Griffith Buck Rose

Design Document

Team Number 41
Reiman Gardens
Mai Zheng
Amy Hartjen, Patrick Origer, Erik Sandberg, Alexander Reynolds, Devin Amdahl, Logan Schmit, Gregory Carter
sdmay24-41@iastate.edu
https://sdmay24-41.sd.ece.iastate.edu

Revised: 11/5

Executive Summary

Development Standards & Practices Used

- Unit and integration testing.
- Branches must be approved to merge to master on Git repo.
- Design as much shippable code or flushed out prototypes possible before each meeting to get as much feedback from clients as possible.
- Keep good abstraction within code to keep refactoring times low.

Summary of Requirements

- The website shall have mobile capabilities for the 105 Griffith Bucks Roses.
- The website shall have pages for each rose, biographies, homepage, and possibly a degree day calculator page.
- Each rose page shall have the provided database information, list of vendors with the available flower and possibly the degree day information for each flower.
- The Degree Day calculator shall automatically tell users when pests should appear, when flowers should bloom, when to spray pesticides etc.
- The page with the list of roses shall have capability to filter through roses using a fluid key and sort alphabetically, by bloom date and other key information.
- The website shall be hosted on-campus or through a third-party provider.
- Users shall be able to easily edit the information stored in the database through a straight-forward form/UI after logging in with their associated credentials.
- All design choices in respect to the display (color, font, image descriptions, etc) shall be compliant with the ADA's web accessibility guidelines.

Applicable Courses from Iowa State University Curriculum

- COM S 227
- COM S 228
- COM S 319
- COM S 309

- SE 329
- SE 339
- SE 409
- SE 363

New Skills/Knowledge acquired that was not taught in courses

- Maintaining consistent and clear communication with stakeholders not directly on the project team.
    - Client
    - TA
    - Advisor
- Utilizing team members' existing skills and knowledge to assign each member to a knowledge area of our project in which they fit best. These areas include:
    - Client Interaction/Team Organization
    - Scrum Master/DevOps
    - Backend/Database Management
    - Front-end Backend Interaction
    - Testing
    - Individual Component Design
    - Research/Third Party Management
- Utilizing modern Scrum practices to define our activities and tasks, determine the time required to complete each of these activities and tasks, and break our larger goals for the project into smaller and more manageable initiatives (user stories).
- Translating customer desires into requirements and constraints.
    - This includes categorizing requirements into functional and non-functional requirements.

Table of Contents

List of figures/tables/symbols/definitions

Figures

Tables

1 Team

1.1 Team Members

Erik Sandberg
Greg Carter
Devin Amdahl
Alex Reynolds
Patrick Origer
Amy Hartjen
Logan Schmit

1.2 Required Skill Sets for Your Project

- Relevant experience working with the following backend technologies: Java, SpringBoot, (JDBC and JPA) and MySQL.
- Relevant experience working with the following frontend technologies:
- React, Bootstrap, CSS, JS, HTML

1.3 Skill Sets covered by the Team

Amy: Overall Front-end experience (design, implementation, architecture, testing, debugging), MySQL, Java, JUnit, JS, HTML communication and collaboration soft skills

Erik: Back-end development experience with MySQL, Java, Spring Boot, JUnit.

Devin: Backend/database experience with Java, SpringBoot, JUnit, and MySQL. Professional experience in embedded development in C. Exposure to JS/HTML/CSS/React Native, Python, and C++. Soft skills include communication, collaboration, and critical thinking.

Alex: Professional experience in designing large scale industry websites in Umbraco. Experience with moq, mstest, and junit. Some experience with sql and database design.

Greg: Full-Stack development using SQL, Java, Kotlin, JS, HTML, CSS, C/++/#, YAML, XML. Experience with Spring Boot, Maven, Gradle, KTor, JUnit, JQuery, React, MySQL, Oracle Databases.

Patrick: SQL, T-SQL, Python, Java, JS, C, HTML, CSS, SpringBoot, JUnit, Cron jobs, and PowerBI.

Logan: Backend development and old/outdated HTML website skills, primarily experienced in Java and C, former experience that I need to relearn in C++, HTML, Spring Boot, MySQL, and JavaScript. Soft skills: Extremely good descriptive/communication skills and critical thinking. I've led teams before but I am not qualified to do so for this project, I know my place and will support the best I can. I don't have any professional work experience from any internship or co-op

1.4 Project Management Style Adopted by the team

Waterfall + Agile Project Management Style

1.5 Initial Project Management Roles

Client Interaction/Team Organization: Amy Hartjen

Scrum Master/DevOps: Devin Amdahl

Backend/Database Management: Erik Sandberg

Front-end Backend Interaction: Greg Carter

Testing: Alex Reynolds

Research/Third Party Manager: Patrick Origer

1.6 Problem Statement

Informative website on the unique roses created by Griffith Buck. The website contains basic information on each rose, as well as when roses will bloom and are vulnerable to pests. Additionally, there will be informative pages on Griffith Buck, horticulture and Reiman Garden employees. Reiman Garden employees need to be able to edit and manage each of the different roses displayed on the site.

1.7 Requirements & Constraints

The general requirements of the website are for all information given to us by the client to be displayed in a user friendly and intuitive way. The website should contain a list of roses which can be organized and filtered by flower's characteristics. Each flower element should be clickable which opens its page to display the given information. A degree day calculator must be created to predict days in which roses will bloom and pests will appear. There also must be additional web pages for a Griffith Buck biography and other Reiman Garden information. The website needs to be manageable by the clients for years to come after this class year without IT Upkeep from our team. The site needs systems for editing the flowers in the database in a clear way so that even those unfamiliar with programming can manage flowers without needing to access the database directly.

Functional:

- Holds & displays the relevant information for each given flower
- Hosted on the desired host name
- Calculates and displays data about when the flowers will bloom and/or get attacked by bugs.

Resource Requirements:

- Host website and database on ISU or Reiman Gardens server
- Hostname

Qualitative/aesthetic Requirements:

- Rounded Images
- Modern Styling

Maintainability Requirements:

- Site should be extremely easy to edit
- Site should involve as little technology that will require updating as possible

Performance

- Performance requirement is that there are no major delays or interruptions
- The website is expected to handle only simple processes that have no reason to slow down the site.
  - The site needs to be functional on desktop and mobile.
  - There are no legal requirements or environmental requirements.

1.8 Engineering Standards

Maintainability: One of the primary requirements from the client was to make a site which is easily maintainable by a non-technical administrative user. It's unreasonable to make a site which requires no upkeep, rather we play on designing the site in a way that even technologically illiterate people can manage the site. Creating front-end pages to manage the database, having clear functionality for utilizing each feature, and designing the site in a way that there aren't ways to break the systems is part of the plan to make the site

Spring & JPA & Java Software Standards: We have chosen Java to be the language primarily used. This happens to also entail use using Spring for server details, as Spring is quite accessible and effective. Spring Data JPA happens to also be great to use like Spring. These tools with their helpful standards are all useful for sharing and displaying the Griffith Buck Rose information on an easy to use website.

User Interfaces: User Interfaces should be intuitive enough that the vast majority of users can intuitively navigate the site. This could be tested by running usability tests on the site after the UI is created. This is important, because the main goal of the website is to limit how many people call Reiman Gardens about the Griffith Buck roses. If the users are unable to navigate the site to find what they are looking for, they will most likely call Reiman Gardens

1.9 Intended Users and Uses

The main users of this website are rose cultivators and Griffith Buck Rose enthusiasts. This user base will not have high technical skills so the website will be designed to be highly intuitive and user friendly. A requirement of the client is that the website must be easily readable and usable by the elderly, because that is a large portion of their user base for them. While this is a specific audience for the website, any person could be interested in the Griffith Buck Roses and it must be usable by any person. Users could use the website for simple information gathering such as cultivation year, color, name, description, height, parentage, descendants, location, hardiness, vendors and disease resistance. Another use could be that a user has a rose in their head but doesn't know their name so they could filter through roses based on their above listed characteristics. A user could use the website to check when a rose will bloom, when to pesticide them, and when pests will appear. Lastly, a user could be looking for information outside of the roses and instead be looking for information on Griffith Buck, Reiman Garden or horticulture. Lastly, there is a use case for admins to be able to edit the back end information of the website. They need to be able to update information, upload photos, login, remove and add vendors, and change map locations.

2 Project Plan

2.1 Task Decomposition

- Degree Day Calculator
  - Algorithm
    - The degree day algorithm is the way to predict when pests will appear, which our client requested as a function for each rose.
  - Visual/Animated Graphic
    - This feature will enable the user to see the results of the degree day algorithm clearly and concisely. Our client wanted this to be understood in a very consumable manner so as not to get confused by the degree day calculations.
- Fluid Key
  - Algorithm
    - This task is required for users to be able to sort flowers by key features of the flower.
  - Frontend Visual Effect
    - This allows users to visualize the list of their sorted flowers in an easy-to-read format. It will automatically hide flowers that do not fit the criteria.
- Backend Admin Access
  - Login Functionality
    - This task is required to allow administrative users to connect to the backend.
  - Frontend UI
    - This task is required to generate a clean UI for the front-end admin to access the database. It will also include input checking to ensure the correct information is sent to the backend.
  - Database Management Functionality
    - This task is required to enable the ability to access and modify data in the database.
- Vendor Dynamic Visualization
  - Manual Selection UI
    - Allows an admin user to edit the vendors accessible to each flower. Admins will manually select which vendors are actively selling specific breeds of roses.
  - Rose Page Dynamic Visual
    - This visual will make it so only vendors who sell the rose on the page you are on will be visible. This is again to increase the readability of the website.
- Project Setup
  - Website Front-end Creation
    - Necessary task to generate the home page, Griffith Buck page, individual rose pages and vendors page.
  - Database initialization
    - The necessary task to store all data relating to buck flowers and any other information.

    - Server Access
        - Necessary task for deploying the database and website access
- Testing
        - Necessary task for ensuring the safety and security of the website and database

2.2 Project Management/Tracking Procedures

For our larger milestones, we will use a waterfall management style because of the dependencies they have on each other. We will have to implement that style for all large tasks. We are using an agile methodology of development for all other requirements for this project. We chose this style of management because our client is giving constant feedback on the project each time we meet. Additionally, the client doesn't have a technical background, so some of the requirements change based on the plausibility of completion. This requires us to need to quickly change requirements and get that feedback from the client. Overall, it was the management style that works best for our team and this project.

To track progress throughout the semester, our team will be using git issues and the issue board to assign tasks and keep track of what needs to be accomplished. We are putting all tasks, technical and non-technical, in that area to make sure we are efficiently managing the project.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Frontend:

- Home page displays gallery of pictures
- Filters show and work properly
- Tips/notices display in easily noticeable and readable text
- Pages navigate to proper links
- Flower pages display gallery and relevant information about the correct flower

Backend:

- Database tables
- Functional endpoints
- Fluid Key searches

2.4 Project Timeline/Schedule

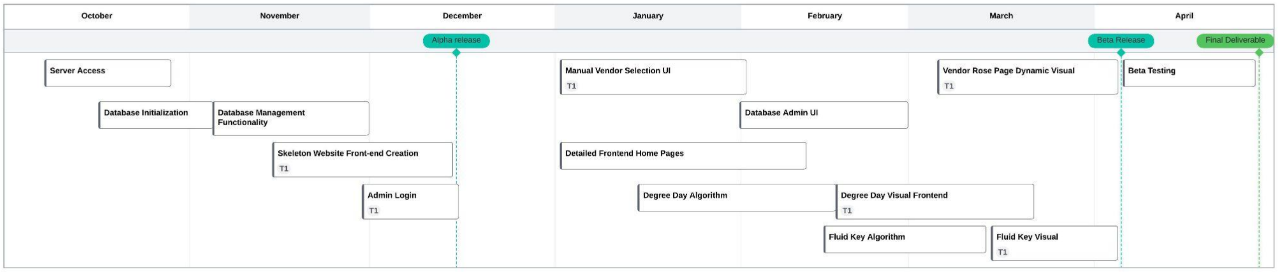

Figure 1: Gantt Chart

2.5 Risks And Risk Management/Mitigation

Regarding our entirely software project, we have no cost-risk with any aspect of our project. The only risk of setbacks are losses of time, either from ourselves making unnecessary development or from requesting information and data from our client which we don't ultimately utilize: 0.3

We'll be using certain algorithms for calculating degree days, these algorithms aren't particularly difficult but will require access to a backlog of weather data for the prior few months: 0.3

Access to the weather data is an unavoidable requirement, we aren't required to pull from any given data provider, but usage of the data is mandatory. With multiple fallback alternatives, this requirement has one of the highest risks of our project but overall a fairly low risk: 0.3

We talked about implementing login captcha for site security. Ultimately we decided on having weekly backups for storing the site's data in case of a danger, this won't entail much risk and should be easy to implement and manage: 0.1

2.6 Personnel Effort Requirements

Task	**Textual Reference & Explanation**	**Total Projected Effort**
Degree Day Calculator *Algorithm* *Visual/Animated Graphic*	To understand the calculations and implement them on the page should be a short process after we setup the access to the weather data https://extension.usu.edu/pests/research/degree-days	50 Hours
Fluid Key *Algorithm* *Frontend Visuals*	This task is deeply intertwined with the core functionalities of the page and will get expanded upon as more and more features are added and the searching is made more polychotomous. The client gave some resources for what they are looking for so we are basing our work off of the example they gave us.	30 Hours

	https://antkey.myspecies.info/en/content/key	
Backend Admin Access *Login Functionality* *Frontend UI* *Database Management Functions*	To create an API for updating the database that is usable by the maintainers without access to IT assistance will mostly take time with frontend design, but will ultimately double the number of pages and functional systems per page.	40 Hours
Vendor Dynamic Visualization *Manual Selection UI* *Rose Page Dynamic Visuals*	This will entail setting up a system for manually linking and listing vendors for each rose's page. It will mostly be made up of complex backend calculations and storage on the database. This work will be entirely separate from our other database work.	15 Hours
Project Setup *Website Creation* *Database Initialization* *Server Access*	Initializing our entire project is a slow drawn out process that takes place over many weeks. It is tedious and very prone to hitting snags that will be difficult to figure out, though the actual process likely won't be overly time consuming to figure out it would be foolish to assume we won't get snagged. Some of this time is also dedicated to database mapping, front-end mockups and overall project design. https://spdload.com/blog/average-time-to-create-a-website/	40 Hours
Testing	Good practice says you should spend more time testing than you do developing, but with small scale projects that estimate doesn't perfectly hold up. We estimate the situation will likely turn out that we will make sure we finish development and then spend the majority of the remaining class time on testing. We also intend to do test driven development starting after the Project Setup Initialization phase. We've estimated the final month of class will be dedicated mostly towards testing. https://www.forbes.com/sites/forbesbusinesscouncil/2022/12/02/software-development-time-estimation-how-long-should-it-take-to-develop-a-product/	N/A

Table 1: Personnel Effort

2.7 Other Resource Requirements

The project is entirely software and will not require any physical materials or parts. For the rose's database we'll utilize a server, and various documents holding information on all of

Dr. Buck's roses as well as the applicable pests that could threaten those flowers. Furthermore, we'll need access to weather data for calculations of degree days for determining upcoming flower blooming and pest hatching estimations. Additionally we will be purchasing a website domain name for hosting the site.

3 Design

3.1 Design Content

Website

- Consistent website structure across all pages (headers, footers, general design choices)
- Ability to navigate to other pages through the website's header and hyperlinked text
- Home page displaying the rose of the day or fact of the day (in relation to one of the Buck roses)
- The ability to filter through the list of Buck roses using flower attributes
  - Fluid Key functionality
- A map of Reiman Gardens displaying the current location of the Buck roses

Backend

- The ability to insert, retrieve, and update information stored in the database
- The ability to handle communication between the frontend and database
- The ability to perform Degree Day calculations
- Consistent endpoint URIs

Database

- A Schema to store information relating to multiple characteristics of Buck Flowers
  - File paths to images on the server
  - Entity of all RHSCC Colors
  - Entity of specific Pests with their Total Degree Days
  - Entity of General Flowers to show parent and descendant relationships
  - Entity of Buck Flowers
  - Entity of Missing Flowers
  - Entity of Regular Flowers
  - Entity of Vendors

3.2 Design Complexity

The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles.

The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

Website: Our website is being built with React and Node.js. The website will also have an interactive map for determining the Degree Days based on a user's selected location.

Backend: Altering the database's data. Handling Degree Day calculations for when the user utilizes the interactive map on the frontend.

Database: MySQL allows us to create a well crafted structure to support the data. Looking through the ER diagram we are referencing when building out the database, how roses have their own descendants and parents adds notable complexity as certain types of roses can be their own parent or descendant.

3.3 Modern Engineering Tools

- Bootstrap
    - Frontend design framework
- React
    - Frontend core
- Node.js
    - Backend endpoint functionality
- Java
    - General coding
- SpringBoot
    - Backend framework
- MySQL (MariaDB)
    - Database management
- Maven
    - Project Management
- GitLab
    - Project hosting and management
- Draw.Diagrams.net
    - Project Design Planning/Diagrams
- Lucidchart
    - Project Design Planning/Diagrams
- Figma
    - Design Mockups

3.4 Design Context

Area	***Description***	***Considerations***
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	This website may bring increased traffic to Reiman Gardens; people may have certain allergies to the cultivations found here. Mitigation against issues with allergies found in Reiman Gardens is resolved by Reiman Gardens. If the website needs specific needs for maintainability

		there may be job opportunities relating to that.
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	The website is primarily meant for people interested in roses. Those interested could be very knowledgeable and experienced in the area to a complete novice. The website is meant to be informative.
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	Our website will hopefully help users in becoming more knowledgeable and aware of how to care for Buck roses. Their improved awareness of how to take care of Buck roses and the methods of dealing with pests will likely improve the environment to a small degree.
Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	The website may lead to increased purchases of Buck roses through listed vendors. Furthermore, the website may lead to an increased number of people visiting Reiman Gardens to see the Buck roses. Lastly, the website's listed vendors and Buck rose locations within Reiman Gardens will need to be updated in accordance with any changes in the listed vendors' supply or the locations of the Buck roses.

Table 2: Project Context

3.5 Prior Work/Solutions

The largest difference between the website we are building and other websites is the degree day calculator that our client has asked us to build. Our client gave us an example of a calculator that the University of Wisconsin created on AgWeather. The shortcomings of this calculator is there is no direct correlative output for the pests. The relationship of degree days to pests is something you must do yourself and is not accounted for by the website. Our website on the other hand will do the correlation between the two data points for you and simply give you an estimate of when pests should appear. The advantage to the website by University of Wisconsin is that it will show more information than ours but that is how we want it since our users will not want or need to see the behind the scenes data and calculations. Their website had a research and data collection focus while our's will be used for education and information.

https://agweather.cals.wisc.edu/thermal-models/get-dds

3.6 Design Decisions

One of the largest design decisions we have made is what information to store in our database. Since this website's main purpose is informational, it was important for us to plan carefully what we needed to store in the database. Most of our client meetings consisted of discussing additional elements we need to add to the database for vendors, sorting/filtering and general display of information. For the displaying of that information we also had to make key decisions about the frontend layout of our website. It was important to design the frontend in an accessible and easily navigable way while still conforming to the way the client wanted it to look. To do this, we made several mockup options and got feedback from our client. Through that circular communication we were able to make a good design decision about the layout of the front-end of our website. Lastly, we will need to decide how to visualize the degree day calculator of the website. This is one of the larger requirements of the client as it requires us to make our own algorithm and user interface. We have two options of how to integrate this information into the front-end once the algorithm is complete. We can either make a separate page for the entire degree day calculator which has information for each pest and some sort of integrated map for selecting degree day information. This would require more backend to frontend communication since the front-end would require information for map display and a list of degree day information. On the other hand, we could have some sort of graphic/animation per each flower that would show how close a flower is to bloom or pests are to appearing. This option would have most of the calculations on the backend and would only send days to bloom/days to pest appear data to the front-end.

3.7 Proposed Design

3.7.1 Design Visual and Description

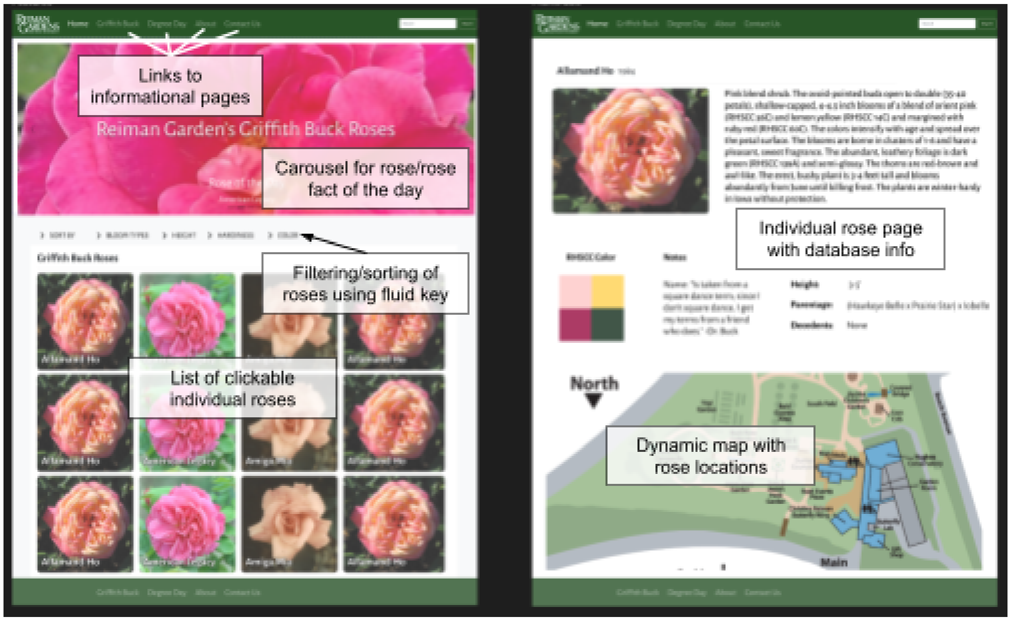
Links to informational pages
Reiman Garden's Griffith Buck Roses
Carousel for rose/rose fact of the day
Filtering/sorting of roses using fluid key
List of clickable individual roses
Individual rose page with database info
North
Dynamic map with rose locations
Main

Figure 2: Frontend Design

The front end design reflects directly on the requirements given by our client and the feedback we have gotten over the course of several mockups. The required functionality you can see within this front-end design is the ability to navigate between informational pages, the display of rose of the day or fact of the day, the ability to filter/sort through the rose list and the list of individual rose components. The figure on the right shows the individual rose page which displays the required information as well as the dynamic map of rose locations. Our client has told us the dynamic map of the flowers might not be possible with the changing layout of Reiman gardens.

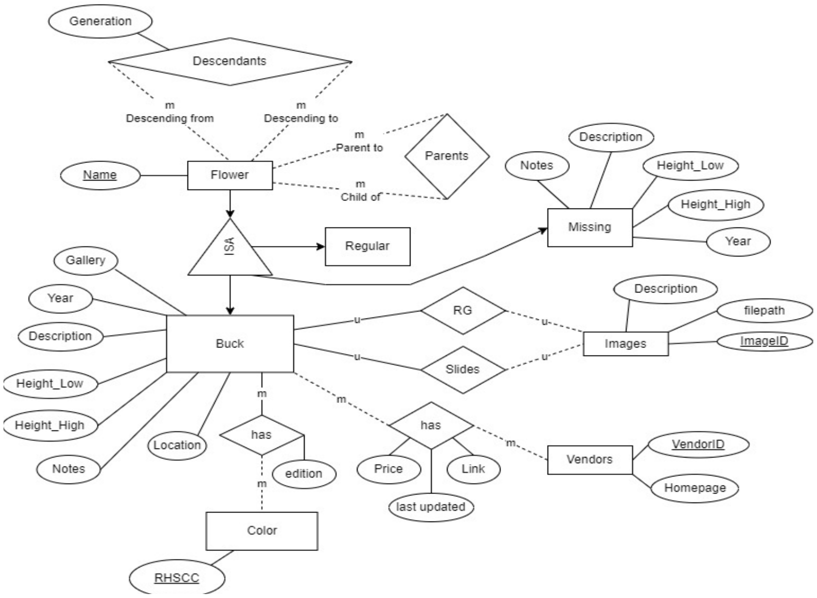

Figure 3: Database Design

The above design is the planned structure of our database in terms of relationships and required information. This design diagram shows the information requirements very well. Each rose needs to have stored data on rose height, year, description, year, gallery, notes, location, color, edition, name, generation and parentage. It also shows the functionality of vendor information where roses have a relationship with vendors who currently carry the specific rose which then can be shown on the rose page. It also shows the structure for showing images on the rose list and the database design will aid in our creation of the fluid key.

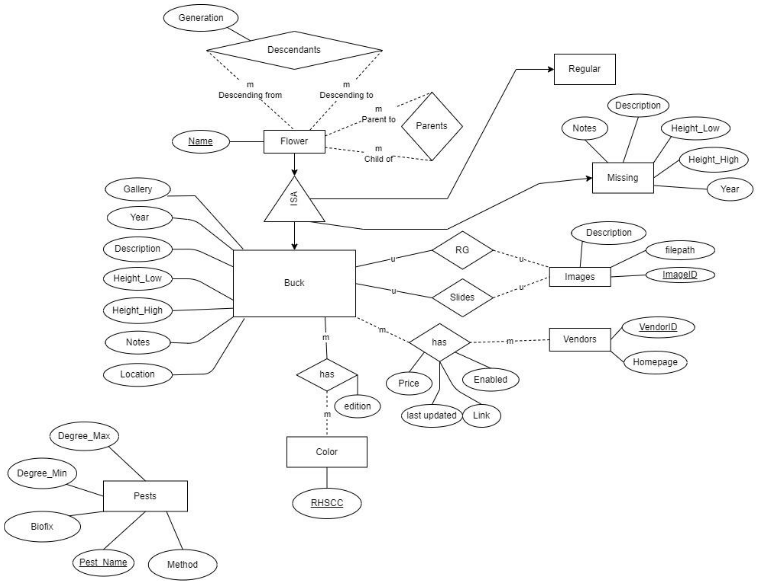

Figure 4: Database Design Second Iteration

The Changes made to the design in the second iteration do not change any of the original design, but add the pests entity necessary for the degree day calculators.

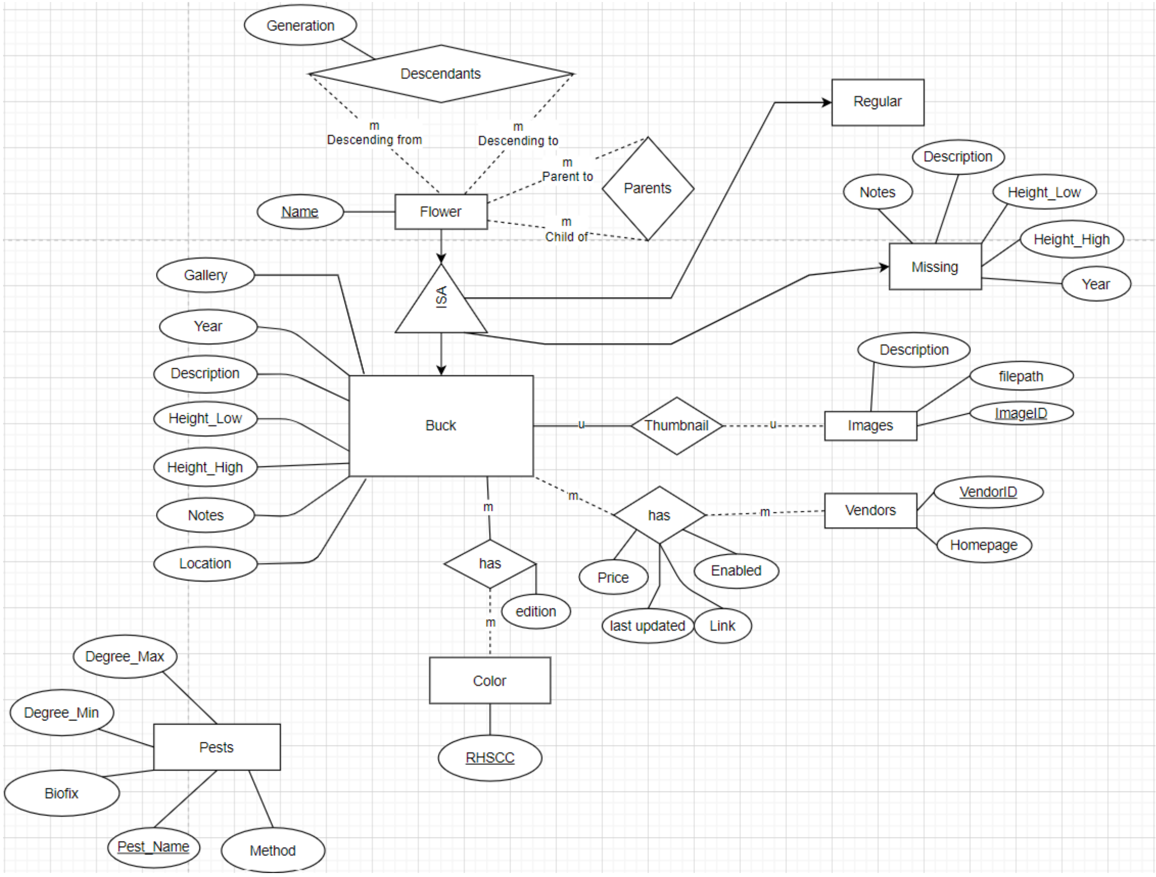

Figure 5: Database Design Third Iteration

The latest design iteration changes the relationship between Buck and Images. The only specific link to an image that Buck flowers need is their thumbnail, any other image will be contained in the gallery folder. This is already an attribute of buck, so this design iteration is mostly for cleaning up the representation.

3.7.2 Functionality

Users could use the website for simple information gathering such as cultivation year, color, name, description, height, parentage, descendants, location, hardiness, vendors and disease resistance. In a real world situation it could be that a user has a rose in their head but doesn't know their name so they could filter through roses based on their above listed characteristics. A large real world user for this website is to check when a rose will bloom, when to pesticide them, and when pests will appear. A user could also be looking for information outside of the roses and instead be looking for information on Griffith Buck, Reiman Garden or horticulture. Lastly, there is a use case for admins to be able to edit the back end information of the website. They need to be able to update information, upload photos, login, remove and add vendors, and change map locations. With the way we have our front-end and back-end designed, users would be able to complete all functionality. We still need to make that decision about the UI for degree days but either way the functionality would be fulfilled.

3.8 Technology Considerations

When creating our initial designs, we considered using wordpress to construct the frontend of the website. Our initial design was to use this website editor to allow the clients to easily edit the contents of their website, however this introduced many trade-offs that would greatly restrict our creative ability in designing the website. For example, individual pages for flowers would have to be manually added and edited by the user, instead of the user being able to fill out a form to make simple and fast changes to the website.

3.9 Design Analysis

Without implementation or testing yet, this design works as it fulfills the main requirements set by the client. This design allows the client to be able to easily edit the contents of the webpage without needing any prior programming experience. The individual rose pages can reflect these changes without the user having to specifically edit that page.

This design is a very good starting design, however modifications may need to be made as the implementation and testing continues. Specifically, the way that the degree day calculator is implemented will need revisions as its design is more developed. Other changes that may need to be made to the website in regards to styling to better match the clients needs. The database design may need to be changed as the implementation develops further, perhaps to better include the attributes of flowers that need to be sorted through the lucid key feature.

4 Testing

4.1 Unit Testing

Units We Will Test:

- General Functionality
    - Degree Days Calculator
    - Admin Editing
    - Filtering roses
- Front-end
    - Navigation
    - Information Display
    - Page Display
    - Admin Login
    - Field entry
    - Button Functionality
- Database:
    - Entities (Model classes):
        - Flower
            - Buck

            - Missing
            - Regular
        - Images
        - Vendors
        - Pests
        - Color
    - Relationships (<entity>Service classes):
        - Descendants
        - Parents
        - RG (thumbnail photos)
        - Slides
        - In Stock (relationship between Buck and Vendor)
        - Has Color (relationship between Buck and Color)

Unit Testing Tools We Will Use for Backend (Spring Boot 3.1.3)**:**

- **Spring Boot Starter Test**
    - **Version:** 3.1.5
    - **Purpose:** Provide a foundational testing suite for Spring Boot applications.
    - **Description:** Maven artifact that includes several testing libraries such as JUnit, Hamcrest, Mockito, AssertJ, etc.
    - **Included Libraries We Will Directly Use For Testing:**
        - **JUnit 5**
            - **Version:** 5.9.3
            - **Purpose:** Unit testing
            - **Description:** Open source unit testing framework for Java.
        - **AssertJ**
            - **Version:** 3.24.2
            - **Purpose:** Make writing and reading tests easier in Java.
            - **Description:** Open source library used for strongly typed, more readable, and more concise assertions than those provided by JUnit.
        - **Mockito**
            - **Version:** 5.3.1
            - **Purpose:** Make the creation and use of mock objects easier in Java.
                - Mock objects are automatically generated.
                - Mockito supports return codes.
                - Tests created as Mocks will not break if interface method declarations are renamed or if interface method declarations' parameters are reordered.
            - **Description:** Open source library used for easily making unit tests involving mock objects easier in Java.
        - **JSONassert**
            - **Version:** 1.5.1
            - **Purpose:** Make the creation of JSON unit tests easier in Java.

        - **Description:** Open source library used for easily creating JSON unit tests in Java. JSONassert converts strings into JSON objects and compares the logical structure and data with the actual JSON.

Unit Testing Tools We Will Use for Frontend (React 18.2.0)**:**

- **React Testing Library**
    - **Version:** 13.4.0
    - **Purpose:** Make the creation of tests easier in React-based web applications.
    - **Description:** Open source testing framework for React that follows the motto "Test functionality, not implementation." In essence, the library performs tests based on DOM nodes rather than rendered components. This allows for refactoring-safe tests.
- **Jest DOM**
    - **Version:** 5.17.0
    - **Purpose:** Make the creation of tests based on DOM nodes rather than rendered components easier in Jest tests.
    - **Description:** Open source testing library that provides a set of custom Jest matchers to extend Jest, resulting in more declarative and easier-to-read tests.
- **Jest**
    - **Version:** 29.7.0
    - **Purpose:** Make the creation of tests easier in JavaScript.
    - **Description:** Open source testing framework built on JavaScript. The framework is designed to primarily work with React and React Native-based web applications.

4.2 Interface Testing

Most Important Interfaces in Our Design:

DBActions, FlowerActions, <entity>Repositories from JPA

How the Composition of DBActions and FlowerActions Will Be Tested

They will be tested in their implementations using JUnits to ensure they are working properly. When we start system testing, the JPA Repositories will be tested as a part of the entire system.

Interface Testing Tools We Will Use:

For the interface testing of our project, the testing libraries and frameworks mentioned in the "Unit Testing" section for our project's backend should be sufficient.

4.3 Integration Testing

Critical Integration Paths in Our Design:

- Test that the flower editing form works and that the data from the database is correctly displayed on the website.

    - **Justification:** The website's main goal is to display Buck rose information so that the target audience doesn't need to frequently call Reiman Gardens anymore.
- Test filter system
    - The filter system is a large system of multiple units working together that needs to be tested because it is a critical component and, if broken, will result in customers calling Reiman Gardens.

Integration Testing Tools We Will Use:

For the interface testing of our project, the testing libraries and frameworks mentioned in the "Unit Testing" section for our project should be sufficient in addition to Spring Test.

- **Spring Test** (Spring TestContext Framework)
    - **Version:** 6.0.13
    - **Purpose:** Make creating unit and integration tests for Spring components easier in Spring Boot applications.
    - **Description:** Framework that supports unit and integration testing of Spring components with JUnit or TestNG. The framework provides consistent loading and caching of Spring ApplicationContexts and provides mock objects that can be used to test your code in isolation.

4.4 System Testing

Systems Level Testing Strategy:

System testing should be completed directly after the completion of large systems and should test the system to make sure it functions as the development team wants.

Set of Sufficient Unit Tests, Interface Tests, and Integration Tests for Our Systems Tests:

- Unit Testing:
    - At least one unit test for each API point
    - At least one unit test for each testable method on the backend
    - At least one unit test for each frontend component
- Integration Test
    - Test each subsystem, including
        - Filter system
        - Pagination
        - Rose editing and viewing
        - Login/account system
        - Degree day
- Interface Test
    - Test communication between highly coupled components

System Testing Tools We Will Use:
For the systems testing of our project, the testing libraries and frameworks mentioned in the "Unit Testing", "Interface Testing", and "Integration Testing" sections for our project should be sufficient.

4.5 Regression Testing

How We Will Ensure New Additions Do Not Break Existing Functionality:

We will ensure that we have full test coverage of base functionality before we start creating functionality that relies on it. As we code, we will create tests for functionality, which means we will always have a test to check that functionality once it is done being programmed. With those tests in place, we can verify that new functionality isn't affecting any old functionality because those base tests are passing.

Our Design's Critical Features and Driving Requirements:

All critical features are driven by the requirements and must not be broken. These include but are not limited to displaying roses, filtering roses, displaying information, administration login, and backend editing, predicting when roses will bloom, and predicting when pests will appear.

Regression Testing Tools We Will Use:

For the systems testing of our project, the testing libraries and frameworks mentioned in the "Unit Testing", "Interface Testing", and "Integration Testing" sections for our project should be sufficient.

4.6 Acceptance Testing

How We Will Conduct Acceptance Testing for Our Project to Involve Our Client:

We have bi-weekly meetings with our clients to update them on progress and give us feedback. During these meetings, we show our client any large functionality or designs to be approved. This ensures that our ideas and application plans should align with our client's ideas. We have additional meetings with our faculty advisor and TA to ensure everything is being done correctly. Our client will review and approve all completed functionality before deeming it complete.

4.7 Security Testing

The admin functionality is the most important place where we will implement security testing. Since this functionality allows backend access to the website, it is the most important place to secure. There will have to be input validation for every area that allows admins to enter information to be sent to the backend. In a perfect world, we could trust all inputs, but if any mistaken inputs could harm our system, it could be detrimental. We will also have to have input validation in the text boxes for login. We will have to limit string length and ensure no spaces, etc. Since admin access gives so much control to our system, most of the security testing will be done to ensure no harmful inputs are being sent to the backend.

4.8 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary concluding that your design is as intended is useful.

Results of Our Testing Plans and How They Ensure Compliance with Project Requirements:

Our extensive testing plans will help deliver a product that meets our client's requirements by:

- Helping ensure issues are identified and fixed prior to the product's official release.
- Helping ensure the product has been created following industry standards/best practices.

- Helping reduce the impact of existing errors.
- Helping ensure that the product creates a positive user experience for the web application's users.

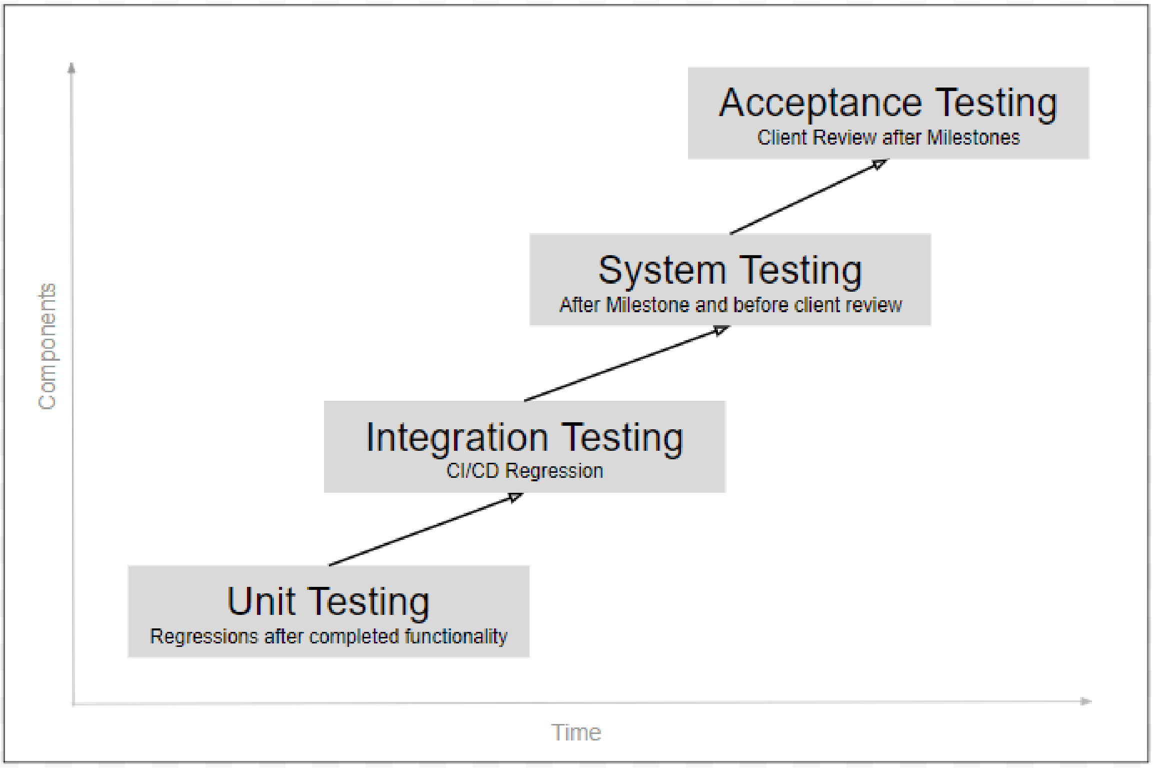
Acceptance Testing
Client Review after Milestones
System Testing
After Milestone and before client review
Integration Testing
CI/CD Regression
Unit Testing
Regressions after completed functionality
Components
Time

FIGURE 6: WATERFALL TESTING SCHEDULE

Acceptance/System Testing	Integration Testing	Unit Testing
Degree Day Calculator	Information Display, Interface, calculation	Button testing, pest prediction, field entry, pest entity, api testing, data validation, calculation
Admin Page	Page design, interface, database storing/front-end updating, login	Button testing, display updating, entity updating, field entry, rose selection, login functionality
Homepage	Page design, rose display, rose/fact of the day	Information display, button testing, field entry, flower entity, image entity, navigation, RG relationship
Misc. Info Pages	Page design, information	Information display, navigation

Filtering	Filtering options, option display, correct filtering	Filter option entities, has color relationship, in stock relationship, color entity, display, button testing
Individual Rose Pages	Page creation, page design, information display	Gallery display, flower characteristics display, individual information components

TABLE 3: TESTING PLAN

Through the above, our testing plans will lead to a more successful product in the eyes of our client, our faculty mentor, our teaching assistant, and our own.

5 Implementation

Degree Day Calculator: The Degree Day Calculator section will be centralized around a map of Iowa, with boundaries of Iowa counties. Location can be derived by inputting the latitude and longitude, inputting the county name, selecting the get my location option that asks the user for access to device location, and clicking on any point of the map of Iowa. Location will be used to locate the nearest weather station and the weather data from that location will be utilized by the Degree Day formula to determine the likelihood of pests being present within the next week.

Lucid Key Flower Filtering and Sorting: A variety of traits will be used for sorting and filtering, such as: Color of different parts of flower, max height, diameter of flower head, time of first bloom, etc. The filtering system for roses will be using AND logic between traits. For example, if pink petals and a max height of 3’ are selected, then only roses that have pink petals and can reach a height of 3’ will be displayed. The filter options will be accompanied by graphics. For example, when selecting the number of blooms in a cluster will be accompanied by a scattering of circles/flower icons numbering the selection options. The sorting can go ascending or descending. Sorting and filtering can both be applied together.

6 Professionalism

6.1 AREAS OF RESPONSIBILITY

Area of responsibility	**Definition**	**NSPE Canon**	**Principle 6: Profession**
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	6.01 talks about using reputable businesses and organizations which help avoid deceptive acts.
Financial Responsibility	Deliver products and services of realizable value and	Act for each employer or client as faithful agents or trustees.	6.07 speaks about only accepting payment when it is well deserved of your work but doesn't talk

	at reasonable costs.		specifically about reasonable costs.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	6.02, 6.08, 6.12 and 6.13 speak on having open communication about your work and sharing software knowledge. They do not specify making public statements.
Health, Safety, Well-being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	6.09 talks about putting the interest of the profession over your own interest. It doesn't necessarily specify everything but directly relates.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	6.06, 6.05 and 6.03 all talk about taking responsibility and acting faithfully. Altogether they cover everything said in the NSPE Canon.
Sustainability	Protect environment and natural resources locally and globally.		6.04 loosely covers this by saying your organizational environment should be favorable to ethical behavior.
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	6.11 and 6.10 talk about following laws and having a professional responsibility to society. This really covers a lot of areas but I think the SE code of ethics also covers much of this area.

Table 4: Areas of Responsibility

6.2 Project Specific Professional Responsibility Areas

Area of responsibility	**Team Performance (High, Medium, Low,)**	**Application to our Project**
Work Competence	High	Work Competence is an extremely important attribute to our project. Our project has numerous high level systems that need to interact with one another, so individual members need to have a high level of competence to ensure each system works as intended and doesn't interact negatively with other systems. Our project leads in these systems all have e
Financial Responsibility	High	We have researched affordable and quality server options for the website, while also speaking with the client on what they have experience with and what they expected out of the server. Also, researching into domain name options led us to considering multiple affordable sellers and a number of quality names revolving around "Buck Rose".
Communication Honesty	High	Our team has been honest and forthcoming with our capability to successfully develop features requested by our client. We are confident in our ability to produce all

		initially requested features. However, an example of communication honesty with our client, is when they asked us to design and add an artificial intelligence that could distinguish the, over one hundred, different Buck Roses from one another. Our team discussed the plan and really wanted to create this for the client but after researching the feasibility of such a task, we presented the client with the required resources and the feature was eventually put aside.
Health, Safety, Well-being	Medium	Our website allows for the filtering of certain traits, such as height and whether the rose possesses thorns. For the instance of thorns, if someone worries about their kids or pets getting into the roses, having easy access to what roses have no roses would be helpful in keeping things safer. In the case of height, the roses could block important lines of sight, so allowing the ability to find roses by height can help potential Buck Roses owners from getting any roses that could pose any danger in their circumstances.
Property Ownership	High	Property ownership is a quality our group has practiced to great success. Any input given by a client is given a high degree of respect and consideration. In the event the client input is inadequate in some way, we work to better understand what they mean and/or offer other ideas that may more effectively represent what the client means. The photos, notes, and other materials given to us by the client are only used appropriately for the purposes of the project.
Sustainability	High	The Buck Rose website project aims to streamline resource utilization at Reiman Gardens. Before the website's implementation, a significant amount of resources were used to answer phone calls inquiring about Griffith Buck Roses. The online availability of this information eliminates the need for such unnecassary phone calls, reducing the environmental footprint associated with distributing information about Griffith Buck Roses.
Social Responsibility	Medium	As with any project that aims to share knowledge with its users, our project has the social responsibility of ensuring the accuracy of all information related to Reiman Gardens, Dr. Griffith Buck, and Dr. Griffith Buck's roses. Furthermore, our project has the social responsibility to ensure that our degree day calculations are correct, as users will utilize this feature to ensure proper care and maintenance of their Dr. Griffith Buck roses.

Table 5: Areas of Responsibility Applications

6.3 Most Applicable Professional Responsibility Area

The area of responsibility that is most applicable to our project is work competence. Since this website won't be used by a large amount of the population its impact on social responsibility, health, safety and well-being as well as sustainability isn't relatively high. What our client is really looking for is a high performing product and an efficient team. For that reason, the most applicable area is work competence which relies on delivering high quality work.

7 Closing Material

7.1 Discussion

Thus far the requirements have all consisted of what the client's design requirements for us have been and how our plans are accommodating those requirements. The client has approved our final mock-design, she has approved the information we're planning to display, and she has clarified any confusion around the site's final functionalities. Thus far everything has gone well, we are on schedule, and the only large hurdles were involved in clarifying that some requests were impossible and unachievable, and thus planning with the client what we would do instead.

7.2 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals. What constrained you from achieving these goals (if something did)? What could be done differently in a future design/implementation iteration to achieve these goals?

This whole semester has consisted of developing each of the points listed previously in this design document, including seeking approval from the client for our design plans, expected requirements, testing plans, interface design, expected functionalities, etc. We have not yet implemented a full scale prototype of the agreed-upon design, as time constraints and other obligations got in the way. We are well underway to achieving the goals set for us by the class and the client, and have confidence that a working prototype will be available on time. Our current plan of action starts with finishing the backend and building a basic frontend to start integration testing early, while we flesh out the rest of the front end, and work out any kinks in the backend.

7.3 References

There have not been any technical references, market survey references, or usage of external research. The only 'external information' has come from our client as information to display on the website.

7.4 Appendices

7.4.1 Team Contract

Team Name Griffith Buck Rose Design Team 41

Team Members:
1) Amy Hartjen
2) Erik Sandberg
3) Greg Carter
4) Devin Amdahl
5) Alex Reynolds
6) Patrick Origer
7) Logan Schmit

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
   a. We will meet face-to-face at 1pm on Sundays at Parks Library room 101h.
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
   a. Discord
3. Decision-making policy (e.g., consensus, majority vote):
   a. Consensus/Compromise
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
   a. Amy will keep meeting minutes, minutes will be kept in the shared google drive

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
   a. Members will attend all meetings on time unless previously agreed upon missing said meeting.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
   a. The team will set expectations for bi-weekly deadlines with each other and our client, which each teammate must follow.
3. Expected level of communication with other team members:
   a. All members are expected to read all discord messages as well as any emails we get related to the project.
4. Expected level of commitment to team decisions and tasks:
   a. Members are expected to articulate opinions on team decisions in a timely manner and are responsible for expressing their opinions as well as the tasks given for deadlines.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
    a. Client Interaction/Team Organization: Amy Hartjen
    b. Scrum Master/DevOps: Devin Amdahl
    c. Backend/Database Management: Erik Sandberg
    d. Front-end Backend Interaction: Greg Carter
    e. Testing: Alex Reynolds
    f. Individual Component Design: Logan Schmit
    g. Research/Third Party Manager(Degree Day and Vendors): Patrick Origer
2. Strategies for supporting and guiding the work of all team members:
    a. Weekly sprint meetings showing individual progress & assigning tasks
3. Strategies for recognizing the contributions of all team members:
    a. Crediting specific members during client meetings so they are getting recognition directly from the client.
    b. Using GitHub issues to maintain who is completing their work

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.
    a. Amy: Overall Front-end experience (design, implementation, architecture, testing, debugging), MySQL, Java, JUnit, JS, HTML communication and collaboration soft skills
    b. Erik: Back-end development experience with MySQL, Java, Spring Boot, JUnit.
    c. Devin: Backend/database experience with Java, SpringBoot, JUnit, and MySQL. Professional experience in embedded development in C. Exposure to JS/HTML/CSS/React Native, Python, and C++. Soft skills include communication, collaboration, and critical thinking.
    d. Alex: Professional experience in designing large scale industry websites in Umbraco. Experience with moq, mstest, and junit. Some experience with sql and database design.
    e. Greg: Full-Stack development using SQL, Java, Kotlin, JS, HTML, CSS, C/++/#, YAML, XML. Experience with Spring Boot, Maven, Gradle, KTor, JUnit, JQuery, React, MySQL, Oracle Databases.
    f. Patrick: SQL, T-SQL, Python, Java, JS, C, HTML, CSS, SpringBoot, JUnit, Cron jobs, and PowerBI.
    g. Logan: Backend development and old/outdated HTML website skills, primarily experienced in Java and C, former experience that I need to relearn in C++, HTML, Spring Boot, MySQL, and JavaScript. Soft skills: Extremely good descriptive/communication skills and critical thinking. I’ve led teams before but I am not qualified to do so for this project, I know my place and will support the best I can. I don’t have any professional work experience from any internship or co-op
2. Strategies for encouraging and support contributions and ideas from all team members:

    a. Encouraging members to speak their opinions and make sure everyone's voice is heard during a team decision.
    b. Through recognizing contributions, members will be able to get credit for their work.
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
    a. Issues that team members have can be brought to attention at the end of weekly sprint meetings

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
    a. Develop and deliver a good working project
    b. Maintain good communication & participation between team members
2. Strategies for planning and assigning individual and team work:
    a. Look at an individual's experiences, strengths, and weaknesses to guide assigning tasks.
    b. Using GitLab issues board to create tasks and then discuss who is best suited to complete them.
3. Strategies for keeping on task:
    a. Follow a meeting agenda for team meetings: Sprint, Discuss any issues with team or project, plan for the next week/make any team decisions, work on any group assignments or group work for the project

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of the obligations of this team contract?
    a. If a specific member is not upholding the team contract, they will be confronted at the end of our weekly sprint meetings to discuss how they can approve and work towards upholding the contract.
2. What will your team do if the infractions continue?
    a. If infractions continue, the team will consider taking credit of the project away from the team member as well as contacting TA and team advisor.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) Amy Hartjen DATE 9/10
2) Erik Sandberg DATE 9/10
3) Alexander Reynolds DATE 9/10
4) Logan Schmit DATE 9/10
5) Devin Amdahl DATE 9/10
6) Patrick Origer DATE 9/10
7) Greg Carter DATE 9/10