Data Analytics for Snow Plow Truck Data

Project plan

Team 23
Client: Henderson Products
Adviser: Dr. Goce Trajcevski
Alex Smola/Design Lead, Alan Peine/Git Master, Colin Heinrichs/Test Engineer, Evan Warych/Report Manager,
Michael Entin/Meeting Reporter, Zach Wilson/Meeting Facilitator & Communication Lead
sdmay18-23@iastate.edu
http://sdmay18-23.sd.ece.iastate.edu

Revised: 10/27/17 Version 2

Table of Contents

List of Figures

Include these sections only as needed.

List of Tables

List of Symbols

List of Definitions

1 Introductory Material

1.1 Acknowledgement

Thank you to James Timmerman and Henderson Products for working with us and providing us with the information for our project. Thank you, Dr. Goce Trajcevski, for your help and guidance with the the direction and management of 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 on a group of 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 Users and Intended uses

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.

delivery: Functionality for data log parsing should be finished by the end of December 2017; however some additional data operations may need to be implemented as we decide how it will be stored in the 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.

delivery: We plan on having our database relationships planned out by the end of December 2018, and our database set up by February 2018.

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.

delivery: We hope to have a working prototype with basic functionality done by: March 2018, and the final version finished by May 2018.

2 Proposed Approach and Statement of Work

2.1 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.

2.2 Constraints considerations

- **Security Requirements** - 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.
- **Responsive Requirements** - Since our project is a web app, our applications should be fast enough to meet the needs of a company. Any request for data should take no longer than 5 seconds to complete and populate on the page.
- **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.

2.3 Technology considerations

- Data Log Parser:
  - ANTLR
    - A popular language parser generator
    - Might be too advanced for what we need
  - Build our Own Parser
    - Can use a language that we know well (Java)
    - Does not need to be complex
    - Can set up configuration files to allow different kinds of data to be recognized

- Data Storage:
  - SQL database
    - Henderson works with SQL a lot
    - Familiar with the syntax and technologies
    - Short setup time
    - Consistent & reliable
  - Data Warehouse
    - Works better with companies in general
    - Can store large amounts of data
    - A new concept most of us aren't familiar with
  - MongoDB
    - Non-relational, which may not be good for this scenario
    - Can work much faster than an SQL database
  - Apache Cassandra
    - Designed to work well with large amounts of data
    - Also Non-relational
    - No experience from team working with it

- Web Application:
  - Frontend
    - AngularJS
      - Uses a MVC architecture for developing frontend applications
      - Experience from multiple team members
    - ReactJS
      - Uses uni-directional data flow for organizing components
      - Organizes views into reusable components.
      - Makes it easy to port to mobile with React Native
  - Backend
    - NodeJS

        - Can have full stack javascript
        - Some of our team has exposure to Node
    - Laravel Framework
        - PHP is usually not fun to code in
        - The framework is really great & can make some good looking apps
        - May not have the graphical functionalities we're looking for; can't be used to create single page apps on its own.
    - Java Spring
        - Nearly all of our team knows Java
        - Used widely, has great documentation
    - Apache Thrift
        - RPC framework that won't limit us in choice of language.
        - Gives us more freedom in terms of software design.
        - Forces us to define our service interfaces and the models that they work with, leading to a better design.
        - Abstracts away communication between software modules. (In our case the client and server)
- AngularJS & NodeJS
    - Javascript is really great for making webapps
    - Angular is one of the leading techs for single page apps
    - Some of our team is very familiar with both and others have some experience in both

2.4 Safety considerations

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

2.5 Previous Work and Literature

One of the main drivers of using big data to gain insights into the optimization and tracking of trucks is US Xpress Transportation. While their product focuses on the performance of their freight trucks, there is still plenty to be observed by their work. Information on this can be found on their website [2] but a more concise write up on their work can be found on a technology news site called datafloq [3].

A few key issues differentiate our product from the one for US Xpress, the main distinction is their real time tracking system versus our projected system of post analysis. However, it was beneficial to learn how they use several different data sources and combined them for analytics, this is a key feature of our application. Furthermore, getting

familiar with the idea of geospatial analysis was helpful as we do plan to map out the trucks data in a similar fashion.

2.6 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.
- **Security Risk**: Our project should limit access to data to the people that need it within Henderson Products and their clients. This could mean locking it down to an internal network, or requiring some sort of authorization to access, or some combination of the two.

2.7 Project Proposed Milestones and evaluation criteria

1. **Familiarize ourselves with the canBus system, and other technologies:** Our project is based on information arriving from a data logger that is adhering the the canBus system. It is important for us to be able to understand how and why things are being done on the logger in order for us to create and informed plan and design. We also need to perform research on what technologies will be most effective.
2. **Project Plan:** The next thing 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
3. **Design Document:** We will want to complete a design document which details out our full specifications that satisfies our functional requirements, design constraints, and when followed will produce a product that performs its intended uses for the intended users.
4. **Research and Wireframe Effective Front-End Designs:** Our client will need to be able to use the website with ease. We feel it is important to get a head start on wireframing the front end and get their feedback on it. With this perfected it may help inform our decision on a good technology to use.
5. **Prototyped Database and Converter:** Our group needs to be sure that how we plan to handle the data will be reliable and efficient before we scale it to the full project. Querying our database efficiently is very important to our project. We will

most likely prototype several different systems before deciding on one that is easy to write and scale for the large amount of data that the snow plows produce.

6. **Complete Final Drafts of Both the Project Plan and Design Document:** With our research completed, website front end designed, and subsystems designed we can finalize our documents. This marks the end of the first semester.
7. **Convert Data:** After making sure that we can convert the data successfully in a prototype environment we will need to scale it for all incoming data packets. This milestone will most likely be the first thing completed during the second semester.
8. **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.
9. **Create Interface:** Once the data is being stored properly the team can start on the actual functionality of our deliverable. We will need to implement the screens we designed for 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.

2.8 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. However, our goal is to stick to our Project Timeline as close as possible and achieve each milestone sooner than expected. This would allow fine-tuning of deliverables and prototyping additional features.

2.9 Objective of the task

Describe the goal of the task. Depending on the type of project, the resultant end product can vary significantly:

– An actual hardware/software product – The design of a product
– A process to accomplish something
– A service to be performed

– A simulation or a set of calculations – Some combination of the above

Task	Objective
1	Obtain a full understanding of the data logging system as well as modern technologies that may help us
2	Complete a first draft of our Project Plan document that mostly details out the full two semesters of work
3	Complete a first draft of our Design Document that clearly lays out our projects specs, requirements, needed technologies, analysis, and process
4	Present a wireframe to Henderson Products that they accept that will be used as the front end user interface of our web application
5	Create a scalable data converter on how we plan on converting the data. Create a scalable database for the large amount of data that can be accessed with simple queries
6	Complete our final drafts of the Project Plan and Design Document to show our successful completion of the first 5 milestones and our preparedness to build our project in the second semester
7	Scale our prototyped solution to a script that can grab the data from the FTP server, convert all types of data packets, and prepare the converted data for transmission to the database
8	Implement what we've decided is the best solution to storing the large amount of incoming data so that it can be easily retrieved for the upcoming web application
9	Implement a front-end, secure solution of our web application that adheres to the decided upon wireframes and technical specifications in our Design Document

2.10 Task approach

Describe any possible methods and/or solutions for approaching the project at hand. You may want to include diagrams such as flowcharts to, block diagrams, or other types to visualize these concepts.

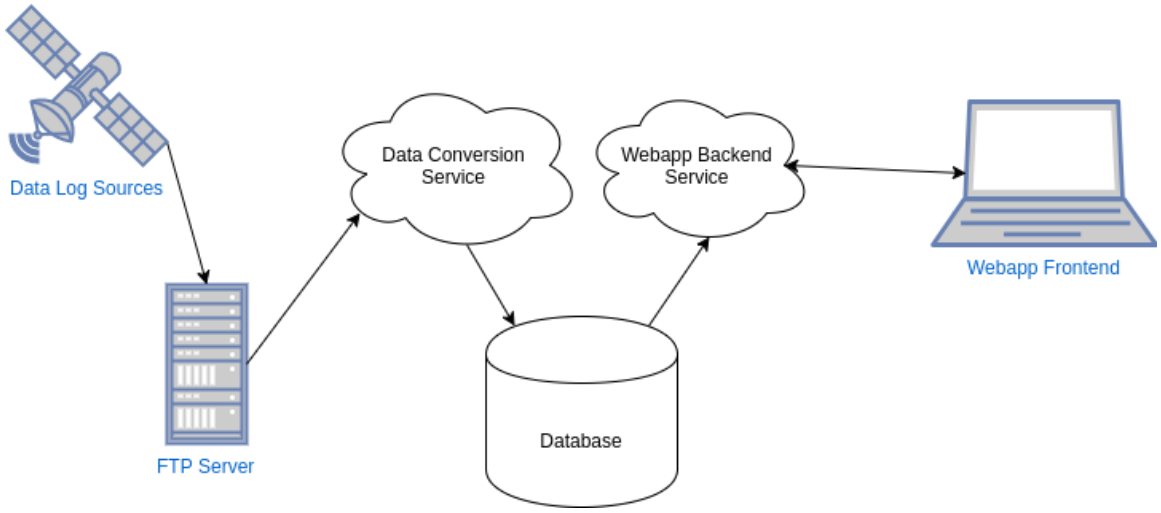


Fig. 1: Design diagram describing the proposed layout of the different parts of our project

The first 6 tasks that we have described so far detail out our plan for this semester. Figure 1 shows our design for the final product. Task 7 relates to the Data Conversion Service. Task 8 is the creation of the database and designing it so that the Webapp Backend Service can easily get the data it needs.. Task 9 relates to the Webapp Frontend and most of the Webapp Backend Service.

2.11 Expected Results and Validation

The desired outcome is a web application that Henderson Products and their clients can securely login to to see relevant data. Members of Henderson Products will want to see information like stalls and jams in their trucks and how long they took to clear or fix themselves. Their clients will be interested in pounds of spread per feet displayed on a map. Some technicians may also see this and want to be able to see histograms of engine data. All of this data is retrieved autonomously, stored in the database and fetched for the web application when needed.

We will confirm that our solutions work with rigorous testing. We have gotten information from Henderson Products on what they want and how they want it. This will aid us in our testing process displayed in Figure 2. Our tests can be grouped into 2 main types of tests: Unit Tests, Integration Tests, and User Tests.

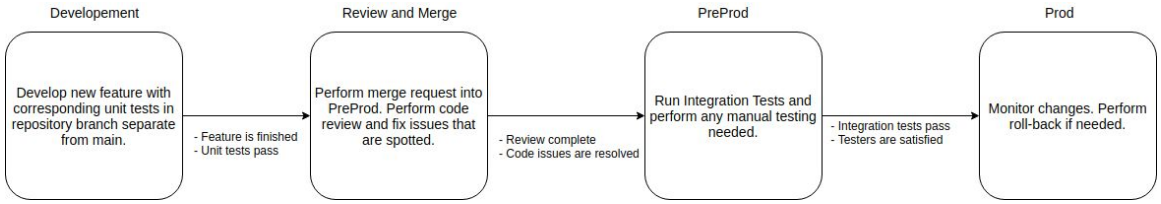


Fig. 2: Our planned test process to use for validation and superior code quality.

- **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.
- **User Tests** - Even if our web application has all the needed features and performs all the tasks, we will not consider our project a success if the application is not user focused and intuitive. We plan to have people unfamiliar with the application login to it at several different times during development. When they do we will give them tasks to complete and data to see. We will gauge how hard it is for them to complete these tasks and ask for suggestions to make it easier. We then want to implement these changes.

2.12 Challenges

We face a number of challenges in our project. Our first one is how we should handle GPS data. We've done some research towards what the best solution toward our GPS data points will be. We read a few things on low and high sampling GPS data. We're hoping that our data will be frequent enough for Google Map's API to interpret and place on roads reliably. Next is how we're going to handle queries to our database. We have a large amount of incoming data. We need to figure out an effective way to store this data so that queries are easy. We need to be able to display the data in any way possible in order our client to see how everything relates. We want to avoid writing complex code in order to handle this. We've researched Data Warehouses as a potential solution to this challenge. Lastly has to do with some of our groups background. We have a software oriented project. The computer engineer and the electrical engineer in our group each have their own respective learning curves.

3 Estimated Resources and Project Timeline

3.1 Personnel effort requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just "X" hours per week for the number of weeks that the task is active

Please See Gannt Chart for duration of tasks

Key:
0 = Minimal Effort Required
1 = Low Effort Required
2 = Medium Effort Required
3 = High Effort Required
4 = All Hands On Deck

Task	Difficulty Level
Data Conversion	3
Database Design	4
Finalize Design Documents	2
Create Database	1
Backend Development	2
Backend Testing	3
Frontend Calls Development	2
Frontend Calls Testing	3
Integration Testing	4
Frontend Design	2
Final Prototyping	1

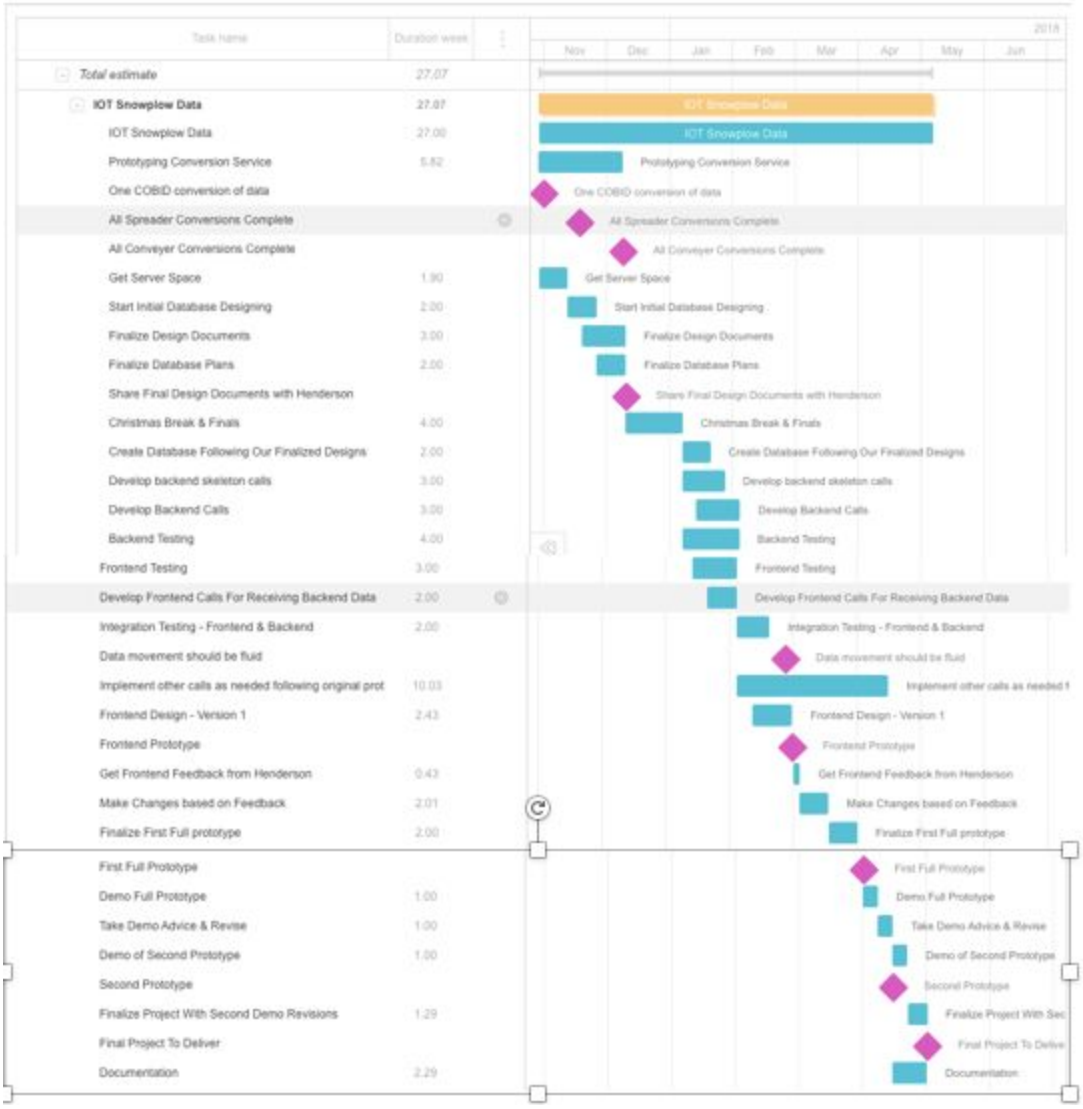
3.2 OTHER RESOURCE REQUIREMENTS

There should be no other resources required for our project. All of the software is open source and we all have our own personal computers to work with.

3.2 FINANCIAL REQUIREMENTS

Currently we do not require any financial help or resources to complete our project.

3.3 Project Timeline



4 Closure Materials

4.1 CONCLUSION

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, as well as make the information easier to read by providing charts or graphs. This information is being used to predict the performance of each individual truck and will help the clients to better monitor their own trucks. We will be formatting the information that they are receiving so that they will be able to see the quantities of each piece of information over time.

4.2 References

1. *Apache Thrift*, Apache Software Foundation, 2016, thrift.apache.org/.

2. "US Xpress Inc - Get Your Freight Shipped - U Can Depend On U.S." *USX Corporate*, www.usxpress.com/.

3. Van Rijmenam, Mark. "Trucking Company US Xpress Drives Efficiency with Big Data ." *Datafloq*, datafloq.com/read/trucking-company-xpress-drives-efficiency-big-data/513.

4.2 Appendices

We currently have nothing for our appendices.