Plantalogue

Software Requirements Specification

Catalogued and Student Workers Database

Due: February 11th, 2016

Maher Algibhah, Flannery Collins,
Michelle Desormeaux, James Murphy

Prepared for
Software Engineering
CPSC 430

1. Introduction

This section gives a scope description and overview of everything included in this Software Requirements Specification (SRS) document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the "Plantalogue" system. This document will cover each of the system's intended features, as well as offer a preliminary glimpse of the software application's User Interface (UI).

1.2 Scope

The software system "Plantalogue" is a Web application which helps the system main user (Dr. Wynn) to keep track of certain plants in the department of The UMW, shows the specific care instructions needed for the plants, assign students to care about certain plants and provides easily accessible information for (and about) the students who are caring for them.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
Task	A work assignment given to a student worker (Privileged User) that relates to the care of a plant in the greenhouse or phytotron.
Software, System, "Plantalogue"	The Catalogued & Student Workers Database web service
User	A person who interacts with the web service at any privilege level.
Administrator	The client, faculty member, or appointed user with create, modify, remove, search, and view privileges for Plants, Users, and Tasks.
Privileged User	Student workers or volunteers who have search and view privileges for Plants, Users, and Tasks.
Basic User	Plant owners, plant donors, or other tangentially-involved Users who have viewing privileges for Plants.

1.4 References

N/A

1.5 Overview

The rest of this System Requirements Specification will contain a general description and specific requirements of the software. The second section- general description - will contain information about the specific requirements, but the details of the software's requirements will be covered at length in the third section- system requirements. The core of the second section will describe the functions of the product, the characteristics of our target users, and any similar products that we are modeling our system after. We will also get into the general constraints that were faced when building this program as well as any assumptions we make and dependencies we have in order to better explain the system and its requirements. The heart of the third section will be to actually lay out each requirement and function of the system at large. This includes all user, hardware, software and communication interfaces used either in the system itself or in the creation of the system. All functional, non-functional and database requirements will be explained here. Each use case and all use case diagrams will be present and explained here. The design constraints and the database specifics will also be included in this section. The appendix will close the System Requirements Specification document.

This document is organized as follows. The introduction contains the purpose, the scope of the project, all definitions, acronyms, abbreviations, references, and a this overview. The general description contains the product perspective, the product functions, the user characteristics, the general constraints, and the assumptions and dependencies. The specific requirements section contains the external interface requirements, (the user interfaces, the hardware interfaces, the software interfaces, and the communication interfaces). This section includes the descriptive list of functional requirements, and all use cases and their diagrams. Non-functional requirements are next, and then the design constraints follow. The logical database requirements are the last thing before the appendices, which close out the System Requirements Document.

2. General Description

The system is a standalone system that will consist of two parts, a web interface and a database, that will be used to categorize plants and the work associated with those plants. Dr. Wynn will have total control over the scheduling of student work in the greenhouse, through our software. The system will also be used to print labels for individual plants.

2.1 Product Perspective

The web application will only be used locally, on the campus of UMW, so it will not interact with any other application. All data input and manipulation will be manual, by a user. The program is not meant to share data, simply to store data. It is only available to Dr. Wynn, and people that are working for her.

The web interface is where the user interacts with the program, and the database is where all of the information about the plants and users will be stored. There will be no other actors that are external to the system. The database does not have any automated functions, so the control of the system is completely up to the user.

2.2 Product Functions

This program's main functions are: to store information and care instructions about each plant in the UMW biology department, and to allow Dr. Wynn to manage her student-work schedule. There will also be a calendar included that will be a centralized location for workers to view what jobs need to be completed. There will be three different users, who all have different functions that they are allowed to perform.

The Administrator User, Dr. Wynn, has full control of the system. This means that all functions are available to her. The Administrator will be able to add and remove plants from the database, as well as edit the information about each plant entry. Likewise, the Administrator controls who uses the program. She can create and remove users as well. This means that anyone who is using the system can be removed by the Administrator at any time. Users will also be available to the Administrator by search.

The Administrator is also the controller of the work schedule; she can create jobs for plants, and then assign them to a worker afterwards. A major functionality of (name) will also be to print labels that contain information on a specific plant. This allows the user to print a label onto a sticker and place it on plant. Labels can also be saved and modified by the Administrator.

A Privileged User will have some functionality available, but not all. The Privileged User can search and view users, plants, and work. However they cannot modify them. They can also print a label, but cannot create or modify one.

A Basic User can simply search and view plants. Also, they see only the jobs that have been assigned to them. They can also print labels. The Basic User also has the ability to print labels, but cannot create or modify them.

2.3 User Characteristics

The users of this software are expected to be involved with the University of Mary Washington's Department of Biology as faculty, student workers or volunteers, or external providers of plants with interest in botanical experiments. Faculty members (in particular, the client) will be registered as Administrators in the system, able to fill and manage database entries about users and plants as well as provide tasks to workers. Their responsibilities outside the system include defining and conducting experiments with plants, and directing students in those experiments and the care of plants.

Students working in the greenhouse and phytotron under faculty instruction will be registered as Privileged Users in the system, able to mark tasks complete or incomplete and leave comments about their state, as well as manage their own personal details. Student workers may be either on the department's payroll, or be working as volunteers or interns to meet graduation requirements or develop experience working with plants.

Outside users will be registered as Basic Users. This grants them access to plant entries in the database, as well as provides them a place to list their personal information. Such users will be plant donors or plant owners leaving their plants in care of the department, or interested non-worker students who requested access to plant data from an Administrator.

2.4 General Constraints

The software must provide a secure web-oriented interface for Privileged Users (student workers) to access task-related information, which includes not only the task specifics, but also schedule information about their work shifts. To preserve their privacy and avoid malicious use of such information, the authentication process must be resistant to penetration and unaffected by injection attempts.

The software must be compatible with the web browser Mozilla Firefox.

The software must be reliable enough to remain operational during student work shifts and when Privileged Users and Administrators have need of its database and label functions, and should remain operational for workers to view schedules and receive task feedback beyond normal Biology Department operating hours. Scheduled downtime for maintenance, updates, or data back-ups should ideally occur during late hours of nighttime.

2.5 Assumptions and Dependencies

We assume that the UMW Biology Department operates only during normal class hours, and that late night hours (after midnight and before dawn) are safe for downtime. We also assume that the users of this system have sufficient bandwidth to maintain a connection to the service.

3. Specific Requirements

As a general overview of the system's requirements, refer to Figures 1, 2, and 3 for reference. Figure 1 represents all of the possible things that an Administrator User can do, Figure 2 represents the Privileged User, and Figure 3 represents everything that a Basic User has access to. The stick figure in the middle of the diagrams is the User of the system. The bubbles represent each piece of functionality that specific user has access to.

Figure 1 - Administrator User Use Case Diagram

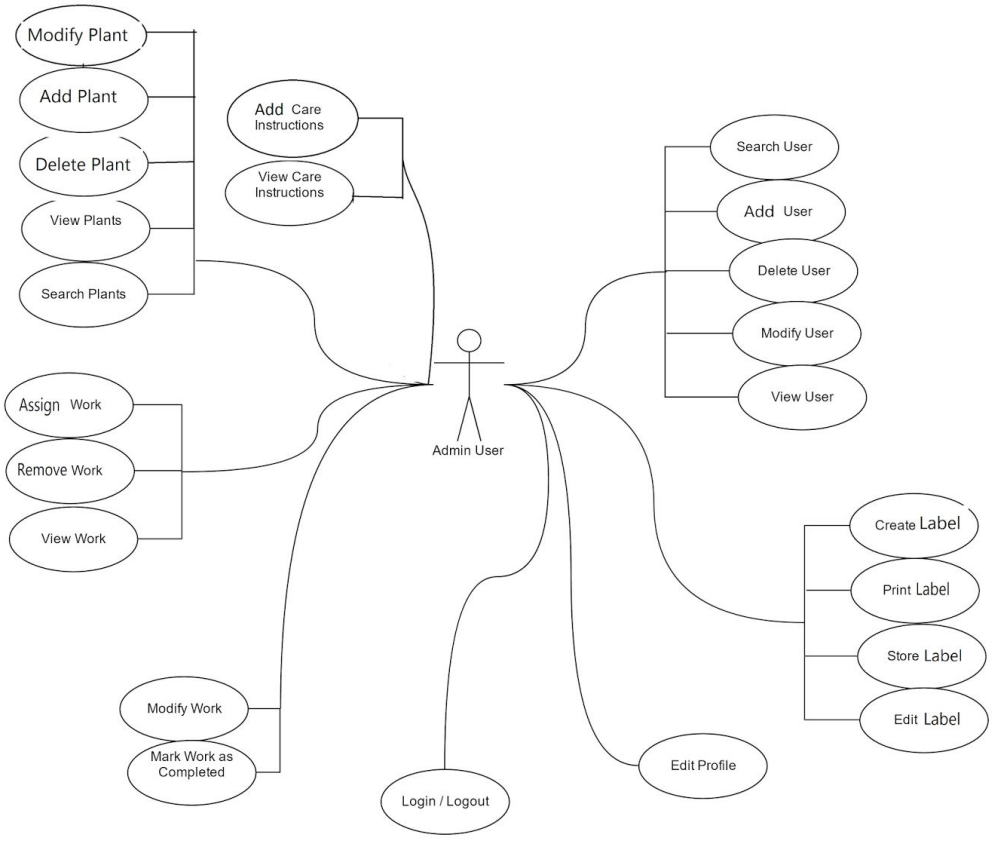

Figure 1

Figure 2 - Privileged User Use Case Diagram

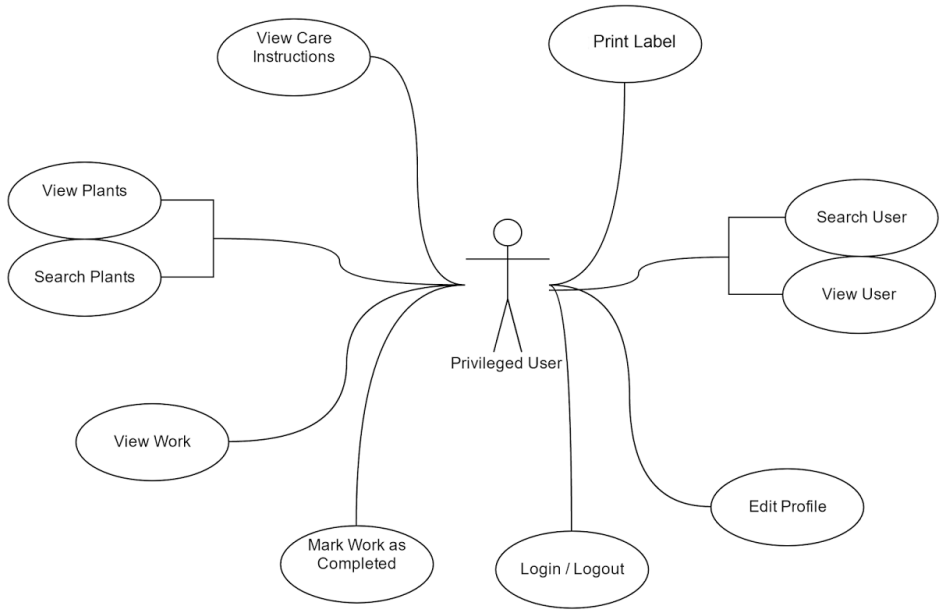

Figure 2

Figure 3 – Basic User Use Case Diagram

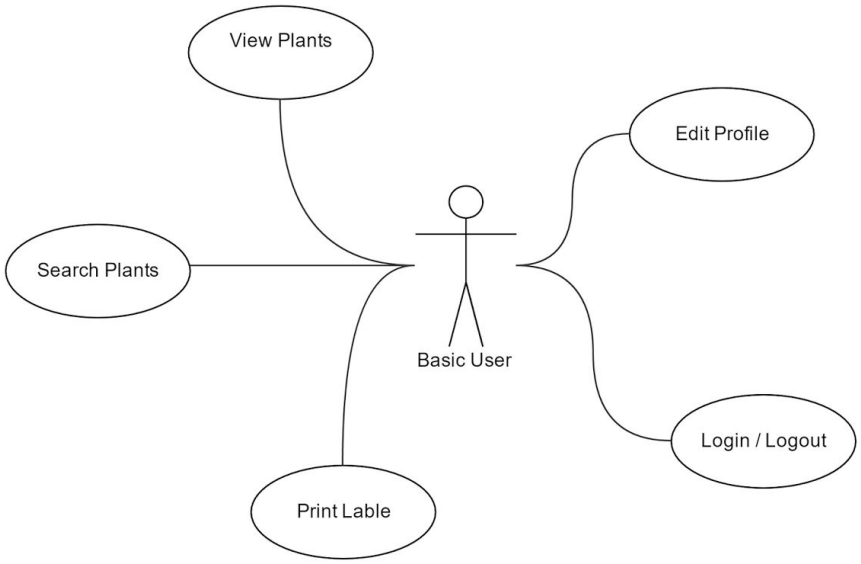

Figure 3

3.1 External Interface Requirements

3.1.1 User Interfaces

There will be a plant information page, a page to display search results, information pages generated for each plant, and profile pages generated for each user. Administrators will have access to a menu with Administrator privilege level functions.

3.1.2 Hardware Interfaces

N/A

3.1.3 Software Interfaces

The system will interact with Amazon Web Services in order to store data.

3.1.4 Communications Interfaces

- Email
- Comment fields in database
- Feedback fields in database

3.2 Functional Requirements

3.2.1 Retrieve password

Author: Maher

3.2.1.1 Introduction - The users should be able to retrieve their password by email.
3.2.1.2 Inputs - The users will select 'retrieve password' to retrieve their password form an user interface.
3.2.1.3 Processing - System shall ask for user's email and check if exisist then send an email to the inputted email with the courrent password.
3.2.1.4 Outputs - User shall recive an email with their password.
3.2.1.5 Error Handling - If the given email does not exist, user will be notified.

3.2.2 Display Plant Entry Page

Author: Michelle

3.2.2.1 Introduction - The system will display a page containing a list of all entries in the plant database.
3.2.2.2 Inputs - The user will choose from the menu to display the plant entries.
3.2.2.3 Processing - All plant entries will be retrieved from the database.
3.2.2.4 Outputs - All plants in the database will be displayed.
3.2.2.5 Error Handling - If there was an error retrieving information from the database, the user will be notified.

3.3 Use Cases

3.3.1 Use Case #1

Use Case Name: Display Plant Information Page

Use Case Author: Michelle Desormeaux

Actor(s): Administrator User, Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: The system will display a page for the plant that has been selected from the list.

Priority: 4

Precondition(s):
· The user needs to have access to the plant list, and have a specific plant selected to view.

Postcondition(s):
· The user will see information about the selected plant.

Basic Path:
1. The user will choose from the list which plant they would like to view.
2. System retrieves plant information.
3. System displays plant's information (ID Number, genus, species, photo if applicable, location – table number, position on table, care instructions, and additional notes specified at plant creation.

Alternative Paths: N/A

Non-Functional Requirements: N/A

3.3.2 Use Case 2

Use Case Name: Remove Plant

Use Case Author: Michelle Desormeaux

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: The system will give the user the ability to delete a plant record from the inventory.

Priority: 4

Precondition(s):
- There needs to be at least one plant in the database.
- Administrator has searched for target plant and is viewing its page.

Trigger(s):
- Administrator selects “Remove Plant” option on target plant’s page.

Postcondition(s):
- The selected plant will no longer be in the database upon successful completion.

Basic Path:
1. System prompts Administrator to confirm target plant deletion.
2. Administrator confirms.
3. System will remove the target plant from the database.
4. System redirects Administrator back to target plant’s page.

Alternative Paths:

2a. Administrator does not confirm plant deletion.

1. Jump to Basic Path 4.

Non-Functional Requirements:
- System will have ability to remove plant from database.

3.3.3 Use Case 3

Use Case Name: Add Plant

Use Case Author: Michelle Desormeaux

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: The system will give the user the ability to add new plants into the database.

Priority: 4

Precondition(s):
- Administrator has information about a new plant to add into the database.

Trigger(s):
· Administrator selects “Add New Plant” option on Administrator menu.

Postcondition(s):
· The database will contain a new plant, added by the Administrator.

Basic Path:
1. The user will be able to fill in these fields: plant ID number, genus and species, picture, location (room number, table number, position on table), care instructions, and additional information specified at creation.
2. The database will check for duplicates.
3. The new plant will be added to the database.

Alternative Paths:
2a. Administrator leaves a field blank.
    1. System prompts Administrator to fill field.
    2. Administrator fills field.
    3. Return to Basic Path 3.

3a. The database had found that the plant the user is trying to add is already in the database.
    1. The user will be notified.
    2. System will redirect the Administrator to the plant list page.

Non-Functional Requirements: N/A

3.3.4 Use Case 4

Use Case Name: Modify Plant

Use Case Author: Michelle Desormeaux

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: The system will give the user the ability to add new plants into the database.

Priority: 4

Precondition(s):
· The user has information they wish to change about a specific plant in the database.

Postcondition(s):

· The database will have updated information about a plant.

Basic Path:
1. The user will click “modify plant”.
2. The user will use the search function to search the database for the plant they wish to modify.
3. The user will be able to choose from one of these fields: ID Number , Genus and Species, Picture, Location (Room Number, Table Number, Position on Table,Care Instructions, and an “other” field.
4. The user will type the new information for the plant in the selected field, and click “update”.
5. The plant’s information will be updated.

Alternative Paths:

2a. The plant was not found.
1. The user will be notified.
2. The user will be asked if they wish to add a new plant.

3a. The user selects “Care Information”
1. The user will be redirected to another screen that contains all of the different fields for a plant’s care information.
2. The user will be able to choose from one of these fields: water, fertilizer, Privileged User caring for plant, owner and contact, duration of experiment, and additional notes made upon creation.

Non-Functional Requirements: N/A

3.3.5 Use Case # 5

Use Case Name: Search Plant

Use Case Author: Michelle Desormeaux

Actor(s): Administrator User, Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: The user will have the ability to search the database for a specific plant.

Priority: 4

Precondition(s):
· The database will have at least one plant to search for.The system will return all entries that this type of user has access to.

Postcondition(s):

· The system will display all matching entries in the database that contained the search term.

Basic Path:
1. User specifies search criteria – genus, species, plant ID, etc.
2. System will search the database for matches (whole or partial) to search criteria.
3. The system will return all entries in the database that have any field that matches the search term.

Alternative Paths:
1a. The system found no matches.
    1. The user will be notified that the system was unable to locate any entries matching the search term.

Non-Functional Requirements: N/A

3.3.6 Use Case # 6

Use Case Name: Display Administrator Control Panel

Use Case Author: Maher Algibhah

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Displaying user interface showing all functions that the Administrator can select to perform.

Priority: # 1

Precondition(s): System is up and running and user is previously logged in as an administrator.

Postcondition(s): Control Panel/Error-message is displayed.

Basic Path:
1. User selects to display the control panel
2. System check if the user is still logged in as an Administrator
3. System shall display the Control Panel

Alternative Paths:
2a. **Authorization fails (**User is not logged in as an administrator anymore)
    1. User selects to display the control panel
    2. System check if the user is still logged in as an Administrator

3. If no, System shall ask the user to login as an Administrator
4. User will enter their username and password
5. System shall check username and password
6. System shall display the Control Panel

3b.**Authorization fails (**password/username not correct)

1. User selects to display the control panel
2. System check if the user is still logged in as an Administrator
3. If no, System should ask the user to login as an Administrator
4. User shall enter their username and password
5. System shall check username and password
6. System shall display Control Panel access is not allowed.

Non-Functional Requirements: N/A

3.3.7 Use Case # 7

Use Case Name: Display User Profile

Use Case Author: Maher Algibhah

Actor(s): Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: Displaying user interface showing employee/volunteers information (username/password/photo profile … etc.).

Priority: # 2

Precondition(s): System is up and running and user previously logged in as a Privileged User or a Basic User.

Postcondition(s): The user will see their profile.

Basic Path:

1. User shall click on 'Profile'
2. System shall check if the user is still logged in.
3. System shall display the user profile

Alternative Paths:

2a. **Authorization fails (**User is not logged anymore)

1. User shall click on 'Profile'
2. System check if the user is still logged in
3. If no, System shall ask the user to login.
4. User shall enter their username and password

5. System shall check username and password
6. System shall display the user profile

3b.**Authorization fails (**password/username not correct)

1. User shall click on 'Profile'
2. System check if the user is still logged in
3. If no, System shall ask the user to login.
4. User shall enter their username and password
5. System shall check username and password
6. System shall display Profile access is not allowed.

Non-Functional Requirements: N/A

3.3.8 Use Case # 8

Use Case Name: Manage User Profile

Use Case Author: Maher Algibhah

Actor(s): Administrator User, Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: After the user login, allowing the user to manage (edit) their information (username/password/photo profile … etc.).

Priority: # 5

Precondition(s): System is up and running and user is logged in. User has information they wish to change about their profile. User's Profile is displayed.

Postcondition(s): The system will have the updated information about the user

Basic Path:

1. User select part of the profile that they desire to update.
2. User input new information.
3. User click 'save' to save the new information.
4. System shall save the new information
5. System shall update the profile to include the latest saved information

Alternative Paths:

2a. User may cancel at any time.

1. Use click 'Cancel' to cancel the process.
2. System shall restore the profile to include the latest saved information

Non-Functional Requirements: N/A

3.3.9 Use Case # 9

Use Case Name: System Login

Use Case Author: Maher Algibhah

Actor(s): Administrator User, Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: Provide user interface that let the user to input username and password which will be matched to user info in the database.

Priority: # 5

Precondition(s): System is up and running.

Postcondition(s): System shall display main interface

Basic Path:

1. User selects to log in
2. User shall input the username and password
3. System shall check if username/password matches the stored username/password
4. System shall display the main interface to the user with indication of successful log-in

Alternative Paths:

2a. Authorization fails (Incorrect username/password with no retying to login)

1. System shall output a message stating that username and/or password is not correct
2. System shall count the failed login attempt
3. System shall ask the user to cancel, retry to login or retrieve username/password
4. User selects to cancel
5. System returns the user to the main interface

3b. Authorization fails (Incorrect username/password with retying to login)

1. System shall output a message stating that username and/or password is not correct
2. System shall count the failed login attempt
3. System shall ask the user to cancel, retry to login or retrieve username/password
4. User selects to retry to login
5. Previous steps are repeated until user login successfully or three consecutive failed login-attempts happen

6. If three consecutive failed login-attempts happen, System shall prevent further login attempts for a specified period.
7. System returns the user to the main interface

4c. Retrieve Username/Password

1. System shall count the failed login attempt
2. System shall ask the user to cancel, retry to login or retrieve username/password
3. User selects to to retrieve username/password
4. System ask for user's email address
5. User shall enter their email
6. System shall check for email existing
7. If email matches, system shall send an email with the username and password
8. System returns the user to the main interface

4c. Fail to Find User Email

1. System shall count the failed login attempt
2. System shall ask the user to cancel, retry to login or retrieve username/password
3. User selects to to retrieve username/password
4. System ask for user's email address
5. User shall enter their email
6. System shall check for email existing
7. If email does not match, system shall display a message stating that the given email is unknown
8. System returns the user to the main interface

Non-Functional Requirements: N/A

3.3.10 Use Case # 10

Use Case Name: Change Password

Use Case Author: Maher Algibhah

Actor(s): Administrator User, Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: All types of users should be able to change their passwords.

Priority: # 5

Precondition(s): User must be logged in

Postcondition(s): The database will have updated information about the user password

Basic Path:

1. User initiates the password change command.
2. User is prompted for old password, new password and to confirm new password.
3. User inputs the old password, new password and confirm new password.
4. System does authentication.
5. If authentication is successful, system update user password.
6. System shall return to the main interface.

Alternative Paths:

2a. Authorization fails (Old password is not correct)

1. User initiates the password change command.
2. User is prompted for old password, new password and to confirm new password.
3. User inputs the old password, new password and confirm new password.
4. System does authentication.
5. If old password is not correct, prompt the user that the old password is not correct.
6. System shall return to the main interface.

3b. Authorization fails (passwords do not match)

1. User initiates the password change command.
2. User is prompted for old password, new password and to confirm new password.
3. User inputs the old password, new password and confirm new password.
4. System does authentication.
5. If the new password is not match, system shall ask the user to re-enter new password and confirm new password.
6. System shall return to the main interface.

Non-Functional Requirements: N/A

3.3.11 Use Case # 11

Use Case Name: System Logout

Use Case Author: Maher Algibhah

Actor(s): Administrator User, Privileged User, Basic User

Other Stakeholders: N/A

Summary Description: User can logout from the system.

Priority: # 5

Precondition(s): System is up and running.

Postcondition(s): System shall display main interface

Basic Path:
1. User selects to log out
2. System shall logout the user
3. System shall return to the main interface

Alternative Paths:
2a. Automatic Logout
   1. System calculate the time passed since user not active
   2. System shall check if the calculated time exceeded a specified threshold
   3. If yes, System shall logout the user
   4. System return to the main interface
   5.

Non-Functional Requirements: N/A

3.3.12 Use Case # 12

Use Case Name: Edit/Modify Task

Use Case Author: Flannery Collins

Actor(s):
- Administrator (Primary)
- Worker (Secondary)

Other Stakeholders: N/A

Summary Description: Administrator shall have the ability to change the details of a task, post creation.

Priority: 2

Precondition(s):
- There must already be a task that the Administrator has previously created. In other words, you cannot edit/modify a task that does not exist.

Postcondition(s):
- The task is saved in the database with the updated information that the user has changed.

Basic Path:
1. User navigates to the list of tasks.

2. User then searches for, and selects which task they would like to modify.
3. The Administrator then has the ability to edit the following information about each task:
    a. watering intervals and volumes
    b. fertilizer types, intervals, and volumes
    c. the name and contact information of a plant owner
    d. the name of an associated student worker assigned to care for the plant
4. Then, the user clicks "Save/Update".
5. The information associated with the task is updated in the database.
6. The updated task information is now available to view by associated users.

Alternative Paths:
2a User Does Not Save/Update
1. User navigates to the list of tasks.
2. User then searches for, and selects which task they would like to modify.
3. The Administrator then has the ability to edit the information about the task.
4. For some reason, the user then does not click "Save/Update"
    a. Either the user changes their mind about the edit, OR
    b. The user exits the program before they save.
5. The information that the user changed about the task is not saved, and the original data for the task is kept the same.
6. The task is not changed, and therefore looks the same when viewed by associated users.

Non-Functional Requirements:
· The Administrator will type into text boxes to edit the information about tasks.
· Images can be uploaded to be associated with the plant from the Administrator's computer.

3.3.13 Use Case # 13

Use Case Name: Assign Task

Use Case Author: Flannery Collins

Actor(s):
· Administrator (Primary)
· Privileged Users, Basic Users (Secondary)

Other Stakeholders: N/A

Summary Description: The Administrator can assign a created task to be worked on by a Privileged User. The Privileged User is then associated with that task.

Priority: 1

Precondition(s):

· There must be a task previously created by the Administrator, which exists in the database.

· There must be a Privileged User created by the Administrator, which exists in the database, and has login access to the system.

Postcondition(s):

· The task is assigned by the Administrator, to the Privileged User. The Privileged User is now associated with the task, and is notified that they have been assigned a task.

Basic Path:

1. The Administrator navigates to the list of tasks.
2. The Administrator then selects which task(s) that they wish to assign to a Privileged User.
3. The Administrator then clicks “Assign Task”.
4. Then, the system prompts the Administrator User with a list of Privileged Users that the task can be given to.
5. The Administrator then selects which Privileged User(s) she wants to work on the task.
6. The Administrator then selects “Save” option.
7. The Task is assigned to a worker, and the database is updated.
8. The Privileged User is notified that a task has been assigned to them.
9. Administrator is redirected to Task menu.

Alternative Paths:

1a. Administrator cancels task assignment.

1. The task is not assigned, and therefore no users are notified about the task being assigned.
2. Jump to Basic Path 9.

Non-Functional Requirements:

· The User will select task(s) that they wants to assign by check boxes.

3.3.14 Use Case # 14

Use Case Name: Remove Task

Use Case Author: Flannery Collins

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Allows Dr. Wynn to delete a Task from the database, whether it is finished, or simply no longer needed.

Priority: 1

Precondition(s):
· There is already a task in the database that the Administrator previously created.

Postcondition(s):
· The task is removed from the database, and is erased forever, therefore removing its associations with the Worker (Basic User). Removed Tasks can no longer be seen by other users.

Basic Path:
1. The Administrator user navigates to the list of tasks.
2. User selects which task that they want to erase.
3. User clicks, "Delete".
4. Task is deleted, and therefore erased from the database.

Alternative Paths:
N/A

Non-Functional Requirements:
- The User will select tasks with a check box.

3.3.15 Use Case # 15

Use Case Name: Update Task Progress

Use Case Author: Flannery Collins

Actor(s):
· Administrator
· Privileged Users

Other Stakeholders: N/A

Summary Description: When the worker in the lab, whether it is Dr. Wynn or a student worker, leaves the biology lab for the day, they will mark the Task as either "In Progress" or "Completed".

Priority: 8

Precondition(s):

· There must be a task or tasks in the system previously created by the Administrator.

o Someone must be working on the previously mentioned task or tasks.

Postcondition(s):

· The task is now marked.

· The Administrator user now knows exactly which tasks have already been completed, and which tasks still need to be worked on.

Basic Path:

1. The user navigates to the page in the system for the task that they are currently working on, or that they just completed.
2. The user clicks "Update Task Progress".
3. The user marks the associated check box for either "In Progress" or "Completed", whichever applies to the current situation.
4. The user clicks "Save".
5. The task progress is updated and saved in the database.
6. System redirects User to task list.

Alternative Paths:

1a. User may cancel at any time.

1. Jump to Basic Path 6.

Non-Functional Requirements:

· The user will choose between "In Progress" or "Completed" by ticking a checkbox.

3.3.16 Use Case # 16

Use Case Name: Give Feedback on Task

Use Case Author: Flannery Collins

Actor(s):

· Administrator (Primary)

· Privileged (Secondary)

Other Stakeholders: N/A

Summary Description: The Administrator can give feedback to the Privileged Users on how their tasks were completed.

Priority: 7

Precondition(s):

- There must be a task that was previously created by the Administrator, and that exists in the database.
- There must already be a Privileged User created by the Administrator, and that exists in the database.
- This created task must already be assigned to a Privileged User. In other words, to be able to give feedback, someone must be working on the task, or have completed the task.

Postcondition(s):

- The feedback is now associated with a Privileged User, and also with the task that they are/were working on. However, the feedback is only viewable by the Administrator, and the Privileged User that is associated with, no other users.

Basic Path:

1. Administrator navigates to the list of Privileged Users.
2. Administrator then selects the Privileged User that they wish to give feedback to.
3. Administrator is then selects "Give Feedback" option.
4. Administrator is then prompted with the list of tasks that is associated with the selected Privileged User.
5. The Administrator then selects which task they want to give feedback on.
6. A text box appears, and the Administrator writes the feedback/comments.
7. The Administrator then clicks "Save/Update".
8. The feedback is saved in the database, and is associated with the task, and the privileged user.

Alternative Paths:

2a. Giving Feedback From Task List

1. The Administrator navigates to the list of tasks.
2. The Administrator then selects the task that they wish to give feedback on.
3. The Administrator is then clicks "Give Feedback".
4. The Administrator is then prompted with the list of Users that are associated with the selected task.
5. The Administrator then selects which User they want to give feedback to.
6. A text box appears, and the Administrator writes the feedback/comments.
7. The Administrator then clicks "Save/Update".
8. The feedback is saved in the database, and is associated with the task, and the privileged user.

2b. User Does Not Click "Save/Update"

1. The Administrator then selects the task that they wish to give feedback on.
2. The Administrator is then clicks "Give Feedback".
3. The Administrator is then prompted with the list of Users that are associated with the selected task.
4. The Administrator then selects which User they want to give feedback to.
5. A text box appears, and the Administrator writes the feedback/comments.

6. For whatever reason, the User does not click “Save/Update”.
    a. Either the User exits the program, or
    b. The User navigates away from the screen.
7. The feedback is not saved, and the original data is still in the database.

Non-Functional Requirements:
· The user will type feedback into text boxes.
· The user will select tasks and users using check boxes.

3.3.17 Use Case # 17

Use Case Name: View Tasks Calendar

Use Case Author: Flannery Collins

Actor(s):
· Administrator User
· Privileged Users

Other Stakeholders: N/A

Summary Description: There will be a Calendar that is accessible by the Administrator User and Privileged Users. There will be a link on User login screen to “View Calendar”.

Priority: 9

Precondition(s):
· The User must have tasks that they added to the system.
· The task database is linked to the calendar, so that when a task is created, it automatically appears on the calendar on the date of creation.
· Nothing will appear on the calendar if there are no tasks

Postcondition(s):
· The User now know what tasks were created on which day, even tasks that they is not working on.

Basic Path:
1. The user navigates to their home screen.
2. The user clicks on the “Calendar” button.
3. The user is taken to the page where the Calendar is located.

Alternative Paths: N/A

Non-Functional Requirements:

1. There will be a Calendar that provides an overview of scheduled tasks, and what day the tasks were created on, to make task viewing easier for the user.
2. The Calendar will provide a view of the entire current month.

3.3.18 Use Case # 18

Use Case Name: Notification Updates

Use Case Author: Flannery Collins

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Every time the Administrator logs in to the system, they are notified with all of the tasks that have been completed since their previous log in. If no tasks have been completed, then the User receives no update.

Priority: 8

Precondition(s):

- There must be tasks that have already been created by the Administrator
- Those tasks must already have been assigned to a Privileged User.
- Also, the student worker (Privileged User) must have already completed the task.

Postcondition(s):

- The Administrator now knows which tasks have been completed, so that they can now delete them.

Basic Path:

1. The User logs into the system with their username and password
2. The software prompts the User on the first page (after log in) with the list of Tasks that were completed since their previous log in.

Alternative Paths:

2a. No Tasks Have Been Completed

1. The User logs into the system with their username and Password.
2. There were no tasks completed since the Administrator's previous log in, so the User is not prompted with any completed tasks.

2b. User Can Not Log In

1. The User enters incorrect log-in information.
2. The User is not able to log on to the software.

Non-Functional Requirements:

- The updates will be presented to the user in a easy to read, formatted text box, containing the job and what students were working on it.

3.3.19 Use Case # 19

Use Case Name: Create Label

Use Case Author: James Murphy

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Administrator can create label for a plant.

Priority: 6

Precondition(s):
- Administrator must be logged in.
- Administrator has searched for a plant and selected its page.
- Plants must be entered in the database.

Trigger(s):
- Administrator selects “Create Label” option on plant’s page.

Postcondition(s):
- A label is associated with a plant record in the database.

Basic Path:
1. Administrator selects “Create Label” option on plant's page.
2. System lists plant data fields for selection.
3. Administrator selects which fields should be present on the label.
4. System provides “Additional details” text field.
5. Administrator (optionally) fills out “Additional details” field.
6. Administrator opts to save the label.
7. Administrator provides a name for the label.
8. System saves label information with plant record in database.
9. System redirects Administrator to plant page.

Alternative Paths:

1a. Administrator may cancel at any time.
    1. Jump to Basic Path 9.

7a. Label already exists with provided name.
    1. System prompts Administrator to choose a different name for the label, overwrite the previous label, or cancel.

2. Administrator chooses a different name for the label.
3. Return to Basic Path 8, *OR*
4. Administrator chooses to retain the same name for the label and overwrite the previous file.
5. System overwrites the previous label,
6. Return to Basic Path 9, *OR*
7. Administrator chooses to cancel the save action.
8. Return to Basic Path 2.

3.3.20 Use Case # 20

Use Case Name: Print Label Batch

Use Case Author: James Murphy

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Administrator may select one or more plants to print labels for.

Priority: 6

Precondition(s):
- Administrator must be logged in.
- Plants must be entered in the database.
- Labels must be created for those plants.

Trigger(s):
- Administrator selects “Print Labels” option from Administrator menu.

Postcondition(s):
- At least one label will be printed or queued to print.

Basic Path:
1. System provides Administrator with a list of plants with labels available to select for printing.
2. Administrator selects plants to print their labels.
3. For plants with multiple labels stored, the system prompts the Administrator to select a label.
4. System provides Administrator with the list of selected labels and prompts them to enter copy quantities for each label, with a minimum of 1 copy per label.
5. Administrator (optionally) changes label quantities
6. Administrator confirms quantities and selects “Print” option.
7. System generates a print preview and displays it to Administrator.

8. Administrator confirms the print job.
9. System sends print job to printer.
10. System redirects Administrator to main menu page.

Alternative Paths:

2a. No Plants

1. System reports to Administrator that no plant entries exist in database.
2. System redirects Administrator to Create Plant page.

2b. No Labels.

1. System reports to Administrator that no labels have been created.
2. System redirects Administrator to plant search page.

Non-Functional Requirements:

1. Labels are printed eight to a standard 8.5" x 11.5" sheet of paper.
2. The software interfaces with the print functionality of Mozilla Firefox.

3.3.21 Use Case # 21

Use Case Name: Add User

Use Case Author: James Murphy

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Administrator may add a new user to the database.

Priority: 3

Precondition(s):

· Administrator must be logged in.

Trigger(s):

· Administrator selects "Add New User" option from Administrator menu.

Postcondition(s):

· A new user will be added to the database.

Basic Path:

1. System prompts Administrator for User's name, telephone number, and email address.
2. Administrator fills these fields.
3. System provides Administrator with privilege options of Administrator, Privileged User, and Basic User to assign to the new User.
4. Administrator selects the new User's privilege level.
5. System asks for a range of dates for a User's work schedule.

6. Administrator provides a start date and an end date.
7. System prompts for the days of the week that the student is scheduled to work.
8. Administrator selects days that the student will work.
9. For each day of the week selected for work, System prompts Administrator for shift times.
10. Administrator selects start and end times for each shift.
11. Administrator selects "Save User" option.
12. System saves new User to database
13. System returns Administrator to Administrator menu.

Alternate Path:

1a. Administrator may cancel at any time.
    1. Jump to Basic Path 13.

5a. New User is not a Privileged User and will not have a work schedule.
    1. Jump to Basic Path 11.

11a. Administrator leaves (a) necessary field(s) empty.
    1. System prompts Administrator to fill empty field(s).
    2. Administrator fills empty field(s).
    3. Jump to Basic Path 12.

3.3.22 Use Case # 22

Use Case Name: Modify User

Use Case Author: James Murphy

Actor(s): Editing User, which may be an Administrator (able to edit any User's information) or a Privileged or Basic user (able to edit their own information).

Other Stakeholders: N/A

Summary Description: Change the information stored in target User database entry.

Priority: 3

Precondition(s):

- Administrator must be logged in.
- At least one User must exist in the database.
- Administrator has searched for target User.

Trigger(s):

- Administrator selects "Modify User" option on the target User's page.

Postcondition(s):

- Target User's information will be altered and overwritten in the database.

Basic Path:

1. System prompts Editing User with name, telephone number, and email address fields for editing.
2. User fills these fields.
3. System prompts Editing User to edit target User work schedule.
4. User changes time range, days of week, and shift times accordingly.
5. User selects "Save Changes" option.
6. System writes changes to target User entry in database.
7. System returns Administrator to User Search menu.

Alternate Path:

1a. User may cancel at any time.
    1. Jump to Basic Path 7.

3a. Privileged Users may not edit their own schedules.
    1. Jump to Basic Path 5.

3b. Basic Users have no schedule to edit.
    1. Jump to Basic Path 5.

3c. Administrators have no schedule to edit.
    1. Jump to Basic Path 5.

3.3.23 Use Case # 23

Use Case Name: Delete User

Use Case Author: James Murphy

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Administrator can remove User from database.

Priority: 3

Precondition(s):

- Administrator is logged in.
- At least one User exists in the database.
- Administrator has searched for target User.

Trigger(s):

- Administrator selects "Delete User" option on User's page.

Postcondition(s):

- User is removed from database.

Basic Path:

1. System prompts Administrator for password to confirm deletion.
2. Administrator enters password to confirm.
3. System removes target User from database.
4. System returns Administrator to User Search menu.

Alternate Path:

1a. Administrator may cancel at any time.
    1. Jump to Basic Path 4.

3.3.24 Use Case # 24

Use Case Name: Search for User

Use Case Author: James Murphy

Actor(s): Administrator

Other Stakeholders: N/A

Summary Description: Non-Basic User can search for target User.

Priority: 3

Precondition(s):

- Non-Basic User is logged in.
- At least one User exists in the database.

Trigger(s):

- Non-Basic User selects “Search Users” option on main menu.

Postcondition(s):

- Non-Basic User finds Users matching search options.

Basic Path:

1. System displays a list of all users, sorted alphabetically by last name.
2. System provides search boxes for first name and last name, email address, and phone number.
3. Non-Basic user fills search boxes of their choice.
4. System filters out users in the list who are not at least a partial match for the input.
5. Non-Basic User finds their target user.

Alternate Path:

1a. No users exist.
    1. System displays message to Non-Basic User indicating no users exist in the database.

4a. No matches found.

1. System displays message to Non-Basic User indicating no users exist matching the provided search query.
2. System suggests that Non-Basic User broaden their search.

3.4 Non-Functional Requirements

- The Software will be designed for Mozilla Firefox.
- It will be required that any user who logs into the system must have sufficient bandwidth to stay connected to the internet.
- We must make sure that we have enough storage space available for our database.
- A login interface will be required and the system will accept a user's information when they enter it correctly.
- The authentication process must be resistant to penetration and unaffected by injection attempts.
- Labels are to be formatted and printed to a standard 8.5 x 11 sheet of paper.
- The software will use the functionality of the web browser to actually print the label.
- The software must be reliable enough to operate past biology department operating hours.
- There will be a calendar interface that allows the Administrator and the Privileged User to view tasks based on their creation date.
    - The ability to "View Calendar" is a Use Case.

3.5 Design Constraints

The system will only be able to hold space for as many plants and Users as the Amazon web service allows. The system's performance will be limited by the nature of the programming language used for implementation and the bandwidth of the User and Amazon web service.

3.6 Logical Database Requirements

The system will use a database to store and retrieve information for the client. Data that deals with confidential user information will be protected with blocks on what each user can search for, as well as what can be displayed.

A. Appendices

A.1 Appendix 1

Privilege Note(s)

A user's privilege, once set, cannot be changed.