**Plant Selection Planner, June 25, 2022**

**Description**

A plant database searcher in which a user can retrieve a list of plants that fall under desired criteria – hardiness zone, sun exposure, herbaceous/shrub/tree, evergreen/deciduous (for woody shrubs/trees), and bloom time. The user can then add plants to a personal collection that they can print and take on a trip to a nursery.

**Personnel**

Haley Gustafson, Ash Carruth, John Sauter

**Motivation for Development**

Amateur gardeners often go to plant nurseries and ask for plants that meet certain criteria. It can be difficult to narrow down the immense number of choices at a nursery while meeting all desired criteria. If a gardener were able to search a list of plants by desired criteria and develop a list of plants that meet their needs, they could take their list to a nursery and ask if their desired plants are available, rather than asking the nursery to narrow their choices down for them, expediting the process for everyone.

**User Stories**

As a gardener,

I want to search for plants filtered by certain criteria

So that I can choose plants that suit my gardening needs

As a user,

I want to add plants to a personal collection

So that I can print that list to take with me to a nursery

As a Nursery Manager,

I want to provide my customers with the plants that meet their needs without wasting a lot of time.

**Technologies**

Foundation, Node.js, Express.js, Handlebars.js, MySQL, Sequelize, nodemailer

**Task Breakdown**

1. Design user flows: what pages will each type of user see, and what actions can be taken on each
2. Design data structures to support the user actions
3. Populate the plant database
4. Design and code the server: this includes specifying the paths
5. Construct and lay out the user pages based on the user flow design, using the paths to the server
6. Test and debug
7. Deploy to Heroku

**Preliminary User Flow Design**

This is a wordy and necessarily vague description of how the application will look from the users’ points of view. It is a complement to the wireframe.

There will be two types of users: gardeners and nursery managers. Each will have its own flow, though a nursery manager can switch to the gardener flow to use the application on behalf of a walk-in customer who is unfamiliar with the software.

The nursery manager will populate the plant database with all the plants in his inventory. If a plant is sold out he can mark it out of stock. He will be able to update the description of each plant, both the written description and (in a future version) the picture. Importantly, he must specify the characteristics of the plant, such as what degree of sunlight it needs. In a large nursery the nursery manager can specify that a plant can be found in a certain location, such as the showroom or the rear terrace.

The gardener will be able to choose plants from the database for her garden. She can divide the garden into zones and choose plants for each. When choosing plants she can specify filters to narrow her selection. For example, she might only wish to see plants that are suitable for her climate. Filters can be combined: she might wish, for example, to see only plants that are deer-resistant and in stock.

When the gardener has chosen the plants she wishes to purchase she can send the list to the nursery as an email message, or use the application from the nursery to show the nursery manager what she wants. To facilitate this, the nursery manager will be able to see the gardener’s selections. This can also be used during a remote interaction: the nursery manager can give suggestions and watch the changes in the gardener’s choices.

**User Interface Design Details**

In this space we will flesh out some details of the user interface.

When a user first connects to the application, he must login or sign up for an account. Signing up requires a username, email address, and password. Both the username and email address must be unique.

The main page for the gardener is the list of chosen plants, with a selector to specify which zones of her garden she wishes to see. She can create and delete zones, and designate a zone as the one she is currently working on. She can further filter the list by specifying criteria such as shade requirements. She can delete a plant from the current zone. She can add plants to the current zone. When adding a plant she performs a search of the plant database. Initially, all of the plants that meet her current criteria will be displayed. She can change her criteria, in which case the list of plants will be updated accordingly.

An important part of the user interface is specifying the viewing filters. There needs to be a convenient and intuitive way to say which kinds of plants are of interest. The web site <https://www.gardenia.net/> is a good resource for criteria. Another good resource is the American Horticultural Society at <https://ahsgardening.org/>. Here is my list, though we will probably implement only a subset, leaving the rest for future work.

1. Garden type:
   1. City and courtyard
   2. Coastal garden
   3. Cutting garden
   4. Formal garden
   5. Gravel and rock garden
   6. Informal and cottage
   7. Mediterranean garden
   8. Japanese garden
   9. Modern garden
   10. Traditional garden
   11. Prairie and meadow
   12. Victory garden (from World War II)
2. Region:
   1. Australia
   2. Europe
   3. North America
      1. Canada
      2. United States
         1. Northeast
         2. Midwest
         3. South
         4. Western states
3. Native to:
   1. Australia
   2. United Kingdom
   3. United States
      1. Alaska
      2. California
      3. Midwest
      4. Northeast
      5. Pacific Northwest
      6. Rocky Mountains
      7. Southeast
      8. Southwest
4. Drought tolerant
5. Deer resistant
6. Salt resistant
7. In stock
8. Maintenance
   1. Low
   2. Average
   3. high
9. Sunlight requirements:
   1. shade,
   2. partial sun,
   3. full sun
10. Water requirements:
    1. Low
    2. Average
    3. High
11. How much winter cold can the plant stand in degrees Fahrenheit? (divide into a for the lower five degrees and b for the upper five degrees) Search here if you are not sure of your hardiness zone: <https://planthardiness.ars.usda.gov/>
    1. -60 (zone 1)
    2. -50 (zone 2),
    3. -40 (zone 3),
    4. -30 (zone 4),
    5. -20 (zone 5),
    6. -10 (zone 6),
    7. 0 (zone 7),
    8. 10 (zone 8),
    9. 20 (zone 9),
    10. 30 (zone 10),
    11. 40 (zone 11),
    12. 50 (zone 12)
    13. 60 (zone 13)
12. How much summer heat can the plant stand? How many days in each year is the temperature over 86 degrees Fahrenheit? You can check a 1997 map if you are unsure:<https://ahsgardening.org/about-us/news-press/cool_timeline/heat-zone-map-developed/> The PDF is at <https://solanomg.ucanr.edu/files/245158.pdf>
    1. 0 (zone 1)
    2. 1 to 7 (zone 2)
    3. 7 to 14 (zone 3)
    4. 14 to 30 (zone 4)
    5. 30 to 45 (zone 5)
    6. 45 to 60 (zone 6)
    7. 60 to 90 (zone 7)
    8. 90 to 120 (zone 8)
    9. 120 to 150 (zone 9)
    10. 150 to 180 (zone 10)
    11. 180 to 210 (zone 11)
    12. 210 or more (zone 12)
13. Plant type:
    1. annuals,
    2. bulbs,
    3. cactus (succulents),
    4. climbers,
    5. conifers,
    6. ferns,
    7. fruit,
    8. herbs,
    9. ornamental dresses,
    10. perennials,
    11. roses,
    12. shrubs,
    13. trees,
    14. palms (cycads),
    15. bamboos,
    16. aquatic plants,
    17. orchids
14. Season of interest:
    1. early spring,
    2. mid spring,
    3. late spring,
    4. early summer,
    5. mid summer,
    6. late summer,
    7. fall
    8. winter
15. Planting place:
    1. Arbors, pergolas, trellises
    2. Banks and slopes
    3. Beds and borders
    4. Bog gardens
    5. Edging
    6. Ground covers
    7. Hanging baskets
    8. Hedges and screens
    9. Patio and containers
    10. Ponds and streams
    11. Rain gardens
    12. Small gardens
    13. Underplanting roses and shrubs
    14. Wall-side borders
    15. Walls and fences
    16. Water gardens

When the gardener is selecting filters she needs to be able to specify multiple choices in each category, for example water requirements might be both low and average. Initially all values will be allowed for all filters, and the gardener can choose to limit the list of plants by removing choices.

When the nursery manager enters a plant into the database, he will specify all the choices that pertain to the plant. For example, a plant might be able to stand winters down to -20 degrees Fahrenheit, in which case zones 5 to 13 would be specified, and heat up to 30 hot days per year, in which case zones 1 to 5 would be specified. By default, no choices are selected, so the nursery manager must specify everything about the plant. This is likely to be tedious, so perhaps there should be a way to copy the criteria from an existing plant when entering a new one.

A gardener should be able to specify her hot and cold tolerance and have that be part of her profile, so the filters will default appropriately. In general, the filter settings should be preserved across login sessions.

**Pages**

For the next stage of the design, we describe each page and how it will operate. The pages have names for our convenience, but these names are not visible to the user.

All pages have a header and footer provided by Handlebar. It contains the material that must be placed at the beginning and end of every page. The navigation header can be tailored by Handlebar to display a logout button only if the user is logged in. Each page provides HTML starting with <main> and ending with </main>.

1. Front: The front page contains a brief description of the application and buttons to choose whether to log in, sign up as a gardener or sign up as a nursery manager.
2. Login: If the user chooses to log in, he gets a page which lets him enter his username and password.
3. Sign\_up\_as\_gardener: If the user chooses to sign up as a gardener, she gets a page which lets her enter her desired username, her desired password, and her email address. Upon successful creation of her account, an email message is sent to her email address. We prevent the application from being used for spam by requiring all email addresses be unique, and having no provision for deleting an account.
4. Sign\_up\_as\_nursery\_manager: If the user chooses to sign up as a nursery manager he gets a page which lets him enter his desired username, password and email address. Upon successful signup an email message is sent to the email address. In the future we will also let the nursery manager choose which nursery he is managing, and have a way to create new nurseries.
5. Gardener\_home: Once a gardener is logged in she will see a screen listing all of the plants she has chosen for the current zone of her garden, filtered according to the criteria for this zone. Initially a gardener will have one zone, with a blank name and all criteria will be chosen, so she will see all plants in the database.
   1. There will be an indication if some plants have been omitted due to the criteria, and a way to temporarily bypass the filters so as to see all the plants for this zone.
   2. There will be a button to take her to the criteria page, where she can change the criteria for this zone.
   3. There will be a way to create new zones, choose a zone to be the current zone, change the name of the current zone and delete the current zone. If there is only one zone it cannot be deleted. When a zone is deleted a remaining zone is chosen as the current zone.
   4. For each plant there will be a way to change the number of such plants to be purchased, and a way to delete a plant from the zone.
   5. There will be a search button, which will take her to the search page. The plants she chooses from the search page will be added to the current zone.
6. Gardener\_search: The search page will start by showing all the plants that meet the current criteria. There will be a button to go to the criteria page to change the criteria. Each plant is shown with its formal name, its common name, and a brief description. In the future we may add a picture of the plant. There will be a checkbox on each plant to select it, and when the gardener is done with this page she will return to her main page with those plants added to it. There will also be a button on each plant to bring up more information about the plant.
7. Gardener\_criteria: The criteria page has a pulldown menu for each category, with a check box for each choice. Multiple choices can be specified.
8. Nursery\_mangager\_home: Once a nursery manager is logged in he sees the list of all plants in the database filtered by his criteria.
   1. There is a criteria button so he can change the criteria, and a button to temporarily suspend the filter and let him see all the plants.
   2. If he chooses a plant to edit he goes to the nursery\_manager\_edit\_plant page.
   3. There is a button to add a new plant.
   4. There is a button to let the nursery manager see the selections of a customer by providing the customer’s name.
9. Nursery\_manager\_customer: When the nursery manager asks to see the selections of a customer he is shown all plants selected by that customer without regard to zones and with the desired quantity of each plant displayed. If he is in contact with the customer he can make suggestions and watch the list change as she updates her selections.
10. Nursery\_manager\_criteria: The nursery manager’s criterion page looks just like the gardener’s criterion page.
11. Nursery\_manger\_add\_plant: When the nursery manager adds a plant he is taken to a page with pull-downs for each category, initially all not checked. There is also a place for the name of the plant (formal and common) and the description. In the future we will add a way to upload a picture.
12. Nursery\_manager\_edit\_plant: The nursery manager can change anything about a plant.

**Paths**

The front end communicates with the server through paths. Given the front end design above, we can specify the paths that the server must provide. The path names start with the name of the page they are used from. Paths that include “api” are for passing information in JSON format between the front end and the server. The other paths are for sending HTML from the server to the front end.

1. Get / (the empty path) loads the front page.
2. Get /front/login loads the login page.
3. Post /front/login validates the login, returning a success or failure indication. If there is already a login using this user name, the old login is invalidated.
4. Get /front/signup\_as\_gardener loads the signup\_as\_gardener page.
5. Post /front/signup\_as\_gardener validates the signup, returning a success or failure indication
6. Get /front/signup\_as\_nursery\_manager loads the signup\_as\_nursery\_manager page.
7. Post /front/signup\_as\_nursery\_manager validates the signup, returning a success or failure.
8. Get gardener/api/criteria returns the current criteria in JSON format from the database. The body of the request has the name of the current zone.
9. Put gardener/api/criteria sends the current criteria to the server, which it saves in the database. The body of the request has the name of the current zone in addition to the criteria.
10. Get gardener/home loads the gardener\_home page. The body of the request has a boolean to say that the criteria are suspended, and the name of the current zone.
11. Get gardener/search loads the gardener\_search page.
12. Get gardener/criteria loads the gardener\_criteria page.
13. Get gardener/plant/:plant\_id loads the page for a particular plant. The body of the request includes the name of the page to return to, which the server includes as hidden text in the returned page so a button on the page can navigate to the correct page.
14. Post gardener/api/:plant\_id adds the plant to the zone. The zone is in the body of the request, as is the number of these plants requested.
15. Delete gardener/api/:plant\_id removes the plant from the zone. The zone is in the body of the request.
16. Put gardener/api/:plant\_id changes the number of these plants requested. The zone and number of plants requested is in the body of the request.
17. Get nursery\_manager/home loads the nursery\_manager\_home page. The body of the request has a boolean to say that the criteria are suspended.
18. Post gardener/api/criteria sends the current criteria to the server, which saves it in the database.
19. Get nursery\_manager/api/criteria returns the current criteria in JSON format from the database.
20. Put nursery\_manager/api/criteria updates the current criteria. The server stores it in the database.
21. Get nursery\_manager/criteria loads the nursery\_manager\_criteria page.
22. Get nursery\_manager/customer/:customer\_id loads the nursery\_manager\_customer page.
23. Get nursery\_manager/add\_plant loads the nursery\_manager\_add\_plant page.
24. Post nursery\_manager/api/plant/:plant\_id adds a plant to the database.
25. Put nursery\_manager/api/plant/:plant\_id modifies the information about a plant in the database.

**Future Development**

Expansion of plant database and selection criteria. Support of multiple nurseries. Addition of plant pictures.

Note: this document is maintained in Google Docs, URL <https://docs.google.com/document/d/1GCexcE_5XEOtCtG3s_m5X1G4Os9K-uVu_rLfUuYzvA4/edit>.

**Wireframe**

