

Data Analytics for Snow Plow Truck-Data

Senior Design - May 2018

Team 23

-

Project Plan Draft #1

Table of Contents

1. Introductory materials	4
1.1. Acknowledgement	4
1.2. Problem statement	4
1.3. Operating environment	4
1.4. Intended user(s) and intended use(s)	5
1.5. Assumptions and Limitations	5
1.5a. Assumptions	5
1.5b. Limitations	6
1.6. Expected end product and other deliverables	6
2. Proposed approach and statement of work	7
2a. Proposed Approach	7
1. Functional requirements	7
2. Constraints considerations	7
3. Technology considerations	7
4. Technical approach considerations	8
5. Testing requirements considerations	8
6. Security considerations	8
7. Safety considerations	8
8. Previous work / literature review	8
There has been no previous documents that are needed to be reviewed prior to working on this project.	8
9. Possible risks and risk management	8
10. Project proposed milestones and evaluation criteria	9
11. Project tracking procedures	9
2b. Statement of Work	9
Familiarize ourselves with the CANBus system	10
Structure initial database for the Web App	10
Build backend code skeleton	10
Research good front-end designs	11
Prototype & Deliver	11
3. Estimated resources and project timeline	11
3a. Estimated Resources	11
3b. Tentative Schedule	13
4. Closure materials	13
• Closing Summary:	13

1. Introductory materials

1.1. Acknowledgement

Thank you to James Timmermann and Henderson Products for working with us and providing us with the information for our project.

1.2. Problem statement

The main goal of the project is to help with the visualization of data that is being received from sensors equipped to select snowplows. Currently the data is being sent from these trucks containing information about the plow including its coordinates, fuel consumption, and select statistics about the plows performance. That data is then being stored on a server, but is currently in an undesired format to be analyzed. Ultimately the product will be a web application that will display these statistics about the truck that can be studied to forecast the trucks performance in the future. With this knowledge, quality issues can be spotted before they affect performance of the truck, saving money from a potential breakdown.

The solution to this as previously stated is to create an application that hosts and helps visualize the data being received. At its current state the data is being stored on a server hosted by Henderson Products. So our application will be constantly pulling new data as it makes its way to that server, and will convert it into a readable format to be stored in our database. We will then design an interface that will allow the info to be easily read and analyzed. Most likely the product will allow clients to log on and allow them to view and monitor their own trucks and data.

1.3. Operating environment

As our project deals with transferring, converting, storing, and accessing data; we don't expect our product to be exposed to any notable conditions. We expect our end product to run on one or more servers, and be capable of being accessed by employees working at Henderson most likely in an office setting. Due to the fact that it will be running on servers, it may be important to account for possible failures in that domain. It's important to note that while the data will be coming from loggers on Henderson vehicles, the actual capturing of data is already implemented and is considered out of scope for this project.

1.4. Intended user(s) and intended use(s)

The intended users of the CANBus data app will be the good employees of Henderson products and possibly even their clients. The clients using the web app, however, is a bit of an assumption (as stated in the next section). Any other intended users could involve the operators of the dispensing units so that they may see the data of the vehicles they operate. It will be important to make sure that only the designated people can use this software so that Henderson's data does not get into the wrong hands.

There could be many end use cases for this app. First of all, it will reduce the need for calculations done by hand by our company contact James. This had been a very large waste of time, taking nearly half an hour to convert 3 minutes of data. Key details about the data could inform the employees when things are not going right with the CANBus system. Whether that means a part on the vehicle is broken the hydraulics readings are not what they should be, it is essential to know when things are going wrong so they can fix it as soon as possible. The data they retrieve from this app will also be used to determine better ways to create products for Henderson's clients, making their snowplows able to withstand the tests of time and the harsh environment of winter.

1.5. Assumptions and Limitations

1.5a. Assumptions

1. We will receive all information on how data converts from hexadecimal bits into relevant data.
2. We will receive access to Henderson Products server in order for us to be able to pull files from it and convert.
3. We will be able to contact our client for important information within a reasonable amount of time.
4. The web application will have users with unique logins to guarantee security and data integrity.
5. Vehicles will each have their own unique identifier.
6. Clients of Henderson Products will be able to view data sent from their own vehicles
7. The maximum amount of simultaneous users shall not exceed 100.
8. The web application will be a single page that allows the manipulation of data to be viewed and read effectively and efficiently.

9. The web application will adhere to the branding of Henderson Products
10. The completed product will not be viewed outside of the United States
11. The website will only need to be in English.
12. New desired features will be given with the understanding that time may be a factor on successful implementation.
13. We will not need to develop for the sensor, all data being sent is all data needed.

1.5b. Limitations

1. The project will be completed in its entirety by May 2018.
2. The project will not require any hardware design.
3. The project budget shall remain at zero dollars.
4. The project will not use more cellular data than the initial transfer from vehicle to server.
5. No data will be lost in transformation or translation.
6. Data will be moved to the web application within 24 hours of creation.
7. Adding a new user will take less than 2 minutes.
8. Clients will only be able to view data from their own vehicles.

1.6. Expected end product and other deliverables

Data Log Parser

The data log parser will take the log files from the trucks, parse them, and return the data in a more manageable form. Initially this parser could return a human readable file. However, the end goal for the parser is to take the data it returns and insert it into a database.

Database

We expect to deliver the design for the database that will allow our client to store, access, and organize the truck data. The database will be populated with data extracted from log files by the parser. The database will be accessed by the client through the web application.

Web Application

The web application will be used by the client to view the data being sent from their trucks. The application will offer different ways for the client to view the information such as: graphs, maps, and tables.

2. Proposed approach and statement of work

2a. Proposed Approach

1. Functional requirements

- **Data Conversion** - Currently data is being hosted on an ftp site in the form of obfuscated log files filled with hexadecimal values. This data needs to be converted into something that makes more sense for humans.
- **Data Storage** - The converted data will need to be stored somewhere where it can be queried for in useful ways for analysis.
- **Data Access** - An interface will need to be build that gives an intuitive way to query for data that will be useful.
- **Data Analysis** - An interface will be needed to outline useful trends in data using graphing tools.

2. Constraints considerations

- **Time** - This project will need to be working and have most of the core functionality before May 2018.
- **Cost** - We will need to consider costs when determining the overall architecture of our project, as this will affect the cost concerning server usage.
- **Software Licenses** - Because our project will be used in a commercial setting, we will need to be sure that the licenses for any of the software libraries we are using gives us rights to use it for this project.
- **Vehicle Access** - We will have limited access to the trucks that the data is being logged from. This shouldn't have any major effects on our project, as the data transfer is already being handled, though having access to the trucks may give us a better idea of what the data is for.

3. Technology considerations

- **Backend** - Currently we are planning on having a backend service built with Nodejs to process and store data, as well as to handle requests from the frontend.

- **Frontend** - We plan on using Reactjs to build a Single Page app for the frontend. By using a frontend framework we hope to simplify development and create a cleaner interface.
- **Database** - We plan on using MySQL to store the converted data, as most of us have experience with it and our client is already using it within their company.

4. Technical approach considerations

- **Development Process** - We plan on employing an agile development process, as nearly all of us have used it and it works well for many new software projects.
- **Development Methodology** - We plan on using test-driven development in the hopes that it will improve the reliability and correctness of our software; as well as help us identify key issues earlier.

5. Testing requirements considerations

- **Unit Tests** - We want to make sure our software works the way we designed it. Writing unit tests for our software will help us do this, and will help us catch problems earlier on.
- **Integration Tests** - Our software has a few different parts that interact with each other, and we want to be sure that when we change one that they will still work together as a system. Writing integration tests will be a good way to ensure that this is the case.

6. Security considerations

Our project should limit access to data to the people that need it within Henderson products. This could mean locking it down to an internal network, or requiring some sort of authorization to access, or some combination of the two.

7. Safety considerations

- As our project is a web application, there are no safety considerations that need to be watched to ensure proper delivery.

8. Previous work / literature review

- There has been no previous documents that are needed to be reviewed prior to working on this project.

9. Possible risks and risk management

- **Developer Time Constraints:** It will be a possible risk that developers lose hours on the project as schedules become more complex and busy throughout

the school year. Time will be the biggest factor when dealing with risks on this project. The goal is to have a steady sprint plan with achievable weekly goals, these will have to be flexible with the developer however to allow for the intended work to get done within their schedule.

- **Learning Curve:** Again time will factor into this, the team is already hard at work to get up to speed with the current state of the product, the technologies planned to be used, and the end goals of the project. This could potentially take longer than planned leading to heavier weekly loads down the line.

10. Project proposed milestones and evaluation criteria

- **Project Plan:** The first milestone to be completed is establish a project plan, we will need our architecture and deliverables to be clearly laid out to create success throughout the project
- **Establish Connections:** The next milestone would need to be a connecting script to be ran. This script will have to pull the data currently hosted on the ftp site, convert it into the desired format, and then store it in the database. It will be evaluated by a successful connection on both sides.
- **Create Interface:** Once the data is being stored properly the team can start on the actual functionality of our deliverable. We will need to create screen designs to first prototype out our interface. From there we can start presenting the data in a way that will ultimately help the client analyze it. User testing and sign offs from the client will be crucial along the way to ensure it is properly formatted to spec.

11. Project tracking procedures

- The project will be tracked using gitlab's built in project tracking features. Issues and feature requests will be logged on the website and assigned to developers in weekly meetings. With these goals assigned we can make sure work is being completed throughout the week.
- Overall the project will be also tracked against a larger timeline of deliverables. This will be laid out, and may change depending on the scope of the project. A loose schedule will still be helpful to track the overall state of the project to get it delivered on time.

2b. Statement of Work

1) Familiarize ourselves with the CANBus system

- a) Objective
 - i) It will be necessary to know what we're working with
 - ii) We've started on this already but it may take some time to fully understand it all
- b) Approach
 - i) Read the documents Henderson has sent us and ask as many clarifying questions as possible
 - ii) Consult the internet as necessary
- c) Expected Results
 - i) Be able to understand the conversions of hexadecimal data to data that is meaningful and understandable
 - ii) Be able to know the format of the data coming in from CANBus

2) Structure initial database for the Web App

- a) Objective
 - i) Create a database that will be a good fit for Henderson Products and meets their standards
 - ii) This will mean they have less work to do and will create a familiar working environment
- b) Approach
 - i) Since Henderson Products uses mainly SQL, we have decided it will be smart to employ it in this project as well
- c) Expected Results
 - i) Using a database they are familiar with will make taking care of the data once we are off the project much easier
 - ii) This will also reduce potential bugs down the line

3) Build backend code skeleton

- a) Objective
 - i) The backend code will do a lot of the heavy lifting and will be essential to the success of this project
 - ii) We need to have a good plan of what this code will look like
- b) Approach
 - i) Having some experience with Node.js, we've decided to use it to handle the backend
- c) Expected Results
 - i) Having a good skeleton will allow us to see what will need to be added to the project and what may or may not be necessary
 - ii) This will also allow us to restructure the database before we write too much code

4) Research good front-end designs

- a) Objective
 - i) Have something to show Henderson so that they can give us feedback on what they're looking for
- b) Approach
 - i) Consult the world wide web for good app design involving graphs and data charts
 - ii) Observe the different data types to determine what graphing tools would be most suitable
- c) Expected Results
 - i) This will give us a good idea as to how to go about making our prototypes
 - ii) Keep Henderson Products in the loop about how the app they're going to be working with every day will function and be designed

5) Prototype & Deliver

- a) Objective
 - i) Create a varying amount of designs and functionalities to show Henderson what our app is capable of offering them
- b) Approach
 - i) Flesh out database and backend code for the final stages of the project
 - ii) Iterate through making different frontend designs and creating the necessary backend calls to accommodate all needs
- c) Expected Result
 - i) Get as close to a final product as possible during the allotted time for this project (as we all know, a software project is never *really* done)
 - ii) Deliver this software to Henderson so they may use it as they please

3. Estimated resources and project timeline

3a. Estimated Resources

Task	Personnel	Resource	Financial
1.Familiarize canBus	3	Henderson Data and Documentation	\$0.00

2. Structure Database	5	Server	\$0.00
3. Build Backend Skeleton	5		\$0.00
4. Research Good Frontend	8	Henderson Opinions	\$0.00
5. Prototype and Deliver	>13	Henderson Data and Documentation, Server, Opinions	\$0.00

We decided it would be best to rate the personnel requirements based on the Agile Fibonacci Sequence. This scale follows the Fibonacci Sequence, with larger numbers indicating more time and more complex tasks. This best fits our needs because as things get more complex the possible setbacks or jumps ahead get larger as well. For instance, developing a pleasing GUI that Henderson Products likes the look of and finds easy to navigate could take a first try or take quite awhile to develop and perfect.

We are developing a software application and don't see any future financial needs in the development process. The IDE's we plan to develop in are all free and the server space is provided for us. We also don't have to pay for internet on campus.

With step one we decided a 3. It will take each of us some time to get fully accustomed to working with canBus and familiar with the canOPEN standards. Henderson has provided us with our needed Data and Documentation. We will use it to become more familiar with their goals for the project.

Step two is our first step in creating the software. Our goal for this step is to have the database completely envisioned. This way when we move into developing the web application we will be able to easily sort and manipulate the data. We plan on doing some prototyping at this stage to make sure things behave as we expect them to. It is possible the prototyping at this stage will change the scope of the step and increase its time; however, this will hopefully decrease time needed on development later. We need server space to house our database prototype in order to implement this step.

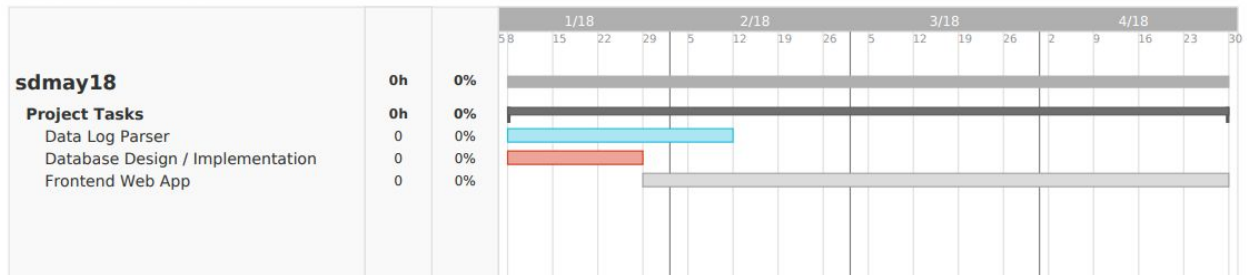
Step three is where we will start to create the documentation for the backend of the web application. Here is where we will name function calls. We will document each function fully so that as we write the code behind it later we will easily be able to tell how the whole application comes together. With this skeleton we will be able to more appropriately be able to plan out tasks and assign team members to their implementation. A successful step three should lessen the work needed in future steps. We want to be very thorough in this step and have a well planned out application. After this step our backlog should swell and programming can really start.

Researching and wire framing the front end is next. We wanted knowledge of how the application will run in the back before we started to develop the front end. Simply wire framing the front end would not take too much time. We think it will be important to display what the actual application would look like on screen. If we are careful about the design it should take less time than we originally have planned. A

green light on our first design could move this down to a 3 or 5. However, we are prepared for the possible delays that this step could take. Getting quick responses on our designs is essential for this step to progress quickly and smoothly.

Our last “step” is a very large one. Without having the database and skeleton designed it is difficult for us to break this into smaller steps. We see this step as being most of our time spent during 492. The actual coding and testing of the software. As we complete more steps this one will become more clear and defined allowing us to later expand our plan and detail out each step more effectively. This step will require all of the previous resources because it will be the development towards the final product.

3b. Tentative Schedule



1. Data Log Parser
 - a. First month of production
2. Database Design / Implementation
 - a. First three weeks of production maximum
3. Frontend Web App
 - a. Last 12 weeks of production, after the database design is complete

4. Closure materials

- Closing Summary:

The goal for our project is to make it easier for our client to understand the data that they are receiving from their trucks. Currently, the information is being stored on a server hosted by Henderson Products, and then manually converted from hexadecimal values. We are going to be creating a web application that will help convert this information faster. This information is being used to predict the performance of each individual truck and will help the clients to better monitor their own trucks.