

Road Surface Estimator

Final Report

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

May 1, 2018

Abstract

Abstract goes here. Less than 300 words.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer massa quam, sodales at sapien a, feugiat pretium arcu. Aliquam augue leo, viverra at scelerisque eu, auctor quis lorem. Sed leo mauris, ultricies sit amet nibh ut, cursus consequat tellus. Integer suscipit interdum nunc vel pretium. Curabitur suscipit, dolor sit amet sodales rutrum, justo leo auctor dui, quis placerat nisl sapien ut purus. Curabitur ut venenatis nulla. Nulla ipsum libero, pretium in enim sed, interdum viverra ante.

Cras quam ex, venenatis id urna in, tincidunt sollicitudin ex. Suspendisse congue justo scelerisque orci varius, et facilisis odio auctor. Curabitur facilisis, orci ac tempor tempus, magna elit eleifend augue, et interdum ipsum sem in risus. Donec ex augue, mollis at enim vel, semper finibus odio. Quisque vel odio vel sapien eleifend condimentum. Pellentesque sed neque viverra, faucibus lorem tristique, mattis sem. Quisque condimentum tortor libero, quis malesuada libero sodales non.

Keywords

Keyword1, Keyword2, Keyword3, Keyword4, Keyword5

Word Count

3000 etc.

Contents

1	Introduction	1
1.1	Objective	1
1.2	Requirements	1
1.2.1	Functional Requirements	1
1.2.2	Non-Functional Requirements	2
1.3	Constraints	2
2	Background	3
2.1	Academic Literature Review	3
2.2	Research	5
2.3	Datasets	6
2.4	Risks	6
2.5	Resource Requirements	7
2.6	Project Tasks	8
3	Method	11
3.1	Internal Design	12
3.2	Software Architecture	13
3.3	Algorithms	13
4	Results	16
4.1	Externally Observable Features	16
4.2	Performance	16
5	Analysis & Discussion	17
6	Future Work	18
7	Conclusion	19
8	References	20

9	Appendices	23
9.1	Production and Deployment	23
9.2	Parameters	23
9.3	User Interface	23
9.4	Externally Available Functions	23
9.5	Internal Testing Procedures	23

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].

This web application that 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 [Papinski, 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 [Papinski, 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, 2018g] 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 Datasets

2.4 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 libraries 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, 2018] prove to be correct.

To mitigate 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.

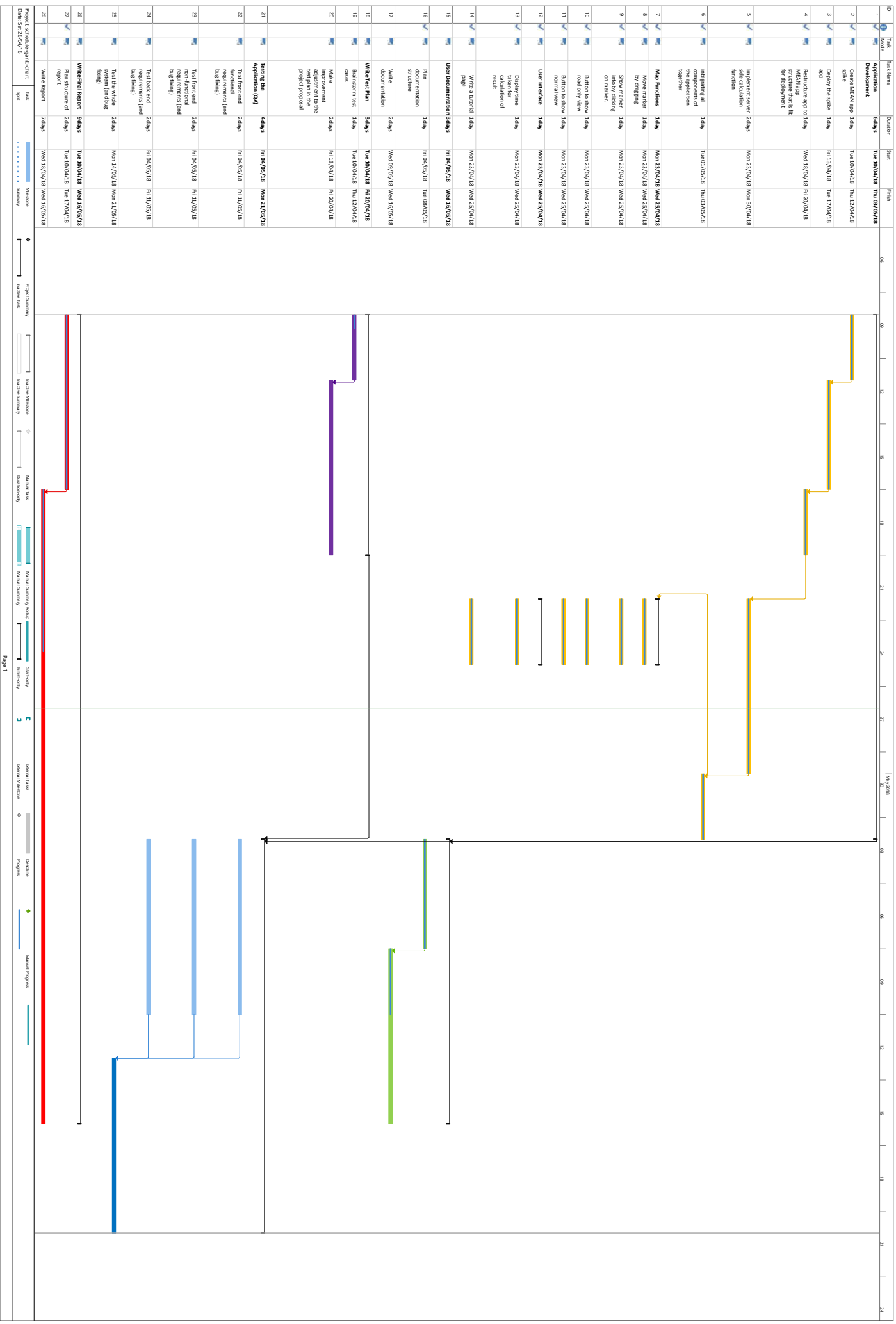
2.5 Resource Requirements

- 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.
- MDBBootstrap 5.2.3[MDBBootstrap, 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.

2.6 Project Tasks

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



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.

3.1 Internal Design

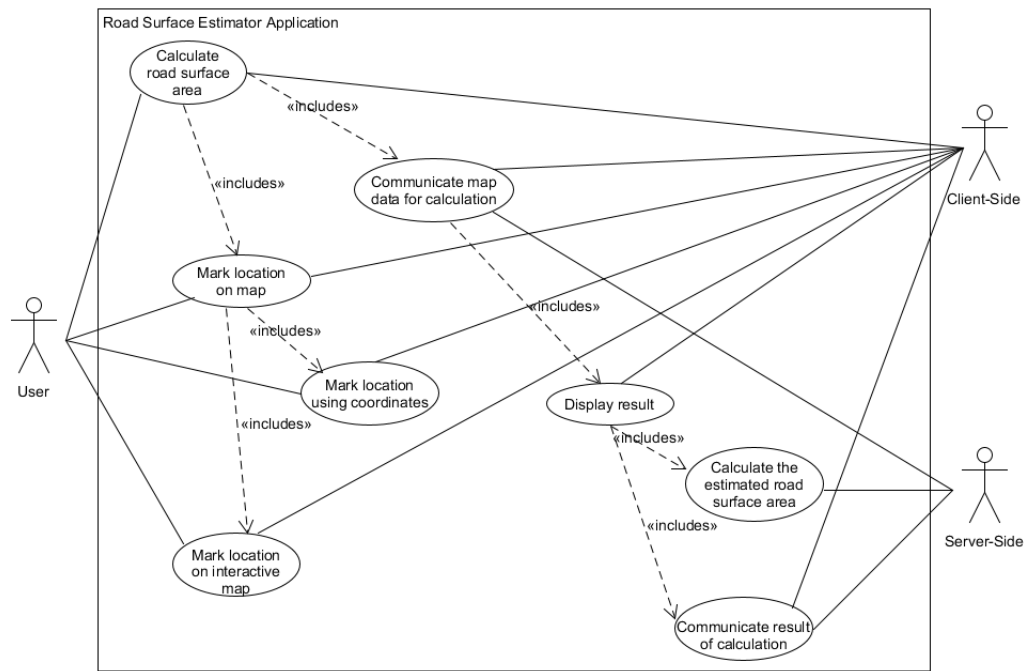


Figure 3.1: Use Case diagram

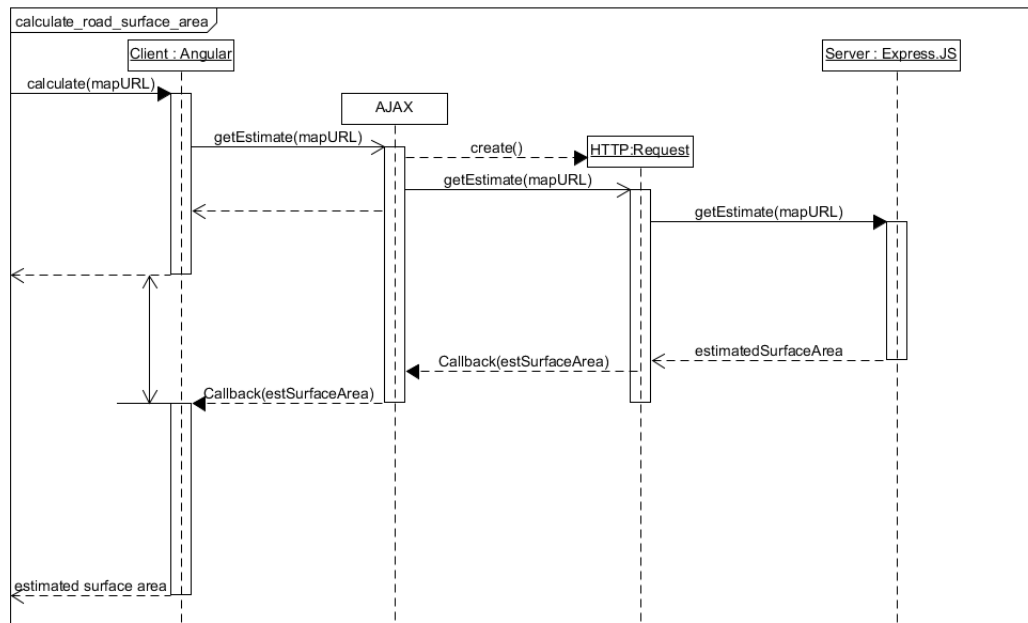


Figure 3.2: Sequence diagram

3.2 Software Architecture

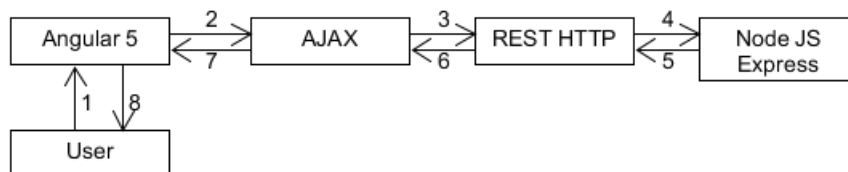


Figure 3.3: Software Architecture diagram

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 checked 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 (3.1)) shows the algorithm steps described above.

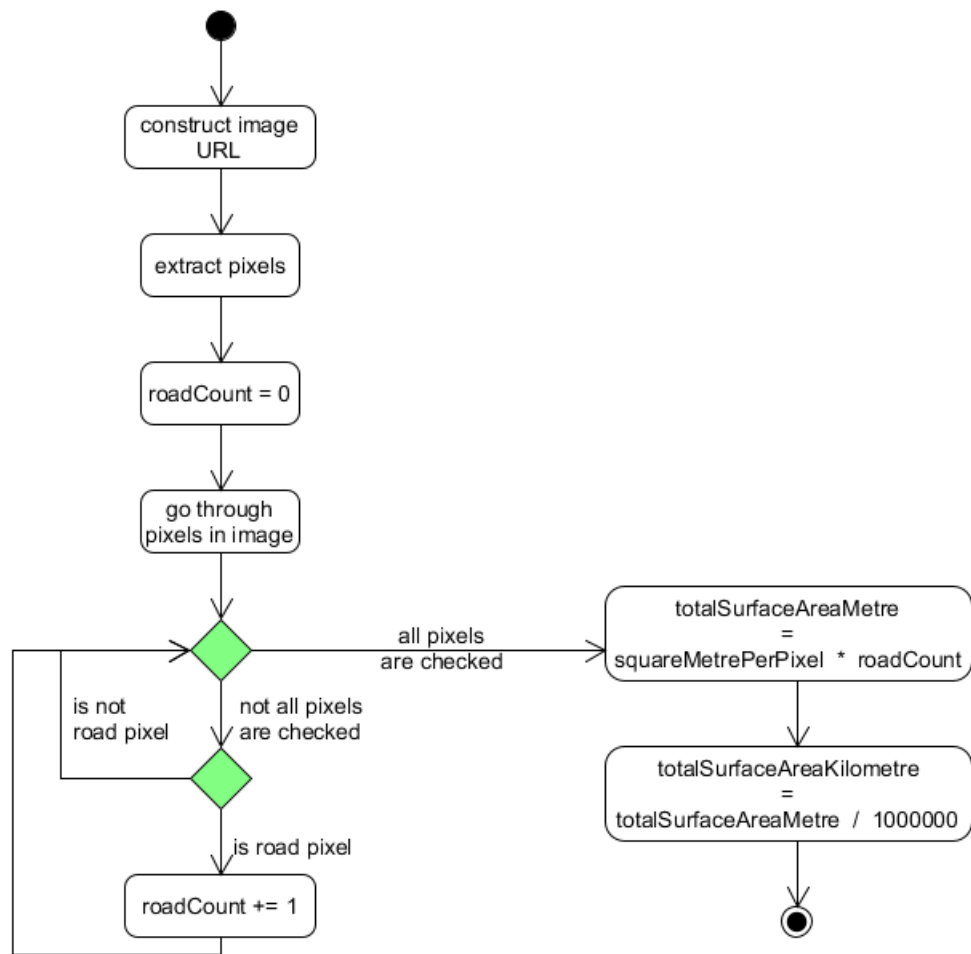


Figure 3.4: The total surface area calculation algorithm.

4

Results

4.1 Externally Observable Features

4.2 Performance

5

Analysis & Discussion

6

Future Work

7

Conclusion

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/>.
- [Broadfoot, 2011] Broadfoot, C. (2011). We need a map.getscale() method. <https://groups.google.com/forum/#!msg/google-maps-js-api-v3/hDR04oHVSeM/os0YQYXg2oUJ>.
- [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). Material design. <https://material.io/>.
- [Holstein, 2018] Holstein, S. (2018). Angular google mapss (agm). <https://angular-maps.com/>.

- [Joyent, 2018] Joyent (2018). Node.js. <https://nodejs.org/en/>.
- [Lee, 2018] Lee, T. (2018). Road surface estimator a project proposal.
- [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://mdbbootstrap.com/angular/>.
- [Microsoft, 2018] Microsoft (2018). Virtual studio code. <https://code.visualstudio.com/>.
- [Papilinski, 2018] Papilinski, 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

9.2 Parameters

9.3 User Interface

9.4 Externally Available Functions

9.5 Internal Testing Procedures