Real Time Route Optimization

FINAL REPORT

Team Number: 18

Client: Adam Ryan   Henderson Products, Inc.

Adviser: Goce Trajcevski

Team Members/Roles:

Junjie Wen - Backend Developer; Data Analyst

Zhanghao Wen - Product Manager; Scribe

Yuhang Xie - Web-App Developer; UI Leader

Xinhe Yang - Web-App Developer; UI Designer

Tianhao Zhao - Communication Leader

Team Email: sdmay19-18@iastate.edu

Team Website: http://sdmay19-18.sd.ece.iastate.edu/

Revised: 04/29/2019 Version 3

Table of Contents

List of figures

List of tables

1. Introduction

The purpose of this project is to develop a system for effective management of a fleet of snow-plowing trucks.

1.1. Acknowledgement

Our client, Mr. Adam Ryan from Henderson Products Inc., provides us with the overall guidance about the different facets of the problem. He guides us through the process of representing (i.e., pre-treating) the data on the map with trucks in snow operations, list scenarios of different situations (e.g weather conditions) to help us achieve the design of snowplows route arrangement. In addition, he will provide us with the real history data with a set of values from the different sensing devices on the trucks for the purpose of enabling a real data simulation for our solutions. For all of this, we are very grateful.

We thank to our adviser, Professor Goce Trajcevski, who provides us with valuable advices on our decision and planning throughout the year. He helps us making decision on how we can approach the tasks in multiple aspects, how we can go through our design process and find a way to present our thoughts, and how we could overcome current challenges and come up with appropriate solutions to the problems.

1.2. Problem Statement

After snowfall, it is important to clear both road segments as well as driveways, so that: (a) travelers' safety is improved, and (b) safe access is enabled for employees, customers and suppliers. Car-slip accidents are quite common during winter seasons, especially in North America. Thus, an effective management of snow cleaning/removal is of the utmost societal importance – however, snow cleaning operations could incur high cost, both in terms of human labor and financial resources. Our client, Henderson Products Inc., is a custom manufacturer who ensures that their customers (including state departments of transportation, cities, counties, and other governmental agencies) have the most complete snow and ice fighting solutions and service available ("HENDERSON PRODUCTS - ABOUT US"). Presently, Henderson Products Inc. is seeking a solution which takes advantage of modern Internet of Things and Big Data paradigms and technologies to improve the operational performance of snow-plowing fleets of trucks in and reduce their service and re-loading time as well as financial costs. At the current state of the affairs, the snow operation dispatchers or the target users of our project, may use some specific ways to collect data in order to get the status of the trucks location, dispenser info, and road conditions during snow operations - however, the data collection, analysis and decision-making are done in a rather ad-hoc manner. Hence, our team is aiming at providing a systematic approach and generate a solution that will allow them to have a better experience of gathering real-time information of trucks and controlling the real-time situation of arrangement of fleets during snow operations. Our solution is intended to enable dispatchers to perform a more effective decision-making in terms of tasks re-assignments and trucks re-routing (both in terms of road segments to be cleaned, as well as reloading with supplies and servicing).

1.3. Operating Environment

Our final product is intended to be a web application and it will be used by the dispatchers as well as managers in certain branches/office of Department of Transportation (DoT). Since it is a web-based design, the hardware such as equipments of sensors are not connected to our design, the actual physical environment will not be directly relevant to our envisioned product from development perspective, except a snow weather condition for utilizing our product. It will require the use of specific local and online system with accessible data that fits and operates the software solution from our design.

1.4. Intended Users and Intended Uses

Our project Real-Time Route Optimization aims at optimizing the routing of the snow plow trucks and reduce the resources wasted during operations. There are three types of users, each with different access interface: (1) general public; (2) dispatchers for arranging trucks; and (3) operator/manager/admin at DOT office, who will have a full access to different information, along with the privilege to come up executive re-routing schemes.

For the public view, citizen can access our website to find out the nearby snow plowing trucks which displayed on the map, as well as the status of which road segments have been cleaned.

The dispatcher's task is to assign the route for directions of each snowplow, and maybe able to show status of truck supplies (e.g. melting salt, dissolved liquid, gasoline). In this process, the snowplow may need to pass the same path and redo snow operations due to bad road conditions. Our project is to help dispatchers properly distribute the tasks of snowplow and reduce waste of resources.

Admin level users, such as office manager, can see the situations during re-routing process, monitor the tasks completion by dispatchers, or help solve unexpected conditions. They may get records of data, do some high-level analysis on specific areas of road for future preparation.

1.5. Assumptions and Limitations

Assumption：

- Users will have internet access

- Users will have a computer system that deal with large amount of data and be able to hold a large number of caches for storing trucks info on the page during a certain time period with real time updating of info.
- The maximum number of simultaneous users will not be limited.
- The users will be able to read instructions on our design of representing the map with trucks and road information.

Limitations:

- Because we are not able to use real time data collected on sensors for developing our design, we need to build up a simulator that conclude all data types of requirements, for future connection with real data purpose.
- Some features of our design have been limited, such that we are not confirmed that we are able to access camera info of the street, or some administration permit from the governor.
- The amount of running cache depends on users' computer system info. Better computer system will have better experience of surfing our map page.

1.6. Previous Work and Relative Literature

Many local governments in north maintain a fleet of snowplows. In the winter season, speed and efficiency of snowplow fleets are of great importance – especially in cosmopolitan cities. Many companies are seeking a solution, which takes advantage of modern Internet of Things and Big Data in order to improve performance of fleets. GIS Cloud has shown how to make a snow fleet management system more efficient with a simple software solution in City of Zagreb. They integrated Map Editor with Fleet and Workforce Management, and they also tracked vehicle in real time and record every vehicle with the path of driving, times, miles, fuel consumption, speed limitation, and status of the vehicle. This solution is similar to what our team are planning to implement.

They also use sensor to capture real-time data. However, the difference is that their sensor is for detecting physical environment like temperature of pavement and air, humidity of the pavements and the freezing point which depends on rainfall, fog and salinity of pavement. Our sensor is focusing more on truck's data like material type, spreader gate height, conveyor speed, spinner speed.

In our project, we are going to visualize information of fleets and environment variables to inform dispatcher how snow event is going.

2. Requirement Specifications

2.1 Functional Requirements

- A web application, which contains:
  - Trucks' information along with movement
    - GPS locations
    - Truck's speed
    - Rate of liquid dispenser
    - Rate of conveyor
  - Road conditions
    - color specified conditions determine whether road was already cleaned.
  - Weather conditions
    - temperature
    - pressure
  - Temporary data of each truck along the path
  - Login page with multiple layers of users
  - History page to store past data
- AWS server implemented by Nodejs, which contains:
  - Simulator to generate truck data for display, since we cannot get live data
  - Authentication for high level managers to login
- MySQL database, which contains:
  - Table for truck GPS and support condition data
  - Table for user information

2.2 Non-Functional requirements

- Contents in App and Web interface design should be easy to read and understand.
- Support showing multiple vehicles at same time.
- Store enough amount of history data.
- User's request can be responded in an acceptable time.

3 Design Plan

3.1 Proposed Design

For implement the algorithm of re-routing, the following information should be gathered and properly incorporated: traffic condition, weather condition, rate of water/salt dispenser, and GPS information of individual truck during snow operations. Therefore, the following tools, API, and framework are used for the purpose of front-end interfaces, backend database and server.

3.2 User Cases

- Login page with multiple layers of users
- Trucks' information along with movement
- Road conditions color specified conditions determine whether road was already cleaned
- Temporary data of each truck along the path which depends on web caches capacity
- History page to store past data
- Rerouting Suggestions from one location to another location.

3.3 Modules

Now, we are considering the main parts of our projects.

3.3.1 Web application

We choose to use AngularJS as our web design interface framework. It has a good html block handling. AngularJS is a Google product, so it has good compatibility with Google map API. In comparison to ReactJS which lacks official documentation, it has a lot of information to help beginner to learn and easy to use with simple html knowledge. It accepts other frameworks with similarity in architecture design. It also has tiny size which allows faster and better performance comparing to ReactJS. It also has a great integration with small interactive parts. This is a lot better than ReactJS which requires deep knowledge of integration. We can simply apply the single application to complicate web interface applications by using AngularJS.

3.3.2 Database

Development: MongoDB is friendly to developing engineer, because of its supply of JSON format data. With the Mongodb, developing speed can get improved. For MySQL, it is a more mature solution. There are lots of documents about MySQL. Besides MySQL is also extendable for different data type in the future. So, for development, MySQL and MongoDB does have significant advantages than the other one.

Maintenance: relational database is very good for maintain because the rules in operation on table. On the other hand, no-relational database is not easy to maintain, and more risk in illegal operation on database. So, for maintain, MySQL is also friendlier to engineer.

Load: Compared with MySQL, MongoDB is better on loading. In our case, a large amount data will be dealt with, plus we are more familiar with MongoDB, so MongoDB will be our best option here.

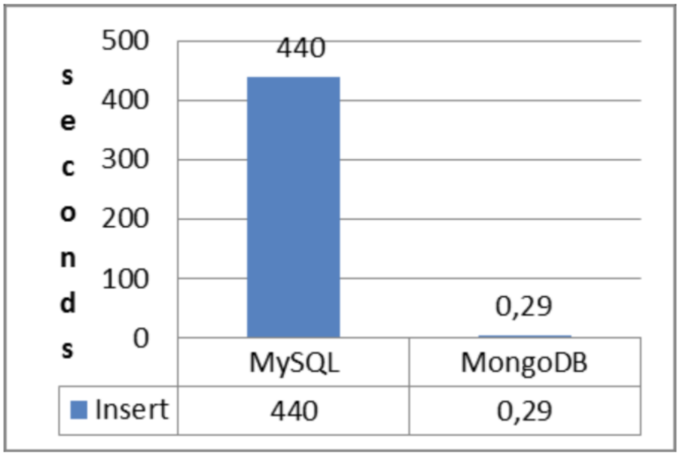

Figure 1: MySQL vs MongoDB

3.3.3 MAP API

We are considering several map APIs including Leaflet, Bing Map, and Google Maps.

Leaflet is a widely used open source JavaScript library used to build web mapping applications.

Bing Maps is a web mapping service provided as a part of Microsoft's Bing suite of search engines and powered by the Bing Maps for Enterprise framework.

Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets, real-time traffic conditions, and route planning for traveling by foot, car, bicycle, or public transportation.

3.3.4 Block Diagram

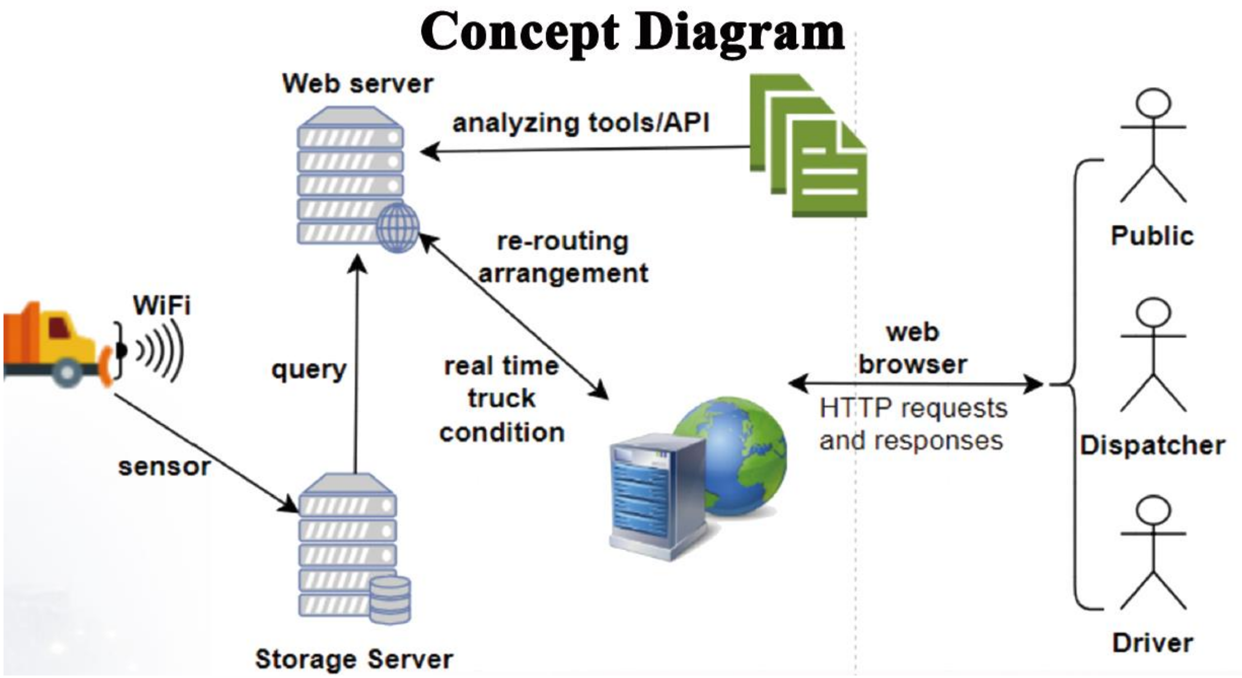

Figure 2: Concept Diagram

3.3.4.1 Web View

Data processing layer takes the simulated history live data from the server temporarily store the data in cache and then logically transfers the data to display page. In our case, the truck's simulated history live data is stored in the cache as an array, and the data processing layer displays the data on Google Maps. The login page requires user to enter username and password to login. The data is compared with the database data through the data processing layer to determine whether the data is the same. The home page mainly displays information on the google map, and the icon shows the specific location and information of the truck. The history page requires the user to enter the time range. The data processing layer will get data of database, display truck's driving route and information on Google map, and display the data analysis of truck during this time, such as average speed and total mileage.

3.3.4.2 Server

1. Router layer used to assign different pages and display different content data. In our case, we can access login page, home page and history page through the router layer.

2. Service layer used to simulate truck data such as GPS position, speed, bearing and spinner speed.

3. Logic layer will get data for logical processing, it stores the truck's simulation data in the database or calls the data needed by front end from the database. It will determine if the password entered by the user is consistent with the database. If it is inconsistent, it will return error message to the front end. It can analyze the data obtained by the database and calculate the average speed or total mileage of the truck.

4. Data access layer temporarily store data in cache, waiting for other components to call the data

3.3.4.3 Data Storage

Classify data and store it in the database or take data from database. The user's username and password are stored in the UserTable, and the user's level is divided into normal users and administrators. The truck information is stored in the TruckTable and contains fields such as Altitude, GPS_head, Acc_mag, Conveyor, Spinner, Prewet and Time.

3.3.5 Architecture Diagram

We use Google Route API to simulate history data in order to display rerouting features. We uses nearby locations according to the history data in order to calculate a feasible route to for rerouting suggestions. With Google Map API, we can see current status of the truck and rerouting path on the map. We also use weather API for displaying specific area weather conditions.

Here is our architectural design diagram.

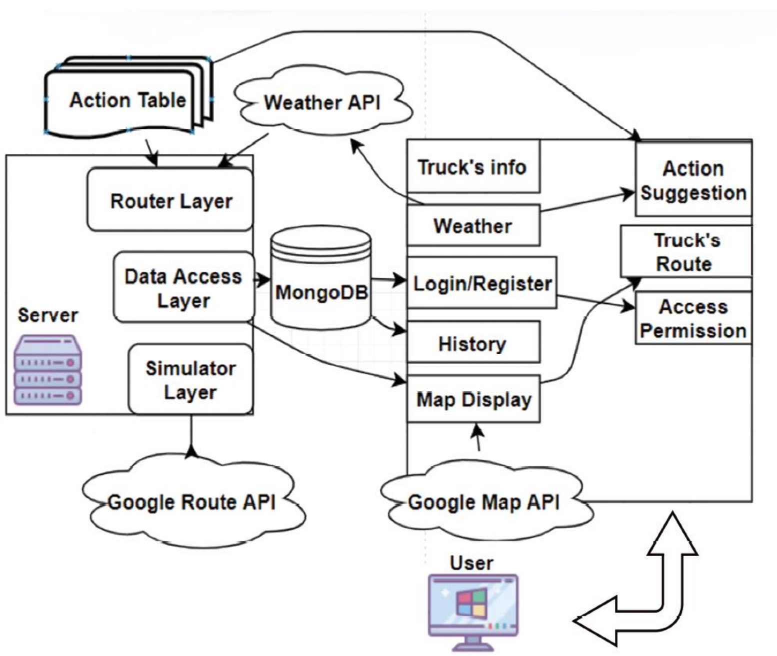

Figure 3: Architectural Diagram

3.4 Module Constraints

We are using AWS server and Google API. It may incur costs during high frequency interaction.

4 Implementation

4.1 Development Process

We developed our design in an agile mode. We started with project proposal and clarification from our Client and adviser, then we got an overall view of our project. We made a progress, then we learned by clarifying more specific goals. We refined the directions of our design and shift emphasis on different fields. Then we made progresses and refined goals for further steps.

4.2 Implementation Issues

4.2.1 Frontend Implementation Issues

- Login page

  The issue is that the model we used at first can only handle a single page. However, client proposes that the program needs to be able to provide different functions based on users' privilege which means we need a login page.
  The solution is to replace the module we originally used with the router module. Router module can place each page in different layers, so that the program can contain multiple pages with corresponding contents.

- Main page
  1. Our goal is to make the truck move along the route smoothly. However, because of the problem of page refreshment, trucks jump to the other coordinate for each movement. In order to solve this problem, we limited the range of data generator to let trucks move in small range. We divide the single movement into several small segments on its distance, in order to make it seems to be real for simulation.
  2. After completing the function of displaying vehicle information on the map, we encountered a new problem - The info window of the truck shows only during the time before its update. After each update (few seconds), it disappears. However， our expect result is to click the truck once and shows it for a long period of time. Therefore, we add a variable to the initial frame of updating and keep its property of last time of whether the info window is open or not.

3. Our expect result is that program can run for a long period without stuck. However, the movement of trucks shows on the page sometimes get stuck because that it produces a lot of virtual memory. To solving this problem, we had tried a lot of algorithms and compared their effectiveness. Finally, we chose First In First Out algorithm to keep the group of data in the same size. This method can ensure that the memory will not overflow and cause slow operation.

4.2.2 Backend Implementation Issues

When we tried to display a truck to show multiple trucks, we encountered a overlapping problem. Multiple trucks show at one location instead of multiple locations. So we redesign the data generator in the simulator. Whenever the generator generates new data, it will include as many sets of data as the number of trucks. This will ensure that each car moves to a different location and takes a different route.

4.3 Standards

The team will use the following standards during development of the project:

- Version Control System
  - Git will be used as the primary means of version control for all project code.
  - The Google software suite (Docs, Sheets, etc.) will be used for all formatted documentation, such as planning and design documents. It has version control features that can accessed from the menu (File → Version history).
- Code Review
  - Development will be done in feature branches.
  - When a feature branch is ready to be merged to the master branch, the author will assign one or more of the other project members as a reviewer.
  - Reviewers must check to ensure the code:
    - correctly implements the desired functionality
    - contains enough tests to ensure correctness
    - passes tests
    - is free of errors
    - integrates successfully with the existing software
  - Reviewers will communicate code issues to the author, who is then responsible for addressing all issues.

○ Once the code has passed inspection, it can be merged to master.

4.4 Risk Management

After analyzing the entire software architecture, there is a framework risk theme for our project. First of all, probably there is a potential risk for security in future. Our sensor data is stored in the database. In authentication and authorization, data is transferred between the client and the server. However, the data is not securely encrypted. If data is intercepted during transmission, this security risk will result in the user's information being easily leaked and used. The amount of traffic to our data could be very large, which can affect server performance and interface access speed. And the communication between the servers is very expensive. In accuracy, the sensor on the truck may be affected by factors such as the weather, so the accuracy of the data we return will be reduced. Therefore, we should carefully analyze these existential analyses and get improved methods.

5 Testing, Validation and Verification

5.1 Interface Specifications

We design a web application by using AngularJS framework. The interface mainly includes a map and a top menu. The map will dynamically display the motion of all trucks within a certain area. Users can access the truck information window by clicking on the record point along truck's way. The records points are generated for each movement of the truck, it includes the truck speed and supply information. Top menu includes the history page with a date selection box, reroute action, home link and other actions.

Here is a screenshot of our homepage as shown below.

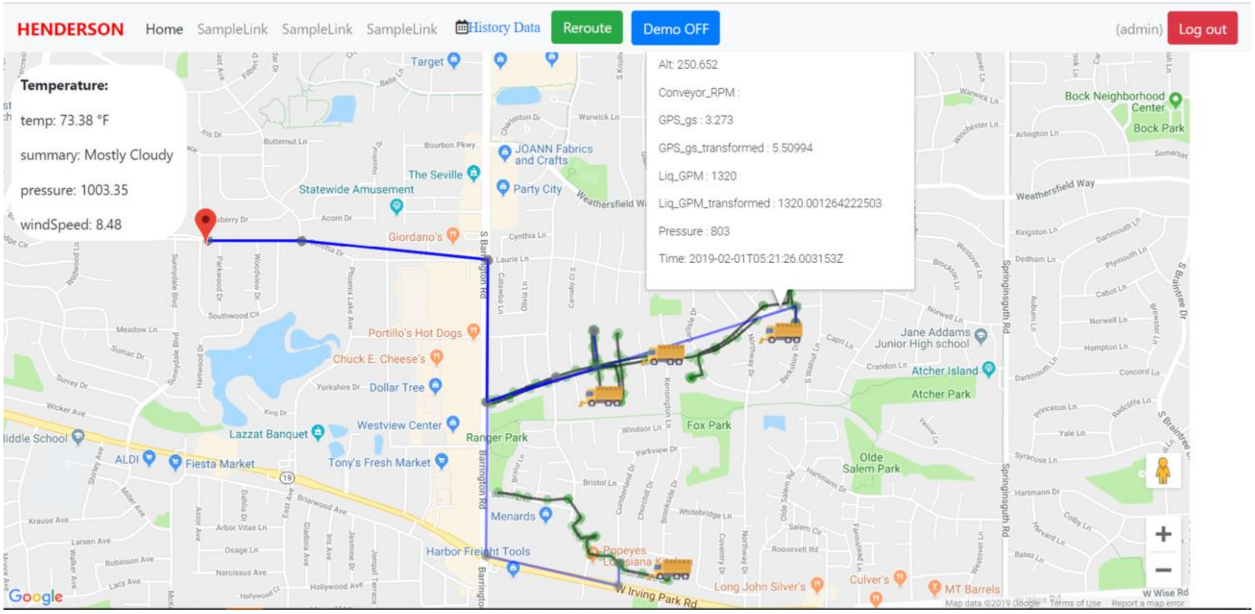

Figure 4: Homepage screenshot

5.2 TEST PLAN

We do tests from multiple aspects including unit test, integration test, system test and acceptable test. By those tests, we can ensure the robustness and effective, and make this project can be used stably. Besides, those functional tests can ensure the project fit the requirement and help the acceptable test. On the other hand, those non-functional test will improve the robustness of this project, and make it more stable to offer service to more people

5.2.1 Unit Test

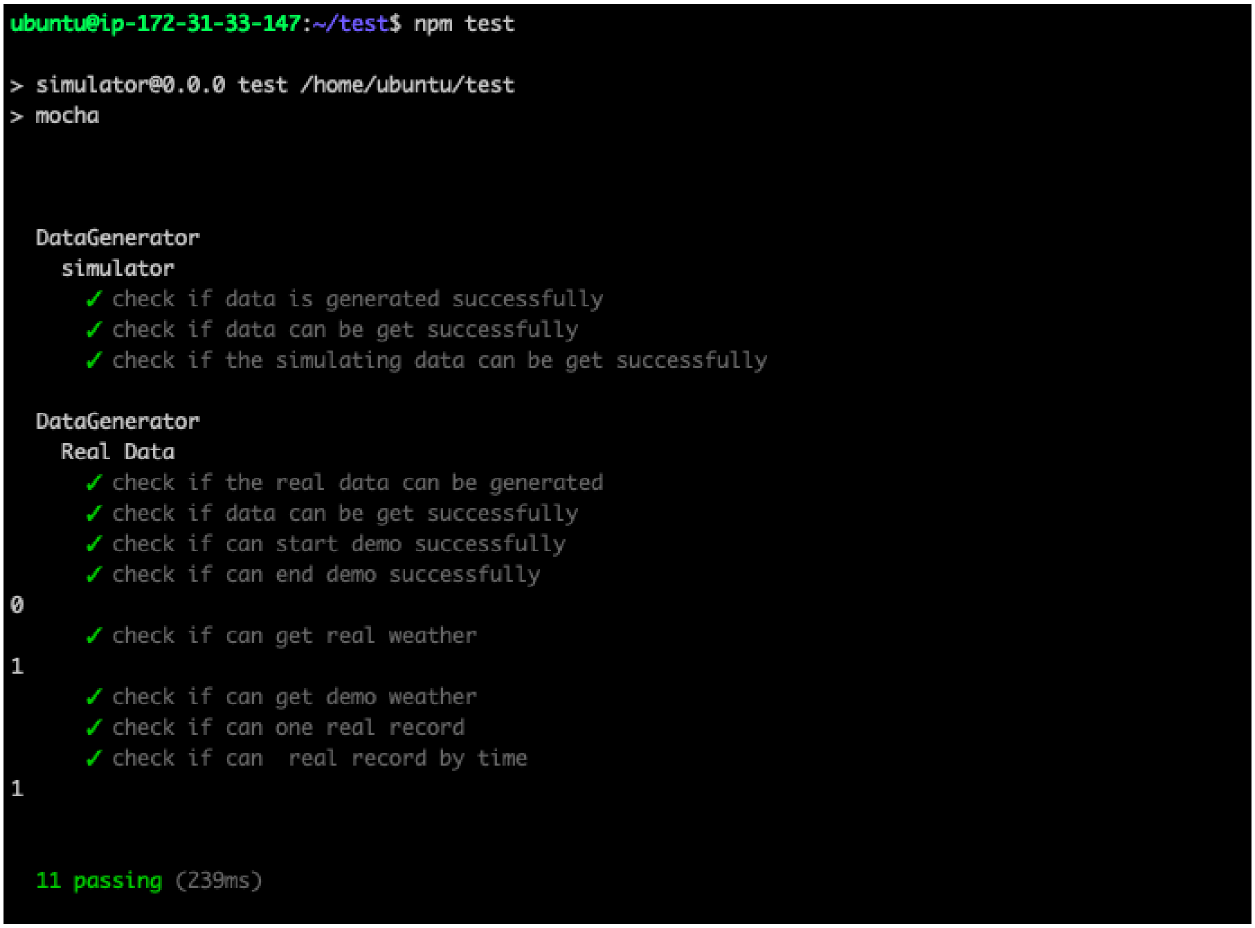

Figure 5: Unit Test screenshot

5.2.2 Integrity Test

We will do functional testing based on the functional requirement, by using the simulating data generator.

Tested requirement:

Display trucks' GPS location on map as well as traffic condition

Expected result:

simulating data generator will send GPS location of trucks, and user should be able to see how truck move on the web.

Tested requirement:

Check out individual truck's supply condition (rate of water/salt dispenser)

Expected result:

simulating data generator will send truck's supply conditions of trucks, and user should be able to check all relative information on the web by clicking the truck.

Tested requirement:

Allow to open weather condition on certain snow operation area.

Expected result:

User will click the weather button and see the snow on certain area.

5.3 Software/framework testing

We considered using spring or Node.JS to build our backend development framework. Finally we chose NodeJS Here is an example of "Hello World" program by using Node.JS and Spring:

Obviously, Node.JS can reduce tons of code compared to Spring.
On the software side, we will use AngularJS to develop the web to provide a line system for trucks. The simulated sensor data obtained is analyzed and algorithm calculated. Display the real-time location of the truck and the data in the car on our web.

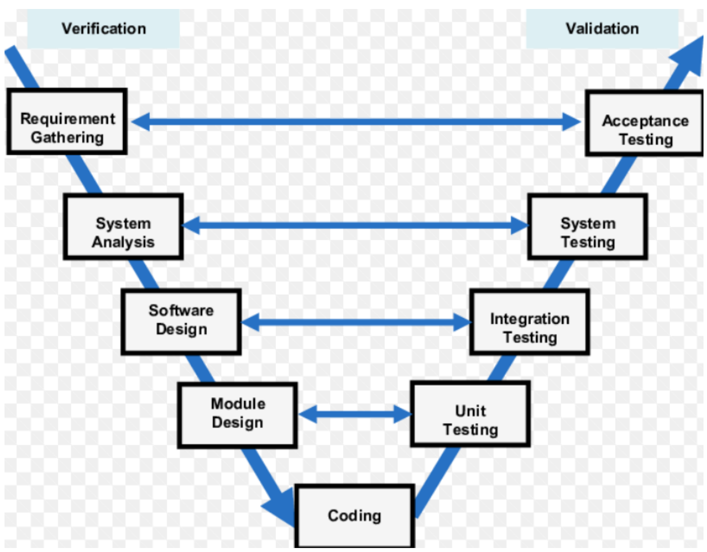

Figure 6: Layers of Testing ("Filling the Gap between Conception and Unit Test.")

5.3.1 Non-Functional Testing

For non-functional test, we test our project based on stress testing and performance test.

Stress testing:

We send tons of request to our server by JMeter, and test the limitation of backend server to see if it can handle enough load.

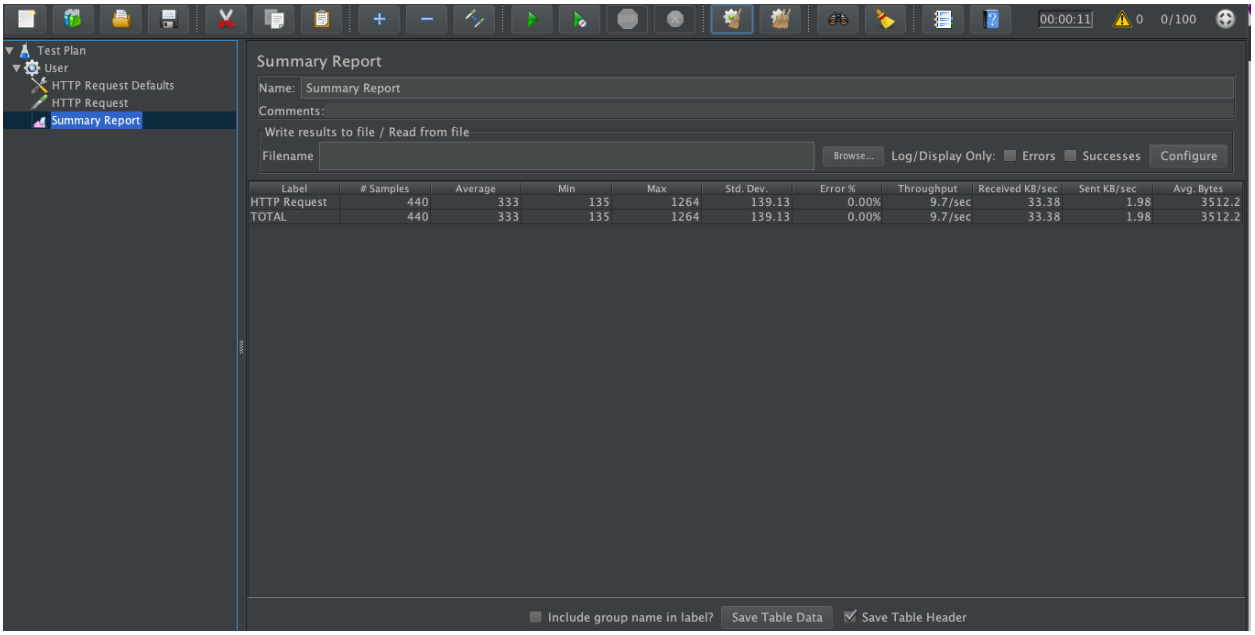

Figure 7: Performance test

Performance test:

We repeatedly use JQuery to get the details information of how the front end page be load. And analyze if the front end page have good performance. And for backend server, we also using JMeter to send multiple type of request, monitoring the performance of server.

5.4 Results

We implemented our features with testing. We could achieve displaying history data as a line input that simulating the movement of the truck with shifting of time. We combined the different variables of input together as one data point with one time point, and we use the GPS locations to locate nearby roads for rerouting process. We could find out one nearby location and point it, to generate a shortest path showing on the map for the rerouting suggestion. The path will find as much crossways as possibles for changing directions toward the destination point.

6 Project Management

6.1 Roles

Client: Adam Ryan Henderson Products Inc.

Advisor: Goce Trajcevski

Backend Developer: Junjie Wen

Product Manager and Scribe: Zhanghao Wen

Web Developer: Yuhang Xie

UI Designer: Xinhe Yang

Communication Leader: Tianhao Zhao

6.2 Responsibility

Junjie Wen work on our server part with NodeJS framework, the main task of him is to build a backend server to generate simulate data. Provided truck's data and re-route logic API for frontend.

Zhanghao Wen is our product manager and scribe. He researched and designed different snow event scenarios and the architecture diagram, proposed solutions to achieve overall project goal.

Yuhang Xie work on the frontend logical part with AngularJS framework. The website through the API to get the truck's data and dynamically let the truck move on the google map.

Xinhe Yang is our UI designer, she designs how our user interface look like, basically she used html and CSS.

Tianhao Zhao is our communication leader, he has contacted with our client and advisor and tell us how our project should be look like from client's perspective.

6.3 Timeline

We followed the design thinking process as shown in Figure 1.

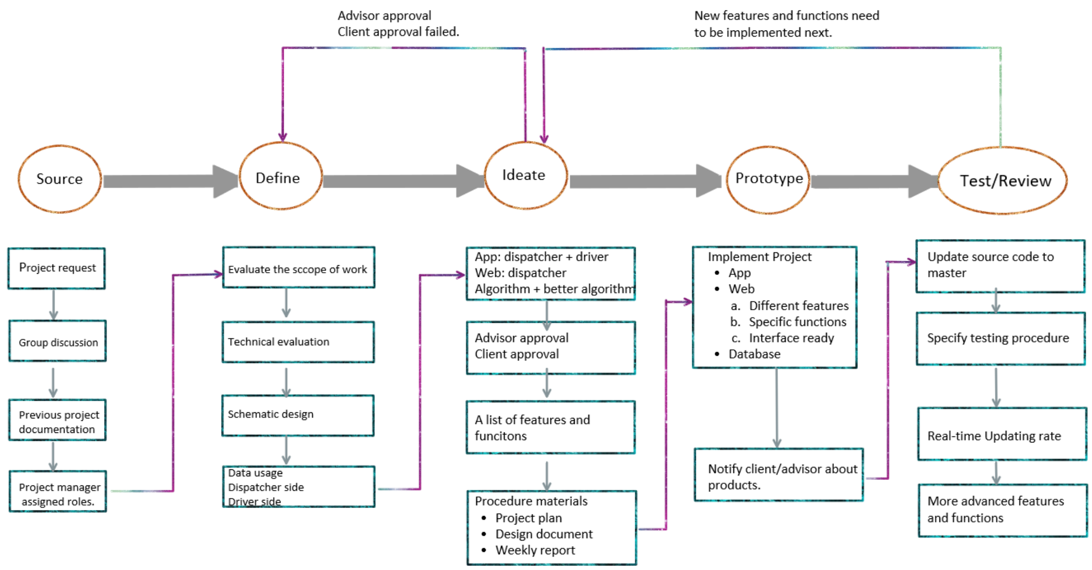

Figure 8: Design Thinking Process

We designed our project as achieving following project milestones:

- Make sure what framework and API need to be used
- Gather data from sensor, store, process, convert, and display data at front end interface
- Display data at certain points in truck's path which is generated by GPS
- Accomplish three types of view for 3 types of user
- Display road and weather conditions to help dispatcher make decisions
- Display rerouting suggestions

We have two semesters timeline as shown in Table 1 & 2.

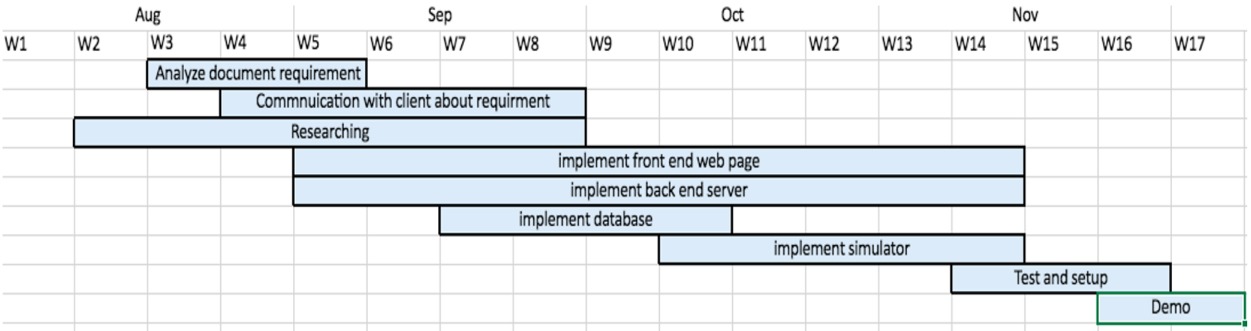

Table 1: First Semester Timeline

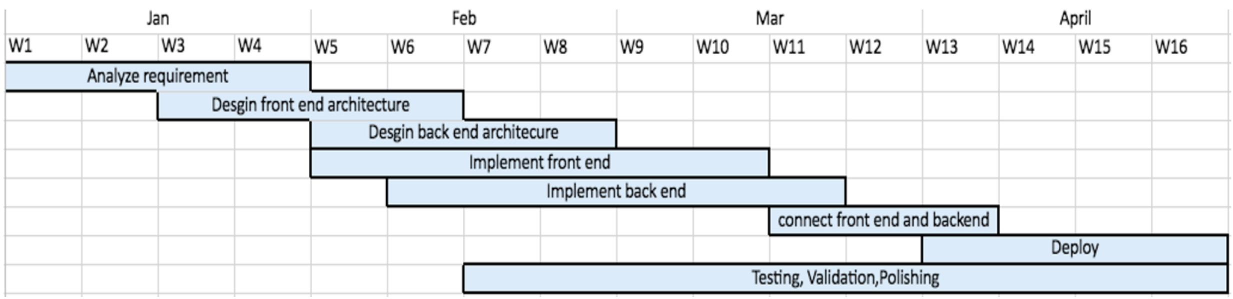

Table 2: Second Semester Timeline

We achieved most all of displaying features. But we come up with difficulty of achieving real time by testing live data with snowplows. Therefore, we changed our goals as utilizing history data to simulate real time scenarios. We also changed our ways of approaching rerouting process. The Table 3 shows our revised second semester timeline.

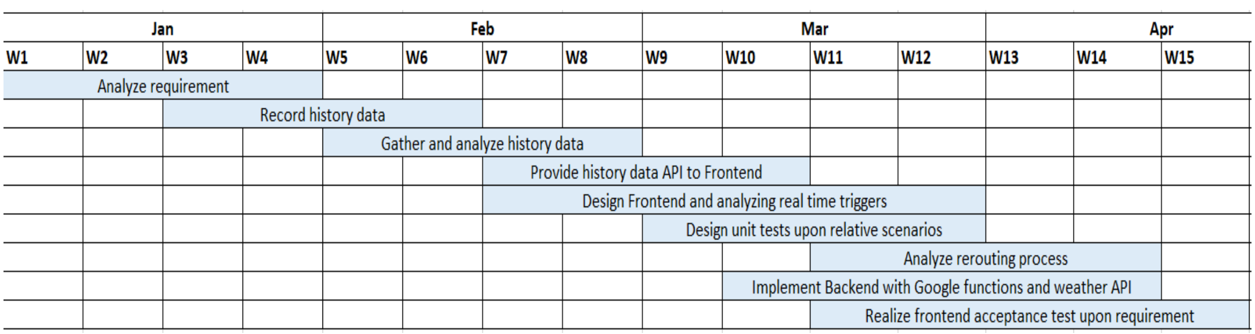

Table 3: Revised Second Semester Timeline

6.4 Lessons Learned

After this year of development and design, with the help of classmates and teachers. We finally completed the task of graduation design. In the process, we encountered a lot of difficulties. After consulting some materials and with the help of teachers and classmates, these problems were solved.

We understand that teamwork is the most important in development projects. When we have a good team spirit, we can better assign tasks and improve development efficiency.

7 Closing material

7.1 Conclusion

The goal of our project is to reduce time cost and financial costs and improve efficiency of snowplow operation combined with sensor system developed last semester.

Real time data from sensors employed in each truck will be took advantage to analyze decision of real-time reroute. MongoDB is used to communicate and store data from AWS server which is provided by Henderson Products. Basic algorithm to meet company's goal will be implemented first and then more advanced algorithm may be discussed later. Main function of our project is achieved by web application. Our overall system can integrate heterogeneous data, provide visual display of parameters of interest, and incorporate the decision-making plan.

Web-based interface can refresh and display the location of each truck and its historical route. And the date picker can be used to search historical data of all trucks for a period of time. It is mainly built for dispatcher who oversees overall snow operation and send assigned routes or reroutes information to drivers.

7.2 Future Work

We are able to analyze, store, and display past data. However, we were not able to display real time snowplows with live data since the data we can fetch so far is from a transfer station that our clients provided for us. Future work could change some API features to adapt live data connections with corresponding rerouting features. Live data testing could be done and use for future improvement.

Currently routing suggestion and route display is based on action table that our clients provided for us and real time weather information that we used by taking advantage of weather APIs. In the future, a more comprehensive algorithm could be considered for re-routing design combining factors such as road priority, real time traffic condition, and emergency.

7.3 References

[1] *How Technology Is Transforming Snow Removal Operations*. EASTBANC Technology, *How Technology Is Transforming Snow Removal Operations*, www.eastbanctech.com/binaries/content/documents/site/technology-insights/what-the-tech/files/transforming-snowplow-fleet-operations/transforming-snowplow-fleet-operations/site%3Afile.

[2] *Using Data, Snow and Ice Control Fleet Managers Achieve Goals and Automate Key Systems*. Certifiedcirus.com, 2017, *Using Data, Snow and Ice Control Fleet Managers Achieve Goals and Automate Key Systems*, certifiedcirus.com/wp-content/uploads/2017/04/Certified-Cirus-Winter-Road-Maintenance-Data-and-Reports.pdf.

[3]"Fighting The Snow With A Fleet Management Solution (Case Study)." *GIS Cloud*, GIS Cloud, 21 Nov. 2018, www.giscloud.com/blog/fighting-the-snow-with-a-fleet-management-solution-a-case-study/.

[4]"Filling the Gap between Conception and Unit Test." *Software Engineering Stack Exchange*, StackExchange.com, softwareengineering.stackexchange.com/questions/276797/filling-the-gap-between-conception-and-unit-test.

[5]"HENDERSON PRODUCTS - ABOUT US." *About_Us*, Henderson Products, Inc., www.hendersonproducts.com/about_us.html.

[6]"Road Conditions, Speeds, Travel Times, Traffic Cameras, Live Streaming Traffic Cameras, Road Closures and Road Work Information Provided by Intelligent Transportation Systems (ITS) a Branch of Colorado Department of Transportation." *Road Conditions, Speeds, Travel Times, Traffic Cameras, Live Streaming Traffic Cameras, Road Closures and Road Work Information Provided by Intelligent Transportation Systems (ITS) a Branch of Colorado Department of Transportation*, COtrip.org, www.cotrip.org/snowplow.htm#/default?SnowplowId=7401.

[7]"Snow Removal Is Essential to Keep a Clear Path to Your Business." *IBHS*, IBHS, 1 Oct. 2018, disastersafety.org/ibhs/snow-removal-is-essential-to-keep-a-clear-path-to-your-business/.

[8]Talley, Becky. "Plow Tracker Site to Let Denver Monitor Snow Plow Locations." *Our Community Now at Colorado*, Our Community Now, 12 Oct. 2018, colorado.ourcommunitynow.com/2018/10/12/plow-tracker-site-to-let-denver-monitor-snow-plow-locations/.

[9]TechMagic. “ReactJS vs Angular5 vs Vue.js - What to Choose in 2018?” *Medium.com*, Medium, 16 Mar. 2018, medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d.

[10]“What Is the Difference between a Distributed System and a Cloud Computing System?” *What Happens to the Planets When a Star Dies? - Quora*, www.quora.com/What-is-the-difference-between-a-distributed-system-and-a-cloud-computing-system.

[11]“Winter Maintenance of Roads and Runways.” *Winter Maintenance RCM411*, Teconer, www.teconer.fi/en/winter.html.

[12]“Xamarin vs React Native vs Ionic vs NativeScript: Cross-Platform Mobile Frameworks Comparison.” *AltexSoft*, AltexSoft.com, 4 Oct. 2018, www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/.