Computer Science Senior Capstone
CS476 - Requirements Engineering

ViralTech

Project: PIMpoint Collaborator Data Entry Point

Requirements Document V1.3

Overview:

The purpose of this document is to highlight the high-level requirements of our project by multiple iterative design cycles and requirements gathering with TGen North.

Team Members:
Jialei Chen
Carl Porter
Colton Spector
Weiheng Su
Scooter Nowak - Capstone Mentor

Client:
Jonathan Todd - Associate Bioinformatician

Northern Arizona University
School of Informatics, Computing, and Cyber Systems

TGen North
Flagstaff, Arizona

December 13th, 2019

Team Lead Signature

Client Signature

Table Of Contents

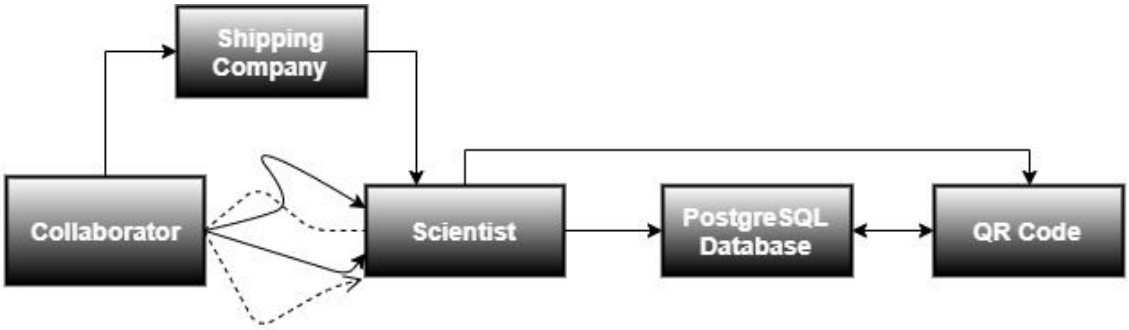
1.0 Introduction

Public health is an ever shifting target. Bacteria, fungi, viruses, and other pathogens are constantly growing, evolving and spreading across the globe. The constant outbreak of germs has become a global problem, and countless researchers and scientists are working on treatments and solutions. It becomes a very important task to sort and sequence the genetics of these pathogens.

The Translational Genomics Research Institute (TGen) is a non-profit 501(c)(3) organization focused on developing earlier diagnostics and smarter treatments. TGen North is the Pathogen and Microbiome division of TGen. TGen North is focused on bacterial and fungal pathogens associated with medicine, public health, biodefense, outbreak detection, and source tracing, developing clinical diagnostic tests and characterizing antibiotic resistance. TGen North works with dozens of partners around the world to address a variety of public health issues. This includes DNA sequencing and analysis of various sample types, patient samples from clinical partners, as well as environmental samples such as bacteria and fungi related to human health. TGen North developed a database and tracking system called PIMpoint in order to collect large amounts of crucial data relating to the pathogens they sequence. PIMpoint is a set of web based interfaces that allow users to interact with the Pimpoint database in a variety of different ways. Data Entry, Data Querying ,Tracking the location of samples, Moving samples, Labeling samples, etc. Different shareholders can use different web interfaces to interact with the database depending on their needs and permissions. Team Viraltech has been tasked with developing the web interface that will allow TGen Norths collaborators to interface with the database.

2.0 Problem Statement

The following figure shows the current workflow for TGen North. The process for collaborators submitting data involves them sending the data in various formats directly to the scientists at TGen North. Once received by TGen North, scientists will then input the respective data into the PIMpoint system. This process requires scientists to spend a lot of their valuable time doing manual data entry instead of science.



Currently, TGen Norths diverse set of collaborators are using equally diverse methods for data sharing. For example, mailed hardcopy print-outs, email, Microsoft Excel documents, and others. Because of this lack of standardization, too much time is wasted as TGen Norths scientists try to decipher and correct the data received. Currently this process is problematic for these main reasons:

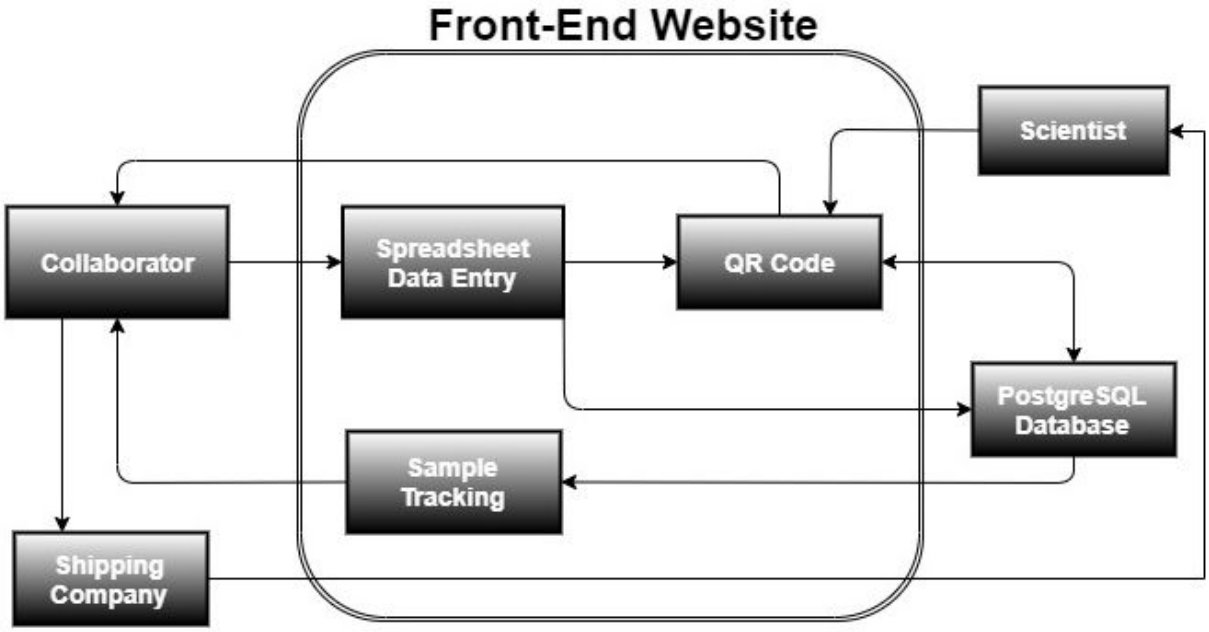
- Because data entry is non-standardized, scientists often have to track down or clarify necessary data that wasn't included properly in the submission
- Scientists waste time every day manually entering data into their database.
- Collaborators are unable to track the progress of their samples once they've sent them to TGen North.

To combat this problem our team is designing a configurable web platform to collect data from contributors in a standardized format, then automatically upload that data to the database. The platform will also produce a printable barcode that the contributor can attach to their samples so that the scientists at TGen North are able to quickly scan the sample and pull up all the information that they need.

3.0 Solution Vision

Collaborators will enter our website and be able to input their data into a spreadsheet. Our website will correctly generate the spreadsheet so that the data is formatted according to TGen North's specifications. Then our system will produce the QR codes, and insert the data into the database. Collaborators are prompted to print a PDF version of the QR codes after submitting data for the samples. This PDF version of QR codes is able to fit easily onto test tubes. Once

printed, the collaborators can attach the QR codes to their samples and send the labeled samples to TGen through a shipping company. The shipping company will then deliver the package of samples to the scientists at TGen North. Scientists will be able to scan the QR codes to quickly and easily pull up all the information they need from the database. After sending the samples to the scientists, Collaborators will be able to view sample progress through the various processes offered by TGen North. This allows Collaborators to actively engage with the system even after they've finished sending the samples.



The key features of our solution are as follows:

- Our system ensures standardized data entry.
- Our system automatically inserts sample data collected from collaborators into the database.
- Our system allows collaborators to see their sample progress.

4.0 Project Requirements

In this section we outline in detail each of the functional, performance, and environmental requirements. We will be referencing the various shareholders who will be interacting with the workflow at large and so before we start we will define those shareholders:

Client: The client is the shareholder who will be deploying the system. The client for this project is TGen North, however we will also be making this system configurable by any client to fit their specific business needs. The client's interaction with the system will be in modifying configuration files that our system will read to generate the website.

User: The user is the shareholder who will be directly using and interacting with the website. The users for this project are the various collaborators who work with TGen North. The user's interaction with the system will be for data submission, sample tracking, QR code receiving and error communication.

Scientist: The scientist has no direct interaction with our system. The scientists for this project are the scientists at TGen North. The scientists use different web interfaces to interact with the database. However this system is designed to facilitate communication between the users and the scientists via the database so it's important to define their role in the overall workflow.

4.1 Functional Requirements

Functional requirements specify the features our website has. Our functional requirements state what our system should be able to do, not how it should do it.

4.1.1 Front-End Website

The Front-End of our website is the general look and feel of the website. This includes the colors, images and general flow.

- **[4.1.1.1]**Client can configure front-end website appearance
    - **[4.1.1.1.1]** Client can configure the colors of front-end website
    - **[4.1.1.1.2]** Client can configure the source file location of all images on front-end website
- **[4.1.1.2]** Client can configure what database the website queries from/inserts into

4.1.2 Data Entry

The data entry section of our website is where users can input the data associated with the samples they are shipping to TGen North. Users often send hundreds of samples at a time so our data entry needs to be able to conveniently accommodate scalable entry.

- **[4.1.2.1]** Users can enter data in a spreadsheet format
- **[4.1.2.2]** Users can click “Create New Package” to add unlimited “Unsent Packages” to their list of packages
    - **[4.1.2.2.1]** Users can view spreadsheet data entry by clicking the “unsent package” item on their list
    - **[4.1.2.2.2]** Users can still view all the data entered into the spreadsheet after leaving the website and coming back before submitting the package
    - **[4.1.2.2.3]** User can click “Send Package Data” to insert all of the data into the database
    - **[4.1.2.2.4]** Website will reject data that isn’t formatted according to the Client configured formatting
    - **[4.1.2.2.5]** Users can see message telling them if their data was entered successfully
    - **[4.1.2.2.6]** Users can see message detailing what was wrong with their data entry if it wasn’t successfully entered
    - **[4.1.2.2.7]** Users can freely add optional fields to their data if the user has the necessary permissions
- **[4.1.2.3]** Client can configure spreadsheet for each individual user
    - **[4.1.2.3.1]** Client can configure spreadsheet to be expandable with optional fields for a given user
    - **[4.1.2.3.2]** Client can configure the types of data that can be entered in each column/attribute
    - **[4.1.2.3.3]** Client can configure the header name of each column/attribute
    - **[4.1.2.3.4]** Client can configure the database destination of each column/attribute
- **[4.1.2.4]** Users can copy and paste entire spreadsheets into the data entry from .xsv files

- **[4.1.2.5]** Users receive email instructions on how to ship samples once package data has been inserted into the database
- **[4.1.2.6]** (Stretch Goal) Users can upload .xsv files that are correctly formatted directly to the website
- **[4.1.2.7]** (Stretch Goal) Website rejects .xsv files that do not meet the configuration requirements set by client
- **[4.1.2.8]** (Stretch Goal) Website parses and inserts correctly formatted .xsv files into the database

4.1.3 Sample Tracking

The current workflow at TGen North doesn't involve the users after they've sent in their samples, until the results are given back to them. Our website will allow users to engage in the process by being able to track where their samples are at any given time.

- **[4.1.3.1]** User can view all of their sample progress through the process
- **[4.1.3.2]** Client can configure the process for each individual User, for each individual package
    - **[4.1.3.2.1]** Client can configure the name of the process the samples are undergoing
    - **[4.1.3.2.2]** Client can configure the amount of steps in the process
    - **[4.1.3.2.3]** Client can configure the name of each step in the process
    - **[4.1.3.2.4]** Client can add a description for each step in the process
- **[4.1.3.3]** User can see samples at each step aggregated into a bar chart
    - **[4.1.3.3.1]** User can see the amount of samples in each aggregation by hovering over the bar for that aggregation in the bar chart
    - **[4.1.3.3.2]** Each bar in the bar chart is sized according to the percentage of samples at that step in the process
    - **[4.1.3.3.3]** Bar chart expands horizontally away from the label of each step defined by the Client

4.1.4 QR Code Generation

After the user submits their data they will be prompted to print out a .pdf version of all the necessary QR codes.

- **[4.1.4.1]** User can print QR codes without having to manually download them
- **[4.1.4.2]** User can print QR codes that are able to fit easily onto test tubes
    - **[4.1.4.2.1]** QR printouts have the parent number on them
    - **[4.1.4.2.2]** QR printouts have the box number on them
    - **[4.1.4.2.3]** QR printouts have cell number on them
    - **[4.1.4.2.4]** QR printouts have directional cap indicator on them
    - **[4.1.4.2.5]** QR printouts have TGen number on them
    - **[4.1.4.2.6]** QR printouts have category type printed on them
- **[4.1.4.3]** User can print .pdf document of all QR codes for their package
- **[4.1.4.4]** User is prompted to print .pdf version of QR codes after clicking “Send Package Data” button
- **[4.1.4.5]** User receives .pdf version of QR codes as an email for redundancy

4.1.5 Error Communication

When errors occur with a sample or with the users view of the website it is important that our website enables efficient communication between all parties involved. Our website will enable the scientists to quickly and effectively communicate necessary information to the users whenever an irregularity occurs in the process. In addition, our website will enable users to quickly and effectively communicate necessary information to the scientists whenever they experience an irregularity on the user side.

- **[4.1.5.1]** Scientists are able to communicate errors with samples to users
    - **[4.1.5.1.1]** Users can see error symbol next to package in list of packages

        - **[4.1.5.1.2]** Users can click expansion button next to package with error symbol to see drop down list of errors indented under package
        - **[4.1.5.1.3]** Users can click the error message to see a window displaying the error information
        - **[4.1.5.1.4]** Users can see the identification number of affected samples in error window or else the keyword “All” indicating all samples are affected
        - **[4.1.5.1.5]** Users can see error message in error window
        - **[4.1.5.1.6]** Users can see error resolution message in error window
        - **[4.1.5.1.7]** Client can configure pre-determined error message/resolution profiles
    - **[4.1.5.2]** Users can send help messages to client by clicking “Help” button
        - **[4.1.5.2.1]** Help messages are transmitted via email to client configured email address
        - **[4.1.5.2.2]** Help messages include a saved state of the website at the time of message sending for client to see

4.2 Performance Requirements

Outlined below are the performance requirements for our website. While the functional requirements outlined what our website needs to be able to do, the performance section outlines the metrics by which it needs to perform.

4.2.1 Front-End Website

The Front-End of our website is the general look and feel of the website. This includes the colors, images and general flow.

- **[4.2.1.1]** User can load website**[4.1.1]** in less than 5 seconds

4.2.2 Data Entry

The data entry section of our website is where users can input the data associated with the samples they are shipping to TGen North. Users often send hundreds of

samples at a time so our data entry needs to be able to handle large database insertions at a reasonable speed.

- **[4.2.2.1]** User can see configured spreadsheet**[4.1.2.1]** for data entry in less than 2 seconds after creating a new package
- **[4.2.2.2]** User can understand where to place sample data**[4.1.2.1]** in less than 5 seconds 8 out of 10 times after viewing spreadsheet for the first time
- **[4.2.2.3]** User can fix data entry in less than 10 seconds after reading unsuccessful data entry message**[4.1.2.2.6]** 8 out of 10 times

4.2.3 Sample Tracking

The current workflow at TGen North doesn't involve the users after they've sent in their samples, until the results are given back to them. Our website will allow users to engage in the process by being able to track where their samples are at any given time.

- **[4.2.3.1]** User can find sample tracking**[4.1.3.1]** in less than 5 seconds after logging into the website for the first time 8 out of 10 times
- **[4.2.3.2]** User can see sample tracking**[4.1.3.1]** in less than 2 seconds after clicking the package
- **[4.2.3.3]** User can find where they can view the amount of samples**[4.1.3.3.1]** at each step 10 seconds after sample aggregation bars are loaded 8 out of 10 times

4.2.4 QR Code Generation

After the user submits their data they will be prompted to print out a .pdf version of all the necessary QR codes.

- **[4.2.4.1]** User is prompted to print .pdf version of QR**[4.1.4.4]** codes in less than 2 seconds after submitting package data
- **[4.2.4.2]** Scientists are able to scan QR codes**[4.1.4.2]** on test tubes in less than 2 seconds after pressing scanner trigger 8 times out of 10 after 10 minutes practicing

4.2.5 Error Communication

When errors occur with a sample or with the users view of the website it is important that our website enables efficient communication between all parties involved. Our website will enable the scientists to quickly and effectively communicate necessary information to the users whenever an irregularity occurs in the process. In addition, our website will enable users to quickly and effectively communicate necessary information to the scientists whenever they experience an irregularity on the user side.

- **[4.2.5.1]** User notices error with package**[4.1.5.1.1]** in less than 5 seconds after logging into the website 8 out of 10 times
- **[4.2.5.2]** User can find error message**[4.1.5.1.3]** in less than 3 seconds after noticing error 8 out of 10 times
- **[4.2.5.3]** Client receives help message**[4.1.5.2.1]** from user in less than 1 hour after the message is sent

4.3 Environmental Requirements

The environmental requirements are requirements dictated by the pre-existing solutions and client needs that our system has to meet. Our project however is highly modular inside of TGen North's existing system. TGen North envisions a solution with different portals for different types of users, meaning that as long as all the portals access the same database the portals themselves can be vastly different. Having said that our project still has several environmental requirements that we must meet.

- **[4.3.1]** Users can login and sign up for the website**[4.1.1]** using Google Authentication
    - **[4.3.1.1]** Website checks authentication tokens every time database queries**[4.1.3.3]** or insertions**[4.1.2.2.3]** are to be made, to ensure token is still valid.
- **[4.3.2]** Website must be able to interface with PostgreSQL**[4.1.2.2.3]**
- **[4.3.3]** User sees spreadsheet similar to "Google Sheets" for data entry**[4.1.2.1]**

5.0 Potential Risks

TGen North solves health problems globally, and when a pathogen outbreak occurs the time it takes to determine the source could mean the difference between a patient living or dying. Because of the nature of our client's work, even a low severity time wasting risk can have deadly consequences. Therefore even the lowest severity risks in our system must be taken very seriously. Our website will handle data transactions between the collaborators and the database. Because of this there are several database related risks we face.

5.1 Duplicate Data

When a collaborator submits data to the website it inserts it into the database. There is a risk that data is duplicated, either by our website or by the collaborator accidentally submitting the same data twice. Because of the nature of DBMS's all records must have a unique key identifying the data, however this does not preclude the system from inserting the same data twice under two separate keys. The QR codes are linked to the unique keys for the data record so they would still pull up one sample, but we could still have two QR codes pointing to different data records for the same sample. This would cause a couple of issues:

1. Scientists could have one package but two sets of data. Without looking deeper into the records they could assume that there is still another package that has been lost in shipment
2. Database space is wasted on duplicated records

The overall severity of this risk isn't very high, mostly causing clerical errors and wasting some time. In order to mitigate this risk we can do a database query on the sample data to see if we pull any records with the exact same data, and then not insert the data and kick back an error to the collaborator.

5.2 Database Read Security

Our system will be able to query the database to pull all collaborator sample records and organize them in our tracking system. This opens up a potential security risk of malicious persons stealing collaborator sample information off the database. Theft of information could cause several issues:

1. Collaborators could lose faith in TGen, damaging their professional relationship
2. TGen could be fined for HIPPA violations depending on the nature of the stolen information

The overall severity of this risk is very high, potentially causing the loss of business or expensive fines. In order to mitigate this risk it's important for our backend to require valid authentication tokens before making any database queries.

5.3 Database Write Security

Our system will be able to insert data into the database to submit collaborator sample data. This opens up a potential security risk of malicious persons submitting mass amounts of fake information. This could cause the database to be filled with useless information requiring deletion of records. The severity of this risk isn't very high, mostly wasting time. In order to mitigate this risk it's important for our backend to require valid authentication tokens before making any database insertions.

5.4 SQL Injection

Our system will be taking user input as strings and appending them to SQL queries sent to the database. This kind of operation creates a situation where SQL injection attacks become possible. SQL injection attacks are a very common security vulnerability with websites. They involve putting SQL queries into text fields for the back end of a website to append to a hard coded SQL query. Because our website will only have query and insertion privileges SQL injections will not be able to modify any existing data or change the database structures. However even with query and insertion privileges only, SQL injections could cause several issues:

1. SQL injections could allow malicious users to view the records of other users on the website
2. SQL injections could allow malicious users to view other information on the database that our backend doesn't query
3. SQL injections could insert records into the database

The severity of this risk is very high, as it has the same consequences as the previously defined risks as well as potentially leaking other database information. In order to

mitigate this risk we need to check all incoming string data for common SQL syntax before appending it to our SQL commands.

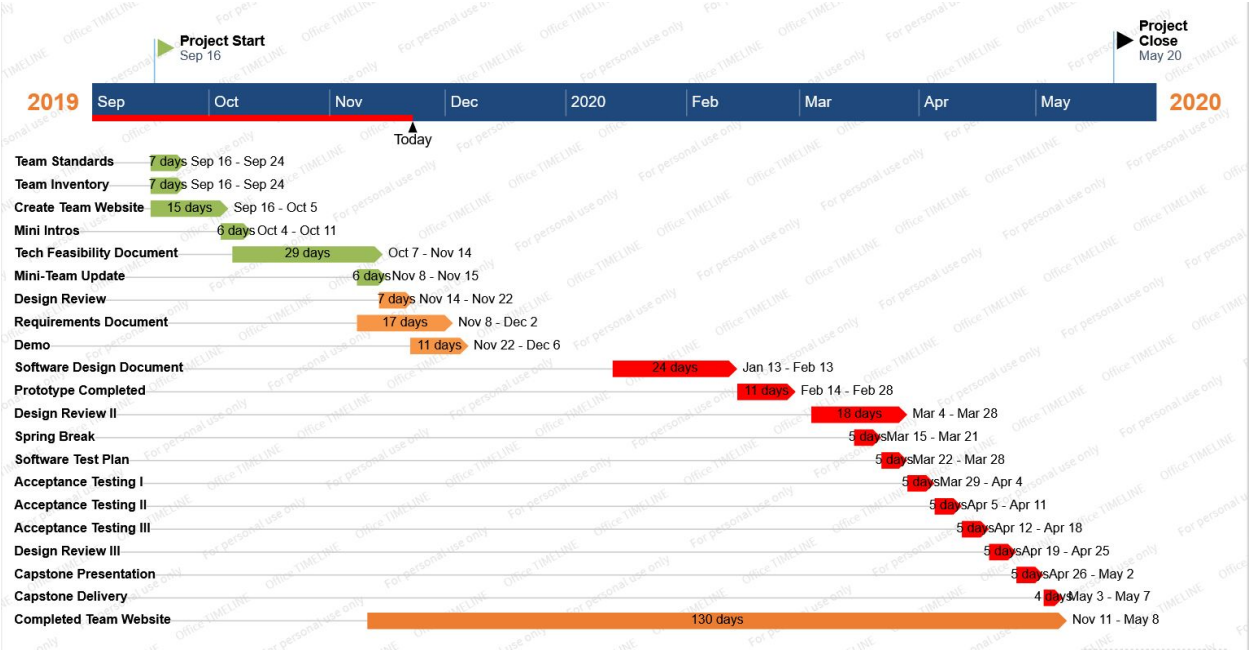
Risk	Severity	Mitigation Plan
Duplicate Data	Medium	Query database for duplicate record before insertion
Unauthorized Database Read	Very-High	Ensure valid authentication token before read
Unauthorized Database Write	Medium	Ensure valid authentication token before write
SQL Injection	Very-High	Search all incoming string data for SQL syntax before appending to SQL queries

6.0 Project Plan

Since September we have been in the process of gathering requirements for our project. We met with TGen North bi-weekly. Our meetings started with general information as well as a tour of the TGen North facilities. From there we began iterative conceptual prototyping in order to workshop the functionality of the website. We went through three iterative cycles, with the first prototype being drawn on paper and the next two modeled in photoshop. Once we had a pretty good idea of how the website would look we translated the visual prototypes to technical requirements. We went over the requirements line by line with TGen North and discussed changes and additions to be made.

Outside of our meetings with TGen North we created team standards and team inventory documents (Image 1, Sep 24th). We created our team website (Image 1, Oct 5th). We wrote a technological feasibility document (Image 1, Nov 11th). We did our design review, presenting our projects progress (Image 1, Nov 22nd). And we completed our tech demo where we implemented software locally to get data from a PostgreSQL server, to the back-end, to the javascript, to the html and back again (Image 1, Dec 9th).

Image 1: Gantz chart of our schedule as of November 20th. Green indicates completed items, orange in progress and red upcoming.



Going into our second semester we are confident that we can bring this project to completion. We plan on hopefully increasing our meetings with TGen north to be weekly instead of bi-weekly. Our second semester is focused on the implementation part of the development process. Our first goal in the second semester is to produce a software design document (Image 1, Feb 13th) that goes into the technical design and implementation of our website. Our design document will involve UML diagrams, explicit data flow scenarios, function documentation for our back-end, etc. While we're completing the design document our plan is to start an iterative implementation cycle. We plan on creating the basic functionality of the front-end and back-end in our first iteration, and then linking them together in our second. We plan on completing our prototype by Feb 20th.

7.0 Conclusion

TGen North, the Pathogen and Microbiome division of TGen, is focused on bacterial and fungal pathogens associated with medicine, public health, biodefense, outbreak detection, and source tracing. They are developing clinical diagnostic tests and characterizing antibiotic resistance.

TGen North works with dozens of partners around the world to address a variety of public health issues. With their current workflow, scientists at TGen North are wasting valuable time every day doing manual data entry. In order to redirect that time away from data-entry and towards life saving science, ViralTech will create a web platform to accomplish this task for them. Our system will allow collaborators to directly enter their sample data into the website, as well as track the progress of those samples. So far we have completed most of our requirements gathering phase. Although requirements are often a shifting target and evolve through the implementation process, we have a solid base to work with as we move forward. In addition to our requirements gathering we have been working with the various technologies and interfaces we will be developing with. We have completed our technological demo and have been able to demonstrate the key connections necessary for our final system. ViralTech is certain that by the end of our prototyping process next semester we will have a functional web application that TGen North can implement into their current system. We are certain that our prototype will improve the overall workflow at TGen North.