PlantHero - Final Report

DECO7180

#Plantlords

Vanessa Ackermann – 46291141

Cameron Edgerton – 42933032

Alvin Yii Yang Jian – 46199108

Chenxiao Liu (Leo) – 44122492

Edward Smart - 45792986

Table of Contents

Description of the Final Product

Overview

PlantHero is an interactive tool that makes gardening easy and in a way that also helps combat the Australian drought crisis. PlantHero has a responsive search function called 'Find my plant' that asks users a selection of prompts and records their responses. After all prompts have been answered, the website will produce a list of waterwise plants that suit the user's unique conditions.

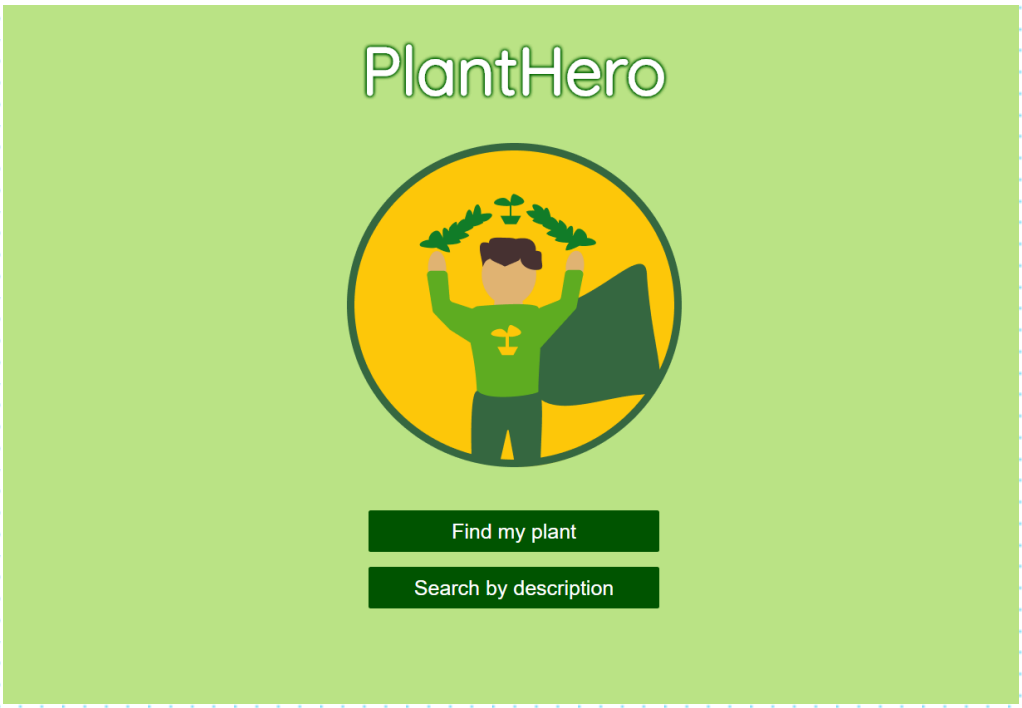

Figure 1: PlantHero home page

PlantHero also has a secondary feature called 'Search by description' which allows users to any plant condition such as name, colour or type and find a list of waterwise plants that match that description. This function can help users in a variety of ways, such as finding out if a plant at their local nursery is waterwise, or if there are any waterwise plants with a particular flower colour that they like.

PlantHero provides a quick and unique way of finding the perfect waterwise plant for you, and requires absolutely no prior horticultural knowledge in order to use it. It truly takes the fuss out of gardening.

Purpose

PlantHero serves two purposes. Firstly, it seeks to make gardening easy in such a way that amateur gardeners can use the website and have confidence that they have chosen the right plant for their needs. Secondly, PlantHero serves the purpose of empowering Australians to fight and manage the Australian drought crisis by encouraging users to plant waterwise plants. By reducing your water usage in the garden, you are helping to conserve

your community's water supplies, and planting waterwise plants is one of the best ways to do this (Queensland Government Department of Energy and Water Supply, 2014).

Target Audience

Description

Our target audience is Australian amateur gardeners (i.e. those with limited horticultural knowledge). There is an assumption that as Australians, our target audience is aware of the drought crisis and the responsibility to be waterwise. Accordingly, these amateur gardeners would inherently be receptive to the idea of ensuring their plants demand minimal water usage.

Demographic

PlantHero is intended to not be limited to a particular demographic, however, we estimate the product will likely be used primarily by Australian individuals aged 20 to 50 years old. This is based on the assumption that people in this age range may be homeowners or renters looking to update their backyard or purchase some indoor plants. Characteristics & Preferences

The archetypal users of the website will be interested in gardening and conserving Australia's natural resources. Ideally, they are a new or hobbyist gardener who enjoys the process of seeing their plants grow and prosper.

As our target audience encapsulates a large demographic with varying technological capabilities, we expect the users will access the website via laptop, desktop or mobile phone.

Goals & Expectations

Being amateur gardeners, users will utilise PlantHero to learn which waterwise plants are suitable to be planted at their desired location. Most of the users will be relatively new to gardening and will be seeking a tool which mitigates the requirement to do in depth plant-suitability research prior to purchasing a plant.

Archetypal Users

Building upon the target audience analysis above, we have developed three personas representing who we feel will be the archetypal users of PlantHero.

Image: Flickr, 2018

Hannah Burns

Age: 29

Occupation: Accountant

Location: Sunshine Coast, Queensland

I like to learn new things

ABOUT

Hannah is an accountant who is a little bit bored with her job. She is looking for a new hobby to distract her from her job and provide something to look forward to doing. She is a very attentive and patient person and prefers to use a desktop computer for browsing the web.

PERSONALITY

Extrovert — Introvert

Sensing — Intuative

Thinking — Feeling

Judging — Percieving

LIKES

-Long walks
-Flower arrangement
-Sewing

DISLIKES

-Poor customer service
-Being late to meetings

TRAITS

Kindness

Adventurous

Optimism

Boldness

GOALS

- Find a new hobby that suits her personality
- Redecorate her home
- Reconnect with nature

FAVOURITE BRANDS

-Kate Spade
-Mecca
-IKEA

Figure 2: Persona #1

Image: CDC, 2020

Clyde Hemmings

Age: 47

Occupation: Chef

Location: Gold Coast, Queensland

Home grown is best

ABOUT

Clyde is an executive chef at a popular Melbourne restaurant which specialises in native Australian ingredients. He is is looking to create his own garden with native ingredients for recipe inspiration, but he is unsure of where to start. He is not very proficient with technology so only uses websites that are easy for him to navigate.

PERSONALITY

Extrovert — Introvert

Sensing — Intuative

Thinking — Feeling

Judging — Percieving

LIKES

-Cooking
-Being creative
-Woodworking

DISLIKES

-Spending his time on unrewarding endeavours
-Noisy neighbours

TRAITS

Kindness

Adventurous

Optimism

Boldness

GOALS

- Create a sustainable garden
- Plant his own food
- Develop some new recipes with the plants in his garden

FAVOURITE BRANDS

-Bunnings
-BCF
-IKEA

Figure 3: Persona #2

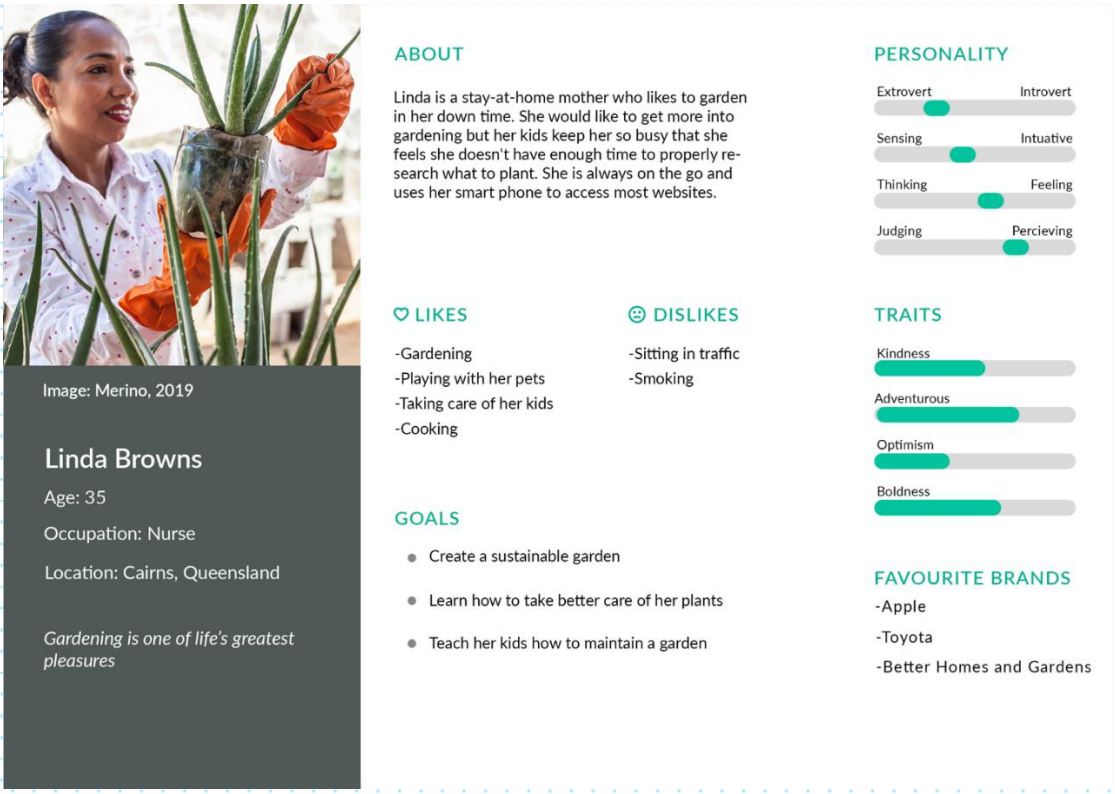

Figure 4: Persona #3

Approach to the Dataset

The dataset we used to build PlantHero was the waterwise plants dataset which we retrieved from the Queensland Government's Open Data Portal (Queensland Government, 2020). The dataset is extensive in that it contains over 4,500 waterwise plant species and 33 columns of additional information about each species of plant.

	A	B	C	D	E	F	G	H
1	Plant ID	Plant Code	Botanical Name	Common Name	Previous Name	Plant Type	Water Needs	Climate Zones
2	2	ABECHI	Abelia chinensis	Chinese Abelia		Shrub	600 to 900mm	Humid Subtropical,Dry Subtropical,Temperate
3	3	ABEFLO	Abelia floribunda	Mexican Abelia		Shrub	600 to 900mm	Humid Subtropical,Dry Subtropical,Temperate
4	4	ABEGRA	Abelia x grandiflora	Abelia		Shrub	600 to 900mm	Humid Subtropical,Dry Subtropical,Temperate
5	5	ABEFRA	Abelia x grandiflora 'Francis Mason'	Golden Abelia		Shrub	600 to 900mm	Humid Subtropical,Dry Subtropical,Temperate
6	6	ABEGOL	Abelia x grandiflora 'Goldsport'	Golden Abelia		Shrub	600 to 900mm	Humid Subtropical,Dry Subtropical,Temperate
7	7	ABEESC	Abelmoschus esculentus	Okra		Annual,Vegetable	2000 to 2500mm	Humid Tropical,Humid Subtropical
8	8	ABEMAN	Abelmoschus manihot	Abika		Vegetable	900 to 1400mm	Humid Tropical,Humid Subtropical
9	9	ABEPAL	Abelmoschus manihot 'Red Vein Palm Leaf'	Palm Leaf Abika		Vegetable	1400 to 2000mm	Humid Tropical,Humid Subtropical
10	10	ABEMOS	Abelmoschus moschatus subsp. Tuberosus	Musk Mallow		Shrub	1400 to 2000mm	Humid Tropical,Humid Subtropical
11	11	ABEMIS	Abelmoschus rugosus 'Mischief'	Rose Musk Mallow		Shrub	1400 to 2000mm	Humid Tropical,Humid Subtropical
12	12	ABEEVE	Abelmoschus rugosus 'Evergreen'	Rose Musk Mallow		Shrub	1400 to 2000mm	Humid Tropical,Humid Subtropical
13	13	ABIPRO	Abies procera	Noble Fir		Medium Tree	1400 to 2000mm	Temperate
14	14	ABRANG	Abroma augusta	Devil's Cotton		Small Tree	1400 to 2000mm	Humid Tropical,Humid Subtropical
15	15	ABUMEG	Abutilon megapotamicum	Big River Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
16	16	ABUVAR	Abutilon megapotamicum 'Variegatum'	Variegated Big River Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
17	17	ABUOXY	Abutilon oxycarpum	Small Flowered Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
18	18	ABUBOU	Abutilon x hybridum 'Boule de Neige'	White Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
19	19	ABUGOL	Abutilon x hybridum 'Golden Fleece'	Yellow Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
20	20	ABUORA	Abutilon x hybridum 'Orange'	Orange Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
21	21	ABURED	Abutilon x hybridum 'Red'	Red Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate
22	22	ABUSAV	Abutilon x hybridum 'Savitzii'	Variegated Abutilon		Shrub	1400 to 2000mm	Humid Subtropical,Dry Subtropical,Temperate

Figure 5: Extract from dataset

We wanted to take advantage of the completeness and complexity of the dataset, but do so in such a way that the complexity was not felt in the user experience. Through our own design and as a result of user testing feedback, we developed two core functionalities (as detailed previously) which we believe take advantage of the richness of the dataset in different ways.

The ‘Find my plant’ function utilises a subset of the dataset to filter specific conditions to provide the user with a tailored recommendation for waterwise plants that suit their conditions. Given the large number of plant species in the dataset and the fact that every column had a distinct value for almost every plant species, we were determined to take advantage of this.

Our approach to the ‘Search by description’ function was somewhat simpler in that we only needed to access two columns of the dataset in order to develop the feature. Regardless of the simplicity of this feature, we believe it serves a core purpose in achieving the vision we have for PlantHero.

Intended Interactivity

The PlantHero website has two core user interactions (functionalities). The first is the ‘Find my plant’ function. The user will be asked to respond to five questions (prompts) by left clicking the answer most appropriate to their scenario. After an answer has been selected, the next prompt will appear until all prompts have been answered. Once all prompts are answered, the website displays a results page with a list of plants that are suitable for the user’s environment based on their responses to the prompts.

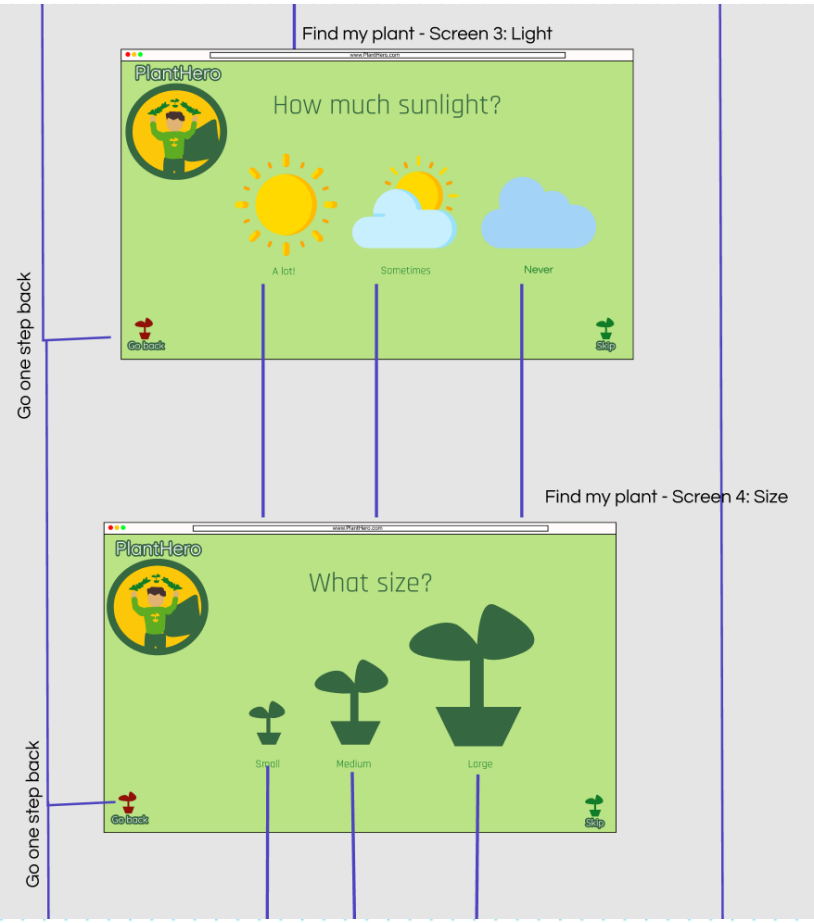

Figure 6: Excerpt from Interaction Plan

The second core user interaction is the ‘Search by description’ function which users will click to take them to a new page which contains a form field. The user will enter the name or description of a plant in the field and then select the ‘search’ option. If the name or descriptor

they searched matches any plant names in the dataset, a confirmation message will appear alongside details of any matching plants. If the value can not be found in the dataset, a message will be displayed advising there are no waterwise plants with a matching description.

Implementation

The final product has a series of fully functional JavaScript API calls and has zero simulated elements. We have broken down the discussion of the implementation according to the two core features of the website.

Find My Plant

The find my plant section is a series of web pages, each containing a single question. The answers that the user chose all need to be stored and subsequently converted into an SQL query that is used to query the dataset via data.qld.gov.au API.

In order to store the user's answers, we created a PHP session variable of the dictionary type. After the user clicks on a particular button, the dictionary is populated with the key - column name in the dataset that this question relates to, and value - value to search for within the column.

For example, in the below code we have a form for our soil type selections. The name and value selections are posted and stored into the session variable after clicking.

```
<form class="inside-outside" action="" method="POST">
    <div class="left-column">
        <div class="left-soil" id="loam">
            <button id="loam-button" class="btn btn-primary" name="soil_type" value="Loam">
                <img src="<?php echo asset_url();?>/img/web/loam.png">
            </button>
            <h2>Loam</h2>
        </div>
        <div class="left-soil" id="clay">

            <button id="clay-button"class="btn btn-primary" name="soil_type" value="Clay">
                <img src="<?php echo asset_url();?>/img/web/clay.png">
            </button>
            <h2>Clay</h2>
        </div>
    </div>
    <div class="right-column">
        <div class="right-soil" id="sand">
            <button id="sand-button" class="btn btn-primary" name="soil_type" value="Sand">
                <img src="<?php echo asset_url();?>/img/web/sand.png">
            </button>
            <h2>Sand</h2>
        </div>
        <div class="right-soil" id="compost">
            <button id="compost-button" class="btn btn-primary" name="soil_type" value="Compost">
                <img src="<?php echo asset_url();?>/img/web/compost.png">
            </button>
            <h2>Compost</h2>
        </div>
    </div>
</form>
```

Figure 7: Soil type interaction

After the values are posted, the controller class redirects the page to the next prompt:

```php
public function soil_type() {

    $this->load->view('templates/header');
    $this->load->view('f_soil');
    $this->load->view('templates/footer');

    if (isset($_POST['soil_type'])){
        $_SESSION['user_input']['Soil Type'] = $_POST['soil_type'];
        redirect(base_url()."find_my_plant/sunlight",'refresh');
    }
```

Figure 8: Page redirection

Once the results page is reached, the session variable is encoded into a JavaScript json object:

```php
<script>var userInput = <?php echo json_encode($session_value); ?>;</script>
<script src ="<?php echo asset_url();?>/js/phpapi.js"></script>
```

Figure 9: Json object creation

Subsequently, this variable is passed into the following function. This function creates the API request URL by iterating through the json object and inserting SQL syntax in appropriate places to generate an SQL query that will be requested of the API.

```javascript
function generateUrl(dict) {
    var url = "https://www.data.qld.gov.au/api/3/action/datastore_search_sql?sql=SELECT DISTINCT \"Common Name\" from \"fd297d03
    var count = 0
    for (element in dict) {
        if (count === 0) {
            url += '"' + element + '"' + " LIKE '%%" + dict[element] + "%%'";
        } else {
            url += " AND " +'"' + element + '"' + " LIKE '%%" + dict[element] + "%%'";
        }
        count += 1
    };
    console.log(url);
    return url
}
```

Figure 10: URL generator

The final URL will usually look something like this:

```
https://www.data.qld.gov.au/api/3/action                    phpapi.js:17:13
/datastore_search_sql?sql=SELECT DISTINCT "Common Name"
from "fd297d03-bf72-40c7-b27e-24cc7023360c" WHERE
"Climate Zones" LIKE '%%tropical%%' AND "Soil Type" LIKE
'%%Compost%%' AND "Light Needs" LIKE '%%Sun%%' AND
"Height Ranges" LIKE '%%1 to 2m%%' AND "Flower colour"
LIKE '%%Yellow%%'
```

Figure 11: Example URL – Find my plant

The following two functions then run. The first function 'CallApi' fetches the data that the user had specified using the URL created in the previous step. This function then calls

'displayResults' which will append the common name of each of the plants to a row on a table in the results html page.

```
function CallApi(url) {
    fetch(url)
        .then(response => {
            return response.json();
        })
        .then(response => {
            console.log(response.result.records);
            if (page == 'results') {
                displayResults(response.result.records);
            }

        });
}

function displayResults(results) {
    if (results.length > 0)  {
        var rows ="";
        results.forEach(element => {
            rows +="<tr>";
            rows +="<td>"+element["Common Name"]+"</td></tr>"
        });
    } else {
        var rows ="<tr><td>There were no waterwise plants to display</td></tr>"
    }
    document.getElementById("results").innerHTML = rows;
}
```

Figure 12: Fetch and display results

We also implemented 'skip' and 'back' buttons. By clicking the 'skip' button, the dictionary and URL are not appended, meaning less criteria are searched and more results appear after the API call. Clicking the 'back' button returns the user to the previous page, and if they choose a different response, the value in the session variable for this prompt will be updated to the new value.

Search by Description

This function operates similarly to the 'Find my plant' function, in that a URL containing the search term specified by the user is created and used to request the API for any data. The primary difference is that we make use of the API's in-built search function rather than using an SQL query:

```
function generateUrl(input) {
    var url = "https://www.data.qld.gov.au/api/3/action/datastore_search?q="+ input + "&resource_id=fd297d03-bf72-40c7-b27e-24cc7023360c";

    console.log(url);
    return url
}
```

Figure 13: API search function

We implemented a form field into which the user enters a characteristic of a plant (such as botanical name, common name or flower colour). When the user clicks the 'search' button, the string they entered is stored into a PHP session variable. This variable is then passed into the function above to generate the URL for the API request. The search will check for any records of the dataset that contain the search term. Similar to the 'Find my plant' function, a JavaScript function then retrieves the resulting plant names and displays them on the page.

If no rows are found which contain the string, a message appears advising no such waterwise plant exists.

Below is an example of what the URL would look like if the user were to search “red flower”:

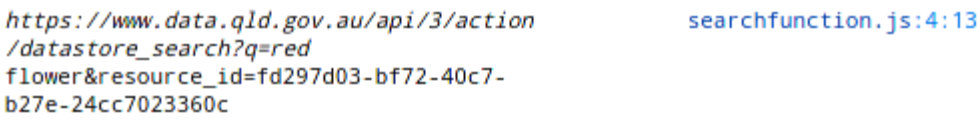

Figure 14: Example URL – Search by description

Third Party Frameworks

Due to the predominantly PHP-based implementation of the website, we utilised the CodeIgniter framework (version 3.1.11) (EllisLab, 2020). CodeIgniter reduces the complexity of front-end and back-end PHP integration by enforcing a model, view and controller (MVC) structure. It contains a variety of powerful libraries which we found effective in our migration from a JavaScript-based website to a PHP-based website.

Final Product Link and Instructions

https://deco7180teams-pfc05t01.uqcloud.net/plant_hero/

When you land on the PlantHero home page you will be presented with two options. Clicking the ‘Find my plant’ option will take you to a new page which asks you to select the coloured button which best reflects your climate zone (per the climate zone map). After doing so, another prompt will appear asking for you to select the type of soil you intend to use while gardening. Alternatively, if you are unsure or do not have a preference for the answer, feel free to click the ‘Skip’ button. You will answer five questions in total before being presented with a results page which details the list of waterwise plants that meet the criteria you searched by.

Clicking the logo at the top left of the screen will return you to the home page. Selecting the ‘Search by description’ button will redirect you to a page with a form field which requests you to enter a characteristic of a plant. Clicking the ‘Search’ button will search the waterwise plants dataset and return a list of waterwise plants whose descriptions correspond with the term you searched by. If no results appear, you will receive a message advising the description you entered does not match that of any waterwise plants in the dataset.

Discussion of Process & Response to Feedback

Challenges, User Testing Feedback & Changes

From the outset, we had a clear plan of how we wanted to structure our design and implementation process. Throughout the process, however, we encountered a number of challenges.

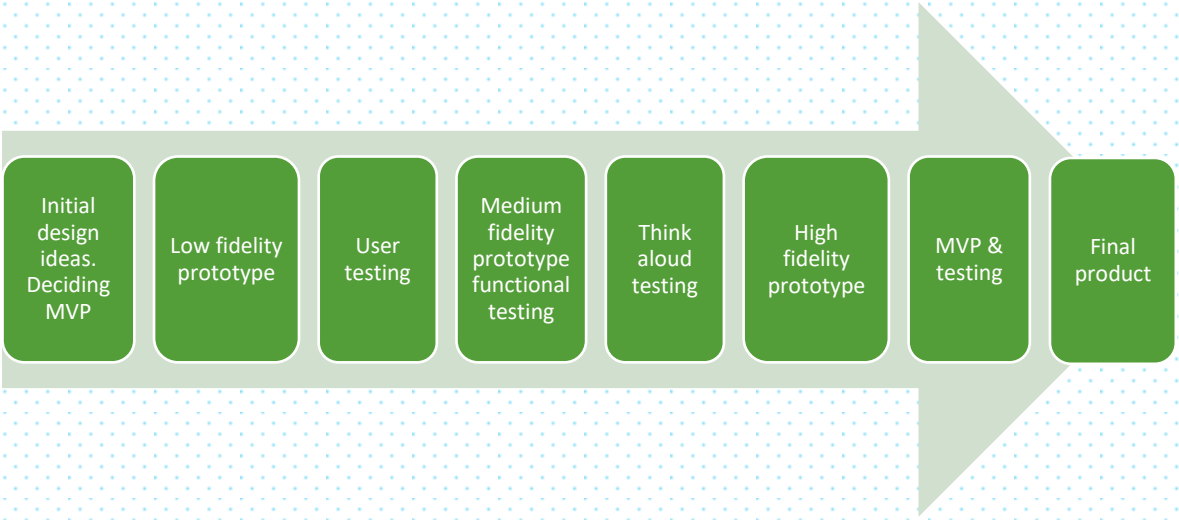

Figure 15: Steps in design process

The main challenge was deciding on a minimal viable product (MVP) that we believed was sufficient to achieve the goals we had set out for PlantHero, but was at a scale that we were able to design and implement fully in the given timeframe. The first iteration of the website had three functions: 'Find my plant', 'Search by name' and 'I'm feeling lucky'. The first function made it to the final product, however, the 'Search by name' and 'I'm feeling lucky' functions did not. The 'Search by name' function was slightly appended to include the ability to search by a variety of plant descriptors (such as type and colour). This was a direct result of our technical implementation methods and we believe it is a much stronger feature on the website than what we initially intended. The appended feature is called 'Search by description' in the final implementation.

The feedback we received from user testing regarding the 'I'm feeling lucky' function was that it did not serve a distinct purpose on the website. We agreed with the feedback and decided to remove it from the scope of the project.

A key challenge we had during the process was ensuring user experience was at the forefront of our design choices. On the outset, we were determined to have a seven-step 'Find my plant' search function which filtered by climate zone, whether the plant will be indoors or outdoors, light levels, size, maintenance requirements, flower colour, foliage colour, and whether it should be a native plant. Through user testing of early prototypes of the website build and our own logical reflections, we decided to narrow the MVP scope to a five-step search function filtering by climate zone, soil type, light needs, plant height and flower colour. The feedback we received was that the indoor-outdoor selection process was a little bit confusing for some users. We also determined that if we restructured the order and wording of the other prompts, we could make it clear to users that plants could be planted inside or outside (mainly by giving them the option to choose a soil type and the plant's light needs), thus negating the requirement to have a separate prompt for this. We also received feedback that the plant type prompt was overwhelming to answer due to the sheer number of possible values the user could select. We also further reflected on our intended target audience (being amateur gardeners) who might not know what some of the different plant types were. For these reasons, we removed the plant type prompt entirely. Given our change in scope for the new 'Search by description' function, we were able to remove the maintenance requirement prompt and native plant prompt as these can now be easily searched for. We also removed the foliage colour prompt because user testers

were often unsure what 'foliage colour' referred to and it also limited the number of results displayed when answered in combination with the flower colour prompt.

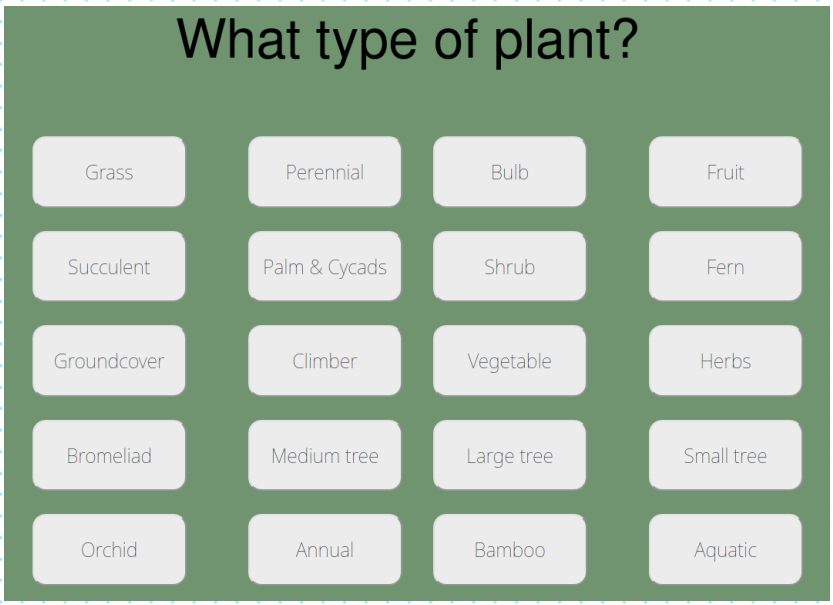

Figure 16: Discarded type of plant prompt

Another issue we had was in relation to the climate zone selection prompt within the 'Find my plant' function. Our initial idea was to have this function utilise a jQuery map plugin or Google Maps API tied to a climate zone API which would take the users co-ordinates and return the corresponding climate zone. We determined that given the scope of the project, it was not an essential MVP component. As a result, we sourced a climate zone map of Australia and created coloured buttons to correlate with the different climate zones on the map and directed users to click the button which best represents their location. We were expecting some issues in user testing in terms of users not understanding the instructions, however, the user testing feedback did not display any negative sentiment regarding the function.

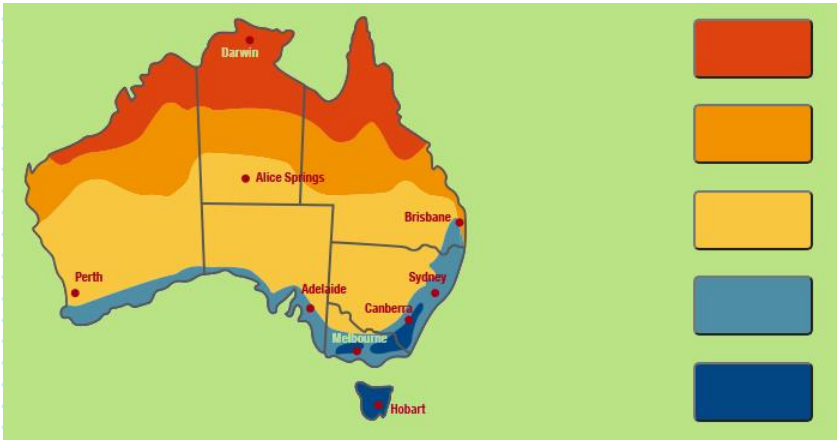

Figure 17: Climate zone map (Healthy Kids Association, 2020) and buttons

Demonstration Feedback & Further Changes

After the demonstration of PlantHero we requested feedback from our peers through the following three questions:

1. What functionality would you like to see added or changed? Why / how?
2. Rate from 1 to 5 (1 being poor, 5 being great) the usability of the 'Find my plant' function. Provide any additional comments.
3. Rate from 1 to 5 (1 being poor, 5 being great) the aesthetic design of the website. Provide any additional comments.

For the most part, feedback was extremely positive. We received a few comments stating that concept of the website was not clear, so we added tooltips on the home page of the final product with a description of what PlantHero is about and how to use it. We feel this was a great improvement with respect to communicating the purpose of PlantHero.

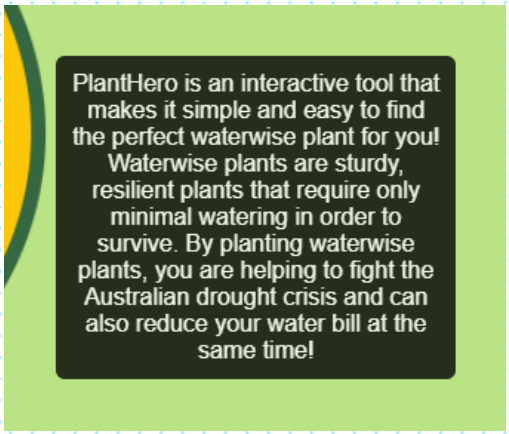

Figure 18: Tooltip explaining purpose

We also received feedback from one individual who suggested we make an educational game focussed on waterwise gardening. Whilst we understand the sentiment from the feedback, we decided PlantHero was to be used as more of a helpful tool for amateur gardeners rather than a game, so we opted not to implement this suggested change.

Reflection on the Final Product

Likes & Dislikes

Overall, we are incredibly happy with the final product. In terms of the aesthetic design, we really like the vibrant, cartoonish theme and icons. We feel like these make the website fun to use and greatly enhance the user experience by maintaining a sense of continuity between pages. We would have liked the 'soil type' page in the 'Find my plant' function to have used similar cartoonish icons, however, we were unable to source any suitable icons and the ones we drew were not as visually appealing as we had hoped. Instead, we opted to use photographs which were sourced online which we all agreed provided a better depiction of the different soil types than any of the other options we explored.

We also like that we kept the website simple in terms of not trying to implement too many functions. We were very focussed on ensuring the purpose of the website was met and that every inclusion on the website contributed to communicating the goals of PlantHero in a meaningful way. We like the 'Find my plant' and 'Search by description' functions because they are both useful in their own right.

Another aspect of the website we like is the function that skips ahead to the results page if less than 10 results will appear after a few prompts have been answered. We feel this improves the user experience because it saves them time answering unnecessary prompts.

We dislike that the results page displays a large number of results in some instances. Optimally we would have liked to limit the number of results displayed, or at least format the results to display in a more appealing way, but we were unable to implement this in time.

Successful & Unsuccessful Aspects and Improvements

Prior to finishing the final product, we set ourselves some criteria for success which, if met, would mean we could call the project a success.

We conducted user testing of the final product on 7 individuals who provided us with feedback which we were able to document. The following is the breakdown of the user testing results against the success criteria:

Success Criteria	Actual measurement	Success? (Yes / No)
Concept		
- 90% of users agreed the website was a useful tool for finding waterwise plants	100%	Yes
Usability & Functionality		
- 95% of users were able to find a plant using the 'Find my plant' function	100%	Yes
- 95% of users were able to find a plant using the 'Search by description' function	100%	Yes
- Average of task difficulty amongst users was 2/5 or less (1 being easy, 5 being hard)	2/5	Yes
User Experience & Design		
- 100% of users reported no major navigation issues	100%	Yes
- 75% of users rated aesthetic 5/5 (1 being poor, 5 being great)	85.7%	Yes

- 50% of users expressed interest in doing more gardening after using the website	71.4%	Yes

Figure 19: Results of final round of user testing

As can be seen, the responses from the user testing feedback show us that the overall product was a success, however, we as a team have reflected and identified a number of areas which we believed were more successful than others.

As previously mentioned, the purpose of PlantHero is to make gardening easy and in a way that also combats the Australian drought crisis. We agree with the user testers and really feel like we have communicated this purpose in the functionality that has been implemented.

The 'Find my plant' function was for the most part a success. Overall, it serves a distinct purpose and provides a useful tool for users to find a waterwise plant to suit their environment. In our opinion, the only unsuccessful aspect of this functionality was the 'climate zone' selection page. As it stands, the user has to manually read a climate zone map of Australia and click a button which corresponds to the climate zone of their location. Optimally, we would have loved to have integrated a location API which returns the co-ordinates of the user's location and integrate that with a climate zone API which reads the co-ordinates and returns the corresponding climate zone. If we had managed to get this functionality operational, we feel like the user experience would have improved.

We also think the responsive design was a success. In today's age, the majority of users would likely use the website on a smart phone, so providing the correct scaling for different screen sizes was incredibly important.

Perhaps the most impactful behind-the-scenes success was the decision to alter the website from being predominantly JavaScript-based to predominantly PHP-based. This change early on in the implementation process benefited us greatly because it made the API call and variable storing during the 'Find my plant' activity incredibly simple. We feel like making this change at the time we did probably saved us a lot of time and allowed us to implement the core functionality at a high standard.

Regardless of all the successes, we were unsuccessful in achieving the MVP we set out to deliver. Our MVP included a Wikipedia or Google API which provided additional information and images about each waterwise plant appearing on the results page. Unfortunately, we were unable to get this this function operational in time and feel as though the user experience suffers as a result.

The following is a list of improvements and additional implementations we would make to the final product if we had more time:

- Integrate the location API to collect user's co-ordinates. Feed these co-ordinates through a climate zone API which would return the corresponding climate zone. Filter that value against the climate zone column in the waterwise plants database using our API call. This was not a part of our minimum viable product, however, as detailed above, we believe this would have had an impact on improving the user experience.
- Integrate Wikipedia or Google API call to get more information and images for each species of plant in the results page.

- Integrate store recommendations in the results page so that users can easily find where they can purchase a plant.
- Limit the results page to display the first 10 results only, with the option to expand the list to view more. We think this would look a lot better than the large list of results it currently displays.

References

Queensland Government. (2020, January 1). *Waterwise plants*. Retrieved from Queensland Government Open Data Portal: https://www.data.qld.gov.au/dataset/6162d790-a123-49c7-b0f0-17e648024404

Queensland Government Department of Energy and Water Supply. (2014). *Being waterwise at home*. Retrieved from waterwise Queensland: https://www.dnrme.qld.gov.au/__data/assets/pdf_file/0007/1407553/being-waterwise-home.pdf

Website Assets

Claire, R. (2020). *Brown Dried Leaves on Ground*. Retrieved from https://www.pexels.com/photo/brown-dried-leaves-on-ground-4846396/

EllisLab. (2020). *CodeIgniter*. Retrieved from CodeIgniter: https://codeigniter.com/

Freepik. (2020). *Flaticon*. Retrieved from Flaticon: https://www.flaticon.com/

Haines, H. (2020). *Brown Soil in Close Up Photography*. Retrieved from https://www.pexels.com/photo/dark-dirty-texture-dust-5209216/

Healthy Kids Association. (2020). *Vegetable Science: Stage 3*. Retrieved from Healthy Kids Association: https://healthy-kids.com.au/vegetable-science-stage3/index.html

Jamieson, C. (2018). *Close-Up Photo of White Sand*. Retrieved from https://www.pexels.com/photo/close-up-photo-of-white-sand-1478450/

Wendt, L. G. (2020). *Brown and Black Concrete Wall*. Retrieved from https://www.pexels.com/photo/climate-desert-dry-dust-5597653/