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: 12/3/17 Version 3

Table of Contents

List of Figures

List of Tables

1 Introductory Material

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. We also discuss the environment this project will operate in, who and what this project is for and our expected end product. We also lay out a set of limitations and assumptions we have laid out for our project.

1.1 Problem Statement

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. It exists as 1 data entry per line that needs to be inputted into Excel and manipulated for around an hour before it is human readable. This still has no analytics and has to be done for each file. The Data Logger also doesn't perform any kind of data redundancy check or compression. This results in large files being sent that contain only zeros.

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.

1.2 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.

1.3 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.

1.4 Assumptions and Limitations

In order to define the scope for our project we set limitations for our project. This allows us to clearly lay out what is expected of our team from ourselves and our client. We've also defined a set of assumptions that are subject to change should a need arise.

1.4.1 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 will not use more cellular data than the initial transfer from vehicle to server.
4. No data will be lost in transformation or translation.
5. Data will be moved to the web application within 24 hours of creation.
6. Adding a new user will take less than 2 minutes.
7. Clients will only be able to view data from their own vehicles.

1.4.2 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.

1.5 Expected End Product and Other Deliverables

Our end product can be separated into 3 main parts. We will have a Data Publishing Service, Database, and Web Application.

Data Publishing Service

The Data Publishing service 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 January 2018; 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 2017, 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

In this section we define our functional and nonfunctional requirements. We also list the standards our team plans on adhering to during development. Our team lists and considers possible useful technology, safety hazards, risks and how to manage them, project milestones, how to approach these milestones, and how we can validate ourselves and our work. Lastly we reflect on challenges we've encountered so far or expect to encounter.

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 Standards Protocols

We intend to adopt ideas from the IEEE standard 1028-2008, which is a standard for software reviews and audits [4]. These ideas will help guide us when performing code reviews and Software inspections, which will be important for us to perform in order to ensure quality software is being produced. To accomplish this we intend to apply ideas from the Technical Reviews section and the Inspections Section. We also intend to apply ideas from the walk-throughs section in order to demo new features for our client.

We also plan on using parts of ISO/IEC/IEEE 29119-2-2013 Software and systems engineering --Software testing --Part 2:Test processes [5], which defines processes for developing software tests.

2.4 Technology Considerations

As said in Section 1.5 our project can be broken down into 3 parts. We considered a number of technological solutions to each of these parts and analyze them below.

- 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
        - EmberJS
            - Front end javascript framework
            - Handles routing, templating, and data-fetching
            - CLI makes it easy to initialize and configure applications
    - 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 [1]
            - 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)

A detailed analysis of the options and final selections can be found in their respective sections of the Design Documentation, section 2.3 "Design Analysis". The application will feature a manually written data parser, which will then store data into a MongoDB

database, hosted by an Amazon Web Server. The application itself will utilize Spring as the backend framework with Ember.js in the front end.

2.5 Safety Considerations

As our project is a web application, there are no physical safety considerations that need to be watched to ensure proper delivery. We plan to address data security and integrity for our project. We hope to be able to have each data logger sign its transmissions so that our Data Publishing Service can verify for certain that it was one of the client's data loggers that sent the information. We also plan on protecting our web application from cross site scripting. Our database and web server will also need input checking to make sure nothing is being injected.

2.6 Previous Work and Literature

The project itself is new, therefore no previous literature was used to construct the current project plan. However, with a new project it was important to research established products on the market that function similarly to our goals. By looking at the competition, the group could learn from their design decisions to figure out what kind of technological issues could potentially arise during our development phase.

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.

2.7 Possible Risks and Risk Management

Our team predicted possible risks that could arise during development of this project. We hope to avoid these risks by recognizing the potential for them before they arise, and adjusting ourselves as needed.

- **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.

2.8 Project Proposed Milestones

We have laid out 6 milestones for our team to progress towards while working on this project. Each of these is a needed step in our development process and should help us recognize any needs for adjustments should we fall behind.

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. **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.
3. **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.
4. **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.
5. **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.
6. **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.

2.9 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.

2.10 Objective of the Task

In Section 2.8 we laid out our project's milestones and why each one is needed. Below in Table 1 we define the objective of each milestone.

Task	Objective
1	Obtain a full understanding of the data logging system as well as modern technologies that may help us
2	Present a wireframe to Henderson Products that they accept that will be used as the front end user interface of our web application
3	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
4	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
5	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
6	Implement a front-end, secure solution of our web application that adheres to the decided upon wireframes and technical specifications in our Design Document

Table 10: Milestone Objectives

2.11 Task Approach

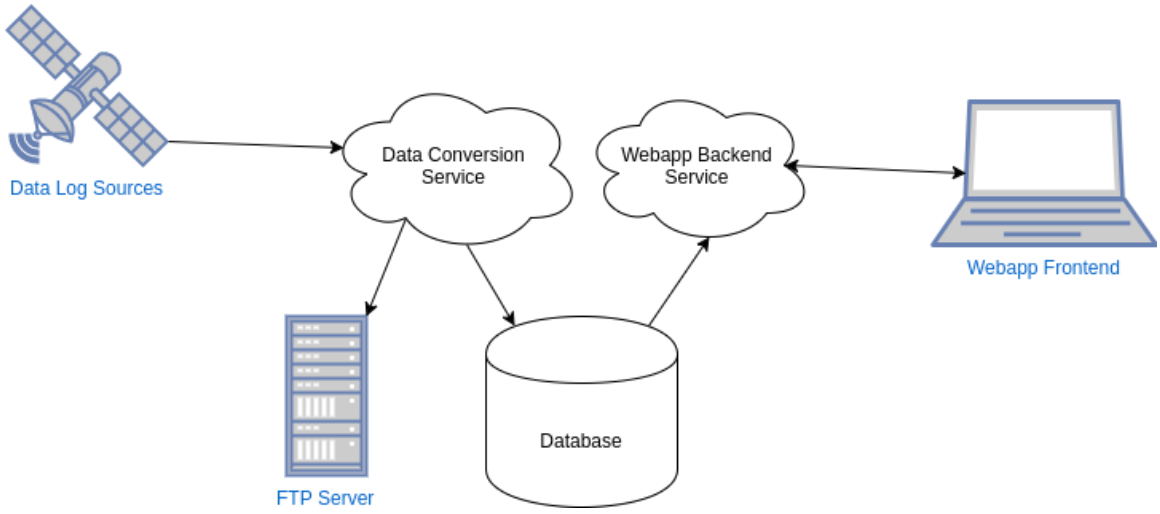

Fig. 1: Design diagram describing the proposed layout of the different parts of our project

The first 3 tasks that we have described so far detail out our plan for this semester. Figure 1 shows our design for the final product. Task 4 relates to the Data Conversion Service. Task 5 is the creation of the database and designing it so that the Webapp Backend Service can easily get the data it needs.. Task 6 relates to the Webapp Frontend and most of the Webapp Backend Service.

2.12 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 3 main types of tests: Unit Tests, Integration Tests, and User Tests.

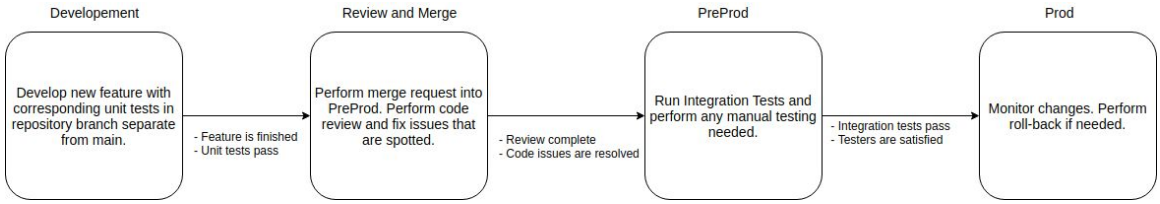

Fig. 2: Our planned test process to use for validation and superior code quality.

- **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.

These tests will be performed in order to fully test both our functional and nonfunctional requirements. We plan to test these as detailed below:

Functional Testing

Data Conversion Testing - To ensure that our data is being converted correctly we plan on writing unit tests that utilize a number of different unit tests.
Data Storage Testing - We plan on creating integration tests to ensure that our connection to the database doesn't break as we make changes. We will also create some unit tests that mock the database connection and ensure it is being called correctly.
Data Access Testing - It will be important for us to test our webapp backend to ensure that all the endpoints are functional. This will involve some unit tests for some of the operations being performed and integration / manual testing to ensure that the service endpoints can be reached.
Data Analysis Testing - Our frontend will involve a lot of manual testing during development to ensure everything is being displayed correctly. We also create unit tests to ensure operations are being performed correctly on the data.

Nonfunctional Testing

Security Testing - To ensure access to our system is restricted to Henderson and their clients, we plan on performing manual testing of our endpoints to ensure that they are locked down to users with correct credentials.
Responsive Requirements Testing - We plan on setting up logging / metrics for our backend service to help us determine bottlenecks and areas where we need to improve to meet our performance requirements.

2.13 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

With our technology, standards, risks, requirements, and milestones all defined we can now estimate what personal effort, resources, and finances we need to complete the project. We can also plot out the work that needs to be done.

3.1 Personnel Effort Requirements

The 3 parts of our end product can be broken up further into a number of tasks that need to be completed as shown in Table 3. Table 2 provides a key for how much effort we expect each task to require.

Please See Figure 4 in Section 5 Appendices 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

Table 2: Key for detailing personal effort requirements

Task	Difficulty Level
Data Conversion	3
Database Design	4
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

Table 3: Difficulty assignment of tasks needed to complete this project

3.2 Other Resource Requirements

For our project we need software, computers, and server space. All of the software is open source and we all have our own personal computers to work with. The server space needed for our Data Publishing Service, Data Storage solution, and Web application is to be obtained through Amazon Web Services (AWS). We plan on scaling them up if the need arises.

3.3 Financial Requirements

The financial requirements for this project are based on available resources (resources that are already being allocated to the project) and proposed resources.

Available Resources

Resource	Cost
Human Data Conversion	$75 / hour of labor
Mobile Data	Unknown

Table 4: Current resources available to the project

The main goal of our project is to remove the need for Henderson Employees to have to manually convert the data. As shown in Table 4, it costs Henderson Products seventy-five dollars per hour of labor spent converting data from the trucks. On average, two hours of work will only result in a few minutes of converted data. The mobile data costs will remain

Proposed Resources

Resource	Cost
AWS Server for Conversion Service	$45 / month
AWS Server for Database	$45 / month

AWS Server for Web Service	$45 / month

Table 5: Proposed resources for the project

As shown by table 5, the additional resources that are required for our projects are three servers that will be purchased from AWS. In total, these will cost 135 dollars per month. These costs will be in addition to any existing costs (such as mobile data costs).

3.5 Project Timeline

Our project timeline has changed a bit over the semester, and that can be observed by comparing figures 3 and 4 in the appendices. Figure 3 was our initial gantt chart depicting our schedule for the project from around the middle of this semester up until we deliver our product to Henderson. Figure 4 is our finalized gantt chart and that is the figure we wish to talk about in regards to when we plan to have certain deliverables available.

As shown from figure 4, we plan to complete eight major deliverables. They are as follows:

1. Complete Data Conversion Service 01/31/2018
2. Finish First Version of Database 02/15/2018
3. Finish Final Version of Database 03/01/2018
4. Finish First Version of Backend 03/09/2018
5. Finish First Version of Frontend 04/01/2018
6. Deliver First Full Prototype to Henderson 04/06/2018
7. Deliver Second Full Prototype to Henderson 04/20/2018
8. Final Project To Deliver 04/27/2018

The time between these deliverables on the gantt chart is filled with tasks and these tasks are separated into the following groups:

1. Data Conversion Service
2. Database Creation
3. Backend Development
4. Frontend Development
5. Prototyping & Delivering

Each of these sections are typically broken down into development, testing, and refactoring sections. We will allow some time at the beginning of next semester to create more specific tasks designed for individuals to complete and these will be mainly defined as issues in our gitlab.

4 Closure Materials

In this section we review and conclude on our plan for this project. We also acknowledge and thank people that have helped us along the way.

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. Our design consists of 3 main parts that can be further broken down into 5 design aspects. These are the data loggers, data ingestion, database, webapp backend, and webapp frontend. During the next semester we will work to implement all 3 parts of using the design decisions we made in these 5 aspects.

4.2 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.

5 Appendices

Currently we already have 1 appendix, but we are open to adding more should the need arise.

5.1 Projected timelines as Gantt Charts

Our initial timeline was a little off and not in line with the goals we had created. It spanned throughout both semesters but focussed on the wrong things. It is displayed in Figure 3. Figure 4 contains our final version of our projected timeline as detailed in Section 3.5.

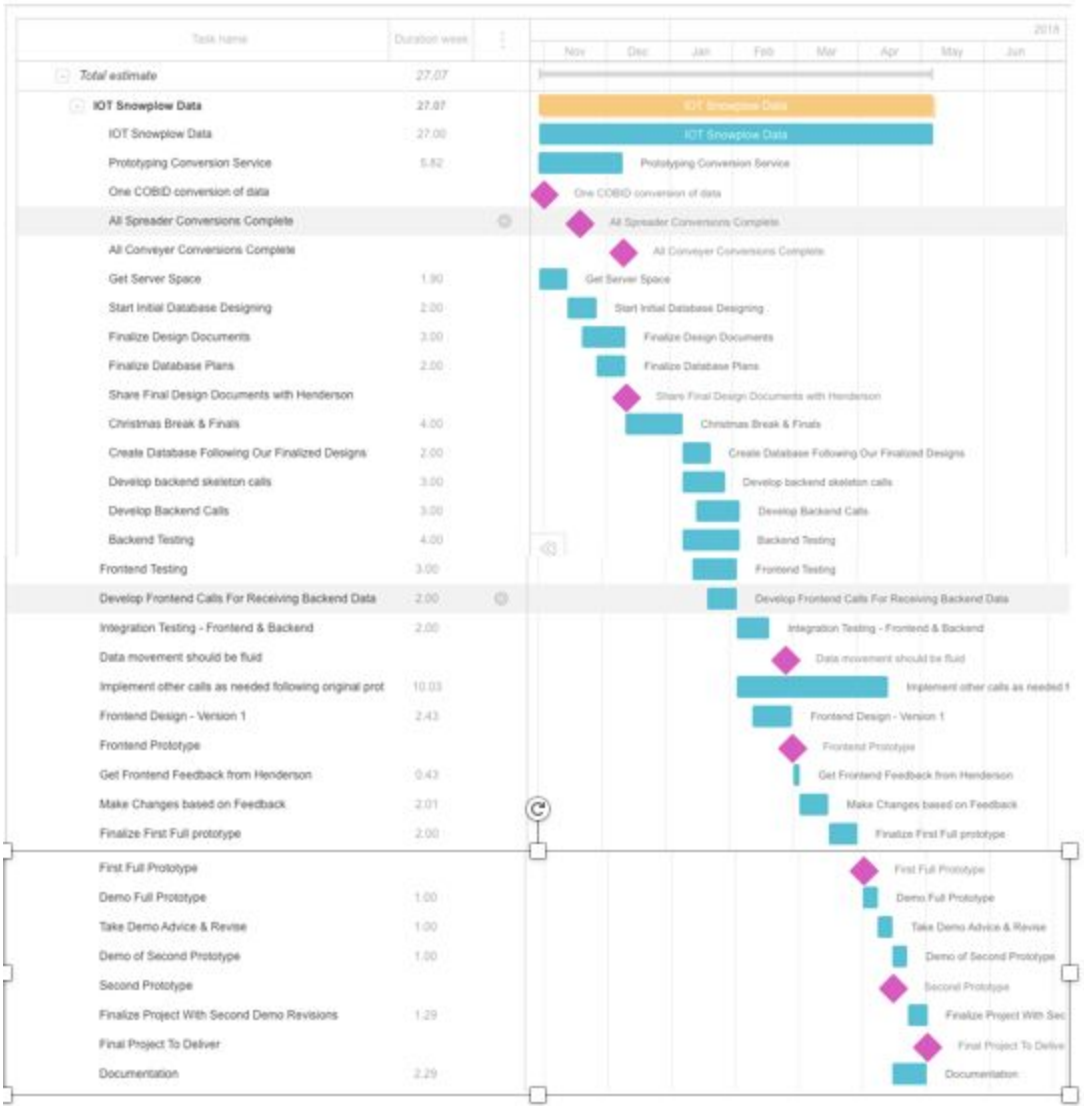

Fig. 3: Gantt Chart Timeline Version 1

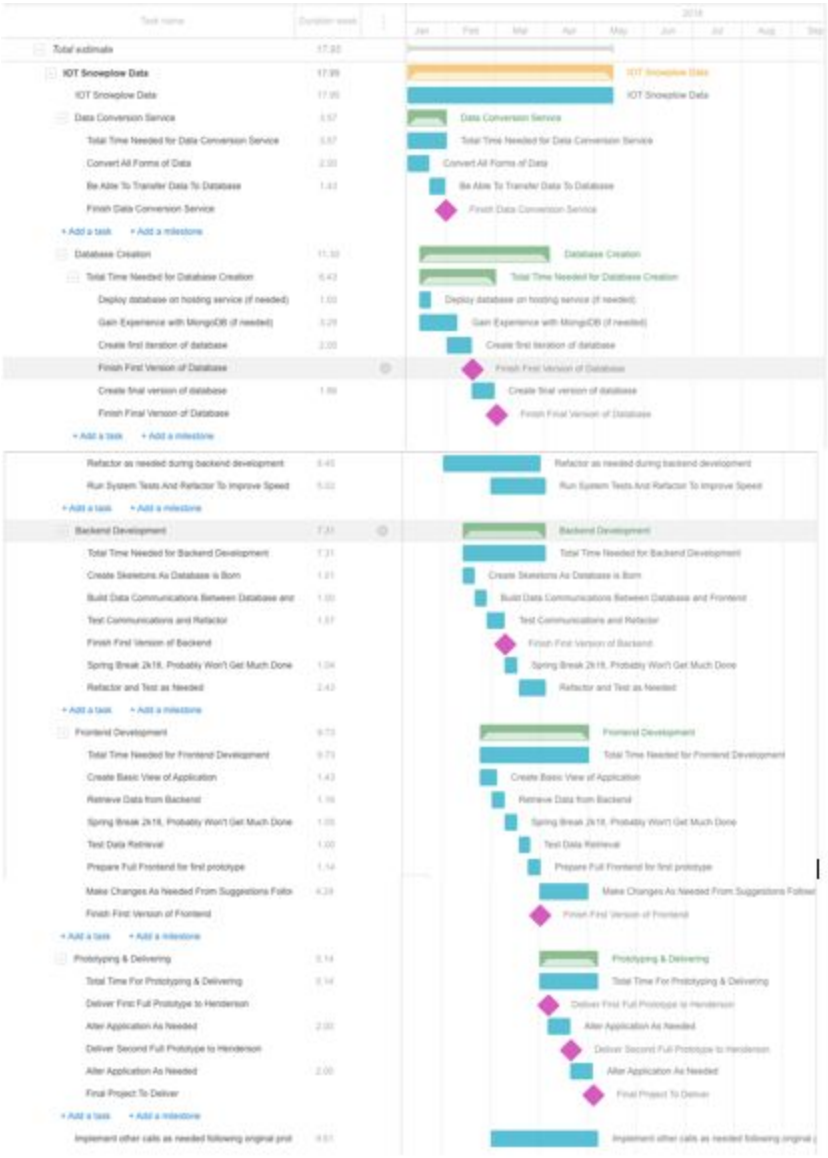

Fig. 4: Gantt Chart Timeline Version 2

6 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. *IEEE Standard for Software Reviews and Audits*, IEEE Standard 1028, 2008.

5. ISO/IEC/IEEE International Standard - Software and systems engineering --Software testing --Part 2:Test processes, IEEE Standard 29119-2, 2013