for The Massachusetts Bay Transport Authority (MBTA). This visualisation displays bus locations and speeds within Boston as a web-like graph. It displays data for the past three hours, and updates itself every hour [3]. A drawback to this approach is the lack of real-time feedback of the traffic trends, by the time the information has become available to the

Figure 2.2: Strava Labs cycling heatmap over Wellington.

members of the public, the data are already out of date. They would not be able to rely on the visualisation to decide on which route to travel. The tool is web based and interactive, it meets requirements R1 and R3. The lack of real-time data means that it fails to fully meet requirement R2.

While the MBTA map is a about visualising a different kind of data (speed and location) to this project (volume and trend), it showcases the use of colour along streets as a means of conveying information. Taking inspiration from the MBTA, in this project I use it as a means of displaying the traffic volume trend of a street.

2.3 Performance

There are multiple performance concerns when it comes to displaying this many data in real time. One issue is balancing the need to support older web browsers, whilst maintaining efficient rendering. Visualisations may use scalable vector graphics (SVG) to produce their overlays. While this is acceptable in situations without too many SVG objects, this can become a huge performance concern when there are many complex objects to render such as all streets in a city. A possible solution is to 'rasterise' the data in the browser onto an HTML5 canvas object, significantly cutting down the computations required for repositioning the map. An HTML5 canvas is only supported in modern browsers, but also has a performance advantage [10]. Both the Strava labs and the MBTA visualisations use a different rendering technique, where they generate rasterised overlay data on the server to display on top of the map tiles on the client. While this works for data which is updated infrequently, it is unsuitable for real-time visualisation due to bandwidth restrictions and the need to generate tiles at various levels of zoom every minute. It is more feasible to inform the client of changes, and have it handle the rendering.

If SVGs are being used, there are still ways to improve performance, such as with the Simplify.js library, designed to reduce the number of points in an SVG whilst mostly maintaining its original shape see Figure 2.4.

Another performance concern is handling real-time updates to the client whilst keeping data transfer and re-rendering cost down. If the server has to resend the entire dataset whenever a single change is made, this results in completely unnecessary data transfer of

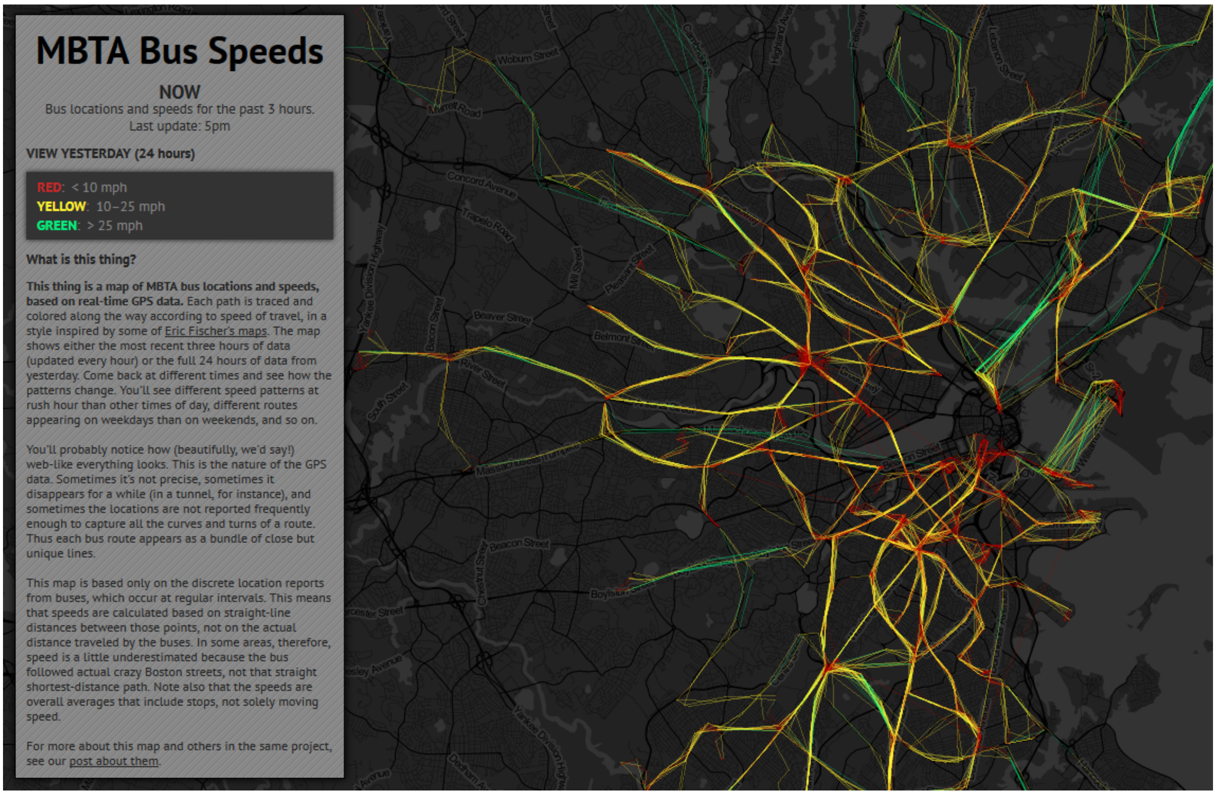


Figure 2.3: MBTA bus speed visualisation.

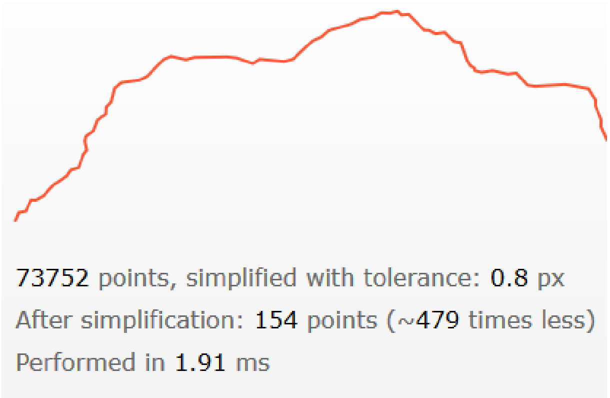


Figure 2.4: A polyline simplification, performed with Simplify.js [2].

unchanged objects between the server and client, and scales very poorly as more clients are connected. To make matters worse, the client would not know what has been changed in the dataset and would either have to compare its current dataset with the new one to identify changes, or re-render the entire thing. Re-rendering is an expensive operation, and should be performed as little as possible. The solution to this problem was to only send changes to the client as necessary, and allow the client to update its own copy of the dataset. This is made possible with a framework which supports a client-side database with insert and update capabilities, and is discussed in more detail in the Full Stack Frameworks section.

2.4 Data

The data provided by AraFlow are in the form of database dumps in comma separated value (CSV) format. This is a plain text format, and given its size, it is too large to operate directly from in an efficient manner. There must be optimisations put in place to perform fast data lookup and modification. This means that the database dumps should be reconstructed into a new database.

Initially the received data dump was lacking any location information other than human readable strings, which did not appear to follow any naming conventions. A large amount of time was invested in attempting a solution to convert and repair these data into a close approximation of their actual representations. After speaking with AraFlow, a new database table was obtained which held the actual geolocation data for each street in the dataset. This has had a fairly significant impact on the development direction for the tool, which is discussed in further detail in section 4.4 of this report.

The data dumps are made up of a series of files, each being a table in the original relational database. The general overview of the most important fields within each table is below.

Route Definitions

- Human readable street names
- Unique street ids
- Connecting street ids
- Length and journey time

Locations

- Unique street id
- Latitude and Longitude

Data

- Unique street id
- Timestamp
- Number of cars detected
- Sum of travel time of all cars

The above data need to transformed into useful visualisations. As this project requires a web-based solution, a summary of possible tools is described below.

2.5 Full Stack Frameworks

There are a number of tools that exist today for scaffolding out web applications. Generally they are made up of multiple smaller tools such as a package manager, CSS preprocessor, front-end reactivity framework, etc. As this project requires operations to be performed at both the server and the client, the most appropriate tool is built with the full stack in mind. The term full stack refers to the technologies in both the front-end and back-end.

One such scaffolding tool is Meteor. Meteor uses Node as a back-end server, MongoDB as a database, and it supports multiple front-end frameworks. Something which sets it apart from other similar tools is its built in support for the distributed data protocol (DDP). The DDP enables client subscriptions to Meteor server databases, meaning that the client copy of the dataset will update in realtime with the server without the developer having to put together a protocol themselves. It also only performs updates as necessary to reduce bandwidth usage. Meteor also has its own package manager (Asmosphere) and encapsulates other tools such as the Node Package Manager (NPM). This means that it will be using copies of these tools which are known to be stable together. [8]

Another possibility is to use a Yeoman generator. Yeoman is different to Meteor in that it is designed to work with different generators. These generators are developed by the community or the official development team to enable various types of applications to be built. They do this by setting up a development environment made up of smaller tools. For example there is a generator specifically for angular full stack applications. This installs the MEAN stack: MongoDB, Express, AngularJS, and NodeJS. These are installed using NPM and not a Yeoman-specific package manager.

Chapter 3

Design

This chapter describes the processes and decisions behind the design of the tool. It also introduces the primary persona, and how it has influenced the outcome of the project.

3.1 Primary Persona - Jane Smith

Jane is a young professional who lives out in Miramar, but works in the central business district. Her work is primarily done on a computer working with specific tools for her field, and she has no software development knowledge. Like many people in her age range, she is used to daily interactions with web applications such as Facebook and Google Maps. She finds most of these web applications intuitive to use and is frustrated when they take a while to get used to.

3.1.1 Potential Use Cases

Scenario one: View detailed real-time traffic volume information for a specific street
Jane navigates to the web application to find out if she should leave for home now, or wait until the traffic congestion has eased up. Upon inspecting the map, she can see that one street along her route looks busy.

1. Jane zooms in on the street of interest and clicks on it.

2. A pin appears on the map with a textual popover providing detailed information. Graphs also appear showing the current traffic volume trend and the usual volume trend.

3. Using this information she can see that the spike in traffic is abnormal and will most likely result in a delay on her way home.

Scenario two: Viewing an overview of historical data
Jane would like to leave for Johnsonville on her day off, but she is unsure when she might have wake up to avoid the traffic.

1. Jane zooms in on the streets of interest and clicks on them.

2. On each street, a histogram of traffic volume is generated and displayed below the map.

Scenario three: Finding a street by name to view detailed traffic information
Jane would like to meet a friend for lunch on Wakefield Street, but she doesn't know where it is or how busy it is. So she would like to find it and view the traffic information.

1. Jane clicks on the search box on the map.
2. She begins typing and the box auto-completes the name, she clicks on the correct suggestion.
3. A pin appears on the map with a textual popover providing detailed information. Graphs also appear showing the current traffic volume trend and the usual volume trend.

3.2 Secondary Persona - Joe Smith

Joe is an emergency services coordinator, his job involves providing navigation directions to emergency services deployed within the city.

3.2.1 Potential Use Cases

Scenario four: Identify route to target destination where traffic is currently not worsening.
Joe needs to find a route to a destination which is unlikely to become congested so that he can navigate a fire truck to a call out.

1. Joe searches the overview map for streets without worsening traffic trends and with low traffic volumes.
2. Joe clicks on the streets for more information via the histogram to suggest whether the street usually has worsening traffic upcoming.
3. Joe finds a route and provides the information to the fire truck.

3.3 Tertiary persona - John Doe

John is a city planner, his job is to identify areas of traffic congestion in Wellington so that he can make decisions about planned upgrades to Wellington's streets.

3.3.1 Potential Use Cases

Scenario five: Identifying peak traffic times for different streets
John would like to identify when streets are most likely to be busy to find intersections where traffic signals may need to be modified during those times

1. John can click on individual street segments to view the histograms for each street.
2. John can use segments connecting to the same intersection to see if one is more busy than the other during specific hours.
3. The traffic system can be modified to ease traffic on certain roads during specific hours.
4. If the traffic signal system supported updating in real-time the feedback from the tool could be used to update the lights based on the current traffic trend.

Scenario six: Identifying areas of traffic during/after road closures
When streets need to close unexpectedly, John would like to know how the traffic congestion is affected. This would allow the WCC to design street layouts to handle unexpected traffic.

John would like to examine the differences in regular routes as a result of a recent Wellington earthquake.

1. John examines the overview to find routes being used as alternatives.

2. John can click on these streets to find the usual traffic volume on the histogram and compare it to the current.

3.4 Overall Architecture

This section describes the full system architecture using real-time traffic data instead of a simulation. This is not to be confused with the prototype which has been produced as part of this project.

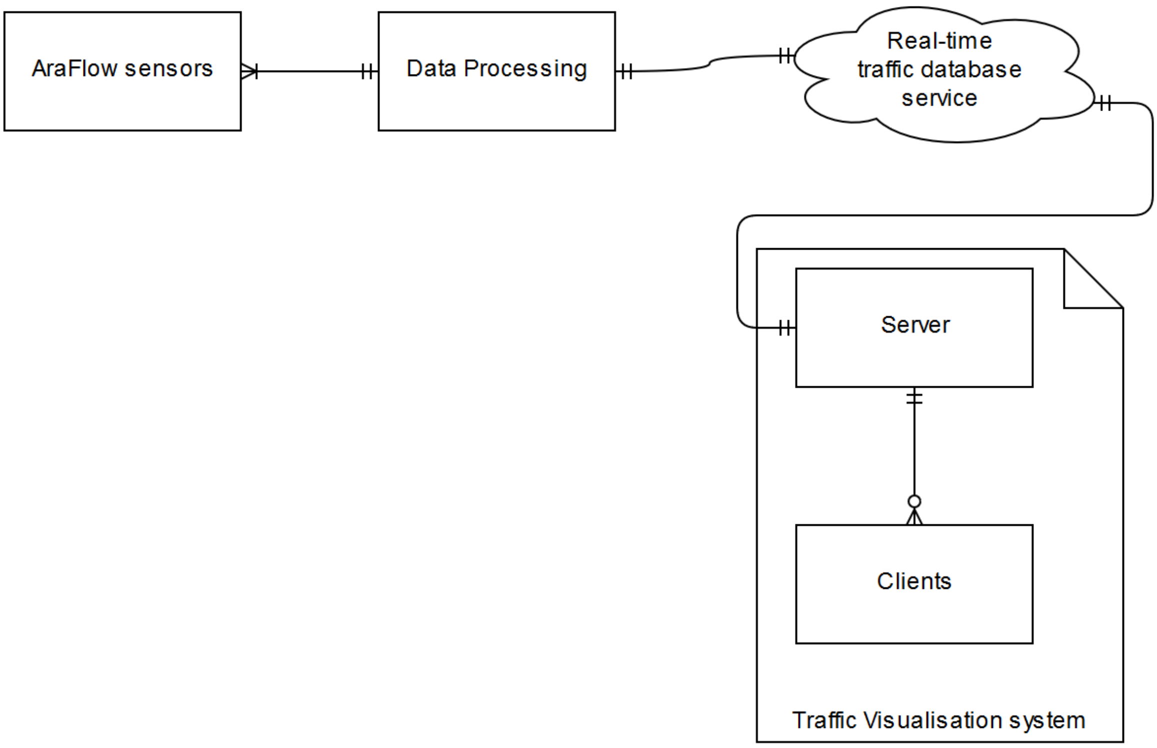


Figure 3.1: A representation of the complete architecture including their entity relations. Read left-to-right.

The architecture in Fig. 3.1 represents the entire system consisting of the sensors around Wellington, the Araflow Data processing system, and a database containing the latest information by the minute. Within the Traffic Visualisation system boundary is the scope of work done for this project, and pertains to the prototype implementation.

3.5 Front-End Component Design

3.5.1 Map Appearance - Flow map

A flow map is used to "show the movement of objects from one location to another, such as the number of people in a migration, the amount of goods being traded, or the number of packets in a network" [7]. It is used in other previously mentioned map tools such as the Strava Labs cycle map [12], or the MBTA bus speed graph [3]. Due to its heavy use for traffic data, I believe it was suitable as a visualisation for this project.

The map is the most crucial visualisation of the web app. The map needs to convey an overview of information in an intuitive and meaningful way that saves the user from having to read through data. I needed to convey two types of data, the current street volume, and the current trend. To display this on each path, I decided that the width and colour of the paths would be the best features of the map to change. The possible range of values representable with width is very small (approx four before it clutters the map), whereas a colour range is large. Due to the limited range and discrete nature of the current volume value, I decided that it should affect width rather than colour. As the trend is a continuous decimal value between -1 and 1, it is better suited to the possible values in a colour space. Lastly, as these paths were being displayed as an overlay to a map, the colours shouldn't conflict with each other, this meant choosing map tiles with minimal information and as few colours as possible. The default Open Street Map tiles contain a vast amount of information such as shops, hospitals, landmarks, etc. These are unnecessary and lead to information overload, they also use up the colour space, making individual paths difficult to distinguish. The solution was to use map tiles in either white or black, which utilise shades of grey to highlight buildings and streets. This leaves plenty of free colour space for the paths to use.

The colours chosen to represent the path were in the colour space of red to green. These were chosen as they are a loose metaphor for traffic lights, with which many users of the tool will already be familiar. Red represents "gaining volume" and green represents "reduction in volume". There are 65,025 possible colour combinations when modifying just the red and green channels in an 8 bit depth per channel colour space. This is unnecessarily large and isn't a scale, rather it is simply all the possible combinations. The scale I chose was very simple, the total of the intensity of the red and green channels must add up to 255 total, a completely negative trend resulted in a red value of 255 and a completely positive trend resulted in a green value of 255. Meaning that when there is no trend, both channels are 128. The "brightness" of the resulting colour is 50% when there is no trend and 100% at both extremes of the spectrum, helping rapidly changing streets to stand out. Brightness is the mean of the three colour channels [15].

The map also has support for zooming, panning, and selecting streets by using the mouse to click and drag. Mobile support uses extra gestures such as pinch and double tap to zoom.

Requirements R2 and R3 are met by the content shown in this section. The Flow map represents traffic and volume for Wellington streets. An explanation of how this is achieved in real-time is in the following chapter.

3.5.2 Detailed street graphs

Each street in the Flow map can be clicked on for a detailed inspection of real-time traffic trends and usual traffic volumes. When a street is clicked on, the server generates datasets by performing queries on the data collection which the client uses to display or update two graphs.

Volume trend graph
Two series are displayed on the chart, the real-time volume for the street, and the exponential moving average. The volume data are discrete due to the nature of traffic volume, however the moving average data are continuous. A line graph was chosen due to its suitability for plotting multiple series against time on one chart [19]. As a line graph is suitable for both discrete and continuous data, it is an obvious choice for this application. The exponential moving average serves two purposes, one is to help smooth the data to get a better idea of the trend when observing the graph, and the other is so that the flow map visualisation can use the values to calculate the street colour.

This meets the requirements R4 and R5 as the chart can represent both traffic volume and trends as different series on the same pair of axes. Once again, real-time support is explained in the following chapter.

Histogram of previous data
This chart was chosen to represent a dataset of the total of all previously recorded cars for the selected street. The data are binned into 30 minute intervals and displayed on a bar chart. The idea behind the visualisation is that a user can see the peak times that traffic volume is likely to be heavy on a street. They can then compare the graph with the real-time line graph to get an idea of whether the current traffic volume is normal. Initially the graph displayed the raw numbers of total traffic recorded per 30 minute bin, this was changed to proportions represented as percentages because the raw numbers hold no meaning as the start date of recording is not defined. The proportion of traffic holds more meaning whilst still displaying an identical distribution. A bar chart was chosen for this visualisation as it allows a user to still get a sense of the traffic distribution whilst also conveying the concept that the data has been summarised into 30 minute bins.

This meets requirement R6. While the data are not updated in “real-time”, a server component is still required to process and generate a dataset for the visualisation as will be described in the next chapter.

3.5.3 Search function

In order to meet requirement R7, the search box was added to the control overlay of the map. The search box supports auto-completion of street names. A user can start typing any part of a street name and another box will appear and populate with suggestions below. They can click on the suggestion or use the arrow keys and press enter to have the map select the street. Just as in the use cases, this can help a user to find a street when they know the name of it but not the location.

3.5.4 Summary of Requirements

In summary of this section, the following table represents the requirements which have been explored. The requirements which remain unexplored are not visualisation components.

R1	Unexplored
R2	Explored
R3	Explored
R4	Explored
R5	Explored
R6	Explored
R7	Explored

R8	Unexplored
R9	Unexplored

3.6 Back-End Component Design

3.6.1 Araflow Sensors and Data Processing

These sensors are placed around Wellington and detect Bluetooth devices as they pass between two points. They send their updates to AraFlow where data processing occurs to remove outliers and untrustworthy data. This is then sent to the Real-time traffic database service.

3.6.2 Real-time traffic database service

This service is accessible via the public internet but will only respond when a service is authorised to use it. It is updated every minute with the cleaned data for the streets in Wellington. The Server is authorised to use the service. The purpose of this database is for efficient lookups of street data so that the server can supply this information to all clients.

3.6.3 Server

The Back-End supplies a stream of updates to the clients for display. The updates are retrieved from the real-time traffic database service every minute. This is done using a server which can handle real-time updating to multiple clients without the need for client polling. Each client is provided with a unique connection and set of data subscriptions, allowing for each user to perform unique inspections of the visualisations.

Both the front-end and back-end components have been discussed, together they meet requirement R1, as they can be considered a client and server respectively. Requirements R8, and R9 are also met thanks to the server's ability to handle multiple unique connections in real-time.

3.6.4 Summary of Requirements

In this chapter so far, all requirements have been explored and met by the design. The design encompasses the full system as can be seen in Fig. 3.1. The following chapter on implementation will discuss the subset of components used for the prototype as well as those which had to be simulated.

Chapter 4

Implementation

This chapter discusses some of the lower level details of the system, it also expands upon the Design chapter by explaining how the visualisations function rather than their effect on user experience. It covers the implementation of the software, mathematics, and algorithms used to transform, transmit, and display the data to the visualisation components.

The data provided are in a form as if from the real-time traffic database service. As I was not supplied with access to the AraFlow database, the real-time database service is simulated by the server. This is primarily due to not having access to the database made by AraFlow. This means that the scope of the prototype is limited to the note as in Fig. 3.1.

4.1 Development Methodology

The project is considered a Human-Computer Interaction (HCI) tool, so it is developed with input and assistance from the HCI group at VUW. The HCI group choose to use a rapid prototyping approach to software development so that changes can be observed by colleagues and feedback can be provided on each small iteration. One advantage to this approach is that if a feature is in the early stages of implementation and the group identify issues or have feedback, changes can be made before the feature has had a large amount of development time invested into it.

4.2 Meteor Server

Of the two full-stack-frameworks discussed in the background section of this report, the system is built using Meteor rather than Yeoman. This decision was made primarily based on Meteor's emphasis on client-server data synchronisation. While it is true that the integration of a tool such as 'Socket.io' into the Yeoman MEAN stack could enable real-time client-server data synchronisation, it also requires more implementation time. This is described succinctly by Dan Dascalescu, "The client and the server communicate data updates seamlessly and automatically, without you having to write any boilerplate data sync code" [6].

Meteor uses Node as a server back end, this is a means of running JavaScript on the server. It also helps to meet requirement R8, as stated on the Node sever about page, "many connections can be handled concurrently" [1]. Requirement R9 is also met by using Meteor due to its protocol for real-time client updates (DDP).

4.3 Data Processing

As was briefly discussed in the Background of this report, the data are in the form of database table dumps in CSV format. They are also from a relational database.

The objective was to process them in such a way that they would function correctly with MongoDB, the database used in the Meteor framework. "MongoDB is an open-source, document database designed for ease of development and scaling" [4]. As MongoDB is non relational and uses documents, it is not does not support functions which relational databases generally use such as join.

This section gives a brief summary of how the collections were translated using the provided CSV files.

4.3.1 Data Collections

Main Data Collection

The main data collection contains all 1.4 million data points, any occasions where no traffic has passed through the sensors is not recorded and has to be accounted for at runtime. As the system needs to iterate through these by their time stamp to simulate real-time updates, and there is no guarantee of correct ordering in the CSV, it was important to support efficient lookups by the epoch value. This was done by creating a non-unique index on the epoch field of the data table.

An example row from the database data table CSV can be found below, Listing. 4.1 shows its equivalent JSON document representation from MongoDB. For each row in the CSV, the MongoDB main data collection uses a document with similar fields to the original.

route_id	a_to_b	epoch	to_timestamp	match_count	jrnytime_sum
441	0	1431319500	2015-05-11 16:45:00+12	1	117

Listing 4.1: A single document in the main data collection

```
{
   "_id" : ObjectId("57f9a7515d3433f05d28aad4"),
   "a_to_b" : 1,
   "epoch" : 1432690980,
   "route_id" : 281,
   "match_count" : 17,
   "jrnytime_sum" : 1678
}
```

In summary of the most important fields:

- "route_id" is unique to every street
- "a_to_b" is a boolean representing the direction of the street
- "match_count" is the number of cars in the past minute
- "epoch" is the timestamp of when the data was recorded.

Many of these field names are used in other collections and have the same meaning.

Statistics Collection

The statistics collection contains documents specifically for the historical data chart. It is generated at the same time as the main data collection is populated because it uses the same raw dataset, but it is processed in a different way. Viewing a histogram for the most frequent travel periods using the main data collection would be an expensive calculation as all documents for that street would have to be sorted by half hour periods and summed up. As the data collection is populated, the statistics collection is populated using the route_ID, a_to_b, hour, and minute fields as keys, where the total_cars field value is incremented by the current match_count as in the data CSV. This optimisation means that if a user wants to view a histogram for a street, only 48 documents need to be queried if the intervals are every 30 minutes.

Trend Data Collection

The trend data collection contains documents with volume and trend chart data for each street. Each document holds a window of the latest volume and moving average values against epoch time values. The window is of size 20 minutes by default, but can be easily modified through the settings file.

Street Data Collection

The street data collection is a representation of the paths, names, and other information required to render the dynamic map on the front-end. This collection had the most amount of time invested into it and has been given a section of discussion below.

4.4 Complications With The Street Data Collection

Due to the lack of availability of the location data while the early prototype was developed, the implementation of the location data loading and processing has changed drastically.

Initially the data dumps I was provided with were incomplete and missing a way of translating the measured street segments into a series of coordinates or latitudes and longitudes. The only way to identify a street segment was to look at the name field in the street definitions file. I researched how this had been handled in the past by another ENGR489 student, and discovered that they used a matching algorithm for OSM map data against provided street names which could handle basic misspellings [9]. Unfortunately their problem was different to mine as their data had more clues such as connected segments and didn't have the same types of errors.

As a workaround I manually cleaned up the definitions file, this was to remove ambiguity and inconsistency in street name descriptions. This came at the cost of precision and accuracy. For example, observe the following database street names:
"001 to 014 Aotea QuayWaterloo Quay/Customhouse Quay",
"007 - 014 Terrace Off - Hunterr/Jevois Quay ", and
"Well 009 ->NZTA Tawa (S) [251 ->287]"
These suffer from inconsistent titling schemes, missing slash characters, hidden trailing characters, spelling errors, words which aren't street names, no mapping of street numbers to particular streets, and overall often requiring prior domain specific knowledge to interpret. They were modified to the following:
"Aotea Quay", "Jervois Quay", and the final one was removed as I was unable to figure out what it was describing and could not create a substitute street name.

The above cleaned data are missing street numbers, this was for two reasons. It was not possible to tell which street the numbers referred to when a name included two streets, and it was uncertain at that stage whether or not geolocation data would become available via AraFlow. This meant that it was unknown how much time should be invested in geocoding the data precisely.

Carrying out a method similar to that in the aforementioned report, I used Overpass Turbo [17] to obtain a geojson file containing Wellington streets. I then ran a matching algorithm against the cleaned street definitions and the geojson to get hold of a subset of OSM street segments with coordinates. I was then able to associate them with the correct route_ids and insert them into the street data collection in MongoDB. A comparison of the matched segments vs the provided segments from Araflow can be seen in Fig. 4.1.

Basic inspection reveals that overall it has correctly identified many of the streets, but there are also at least two incorrectly identified streets.

As was explained earlier, Araflow eventually provided a database table dump in CSV form containing route ids, latitudes, and longitudes. This table dump is used in conjunction with the route definitions to form the Street Data Collection at server startup. In a relational database the configuration would most likely have kept the definitions and the locations as separate tables, and a join on the route_id field could be used to make a combined table. This is not possible in MongoDB, and is not applicable given that the paths are rendered directly using the JSON output of the collection.

There are some issues I have identified with the schema AraFlow created. The "a_to_b" field makes the assumption that every monitored street definition is bi-directional. This is bad as many streets in Wellington are one-way. Another issue is that some street names include the street direction, such as northbound or southbound, the name is uni-directional but the database expects two records for every "route_id", this results in the issue of placeholder data.

In cases where there is only one direction to record, AraFlow appear to have worked around the schema assumption by creating the other street segment with a single point. As a single point cannot make up a line, they do not show on the map.

4.5 Subscriptions

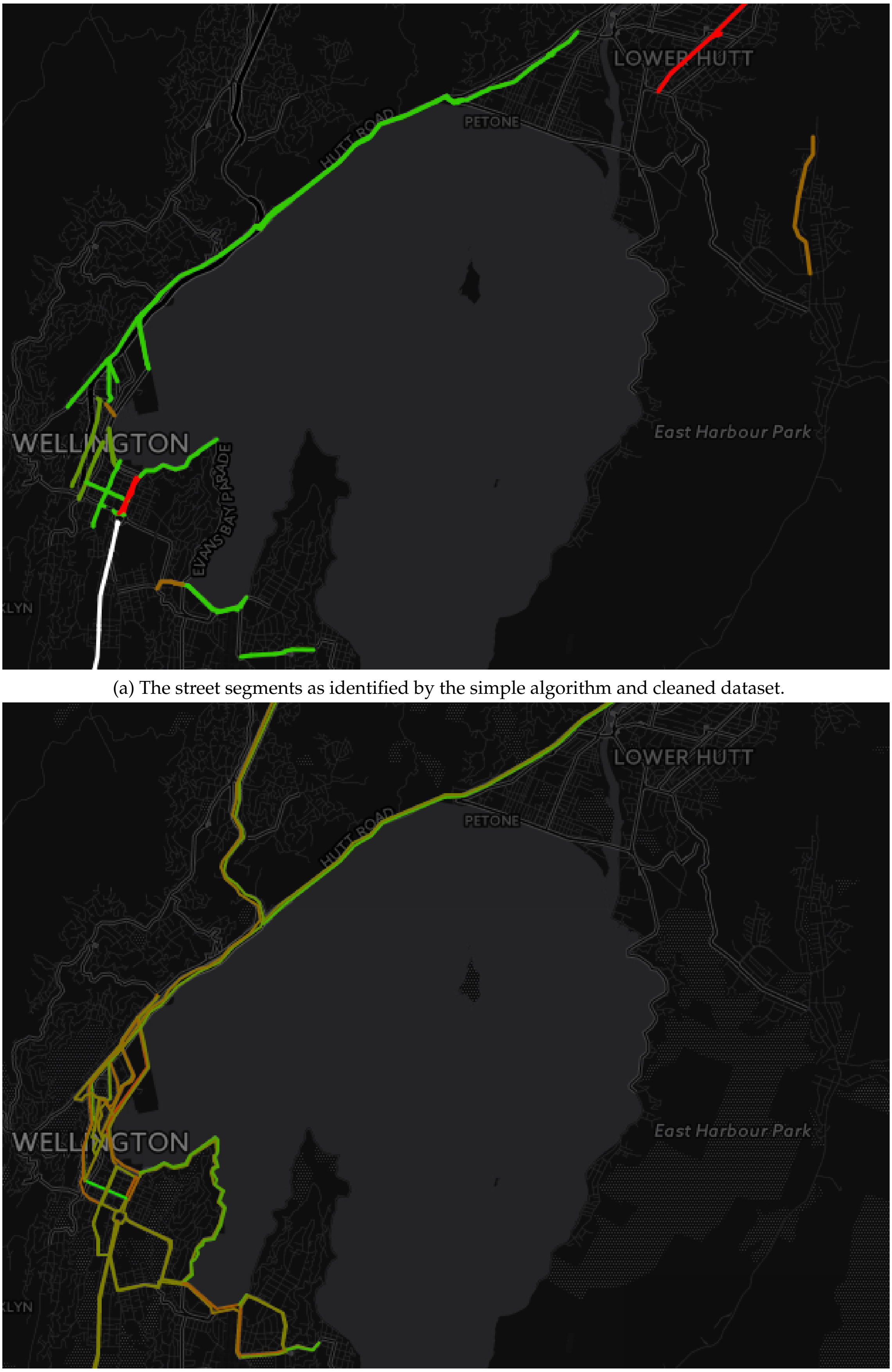
Meteor supports subscription to data collections within MongoDB. What this means is that if a value is updated in the collection, all clients are sent the update via the DDP. A subscription can be parametrised and limited such that a client is only viewing a fraction of the collection, this is performed by creating a database cursor specific to each client. I use this concept in subscriptions to the Trend Data collection.

4.5.1 Trend Data

When viewing the visualisation a client can subscribe to the Trend Data collection by clicking on a street. On every update to the Trend Data collection, the client's cursor query is rerun and the data is returned to the client via the DDP.

4.5.2 Street Data

The Street Data collection also contains a document for every monitored street. However the collection is structured such that it can be directly fed into Angular-Leaflet's path attribute as JSON. Unlike the Trend Data collection, this subscription cannot be parametrised and



(a) The street segments as identified by the simple algorithm and cleaned dataset.

(b) The actual street segements the sensors are measuring, as drawn by Araflow.

Figure 4.1: A comparison of estimated location data vs Araflow provided location data.

filtered. This is intentional as all streets must be available to the client for viewing on the map. The documents contain information such as coordinates, colour, and weight (width).

4.6 Meteor Methods

Meteor supports the ability to asynchronously call functions on the server from the client with an expected return value. This is used for the histogram feature of the system.

4.6.1 Statistics Collection

The statistics collection does not need to update in real time as it is viewing a summary of the historical data for a particular street. As a performance optimisation, rather than using a subscription which would need to be rerun at every server loop, a meteor method was set up to run the query on the Statistics collection and return the result to the client.

4.7 Summary of the server startup sequence

Upon launch, the server will check its database for each table, if there are any tables missing it will attempt to load it in from the CSV files in the private server directory. A challenge while developing the system was to handle efficient loading of the data. At the beginning of this project there were few resources about efficient loading of data from Meteor to MongoDB, the suggestion from the HCI group was to use bulk loading of data. For a long while I couldn't find information on how to use this feature of the Mongo driver. This meant that data had to be loaded in sequentially without driver bulk optimisations. This took approximately 37 minutes on average over three trial runs to load in the entire data CSV. This was not very much of a concern as once the data were loaded, the database performed well.

Information eventually emerged revealing how to use bulk loading. This brought the database loading time down to an average of approximately 4 minutes. Unfortunately this approach is RAM intensive and a system requires at least 2GB of RAM to use this bulk loading option without thrashing.

The overall process of server startup and operation when the database is empty can be described as follows.

- The street definitions file is loaded into the street data collection. The trend data collection is created concurrently.
- The location data are loaded and merged with the street data collection.
- The main data collection is loaded, this can be done using the bulk loading optimisation if the system has enough RAM.
- The server begins asynchronously iterating through the data and updating the street data table which the clients are subscribed to.

If the database tables are already populated at launch, only the final step will occur.

4.8 Front-end Processing and Visualisation

The tool must be accessible via web browsers as defined in requirement R1, the foundations behind the front-end are implemented using HTML, JavaScript, and CSS. These are complimented with various libraries, the most imporant of which is AngularJS which handles the real time reactivity.

The AngularJS framework also helps break down the code into manageable components thanks to its modules design pattern. This particular AngularJS implementation is configured to work within the Meteor framework seamlessly so that it is simple to use Meteor components such as database subscriptions.

4.8.1 Map

The tool uses Leaflet as a JavaScript library for displaying the map. More specifically, it is an Angular packaged version designed to handle reactivity with a changing dataset. Leaflet is an open-source project which uses map tiles generated from OpenStreetMap data. Leaflet supports zooming, panning, and events for selection of paths, these features meet requirement R3.

The data in the street data collection subscription are provided as a series of poly lines, this is in JSON format and is generated on the server to suit the leaflet API without client modification. That is, the data structure is made to suit leaflet's vector format, and the colour is generated by the server rather than the clients. The purpose of this is so that the clients have less processing to do to generate the visualisation, rather than having to read in and create JSON objects on the fly. This broadens the scope of compatible devices, but also puts more processing strain on the server.

4.8.2 Graphs

The graphs as displayed on the front-end are powered by D3.js and a library to implement Angular reactivity and reduced boilerplate code Angular-NVD3 [11]. The data are taken from either the Trend Data subscription or the Statistics Data collection and are passed to the library. Processing is then done accordingly per datum using custom implemented callback functions.

4.8.3 Search

The search function is implemented using the Angular directive Angucomplete, it searches through the name fields provided from the Street Data Subscription.

4.9 Real-time

The server iterates through the main data collection by running a query on the epoch field and incrementing it by 60 on each loop. This accounts for every minute in the dataset as the epoch value is in seconds. The query will return an undetermined number of documents as many streets will not have any data associated with them for that particular minute. Averages, width, and colour are calculated and the Street Data collection is then updated with the new data. Another query is run, to find the streets without data for that minute, zero values are interpolated to ensure that calculations are handled correctly, and again, averages, width, and colour are calculated and inserted into the street data collection.

4.9.1 Street Colour and Width Generation

The colour for every street in the street data collection is calculated on each iteration using the trend value. The trend value is calculated using the difference between the final and initial of the exponential moving average, and is discussed in further detail later in this section. As the street data collection is subscribed to by the clients it updates as a map overlay in real-time. A code snippet of the colour generation is available in listing 4.2.

Listing 4.2: Code for path colour generation. Output is as a CSS RGB string.

```
var red = Math.floor((255 * trend));
var green = Math.floor((255 * (1 - trend)));
var blue = 0;
return "rgb("+red+","+green+","+blue+")";
```

The width is handled in a similar way to the colour, however some further work had to be put in to decide on optimum widths. I ran some test queries on the dataset to find the range of volume values, the result revealed that this is 0-17. As the minimum discernible adjustment in width is 1 pixel, if the width value was used directly then it would be possible to have streets which are 17 pixels wide. This is unreasonable as it would cover far too large an area of the map. Instead I decided to bin the data into ranges, by looking at the distribution, available in Fig. 4.2.

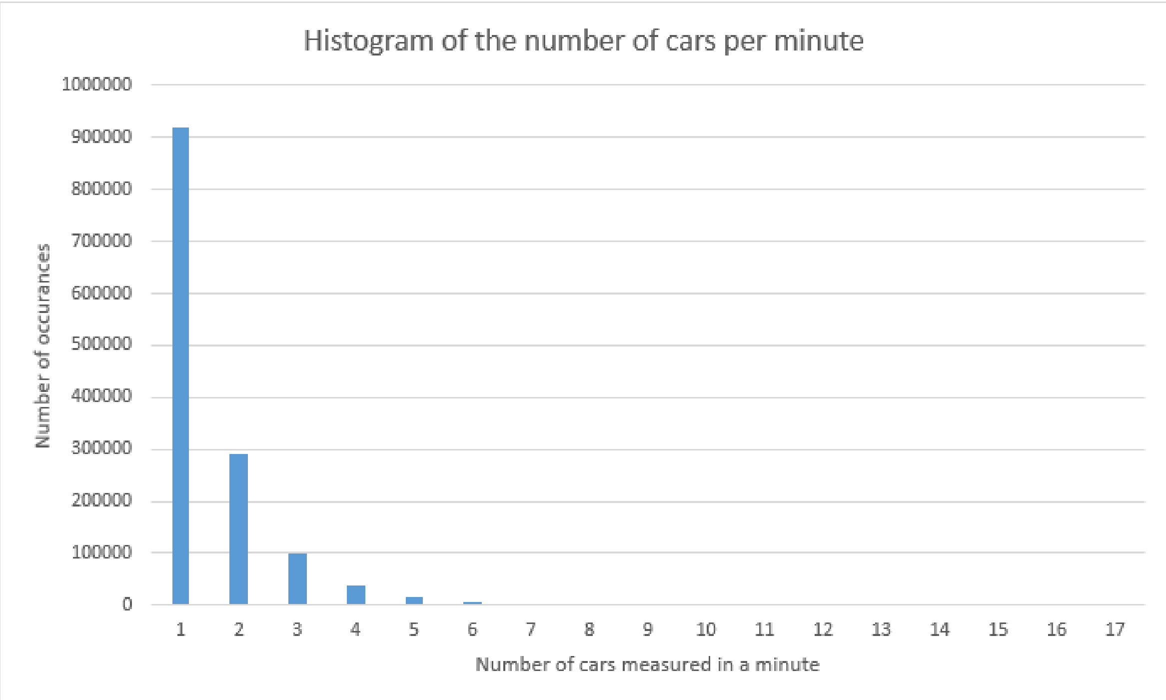


Figure 4.2: A histogram of the frequency of volume values in the dataset.

Bins	Proportion
1	67.0%
2	21.2%
3	7.2%
>3	4.5%

The distribution shows clearly that values above 3 are very rare. This also doesn't account for the number of occurrences of 0. It is possible to estimate the total number of values inclusive of 0 and subtract the total number recorded in the dataset to get an approximation of zero values.

$(((final\ epoch - initial\ epoch) \div 60) \times \#\ streets) - (size\ of\ main\ data\ collection)$

$(((1435665540 - 1430395200) \div 60) \times 84) - (1372043) = 6,006,433$

The above calculation suggests that 6 million zero values have to be interpolated into the data by the server over the course of the simulation. This is a very large figure given that the most frequently occurring value (1) has fewer than one million occurrences.

Based on this discovery, the data were binned into five categories: 0, 1, 2, 3, >3. The width of a street is directly proportional to the volume with the pixel width being the volume plus 1 until it reaches size 4, where the values are then clamped to width 5. The code snippet for this operation can be seen in Listing. 4.3.

Listing 4.3: Code for path width generation. Output is as a Number.

```
function generateWidth(num){
   if(num < 4){
      return num + 1;
   }else{
      return 5;
   }
}
```

4.9.2 Exponential Moving Average

The Exponential Moving Average (EMA) is used for two purposes, to display on the real-time volume graph as a more smooth variation of the raw volume values, and to help calculate the colour for each street. The reason an EMA is used is to apply weighting based on the age of the data, this is so that more recent values have a greater influence on the output of the calculation as with traffic the most recent events are the most important in predictions.

The following function is used to calculate the EMA.

for $t > 1, S_t = \alpha \times Y_t + (1 - \alpha) \times S_{t-1}$ where $\alpha = 0.4$

The alpha value determines the rate of decay where a higher value discounts older observations faster. Y_t is the volume at time t. S_t is the value of the EMA at time t. As $t = 1$ is undefined, it is simply set to the first observed actual value.

The EMA is updated at the same time as the volume for every street. i.e. every loop in the simulation.

4.10 Application Aesthetics

The appearance of the tool went through multiple iterations as part of the rapid prototyping development model. Stage one was purely a prototype and made no attempt to produce a quality user experience. Stage two integrated the Twitter Bootstrap framework, this was primarily to handle layout rather than improve design or user experience. Stage three was the integration of Google's Material Design through the Material Design Lite libraries. The final stage was the theme customisation of the Material Design to suit aspects of the visualisations which needed to remain constant.

When deciding on a front-end framework for displaying the content of the tool to the user, I initially decided to use Twitter Bootstrap, a popular choice among many websites with 17.1% of the top million websites making use of it [18]. The prototype used Bootstrap for a long while as the development focus was around the map interactions.

Towards the later stages of development when the map and charts had finally been implemented, I researched modern user experience and digital design. According to the Nielson Norman Group, a well respected User Experience training group, modern digital design can make use of Flat 2.0 design while also keeping a positive user experience [13]. They state that original Flat Design was aesthetically pleasing, but introduced many usability problems. With the introduction of Flat 2.0, "Designers are finding they can be 'authentically digital' and explore the unique opportunities of the medium without compromising usability" [13]. This is unlike the original Flat Design trend which is known to reduce user efficiency [14].

This lead to the decision to implement Google's Material Design Language in the tool. Google publish a lightweight library to assist with the implementation of Material Components called Material Design Lite (MDL), they also publish another which is specifically designed for AngularJS called Angular Material. Given the size of Angular Material, the unnecessary components, and the performance implications of using yet another large library, MDL was chosen over Angular Material.

Finally, the app was themed for visual consistency and a reduction in the overall colour palette. As the map uses a dark theme from pre-generated tiles, the remainder of the components in the application were themed around these colours. Care was taken to ensure that Flat 2.0 was still applicable and that there was a sense of depth of the components.

4.11 Live Example of the Tool

The completed tool has been deployed to an Amazon Elastic Compute Cloud instance and is available at the following URL: `https://goo.gl/fDF580`. It plays back traffic at 12 times the actual speed, for demonstration purposes.

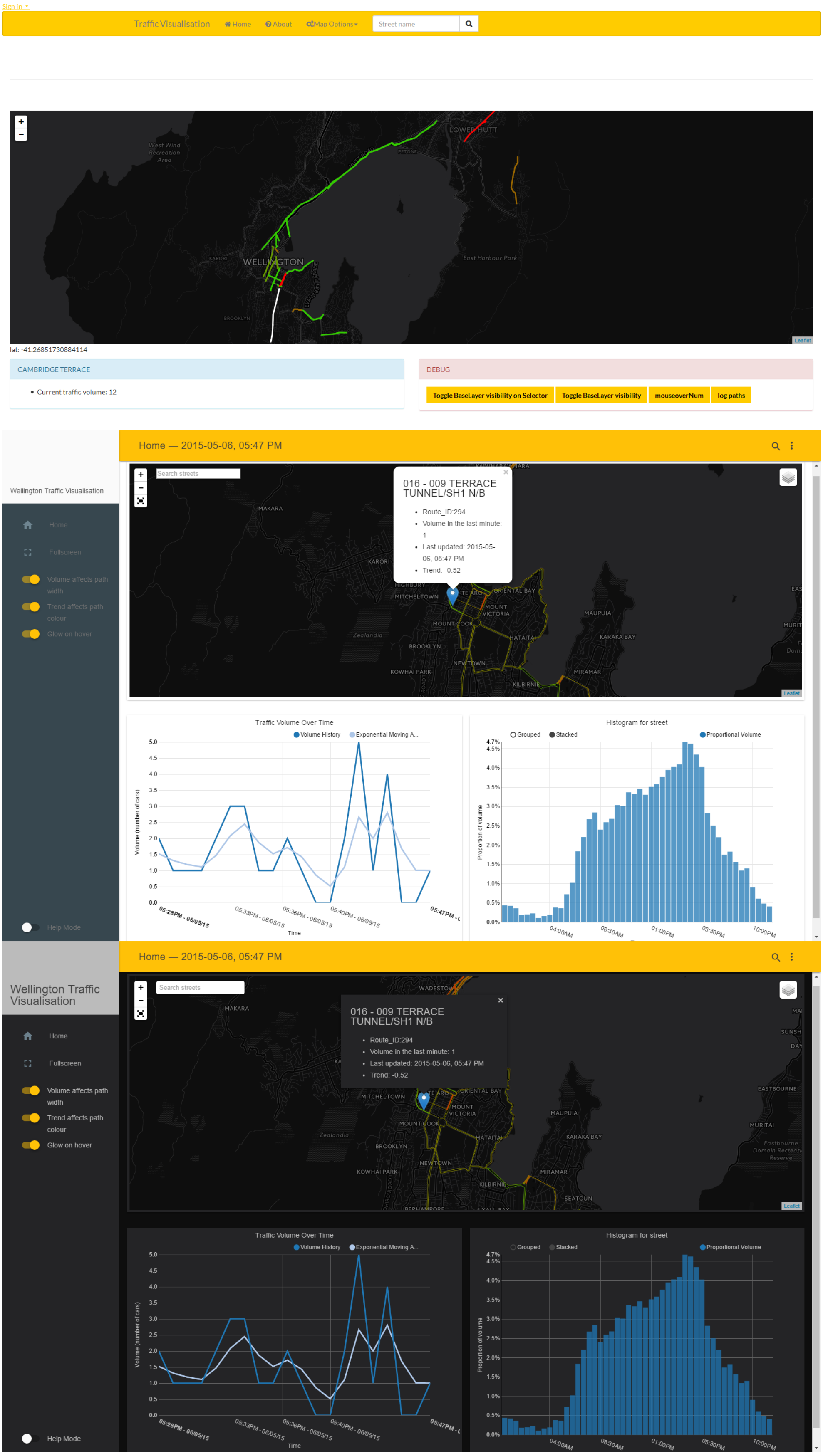


Figure 4.3: The various front-ends produced as iterations of the rapid prototyping process.

Chapter 5

User Testing and Evaluation

An evaluation of the tool has been carried out. The intention was to perform a basic usability study to identify areas needing improvement, and then to apply the most urgent changes. Then following that, perform another evaluation involving a quantitative analysis to identify lengthy operations when using the system. Due to shifting requirements between each review period from the Human Ethics Committee, the ethics approval process took longer than anticipated, and a follow-up evaluation was not carried out.

As a result, this evaluation focuses mostly on user impressions and how easily users can interact with the system without an explanation. This is suited to the primary persona, as a typical user should be able to understand the system with no prior knowledge in the domain. While the interim report discussed the possibility of developing this tool for not just members of the public, but also for domain experts such as street planners, testing against them was decided against in a discussion with my supervisors. As such, there is no separate study on domain experts with the tool.

5.1 Usability Testing

The primary purpose of this evaluation is to identify how well a user without domain knowledge can use the tool which meets the requirements set for this project. It does this by breaking the study into two parts:

- Identification of how well a user can pinpoint areas of traffic congestion and the properties of the street segments involved.
- Identification of areas needing improvement, which are negatively impacting the user experience.

5.1.1 Participants

The initial target test group was to be students throughout the university. Unfortunately ethics approval was denied due to the lack of a flyer, upon the advice of my supervisor, the recruitment strategy was modified to approaching students instead. The scope of students was also limited to those found in the ECS labs at VUW. Through this recruitment process, a total of 12 users were found, and took part in the usability study. All students were male, and aged between 18-30 years old.

5.1.2 Human Ethics Approval

After submitting the initial application to gain human ethics approval for the usability study, two resubmissions were required.

The initial application was filed on the 20th of August. Approval to forward the submission to the Human Ethics Committee (HEC) was approved on the 1st of September by my supervisor Kris Bubendorfer. The application was denied on the 12th of September, and amendments were made. After two rounds of amendments and submissions, HEC approval was obtained on the 12th of October.

5.1.3 Evaluation Method

Configuration

To help speed up the evaluation process, the system was deployed to an Amazon Elastic Compute Cloud 2 (EC2) instance in Sydney. The reasoning for this is so that when ECS students are asked to participate, all they need to do is navigate to the web application in Google Chrome. The lab computers were all using Arch Linux with Kernel 4.6.2-1-ARCH, and the Google Chrome browser version 51.0.2704.84, confirmed by running "uname -r" and "google-chrome -version" in the terminal. Using the same operating system and browser helped to ensure a similar if not identical user experience between users.

Task List

As the system is running a real-time simulation, there can not be an expected answer to many questions as the data are constantly changing. The best way to verify that a user is correctly using the system is to observe them and record if they were correct.

The users are asked to perform a series of tasks, available in Appendix []. It should be noted that the speed section of the survey was ignored, it was included for the ethics application as at the time I was uncertain of whether or not I was allowed to include speed data in the completed system. I recorded their verbal responses and if they were correct on paper as they focussed on interactions with the system. The following are the tasks they were asked to perform:

- **T1** What is the name of a street with high traffic congestion?
  This relates to requirement R2, and Scenario One in the user model.
- **T2** Locate and click Vivian Street on the map. How much traffic volume has there been in the past minute?
  This relates to requirement R3, R4, and R7 if the user needs to search. It is also an example of Scenario One in the user model, and Scenario Three if the user has to search.
- **T3** What is the current traffic trend on Vivian Street?
  This relates to requirement R5, and is also an example of Scenario One in the user model.
- **T4** At which point in time does the traffic volume peak?
  This relates to requirement R6, and uses historical data like in Scenarios Two or Five of the user model.
- **T5** How easy is it for you to identify traffic volume trends?
  This relates to requirement R5.

Finally, the users were asked to provide feedback on the system with the following question: "Can you make any comments on your experience with the system? Please provide feedback to help with future improvements." The intention with this open ended question was to help make improvements before another usability study would be carried out.

5.2 Results

There are both qualitative and quantitative results for the usability study. They are analysed and discussed in more detail below.

5.2.1 Qualitative Results

In this section the project requirements involving the client side of the system are used as a way to group the results of the survey, and provide an analysis of the tool's ability to meet the project requirements.

R2 – visual representation of traffic on a map.

Task T1 asks the users to find a street with high traffic congestion. In all cases except one, the users figured out that a red line indicates a negative trend. Many of the users commented that there should be a key to signify this as well. The users who understood which streets were exhibiting signs of high congestion correctly clicked on the street and could name it.

R3 – Map interactivity, zoom, pan, selection, R7 – textual search

Task T2 causes the users to search the map for a street, many did this by panning and zooming the map successfully. Others spent a short while manually searching, before trying the search function.

R4 – Current traffic volume graph

As a continuation of task T2, once the street was found, users correctly identified the real-time graph for the volume and gave a correct response. This was true for all users except the one who failed to understand the colouring system from before.

R5 – Current traffic trends graph

Task T3 asks the user to provide the current traffic trend for a street. Many users took a while to answer this question, most likely due to the ambiguous phrasing. The number isn't actually available on the graph as it should be, however all users figured out that the number is available on the popover for a street, except for the one user who failed to understand the colouring system from before.

R6 – Historical traffic trends graph

Task T4 asks the user to find when traffic trends peak for a given street, all users were able to successfully identify the peak. For some users it wasn't obvious that they could hover over the histogram to identify the precise time which a value occurred.

5.2.2 Quantitative Results

The quantitative results can be taken from how often users correctly carried out the tasks which were asked of them. The Likert score of how easy they found the identification of trends is also used to quantify the user experience.

Figure 5.1 shows a summary of users who were able to successfully complete the tasks in the Task list. It shows that there was only one user who could not successfully complete the first three tasks.

Figure 5.1: A chart of the number of correctly completed tasks by task number.

The Likert results for the survey reveal that there appears to be no correlation between how difficult a user finds navigation around Wellington and how easily they feel they can identify trends with the tool, as can be seen in Fig. 5.2. There is even a case where a user finds it very difficult to navigate Wellington, but they find the tool useful for identifying trends.

Quantification of comments

I decided to group the suggestions for improvements by category and find the frequency of each suggestion. After reading through the comments, I decided on the following categories:

- Difficult to click on streets
- Needs a key for width/trend
- Help mode needs improvement
- Popover on map pin is not obvious
- Ability to compare streets
- Explanation of Graphs

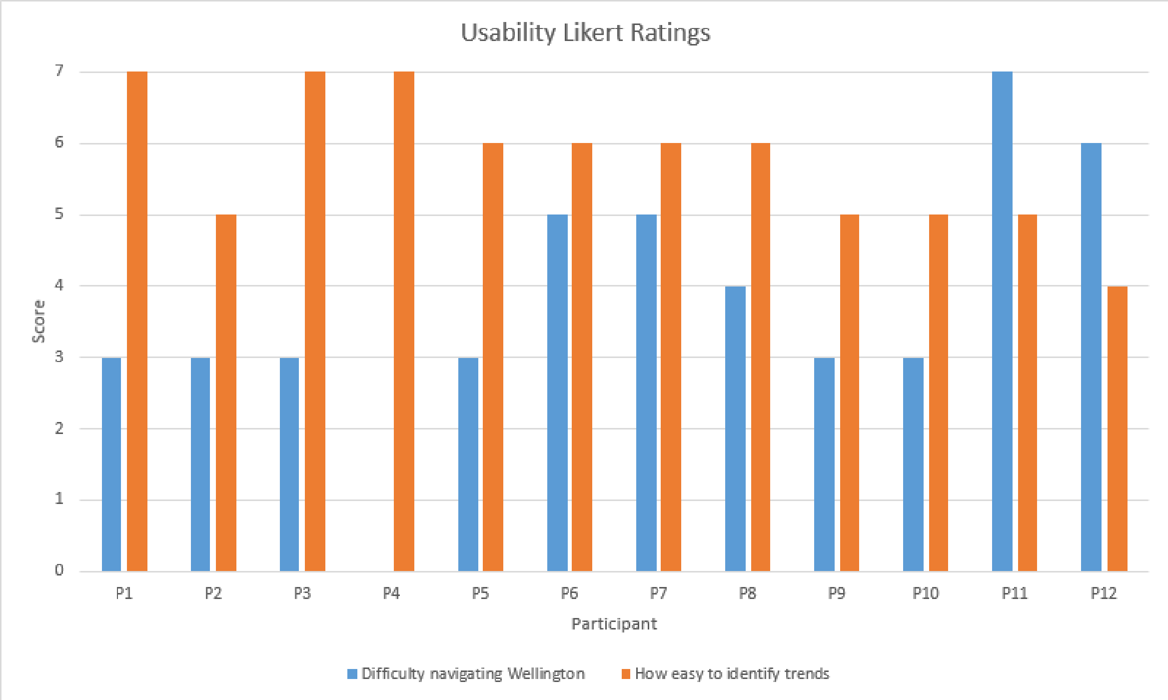


Figure 5.2: A grouped bar chart of the Likert scores by participant number.

- Items should be resized to reduce scrolling
- Filter switches confusing
- Should handle zoom out with double left click

These categories were then plotted to help identify the biggest problems with the system as can be seen in Fig. 5.3.

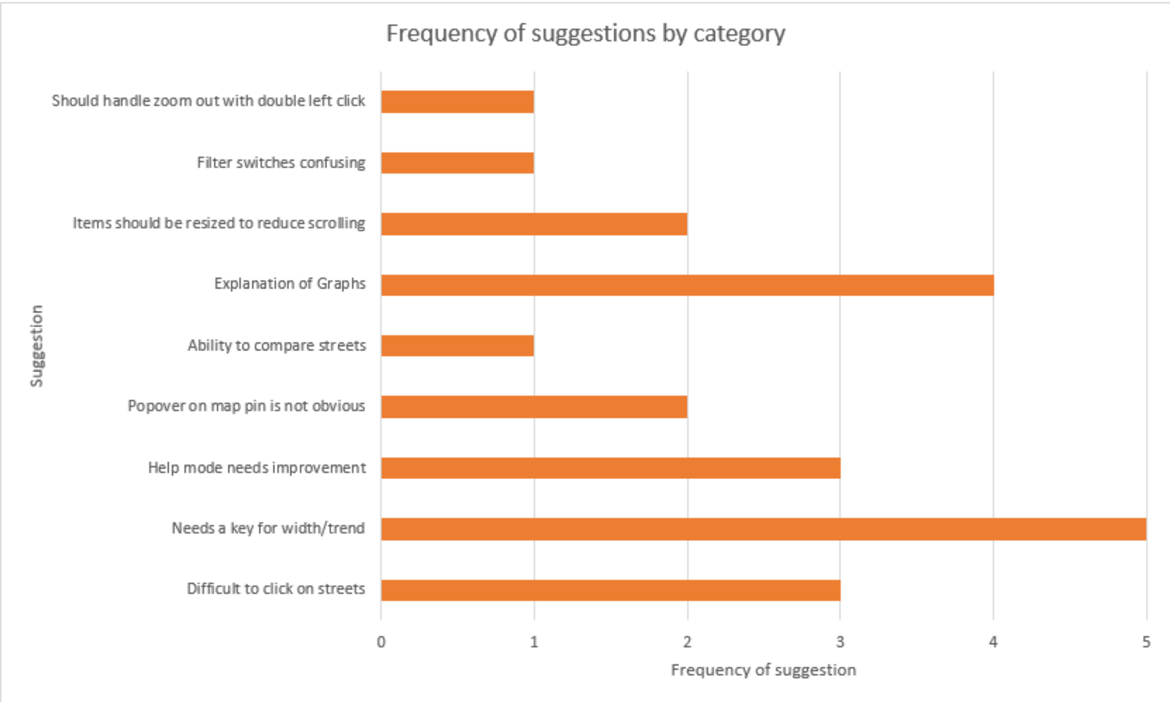


Figure 5.3: A histogram of the suggestions after being categorised.

Chapter 6

Future Work and Conclusion

6.1 Trends and Graphs

The ability to have the application calculate a route based on traffic congestion would be desirable. The information required for this feature is available in the datasets provided by Araflow, the route definitions contain connecting streets. A search algorithm which accounts for the current volume and trend could be employed to find an optimum route. Unfortunately the number of streets in the dataset is so small that there is very good chance that a route does not exist that could help the user.

6.2 Real-time updates

The server is designed in such a way that it could be supplied with data from a 3rd party and stream it to clients instead of using a MongoDB collection to simulate a 3rd party. The tool isn't particularly useful unless provided with real-time data as most of the visualisations are about helping users make decisions about Wellington traffic in its current state.

6.3 Historical Data by date

Similar to the previous ENGR489 project, it would be nice to integrate functionality to represent more in depth historic data. This project was focussed on real-time information, but bundling the functionality of the two projects could result in a richer overall user experience.

6.4 Conclusion

This project followed a rapid prototyping process incorporating a design, prototypes, and an evaluation.

The problem stated that the tool was needed to help road planners, emergency services, and potentially regular members of the public to make decisions around travelling within Wellington. Testing was only carried out on those who had a lack of domain knowledge. It can be argued that if the tool is usable by those lacking domain knowledge, those with domain knowledge are likely to be able to use the tool.

The requirement R1 has been met by using a client-server architecture with real-time synchronisation support to update a series of visualisations. Requirements R2, R3, and R7 are met by using the real-time data with a reactive framework and mapping tool which supports interactivity, the usability study has shown that users understand intuitively how to use the tool, despite not fully understanding the meaning of width and colour. Requirements R4 to R6 are met by using data processing, graph tools, and the real-time synchronisation as stated earlier, the usability study has shown that most users can understand what the data represents. The final requirements are supported by the server architecture and the choice to develop asynchronous implementations.

In summary, this project has resulted in the following:

- The design of a complete system enabling members of the public to access a real-time visualisation of traffic in Wellington streets.
- A prototype of the visualisation system with a simulated real-time traffic data stream to account for non-existent components.
- A user study with an investigation of the resulting qualitative and quantitative data.

Bibliography

[1] About node.js. https://nodejs.org/en/about/, 2016.

[2] AGAFONKIN, V. Simplify.js. http://mourner.github.io/simplify-js/, nd.

[3] ANDY WOODRUFF, T. W. Mbta bus speeds. http://bostonography.com/bus/, nd.

[4] ANON. The mongodb 3.2 manual. https://docs.mongodb.com/manual/, 2016.

[5] BARCELÓ, J., MONTERO, L., MARQUÉS, L., AND CARMONA, C. Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. *Transportation Research Record: Journal of the Transportation Research Board*, 2175 (2010), 19–27.

[6] DASCALESCU, D. Meteor.js vs. the mean stack. https://wiki.dandascalescu.com/essays/meteor_js_vs_the_mean_stack, 2016.

[7] DOANTAM PHAN, LING XIAO, R. Y. P. H. T. W. Flow map layout. http://graphics.stanford.edu/papers/flow_map_layout/, 2005.

[8] GSCHMIDT. Meteor is now on npm. https://github.com/meteor/meteor/pull/516#issuecomment-12919473, 2013.

[9] HYSON, J. Wellington traffic visualisation. Tech. rep., Victoria University of Wellington, November 2015.

[10] KELLEY, E. High performance map interactions using html5 canvas. http://chairnerd.seatgeek.com/high-performance-map-interactions-using-html5-canvas/, 2014.

[11] KRISPO. Angular-nvd3. http://krispo.github.io/angular-nvd3/, 2016.

[12] LABS, S. Global heatmap. http://labs.strava.com/heatmap, nd.

[13] MEYER, K. Flat design: Its origins, its problems, and why flat 2.0 is better for users. https://trends.builtwith.com/docinfo/Twitter-Bootstrap, 2015.

[14] MEYER, K. Long-term exposure to flat design: How the trend slowly decreases user efficiency. https://www.nngroup.com/articles/flat-design-long-exposure/, 2015.

[15] POYNTON, C. Color faq - frequently asked questions color. http://www.poynton.com/notes/colour_and_gamma/ColorFAQ.html, 2006.

[16] SCHÄFER, R.-P., THIESSENHUSEN, K.-U., AND WAGNER, P. A traffic information system by means of real-time floating-car data. In *ITS world congress* (2002), vol. 11, p. 14.

[17] TURBO", O. Overpass turbo. https://overpass-turbo.eu/, 2016.

[18] WITH, B. Twitter bootstrap usage statistics. https://trends.builtwith.com/docinfo/Twitter-Bootstrap, 2016.

[19] YAU, N. 11 ways to visualize changes over time a guide. http://flowingdata.com/2010/01/07/11-ways-to-visualize-changes-over-time-a-guide/, 2010.

Appendix A

Human Ethics Approval

ResearchMaster Enterprise: 5.18.0

User: longuegeof Mr Geoffrey Longuet Logout

Home | **Ethics** | **Postgraduate Forms** | **Research Outputs** | **Help**

Applications

- Create Application
- My Applications
- **Approved**
- For Review
- For Assessment Review
- All Applications

Application ID	Application Title	Status	Primary Investigator	Process Stage	Stage Due Date	Template Name	Date Created
TYPE: Human Ethics							
0000023452	Wellington Traffic Visualisation	Approved	Mr Geoffrey Longuet	Approved (HEC)		Short Form (Category B)v4	31/07/2016

Contact Vic | Disclaimer | About This Site | Request a Change | Site Accessibility

ETH001M

Appendix B

Consent Form

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Wellington Traffic Visualisation

CONSENT TO INTERVIEW

Researcher: *Geoffrey Longuet, ECS, Victoria University of Wellington.*
geoffrey.longuet@ecs.vuw.ac.nz
Supervisors: Kris Bubendorfer, *ECS, Victoria University of Wellington.*
kris.bubendorfer@ecs.vuw.ac.nz
Stuart Marshall, *ECS, Victoria University of Wellington.*
stuart.marshall@ecs.vuw.ac.nz

- I have read the Information Sheet and the project has been explained to me. My questions have been answered to my satisfaction. I understand that I can ask further questions at any time.
- I agree to take part in an interview where my verbal answers will be written down.

I understand that:

- I may withdraw from this study at any point, I have up to two weeks after data collection to withdraw from the experiment if I no longer wish to participate.
- The information I have provided will be destroyed 1 year after the research is finished.
- Any information I provide will be kept confidential to the researcher and the supervisor. I understand that the results will be used for an Honours report and a summary of the results may be used in academic reports and/or presented at conferences.
- My name will not be used in reports, nor will any information that would identify me.
- I would like a copy of the transcript of my interview: Yes ☐ No ☐
- I would like a summary of my interview: Yes ☐ No ☐
- I would like to receive a copy of the final report and have added my email address below. Yes ☐ No ☐

Signature of participant: ________________________________

Name of participant: ________________________________

Date: ______________

Contact details: ________________________________

Appendix C

Information sheet for participants

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Wellington Traffic Visualisation

INFORMATION SHEET FOR PARTICIPANTS

Thank you for your interest in this project. Please read this information before deciding whether or not to take part. If you decide to participate, thank you. If you decide not to take part, thank you for considering my request.

Who am I?

My name is Geoffrey Longuet and I am an Honours student in Software Engineering at Victoria University of Wellington.

What is the aim of the project?

To be able to visualise Wellington Traffic data from a simulated real-time data stream. This is to assist a regular user to identify traffic flow and its trends.
This research has been approved by the Victoria University of Wellington Human Ethics Committee.

How can you help?

This is a directed survey in which participants will be asked to complete a series of tasks using the system as well as questions about what features they can identify in the system. Participant responses will be recorded, i.e. written down. They will also be asked for basic participant information, and comments on how difficult they found the system to use and what they would like changed. Participants are able to refuse to answer any given questions and may withdraw from the study without question within a fortnight of data collection. If you withdraw, the information you provided will be destroyed or returned to you.

The above process will take approximately 10 minutes.

What will happen to the information you give?

This research is confidential. This means that the researchers named below will be aware of your identity but the research data will be aggregated and your identity will not be disclosed in any reports, presentations, or public documentation. However, you should be aware that in small projects your identity might be obvious to others in your community.

Only my supervisors and I will read the notes or transcript of the interview. The interview transcripts, summaries and any recordings will be kept securely and destroyed 1 year after the research ends.

What will the project produce?

The information from my research will be used in my Honours report.

If you accept this invitation, what are your rights as a research participant?

You do not have to accept this invitation if you don't want to. If you do decide to participate, you have the right to:

- choose not to answer any question;
- withdraw from the study within two weeks of this interview;
- ask any questions about the study at any time;
- read over and comment on a written summary of your interview;
- agree on another name for me to use rather than your real name;
- be able to read any reports of this research by emailing the researcher to request a copy.

If you have any questions or problems, who can you contact?

If you have any questions, either now or in the future, please feel free to contact either:

Student:	**Supervisor:**
Name: Geoffrey Longuet	Name: Kris Bubendorfer
University email address: geoffrey.longuet@ecs.vuw.ac.nz	Role: Associate Professor
	School: School of Engineering and Computer Science
	Phone: +64 4 463 6484
	kris@ecs.vuw.ac.nz

Human Ethics Committee information

If you have any concerns about the ethical conduct of the research you may contact the Victoria University HEC Convener: Associate Professor Susan Corbett. Email susan.corbett@vuw.ac.nz or telephone +64-4-463 5480.

Appendix D

User survey

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Wellington Traffic Visualisation

Pre-experiment questionnaire

1. Which age group are you in?
    a. 18-30
    b. 31-49
    c. 50 and older
2. What is your gender
    a. Male
    b. Female
    c. Other (please specify)
3. How long have you lived in Wellington?

4. How difficult do you find it to navigate Wellington streets?

Very Difficult	Difficult	Somewhat Difficult	Neutral	Somewhat Easy	Easy	Very Easy

Study Tasks

The following study tasks are tasks which we will ask you to do, we will observe your actions, and write down your answers.

Volume

1. What is the name of a street with high traffic congestion?
2. Locate and click Vivian Street on the map. How much traffic volume has there been in the past minute?
3. What is the current traffic trend on Vivian Street?
4. At which point in time does the traffic volume peak?
5. How easy is it for you to identify traffic volume trends?

Very Difficult	Difficult	Somewhat Difficult	Neutral	Somewhat Easy	Easy	Very Easy

Speed

1. What is the name of a street with slow traffic?

2. Click on the chosen street. What is the current traffic speed trend?
3. How easy is it for you to identify traffic speed trends?

Very Difficult	Difficult	Somewhat Difficult	Neutral	Somewhat Easy	Easy	Very Easy

Miscellaneous

1. Can you make any comments on your experience with the system? Please provide feedback to help with future improvements.

Appendix E

Ethics Application Form

ResearchMaster

Human Ethics Application

Application ID :	0000023452
Application Title :	Wellington Traffic Visualisation
Date of Submission :	N/A
Primary Investigator :	Mr Geoffrey Longuet
Other Investigators :	Dr Kris Bubendorfer

Research Form

Type of form

The screening questionnaire has initially assessed your application as Category B. Welcome to the Category B form. This is a shorter version of the full form. If your application is confirmed as Category B, your application will not require review at a meeting of the full committee.

If the Committee looks at your application and decides that it does need review at a meeting of the full committee, you will be notified and asked to complete the full Category A form.

Please complete this short form and ensure that all researchers have signed off. Once you have completed this, go to the Action tab and click on 'Submit for review' If you are a staff member, your application will then be automatically forwarded to the committee (if you are a student, it will go to your supervisor for approval first).

If you made a mistake in your screening questionnaire and think that your application does need to be considered at a meeting of the full committee, go to the Action tab and click on 'Switch to Cat A'. You will then be automatically taken to the full form.

1. **IMPORTANT: Please select type of research below and click on 'Save' to access the rest of the form.**

   Research Only

Application Details

2. Ethics category code*

   Human

3. Application ID

   0000023452

4. Committee

   *Human Ethics Committee

5. Title of project*

   Wellington Traffic Visualisation

6. School or research centre*

   Engineering and Computer Science

7. Please add any additional people involved in this project. Ensure that all are listed with the correct role. **If you are a student, do not add your supervisor here: you will be asked to add this information on the next page.**

   Please ensure that only one person is listed as Principal Investigator.

   To add a person, search for their Victoria ID if known, otherwise *either* their first *or* last name (whichever is the most unusual). Click on the magnifying glass to search for results.

   Press the **green tick** at the bottom right corner to save the person record.

   Add anybody who is involved in this project as:

   - Associate Investigator
   - Other Researcher
   - PhD Student
   - Masters Student
   - Research Assistant

   Click on the help button if you are having difficulty adding people to the list.*

1	Given Name	Geoffrey
	Surname	Longuet
	Full Name	Mr Geoffrey Longuet
	AOU	SCPS
	Position	Principal Investigator
	Primary?	Yes

8. Is the principal investigator a student?*

- (●) Yes
- ( ) No

Next time you save this form or move to a new page, a Student Research page will appear after this one. Please complete the two questions on the Student Research page.

Student Research

8a. What is your course code (e.g. ANTH 690)?*

ENGR 489

8b. Please add your primary supervisor (the supervisor who should review this application).

If your supervisor is also the Head of School or the school ethics officer, you will need to discuss with your School who should approve this application as Head of School or delegate. The supervisor and Head of School or delegate **must not be the same person.**

To add your supervisor, search for their Victoria ID if known, otherwise *either* their first *or* last name (whichever is the most unusual).

Press the **green tick** at the bottom right corner to save the person record. *

1	Given Name	Kristian
	Surname	Bubendorfer
	Full Name	Dr Kris Bubendorfer
	AOU	SECS
	Position	Supervisor

8c. What is your email address? (this is needed in case the committee needs to contact you about this application)*

glonguet603@gmail.com

Note that system-generated emails (eg approval notifications) will not necessarily come to this address. System-generated emails will come to the email address stored for you in Student Records. To change the record in Student Records, log into My Victoria, and click on Student Records. You will be able to update your email address from there.

Project Details

9. Describe the benefits and scholarly value of the project*

To be able to visualise Wellington Traffic data from a simulated real-time data stream. This is to assist a regular user to identify traffic flow and its trends.

10. Describe the method of data collection. Note that later in this form, in the Documents section, you will need to upload any relevant documentation such as interview schedule, survey, questionnaires, focus group rules, observation protocols etc. Delays are likely if the interview questions are missing from the Documents section. *

They will also be asked to fill out a questionnaire to provide basic participant information.

There is also a directed survey in which participants are asked to perform tasks and their actions are recorded. These "study tasks" are tasks which we will ask them to do, we will observe them and write down their answers.

11. Describe the objectives of the project*

To produce a tool for visualising traffic data in simulated real-time.

12. Who will the participants in this project be?

VUW students found in the ECS department of the Kelburn campus. I will attempt to have 15 participants.

13. How will participants be recruited?

Requesting survey participation from students in the ECS department.

Documents

14. Please upload any documents relating to this application. A sample Participant Information Sheet and Consent Form are available on the Human Ethics web page.

Please ensure that your files are small enough to upload easily, and in formats which reviewers can easily download and review. To replace a document, click the tick in the column to the right of the document title. A green arrow will appear - click this arrow to upload a new document. To add a new document click on 'Add New Document', at top right of the documents window. Then enter the document name in the box that appears and click the green tick. A green arrow will appear to the right of the file name which allows you to upload the new file. *

Description	Reference	Soft copy	Hard copy
Participant information sheet(s)	Information_Sheet.pdf	✓	
Participant consent form(s)	Consent_Form.pdf	✓	
Questionnaire or survey	Survey.pdf	✓	
Consent Form V2	ConsentFormV2.pdf	✓	
Information Sheet V2	InformationSheetV2.pdf	✓	
Information Sheet V3	InformationSheetv3.pdf	✓	
Survey V2	Surveyv2.pdf	✓	

Key Dates

15. If approved, this application will cover this research project from the date of approval

16. Proposed end date for data collection*

    14/10/2016

17. Proposed end date for research project as a whole*

    11/11/2016

Signoff

If you have any feedback about this online form, please email it to ethicsadmin@vuw.ac.nz

18. This section records sign-off by all other researchers involved in the project (the other team members listed at Q.6). Principal investigators do not need to complete this section - you signoff by submitting the application.

    If co-researchers are external to Victoria University they may be unable to access this site. In this instance, the Principal Investigator may sign off on their behalf. Please upload evidence of the co-researchers' signoff (e.g., a scanned email) to the Documents page.

    To sign off, do ALL 5 of the following 5 steps::

    1. Click on the pencil icon on the far right of the line with your name on it
    2. Click on I Accept
    3. Add the date
    4. Click on the green tick icon on the bottom of the signoff window
    5. Go to the Actions tab and click on 'Notify lead researcher that signoff is complete'

    This question is not answered.

Please add the Head of School or delegate - the person in your School who is responsible for Human Ethics. This person will be notified when your application is approved, and will have online access to the form.

This question is not answered.

Please ensure that you **save your application before submitting it**. Once you have saved your application, to submit it, click on 'Actions' on the left hand side of the screen and then 'Submit for review'.

Applicant Declaration

Declaration for student applicant

I have read the Human Ethics Policy and Guidelines and discussed the ethical implications of my research with my supervisor. I understand my obligations and the rights of the participants and agree to undertake the research in accordance with the Human Ethics Policy and Guidelines. I confirm that the information contained in my responses to the Human Ethics self-assessment form and all other documents pertaining to my application is, to the best of my knowledge, accurate and comprehensive and not misleading.*

☑ Yes

Please complete declaration(s) below.

Supervisor Declaration

Declaration for supervisor

I have assisted the student in the analysis of the ethical implications of this research. As supervisor of this research I will ensure that the research is carried out in accordance with the Human Ethics Policy and Guidelines. I confirm that the information pertaining to this application is, to the best of my knowledge, accurate and comprehensive and not misleading.*

☑ Yes

To submit your application, click on the Action tab and then click on 'Submit'.

Amendment or extension request (available only for approved applications)

17. Are you applying for an extension, an amendment, or both?*

- ◯ Extension
- ◯ Amendment
- ◯ Both an extension and an amendment

This question is not answered.

Please check that you have answered all mandatory questions and have saved the application before submitting your form. Upload any amended documents (e.g. Pariticpant Information Sheet) at Question 40 on Documents page. To submit your form, click on the Action tab and then click on Submit for review

Subsequent Amendments (further requests after initial amendment request has been approved)

If you have already had an extension or amendment in the past, please answer the questions below:

18. Do you have a second amendment request to make?

- ◯ Yes
- ◯ No

This question is not answered.

19. Do you have a third amendment request to make?

- ◯ Yes
- ◯ No

This question is not answered.

20. Do you have a fourth amendment request to make?

- ◯ Yes
- ◯ No

This question is not answered.

Committee use

Assessment of convenor or administrator: should this application be assessed as Category A, or Category B?

- ◯ Category A
- ◯ Category B

This question is not answered.

Assessment of committee reviewer: Do you recommend approval of this application?

- ◯ Yes
- ◯ No

This question is not answered.

Any other comments

This question is not answered.

For convenor/adminstrator: Please note whether this application has been processed as Category A (type 'A'), or Category B (type 'B').

This question is not answered.

Date Approved

12/10/2016

Date Approval To

This question is not answered.