GHOST: Global Hepatitis Outbreak and Surveillance Technology Detailed Design Report

Team 23:

Andrew Amontree
Jeongsoo Kim
Ernest Lai
Sarthak Mohapatra
Lovissa Winyoto
Georgia Institute of Technology

In partnership with:

David S. Campo, Ph.D.
Centers for Disease Control (CDC)

Table of Contents

List of Figures

Abstract

The Division of Viral Hepatitis at the Centers for Disease Control (CDC) has created a system, Global Hepatitis Outbreak and Surveillance Technology (GHOST), in which public health researchers can perform individual analyses based on collected and existing patient data in order to more easily track outbreaks across the US. GHOST accomplishes two goals: first, it acts as the front-end for a database of previously sequenced patient genomes. Second, once a researcher navigates to a specific genomic analysis, GHOST allows visualization of the data in the form of a node-link diagram of patient-to-patient relationships, where nodes are patients and links represent the strength of their genomic relationship. The purpose of this report is to identify and define the components that comprise GHOST and outline the extent of our intended modifications to the current system.

Introduction

Hepatitis C is a contagious liver disease that ranges in severity from a mild illness lasting a few weeks to a serious, chronic illness that attacks the function of the liver. It results from infection with the Hepatitis C virus, which is spread primarily through contact with the blood of an infected person[1]. It is a disease that affects millions of people, often without their knowledge.

The current system in place requires the forwarding of data to the CDC and is a process that can potentially take months. The steps required to determine the similarity in strains of Hepatitis between two cases make pinpointing the source in the event of an outbreak much slower and harder. The GHOST system eliminates the need to send in test results and have the CDC perform analysis, instead having a practitioner input gathered data and receive feedback directly from the system. We will be working on redesigning and refactoring UI elements in the front end of the GHOST system in order to create a more simple and intuitive user experience, with a tentative release schedule and designs detailed below.

[1] http://www.cdc.gov/hepatitis/hcv/cfaq.htm

Background

In the prior semester, we have worked with Dr. David S. Campo, Ph.D and his team from the Centers for Disease Control to isolate the main issues with the current website and visualization, and envision the design changes necessary for a better user interface and resulting experience. Through meetings, both at Georgia Tech and the CDC's office, we have decided to focus our efforts on redesigning the dashboard, the landing page, and the node-link visualization components of the website. Over the course of a semester, we have produced a statement of work, a feasibility report, and an oral presentation for this project. Finally, we have planned the project implementation sprints that we will use this semester.

Systems and Architecture

Static System Model (Static System Architecture)

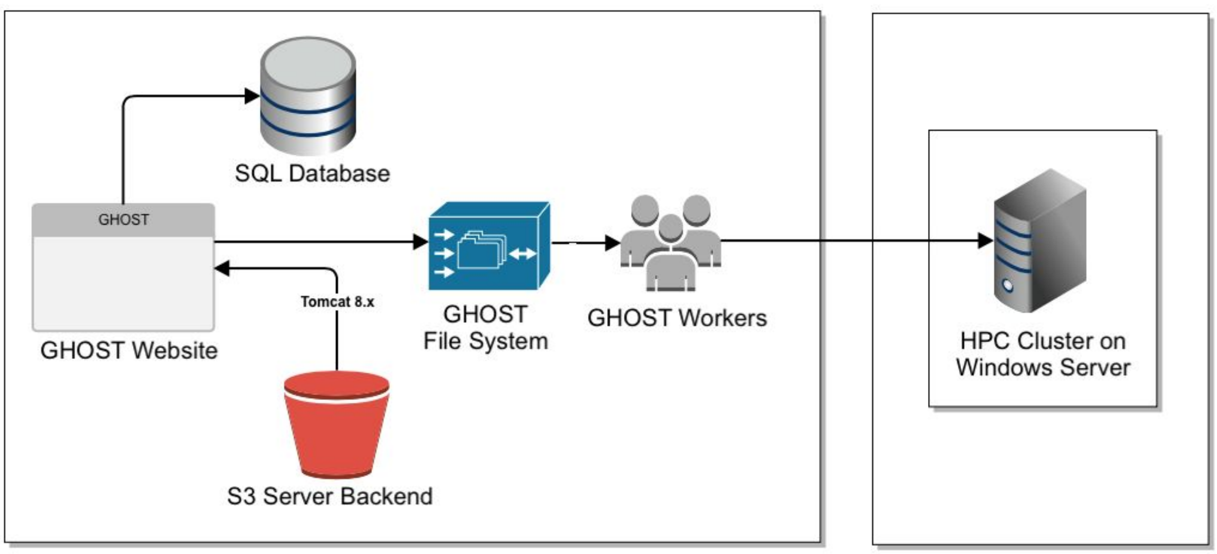

Figure 1. GHOST Static System Architecture Diagram.

The diagram in Figure 1 shows the high level architecture of the existing GHOST project and describes the interaction between the frontend and backend components. Although we will be working exclusively with the website portion of the front end, we have included a diagram in Figure 1 to demonstrate the makeup of the larger GHOST system.

User Scenario (Dynamic System Architecture)

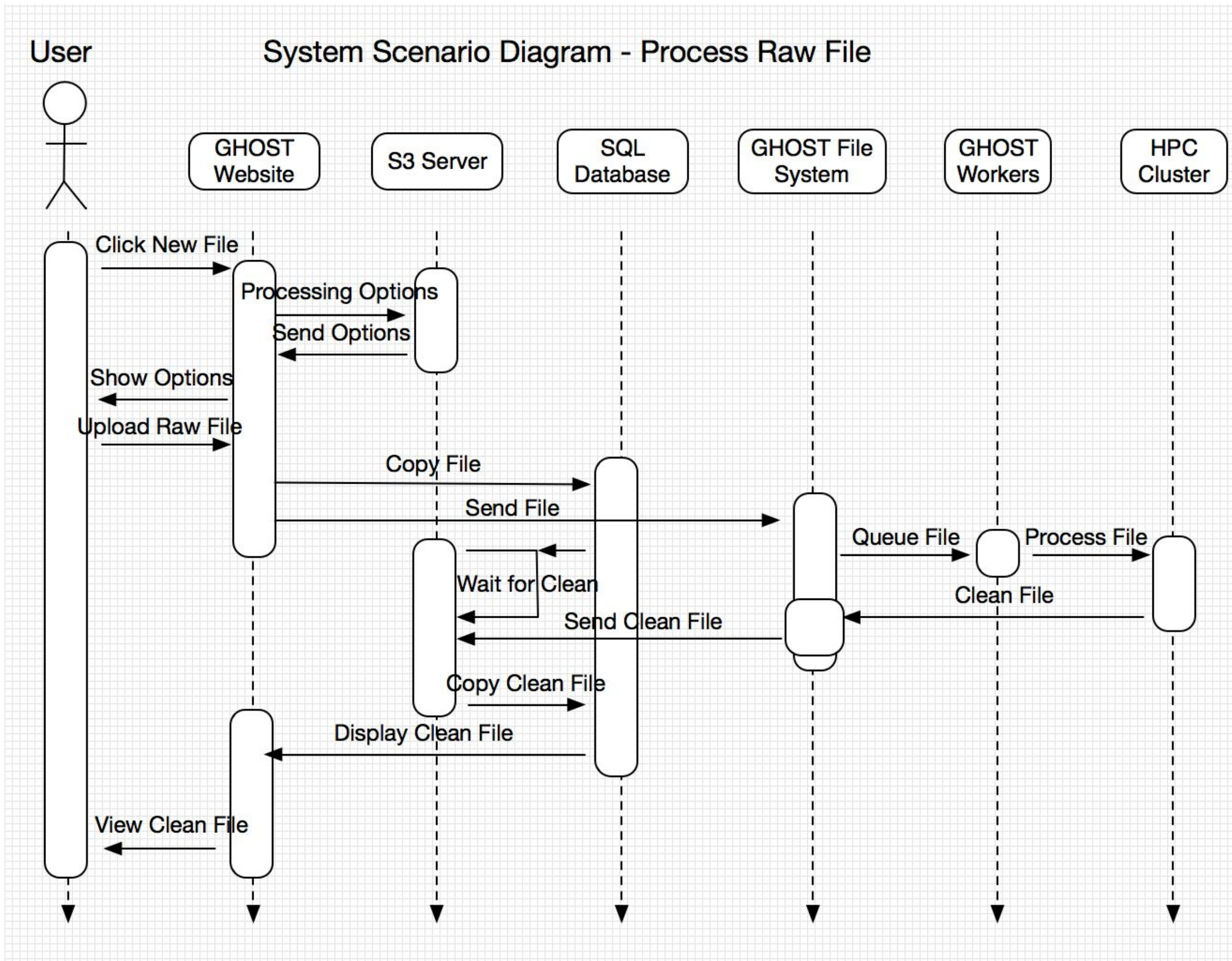

Figure 2. GHOST Dynamic System Scenario Diagram.

Figure 2 above shows the dynamic workflow of processing a raw file from a component perspective. The user begins by choosing to create a new job, and the appropriate options are shown to the user based on the type of job selected. The user then selects the appropriate analysis options to be performed and uploads the file. The raw file is copied over to a SQL database, and sent over to the GHOST file system. GHOST workers then queue the file for analysis to be performed by a High Performance Computing (HPC) Cluster. A processed file called a “clean” file is then returned back to the GHOST file system, copied to the SQL database, and displayed to the user via the GHOST website.

Component Detailed Design

As our project only involves modifications to the website portion of the GHOST system, this section has been separated into the detailed designs of the three components of the GHOST website: Dashboard, Visualization, and Landing Page. Below are the descriptions of each component of the website and the component's detailed design:

Dashboard:

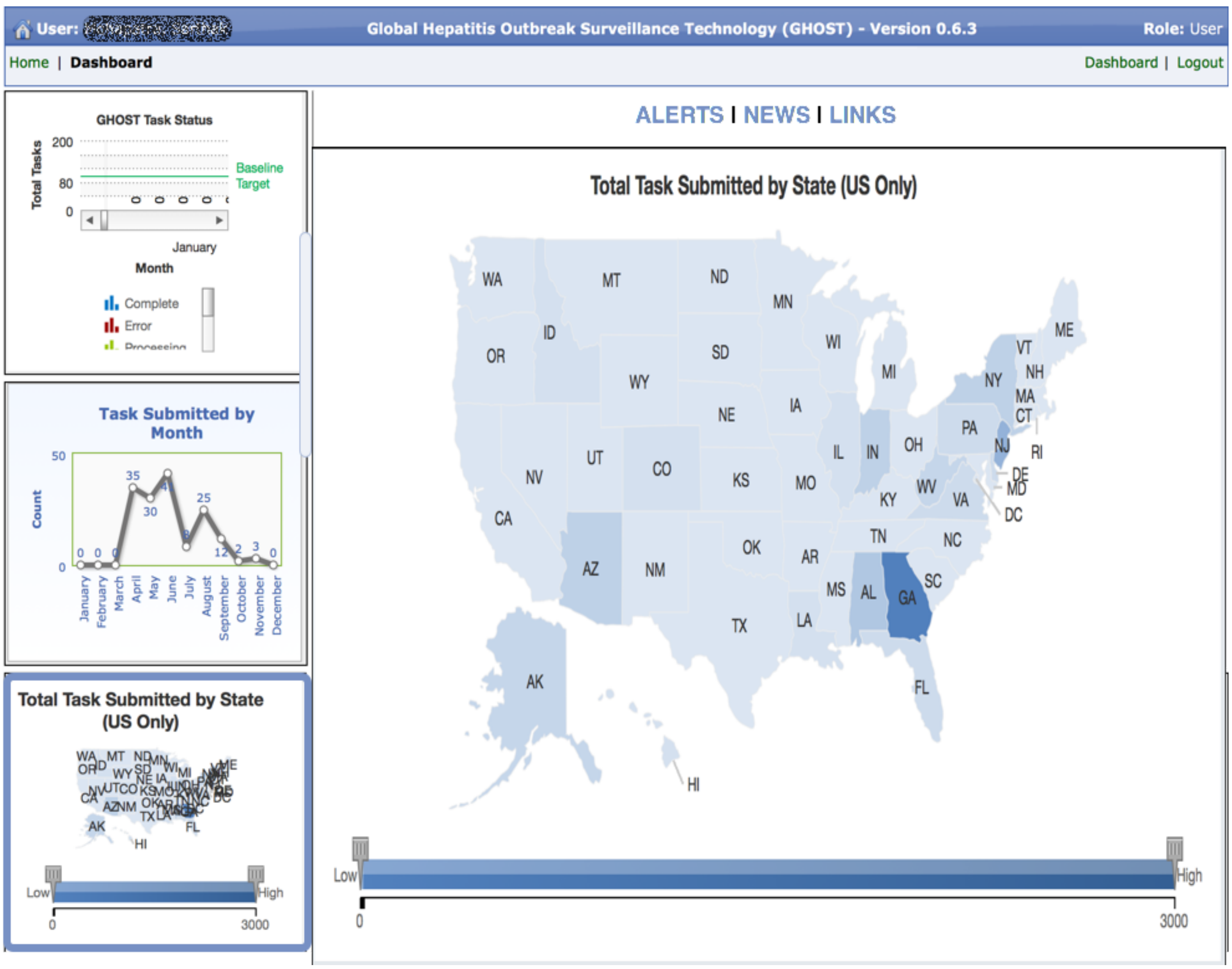

Figure 3. Updated GHOST Dashboard.

The purpose of the dashboard is to monitor various parts of the GHOST system including metadata such as submission frequency and task status in addition to updates, such as alerts and news. The proposed design shown in Figure 3 includes a modular sidebar, which allows for the addition and modification of chart and graph types without affecting the layout of the total page and increases familiarity of the system. Additionally, the Alerts, News, and Links were moved from the navigation menu bar to the section shown above in order to increase the consistency of the system by reducing the confusing internal redundancy of the system. These links will no longer be accessible across every web page within the application, emphasizing their implicit relationship to the dashboard web page and its contents.

Visualization:

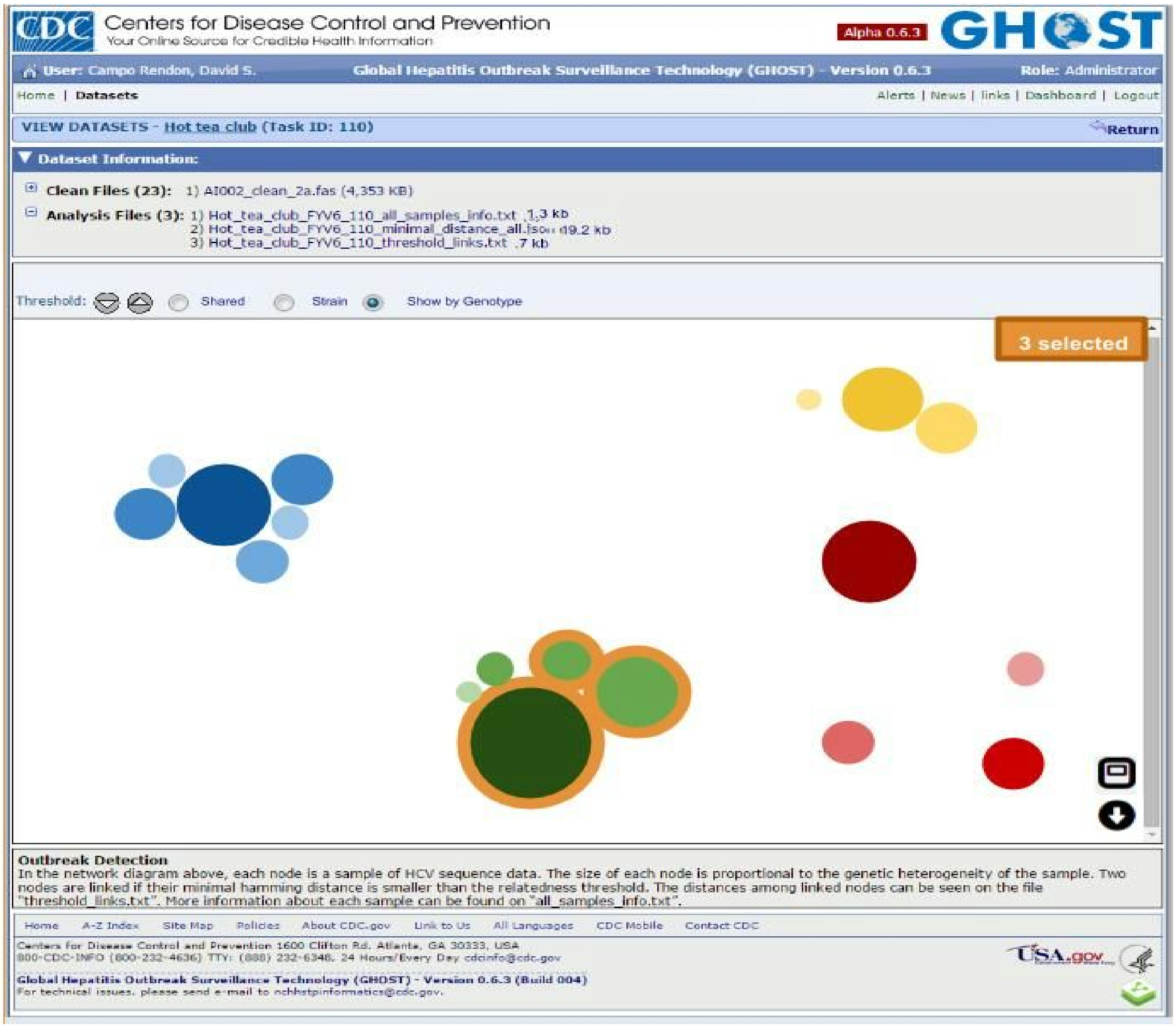

Figure 4. Updated GHOST Node-Link Diagram.

The project uses a node-link visualization in d3.js to visualize a Hepatitis C outbreak. As shown in Figure 4, nodes represent individual Hepatitis C diagnoses and the distance between two nodes reflects shared strains indicating the spread of a particular strain in the form of an outbreak. By utilizing this tool and examining these node-link clusters, user will be able to more accurately chart the path of an outbreak in order to intervene as appropriate.

The visualization will implement a highlighting schema as the method of selection, with the number of nodes selected shown in the top right of Figure 4. The selection schema will make use of paradigms such as control-clicking to select or deselect individual nodes, as well as boxing functionality for mass selection in a form similar to those used in other visualization systems in order to increase the familiarity aspect of the interface. Channels such as hue will denote similarity in strains, and the size of a node will denote the number of cases of a particular genotype of Hepatitis in order to allow the viewer to better interpret these visual traits in a functional way. Specifically, the links that acted as the core display mechanism of the old visualization have been replaced by representing the outbreak degree in the saturation of the hue. The darker/more saturated a node, the higher the degree, representing a source of an outbreak due to the widespread transmission to multiple cases. The positions of the clusters will be used to reinforce similarities in clusters (i.e. different shades of blue will be clustered together, while the green cluster is separate) using D3’s force layout functionality in order to allow users to more immediately locate these sources. Finally, additional channels can be made use of such as shape or position as more information is encoded into the visualization.

Several widgets have also been added to the bottom right of the visualization for ease of functionality. The first widget toggles between modes of presenting the node-link relationships. One mode presents the nodes as spatial clusters, while the other mode presents actual node-links instead of using position to denote similarity. The second widget is a download button to download a screenshot of the visualization. The combination of these two extra widgets will both increase the robustness of the system by allowing for additional visualization methods to be appended by the client team in the future and increase the learnability of the system for novice users by adding to the operation visibility of the interface.

Landing Page:

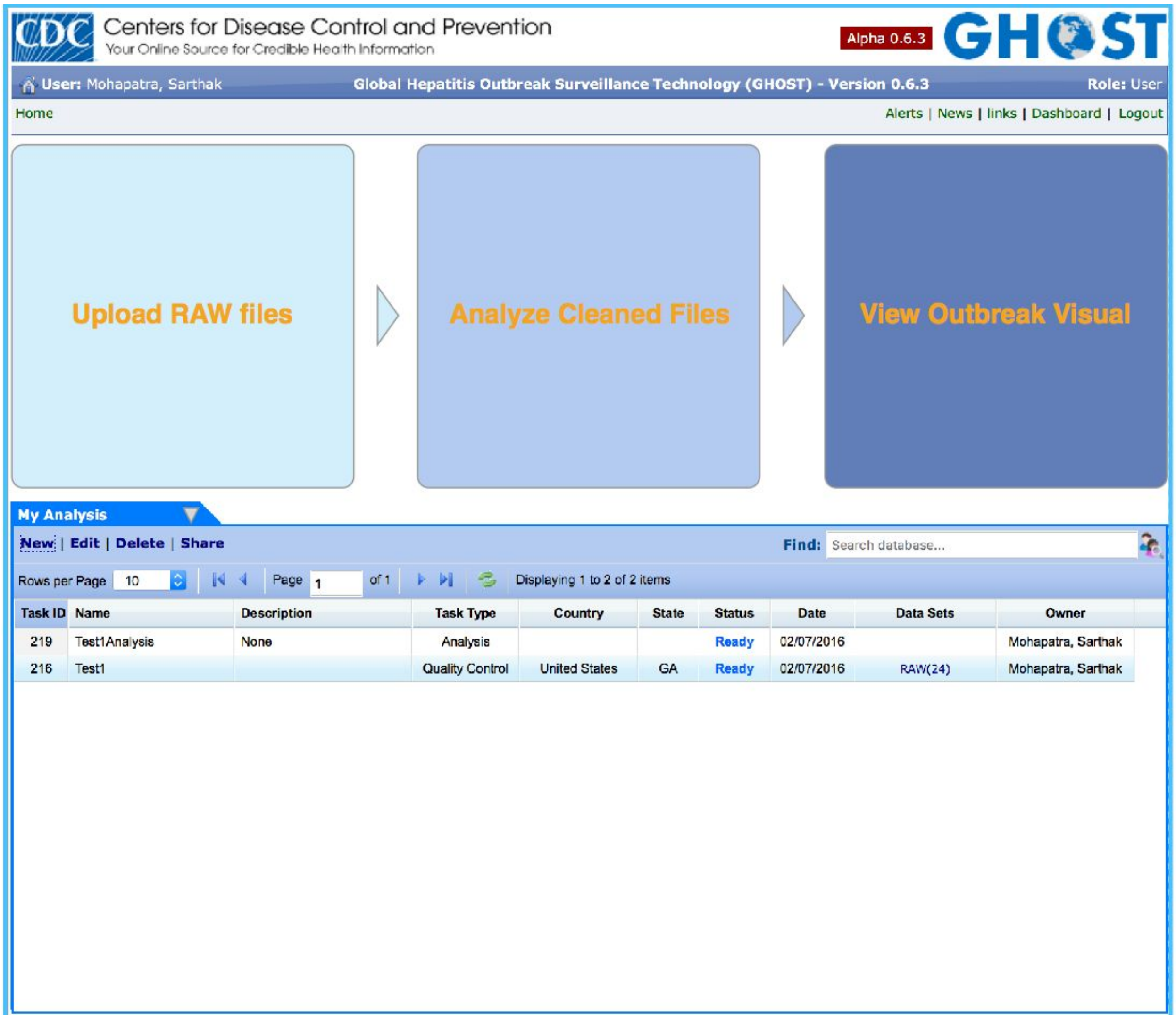

Figure 5. Updated GHOST Landing Page.

The landing page must give new users a walkthrough of how the system works and should be simple enough for novice users to easily interact with and robust enough to capture all possible actions within the core GHOST system. Its primary purpose is to act as an entry point for users to access the outbreak detection functionality of the system by allowing the user to upload raw files, receive and analyze clean files, and view the resulting outbreak visualizations. The proposed design in Figure 5 above demonstrates our changes to the existing system. The original database visualization that dominated the full landing page will be shifted down to the bottom of the page. This database visualization will also now allow minimization in order to reduce the visual complexity of the page while maintaining the substitutivity of the system for advanced users that wish to interact with the full original feature set of the system. In place of an intricate menu system that requires a high level of knowledge about the system, the various available tasks have been reduced down to the key actions needed by core users in order to operate the system in order to reduce the barriers to learnability of the system. In addition to this the action boxes around the tasks have pointers acting as metaphors for indicating the task flow of the system.

Next Steps

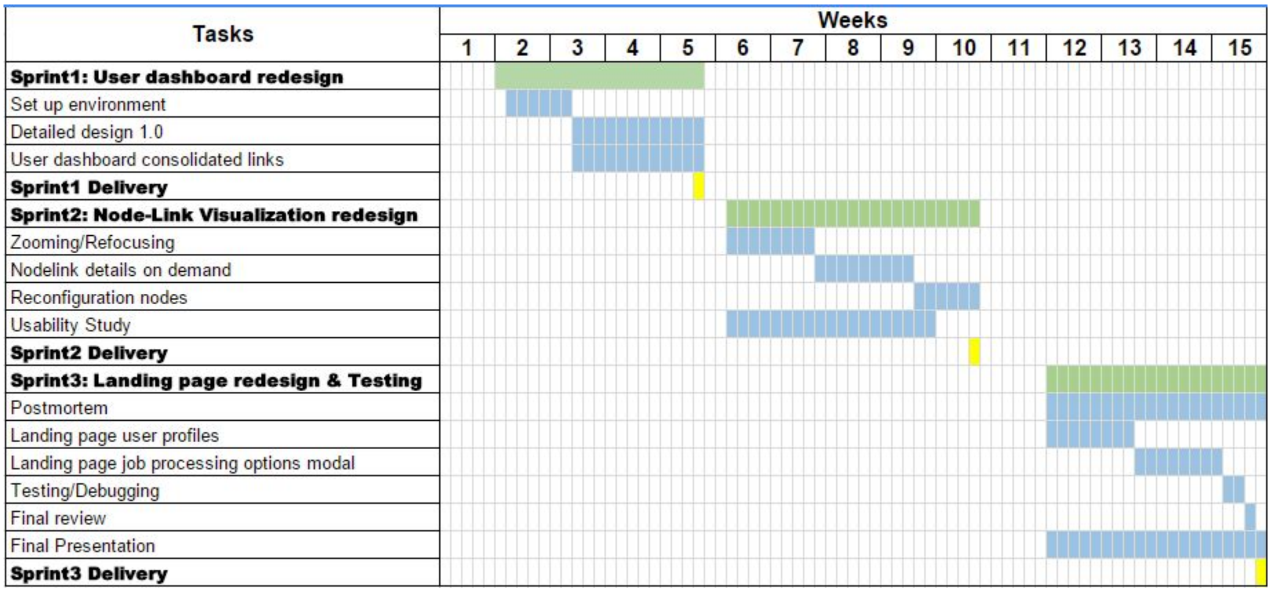

Figure 6. GHOST Implementation Gantt Chart.

The Gantt chart represented in Figure 6 shows how we are managing our tasks and deliverables through this semester. The deliverables have been separated into three sprints. During sprint 1, we will add a modal introduction to landing page and organize all relevant information in the user dashboard to route our users more efficiently. During sprint 2, we will focus on redesigning the node-link visualization. The tasks for sprint 2 is separated into three parts: zooming/refocusing, node-link details on demand, and reconfiguration nodes. During sprint 3, we will work on redesigning the landing page and ensure that all functional and nonfunctional requirements are implemented by testing and revising our implementations. The rows in the first columns show all the tasks that we have worked and will work on for each sprint. Each green colored cell shows the total work period of each sprint and each blue colored cell shows a work period of a particular task.