Road Surface Estimator

Final Report

THANAKRIT LEE
26529009
tlee38@student.monash.edu
Monash University

May 7, 2018

Abstract

The purpose of the project is to develop an application that estimate the total surface area of a 1 square km^2 area on the map. There are many different ways to go about implementing a solution. Neural Networks could be use to find roads in a map image, but in this project a simpler solution is implemented.

A server is use to do the surface area calculation by extracting road pixels from an image of the map. The calculation result is send to the application front end client, featuring an elegant and responsive display of user interface to work with. A separation of client and server is implemented so that heavy calculation processing work can be done remotely, faster, more efficient, and doesn't rely on the performance of the user device.

Keywords

Road, surface area, latitude, longitude, google maps

Word Count

5299 words

Contents

1

Introduction

Re-surfacing roads requires the knowledge of the surface area of roads to resurface. Getting the surface area of roads can be made more efficient using aerial/satellite views technology, where the surface area can be calculated. The efficiency of having a total estimated surface area of roads to resurface allows road surface materials to be prepared in a more precise manner, reducing materials wastes and thus reducing total road resurfacing costs.

1.1 Objective

The objective of this application is to allow user to select a point on the map and to work out the total area of roads in the nominated square kilometre, where the specified point is the centre, to give a quote of the road resurfacing cost [Paplinski, 2018].

The web application allows user to designate point on a map for a total estimated surface area of roads in the square kilometre (km^2). The web application will be implemented with Google Maps, which allows for user-to-map interactivity, and for getting data requires for the surface area calculation.

1.2 Requirements

1.2.1 Functional Requirements

- The application is able to display a graphical map.
- The application allow user to interact with the map e.g. click on the map, drag and move around on the map, zoom in and zoom out, change map styling (roadmap and satellite view), add marker, move marker by dragging it, click on marker to show marker info.
- The application allow user to mark the centre point of the square kilometre area on the interactive map.

- The application allow user to input longitude and latitude coordinates.
- The application is able to display the user input coordinates on the interactive map.
- The application is able to calculate the surface area of the road in the selected square kilometre (km^2).
- The application is able to display the calculated road surface area.

1.2.2 Non-Functional Requirements

- The application's graphical user interface is easy to navigate.
- The application is doesn't take long to process the surface area calculation, i.e. low response time.
- The application is responsive.
- The application is able to be use on mobile devices.
- The application provides an easy to understand tutorial on how to use the application.
- The application is hosted on the web, and user can access via the internet.

1.3 Constraints

A constraint for the project is that this project is done as part of a contract for a local council in Victoria, Melbourne, Australia. This means that the roads to resurface must be the correct category of roads classified by the local council (i.e. the declared roads being free ways, arterial roads and some non-arterial state roads) [VicRoads, 2016].

2

Background

2.1 Academic Literature Review

- FIT3036 Project Intro Document [Paplinski, 2018].

  The document gives a brief introduction to what the project will be about and the main specifications for the project. This document was use as the main reference for getting the requirements for the project's application.

- The Google Maps JavaScript API Documentation [Google, 2018e].

  The documentation pages was use to study the API and how to use it in the project. Google Maps API was chosen, because it was one of the recommended technologies to use in the project specified in the project intro document [Paplinski, 2018].

- Road Detection Using Deep Neural Network In High Spatial Resolution Images [Rezaee and Zhang, 2017].

  This was the first article I read about road detection. I initially had the idea that the road detection was going to be done using satellite images. This article provide me with an overall look idea of how road detection was done using satellite images and deep Convolutional Neural Network (CNN).

  Although in the end I didn't end up implementing the application using satellite images and neural network, this article gives me a perspective of the different ways the application could be implemented.

- A Medium [A-Medium-Corporation, 2012] article. Node.js get-pixels: Getting Pixels at Specific Sectors of an Image using ndarray [Levine, 2015]

  This Medium article is a tutorial describing the way to get each pixel of an image using Node.js [Joyent, 2018] and the package: get-pixels [scijs, 2018a]. I use this article to understand how to get each pixel of the map image using get-pixels.

The article was useful in that it is simple, to the point, and provide lots of usage examples, while the get-pixels documentation doesn't provide lots of example for me to understand its usage.

I believe my lack of understanding using the get-pixels documentation was that it doesn't provide any examples for the image pixels data structure [scijs, 2018b], while the Medium article does.

- Google Maps Static Api Documentation [Google, 2018f].

  This documentation was use to understand how to create a static image of Google map. The static image is needed so that it could be processed in the server to detect the road pixels. The documentation is simple to understand, since there isn't any new concept introduce. The map URL property is almost the same as the JavaScript API, but is converted into a static image instead.

- Angular 5 tutorial and documentation [Google, 2018a]

  I've decided to use the Angular (5) framework, because I want to make my development process of the application easier. I've also decided to use Angular, because I've recently started learning the technology and wanted to implement my knowledge.

  The tutorial provide a basic concept and usage of the framework, but I've also use other sources such as YouTube videos and online tutorial.

  The documentation is easy to understand as long as one have a grasp of the previous pre-requisite concepts.

- A YouTube video tutorial on Angular Google Maps (AGM). Google Maps & Angular | ANGULAR SNIPPETS [Academind, 2017].

  The video shows how to get AGM setup and goes through the basic usage of the package: showing the map on a html page, adding markers to the map, and centring the map on a latitude/longitude coordinates.

- A video and online tutorial on how to use material design [Google, 2018h] bootstrap [Twitter, 2018] with angular by using the library MDBootstrap [MDBootstrap, 2018]. Material Design Bootstrap 4 and Angular 5 Tutorial - MdBootstrap [coursetro, 2018].

- Bootstrap 4 documentation [Twitter, 2018].

  The documentation is use for when I've encounter Bootstrap related problem and the MDBootstrap documentation isn't clear or have enough information.

- Google Maps APIs Styling Wizard [Google, 2018d].

  The styling wizard is a tool use for creating a style property of a Google Maps. The wizard shows options such as colours and visibility of elements

on the map. I've used this styling wizard for creating an appropriate style for my map where the roads are shown clearly and everything else is transparent. This style allows my application to classify the road pixels more efficiently.

- Google Chrome DevTools [Google, 2018c].

  "Chrome DevTools is a set of web developer tools built directly into the Google Chrome browser. DevTools can help you diagnose problems quickly, which ultimately helps you build better websites, faster." [Google, 2018c].

  I've use the tools for debugging my web application, and for testing the mobile view of the application.

  The debugging feature that I've mainly use is the console output. I've use the console to log variables in my application to see if they're the expected variables, and to read the error logs when my application is bugged.

  I've discovered the mobile view only recently and thought that it would help me develop my application better to also be use on the mobile platform. The mobile view shows the application running on mobile screen device. The application is compacted to fit the screen. There are various mobile devices to choose from such as the iPhones, iPad, Galaxy, Pixel etc.

2.2 Research

- MEAN stack.

  During the 2017 December break I was studying the MEAN web application stack. "MEAN" stands for MongoDB ExpressJS Angular NodeJS. A "stack" is a set/combination technologies that are use to together to create a web application. MongoDB is use as the application database. ExpressJS via NodeJS is use as the API server to connect the client-to-server-to-database. Angular is use as the interface for the client-side.

  My understanding of how a web application connects with each other have given me insight into how to better develop a web application, with better structure and business logic.

  The MEAN stack have influenced me in choosing the technologies I have chosen for this project (Angular, NodeJS, ExpressJS).

2.3 Risks

Risk List		
ID	**Risk**	**Trigger**
R0	Project scope creep	Project scope wasn't defined properly.
R1	Software libraries, modules or frameworks changes and is now incompatible with the project	The software developers making changes to the software.
R2	Working computer breaks	Dropping the computer, spilling water on it.
R3	Project going over schedule	Improper time management of the project, not regularly updating and checking on the project schedule Gantt Chart.s

Risk List		
ID	**Probability (low/medium/high)**	**Impact (low/medium/high)**
R0	low	high
R1	low	medium
R2	low	high
R3	medium	high

Risk List	
ID	**Mitigation Plan**
R0	Make sure to get most (if not all) the project requirements from the project stakeholders. If a compulsory requirement is added, make it a priority over lower priority requirements.
R1	Use versioning tool to keep track of the version of the libaries the project is using.
R2	Take good care of the computer. Backup the project code regularly. Use Git remote repository (such as GitHub[GitHub, 2018]) to backup the project and regularly make commits and push to the remote repository.
R3	Regularly check the project schedule to see where the project progress is at and compare it to the current time to see if the project is behind, ahead or on schedule. Make a commitment to at keep the project on schedule. If the project is behind schedule then work over time to catch up to the project schedule time line.

The only risk that was actually encountered in the project was R0: Project scope creep.

During the project a new requirement was added. The requirement was to add a function to the application that allow the user to change the view of the map from

"normal" view to a view that only shows roads. This requirement was added to give user more interactivity with the map, which in turn increase user experience (UX).

This scope creep was obviously the result of not property defining all the requirements needed in the requirement gathering phase. The risk trigger hypothesised in the risk document (from the project proposal) [Lee, 2018a] prove to be correct.

The mitigation I did for the risk was that I accepted the risk that a new requirement was added to the project, and I assign a priority level to the new requirement in comparison to other requirements. I then work on implementing the highest priority level requirement then moving on to the lower ones. I used the same risk mitigation method as described in the risk document.

Instead of having a high impact on the project as hypothesis, the impact of scope creep was rather low. There was no effect at all to the schedule of the project. I further hypothesis that the impact of the scope creep might have been low because of the scale of the introduced requirement. The introduced requirement was a small feature and only require 1-2 hours to implement and test, no where large enough to effect the final project schedule.

2.4 Resource Requirements

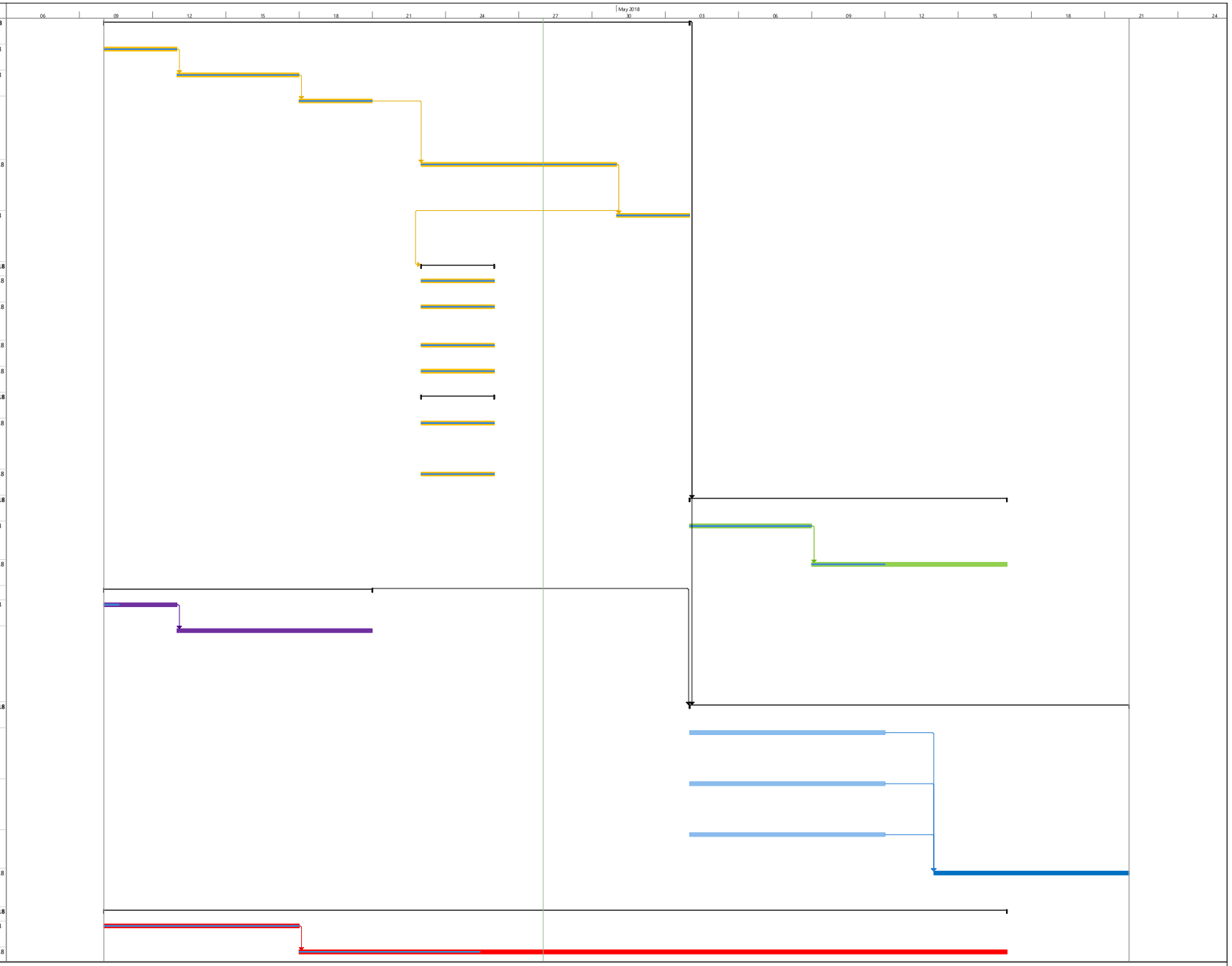
- Computer for developing the project.
- Internet access to download and use libraries the project depends on, and to access the libraries documentation.
- Angular 5.2.0[Google, 2018a], a front end framework use for creating a good user interface and implementing business logic.
- Node.js 8.9.4[Joyent, 2018], a JavaScript runtime for use with Express.js to create a server API and implement the back end of the web application.
- Express.js 4.16.3[StrongLoop, 2018], for creating the API for communication between client and server.
- MDBootstrap 5.2.3[MDBootstrap, 2018], use the CSS framework for faster development of the app.
- AGM 1.0.0[Holstein, 2018], a wrapper of Google Maps JavaScript API for Angular 2+.
- Google Maps JavaScript API [Google, 2018e]. The JavaScript API is use to display a Google map on the web application for user to interact with.

- Google Maps Static API [Google, 2018f]. The Static API is use to create a static image URL from the JavaScript API so that the image could be process to find the roads area.
- get-pixels [scijs, 2018a]. A NodeJS package that get all pixels from an image. I use this package to extract the pixels from the Google map static image for classifying the road pixels.
- Syntactically Awesome Style Sheets (SASS) [Sass, 2018]. A style sheet language that is use with MDBootstrap to style html pages.
- Microsoft Virtual Studio Code (VS Code) [Microsoft, 2018]. An open source code editor that I use to edit the web application code.
- Google Chrome [Google, 2018b]. A internet browser for testing the functionality and user interface of the application.
- Goole DevTools [Google, 2018c]. I use the tools for debugging my application.
- Git [Torvalds, 2018]. A version control system (VCS) that I use to keep track of the versions of the application. A good practice for software development is to use VCS while developing the software. Using git it has allow me to easily revert my application back to the previous version, and to easily add in bug fixes into the application.
- GitHub [GitHub, 2018]. A Git remote repository. I use GitHub to host my Git repository remotely on the GitHub server for backup. I want to have a remote backup repository to mitigate the risk that something might go wrong with the computer I'm working on.
- GitKraken [Axosoft, 2018]. A Git Graphical User Interface (GUI). I use GitKraken to streamline the application development process so that I can develop the application faster.
- Heroku [Salesforce, 2018]. A cloud platform as a service (Paas) [Wikipedia, 2018] use as a web application deployment model. I use Heroku to host my web application.

2.5 Project Tasks

The tasks required to be done for the project are listed in the Gantt chart below.

ID	Task Mode	Task Name	Duration	Start	Finish
1		**Application Development**	**6 days**	**Tue 10/04/18**	**Thu 03/05/18**
2		Create MEAN app spike	1 day	Tue 10/04/18	Thu 12/04/18
3		Deploy the spike app	1 day	Fri 13/04/18	Tue 17/04/18
4		Restructure app to MEAN app structure that is fit for deployment	1 day	Wed 18/04/18	Fri 20/04/18
5		Implement server side calculation function	2 days	Mon 23/04/18	Mon 30/04/18
6		integrating all components of the application together	1 day	Tue 01/05/18	Thu 03/05/18
7		**Map Functions**	**1 day**	**Mon 23/04/18**	**Wed 25/04/18**
8		Move marker by dragging	1 day	Mon 23/04/18	Wed 25/04/18
9		Show marker info by clicking on marker.	1 day	Mon 23/04/18	Wed 25/04/18
10		Button to show road only view	1 day	Mon 23/04/18	Wed 25/04/18
11		Button to show normal view	1 day	Mon 23/04/18	Wed 25/04/18
12		**User Interface**	**1 day**	**Mon 23/04/18**	**Wed 25/04/18**
13		Display time taken for calculation of result	1 day	Mon 23/04/18	Wed 25/04/18
14		Write a tutorial page	1 day	Mon 23/04/18	Wed 25/04/18
15		**User Documentation**	**3 days**	**Fri 04/05/18**	**Wed 16/05/18**
16		Plan documentation structure	1 day	Fri 04/05/18	Tue 08/05/18
17		Write documentation	2 days	Wed 09/05/18	Wed 16/05/18
18		**Write Test Plan**	**3 days**	**Tue 10/04/18**	**Fri 20/04/18**
19		Brainstorm test cases	1 day	Tue 10/04/18	Thu 12/04/18
20		Make improvement adjustment to the test plan in the project proposal	2 days	Fri 13/04/18	Fri 20/04/18
21		**Testing the Application (QA)**	**4 days**	**Fri 04/05/18**	**Mon 21/05/18**
22		Test front end functional requirements (and bug fixing)	2 days	Fri 04/05/18	Fri 11/05/18
23		Test front end non-functional requirements (and bug fixing)	2 days	Fri 04/05/18	Fri 11/05/18
24		Test back end requirements (and bug fixing)	2 days	Fri 04/05/18	Fri 11/05/18
25		Test the whole system (and bug fixing)	2 days	Mon 14/05/18	Mon 21/05/18
26		**Write Final Report**	**9 days**	**Tue 10/04/18**	**Wed 16/05/18**
27		Plan structure of report	2 days	Tue 10/04/18	Tue 17/04/18
28		Write Report	7 days	Wed 18/04/18	Wed 16/05/18



Project: schedule-gantt-chart
Date: Sat 28/04/18

Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress

As of writing the final report I'm behind schedule on my progress of the:

- Final report.

  I'm currently behind my final report schedule progress by 2 days, but I'm expected catch up.

- Test plan

  I haven't started on rewriting my test plan from the project proposal yet.

- Testing the application

  Although I haven't officially tested the application properly yet, I have had my application gone through usability tests and the result were good. I've also gotten criticism from the users and have made improvements to my application user interface to make my application more user friendly.

- Application documentations.

  I have only documented my application by 50% so far.

My plan to get my project back on schedule is to first complete the final report, then complete the test report, test my application, and then document my application.

I know that the project will be back in schedule in a week time.

3

Method

I've used the Agile software development framework methodology in developing the project software product. For each sprint (week) I make small increments to the application. The increments are the functional and non-functional requirements. As I implement the requirements to the application I test each requirements to make sure that they are working as expected.

The project schedule Gantt chart figure does not represent the Agile development life cycle I went through, but is a good basis as a list of tasks required for the project.

I've made changes to the sequence diagram (figure (3.2)) and software architecture diagram (figure (3.3)). I added the AJAX (Asynchronous JavaScript And XML) [Mozilla, 2018] technology to the diagrams. AJAX is the interface between the client HTTP call and the actual HTTP call that AJAX does asynchronously. AJAX allows the application to asynchronously calls HTTP request, which provides the advantage of having higher performance web application, because the client don't have to wait (synchronously) for the HTTP response.

I've added AJAX to the diagrams now in the final report because I've only learned about the technology while I was developing the project's application. Developing web applications is still a new territory for me.

3.1 Internal Design

The web application will have a client-side and server-side. Angular 5 is use as the front-end client-side and Node JS Express is use as the back-end server-side. The client and server will communicates with each other through HTTP. A server is implemented in this web application so that it is properly structured and follow the separation of concerns principle. The front-end take cares of the business logic (displaying data, taking user inputs) while the back-end take cares of the calcula-

tion and processing of data.

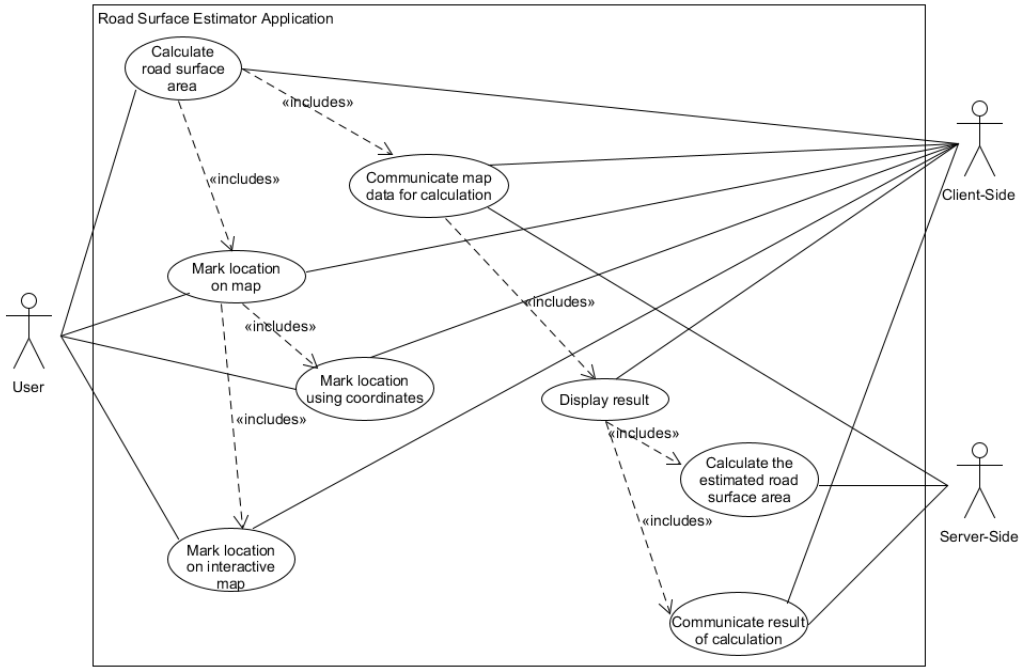


Figure 3.1: Use Case diagram

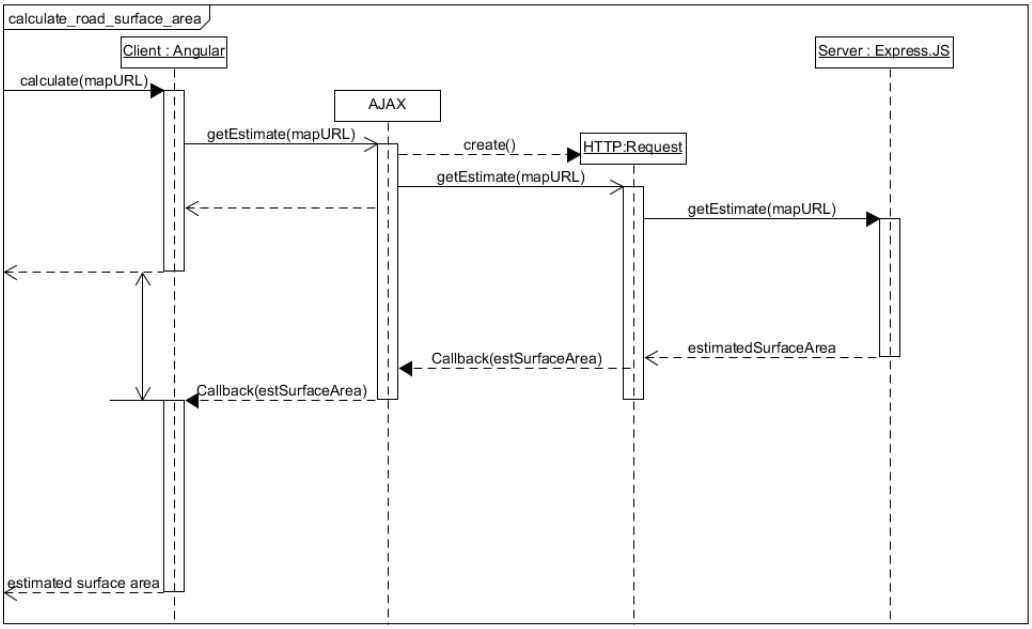


Figure 3.2: Sequence diagram

3.2 Software Architecture

The web application will use the Angular 5 web framework to implement the front-end client-side of the application. The back-end server-side will be implemented using Node JS Express, and the communications between the server and the client will be done through HTTP REST. Using the Google Maps API, the client creates a map static image URL and pass it (through HTTP) to the server for processing. The server will process the image and calculate the total surface area of roads in the map, and return the result back to the client as a response.

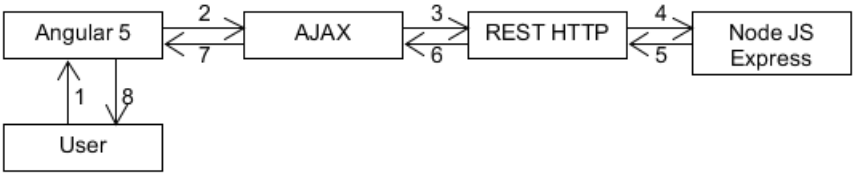


Figure 3.3: Software Architecture diagram

3.3 Algorithms

In processing the map image to calculate the estimated total surface area of roads in a nominated square kilometre, I've use get-pixels (a NodeJS package) to extract all pixels from the map image.

With all pixels from the map I run through all pixels from the top left corner of the map to the bottom right corner of the map once. While running through each pixel the pixel is check if it is visible or not, because only road pixels in the image are visible (this was done by styling the map using the styling wizard). If the pixel is visible, it is a road pixel and a counter is incremented by 1, if the pixel is not visible then do nothing. Once all pixels have been checked, the count of road pixels are multiplied by the square metre value of each pixel, given by the formula [Broadfoot, 2011] :

$$squareMetrePerPixel = metresPerPixel \times metrePerPixel$$

$$metresPerPixel = \frac{156543.03392 \times \cos(lat \times \frac{\pi}{180})}{2^{zoom}}$$

where

- *lat* is the latitude is the nominated square kilometre centre coordinates latitude.
- *zoom* is the current zoom level of the map.

After the multiplication the result is the estimated total surface area of all roads in the nominated square kilometre in square metres. To convert to square kilometres divide the result by 1000000.

The activity diagram (figure (3.4)) shows the algorithm steps described above.

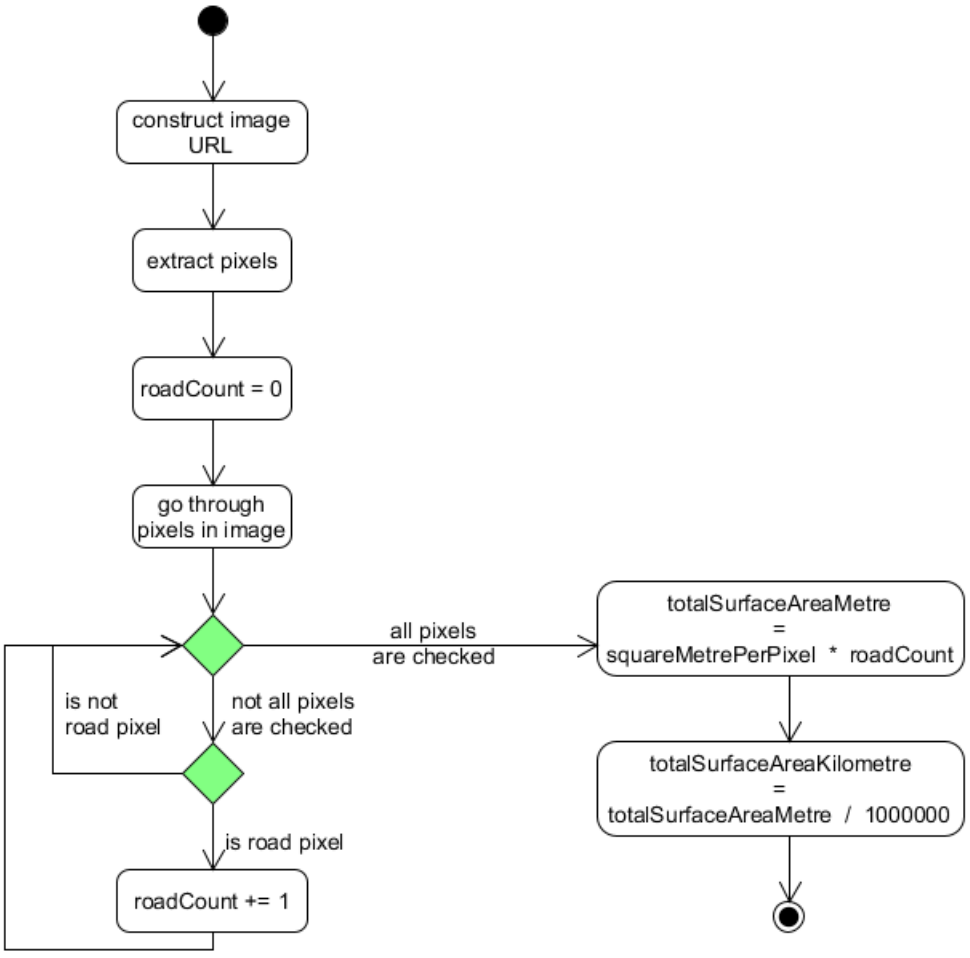


Figure 3.4: The total surface area calculation algorithm.

4

Results

All project requirements, both functional and non-functional are implemented in the application within the project schedule.

There was a slight project scope creep. I introduce a new functional requirement, where the application provides a button for user to change the style of the map. This scope creep didn't affect the project schedule, since I've almost got all requirements done at that time.

The outcome of the project was a lot better than I expected. I put a lot more emphasis on developing a good user interface than I originally intended. However, the effort I put into developing the user interface wasn't taken away from the other requirements. While working on various requirements during the project life, I continuously check the project's Gantt chart to see where I'm at in the project schedule. By checking the project schedule I remind myself where I'm at and what need to be done.

The quality of the project is acceptable. All the requirements are implemented in the application and are satisfied.

I'm satisfied with the outcome of the project, however I believe I could have done more with the user interface. I thought that the interface seems a little too static and needed some animations to make user experience better and more fun to use the application.

4.1 Externally Observable Features

The three types of input in the application are:

- Google Maps
- Buttons

- Form Inputs

Google Maps

User can interact with the Google Maps using their mouse pointer and the scroll wheel. On mobile device, user interact using their fingers.

- Computer Internet Browser
  - click on the map to move set marker at clicked coordinates.
  - click on marker to bring up marker info. The info is the marker's coordinates on the map.
  - click and hold on a marker to drag it. This allow user to drag the marker around.
  - using the scroll wheel to zoom. On the computer internet browser user must also hold "control/command" on their keyboard while scrolling wheeling to zoom.
  - user can click and hold on the map to move the map around.
- Mobile Device Internet Browser
  - touch the map to move the marker to new place.
  - touch the marker on the map to show marker coordinates info.
  - touch and drag fingers on a marker. This action drag the marker across the map with the finger.
  - using two fingers and making a pinching and un-pinching motion, the user can zoom the map out and in, respectively.
  - using two fingers and dragging them across the map will move the map around.

Buttons

The button inputs are:

- a button that change map view type, showing the map in either roadView or satelliteView.
- a button that change map style, changing the style between the default style and a style which shows only the roads.
- a button that shows the calculation results and the elapsed time.

There are other input buttons such as the navigation bar links, the links in the footer, and images with links on them.

Form Inputs There is a form with 2 input fields in the application. The input fields takes input value of type number. The input represents the latitude and longitude coordinates of the marker on the map.

The only output from this application is the result of the calculation and the elapsed time of the calculation on the server. The result shows the surface area of roads in km^2 in the 1 km^2 area around the marker coordinates. The elapsed time shows the elapsed time of the calculation in seconds (s).

4.2 Performance

The complexity of the road surface area calculation is $O(n)$. The algorithm runs through the pixels array once, and does the area calculation afterwards. The algorithm is describe in the *algorithm chapter*.

The performance result I've chosen to record are of the web application hosted Heroku [Salesforce, 2018] and accessed via the internet on Google Chrome [Google, 2018b] internet browser. I've chosen the remotely hosted version over a locally hosted version, because I wanted to test the application in a real world situation. The user would most likely use the application provided over the internet rather than locally installing the application and using. It's far easier to use it over the internet, since user only have to go to the application URL address in the internet browser.

Figure (4.1) and figure (4.2) shows a line graph of the elapsed time of the calculation versus the total surface area of the roads. The result shows that there's a slight increase in elapsed time as the surface area increases. Generally the server calculation elapsed time is under the 0.25 seconds mark, and the overall calculation plus the network communication elapsed time is under 0.5 seconds.

The Thailand Data table shows the data points for the Figure (4.1) line graph and the Melbourne Data table shows the data points for the figure (4.2) line graph. Summing up the both server time and overall time, then dividing by the number of data points result in the mean average:

- Thailand Data:

  Server time mean average: 0.1818 seconds

  Overall time mean average: 0.4881 seconds

- Melbourne Data:

  Server time mean average: 0.192 seconds

  Overall time mean average: 0.6201 seconds

There were outliers in the data. The outliers are DT9, DM6, and DM8. These outlier shows the overall time being above 1 seconds, while the server time stays

relatively low in comparison. My hypothesis for the occurrence of the outliers is that the Heroku server which I've hosted the application on or the internet browser that I'm using is affecting the network communication time. I've also notice that the these outlier occurs when the internet browser is idle for more than 60 seconds.

Figure (4.3) and Figure (4.4) shows line graphs of the performance data without the outliers. With no outliers, the graphs suggest that there's a correlation between the surface area and the calculation time. As the road surface area increase, so does the calculation time. My hypothesis for this correlation is that the calculation involve multiplication of each road pixels, and with higher road pixel counts (more road surface area) the multiplication required increases. The mean average of the data points with no outliers are:

- Thailand Data:

  Server time mean average: 0.1731 seconds

  Overall time mean average: 0.42367 seconds

- Melbourne Data:

  Server time mean average: 0.176625 seconds

  Overall time mean average: 0.42425 seconds

Without the outliers the mean averages are lower.

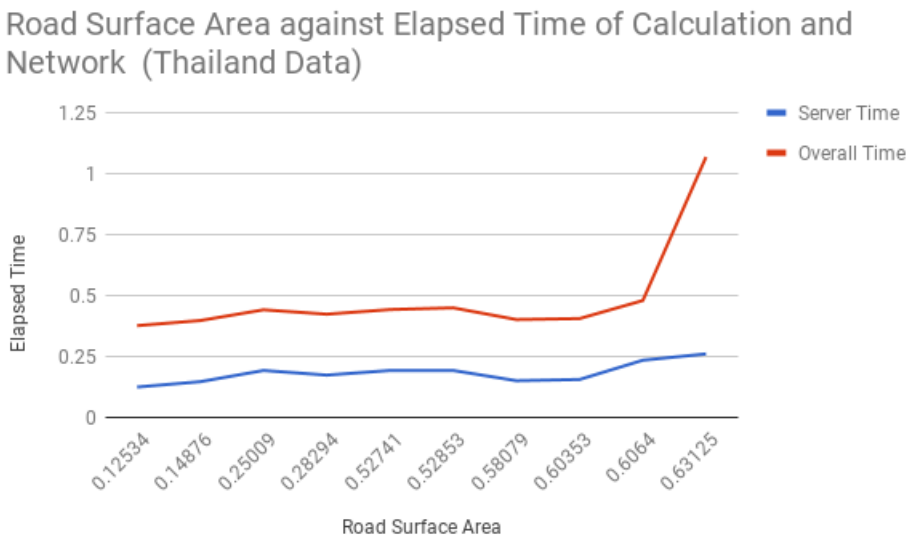


Figure 4.1: Road Surface Area against Elapsed Time of Calculation and Network (Thailand Coordinates)

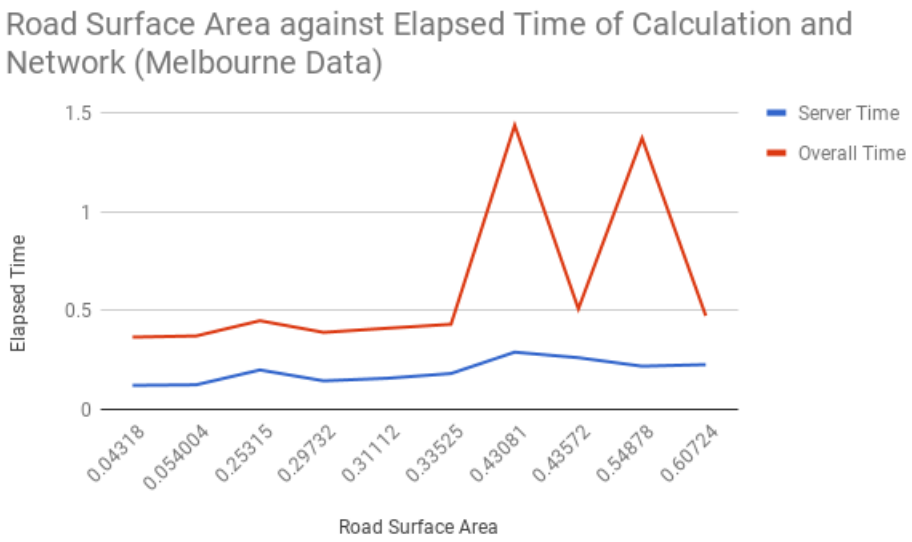


Figure 4.2: Road Surface Area against Elapsed Time of Calculation and Network (Melbourne Coordinates)

Thailand Data					
ID	**Latitude**	**Longitude**	**Result**	**Server Time**	**Overall Time**
DT0	13.91105564	100.4813004	0.12534	0.124	0.376
DT1	13.69002205	100.7501221	0.14876	0.146	0.397
DT2	13.6788381	100.5640411	0.25009	0.192	0.441
DT3	13.69017971	100.5779457	0.28294	0.173	0.423
DT4	13.79003496	100.7247162	0.52741	0.192	0.442
DT5	13.71252776	100.6244659	0.52853	0.192	0.449
DT6	13.71266717	100.6443787	0.58079	0.15	0.401
DT7	13.70065954	100.6938171	0.60353	0.155	0.405
DT8	13.74535151	100.5509949	0.6064	0.234	0.479
DT9	13.76135851	100.5976868	0.63125	0.26	1.068

Melbourne Data					
ID	**Latitude**	**Longitude**	**Result**	**Server Time**	**Overall Time**
DM0	-37.94177554	145.2890396	0.04318	0.121	0.365
DM1	-37.89789987	145.3683472	0.054004	0.124	0.371
DM2	-37.88183827	145.0365277	0.25315	0.199	0.448
DM3	-37.86298759	145.2200317	0.29732	0.144	0.389
DM4	-37.85496375	145.0688001	0.31112	0.157	0.41
DM5	-37.88508993	145.004942	0.33525	0.181	0.43
DM6	-37.81249633	144.9469185	0.43081	0.289	1.436
DM7	-37.84321388	144.9910375	0.43572	0.261	0.508
DM8	-37.81379155	144.9520703	0.54878	0.218	1.371
DM9	-37.8283008	144.9654599	0.60724	0.226	0.473

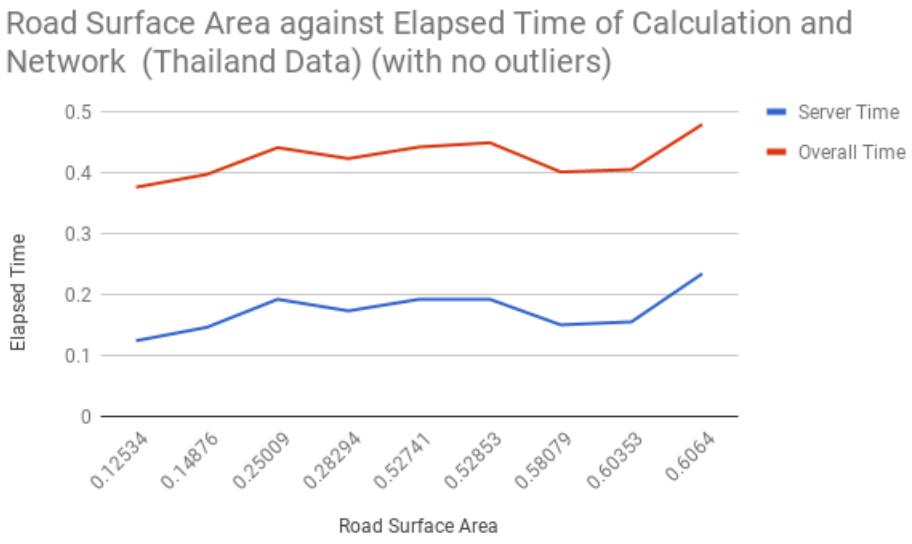


Figure 4.3: Road Surface Area against Elapsed Time of Calculation and Network (Thailand Coordinates) (No Outliers)

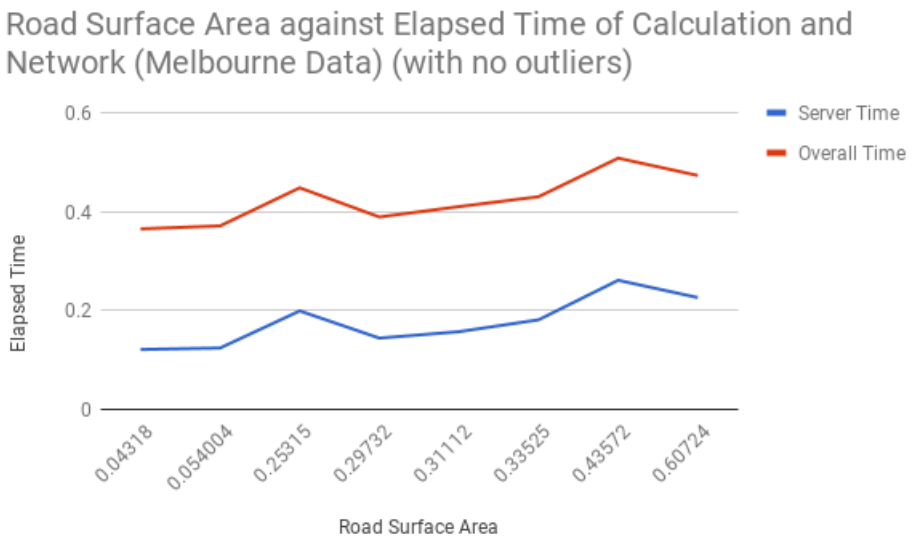


Figure 4.4: Road Surface Area against Elapsed Time of Calculation and Network (Melbourne Coordinates) (No Outliers)

5

Analysis & Discussion

The result of the calculation for the total road surface area is only an estimate. For classifying the road pixel in the map, I've styled the map so that only the roads are visible. However, through closer investigation I've found that the map also shows walking pathway through parks and bridges as roads, and when the level of zoom is high enough (zoom in closer) some building layouts are shown. I've gone through the styling wizard [Google, 2018d] options again to make sure that all buildings are hidden and not shown and I've confirm that all options are correct, but the same building layout are still shown.

However, after having gone through the server side code where it constructs the image URL, I've found that the building layouts are not shown because the zoom level is too low for it to be shown. Walking pathway through parks and bridges are still shown though.

The road surface area result is accurate enough because of the low amount of non-roads objects (walkway) in the map. Provided also that the user of this application should understand that it's an estimated and not an exact value. User can, if they choose to, prepare the exact amount of materials required for surfacing roads. The roads are already classified with a small amount of area being walkways, therefore there should be a small amount of excess material left over. This excess of material should be a small amount, because of the small surface area walkways contributes to the total estimated amount.

6

Future Work

In future iteration of the program I would like to:

- Increase the accuracy of the road surface area result.
- Make changes to the user interface so that it is more suitable for both computer and mobile device.
- Convert the application to a progressive web app (PWA) [Google, 2018j].
- Shows a 1 km^2 square border line around the map marker.
- Shows the result automatically as the marker coordinates changes.
- Keeps record of the result data and visualise it.
- Implement better Development Operations (DevOps).

Increase result accuracy
To increase the result accuracy I might have to find a way to get remove the walkways from the road classifications. To go about doing this I will research the Google Maps API maps object further to see if I could find out what kind of object the walkways are and if I can hide it.

UI being more suitable for computer and mobile devices view
I considering implementing a feature that checks what device the user is on and display an appropriate version of the application to the user so that the view is optimise for the device and screen size.

Convert application to PWA
My application stack is a *MEAN* without the *M* (MongoDB, use as database). I've had a look at web tutorials and videos guide on converting application to PWA. I'm considering creating a branch of the application and experimenting with converting the application to a PWA.

1 km^2 square borderline
I've seen other students in class implemented this feature in their application, and I've asked them about it. I was given a brief explanation on how to implemented it. The problem that I foresee for implementing the Google Maps geometry is that the Google Maps API I'm currently using is a library Angular 5 wrapper of Google Maps [Holstein, 2018]. I'll have to find out if drawing geometry can be done with the wrapper.

Shows result as marker moves
This feature is easy to implement. I can attach a event listener to the dragging event of the marker and call a calculation each time the marker is move. However, the problem with continuously making a calculation as the marker moves is the calculation elapsed time. As shown above in the performance chapter, the overall elapsed time for calculation and network communication on mean average is around under 0.5 seconds. The calculation elapsed time might be a problem, affecting the user experience. Experiments need to be done having users test out the feature and see if its too slow for their liking.

Visualise the result data
Implement a local storage on the client side or MongoDB in the back end server. Storing user calculation results there and visualise it using D3 [Bostock, 2018] or some other data visualisation libraries.

Better DevOps
Implement a tool chain in the application development operation. Implement testing tools such as Travis CI for testing the application and Docker for packaging the application.

Another future work of the project I would like to discuss is the recently introduced Google Maps Platform on 02\05\2018. The new platform combines the various Maps API into products: "We're simplifying our 18 individual APIs into three core products—Maps, Routes and Places, to make it easier for you to find, explore and add new features to your apps and sites. And, these new updates will work with your existing code—no changes required." [Google, 2018g]. Note that the last sentence says the existing code will still work as intended, so there's no change needed for the project.

The new platform also introduce a new pricing plan, where instead of a free tier, the developer get a $200 of free usage each month: "With this new plan, developers will receive the first $200 of monthly usage for free.", "With this new pricing plan you'll pay only for the services you use each month with no annual, up-front commitments, termination fees or usage limits." [Google, 2018g].

However, one should also take into consideration the maximum limits of loads available for the free tier (free $200 monthly). The maximum amount of loads for Dynamic Maps, which is the JavaScript Maps API is 28,000 loads for month [Google, 2018i]. This maximum limitation of loads should be taken into consideration if the application in this project is to be use by large user groups, since a limit might be reach.

Further study of the new Google Maps Platform is needed to gain more insight into the change.

7

Conclusion

In conclusion I thought the project went very well and even exceeded my expectation. I've added a lot more features (requirements) than the specified requirements in the project proposal. The project was completed a head of schedule by 2 weeks. Being so ahead of schedule, perhaps more features from the future work section could be implemented.

Developing the application have been a good experience and a good opportunity to implement what I've learned during my degree course and what I've learned on my own. I've applied what I've learn in project management unit with this project, creating concise documentations and sticking to schedule.

I know that there are a lot more that I can do to the project to improve it. This is a good thing, since one should always strive for betterment and improvement in one's self and in one's work. I will continue to work on the project and make improvement to it.

8

References

Bibliography

[A-Medium-Corporation, 2012] A-Medium-Corporation (2012). Medium. https://medium.com/.

[Academind, 2017] Academind (2017). Google maps & angular angular snippets. https://www.youtube.com/watch?v=lApggVS0icc.

[Axosoft, 2018] Axosoft (2018). Gitkraken. https://www.gitkraken.com/.

[Bostock, 2018] Bostock, M. (2018). D3: Data-driven documents. https://d3js.org/.

[Broadfoot, 2011] Broadfoot, C. (2011). We need a map.getscale() method. https://groups.google.com/forum/#!msg/google-maps-js-api-v3/hDRO4oHVSeM/osOYQYXg2oUJ.

[coursetro, 2018] coursetro (2018). Material design bootstrap 4 and angular 5 tutorial - mdbootstrap. https://coursetro.com/posts/code/132/Material-Design-Bootstrap-4-and-Angular-5-Tutorial---MdBootstrap.

[GitHub, 2018] GitHub (2018). Github. https://github.com/.

[Google, 2018a] Google (2018a). Angular. https://angular.io/.

[Google, 2018b] Google (2018b). Chrome. https://www.google.com.au/chrome/.

[Google, 2018c] Google (2018c). Google chrome devtools. https://developers.google.com/web/tools/chrome-devtools/.

[Google, 2018d] Google (2018d). Google maps apis styling wizard. https://mapstyle.withgoogle.com/.

[Google, 2018e] Google (2018e). Google maps javascript api. https://developers.google.com/maps/documentation/javascript/.

[Google, 2018f] Google (2018f). Google maps static api. https://developers.google.com/maps/documentation/static-maps/.

[Google, 2018g] Google (2018g). Introducing google maps platform. https://mapsplatform.googleblog.com/2018/05/introducing-google-maps-platform.html.

[Google, 2018h] Google (2018h). Material design. https://material.io/.

[Google, 2018i] Google (2018i). Pricing for maps, routes, and places. https://cloud.google.com/maps-platform/pricing/sheet/.

[Google, 2018j] Google (2018j). Progressive web apps. https://developers.google.com/web/progressive-web-apps/.

[Holstein, 2018] Holstein, S. (2018). Angular google mapss (agm). https://angular-maps.com/.

[Joyent, 2018] Joyent (2018). Node.js. https://nodejs.org/en/.

[Lee, 2018a] Lee, T. (2018a). Road surface estimator a project proposal.

[Lee, 2018b] Lee, T. (2018b). Road surface estimator test report.

[Levine, 2015] Levine, M. (2015). Node.js get-pixels: Getting pixels at specific sectors of an image using ndarray. https://medium.com/@mackplevine/node-js-get-pixels-getting-pixels-at-specific-sectors-of-an-image-using-ndarray-

[MDBootstrap, 2018] MDBootstrap (2018). Material design for bootstrap 4 (angular). https://mdbootstrap.com/angular/.

[Microsoft, 2018] Microsoft (2018). Virtual studio code. https://code.visualstudio.com/.

[Mozilla, 2018] Mozilla (2018). Ajax. https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started.

[Paplinski, 2018] Paplinski, A. P. (2018). Fit3036 - 3rd year project: Road surface estimation.

[Rezaee and Zhang, 2017] Rezaee, M. and Zhang, Y. (2017). Road detection using deep neural network in high spatial resolution images. *Joint Urban Remote Sensing Event (JURSE)*.

[Salesforce, 2018] Salesforce (2018). Heroku. https://www.heroku.com/.

[Sass, 2018] Sass (2018). Sass. https://sass-lang.com/.

[scijs, 2018a] scijs (2018a). get-pixels. https://github.com/scijs/get-pixels.

[scijs, 2018b] scijs (2018b). ndarray. https://github.com/scijs/ndarray.

[StrongLoop, 2018] StrongLoop (2018). Express.js. https://expressjs.com/en/4x/api.html.

[Torvalds, 2018] Torvalds, L. (2018). Git. https://git-scm.com/.

[Twitter, 2018] Twitter (2018). Bootstrap. https://getbootstrap.com/.

[VicRoads, 2016] VicRoads (2016). Register of public roads. https://www.vicroads.vic.gov.au/about-vicroads/acts-and-regulations/register-of-public-roads.

[Wikipedia, 2018] Wikipedia (2018). Platform as a service. https://en.wikipedia.org/wiki/Platform_as_a_service.

9

Appendices

9.1 Production and Deployment

Install Node.js from website.

To install the application run ‘npm install‘ at the root directory of the application. This install all node module dependencies required for running the application. All dependencies for the application can be found in the package.json file at the root directory of the application.

Only do the commenting out codes, if the project was taken directly from the Git Hub repository. Otherwise, skip this part.
In file *.\src\app\app-services\area-calculation\area-calculation.services.ts*:
comment out the line :
private serverApi = 'api/area_calculation/';
and uncomment the line :
private serverApi = 'http://localhost:8080/api/area_calculation/';
This configuration is required to redirect application to the correct server API address.

To run the application, it requires the client server and back end server to be run at the same time.

Run ‘node server.js‘ for a backend Express.JS server API.

Run ‘ng serve‘ for a dev server. Navigate to ‘http://localhost:4200/‘. The app will automatically reload if you change any of the source files.

9.2 User Interface

The web application consist of a set of web pages, a navigation bar at the top of the screen (figure (9.1)), and a footer at the bottom of the screen (figure (9.2)).

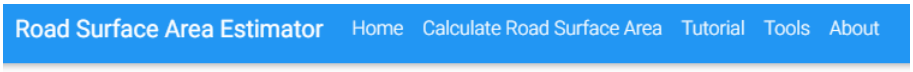


Figure 9.1: The navigation bar at the top of the page.

The navigation and footer can be use to explore the pages of the application.

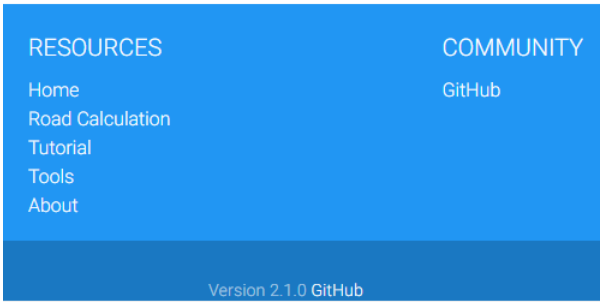


Figure 9.2: The footer at the bottom of the page.

A page in the application is dedicated as a tutorial to the usage of the road surface area calculation function (figure (9.3)). The tutorial include images and descriptions to describe the usage.

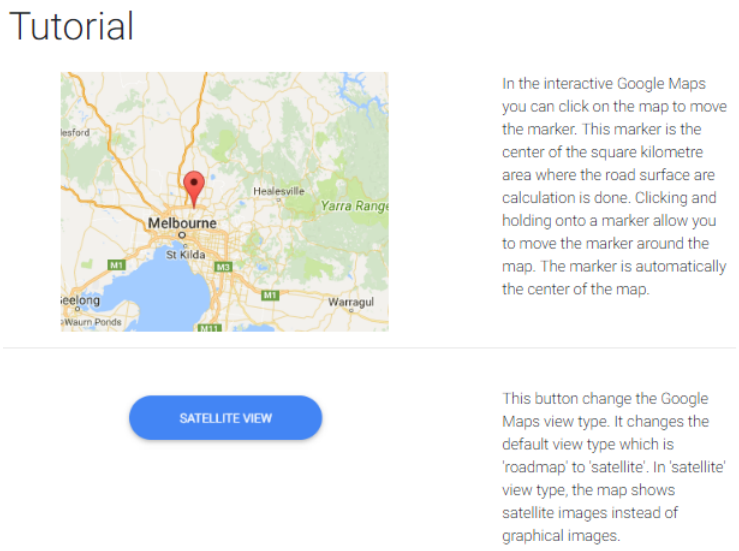


Figure 9.3: The tutorial page.

9.3 Externally Available Functions

There are currently no external functions available for the server. I didn't consider adding any, because I've not implemented any before and because I wasn't thinking about it. Also the application is only of a small scale and won't likely benefit much from any external functions.

9.4 Internal Testing Procedures

I haven't implemented any internal testing procedures for the application. I've tested the application using primitive methods such as manually testing the functions and console logging variables to check the state of the application. The tests are described in the test report [Lee, 2018b].