**Pollinator Garden Planner**

Final Report

North Carolina State University

Department of Applied Ecology

CSC 492 Team 5:

Kevin Bartushak

Noel Drotor

Tammy Le

Carolyn Thompson

North Carolina State University

Department of Computer Science

December 14, 2018

# **Executive Summary**

Primary authors: Noel Drotor, Tammy Le, Carolyn Thompson

Editors: Tammy Le, Carolyn Thompson

The health and population of managed honey bees and wild bees have declined over the past ten years.[[1]](#footnote-1) Scientists and the public have started programs such as installing pollinator gardens to respond to these widespread declines. These pollinator gardens typically include perennial plants known to be a good source of pollen and nectar for bees and other pollinators such as hummingbirds and bee flies. Creating these habitats can be effective at providing nectar and shelter for pollinators;[[2]](#footnote-2) however, through their local outreach, the NCSU Department of Applied Ecology (DAE) has “noted that garden design remains a barrier for many homeowners who are unsure how to create an aesthetically appealing garden using recommended pollinator plants.” The DAE is sponsoring a senior design project to develop a web-based garden design and visualization app for novice and expert gardeners and homeowners to promote the health of pollinators by creating beautiful home gardens with pollinator plants.

The senior design team’s solution to the DAE’s request is a web application, called Pollinator Garden Planner, that will meet the following functional requirements: creating an account, logging in, creating a garden project, open an existing garden project, filtering the plant list, searching for plants by name, viewing the garden space, checking the garden (against criteria that makes a successful pollinator garden), and generating a shopping list. The team populated a sample database with nine pollinator plants chosen by the team’s main contact from the DAE as a starting point for the minimum viable product (MVP).

As of December 14, 2018, the team solidified functional requirements through use cases; non-functional requirements and constraints have been explicitly defined. The team has finalized wireframes for all requirements and has made modifications based on usability testing and sponsor feedback. The team has implemented the low-level design of the Angular modules and components, REST endpoints, and MySQL database tables that support the application. The features that have been implemented and tested include the following: creating and viewing garden projects, dragging and dropping plants from the plant list to the canvas, and searching and filtering the plant list. The team has developed and executed comprehensive black box and white box test plans. Supplemental guides have been written to help the next senior design team understand the workings of this project.

# **Project Description**

Primary author: Noel Drotor

Secondary authors: Tammy Le, Carolyn Thompson

Editors: Tammy Le, Carolyn Thompson

## **Sponsor Background**

The company sponsor of the senior design project is the NC State Department of Applied Ecology (DAE), located at David Clark Labs. The sponsor has found an imminent problem of declining pollinator populations. To prevent endangerment or extinction of these species, the DAE concluded that a garden design tool focused on using pollinator plants would help address this problem.

Dr. Elsa Youngsteadt has been the team’s primary sponsor contact and advisor from the Department of Applied Ecology; additionally, she has collaborated with Ms. Anne Spafford who has provided supplemental support in pollinator and garden research. They have worked closely with the senior design team this semester to develop a solution in the form of a pollinator garden design web application to save the pollinators.

**Dr. Elsa Youngsteadt,** Primary Sponsor

Assistant Professor and Extension Urban Ecology Specialist

Dept. of Applied Ecology

2301 Gardner Hall

(919) 515-1661

ekyoungs@ncsu.edu

**Ms. Anne Spafford,** Collaborator

Associate Professor

Dept. of Horticulture Science

110 Kilgore Hall

(919) 515-1190

amspaffo@ncsu.edu

## **Problem Statement**

With the decline in pollinators and their health, many programs have encouraged homeowners and individuals to create pollinator gardens to increase habitats for pollinators. However, through their local outreach and extension projects, the DAE has recognized the challenges homeowners face in designing suitable pollinator habitats. Although other online garden design tools already exist, they either do not include plants that specifically attract pollinators or lack other desirable features, such as the ability to view a planned garden from different angles.

## **Project Goals & Benefits**

The sponsor asked the senior design team to develop a garden design and visualization web application called Pollinator Garden Planner that will build upon the lacking functionality found in other existing tools, including not having enough pollinator plants and not being able to view the garden from different angles. Additionally, they requested that the application offer diverse options not found in other tools; it will guide novice homeowners and expert gardeners alike in creating an aesthetically pleasing pollinator garden and allow the user to generate a shopping list of plants to ease the process of purchasing and installing their garden. The Pollinator Garden Planner will interact with a database that contains accurate plant data and mainly includes pollinator plants. The project will persist over multiple semesters to fully implement the application; upon completion, the code base will be delivered to the sponsor, who will host the application through the NCSU’s College of Agriculture and Life Sciences’ (CALS) department and make the application available for public use.

Specifically, the sponsor and the team have identified the following foundational features of the Pollinator Garden Planner:

* Authentication
  + Create account
  + Login
* Garden project space
  + Create new garden
  + Open and edit existing gardens
  + Search and filter plant list
  + View garden from top-down and frontal perspectives
  + View garden in different seasons
  + Check garden (against criteria for good pollinator gardens to guide users)
* Shopping list
  + Print or email shopping list

The senior design team’s garden design tool will benefit researchers, gardeners, and curious homeowners who want to help sustain pollinator populations while creating beautiful gardens. Additionally, NCSU will benefit from this software by claiming the software as intellectual property with the ability to modify and expand it as more research is conducted on pollinators. As the application is utilized more, the local population of both humans and pollinators will benefit from a mutualistic relationship that reaps resources from both parties.

## **Development Methodology**

The senior design team met with the Dr. Youngsteadt weekly on Monday mornings during which the team provided updates, received feedback on their progress, and discussed next steps and outstanding questions. Additionally, a line of contact with Dr. Youngsteadt was maintained through email for any questions requiring quicker responses.

First, the team established the set of requirements, the front-end and back-end designs, and the user interface designs for the entire system. Then, team followed an agile development approach. The team broke the application into smaller features and developed the features end-to-end through iterations. A single iteration spanned three weeks and included task planning for the iteration, finalizing the design, implementing code, and fully testing the feature.

## **Challenges & Resolutions**

|  |  |
| --- | --- |
| **Challenge** | **Resolution** |
| **Development Risk**. One development risk is viewing the garden space from multiple perspectives (namely the top and front views). The team was given an existing NCSU plant database that included a picture for each plant. In order to make the view seamlessly consistent when transitioning between views, it would require multiple illustrations from different perspectives that flexibly scale with any image alterations. Currently, this poses as one of the harder tasks for the team because it requires calculating how the images look from different angles and make sure depth and positions are maintained accurately from both perspectives. | Dr. Youngsteadt found an artist to produce illustrations of each sample plant from a top-down perspective and side-view. Vectors can be used to address concerns about flexible image scaling. JPEGs can be loaded faster on websites due to their low file size, and image quality loss may not be a big issue if the users do not scale the images by a large amount. PNG images will be high quality but due to their high file size, the website’s loading time may be slowed down. As for representing depth of plant images, that can be accomplished mathematically using a grid and coordinates. |
| **Hosting**. The team’s web application will be maintained and hosted through the CALS IT Department, who originally requested that our application be compatible with WordPress. | The team and the CALS IT department reached the agreement that the team would implemented the application on their own virtual machine (VM) (provided by CSC IT during development). At the end of each semester of this project, the team will hand over the code base to CALS IT for them to store on their servers. |
| **Technology.** Some of the technologies, such as Angular, Jasmine and Karma, are new to the team. | One of the team members, Kevin Bartushak, implemented web applications with similar architectures this summer during an internship and guided the rest of the team through learning the technologies and setting up our system. The team also completed tutorials on Angular. |

### **Legal Issues**

The project has been completed within the team’s scope for the DAE. The project has an implied confidentiality agreement[[3]](#footnote-3), as the data supplied to the team comes from a private database still undergoing its prototype stage. Additionally, it was decided by the sponsor and Ms. Heil that the team would not automatically own their intellectual property[[4]](#footnote-4) (IP) on this project because the team’s work would be part of a larger NCSU effort; however, the team has agreed to formally over their IP to ensure that DAE maintains IP of the project. This was complete as of December 10, 2018.

# **Resources Needed**

Primary author: Carolyn Thompson

Secondary authors: Noel Drotor, Kevin Bartushak

Editor: Tammy Le

|  |  |  |
| --- | --- | --- |
| **Resource** | **Justification** | **Status** |
| MySQL database | According to Oracle[[5]](#footnote-5),  “MySQL is the world’s most popular open source database. With its proven performance, reliability, and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties. Additionally, it is an extremely popular choice as embedded database, distributed by thousands of ISVs and OEMs.”  The team deemed MySQL the best choice because it is easy to use, and the team has past experience working with MySQL database. It is a service offered by NCSU[[6]](#footnote-6); this is important because ultimately the application will be hosted by NCSU. | Database has been created and lives on the Windows VM provided by the Department of Computer Science (CSC) IT staff. |
| Node.js Express package | This is a lightweight framework for building mobile and web applications. Node.js is needed in general to connect the Angular application to the database. | The team has downloaded and installed this package. Express has become a dependency located in the package.json file of the code. |
| Node.js Body-parser package | Body-parser takes a group or single piece of data and combines it into one body to be sent to the server side. If this was not used each piece of data would have to be tracked and checked to see if the data arrived correctly in the right format and exact value through the REST API request. | The team has downloaded and installed this package. The Body-parser package has become a dependency located in the package.json file of the code. |
| Node.js MySQL package | Since the database that the team have chosen is MySQL, the REST API must connect to the database. This package completes the connection as well as the ability to make queries and execute transactions. Additionally, it is well documented[[7]](#footnote-7). | The team has downloaded and installed this package. The MySQL package has become a dependency located in the package.json file of the code. |
| Angular 6 and HTML | Angular’s feature module structure works well with the front-end heavy project. The project has a handful of main functions that connect to a database endpoint. | The team has downloaded and installed Angular 6. |
| Windows VM | CALS IT utilizes a Windows-based server on which the team’s system will live after implementation. A Windows VM was requested to match the environment the web application on which it will run. | The team received a Windows VM from CSC IT. |

# **System Requirements**

Primary author: Tammy Le

Secondary authors: Kevin Bartushak, Noel Drotor, Carolyn Thompson

Editor: Tammy Le

## **Overall View**

The Department of Applied Ecology at NCSU requested a garden design and visualization tool as a web-based application through which users can browse recommended pollinator plants to design and view a garden space. Users can filter plant options by several criteria, search plants by name, view the garden space from multiple perspectives, and check their garden against preliminary criteria before installing their designed garden. Users can also generate a shopping list to ease the installation process.

## **Functional Requirements**

### UC1. Create Account

#### 1.1 Preconditions

None.

#### 1.2 Main Flow

The user creates a new account [S1][S2].

#### 1.3 Sub-flows

* [S1] The user clicks the “Create new account” button on the login page.
* [S2]: The user enters an email address, password, and confirmed password (repeated password). The user clicks the “Create account” button [E1][E2][E3][E4]. User is navigated to the My Gardens page (UC3).

#### 1.4 Alternative Flows

* [E1] If the given email address is not a valid email address, the system shall display, “Please enter a valid email address.”
* [E2] If the email address given is already in the system, the system shall display, “This email address already has an account associated with it. Please enter a valid new email address.”
* [E3] If the password doesn’t meet criteria specified in section 1.5, the system shall display, “Password must be at least 8 characters and contain at least one of the following: number, upper-case letter, and special character.”
* [E4] If the password and confirmed password do not match, the system shall display “Your password and confirmed password must match.”

#### 1.5 Data Format

|  |  |
| --- | --- |
| **Field** | **Format** |
| email | Up to 50 alphanumeric characters and symbols ., \_, and @ |
| password | At least 8 characters. Must contain at least one of each the following: number, uppercase letter, lowercase letter, special characters (including !, #, @, $, punctuation characters) |
| repeated password | Same as password |

### UC2. Login

#### 2.1 Preconditions

A user has created an account (UC1) and exists in the system.

#### 2.2 Main Flow

A user enters their email and password to gain access to their account in the Pollinator Garden Planner [E1]. The user can retrieve their password if they forgot their credentials [S1]. Sessions inactive for more than 30 minutes will be terminated [S2]. Upon successful authentication, the user will be directed to the My Gardens page [UC3]. An authenticated session ends when the user logs out of or closes the system.

#### 2.3 Sub-flows

* [S1] The user enters their email address and clicks “Retrieve password” [E1]. The system will send the user an email with a temporary password and a link to reset their password.
* [S2] The user will be automatically logged out and redirected back to the login page.

#### 2.4 Alternative Flows

* [E1] If the user enters incorrect credentials, the system shall display, “Incorrect email or password.”

#### 2.5 Data Format

|  |  |
| --- | --- |
| **Field** | **Format** |
| user | A valid user in the system |

### UC3. Create New Garden Project

#### 3.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user is directed to the My Gardens page.

#### 3.2 Main Flow

The user creates a new garden project [S1][S2].

#### 3.3 Sub-flows

* [S1] The user selects “New garden project.”
* [S2] The user provides a name for the project [E1][E2][E3]. The user then clicks “Create project” and the user is directed to the Edit Garden page (UC5).

#### 3.4 Alternative Flows

* [E1] If the project name given is already the name of an existing project for that user, the system shall display, “You already created a project with the same name. Please provide a unique project name.”
* [E2] If the project name given is over 50 characters long, the system shall display, “The project name must be between 1 and 50 characters.”
* [E3] If the user clicks “Create project” before entering a name, the system shall display, “The project name must be between 1 and 50 characters.”

#### 3.5 Data Format

|  |  |
| --- | --- |
| **Field** | **Format** |
| Project name | Between 1 and 50 characters |

### UC4. Open Existing Garden Project

#### 4.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has created garden project(s) (UC3).

#### 4.2 Main Flow

The user is either directed to the My Gardens page after logging into the system or can navigate to the My Gardens page by clicking “My Gardens” in the toolbar. The user opens the desired garden project [S1]. The user can delete a garden project from the My Gardens page [S2].

#### 4.3 Sub-flows

* [S1] From the list of garden projects on the My Gardens page, the user clicks the garden project they wish to open. The garden project will open in the Edit Garden page (UC5).
* [S2] From the list of garden projects on the My Gardens page, the user clicks the trash button for the garden project they wish to delete; they will be prompted “Are you sure you want to delete this project?” The user clicks “Confirm” to delete the project or “Cancel” to cancel the action.

#### 4.4 Alternative Flows

None.

#### 4.5 Data Format

None.

### UC5. Edit Garden

#### 5.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has either created a new garden project (UC3) or opened an existing garden project (UC4). The user is directed to the Edit Garden page where their project is displayed.

#### 5.2 Main Flow

The user can draw or erase a garden outline using the toolbar on the top of the canvas [S1][S2]. The user can view information about the plant by hovering over the plant image in the plant list [S3]. The user drags and drops plants from the left sidebar into the canvas [S3]. Once a plant has been placed, the user can click on it to view information, add comments, move its position on the canvas, or remove it from the canvas [S4].

#### 5.3 Sub-flows

* [S1] The user uses the circle, curved line, or circle brush to draw a garden shape. They can select a color or texture to fill in the shape.
* [S2] The user deletes a garden shape by selecting the eraser and clicking on the shape they wish to erase.
* [S3] Hovering the mouse over a plant image in the plant list (in the left sidebar) will show details of that plant.
* [S4] A list of plants is presented in the left sidebar. The user can search for specific plants (UC6) or filter the plants that are displayed in the sidebar (UC7).
* [S5] The user drags the desired plants from the plant list onto the garden space [E1].
* [S6] Clicking on a plant that has been placed on the canvas will display a small menu with options to view plant information, add comments for the plant, move the plant, or remove the plant.

#### 5.4 Alternative Flows

* [E1] If the plant is not placed in the boundaries of the canvas, the plant is not placed onto the canvas.

#### 5.5 Data Format

None.

### UC6. Search Plants

#### 6.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has either created a new garden project (UC3) or opened an existing garden project (UC4). The user is directed to the Edit Garden page where their project is displayed.

#### 6.2 Main Flow

The user searches for plants using the plant’s common name or scientific name [S1]. A list of plants with names matching or containing the searched name will appear in the left sidebar.

#### 6.3 Sub-flows

* [S1] The user types the name (common or scientific) of the plant or type of plant by which they wish to filter the list of plants in the system. A list of plants with names matching or containing the searched name will appear in the left sidebar [E1].

#### 6.4 Alternative Flows

* [E1] If there are no plants with common or botanical names that contain the search term (case insensitive), the system displays “No plants match your search.”

#### 6.5 Data Format

|  |  |
| --- | --- |
| **Field** | **Format** |
| Search term | String (can contain letters, numbers, punctuation characters) |

### UC7. Filter Plants

#### 7.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has either created a new garden project (UC3) or opened an existing garden project (UC4). The user is directed to the Edit Garden page where their project is displayed.

#### 7.2 Main Flow

The user can filter the list of plants that are displayed in the left sidebar [S1][S2][S3][S4][S5][S6][S7].

#### 7.3 Sub-flows

* [S1] The user specifies one or more bloom seasons from the options specified in section 7.5 [E1].
* [S2] The user specifies one or more plant colors from the options specified in section 7.5 [E1].
* [S3] The user specifies a plant height range from the options specified in section 7.5 [E1].
* [S4] The user specifies one or more plant types from the options specified in section 7.5 [E1].
* [S5] The user specifies one or more soil types from the options specified in section 7.5 [E1].
* [S6] The user specifies a plant native-ness from the options specified in section 7.5 [E1].
* [S7] The user specifies one or more plant hardiness ranges from the options specified in section 7.5 [E1].

#### 7.4 Alternative Flow

* [E1] If there are no plants that match the filtering options selected in the plant search, the system displays “No plants meet your filter criteria.”

#### 7.5 Data Format

|  |  |
| --- | --- |
| **Field** | **Format** |
| Bloom season | Fall, Winter, Spring, Summer |
| Plant color | Red, Blue, Purple, Pink, Yellow, White, Orange, Green, Other |
| Plant height | X - Y feet, where X and Y are integers from 0 - 10 |
| Plant type | Annual, Perennial, Tree, Shrub, Vine |
| Soil Type | Wet, Moist, Dry (can select range of soil type) |
| Native-ness | Native, Non-native |
| Hardiness | 6a, 6b, 7a, 7b, 8a, 8b |

### UC8. View Garden Space

#### 8.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has either created a new garden project (UC3) or opened an existing garden project (UC4). The user is directed to the Edit Garden page where their project is displayed.

#### 8.2 Main Flow

The user switches the perspective view of the garden [S1][S2]. The user switches the season view [S3][S4][S5][S6].

#### 8.3 Sub-flows

* [S1]: The user selects option to change garden perspective view to frontal view.
* [S2]: The user selects option to change garden perspective to top down view.
* [S3]: The user selects option to change garden season view to Spring.
* [S4]: The user selects option to change garden season view to Summer.
* [S5]: The user selects option to change garden season view to Fall.
* [S6]: The user selects option to change garden season view to Winter.

#### 8.4 Alternative Flows

None.

#### 8.5 Data Format

None.

### UC9. Check Garden

#### Note: Check Garden criteria have not been finalized. Future work is required to solidify this requirement with the sponsor.

#### 9.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has either created a new garden project (UC3, UC5) or opened an existing garden project (UC4). The user is directed to the Edit Garden page where their project is displayed.

#### 9.2 Main Flow

The user clicks the check garden button in the toolbar on the top of the main canvas. A pop-up window will appear on top of the Edit Garden page [S1][S2][S3][S4][S5].

#### 9.3 Sub-flows

* [S1] The Check Garden window will show that the Diverse Seasons criterion is met if the user has met the diverse seasons criterion of a bloom season count greater than 2 for the plants in the garden. [E1].
* [S2] The Check Garden window will show that the Critical Mass criterion is met if the user has met the critical mass criterion of having more than one of each plant species present in the garden [E2].
* [S3] The Check Garden window will show that the Aesthetic criteria are met if the user has met the aesthetic criteria of having multiple colors and multiple heights represented in their plant selections [E3][E4].
* [S4] If the user is satisfied with their garden, they can proceed to the Shopping List page by clicking “Go to Shopping List.”
* [S5] If the user would like to edit their garden again, they can navigate back to the Edit Garden page by clicking “Back to Edit Page.”

#### 9.4 Alternative Flows

* [E1] If the garden does not have plants for at least two different bloom seasons, the system will indicate that the Diverse Seasons criterion has not been satisfied.
* [E2] If the garden does not have more than one of each plant species, the system will indicate that the Critical Mass criterion has not been satisfied.
* [E3] If the garden contains plants of all the same color, the system will indicate that the Aesthetics criteria have not been satisfied.
* [E4] If the garden contains only short plants (0 – 4 feet), only medium plants (4 – 8 feet), or only tall plants (greater than 8 feet), then the system will indicate that the Aesthetics criteria have not been satisfied.

#### 9.5 Data Format

None.

### UC10. Generate Shopping List

#### 10.1 Preconditions

A user has created an account (UC1) and successfully logged into the system (UC2).

The user has either created a new garden project (UC3, UC5) or opened an existing garden project (UC4). The user is directed to the Edit Garden page where their project is displayed.

#### 10.2 Main Flow

The user clicks the shopping list button in the toolbar on the top of the main canvas. A pop-up window appears with message "Do you want to check your garden before continuing to your shopping list?" [S1][S2]. The user is directed to the Shopping List page, where they can see a list of all the plants they have placed into the garden for the project [S3][S4][E1]. The user can export the generated shopping list [S5]. The user’s comments on individual

#### 10.3 Sub-flows

* [S1] If the user clicks “Check garden,” the user will be navigated to the Check Garden page (UC9).
* [S2] If the user clicks “Continue to shopping list,” the user will be navigated to the Shopping List page.
* [S3] For each plant in the shopping list, the plant’s common and scientific names, thumbnail, and quantity are displayed.
* [S4] If the user would like to mark a plant as “Already Bought,” the use clicks “Already Bought” next to the plant name.
* [S5] The user can click the export button to either save the shopping list to their computer as a file (.docx, .doc, .pdf) or print the Shopping List page.

#### 10.4 Alternative Flows

* [E1] If there are no plants placed on the canvas, the shopping list will display “No plants have been added to your garden yet.”

#### 10.5 Data Format

|  |  |
| --- | --- |
| **Field** | **Format** |
| Plant name | Up to 100 alphabetic characters |
| Quantity | Integer greater than or equal to 0 |
| Already Bought | Boolean |
| Remove | Boolean |

## **Non-Functional Requirements**

* NFR 1. The system shall hash and salt passwords to preserve the integrity of user authentication information.
* NFR 2. The system shall have a privacy policy mirrored to NCSU’s privacy policy found at https://www.generalcounsel.ncsu.edu/legal-topics/business-matters/contracts/confidentiality-or-non-disclosure-agreements/.
* NFR 3. Each user’s information is considered private, sensitive information and must not be displayed in URLs and on most pages.
* NFR 4. The system shall comply with Web Content Accessibility Guidelines (WCAG) found at https://www.w3.org/WAI/standards-guidelines/wcag/.
* NFR 5. The system shall preserve the image quality of the plant illustrations when scaled to small and medium sizes.

## **Constraints**

* C 1. The application must run on the CALS IT Department machines, which run Windows servers.
* C 2. The schema of the database must closely follow the format of the NCSU Department of Horticulture plant database.

# **Design**

Primary authors: Kevin Bartushak, Noel Drotor, Tammy Le

Editor: Tammy Le

The senior design team has designed all the features of the web application desired by the sponsor. Due to the scope of the project and time constraints, this project is a multi-semester project. Therefore, the team has developed comprehensive front-end and back-end designs along with wireframes of all the interfaces of the application for current and future developers of this system.

## **System Architecture**

The senior design team created the front end of the web application using Angular and TypeScript to provide functionality for the garden design tool. The application can be broken down into features involving user interactions; therefore, Angular’s feature modules work well with the team’s project. The application communicates with the API layer, which is hosted using a Node.js server and Express.js due to their lightweight frameworks, ease of installation and use, and compatibility with Angular. The API layer relays data between the front-end application and a MySQL database that the team has created to store garden, plant, and user information. The team has selected a MySQL database due to the team’s familiarity with the technology and its compatibility with Express.js and Windows machines.

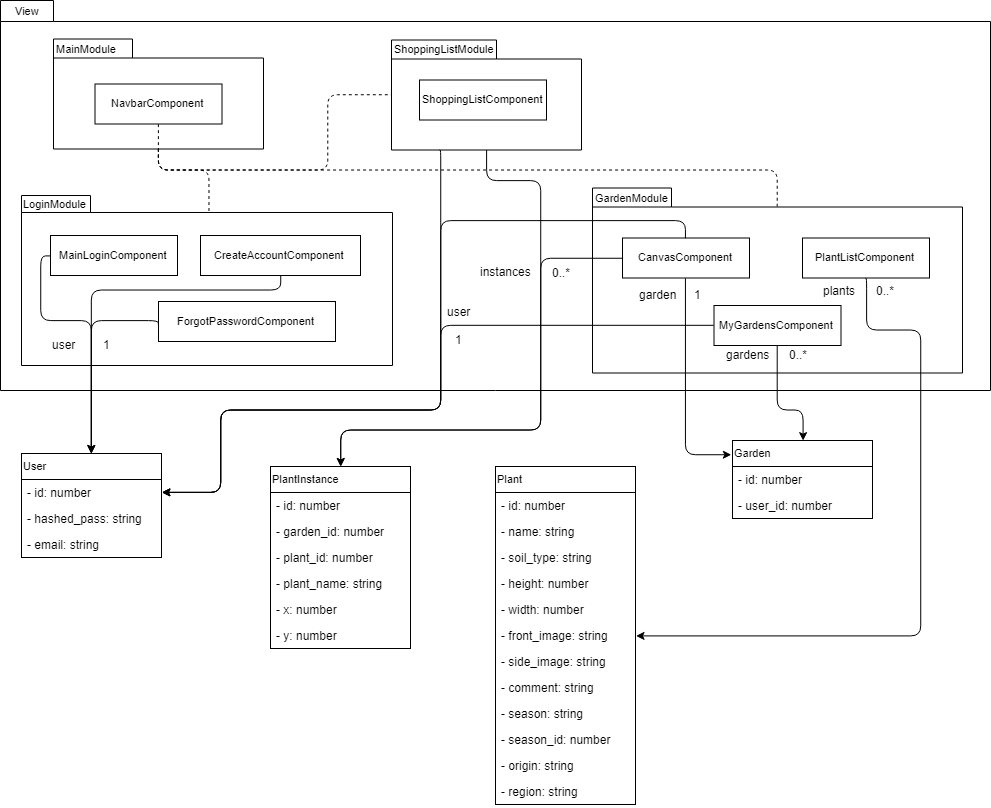


**Figure 1. Diagram of System Architecture**

## **Project Structure**

### **Front-end Structure**

Angular applications group their components into modules, called “Feature Modules.” These feature modules represent cohesive functionalities of a software application; because of this, they fit very closely to the team’s model of system requirements. Using the requirements list, the team broke the application into four feature modules seen below in Figure 2:



**Figure 2. Feature Module & Database Endpoint Relationships**

**MainModule.** This module contains any components that transcend single functionalities in the system. For example, the NavbarComponent is displayed on every page of the application. As the NavbarComponent is simply for navigation and user interaction, it does not need a data model associated with it.

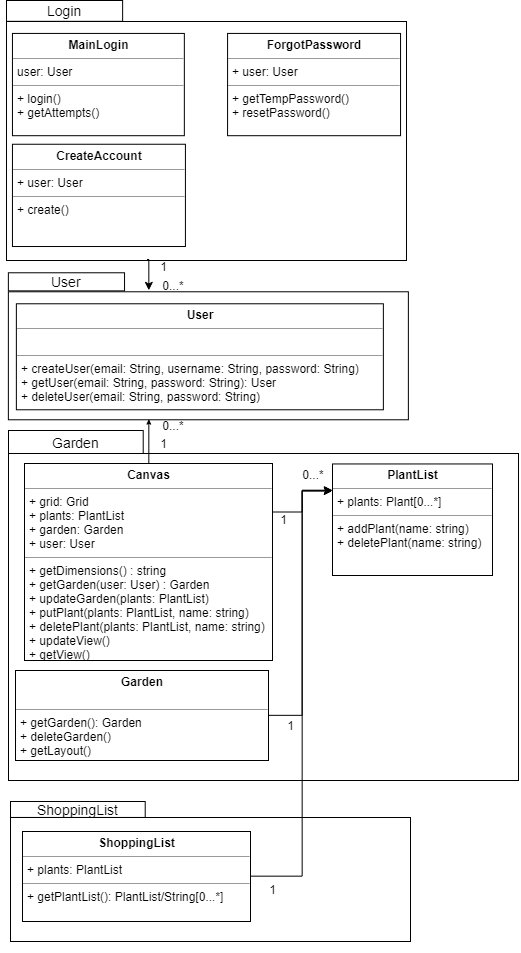
**LoginModule.** This module contains the components that allows the user to log in to the system, create an account, or change their password. All the components in this module use the User class, which contains the fields needed to store a user in the database.

**GardenModule.** The most feature-rich and complex module in the system, this contains the garden planning canvas and plant list for the user to create their garden. Additionally, it contains the popup that prompts the user to load or create a new garden. Components of this module need to interact with the Plant class, which contains information about all of the available plants in the database, the PlantInstance class, which represents each plant that the user places in the canvas, and the Garden class, which is used to save and load gardens that the user has created.

**ShoppingListModule.** This module currently only contains a ShoppingListComponent, which is used to display a list of plants the user has currently placed in the active garden. The team discussed putting this component in the GardenModule but decided that having its own module would be less confusing and allow room for the application to grow in the future.

### **Back-end Structure**

Each class of the front-end data model is stored in the database via the API layer. This API implements CRUD operations to get and store information in/from the database based on REST calls from the front-end. The API is broken up by classes (as described in Figure 3 below), with each set of operations for a class separated into its own component.

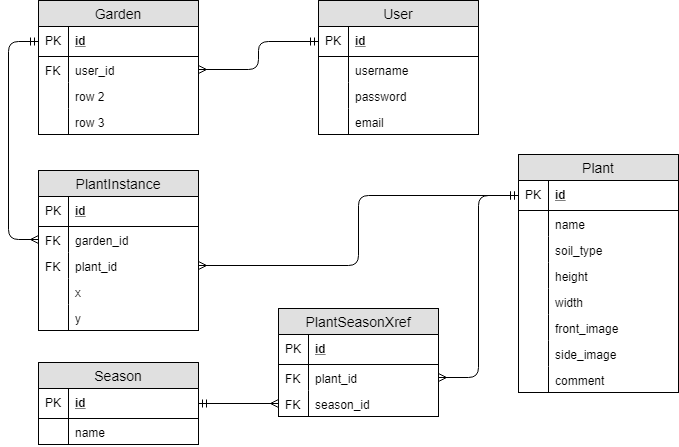


**Figure 3. Overview of Classes and the Relationships Between Them**

According to the diagram, there is a relationship in the API calls between the Login and Garden component by the connection of the User component/class. There is not a direct relationship between the Login Component and the Shopping List component/class because the garden must be loaded first by the user to retrieve the plant list to place the order.

The entity relationship diagram in Figure 4 below shows the structure of tables in the MySQL database. The relationships between classes are shown with the following notation:

* **One-to-Many .** For example, one User may have many Gardens, but one Garden can only have one User.
* **PK.** This notation indicates the given field is a primary key in the database, used to uniquely identify the record
* **FK.** This notation indicates the given field is a foreign key, referencing a field from another table (linked by a relational arrow)



**Figure 4. Structure of the MySQL Database**

Additionally, the team used a cross-reference table to describe the relationship between Plant and Season. This was necessary because Season and Plant have a Many-to-Many relationship. That is, one Plant can belong to many Seasons, and each Season can have multiple Plants associated with it. In order to implement this in an SQL table, a relational table is needed to stores the IDs of the Plant and Season that are related.

### **GUI Wireframes & Flow**

The wireframes found in Appendix A describe the high-level designs of each interface and the flow through which a user interacts with the system. Below is a description of each wireframe.

**Login page.** (Figure A-1) This page contains text fields for the user’s email address and password. Once the user enters their credentials, they will press “Login” to authenticate into the system, which will direct the user to the My Gardens page. Additionally, there is a “Create Account” link that directs the user to the Create Account page and a “Forgot Password?” link that directs the user to the Reset Password page.

**Create Account page.** (Figure A-2) This page contains text fields for the user’s email address, password, and confirmed password. Once the user enters their information, they will press “Create Account,” which will create their account in the system and direct them back to the Login page from which they can authenticate with their credentials into the system.

**Reset Password page.** (Figure A-3) This page contains a text field for the user to enter their email address. Once they enter their email, they will press “Reset Password,” which will trigger the system to send an email to the user’s email (given that the email address the user entered is associated with a valid user in the system) with a temporary password and a link that directs them to reset their password.

**My Gardens page.** (Figure A-4) This page appears as an overlay on top of the page underneath it (either the Edit Garden page or the Shopping List page). The user can choose to create a new garden project or open a project they have previously created from the grid menu of projects. The option to create a new garden project is represented by a plus sign. If there are no existing projects for the user, only the new garden project option will appear.

**Name Garden Project page.** (Figure A-5) This pop-up window appears when the user selects to create a new garden project. It contains a single field for the project name. Once a name is inputted, the user clicks “Create project” and is navigated to the Edit garden page.

**Edit Garden page.** (Figure A-6) This page contains multiple components. The left sidebar is where the user filters, searches, and selects plants to drag and drop onto the canvas. The search bar sits at the top of the left sidebar. Below the search bar are drop-down boxes for different filtering criteria, which include the following: bloom season, plant color, minimum and maximum plant height, soil type, plant type, plant native-ness, and minimum and maximum plant hardiness. The main canvas where the user can draw garden shapes and onto which users can drag and drop plants spans most of the interface to provide more space for editing a garden project. On top of the canvas will be a canvas toolbar that contains the following: tools to draw circular, rectangular, or free-form shapes; a texture picker where users can choose to fill in drawn shapes with grass or various stone textures; options to view the garden in Winter, Spring, Summer, or Fall perspectives; a button the user can press to switch between top and frontal view; a Check garden button the user can press that will cause the Check Garden page to appear as an overlay above the Edit Garden page; and a shopping cart button the user can press to navigate to a pop-up window that prompts the user whether they wish to check their garden before proceeding to the Shopping List page. Clicking “Check garden” will direct the user to the Check Garden page. Clicking “Go to shopping list” will direct the user to the Shopping List page.

**Check Garden page.** (Figure A-7) This page appears as a pop-up window. It will inform the user if the diverse seasons criterion, the critical mass criterion, and the aesthetic criteria have been met with a green checkmark (for criteria met) or a red “X” and a message (for criteria not met). The message will inform the user ways to pass all the criteria. Clicking “Back to garden” will direct the user back to the Edit Garden page. Clicking “Go to shopping list” will direct the user to the Shopping List page.

**Shopping List page.** (Figure A-8) This page shows a checkbox list of each plant in the garden. Each item on the list will contain the plant thumbnail, plant name, and plant quantity. Click on the checkbox next to the plant will place a checkmark in the box (if it was previously unselected) or remove the checkmark (if it was previously selected). This allows the user to indicate whether they have purchased that plant yet. Clicking the envelope icon will send the shopping list as an email to the user. Clicking the printer icon will prompt the user to print to shopping list. Clicking “Back to Edit Garden page” will direct the user back to the Edit Garden page.

**Navigation Bar.** A navigation bar persists in the Edit Garden page and the Shopping List page. From the navigation bar, the user can press “My Garden,” which will direct the user to the My Gardens page from which they can choose a different garden project to open, or the “Logout” button, which will log the user out of the system and direct them back to the Login page.

# **Implementation**

Primary authors: Noel Drotor, Tammy Le

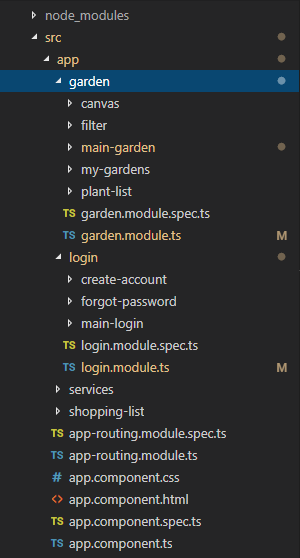
Editor: Tammy Le

The senior design team defined three iterations that lie within the scope of a semester-long project. All three iterations have been completed as of December 14, 2018. Each iteration produced an end-to-end feature of the web application. The iterations are seen in Table 1 below. The following use cases will be implemented by a future senior design team: UC1, UC2, UC8, UC9, and UC10.

**Table 1. Fall Semester 2018 Iterations**

|  |  |  |
| --- | --- | --- |
| **Iteration** | **Description** | **Use Case(s)** |
| 1 | Viewing 1+ garden projects | UC3. Create new garden project  UC4. Open existing garden project |
| 2 | Drag-and-drop plants onto garden | UC5. Edit garden |
| 3 | Search and filter plants | UC6. Search plants  UC7. Filter plants |

Figure 5 below shows the project directory and file structure of the web application. Although the team has not implemented all the requested features of the web application, they have created directories for each of the front-end modules specified in the design: login, garden, and shopping list. Each module directory contains .css, .html, .spec.ts, and .ts files in which the components specified in the design are created. The module directories live inside the pollinator-garden directory. App.js, package.js, and all the .js files for each route in the web application live inside the api directory. The pollinator-garden and api directories live in the root directory of the project. As described in the Resources Needed section, the configuration files that exist for the project are the dependencies required for the application. The Angular web application is connected to the database through Node.js. The team has installed Express, Body-parser, and MySQL packages in Node.js; they are located in the package.json file in the api directory as dependencies.



**Figure 5. A Glimpse of the Project Folder Structure**

Due to time constraints, the team hasn’t implemented the authentication feature of the system (UC1 and UC2); however, the team has included authentication in the front-end designs, back-end designs, and wireframes. REST endpoints and MySQL data tables have been created for future teams to implement authentication.

## **Security Considerations**

There are privacy and security implications the team considered when constructing the requirements and design surrounding authentication, namely the descriptions and wireframes for UC1 and UC2. Additionally, the user table in the MySQL database is designed to only contain the user’s email address and their hashed password to preserve confidentiality and security. To maintain user privacy, the system requires only an email address and password from each user, since other identifying information is irrelevant to the purpose of the application. Although there is only one type of user outlined in the system requirements, to maintain the integrity of the application, the team has implemented an AuthGuard service to protect against unauthorized routing. Only necessary REST calls should be implemented; for example, users should never be able to POST a plant to the database, as the system only reads plant data from the database. Additionally, the team has outlined complex criteria a user’s password must meet to ensure its security; passwords must be at least eight characters and contain at least one of each of the following: number, uppercase letter, lowercase letter, and special characters (!, #, @, $, punctuation characters).

# **Testing**

Primary authors: Carolyn Thompson, Tammy Le

Editor: Tammy Le

## **Acceptance Test Plan**

**Test Plan Executed:** December 7, 2018

**Precondition:** The system shall be loaded with the following plant data before the following tests are executed.

**Table 2. Sample data for black box tests**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name (botanical)** | **Common name** | **Bloom season(s)** | **Color** | **Height** | **Soil type** | **Hardi-**  **ness** | **Native-**  **ness** |
| *Zizia aurea* | Golden alexander | Spring | Yellow | 1.5-3’ | Moist | 3 | Native |
| *Gelsemium sempervirens* | Carolina jessamine | Spring | Yellow | 12-20’ | Moist | 7 | Native |
| *Vaccinium ashei* | Rabbiteye blueberry | Spring | White | 3-12’ | Dry-  moist | 6 | Native |
| *Cosmos bipinnatus* | Cosmos | Summer | Pink | 1-4’ | Moist | 2 | Non-  native |
| *Eryngium yuccifolium* | Rattlesnake master | Summer | White | 4-5’ | Dry-  moist | 3 | Native |
| *Asclepias tuberosa* | Butterfly weed | Summer | Orange | 1-2.5’ | Dry-  moist | 3 | Native |
| *Symphyotrichum oblongifolium* | Aromatic aster | Fall | Yellow | 1-3’ | Dry-  moist | 3 | Native |
| *Eutrochium dubium* | Joe-pye weed | Fall | Purple | 3-4’ | Moist-  wet | 3 | Native |
| *Solidago rugosa* | Rough goldenrod | Fall | Yellow | 2.5-4’ | Moist | 4 | Native |

### **Black Box Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| createAccountValid | **Precondition**: None  1. User clicks sign up button on login page and enters:  Email: janeSmith@gmail.com  Password: saveBees18!  Confirm Password: saveBees18!  2. User clicks “Create account” button. | User account is created and user is navigated back to login page. | Not implemented this semester |
| createAccountInvalidEmail | **Precondition**: None  1. User clicks sign up button on login page and enters:  Email: jsmithgmail.com  Password: saveBees  Confirm Password: saveBees  2. User clicks “Create account” button. | The message “Please enter a valid email address.” is displayed below the email text field. | Not implemented this semester |
| createAccountInvalidPassword | **Precondition**: None  1. User clicks sign up button on login page and enters:  Email: janeSmith@gmail.com  Password: short  Confirm Password: short  2. User clicks “Create account” button. | The message “Password must be at least 8 characters and contain at least one of the following: number, upper-case letter, and special character.”  is displayed below the password text field. | Not implemented this semester |
| createAccountInvalidConfirmedPassword | **Precondition**: None  1. User clicks sign up button on login page and enters:  Email: janeSmith@gmail.com  Password: short  Confirm Password: short1  2. User clicks “Create account” button. | The message “Password must be at least 8 characters and contain at least one of the following: number, upper-case letter, and special character.”  is displayed below the password text field. | Not implemented this semester |
| loginValid | **Precondition**: User Jane Smith has an account.  1. User enters credentials:  Email: janeSmith@gmail.com  Password: saveBees18!  2. User clicks “Login” button. | User is navigated to the My Gardens page. | Not implemented this semester |
| loginInvalidEmail | **Precondition:** User Jane Smith has an account.  Correct email: janeSmith@gmail.com  Correct password: saveBees18!  1. User enters credentials:  email: notJane@gmail.com  Password: saveBees18!  2. User clicks “Login” button. | The message “Incorrect email or password.” is displayed under the username text field. | Not implemented this semester |
| loginInvalidPassword | **Precondition**: User Jane Smith has an account.  Correct email: janeSmith@gmail.com  Correct password: saveBees18!  1. User enters credentials:  email: jsmith@gmail.com  Password: savebees  2. User clicks “Login” button. | The message “Incorrect email or password.” is displayed under the username text field. | Not implemented this semester |
| createNewGardenProject1 | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page.  1. User selects “New garden project.”  2. User enters “Fall garden” as the garden project name and clicks “Create garden.” | The Name Garden Project page appears as a pop-up window.  The My Gardens page overlay will disappear and the Edit Garden page will appear with a blank canvas. | Name Garden window appears as pop-up.  Name Garden and My Gardens pop-up windows disappear and Edit Garden page appears with empty canvas. |
| createNewGardenProject2 | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page.  1. User selects “New garden project.”  2. User enters nothing as the garden project name and clicks “Create garden.” | The Name Garden Project page appears as a pop-up window.  “The project name must be between 1 and 50 characters.” appears on the screen. | Name Garden window appears as pop-up.  The Name Garden window displays “Garden name must be between 1 and 50 characters.” |
| createNewGardenProject3 | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page. She has previously created a project called “Fall Garden.”  1. User selects “New garden project.”  2. User enters “Fall Garden” as the garden project name and clicks “Create project.” | The Name Garden Project page appears as a pop-up window.  “You already created a project with the same name. Please provide a unique project name.” appears on the screen. | Name Garden window appears as pop-up.  The Name Garden window displays “A garden project already exists with the given name. Please enter a new name.” |
| openExistingGardenProject | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page. She has previously created a garden project named “First Garden.”  1. User selects “First Garden” from the list of garden projects. | The My Gardens page overlay will disappear and the Edit Garden page will appear with “First Garden” loaded on the canvas. | Name Garden window appears as pop-up.  Edit Garden page appears with the First Garden canvas loaded onto the screen. |
| searchPlantsNoFilter | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page. She has previously created a garden project named “First Garden.”  1. User types “cosmos” into the search bar. | The list of plants below the search bar will contain the following plant:   * *Cosmos bipinnatus* (Cosmos) | Only Cosmos appears with its plant image in the plant list. |
| searchPlantsFilter | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page. She has previously created a garden project named “First Garden.”  1. User selects the following filter options:  Bloom season: Spring  Bloom color: Yellow  2. User types “jessamine” into the search bar and hits enter. | The list of plants below the search bar will contain the following plants:   * *Zizia aurea* (Golden alexander) * *Geisemium sempervirens* (Carolina jessamine)   The list of plants below the search bar will contain the following plant:   * *Geisemium sempervirens* (Carolina jessamine) | After selecting filter options Spring and Yellow, Carolina jessamine and Golden Alexander appear in the plant list.  After searching, only Carolina jessamine appears in the plant list. |
| searchPlantsNoResult | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. She is directed to the My Gardens page. She has previously created a garden project named “First Garden.”  1. User selects the following filter options:  Bloom season: Spring  Bloom color: Yellow  2. User types “goldenrod” into the search bar and hits enter. | The list of plants below the search bar will contain the following plants:   * *Zizia aurea* (Golden alexander) * *Geisemium sempervirens* (Carolina jessamine)   The message “No plants match your search.” appears in the area below the search bar and filter drop-down boxes. | After selecting filter options Spring and Yellow, Carolina jessamine and Golden Alexander appear in the plant list.  Nothing shows up in the plant list. |
| filterPlants1 | **Precondition**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User selects the following from the drop-down options:  Bloom color: Yellow  Bloom Season: Fall | She sees the list of filtered plants below the search bar, which contains the following:   * *Solidago rugosa* (Rough goldenrod) | Only Rough goldenrod appears in the plant list. |
| filterPlants2 | **Precondition**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User selects the following filtering options:  Bloom color: White  Bloom Season: Summer  Height: 0 - 5 feet  Soil type: Moist | She sees the list of filtered plants below the search bar, which contains the following:   * *Eryngium yuccifolium* (Rattlesnake master) | Only Rattlesnake master appears in the plant list. |
| filterPlants3 | **Precondition**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User selects the following from the drop-down options:  Bloom color: Orange  Bloom Season: Spring | The message “No plants meet your filter criteria.” appears in the area below the search bar and filter drop-down boxes. | No plants appear in the plant list. |
| editGarden1 | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User drags and drops:  *Solidago rugosas* (Rough goldenrod)  within the boundaries of Canvas area. | The Rough goldenrod plant appears on the Canvas. | Rough goldenrod appears on the canvas where it was dropped. |
| editGarden2 | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User draws a circular area in canvas area using circle drawing tool on the canvas toolbar. | Circular area appears in canvas area. | Not implemented this semester. |
| editGarden3 | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User draws grass texture within Canvas area using texture brush found on the canvas toolbar. | Drawn grass texture appears in canvas area. | Not implemented this semester. |
| editGarden\_deletePlant | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  1. User drags and drops:  *Solidago rugosas* (Rough goldenrood)  within boundaries of Canvas area.  2. User double clicks on the Rough goldenrodplant on the canvas to delete it. | The Rough goldenrod plant appears on the Canvas.  The Rough goldenrod plant disappears from the Canvas. | Rough goldenrod appears on the canvas where it was dropped.  The plant disappears from the canvas. |
| changePerspectiveFrontal | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page.  User is currently in top-down view.  1. User drags and drops:  *Solidago rugosas* (Rough goldenrood)  within boundaries of Canvas area.  2. User clicks view icon on canvas toolbar to change canvas to frontal view. | The *Solidago rugosas* plant appears on the Canvas.  Entire canvas area changes perspective to front view. The image of the *Solidago rugosas* plant switches to its front view counterpart. | Not implemented this semester. |
| changePerspectiveTop | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. She has opened a new garden project named “First Garden” and is directed to the Edit Garden page. User is currently in frontal view.  1. User drags and drops:  *Solidago rugosas* (goldenrood)  within boundaries of Canvas area.  2. User clicks view icon on canvas toolbar to change canvas to top-down view. | The *Solidago rugosas* plant appears on the Canvas.  Entire canvas area changes perspective to top-down view. The image of the *Solidago rugosas* plant switches to its front view counterpart. | Not implemented this semester. |
| changeSeasonSpring | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Second Garden” that contains 3 *Zizia aurea* (Golden alexanders) plants. Spring is not selected for season view.  1. User clicks Spring icon on the canvas toolbar for viewing garden in Spring. | Images of 3 *Zizia aurea* plantsin Canvas area are switched with their Spring images. | Not implemented this semester. |
| changeSeasonSummer | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Second Garden” that contains 3 *Zizia aurea* (Golden alexanders) plants. Summer is not selected for season view.  1. User clicks Summer icon on the canvas toolbar for viewing garden in Summer. | Images of 3 *Zizia aurea* plantsin Canvas area are switched with their Summer images. | Not implemented this semester. |
| changeSeasonFall | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Second Garden” that contains 3 *Zizia aurea* (Golden alexanders) plants. Fall is not selected for season view.  1. User clicks Fall icon on the canvas toolbar for viewing garden in Fall. | Images of 3 *Zizia aurea* plantsin Canvas area are switched with their Fall images. | Not implemented this semester. |
| checkGardenAllCriteriaMet | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 3 *Zizia aurea* (Golden alexanders), 2 *Eryngium yuccifolium* (Rattlesnake master), and 4 *Symphiotrichum oblongifolium* (Aromatic aster) plants. | The Check Garden page will appear as a pop-up window and say the following:  “Diverse seasons criterion is met.  Critical mass criterion is met.  Aesthetic criteria are met.” | Not implemented this semester. |
| checkGardenSeasonsNotMet | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 3 *Zizia aurea* (Golden alexanders) and 2 *Vaccinium ashei* (Rabbiteye blueberry) plants.  1. User clicks “Check garden” button on the canvas toolbar. | The Check Garden page will appear as a pop-up window and say the following:  “Your garden has not met the diverse seasons criterion. Make sure your garden contains different plants that can bloom in different seasons.  Critical mass criterion is met.  Aesthetic criteria is met.” | Not implemented this semester. |
| checkGardenCriticalMassNotMet | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 1 *Zizia aurea* (Golden alexanders) and 2 *Vaccinium ashei* (Rabbiteye blueberry) plants.  1. User clicks “Check garden” button on the canvas toolbar. | The Check Garden page will appear as a pop-up window and say the following:  “Diverse seasons criterion is met.  Your garden has not met the critical mass criterion. Make sure you have more than one of each plant in your garden.  Aesthetic criteria is met.” | Not implemented this semester. |
| checkGardenAestheticNotMet | **Preconditions**: User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 3 *Zizia aurea* (Golden alexanders) and 2 *Solidago rugosa* (Rough goldenrood) plants.  1. User clicks “Check garden” button on the canvas toolbar. | “Diverse seasons criterion is met.  Critical mass criterion is met.  Your garden is lacking variety in plant color. Try adding plants of different colors to your garden.” | Not implemented this semester. |
| shoppingList | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 3 *Zizia aurea* (Golden alexanders), 3 *Cosmos bipinnatus* (Cosmos), and 2 *Solidago rugosa* (Rough goldenrood) plants.  1. The user clicks the shopping list button at the top right of the canvas. | User is navigated to Shopping List page.  The shopping list shows the following:   * *Zizia aurea* with a quantity value of 3 * *Cosmos bippinnatus* with a quantity of 3 * *Solidago rugosa* with a quantity of 2 | Not implemented this semester. |
| emailShoppingList | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 3 *Zizia aurea* (Golden alexanders), 3 *Cosmos bipinnatus* (Cosmos), and 2 *Solidago rugosa* (Rough goldenrood) plants.  1. The user clicks the envelope/mail icon. | The shopping list is emailed to janeSmith@gmail.com. | Not implemented this semester. |
| printShoppingList | **Preconditions:** User Jane Smith has an account and has successfully logged into the system. User has created a garden project named “Jane’s Garden” that contains 3 *Zizia aurea* (Golden alexanders), 3 *Cosmos bipinnatus* (Cosmos), and 2 *Solidago rugosa* (Rough goldenrood) plants.  1. The user clicks the printer icon. | A separate webpage with printing options opens. The plant list appears in the page with the following:   * *Zizia aurea* with a quantity value of 3 * *Cosmos bippinnatus* with a quantity of 3 * *Solidago rugosa* with a quantity of 2 | Not implemented this semester. |

## **Unit Test Plan**

### **Unit Testing Tools**

*Jasmine*

* Jasmine is a known testing tool that works well with Angular. Jasmine attempts to describe tests in a human readable format so that non-technical people can understand what is being tested.

*Postman*

* Postman is an API development environment that includes a mock server for development and automated testing of REST APIs. Postman offers an interface to run tests for behavior and performance, and it generates test reports for analysis.

### **Coverage Tool**

*Karma*

* Karma is used for both unit testing and as a coverage tool. Karma can launch and connect to a specific set of web browsers, run tests, and then gather the report.

### **Manual Tests**

Since the app is visualization heavy, there are aspects the team cannot unit test. At this point in time, the team was not able to automate testing for the following:

* Canvas visualization loading
* Dragging and dropping plants into canvas area

These actions have been tested manually to check desired functionality.

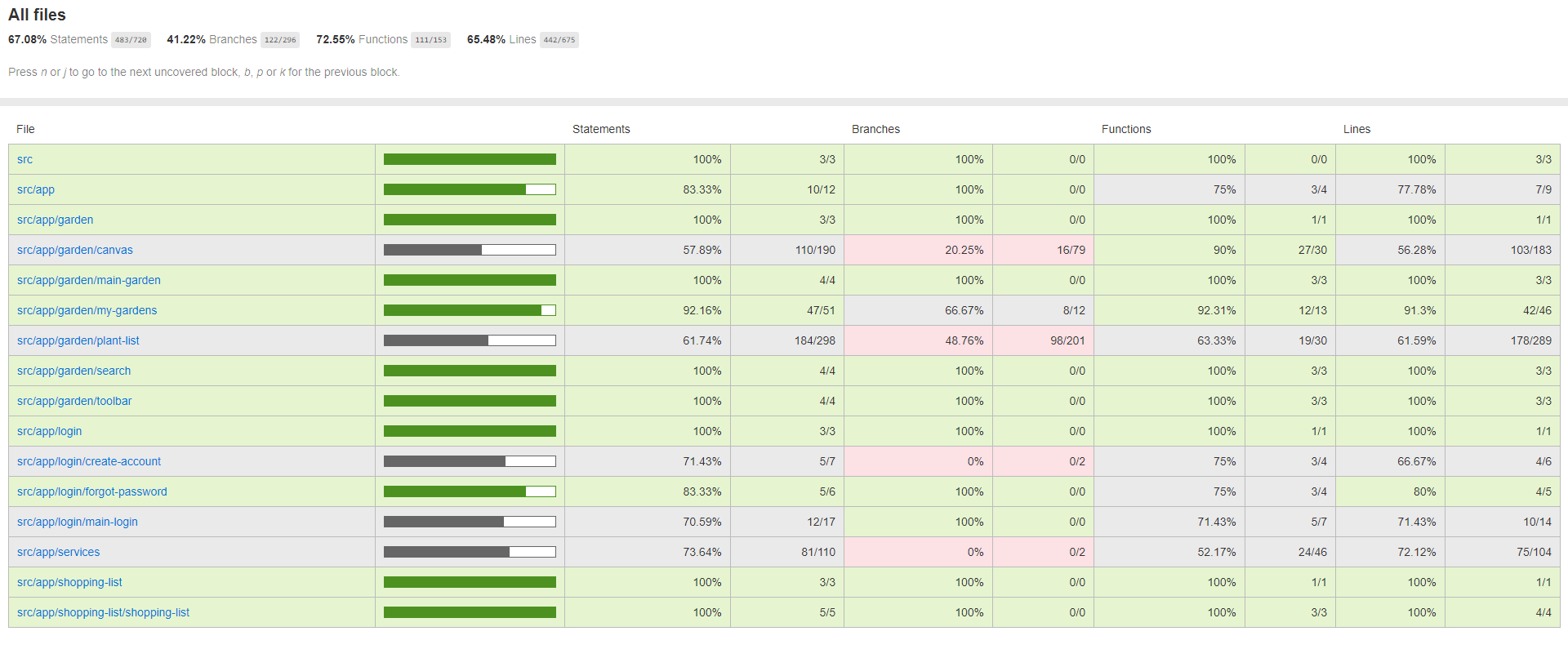
### **Overall Coverage Estimates**

Writing thoughtful tests is a main priority, and an estimated 80% minimum coverage may fall out of that priority. Better tests and better coverage are always the goal, but the 80% is set to establish a standard for the team.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **class, %** | **method, %** | **block, %** | **line, %** |
| all classes | 80% | 80% | 80% | 80% |

### **Unit Testing Results**

Figure 6 below shows the results of the white box unit testing done with the Karma/Jasmine suite. As discussed above, our goal was to achieve around 80% coverage to ensure good testing practices and thoroughness. However, limitations of the testing framework prevented the team from testing event listener functions, such as mouse move and click functions, which were used extensively when programming our canvas functionality. Due to this restriction, the team were unable to reach the desired coverage metric for the canvas and plant-list components.

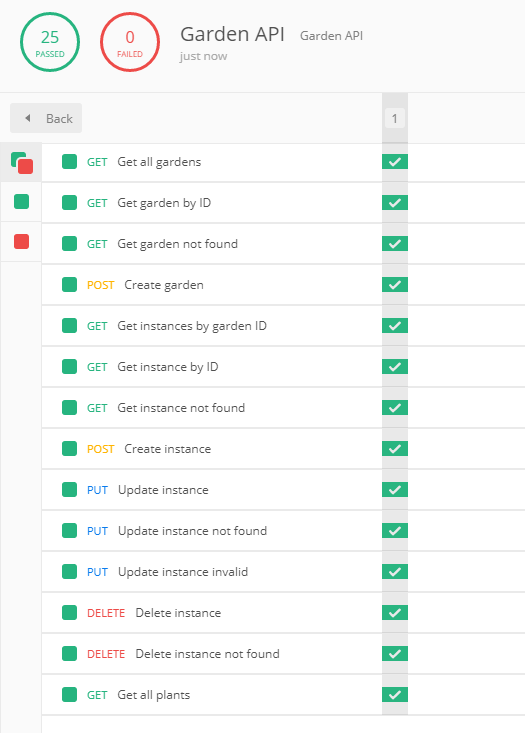
****

**Figure 6. Karma/Jasmine Unit Test Coverage results**

Aside from this, the Jasmine testing framework provided an easy method of mocking component dependencies, allowing each component to be rendered and tested individually, without loading the entire app for each test. Combined with our manual black box testing and API testing using Postman, the team was able to adequately ensure the functionality of features implemented this semester.

### **API** **Automated Test Results**

Figure 7 below shows the results of the automated API tests written using the JavaScript testing feature of Postman. These tests were saved into a collection, which then runs each call in sequence. For each call, all possible branches were considered, which includes returning different HTTP statuses based on the validity of the call and the data returned.



**Figure 7. Postman Automated API Test Results**

## **Usability Test Plan**

In addition to testing the functionality of the system, the team tested how easily users can interact with the system’s interfaces and complete tasks. The team and the sponsor recruited students and non-students from the senior design class in CSC and DAE. The goal was to recruit between five and seven individuals with varying backgrounds (i.e. students, non-students, garden enthusiasts). The pre-test survey, post-test survey, and testing script the team developed can be found in Appendix B. Table 4 below shows the timeline for preparing and conducting usability tests.

**Table 4. Usability Test Timeline**

|  |  |
| --- | --- |
| **Week** | **Task(s)** |
| October 1 - 7 | * Sponsor to recruit students and non-students from DAE for usability test * Team to recruit students and technical advisors from CSC Senior Design (with Ms. Heil’s assistance) for usability test |
| October 8 - 14 | * Aim to create a minimum complete end-to-end front-end implementation for each wireframe for use during usability testing * Print paper wireframes for usability testing to supplement the use of the minimum front-end implementation |
| October 15 - 19 | * Monday (10/15) and Wednesday (10/17) - conduct usability tests with participants from CSC Senior Design (both students and/or faculty) * Friday (10/19) - conduct usability tests with participants from AE |
| October 22 | * Analyze results from usability tests and make design changes and/or justifications (if any) |

### **Usability Test Results**

The team conducted usability tests with seven participants with varying technical and gardening backgrounds. None of the participants had experience using a garden design tool, and only one had taken care of a pollinator garden previously. The team asked participants to talk aloud while completing the usability tasks; this provided additional insight to the participants’ thought process when interacting with the system. There were several common findings in the participants’ responses; the findings and the team’s response to these findings are found in Table 5 below:

|  |  |
| --- | --- |
| **Finding** | **Team’s Response** |
| Bloom season filter option often confused for the season view buttons. The icons used for the different seasons were hard for some participants to understand; specifically, the participants with a background in Applied Ecology and pollinator gardens interpreted the season icons as “Moisture” and “Sunlight” settings for the garden. | “Filter Options” heading has been added above the filtering options. The season view buttons have been changed to a drop-down box with text. |
| Participants knew to select the cube icon when asked how to change the perspective of the icon from front to top-down view; however, further conversation showed that many participants selected the icon out of a process of elimination. They also discussed that they would mostly use trial and error to determine a button’s functionality due to it being a web application. When asked for an icon that might make more sense, participants couldn’t think of anything significantly better; some suggested the icon Google Maps uses to change perspective. One even suggested dragging the whole canvas. | After discussion with Dr. Youngsteadt, the team concluded that the cube icon works well in the design and will not be changed at this time. |
| Participants understood the drawing tools in the canvas (shape tools, brush tool); however, most did not understand the texture box. Similar to the perspective icon, participants said they would probably play around with the different buttons if it were an actual application. They used their knowledge of the Paint application found on Windows computers as a basis of understanding. | The team recognizes that the lack of color and static design on the printed prototypes hindered the participants’ understanding of this feature. The team decided not to change this icon because it is still the best icon the team could think of that represents this functionality. |
| Participants confused the Shopping List for a Shopping Cart. When asked how to check off items in the list they had already purchased, many were confused because the shopping cart icon had suggested to them that they were preparing to purchase the plants from the application. Some users questioned whether “checking a plant off the list” would remove the plant from the garden. | The team changed all references to the Shopping List to be consistent (i.e. use only the phrase “Shopping List,” not “Shopping Cart”). The icon to navigate to the shopping list has been changed from a shopping cart icon to a list icon on the Edit Garden page. Additionally, instead of using checkboxes to indicate a plant has been purchased, the team used textual buttons that read “Already Bought” to indicate the plant has been purchased and “Remove” to indicate the plant should be removed from the garden design. |
| “Critical mass” was a confusing term to participants, especially those with a background in pollinator gardens and Applied Ecology. Additionally, some participants suggested additional criteria that the Check Garden functionality could incorporate. | After discussing the feedback on Check Garden criteria with Dr. Youngsteadt, the team decided to mainly stick with the criteria she had defined to start while the application is still in its early stages of development. Many of the suggestions would require additional fields in the database. Dr, Youngsteadt has requested for additional time to re-define the criteria and will postpone the updates for this requirement to next semester. |

Additionally, the team discussed the following additions to the system requirements and functionality with Dr. Youngsteadt based on feedback from the usability tests:

* When the user hovers over a plant in the plant list, additional photographs of the plant and plant details will appear in a box.
* When the user selects a plant that has been added to the canvas, the user will see additional photographs of the plant and plant details; additionally, the user will be able to add comments to that particular plant (for their own use when referring to the garden design or during installation).
* The Shopping List page will have two sections: the list of plants the user needs to purchase or attain for the garden and a comments section where any comments the might have made on plants in the Edit Garden page will populate for reference.

Overall, the participants found the web application easy to use and enjoyable. Most of the participants found it easy to understand how to draw garden spaces, how to add plants, how to filter and search plants, and how to change season or perspective views. The complete set of responses, represented as graphs and tables, can be found in Appendix B.

# **Suggestions for Future Teams**

Primary author: Carolyn Thompson

Editor: Tammy Le

The senior design team recommends completing Angular’s tutorials to become familiar with the module and component structure of the project. The supplemental guides will help with accelerated understanding of the project as well. Additional usability testing is important for this web application due to its very interactive and visual nature, and it is likely that it can be conducted using the actual web application rather than wireframes during the next team’s semester.

Additionally, the team recommends keeping in touch with the CALS IT team. Contact information can be found below along with helpful documentation that should be referenced and updated as progress of changes are made throughout the semester:

* [CALS IT Informative Document](https://docs.google.com/document/d/1yQBV1NLkMWYY4NkuCYrGTVlvtA_JA-cN0rOGl-aGvMo/edit?usp=sharing)
* CALS IT Contacts:
  + Rebecca O'Connell, rloconne@ncsu.edu
  + Scott Niven, niven@ncsu.edu
  + Jamie Dennis, jddennis@ncsu.edu
  + Scott McCollum, mccollum@ncsu.edu
  + Rob Ladd, rcladd@ncsu.edu

Future features that are important to the sponsor include the ability to view the garden in multiple perspectives (UC8), viewing the garden in different seasons (UC8), checking the garden against criteria for success (UC9), generating a shopping list (UC10) and successfully creating a user account and logging in (UC1 & UC2).

Additional features that may be of interest include the ability to leave comments on specific plants on the Edit Garden page, downloading an image of the garden from the canvas, creating an educational guide about pollinator gardening for users to access in the web application, and showing users the species of pollinators their garden will attract based on the plants they have added.

## **Team Contact Information**

Tammy Le, Team Leader, thle@ncsu.edu, 610-312-4578

Kevin Bartushak, Development Manager, [kabartus@ncsu.edu](mailto:kabartus@ncsu.edu), 919-896-5617

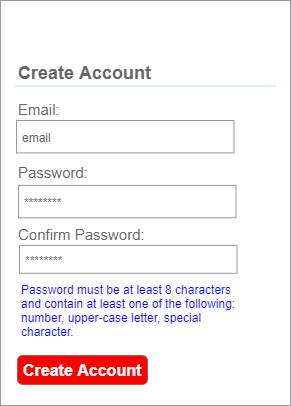
Noel Drotor, Quality/Process Manager, [cndrotor@ncsu.edu](mailto:cndrotor@ncsu.edu), 919-455-6365

Carolyn Thompson, Planning Manager, [cmthomp7@ncsu.edu](mailto:cmthomp7@ncsu.edu), 910-584-2092

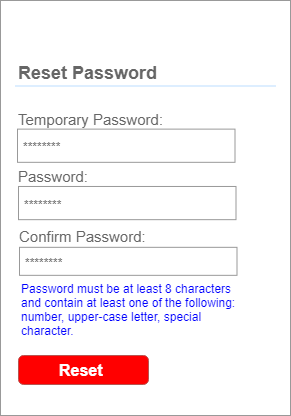
# **Appendix A: Wireframes**

# 

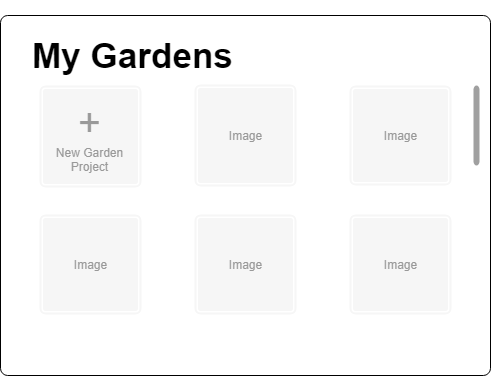
**Figure A-1. Login wireframe**

****

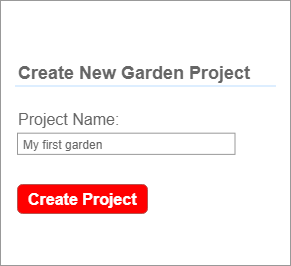
**Figure A-2. Create Account wireframe**

****

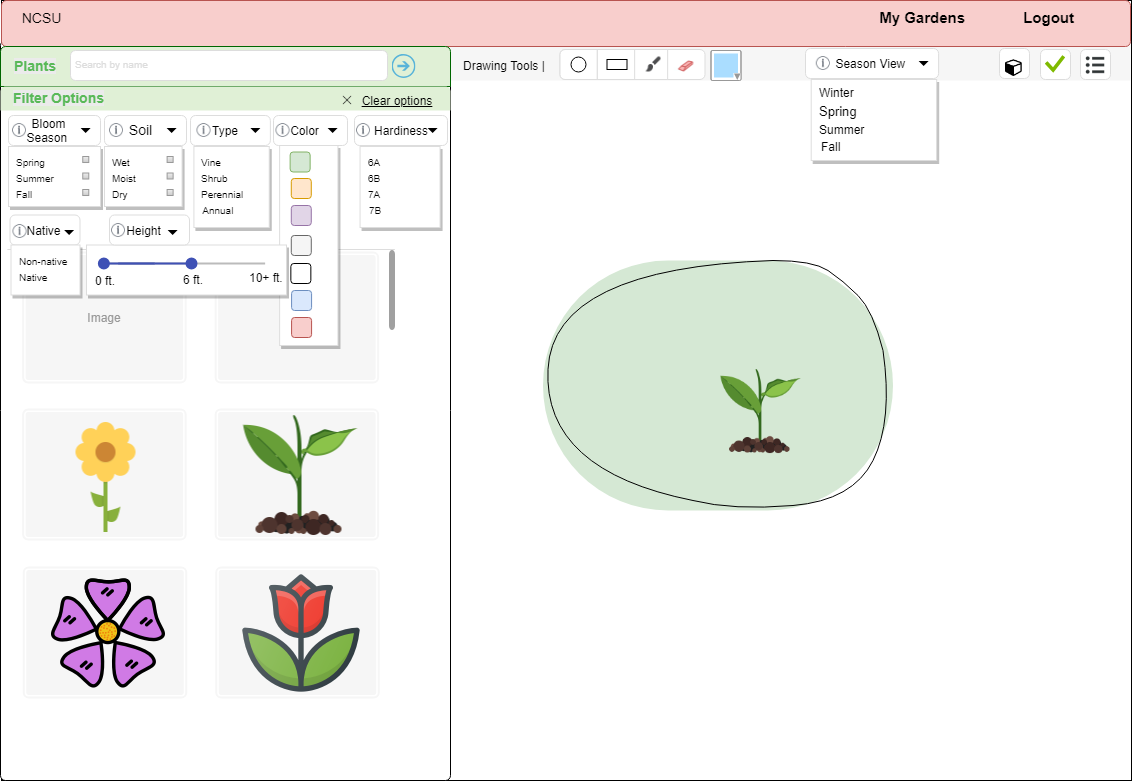
**Figure A-3. Reset Password wireframe**

****

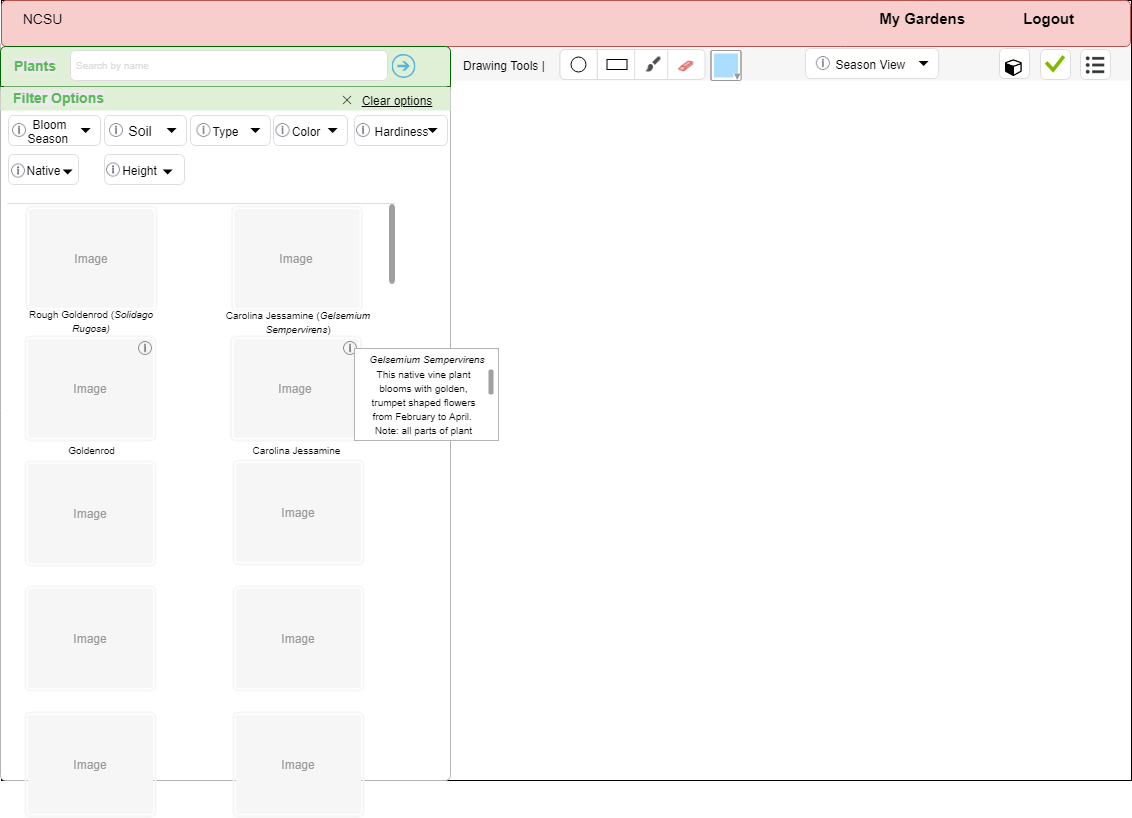
**Figure A-4. My Gardens wireframe**

****

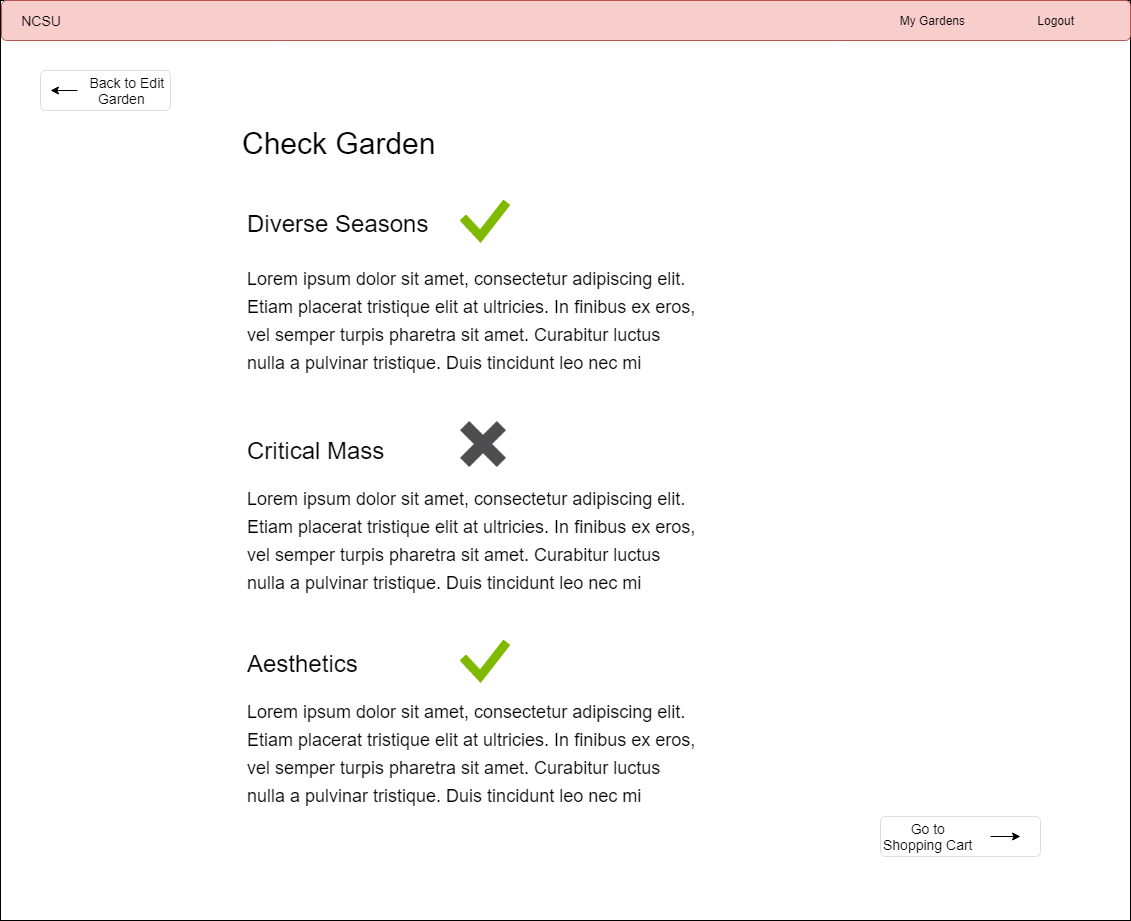
**Figure A-5. Name Garden Project wireframe**

****

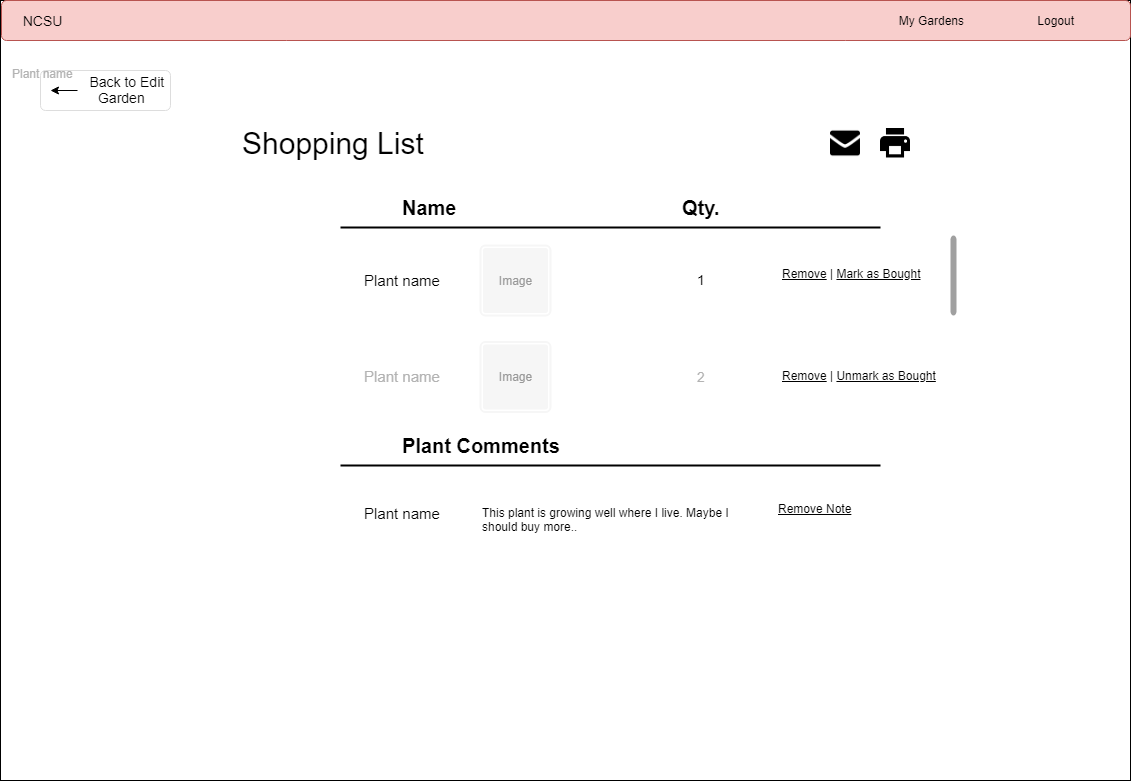
**Figure A-6a. Edit Garden wireframe**

****

**Figure A-6b. Edit Garden wireframe with name info**

****

**Figure A-7. Check Garden wireframe**

****

**Figure A-8. Shopping List wireframe**

# **Appendix B: Usability Test Artifacts**

|  |
| --- |
| Project Title: Pollinator Garden Planner  In this usability test, you will be asked to perform tasks using paper wireframes and to talk aloud while completing tasks. Participation in this study is voluntary.  *What is the purpose of this usability test?*  The purpose of the usability test is to assess if the Pollinator Garden Planner web application and interfaces are meeting the needs of users. Data collected will be used to improve our app design to create a better experience for users.  *Confidentiality*  The information in the study records will be kept confidential to the full extent allowed by law. No reference will be made in oral or written reports which could link you to the study.  I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Consent to Participate: “I have read and understand the above information. I agree to participate in this study with the understanding that I may choose not to participate or to stop participating at any time without penalty.”   1. Select one of the following:    1. I consent to participate in this study.    2. I do not consent to participate in this study.   *Note: If the participant does not consent to participating, the form will be submitted, and the testing will end. Otherwise, the survey will continue below.*  Thank you for participating in this usability study for our Pollinator Garden Planner web app. In the following survey, you will be asked some questions about your level of experience with gardening. This information will not be personally identifiable to you.  The following definitions are provided for your reference.  Pollinator Garden: A pollinator garden is planted and designed, with specific nectar and pollen producing plants, in a way that attracts pollinating organisms known as pollinators.  Pollinator: A pollinator is an animal that helps carry pollen from the male part of the flower (stamen) to the female part of the same or another flower (stigma). The movement of pollen must occur for the plant to become fertilized and produce fruits, seeds, and young plants. Pollinators include bees, wasps, moths, butterflies, birds, flies, bats and more.  The following question is a short-answer question.   1. Please describe your experience with gardening.   The following questions are YES/NO questions.   1. Have you used a garden design app before? 2. Have you taken care of a pollinator garden? |

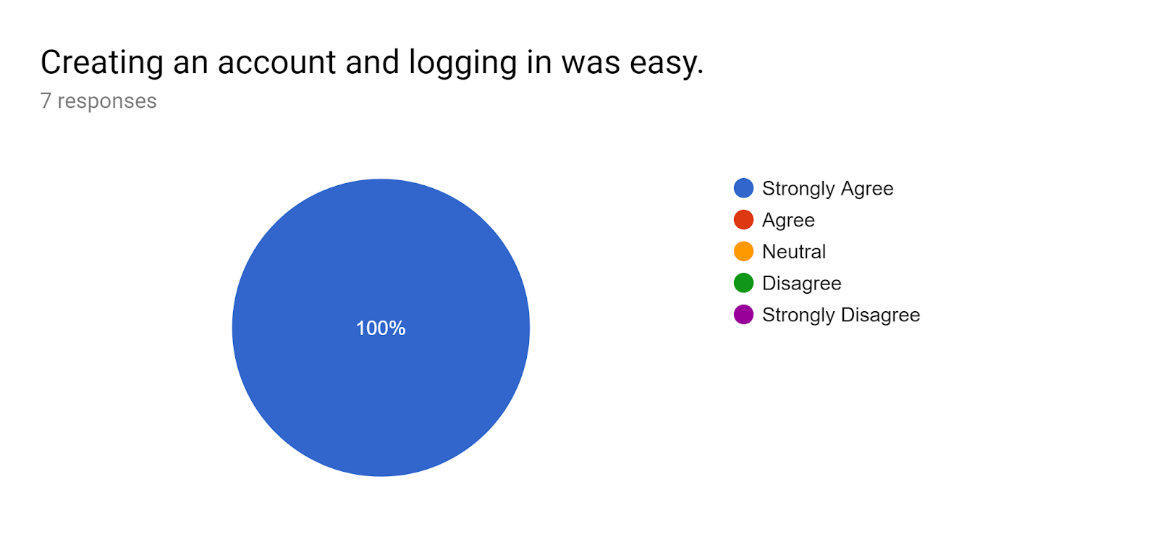
**Figure B-1. Pre-Test Survey (includes consent form)**

|  |
| --- |
| Select Strongly Agree, Agree, Neutral, Disagree, or Strongly Disagree for the following questions.   1. Creating an account and logging in was easy. 2. Creating a new garden project was easy. 3. Drawing a garden area was easy. 4. The garden drawing tools were easy to understand. 5. It was easy to understand how to add plants to the garden. 6. The plant filtering options were easy to understand. 7. The plant filtering options were easy to use. 8. Searching for specific plants was easy. 9. Viewing the garden in a different season was easy. 10. Viewing the garden from a different perspective (top-down or side view) was easy. 11. It was easy to understand how to check the garden before generating the shopping list. 12. It was easy to understand how to generate a shopping list. 13. It was easy to understand how to email or print the shopping list. 14. The “My Gardens” button on the navigation bar was easy to understand. 15. I would use the Pollinator Garden Planner. 16. I enjoyed the overall experience of using the Pollinator Garden Planner.   The following are short-answer questions.   1. What did you like about the Pollinator Garden Planner? (Think about specific features or the ease of use that improved your experience with the application.) 2. What could be improved in the application? |

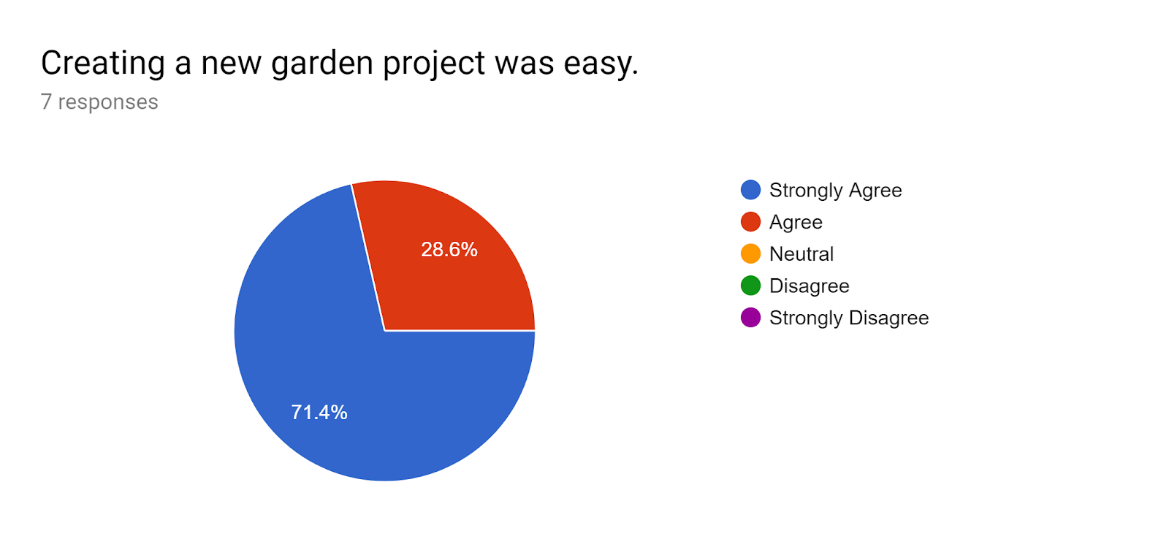
**Figure B-2. Post-Test Survey**

|  |
| --- |
| **Introduction**  Hi \_\_\_\_\_, thank you again for taking the time to participate in this study. Before we begin, I will give you a brief overview of the test and how it will work.  This session will be straightforward — I’ll be giving you a broad task to complete and then asking questions as we go along. Before I tell you the task, I’ll be giving you a little bit of context behind it, such as why you might be doing it and what you hope to achieve.  It’s important to know that we are only testing the site, not you. You can’t do or say anything wrong here. Please feel free to let me know at any time if there’s something you like, dislike, if you’re confused, etc.  Also, I’d like you to “think aloud” as much as possible. By that, I mean that I’d like you to speak your thoughts as often as you can. For example, you may be looking at a page, suddenly see something you didn’t see before and want to click on it. In that case, saying something like “this caught my eye so I’m going to see what it is” would be very useful.  If at any point you have questions, please don’t hesitate to ask. You may also choose to end the study at any time. Do you have any questions so far?  Ok, Let’s get started.  Let’s begin with some background information about our web app, Pollinator Garden Planner.  Explanation of app: Our web app’s goal is to allow users to design pollinator gardens that are both successful and beautiful. A pollinator garden is a garden that attracts animals that transfer pollen, such as bees, pollen wasps, hummingbirds, etc.  Background Scenario: You have access to a web app called “Pollinator Garden Planner.” You want to design a pollinator garden for your home and you also want to create a shopping list of plants you used in your design. You do not have an account with the web app yet.  **Task 1:** How would you login to the web app? [after] Does the login/create account process make sense?  Expected Actions:   1. Click “Create Account” 2. Input username, email, password, confirm password 3. Click “Sign Up” 4. Enter email and password 5. Click “Login”   **Task 2:** How would you create a new garden project? [after] Was that thumbnail easy to find and understandable?  Expected Actions:   1. Click thumbnail with plus symbol in the middle   **Task 3:** Let’s say you want to draw a garden area. How would you do that? [after] Are these icons easy to understand?  Expected Actions:   1. Click any of these tools: circle, rectangle, freehand draw, texture painter   **Task 4:** How you would you add plants to your garden? [after] Is that process understandable?  Expected Actions:   1. Click and drag a plant thumbnail into the garden canvas area   **Task 5:** How would you make plants with certain qualities appear? [after] Are these filter options easy to use? What filter option(s) are relevant for you?  Expected Actions:   1. Change any of the filter options: bloom season, height, color, type, soil, hardiness, native   **Task 6:** Let’s say you have a specific plant you want to add to your garden. How would you look for that plant?  Expected Actions:   1. Type plant name in search bar at the top of the plant thumbnail area   **Task 7:** Let’s say you want to see how your garden looks in a different season. How would you change the season view of your garden? [after] Is it easy to understand what these icons mean?  Expected Actions:   1. Click any one of the season icons: snowflake, green leaf, sun, orange leaf.   **Task 8:** Let’s say you want to change the perspective view of your garden from top-down to a side view. How would you change that perspective? [after] Is that icon easy to locate and understand?  Expected Actions:   1. Click 3D cube near save button in top right corner of dashboard   **Task 9:** Let’s say you want a shopping list with all the plants you’ve add to your garden design. How would you generate a shopping list? [after] Is that icon easy to understand?  Expected Actions:   1. Click shopping cart icon in top right corner of dashboard   **Task 10:** Now you have the option for the webapp to check your garden before proceeding to the shopping list. Let’s say you want to check your garden. How would you do that? [after] Would you want more information on what “check my garden” means to make this decision?  Expected Actions:   1. Click “Check My Garden”   **Task 11:** From this check garden page, how would you proceed to the shopping list?  Expected Actions:   1. Click “Go to Shopping List”   **Task 12:** Let’s say you have some of the plants on your shopping list. How would you indicate that you already have them?  Expected Actions:   1. Click any number of checkboxes to the left of the listed plant(s)   **Task 13:** How would you email your shopping list to yourself? [after] Is this icon easy to understand?  Expected Actions:   1. Click envelope icon   **Task 14:** How would you print your shopping list? [after] Is this icon easy to understand?  Expected Actions:   1. Click printer icon   **Task 15:** Let’s say you want to create another garden. How would you do that from here? [after] Is it clear what the button “My Gardens” means?  Expected Actions:   1. Click “My Gardens” at the top right corner of the screen |

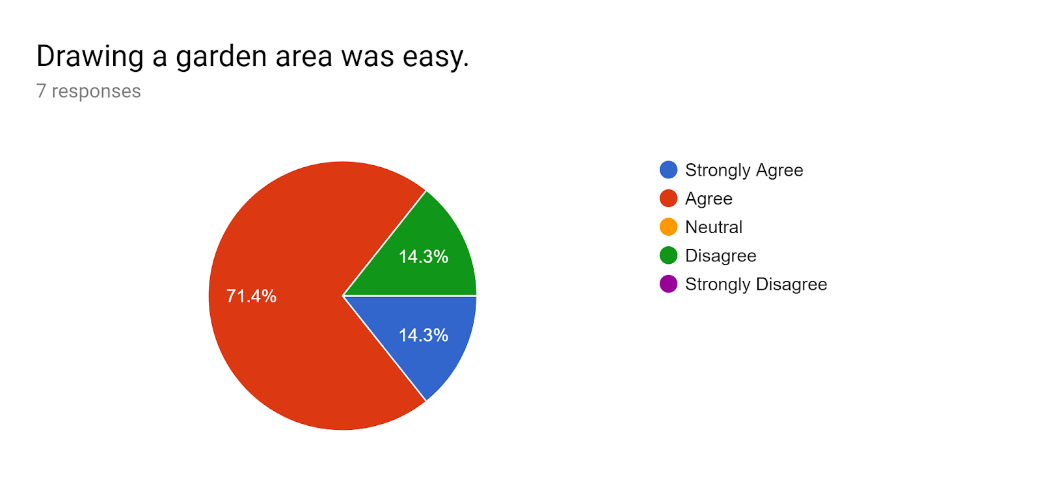
**Figure B-3. Usability Test Script**



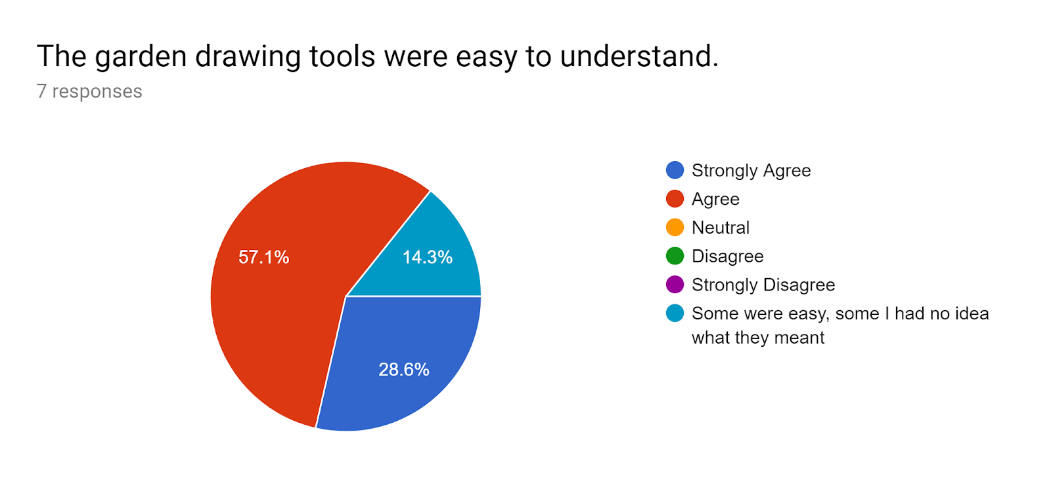
**Figure B-4. Participant responses to “Creating an account and logging in was easy.”**



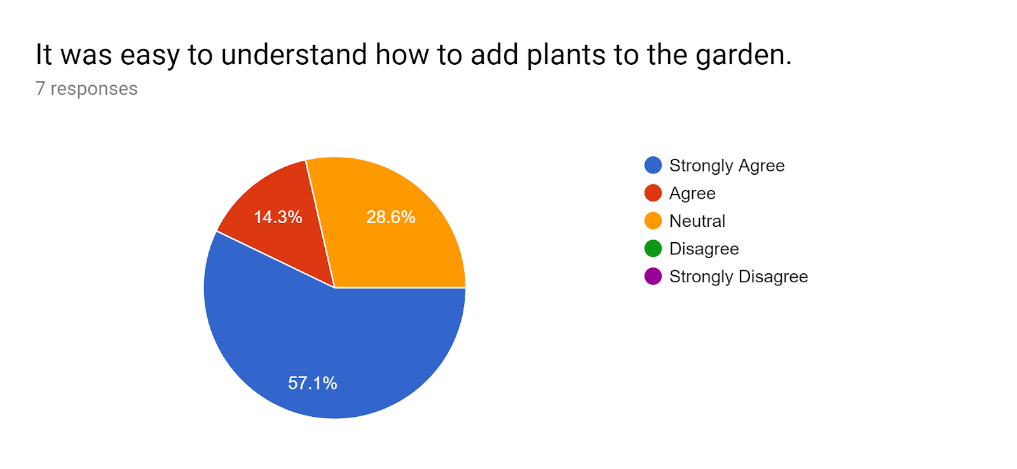
**Figure B-5. Participant responses to “Creating a new garden project was easy.”**



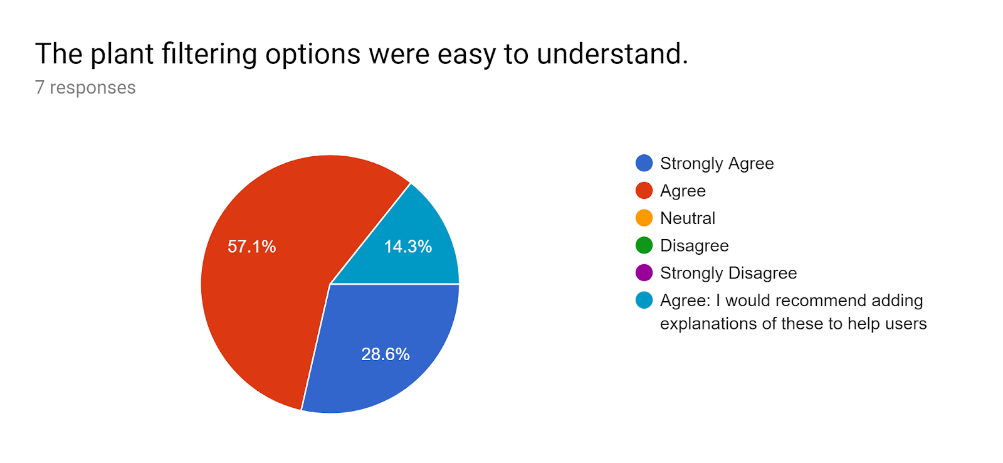
**Figure B-6. Participant responses to “Drawing a garden area was easy.”**



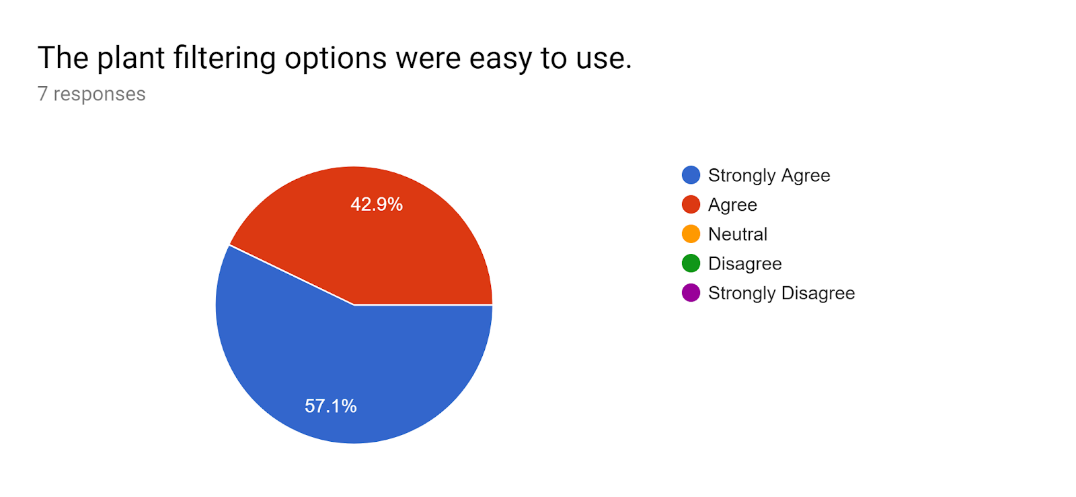
**Figure B-7. Participant responses to “The garden drawing tools were easy to understand.”**



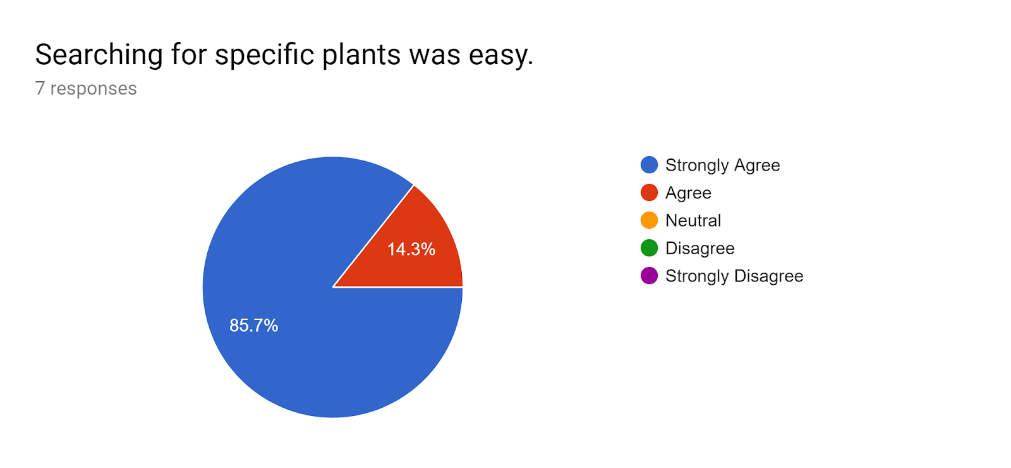
**Figure B-8. Participant responses to “It was easy to understand how to add plants to the garden.”**



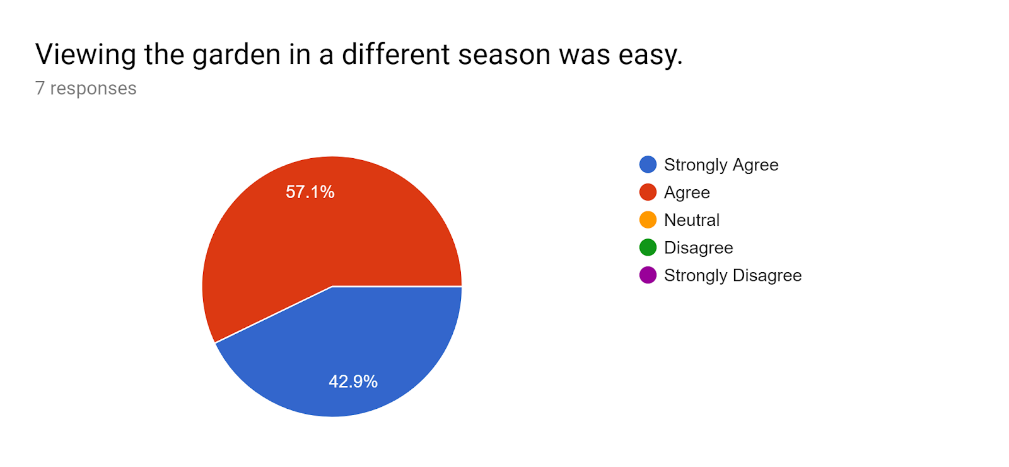
**Figure B-9. Participant responses to “The plant filtering options were easy to understand.”**



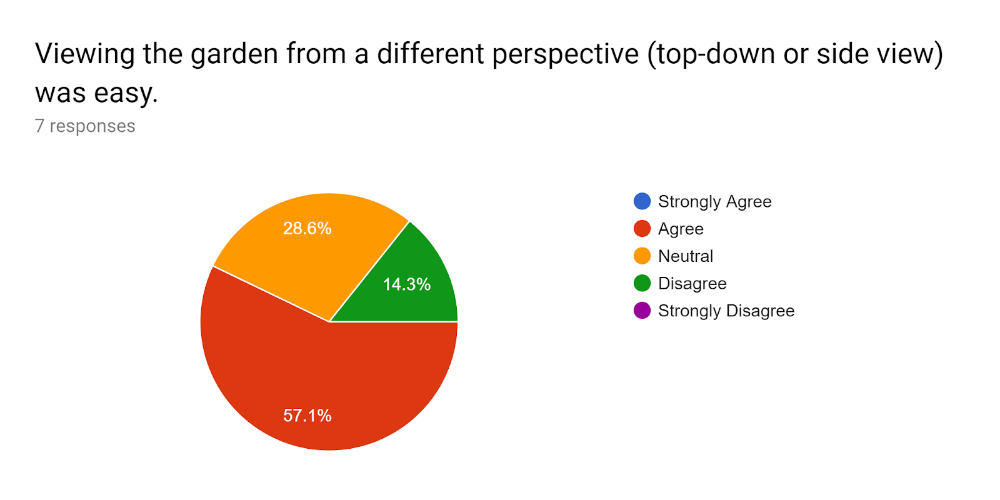
**Figure B-10. Participant responses to “The plant filtering options were easy to use.”**



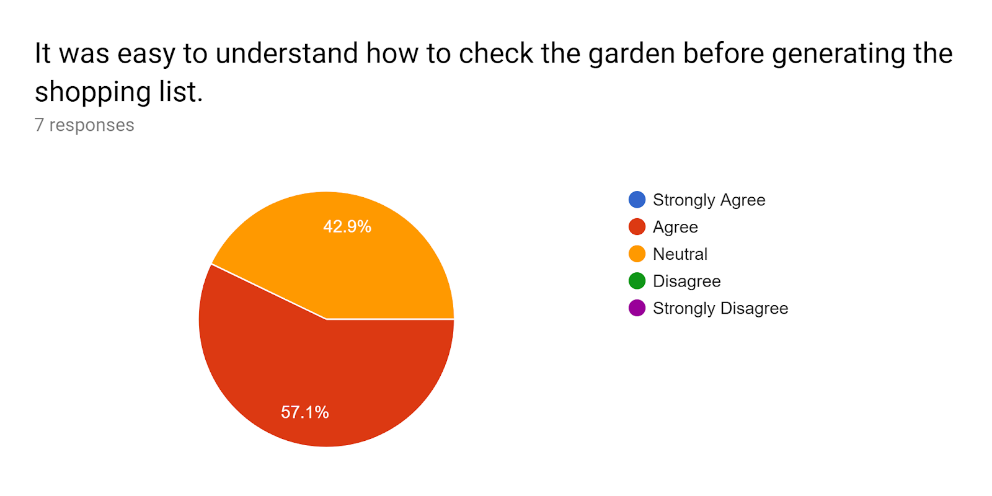
**Figure B-11. Participant responses to “Searching for specific plants was easy.”**



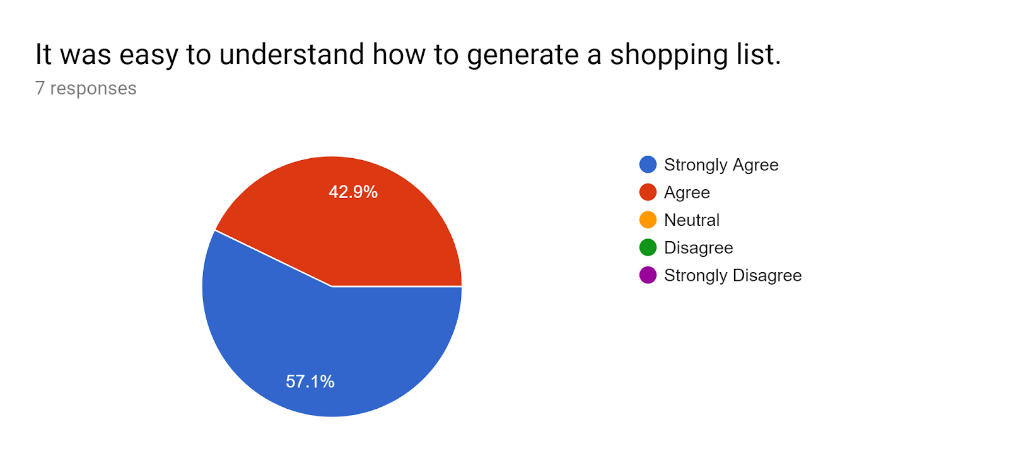
**Figure B-12. Participant responses to “Viewing the garden in a different season was easy.”**



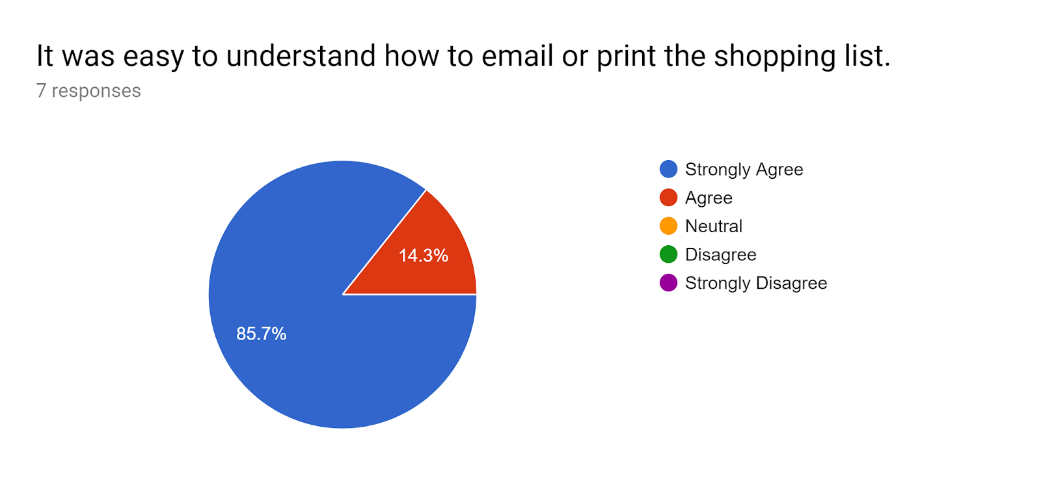
**Figure B-13. Participant responses to “Viewing the garden from a different perspective (top-down or side view) was easy.”**



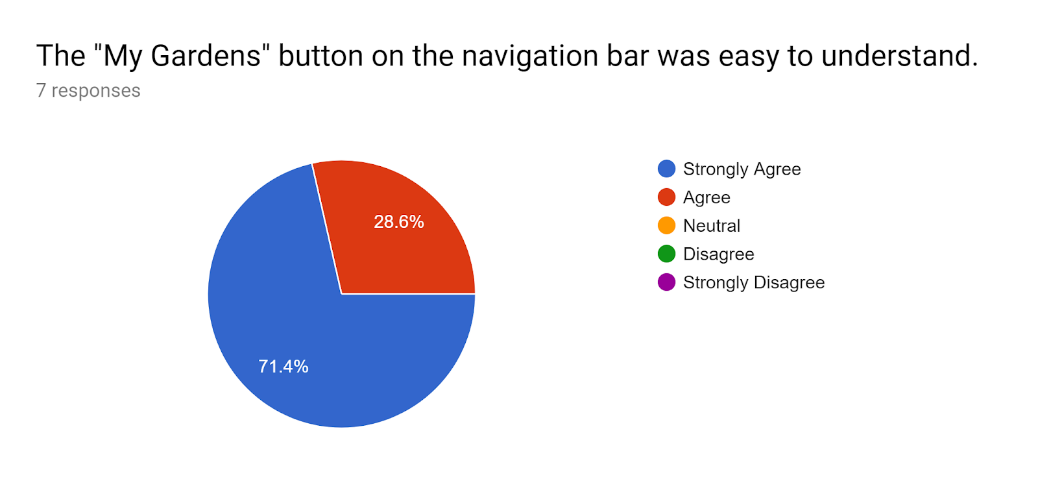
**Figure B-14. Participant responses to “It was easy to understand how to check the garden before generating the shopping list.”**



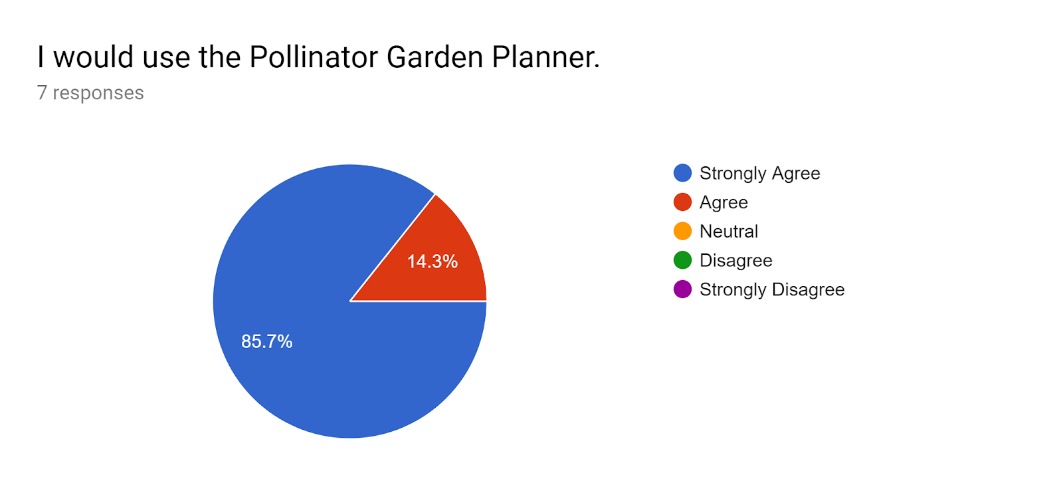
**Figure B-15. Participant responses to “It was easy to understand how to generate a shopping list.”**



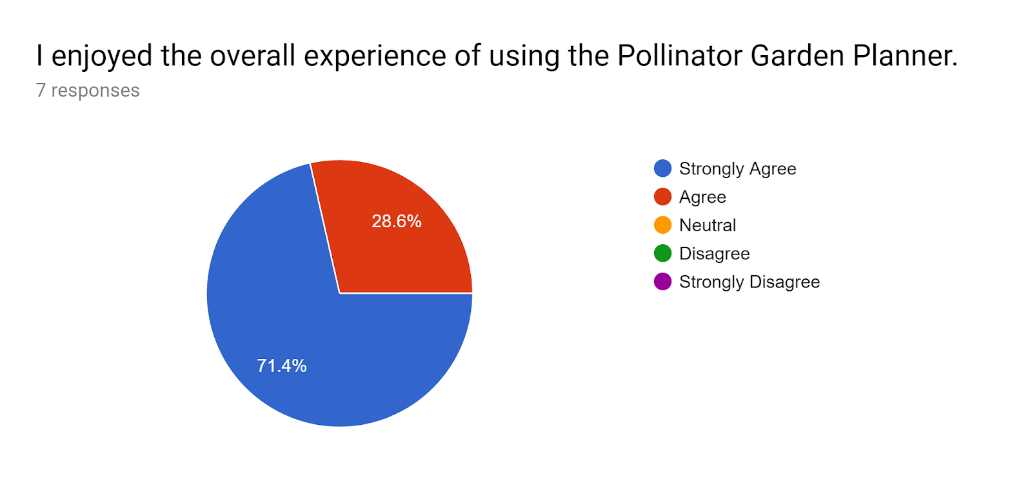
**Figure B-16. Participant responses to “It was easy to understand how to email or print the shopping list.”**



**Figure B-17. Participant responses to “The ‘My Gardens’ button on the navigation bar was easy to understand.”**



**Figure B-18. Participant responses to “I would use the Pollinator Garden Planner.”**



**Figure B-19. Participant responses to “I enjoyed the overall experience of using the Pollinator Garden Planner.”**

**Table B-1. Responses to “What did you like about the Pollinator Garden Planner?**

|  |
| --- |
| I liked that it lets you know how well it will actually feed pollinators through the critical mass and bloom seasonality features |
| It is awesome and needed. YES! I gave details during the session. |
| I liked ease of the image dragging process, and the flow of creating an account/logging in. |
| I like that I can view the garden at different seasons, and I like that I can choose plants based on criteria (especially native vs non-native). |
| It was mostly intuitive and easy to use. Icons were nice and made sense. |
| I enjoyed the filtering options and the possibility of drawing the garden shape |
| Easy and features are mostly intuitive. Checking across seasons is nice. And checking to see if your garden fulfills some criteria is really awesome. |

**Table B-2. Responses to “What did you like about the Pollinator Garden Planner?”**

|  |
| --- |
| This might exist, but it would be nice to have a blurb that would educate people a little bit about why things like critical mass and flower seasonality are important before they start creating the garden. I worry that someone either will not know to check the validity of the garden or will design it based only on plants they like and then be hesitant to revise their design after checking it. Although, admittedly, people using this app will probably have a good sense of this already. A filter based on the type of pollinators the plant would support would be great. Finally, it may be nice to have some typical garden objects (e.g. benches, fountains, trellises) that people can add to better visualize their garden. Overall, great job! This app certainly will be easy to use for most people! |
| I gave details during the session. |
| Make the plant filtering options more clear that is there purpose. |
| Eraser, descriptive note on "checking" garden, direction on how to change the view. |
| Some terminology could be kept more consistent. |
| the seasonality icons were not immediately clear, and I'm not sure how the plants get added to the garden |
| Actually photographs for the plant search side of things, but simple drawings for the garden design. Would be even cooler if you could hover over the simple drawing in the planner side that could show you the actual image. |

1. Potts et al. “Trends in Ecology and Evolution.” *Science Direct*, vol. 25, issue 6, 2010, pp. 345 - 353, www.sciencedirect.com/science/article/pii/S0169534710000364. Accessed 10 Oct. 2018. [↑](#footnote-ref-1)
2. “Campus Bee Hotels are Open for Business.” *Applied Ecology News*, cals.ncsu.edu/applied-ecology/news/campus-bee-hotels-are-open-for-business/. Accessed 10 Oct. 2018. [↑](#footnote-ref-2)
3. generalcounsel.ncsu.edu/legal-topics/business-matters/contracts/confidentiality-or-non-disclosure-agreements/ [↑](#footnote-ref-3)
4. generalcounsel.ncsu.edu/legal-topics/research-and-technology-transfer/ [↑](#footnote-ref-4)
5. https://www.oracle.com/mysql/ [↑](#footnote-ref-5)
6. oit.ncsu.edu/campus-it/web-services/mysql-sla/ [↑](#footnote-ref-6)
7. www.npmjs.com/package/mysql#introduction [↑](#footnote-ref-7)