4.1 Plan Introduction

This Software Development Plan provides the details of the planned development for the GRNsight software CSCI which provides a web application and service for users to upload, view, and edit graphs of gene regulatory networks.

Existing applications that perform similar tasks are difficult to use for small GRNs with only a few genes represented, and often have very high learning curves. Additionally, they require a standalone download that may not be compatible with every user’s computer. GRNsight aims to provide users with a simple, clean, and easy to use application for visualizing small- to medium-scale gene regulatory networks.

As a part of development, a series of deliverables will be created. The project proposal and requirements specification documents have already had initial drafts delivered. The requirements specification document will be resubmitted by April 3rd. This document, the software development plan, will be submitted by March 3rd. A resubmit will be submitted by March 20th. The preliminary demonstration presentation will be presented either April 3rd, April 10th, or April 17th, depending on presentation order. Final product delivery will be May 3th. Written status reports will be delivered every Monday starting March 20th, ending on May 4th.

4.1.1 Project Deliverables

This section will explain in detail the project deliverables that will be completed and submitted throughout the production of the GRNsight software.

4.1.1.1 Project Proposal Document

The project proposal document was submitted on January 23rd. It contained initial information about the scope of the project, the features that are intended to be complete, and an initial set of information about the project itself. It also detailed the rational behind the project. The proposal was presented as a document as well as a presentation given to the class on January 23rd.

4.1.1.2 Requirements Specification Document

The requirements specification document was submitted February 6th. This document contains the information about the requirements of the project, including required features, performance goals, and hardware and software. It also contained projected goals that may be completed over the course of the 402 class. A resubmit of this document will by completed by April 3rd, containing improvements and changes requested by the primary user, Kam Dahlquist, as well as the professor, BJ Johnson.

4.1.1.3 Software Development Plan

This document is the software development plan. It was to be submitted by March 3rd. This document contains the information about the projected plan of development for the GRNsight software as it currently stands. It contains information about the schedule of the development tasks, as well as the deliverables to be given to the customer. A resubmit of this document containing improvements and changes will be completed by March 20th.

4.1.1.4 Preliminary Demonstration Presentation

A preliminary demonstration presentation will occur either on April 3rd, April 10th, or April 17th, depending on where the schedule of presentations places the presentation on GRNsight. Exact specifications on this presentation have not yet been released. However, it is likely that this presentation will display the current progress on the software, and detail any progress that is intended to be completed form the date of the presentation to the final product delivery.

4.1.1.5 Final Product Delivery and Presentation

The final product delivery will take place at noon on May 3th. A poster presentation detailing the accomplishments of the semester, as well as any challenges or lessons that were learned throughout the development process, will take place at 3pm on May 5th.

4.1.1.6 Oral Status Reports

Throughout the entire time of development, oral status reports will be given each week. These status reports will contain information on the current task or tasks being completed, what tasks have been completed since the previous status report, and what challenges are currently being faced in completing the current task, if any.

4.1.1.7 Written Status Reports

Beginning March 20th, there will be weekly written status reports. These written status reports are very similar to the oral status reports. They will contain accomplishments since the last status report, tasks that are to be completed before the next week, and any predicted or current challenges that are faced.

4.2 Project Resources

The following sections will detail the project resources necessary for the completion of the project.

4.2.1 Hardware Resources

Development:

* Macbook Air, macOS Sierra v10.12.3, 1.4GHz Intel Core i5 Processor, 4GB 1600 MHz DDR3 RAM.

Deployment:

* Server hosted on AWS
* Ubuntu 14.04.2 LTS, 7.74 GB RAM
* Details of the machine hosting the server are currently unknown; Information will be obtained shortly.

4.2.2 Software Resources

* Operating System: macOS Sierra v10.12.3
* Editor: Atom v 1.14.4 x64
  + Packages:
    - Stylus v3.1.0
    - Atom-jade v0.3.0
    - Language-git v0.19.0
    - Git-diff v1.2.0
* Project Software Requirements:
  + Node v7.4.0
  + Npm v4.0.5
    - NPM Packages:
      * Chai v1.10.0
      * Cors v2.8.1
      * Cytoscape v2.7.14
      * Express v3.3.8
      * Googleapis v1.1.5
      * Istanbul v0.4.5
      * Jade v1.11.0
      * Jquery-Extend v2.0.3
      * Mocha v2.5.3
      * Moment v2.17.1
      * Multiparty v4.1.3
      * Node-xlsx v0.7.2
      * Nodemon v1.11.0
      * Parallelshell v2.0.0
      * Should v11.2.0
      * Stylus v0.54.5
      * Supertest v3.0.0
      * Xml2js v0.4.17
      * Xmlbuilder v8.2.2
  + Google Chrome v56.0.2924.87 (64-bit)
  + Mozilla Firefox v51.0.1 (64-bit)
  + Libraries:
    - Bootstrap v3.2.0
    - D3.js v3.4.4
    - Bootstrap CSS v3.1.1
    - Bootstrap File Input
    - jQuery v1.11.0

4.3 Project Organization/Human Resources

This section contains the information on the organization of the project. Typically this would contain information about each team member, their role, and what tasks they are to complete. Though GRNsight is a team project containing six individual members including myself, I am operating mostly alone in the development of this part of the project.

There are three major aspects to the project:

1. Graph editing and visualization tools
2. Additional Graph layouts
3. Saving session data

4.3.1 Graph editing and visualization tools

Graph editing and visualization is the first major task to be completed.

The current graph visualization tools do not allow for sufficient zooming and scrolling. They also limit the graph to a small window that is not optimal for most graphs that are uploaded. The graph visualization tasks are as follows:

1. Adjust size of viewport (visible area) to have different sizes optimized for different screens
2. Automatically detect the browser size and select the best size of the viewport
3. Allow for the drafting board (drawable area) to adapt based on the size of the graph. If a node bumps up against the edge of the drafting board, it expands the size of the drafting board.
4. Allow for the drafting board to be larger than the viewport, up to a fixed maximum
5. Enable zooming using mousewheel and a slider
6. Enable scrolling using onscreen arrows, the arrow keys, and left-click and drag

4.3.2 Graph Layouts

Following the completion of sizing, as well as zooming and scrolling, additional graph layouts will be considered. As of yet, the only additional graph layout being considered is a hierarchical tree layout.

1. Investigate how to handle interconnected tree nodes in d3.js
2. Create interconnected hierarchical force tree layout
3. Ensure all existing properties operate as intended on new layout

4.3.3 Saving Session Data

Currently, when a session is closed, the resulting graph is not stored on the server or exported in any way, causing the user to lose all their current progress. Session data will need to be stored to allow for the user to continue from where they left off, or to have a way to view the created graph without taking a screenshot or printing.

1. Allow for saving the graph data, allowing for re-import at a later time. Graph data will be saved in ~~SIF or~~ GraphML.
2. Enable import of graph data that has been exported
3. Create tests to ensure graph data is equivalent to original data
4. Allow for saving image of graph to SVG or other vector format
5. Allow for saving of session data – the current state of the nodes and edges, as well as the current values of the force parameters, viewing/hiding weights, size of viewport, size of drafting board
6. Enable import of session data that has been exported

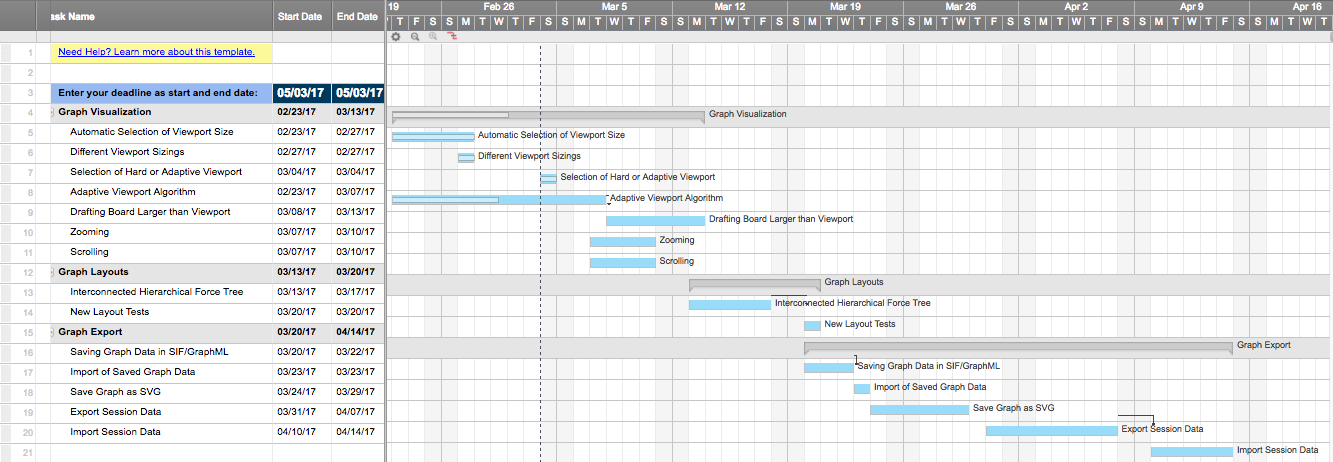
4.4 Schedule

This section provides schedule information for the GRNsight project.

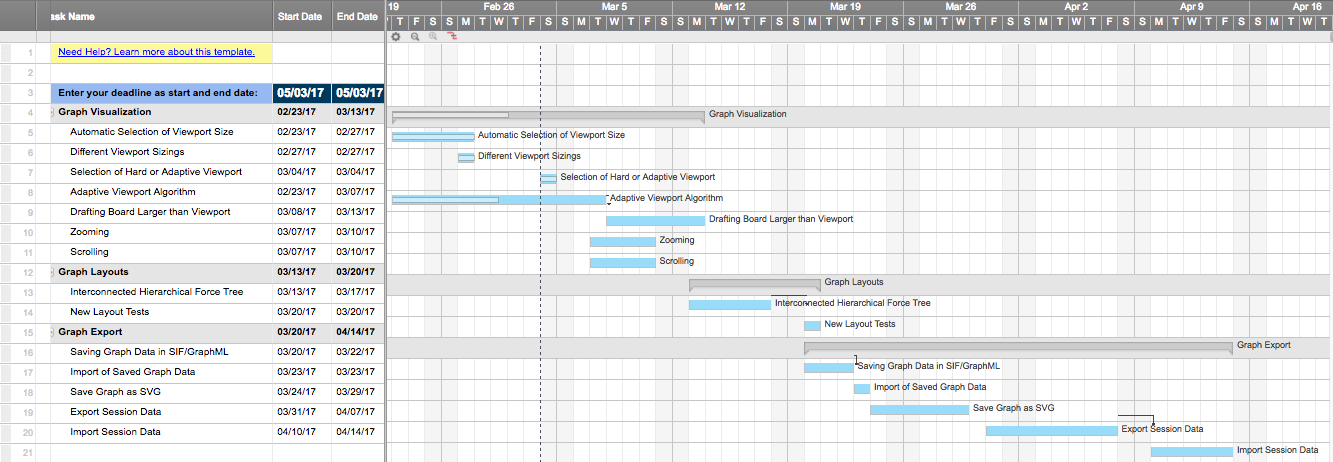
4.4.1 GANTT Chart

As the full GANTT chart is too large to fit here, it will be broken into several sections.

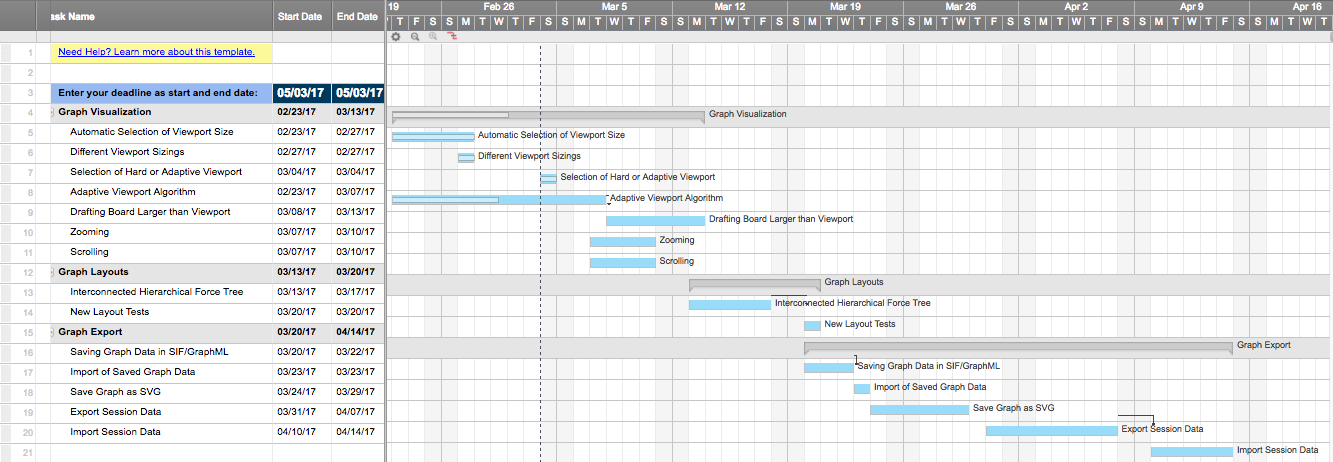
4.4.1.1 Full task breakdown



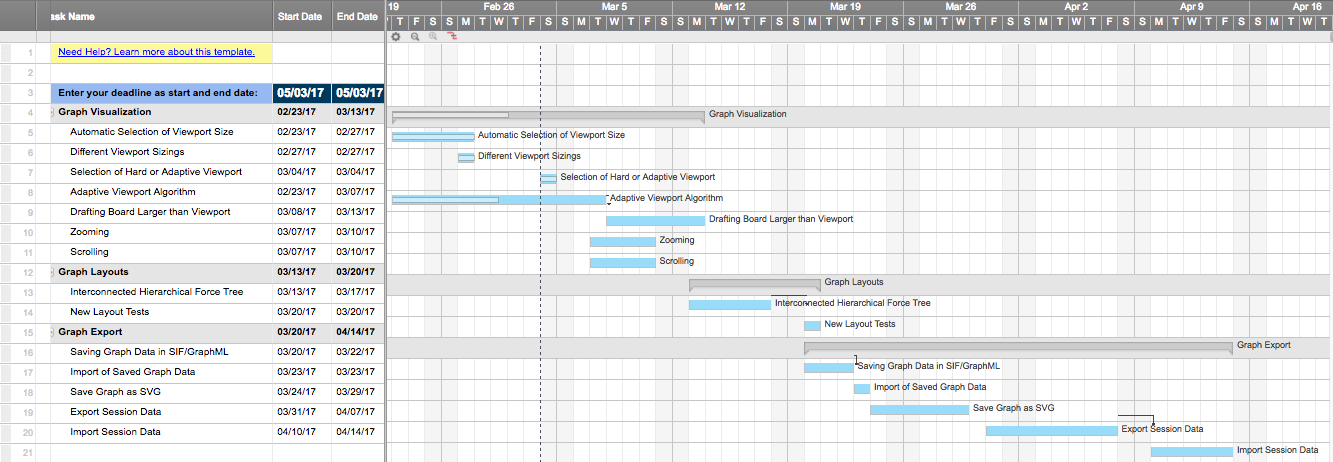
4.4.1.2 Graph Visualization



4.4.1.3 Graph Layouts



4.4.1.4 Graph Export



4.4.2 Task/Resource Table

|  |  |  |
| --- | --- | --- |
| Task | Hardware | Software |
| Automatic Selection of Viewport Size | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Bootstrap  jQuery  D3.js  Google Chrome  Mozilla Firefox |
| Different Viewport Sizes | Macbook Air  AWS Server | Atom  macOS Sierra  Stylus  Node  Npm  Jade  Google Chrome  Mozilla Firefox |
| Selection of Hard or Adaptive Viewport | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  Jade  Google Chrome  Mozilla Firefox  jQuery |
| Adaptive Viewport Algorithm | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  D3.js  jQuery |
| Drafting Board Larger than Viewport | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  D3.js  jQuery |
| Zooming | Macbook Air  AWS Server | Aom  macOS Sierra  Node  Npm  D3.js  Jade |
| Scrolling | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  D3.js  Jade |
| Interconnected Hierarchical Force Tree | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  D3.js  Jade |
| New Layout Tests | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  Chai  Mocha |
| Saving Graph Data in SIF/GraphML | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  jQuery  D3.js |
| Save Graph as SVG | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  jQuery  Jade  Stylus  D3.js |
| Export Session Data | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  jQuery  D3.js  Jade  Stylus |
| Import Session Data | Macbook Air  AWS Server | Atom  macOS Sierra  Node  Npm  jQuery  D3.js  Jade  Stylus |