Real Time Route Optimization

PROJECT PLAN

Team Number: 18
Client: Adam Ryan
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: 11/30/2018 Version 3

Table of Contents

List of Figures

List of Tables

1. Introductory Material

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. In addition, he will provide us with the AWS and database runs from the different sensing devices on the trucks for the purpose of enabling a testing of our solutions. Besides, he guides us through the process of representing (i.e., pre-treating) the data on the map with trucks in snow operations, to help route arrangement of snow plows. 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 semester. He helps us making decision on how we can approach the tasks in multiple aspects, how we can go through our thinking process of finding main cores of objects in designing, and how we could overcome current challenges with overall thinking of the project.

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. Proposed Solution

Our solution includes building a web-based monitoring system that will help our users make an efficient and "smart" decision, with a simulation of databases that generate required types of data for fitting real utilizing purpose. Various sensors have been deployed in the individual trucks of each fleet. They are used for reading data from trucks. For real-time situation, sensors data are being continuously gathered and sent back to the server. Our system will aim at retrieving sensors data from AWS database where they are being collected, along with performing multiple analytic tasks. For better monitoring of the fleets' performance and for guiding operators selecting appropriate interface of viewing data in various-level details, we will enable a visualization of list of historical/current data. We may also build a cross-platform mobile application as our supplementary solution, which helps fleet drivers to get assigned tasks and to communicate with the operators, or for convenience purpose.

We will also look forward to implement more advanced algorithms by integrating into our solutions, so that we can provide extra feedbacks to users in order to add features in representation of fleets' performance during snow operations. Our solution may be able to provide feedback real-time analysis in order to suggest what level of dispenser rate should be set at certain road path during snow operation.

1.4. Operating Environment

Our final product is 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.

1.5. 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.

1.6. 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 has 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.

1.7. Expected End Product and Other Deliverables

A web application map with single or multiple interfaces mainly designed for three types of group. The first group is for public. Citizens can access our website to find out the nearby snow plow which displays on map as well as which road has been cleaned. The second group is dispatcher. They can know real time road conditions and do re-routing arrangement to optimize snow operations. The third group as admin managers, have full access to control and monitor fleets during snow operation.

A MySQL database which stores all types of data about truck information and users' information. Data could be queried by AWS to monitor and analyze performance of fleets.

The web application could get data from the server with same data types in backend simulator, and provide detailed truck and road information.

2. Proposed Approach and Statement of Work

2.1. Objective of the Task

Develop a cross platform web application for the Henderson Products Inc. The web application will provide a way for dispatcher to assign routes to fleet drivers, monitor fleet data during snow operations. Data need to be displayed include conveyor speed, spinner speed, GPS location, heading direction, altitude, plow up/down, material rate. Citizen can also access our website to find out the nearby snow plow which displays on map as well as which road has been cleaned.

We may develop a cross platform mobile application for fleet drivers to monitor their own truck supply conditions if needed.

2.2. Functional Requirements

- A web application, which contains:
  - Trucks' information with GPS data and dispenser data
  - Road conditions with traffic and road temperatures
  - Weather conditions
  - Temporary data of each truck along the path
  - Home page
  - Login page with multiple layers of users
  - History page to store past data
- A mobile application (if needed), which contains:
  - Trucks' GPS location on map
  - Individual truck's supply condition
  - Company's name
  - Communicate with operation center
- AWS server implemented by Nodejs, which contains:
  - Simulator to generate truck data for display, since we cannot get real data
  - Authentication member/company to login and logout
- MySQL database, which contains:
  - Table for truck gps and support condition data
  - Table for user information

2.3. Constraints Considerations

One of the constraints that we will have in terms of the non-functional requirements will be upload speed. Path of fleets are generated and represented by a series of dots. Each dot should include all the data mentioned above (GPS, material rate, equipment speed). So one truck will have one path and there is a fleet of trucks. We will need to store historical data (past snow event). Our client wants us to record dots as many as sensors send to accurately monitor truck condition. Therefore, it is necessary for us to think about how to implement our server and website to query from database and display at front end to deal with such a great amount of data.

We may not be able to get some physical data, such as road conditions sent back from radar, camera access on the street, since they may be relevant to government controls.

The standard protocols that we will follow during developing our project is agile protocol. We have multiple members of our team who are agile certified therefore we will use their expertise to maintain the agile process. The agile process is taught in school in many of our classes therefore they meet IEEE and ABET criteria.

2.4. Previous Work and Literature

Many local governments in north maintain a fleet of snow plows. In the winter season, speed and efficiency of snow plow fleet 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's 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 solution, we also track GPS and update on map in real time. One function of our solution called "Snow Event" could also report event starting time and ending time.

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

2.5. Proposed Design

The web application is going to be developed using JavaScript, with the AngularJS framework used on the client side and Node.js used on the server side.

The mobile application (if needed) is going to be developed with the Ionic framework used on the client side and Node.js used on the server side, our database is the MySQL database.

The Ionic main programming language is TypeScript, which is generally a superset of JavaScript that compiles to plain JavaScript. TypeScript increases the quality of the code because it helps to spot and eliminate mistakes during code typing. Using TypeScript is optional, and the application can be written in simple JavaScript.

The core reason using Ionic is that it provides us high efficiency of coding, since the percentage of code reusing for Ionic among all other cross-platform mobile frameworks is the highest. Although performance statistics is relatively low compared to others, it is acceptable in our project.

The reason we choose MySQL as the database for our project, is that the maintenance of the data is more important than the actual transaction management (i.e., loading, concurrency, etc.). Given that the main users will be the snowplow dispatchers and the managers, and the data will be loaded from the sensors, the assumption of not being concurrency-heavy seems to be justified at the moment. In addition, this selection allows for easier portability, with extra developments after the project has been completed.

Google Map API is used because it can handle a huge number of markers (10000+) at same time and it is more efficient than other map APIs. Our product needs to handle a large number of markers at same time and each of them represents the position of the truck. When the map is zoomed out, the user cannot clearly see how many trucks are in a certain area, since the number of markers in that particular area will be reduced to one. Because the google map API has a Marker Clustering feature, it can show the number of markers in a certain area when map is zooming out.

2.5.1. Block Diagram

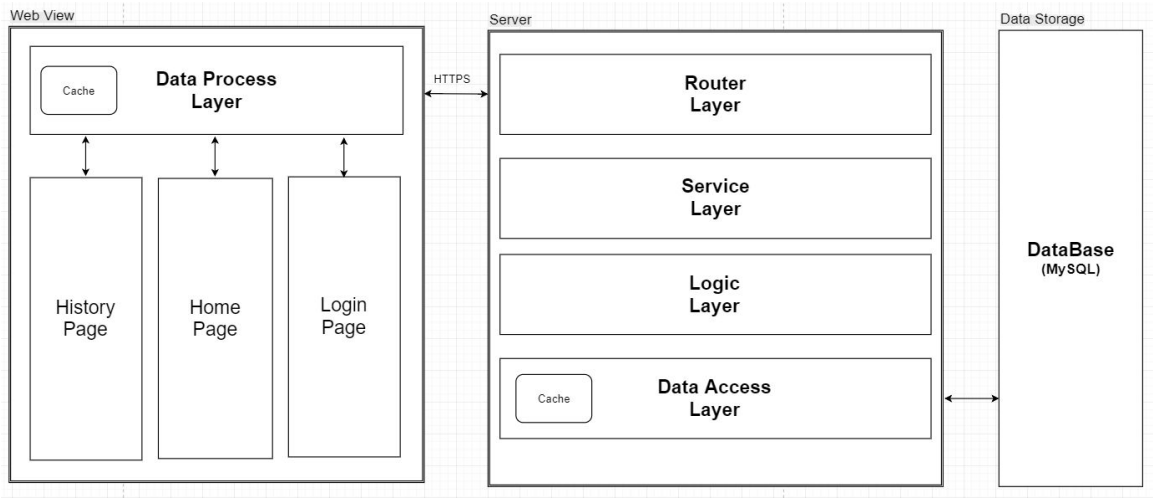

Figure 1: High Level Architecture Diagram

2.5.1.1. Web View

Data processing layer takes the simulated data (such as location or speed of truck) 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 real-time 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 type is identical. 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 from 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.

2.5.1.2. 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.

2.5.1.3. DataBase

The purpose is to classify data and store it in the database or to 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.

2.6. Technology Considerations

2.6.1. API

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

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

Bing Map 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.

2.6.2. Mobile Application

Our mobile application is mainly targeted and designed for truck driver, so put different drivers may use different mobile operating system into considerations, we could build iOS app by using Objective C and Android app by using Java.

Figure 2: Mobile App Framework Comparison

We note that we could also build a hybrid app. A hybrid app is a program that is built using HTML5, CSS and JavaScript and wrapped in native container which can used in different platforms. We have choices like Xamarin, Ionic, Nativescript framework. We expect to make a final selection based on the interaction with the client in the next two weeks.

2.6.3. DataBase

Development: Mongodb is a rather friendly environment for development 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 do have significant advantages than the other one.

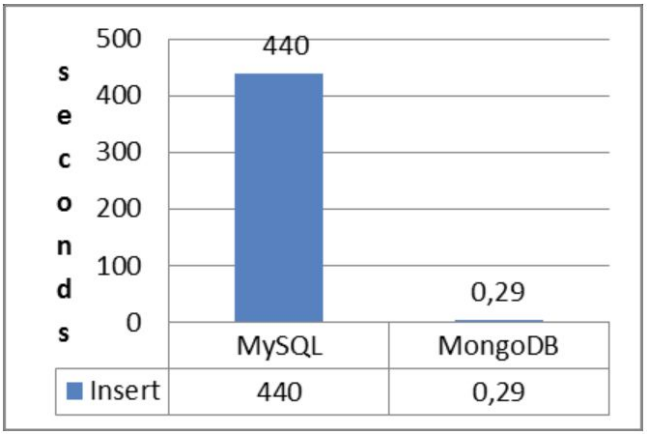

Figure 3: MySQL vs MongoDB

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.

2.7. Safety Considerations

Our team does not have many safety concerns due to having limited hardware components needed to use our project. The only safety concerns we have is for our driver to be operating trucks while using our project.

Considering Cyber-Security problem, there are two factors we need to pay attention. First, we need care about the data security. Different company shall not see another groups' information. Second, we need consider snowplow driver's security. The route we give should be safe to work.

2.8. Task Approach

Our design thinking process shows as Figure 4 below.

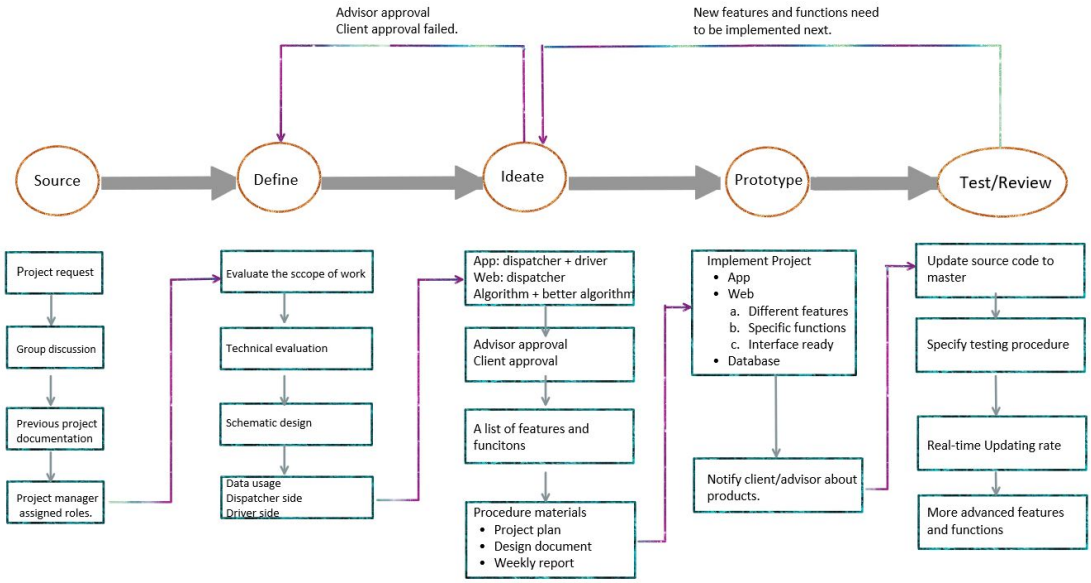

Figure 4: Designing Thinking Process

2.9. Possible Risks and Risk Management

After analyzing the entire software architecture, there is a framework risk theme for our project. First of all, probably there is a security risk. 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.

2.10. Project Proposed Milestones and Evaluation Criteria

- 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 conditions and help dispatcher make decision

2.11. Project Tracking Procedures

We are using a Trello board to record the progress of project and Minutes of Meetings with our client and advisor. We also use Trello to track which task is ready to be picked up, which tasks are assigned to different team members, what tasks have been completed.

Our teams write weekly report that detail individual work that has been done and our overall team progress in each week.

We have a phone meeting with client every week to update our progress. Client and our team ask question to each other to help us clarify current issues, popular solutions existing in the world, and project specifics.

We also have a communication leader that lead us to discuss course deadlines, assigned tasks, prepare presentation, and a project manager to make project decisions.

2.12. Expected Results and Validation

Our desired outcome is to have a fully working product that meets all of our functional and non-technical requirements. We believe that if we meet all of our requirements and scope which have been refined by our team multiple times and recently approved by our client the product will be satisfactory. The client has requested a web application that mainly used for management level which will be able to see path that fleet go through in each snow event as well as displaying various of data once clicking on certain point on path. Our team will try to design and implement more advanced algorithm to make real-time re-route an automated process, which is easier for our client to control fleets, reduce costs, and improve performance.

1. Display route of each truck on map:

We decide to use Google map API to implement this function. The API has function of recording historical points. All we need to do is to connect these points with different colors to achieve the historical route displayed function for each truck.

Our expected result is that the software can display points on map based on location data obtained from the server each time. When truck on the map moves to next point, the historical point and the current point are connected by lines of different colors.

2. Display history data of each historical point

In order to obtain data from server side, we will use http to send request. After obtaining data, we use the function of the informed window that comes with Google map API to implement historical data display requirements. The expected result is that when user click on a historical point, a prompt window with historical truck status data will be displayed in the upper left corner. This window will not move as truck position is updated.

3. Login with different groups of users

Using http to send username and password to server side and the server matches the information with user data in database and obtains user's privilege level. The server then returns the privilege level to front end, which displays the corresponding function according to different level.

Our expected result is that the program can display different functions based on user's privilege level. When user's level is public, the program only displays truck location, map, time and altitude. When privilege level is “user”, the program will displays status of truck in addition to all the data showing in the public privilege. For example, speed of spinner and the amount of water sprayed. When logged in user is “admin”, the program will not only display the entire

contents of the public and user levels, but also the historical data and historical route of each truck.

4. Display historical data for a specific time period

Using the calendar component in bootstrap to implement this functionality. After clicking submit button on the component, front end will send request via http and get historical data of certain period of time, which will display on new interface. The expected result is that when we use the date picker to select a specific time period, program will generate a new interface and display the historical data obtained from database.

2.13. Test Plan

On high level, we will carry out project plan based the idea of Figure 6. While developing, we will build unit test for our code for every module. And after finish complete component prototype(backend, database, simulating data generator), we will build integration test for them individually. After that, we will work build system testing for our overall project. Finally, do acceptance testing with the help for our client.

Unit test: A bunch of unit tests should be built, which will cover all platforms, including android, IOS, front-end and back end code. And should be executed every time the code be changed.

Integration test: There should be individual integration test for different platform, including Android, IOS, Front-end and Backend.

System test: Test engineers should implement test code based on black box test policy.

Acceptance testing: The whole team will work on final evaluation of acceptance.

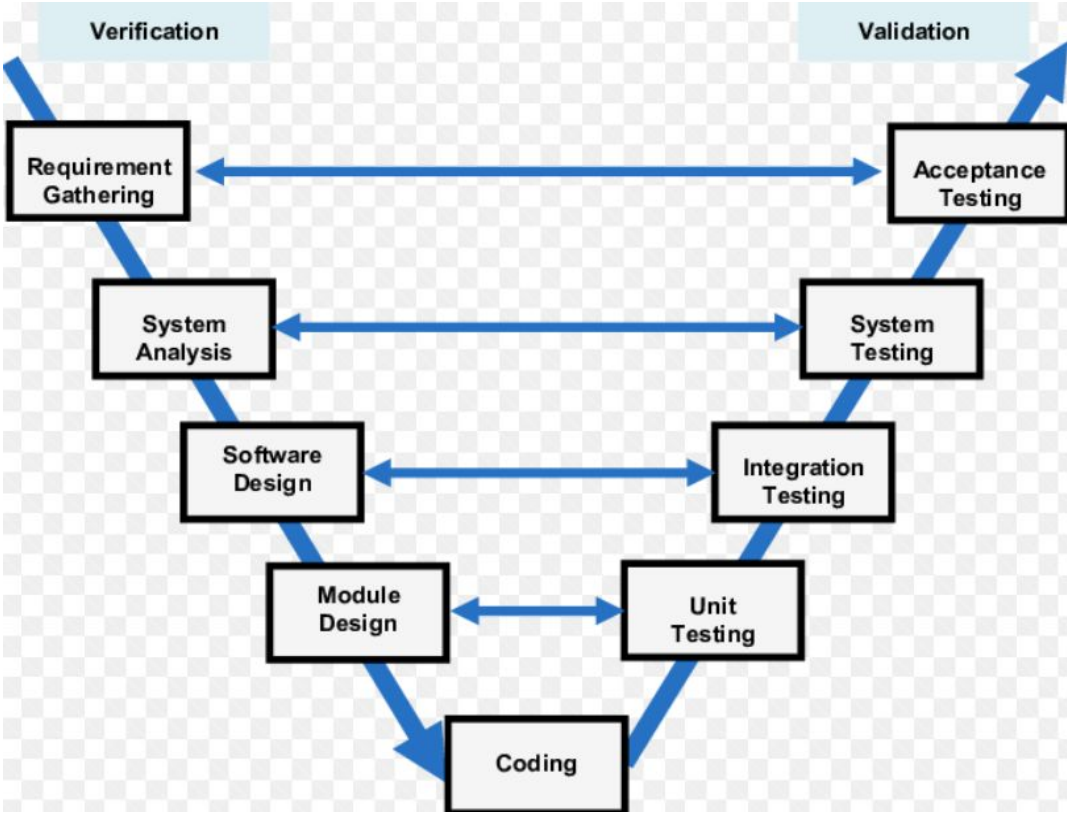

Figure 5: Testing Diagram

3. Project Timeline, Estimated Resources, and Challenges

3.1. First Semester Timeline

The following is a Gantt chart outlining the proposed timeline for the project's development through the first semester. The blue bars indicate project phases

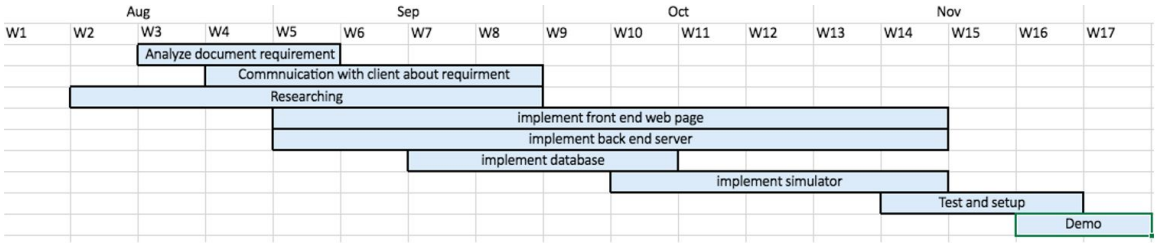

Table 1. First Semester Timeline

The first few weeks of the project, we will spend time understanding the requirements, researching, and prototyping. Most of our design thinking happens here and we make sure to meet with the client often in order to get the requirements. After the first few weeks, we start working building front end interfaces, including a web page and a cross-platform mobile app to make dispatcher side (manager) of snow operation "smarter" by showing truck condition at certain time points in its path.

Meanwhile, we will work on build a backend server to deploy the web page, and also build a database, based on the back end server. After that, we will build a microserver for simulator to motivate front end.

Basic functions of our project are going to be implemented and tested before the end of first semester. More work for the advanced data analysis from various algorithms will continue into the next semester.

3.2. Second Semester Timeline

The following is a Gantt chart outlining the proposed timeline for the project's development through the second semester. The blue bars indicate project phases.

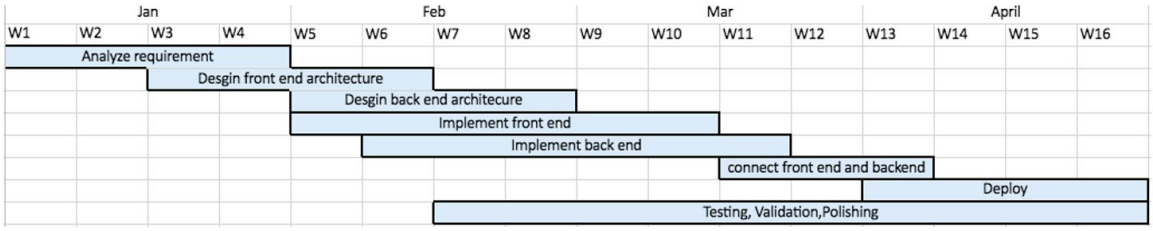

Table 2. Second Semester Timeline

The plan for the first 2 months is to continue working on the web and mobile application to help dispatcher to monitor fleet of truck during snow operation. At this point, data from client is probably more detailed and we need to design more advanced algorithm based on various real-time scenario to help dispatcher make decisions.

3.3. Feasibility Assessment

Basically, this project is divided into 2 parts in 2 semesters. We expect to have a fully functional, production ready, mobile application as well as web application by the end of second semester. Our system will employ the use of a frontend backend architecture allowing our team to make changes to either system without the need to change the other.

We expect to have all the basic functional requirements met at the end of first semester, and it includes web pages with 3 types of map view for 3 groups of users: public, member, and full-access dispatcher which give information about fleet at different level. Therefore, we need to think about permission and how they can access different pages. Once snow, dispatcher can create a weather event by our website which can send route and other related information to driver. What's more, dispatcher can access to our webpage and see the path that fleet go through and click certain point on that path, data like spreader speed and plow height will show up. This will help central manage office monitor performance of fleet.

Overall challenge for each of our group member is that we need to teach ourselves a lot of new skills in terms of programming language, usage of API, UI design ,etc. The knowledge from college helps us how to define what problems is and we need to access and solve all kinds of problems based on newly learned skills.

Another potential challenge is that none of team member have touched snowplow before, which is what our products will be used for. So speaking to deeper technological understanding, the actual contents of our design become not very strong.

Our actual physical challenge is that data is given later than what our team expected, which we build our own data simulator to generate the data we need.

Another challenge in the future will be that there will be so many aspects that we need to consider and analyze and apply those data into re-route algorithm. The aspects include traffic condition, which may affect some trucks' working time, weather, which leads to snow plow spreader rate, gate height setup, spinner speed, etc. Another important aspect is road priority. Once operation started, we need to determine which roads within snow operation area, including roads in rural area, main road in a city (for example Lincoln way in Ames), and highway, have what level of priority.

3.4. Personnel Effort Requirements

As seen below, Table x is a table of the major tasks that we need to accomplish and implement in order to complete our project. In addition, we also have workshops, preparing project presentation, listening tech-talk, research, and documentation.

Task	Description	Estimated time
Create Requirements	Create a list of the requirements needed for the software	10 hours
Research	Background information and software tools analysis, framework, and API choice	20 hours
Design algorithm	Design the data analytics algorithm. Improve algorithm based on new types of data	20 hours
Front-end application	A web page designed and implemented mainly for central manage system and a mobile application designed and implement for end-user	40 hours
Transfer Data	Get data communicate between database, server, and front-end application	20 hours

Design database architecture	We will need to create a database that can manage the different types of information along with dealing with the large amount of data	20 hours
Testing	Test accuracy of the algorithm or improved algorithm	15 hours

Table 3: Major Tasks

3.5. Other Resource Requirements

- External resources to maintain the team's documentation and Git instance.
- Sensor from Henderson for testing
- Google map API
- Weather API

3.6. Financial Requirements

The main financial resources needed to complete this project are minimal during the development cycle with our team only needing one server to host both our frontend and backend. Our client provides us AWS (Amazon Web Server) and it costs $650 per year. At this stage, all the software that we will be using to implement our solution is free for us to use.

4. Closure Materials

4.1. Conclusion

The overall goal of our project is to reduce time cost and financial costs and improve efficiency of fleets of snow plow during snow operation. The major problem our client is facing is that they have limited control of fleets of truck once dispatcher assign which roads they need to clean. Based on weather condition and different types of road, driver should setup equipment rate differently. In order to monitor and guide truck driver, we deployed sensors in fleets which can read all data and send them back to client, so they can analyze and improve fleets' performance in the future. We are currently working on the interfaces where the client can access that information.

Or even better in the next semester, we will think about and design more advanced algorithm that can improve fleets' performance real-time during snow operation with feedback available to our client in a real-time manner.

Real time data from sensors deployed in each truck will be took advantage to analyze decision of real-time reroute. MySQL is used to communicate and store data from AWS server which is provided by Henderson Products. Main interfaces of our projects will be achieved in two form: web application and mobile application. Mobile app will be built for snow driver to ensure that they understand condition of truck such as supplies (sand, salt). Web will be built for dispatcher and manager who oversees overall snow operation and send assigned routes or reroutes information to drivers.

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

[5]"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.

[6]"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/.

[7]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/.

[8]“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.

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