Project Plan

for

Computer Science Internship Website

Version 1.0

Prepared by Clare Arrington, Harrison Crosse, John Herrin

University of Mary Washington

February 2019

Table of Contents

1. Introduction

1.1 Purpose

This project plan document serves as the guidelines for the implementation team of the computer science internship website. It is based off of the previous software requirements specification provided by the requirements team. After re-evaluation of the document and more detailed discussion with our client, we have an updated list of requirements and a plan of action towards completing our project.

1.2 Scope

Our system's primary goal is to serve as an internship portal for computer science students from the University of Mary Washington. It will be a small website that allows students to post reviews about their previous internships and find information on companies where other students have had internships based on recency and location. Our client, Dr. Jennifer Polack, is a computer science professor who intends to serve as the administrator of this site.

1.3 Document Overview

Project Description (Section 2) gives an overview for our project by discussing the client, users, functional requirements and general constraints. Project Schedule (Section 3) presents our proposed scheduling of the project, along with various justification and reasoning on how we reached these conclusions. Appendix (Section 4) is for extra information one may want to better understand our document.

2. Project Description

2.1 System Overview

The system will be a simple website with only several pages of content. Users will be able to create accounts if they own a UMW email. Subsequently, these users will be able to put pins on a virtual map at the location of the company they worked at. This will provide a base to write a review of their experiences at the internship. Only logged in users will be able to post reviews, however, any visitor to the site will be able to view the interactive map and click on pins to view the reviews for that specific company.

2.2 Client Characteristics

Dr. Jennifer Polack is a computer science professor who noticed the lack of easily accessible information about past internships that students have worked for. She was aware of the features she wanted from the site as well as the general design. We will be using a hosting method she is not familiar with, so one of our challenges at the end will be making sure she is able to use it once the project is completed.

2.3 User Characteristics

The intended end users of the system are UMW computer science students that wish to gain more information about internships experiences at different hiring companies. Additionally, users will be students that share information about their personal experience as interns. We are assuming these students, as computer scientists, are fairly tech literate. Because of that, we believe they will be knowledgeable with basic web interfaces, can fill out a form, and can navigate a search setup similar to Google Maps.

2.4 Functional Requirements

The design allows a potential user to register with an email address ending in @mail.umw.edu. The email address is verified either through email or through the administrator. Once a user is approved as a reviewer, the ability to log in, submit reviews and view information regarding companies is granted.

Authenticated reviewers have access to fill out an internship review form. The form contains different categories to rate their internship experience, with each category having a drop-down menu of client-approved responses. Employer contact information will be requested from the reviewers; information will not be publicly available without approval. Once the form is completed, an email will be sent to the employer requesting approval to display contact information on the UMW Internship Website. They will not be informed of the contents of the review. Additionally, submitted review forms are placed in a pending state, waiting for administrator approval.

The administrator approval is needed to make the review information publicly available. Our client will have a management page that she can access when logged in where she can view pending reviews. She will be able to see the contents of an individual review when selected and can decide to edit the contents before publishing. If she does not wish to show a review, she can choose to deny it. Users will be notified once their review has been acted upon, regardless of decision.

As a guest user, past UMW internship experiences are searchable on the Internship Website. The guest user can search by location, tags, and/or company. The search results are displayed by most recent review first. On the right of the screen, a map is shown with dropped pins of internship job site locations corresponding to the listed results on the left side. If the guest user needs to submit a review of a past internship, the user selects the login button and uses previous credentials. If credentials have not been established, the user selects the request access button located next to the login button on the bottom right of the screen.

2.4.1 Internship Reviews

The internship reviews are the core data of the site. Every review available will be displayed in a list format on the main page, Figure 1, where users can scroll through to search for anything interesting. Figure 2 shows how users can search by details like company name or location to be given a more filtered group of reviews. If a user wants the full information available, they can select one review and view it such as in Figure 3.

2.4.1.1 Review List

1. As a user, I want reviews sorted chronologically, so I can see the most recent activity.
2. As a user, I want to see the company name, rating, tags and location of each internship.
3. As a user, I want to search by any feature in 2.4.1.1.2 to find relevant reviews for me.
4. As a user, I want to see reviews filtered by my search query.
5. As a user, I want to see a specific list of reviews when a company is selected.

2.4.1.2 Specific Review

1. As a user, I want to select one review and see programming languages used, length of internship, likes, type of environment (desk, open space, etc.), prerequisites taken before getting the internship, paid / unpaid status, valuable skills learned, and a summary.
2. As the administrator, I want most values to be from a standard selectable list, so the reviews are more uniform.

2.4.2 User Capabilities

This section covers the distinction between actions that the admin, guests, and logged in users are capable of doing.

2.4.2.1 Guest User

1. As a guest user, I want to log in so I can have access to the internship review form.
2. As a guest user, I want to request access so that I can join the internship review group.

2.4.2.2 Logged In User

1. As a logged in user, I want to be able to submit an internship review form for approval.
2. As a logged in user, I want to log out of the system.
3. As a logged in user, I want to search for internship reviews.
4. As a logged in user, I want to be reminded that reviews are approved by UMW faculty.

2.4.2.3 Admin User

1. As an admin user, I want to be able to approve/edit/deny internship review posts before letting them become publicly available.
2. As an admin user, I would like to privately store the contact information of company employees if the approve.

2.4.3 Map Functionality

The map is the largest add on for the website. This is meant to directly improve the search experience of users, so they can reference internship locations immediately without the use of another tool.

1. As any type of user, I want to be able to view a map with pins designating the location of companies which have reviews written about internships.
2. As a logged in user, I want to to place pins on the map where I have been an intern.
3. As any type of user, I want to click an existing pin in order to alter the reviews displayed on the side to match the company selected.
4. As any type of user, I want to zoom in or out of an area on the map in order to display the reviews relevant to the area selected.
5. As an admin user, I want to be able to remove unnecessary pins from the map.
6. As an admin user, I want to be able to add required pins to the map.

2.4.4 User Interface

It is expected that there will be at least three separate web pages that accomplish different goals. These can be viewed in the wireframe section 2.4.4.4.

2.4.4.1 Splash page

1. As any type of user, I want to be initially shown the splash page on entering the website.
2. As any type of user, I want the splash page to display reviews, a map, and login/registration functionality.
3. As a guest user, I want to be able to create an account with my UMW email.
4. As a registered user, I want to be able to log in from the splash page.

2.4.4.2 Admin page

1. As an admin user, I want to view approved and unapproved reviews on a page that only admins have access to.
2. As an admin user, I want to approve/disapprove reviews with a check box located near the review.

2.4.4.3 Review Page

1. As a logged in user, I want to submit a review with a form located on the review page.
2. As a logged in user, I want to fill out the information covered in Requirements 3.1.2-3 and submit the form for admin approval.

2.4.4.4 Wireframes

Figure 1: Splash Page

This is the primary page that users will be on and serves as the framework for every other page. It features a list of reviews on the left side and a map component on the right. Users can log in or register in the top right corner. They can search or, if logged in, can submit a review. The final button simply links to the computer science wiki which is discussed in 4.3.

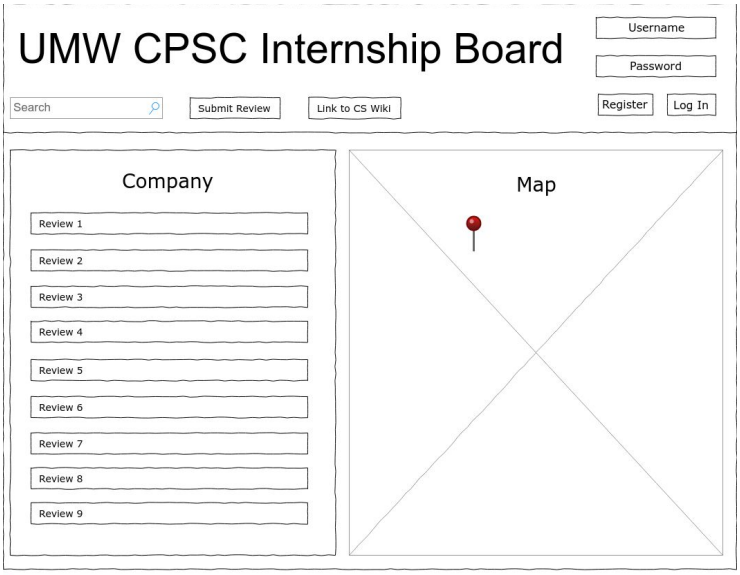

Figure 2: Selected Pin / Searched Company

This shows how the left list changes based on map input.

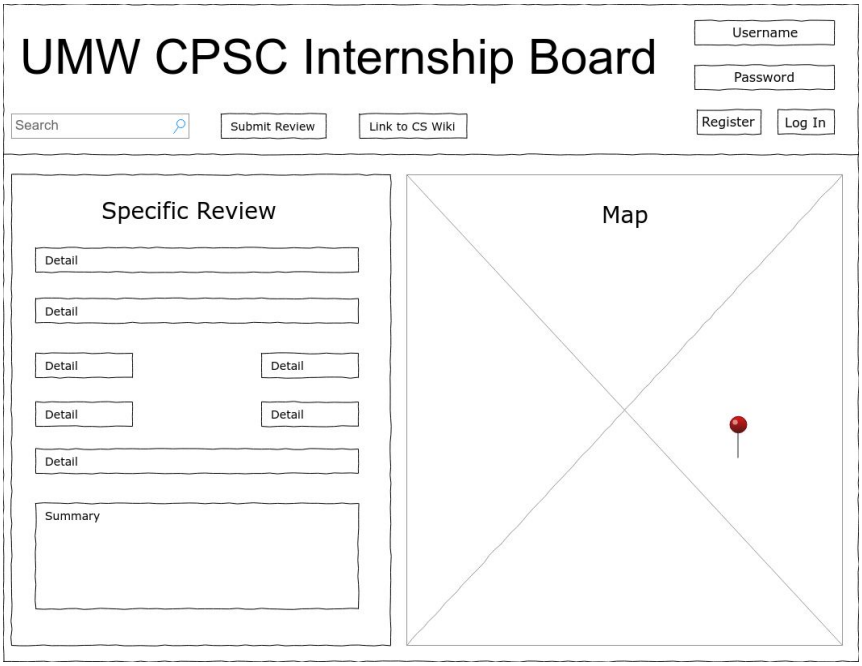

Figure 3: Selected Review

The left bar can also change to show the full information of a single review.

Figure 4: Admin Page

The admin page features a list of pending reviews and a view of a selected review that our client can edit.

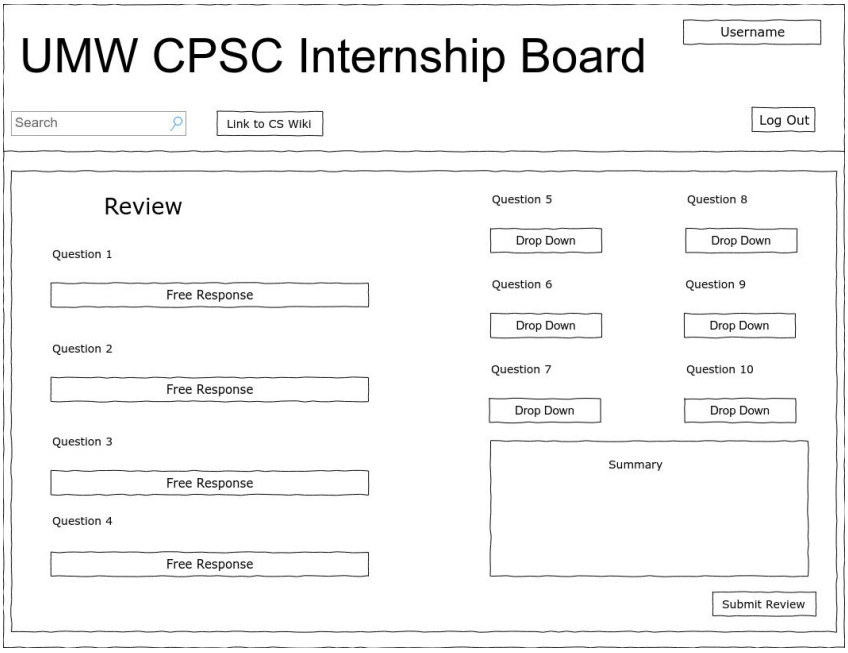

Figure 5: Review Submission Page

This page requests information from the user when submitting a review.

2.5 General Constraints

The project team recognizes that with the budget constraints and limited time to implement a secure website, the following assumptions are made:

- The use of a map API, database management system and web hosting needs to be between $0-20 per month total. If this project is successful, we assume we can maintain this budget past any discounted prices giving during a limited trial period.
- Software security is important, however, given the time period needed to complete the project and the required personnel/skills to actively monitor and secure such a site may not be present. This project design assumes that no PII or other sensitive data will be stored on this database.

3. Project Schedule

3.1 Approach

Our implementation plan is structured into one planning phase and two production phases, based on priority. The planning phase involves creating a foundation for our future work, such as establishing hosting and creating a basic working layout for the website. Following this, we identified the primary project components with the help of our client and have put those in the prioritized development group. This consisted of a working website and database, account usage, admin controls, and review submission. The second production phase are those features that would be ideal to have, but are not completely relied upon by the core functionality of the website. This includes the map, sorting, searching, selecting specific reviews, and emailing.

We will be using Github as our version control system and OpenProject as our project management system. Our team meeting day is Friday and our end day for weekly sprints is on Monday, that way if someone has questions they can ask before the weekend begins. We assessed that this was the most ideal for giving ourselves enough time to meet requirement deadlines.

3.2 Milestones, Deliverables, and Work Breakdown Schedule

The milestones, deliverables and work breakdown schedule have been combined into one section that covers each individual component in every step. For the most part, these milestones are in order of iterative steps. Near the later part of our project, we begin to overlap work on the map with our main section. We believe that work will be able to be split between the two, so we can push towards integration.

3.2.1 Planning Milestone

This milestone will pertain to the creation and delivery of this project plan. This document will help to establish the process our team will use to create the software that has been requested by our client.

- Deliverable: Project Plan Document
- Schedule Allotment: 2 weeks
- Scheduling Justification: Coordinated meetings, project research and document creation will demand at least two weeks to allow for a reliable plan for the future of the project.

3.2.2 Project Skeleton Milestone

This milestone will consist of getting a simple project setup for local development, as well as getting it committed to github in order to establish a reliable version control system for the project.

- Deliverable: Github Repository
- Schedule Allotment: 3 days
- Scheduling Justification: The tasks for this milestone should be completable relatively quickly. The basic project can be created with a few simple commands provided by the NPM package manager. Commiting the project to Github will be quick as well. Three days has been allocated to account for any unlikely issues which should be simple to resolve.

3.2.3 Site Hosting Milestone

This milestone will consist of choosing a hosting service, as well as deploying the existing github project to this service. A database will also need to be established on the web host to store all of the data relevant to users and their reviews.

- Deliverable: A basic/empty website that the client can use to confirm the domain has been claimed and work has begun.
- Schedule Allotment: 1 week
- Scheduling Justification: This milestone will be slightly more intensive than the Project Skeleton Milestone. The most labor-heavy task will be creating and configuring the database containing our user data.

3.2.4 User Creation Milestone

For this milestone, the visitors to the project website will now be able to create a user and login. To complete this milestone, the website database will need to be updated to store the login data being provided by the user.

- Deliverable: A website that allows for user creation/login.
- Schedule Allotment: 2 weeks
- Scheduling Justification: The work required to complete this will consist of creating the necessary fields displayed on the webpage, and passing the user data along to the database hosted on the website. The database will need to be heavily tested to ensure that it is reliably securing the login information, and to protect it against SQL injection. For security aspects, 2 weeks has been allocated in order to properly test these new features and safeguard the data.

3.2.5 Review Creation Milestone

For this milestone, the user will now be able to create a basic review for a company. New fields will need to be added to the website, and the database will need to be updated to reflect the additional data being provided by the user. The user should now also be able to view/search/filter a feed of previously created reviews from all users.

- Deliverable: A website that allows for registered users to create a company review, and view reviews from other users.
- Schedule Allotment: 3 weeks
- Scheduling Justification: The implementation for this encompasses a lot of new features. Thus the three week estimate. The database will once again need to be updated to accept new fields of data for the reviews, and protected against SQL injection. This milestone was also given high priority by the client as one of the earliest things to be completed. A larger buffer was given for this milestone due to its importance, and the fact that it encomposses most of the prominent visuals that users will see after logging in.

3.2.6 Admin Capabilities Milestone

After the completion of this milestone, the website should now be able to differentiate between admin users, and regular users. Admins will be able to accept/deny/edit or delete user submitted reviews. The database should be updated to reflect this new type of admin/superuser when processing a login. Editing of individual reviews should now be possible for the users that originally created them as well.

- Deliverable: A website that allows admin users the ability to deny/accept user created reviews.
- Schedule Allotment: 2 weeks
- Scheduling Justification: This milestone will be exceptionally important to the client, as it they will be the only user with admin capabilities. The features that are being incorporated with this milestone will likely not be incredibly difficult to complete. However, due to the significant power that some of these features contain, they must be properly tested to ensure they are working as intended.

3.2.7 Email Verification Milestone

With the completion of this milestone, the website should now be sending verification emails to the users who have signed up. This process is necessary to confirm that the user who is creating a review is actually a student enrolled at the University of Mary Washington.

- Deliverable: Website can send emails
- Schedule Allotment: 1 week
- Scheduling Justification: The completion of this milestone should not require a large amount of time. The task can likely be solved with a single ‘email focused’ open source coding library that can accomplish the task of sending out an email. The email shall contain a unique access code that, once provided to the website, can establish that the user is authentic. One week should be sufficient in implementing and testing this milestone.

3.2.8 Map Capabilities Milestone

This milestone will pertain to the implementation of a 'Google Map' layout that the user can utilize to pinpoint the location of the workplace they are reviewing.

- Deliverable: A website where the client can create a test review and pin a location on a map pertaining to the company location.
- Schedule Allotment: 3.5 weeks
- Scheduling Justification: Studying and implementing the API for Google Maps may require a significant amount of time in order to provide a satisfactory user experience.

3.3 Gantt Chart

The Gantt chart below was created with Open Project. It provides an overview for the schedule of our project week by week. We divided the implementation of our project into two phases, one epic, and seventeen tasks to be completed across nine weeks. The order of the tasks was largely dictated by their dependencies and priority.

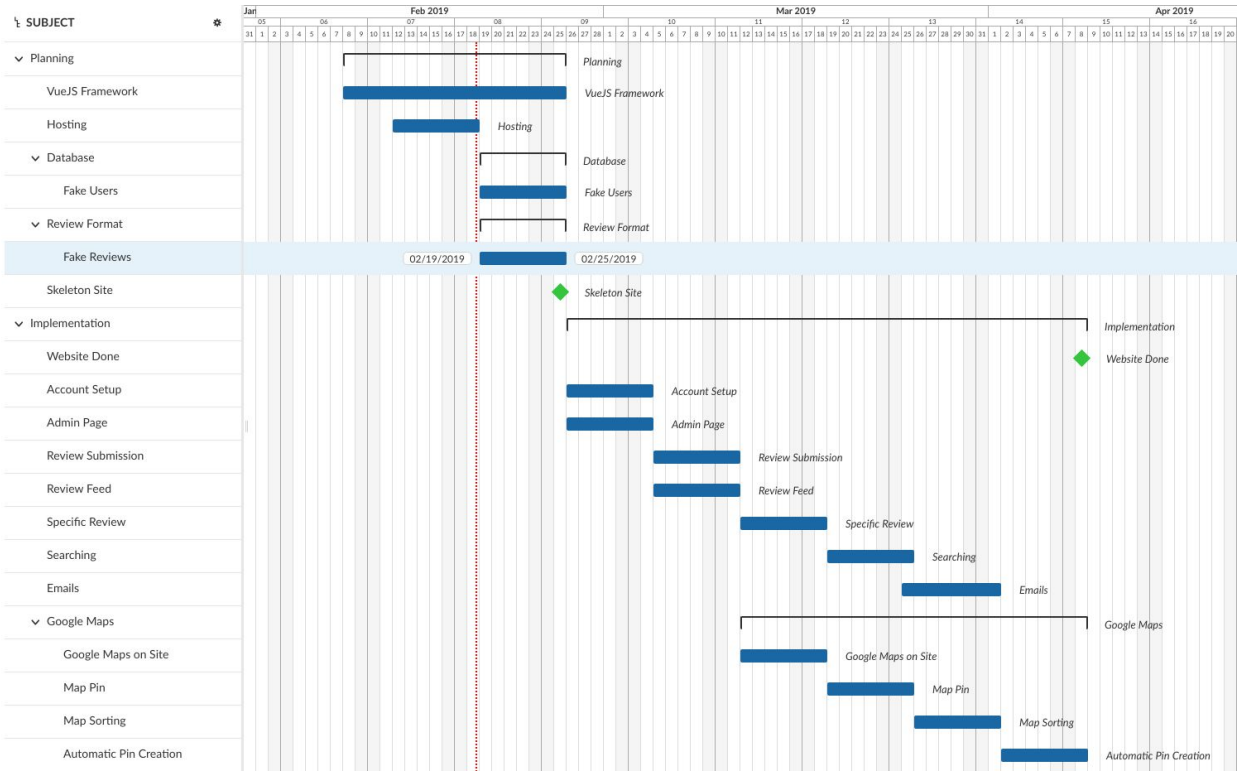

Figure 6: Gantt Chart

Week 1: Planning Phase
- a. VueJS Framework

Week 2: Planning Phase
- a. VueJS Framework
- b. Hosting

Week 3: Planning Phase
- a. VueJS Framework
- b. Database
  - i. Fake Users
- c. Review Format
  - i. Fake Reviews

Week 4: Implementation Phase
- a. Account Setup
- b. Admin Page

Week 5: Implementation Phase
- a. Review Submission
- b. Review Feed

Week 6: Implementation Phase
- a. Begin Google Maps epic
  - i. Google Maps on site
- b. Specific Review

Week 7: Implementation Phase
- a. Google Maps epic
  - i. Map Pins
- b. Searching

Week 8: Implementation Phase
- a. Google Maps epic
  - i. Map Sorting
- b. Verification Emails

Week 9: Implementation Phase
- a. Google Maps epic
  - i. Automatic Pin Creation
- b. Overflow time for non-map features

3.4 Task Dependency Diagram

The diagram below depicts all of tasks in the project and shows which other tasks they are dependent on. The tasks are color-coded by the length of their longest dependency chain.

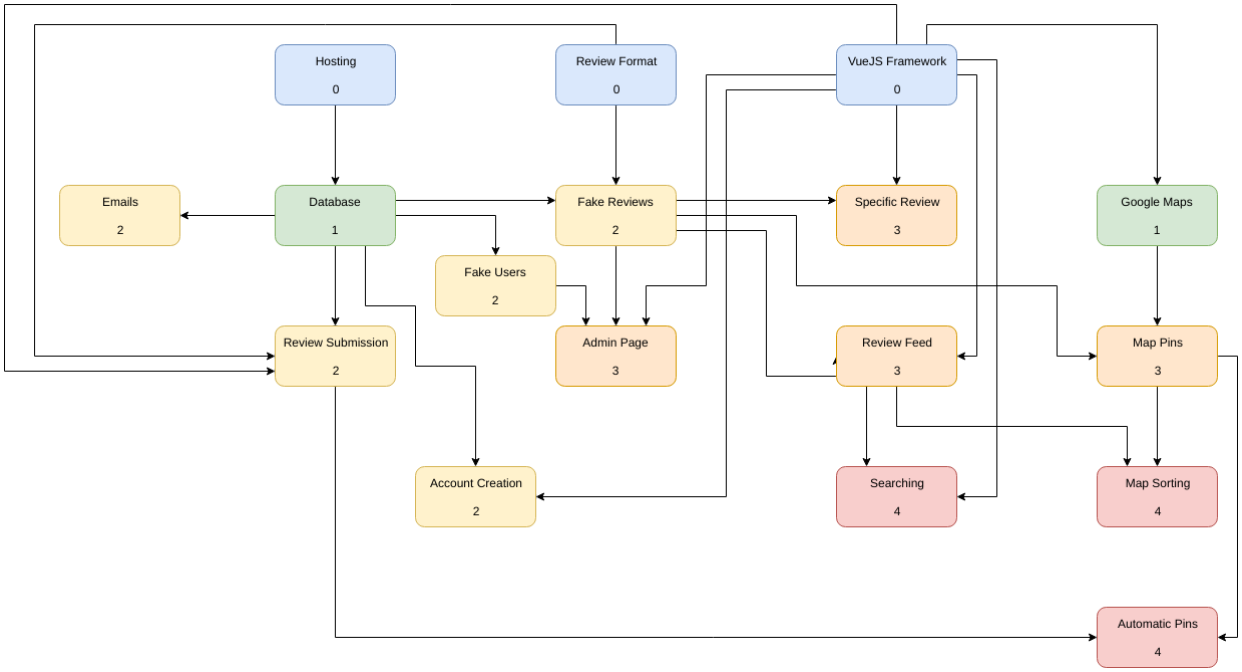

Figure 7: Task Dependency Diagram

0. **Independent Tasks (Blue)**:
    a. Hosting
    b. Review Format
    c. VueJS Framework

1. **Length-One Dependent Tasks (Green):**
    a. Database
        i. Dependent on web host selection.
    b. Google Maps
        i. Dependent on the underlying framework being built.

2. **Length-Two Dependent Tasks (Yellow):**
    a. Account Creation
        i. Dependent on both the database being configured and the VueJS framework being completed.
    b. Emails
        i. Dependent on the database for email storage.
    c. Fake Reviews

   i. Dependent on the review format being finalized and on the database being configured.
d. Fake Users
   i. Dependent on the database being configured for storage of the fake users.
e. Review Submission
   i. Dependent on the review format being finalized, the VueJS framework being completed, and the database being setup.

3. **Length-Three Dependent Tasks (Orange):**
   a. Admin Page
      i. Dependent on fake reviews, fake users, and the VueJS framework being created.
   b. Map Pins
      i. Dependent on the map being finished and fake reviews for locations.
   c. Review Feed
      i. Dependent on the existence of fake reviews and the VueJS framework being set up.
   d. Specific Review
      i. Dependent on the existence of fake reviews and the VueJS framework being set up.

4. **Length-Four Dependent Tasks (Red):**
   a. Automatic Pins
      i. Dependent on review submissions and map pins being completed.
   b. Map Sorting
      i. Dependent on map pins and the review feed being finished.
   c. Searching
      i. Dependent on the review feed and the VueJS framework being finished.

4. Appendix

4.1 Glossary

Administrator User - A user with administrator privileges, typically a UMW faculty member. The administrator manages the site and approves/denies registration forms and review forms.

API - Application Programming Interface

Authenticated Reviewers - Users approved to sign into the site. These users can only submit an internship review form for administrator approval.

CPSC - Computer Science

Guest User - A user not logged into the site. Permissions - View publicly available data.

Necessary Pins - A pin that is required to be placed on the map in order for a review to be displayed.

PII - Personal Identifiable Information

UMW - University of Mary Washington

Unnecessary Pin - A pin on the map that has no review linked to it.

4.2 Author Information

Clare reformatted the document, added additional details to the previously completed sections of 1 and 2, and wrote the approach. John completed the milestones, deliverables, and work breakdown structure. Harrison created the Gantt chart and Task Dependency diagram and wrote their respective sections.

4.3 Additional Documents

Our client suggested that we link to the Computer Science MediaWiki, so students can find posted internships. http://cs.umw.edu/mediawiki/index.php/Career_and_Internship_Information is the link for this specific webpage.