Road Surface Estimator

A Project Proposal

Thanakrit Lee

April 13, 2018

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.

This document propose a 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.

2

Project Requirements

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.

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.

3

Project Plan

3.1 Overview

The objective of the project is to develop a web application that allows user to select a square kilometre area on the interactive map, calculate the total estimated surface area of roads in the area, and display the result.

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]

3.2 Risk Analysis

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.

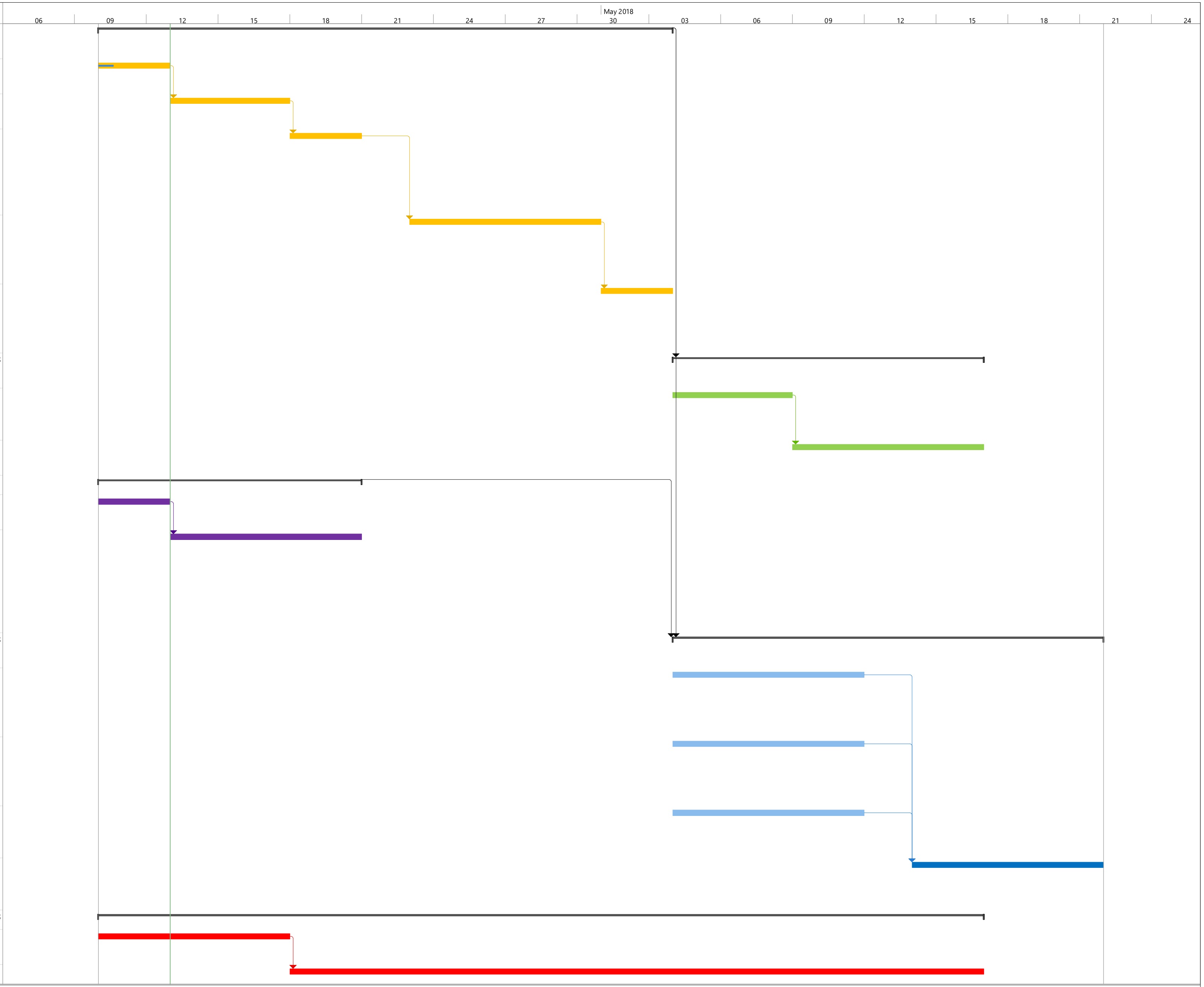
3.3 Resource Requirements

- 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+.

3.4 Schedule

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		**User Documentation**	**3 days**	**Fri 04/05/18**	**Wed 16/05/18**
8		Plan documentation structure	1 day	Fri 04/05/18	Tue 08/05/18
9		Write documentation	2 days	Wed 09/05/18	Wed 16/05/18
10		**Write Test Plan**	**3 days**	**Tue 10/04/18**	**Fri 20/04/18**
11		Brainstorm test cases	1 day	Tue 10/04/18	Thu 12/04/18
12		Make improvement adjustment to the test plan in the project proposal	2 days	Fri 13/04/18	Fri 20/04/18
13		**Testing the Application (QA)**	**4 days**	**Fri 04/05/18**	**Mon 21/05/18**
14		Test front end functional requirements (and bug fixing)	2 days	Fri 04/05/18	Fri 11/05/18
15		Test front end non-functional requirements (and bug fixing)	2 days	Fri 04/05/18	Fri 11/05/18
16		Test back end requirements (and bug fixing)	2 days	Fri 04/05/18	Fri 11/05/18
17		Test the whole system (and bug fixing)	2 days	Mon 14/05/18	Mon 21/05/18
18		**Write Final Report**	**9 days**	**Tue 10/04/18**	**Wed 16/05/18**
19		Plan structure of report	2 days	Tue 10/04/18	Tue 17/04/18
20		Write Report	7 days	Wed 18/04/18	Wed 16/05/18



Project: schedule-gantt-chart.m
Date: Fri 13/04/18

Task	▬	Summary	▬	Inactive Milestone	◇	Duration-only	▬	Start-only	[	External Milestone	◆	Manual Progress	▬
Split	······	Project Summary	▬	Inactive Summary	▬	Manual Summary Rollup	▬	Finish-only	]	Deadline	⬇		
Milestone	◆	Inactive Task	▭	Manual Task	▬	Manual Summary	▬	External Tasks	▬	Progress	▬		

4

External Design

4.1 User Interface

The user interface of the web application will be responsive. It can be view on both the computer web browser and the mobile phone web browser.

Figure (4.1, 4.2) shows the web application user interface on the computer web browser. The "Calculate Road Surface Area" page is for using the main functionality of the application (getting the surface area). It includes a Google Maps for user to interact with, the user can zoom, move around, add centred marker (the map is centred to the marker when it's added) on the map, move the marker around by clicking on it and dragging it, and click on a marker to get more information about the marker (coordinates).

Figure (4.3, 4.4, 4.5, 4.6) shows the web application user interface on the mobile phone web browser. Depending on how small the size of the screen of the mobile device is the navigation bar at the top of the page will compress and becomes a navigation bar that drop downs the navigation links. The suer can drop down the navigation links by tapping on the "burger" icon on the right side of the navigation bar.

User can choose to add marker by tapping on the interactive Google Maps or by entering the latitude and longitude coordinates in the input field. The input fields is reveal and hidden with the use of the "USE COORDINATES INSTEAD" button.

This user interface mockup is not how the finalise version of the application will look like. Changes to colour scheme and layout of components are likely to change to increase the application usability, e.g. user experience, making the application easier to use, pleasant to look at and use etc.

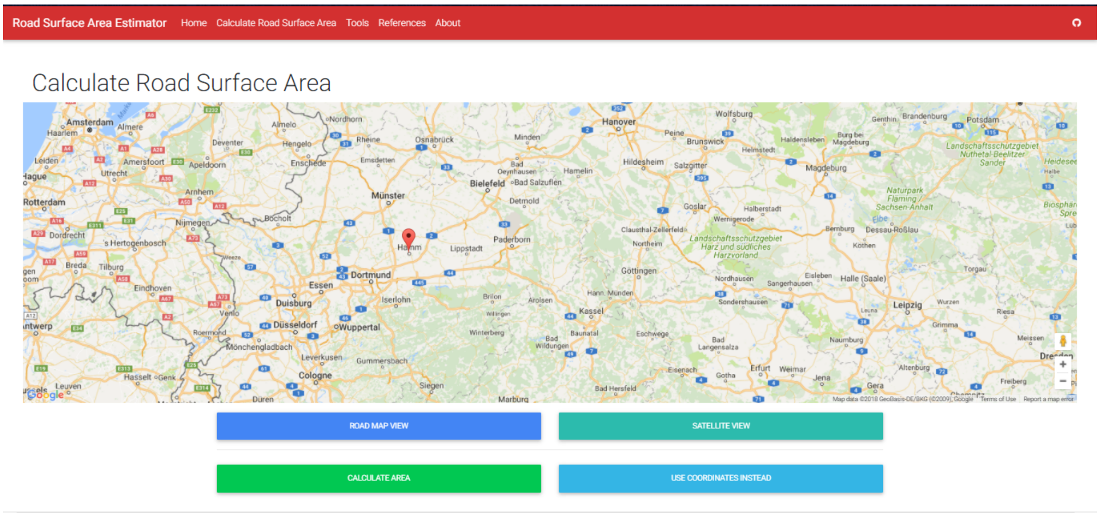


Figure 4.1: Computer web browser view 1

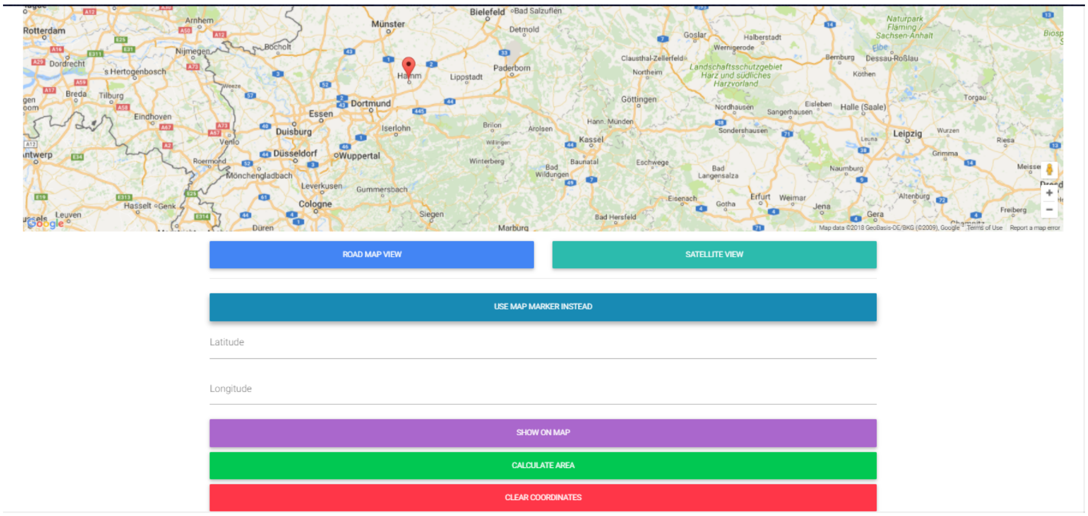


Figure 4.2: Computer web browser view 2

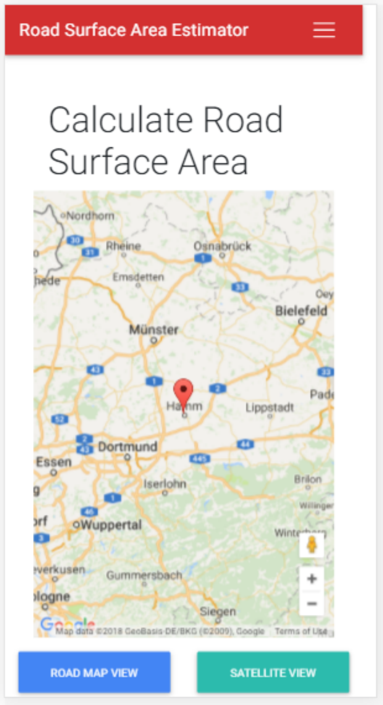


Figure 4.3: Mobile phone view 1

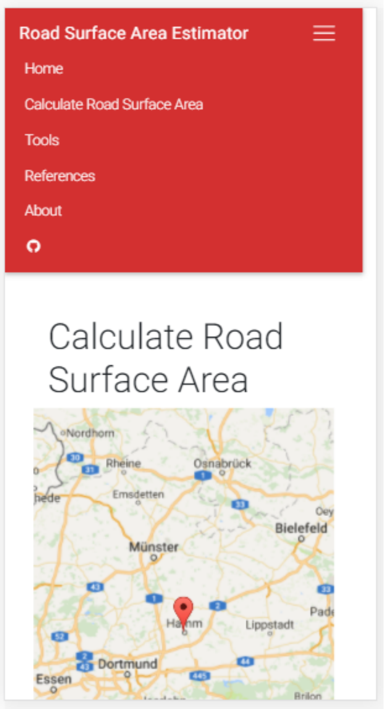

Road Surface Area Estimator
Home
Calculate Road Surface Area
Tools
References
About
Calculate Road Surface Area

Figure 4.4: Mobile phone view 2

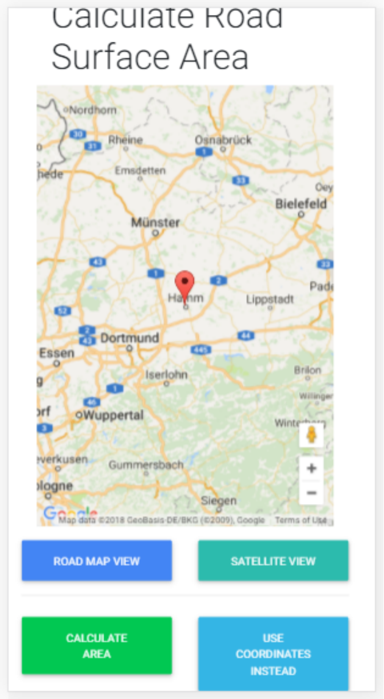


Figure 4.5: Mobile phone view 3

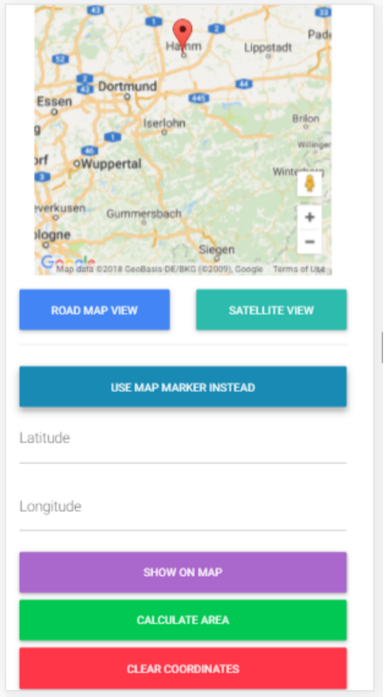


Figure 4.6: Mobile phone view 4

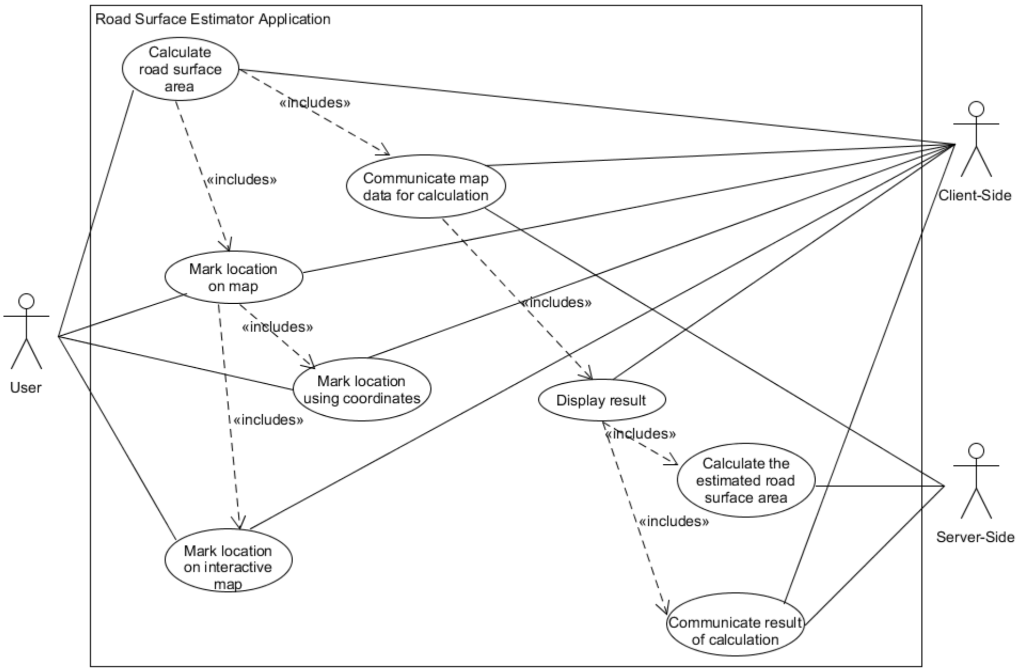
4.2 Performance

Time complexity of the server processing the image and making the surface area calculation is $O(n)$, where n is the number of pixels in the map image. It is of that time complexity because the server will run through all pixels in the image array once and collects "road pixels", then it will add all the surface area of the collected "road-pixels" together to get the estimated total road surface area.

5

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 calculation and processing of data.



Road Surface Estimator Application
Calculate road surface area
«includes»
«includes»
Communicate map data for calculation
Mark location on map
«includes»
«includes»
Mark location using coordinates
«includes»
Display result
«includes»
Calculate the estimated road surface area
«includes»
Mark location on interactive map
Communicate result of calculation
User
Client-Side
Server-Side

Figure 5.1: Use Case diagram

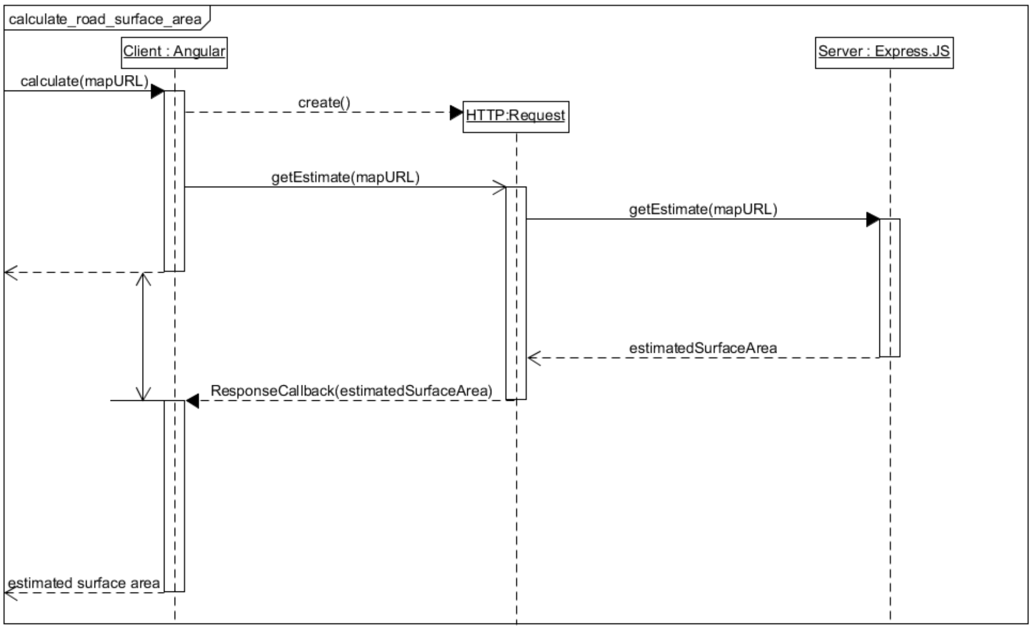


Figure 5.2: Sequence diagram

6

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 the process the image and calculate the total surface area of roads in the map, and response the result back to the client.

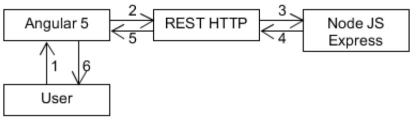


Figure 6.1: Software Architecture diagram

7

Test Plan

7.1 Test Coverage

Test Coverage List		
ID	**Requirement**	**Product Life Stage To Verify At**
Re0	Display graphical map	Development
Re1	Have user-to-map interaction	Development
Re2	Have coordinates user input	Development
Re3	Turn user input coordinates into marker on the map	Development
Re4	Calculate estimated total surface area of roads in user selected km^2 on the map	Development
Re5	Display surface area result	Development
Re6	Doesn't take "long" to do the surface area calculation	Testing
Re7	Web design is responsive	Testing
Re8	Can be use on mobile device	Testing
Re9	Provides a easy to understand tutorial	Testing
Re10	Can access the application on the internet	Testing

7.2 Test Methods

Test Methods List	
ID	**Testing Implementation Method**
Re0	Unit Testing
Re1	Integration Testing
Re2	Unit Testing
Re3	Integration Testing
Re4	Integration Testing
Re5	Unit Testing
Re6	Performance Testing
Re7	Usability Testing
Re8	System Testing
Re9	Usability Testing
Re10	System Testing

7.3 Sample Test Cases

Test Sample Cases List	
ID	**How the data will be collected**
Re0	By checking that the web application displays a graphical map (Google Maps).
Re1	Interact with the map on the web application page. Interaction includes scrolling to zoom in and out, dragging to move around the map, clicking on the map to add a marker and centre the map on the added marker, change map type and style, dragging a marker, clicking on a marker to show more info.
Re2	Check that the app page has input options for longitude and latitude co-ordinates.
Re3	Click on the "show on map" button to test the functionality. Check that a new marker is added on the map. Check that the coordinatse of the new marker is the same as the user input coordinates by clicking on the marker to show the marker info which includes its coordinates.
Re4	Check that the map area to calculate is 1 km^2. Check that the result is a correct estimate.
Re5	Check the app page to see that a result is shown there.
Re6	Check that the app doesn't take more than 5 seconds to calculate and return the result, by timing the calculate function.
Re7	Change the size of the web browser on the laptop to see if web app is responsive. Change the size of the mobile device in chrome developer tools[Google, 2018b] to see if web app is also responsive to different phone screen size.
Re8	Use chrome developer tools and change to mobile view and check that the web app works on the mobile devices. Using a real physical phone, on the phone web browser go to the web app address and check that the app works on the phone.
Re9	Get a user to use the application and ask for inputs, comments, and opinions afterwards.
Re10	On both a computer and mobile phone, use the web browser and go to the address of the web application and test that the application works by checking all functionalities.

8

References

Bibliography

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

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

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

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

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

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

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

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