Plant Oasis – by Daniel Cullen, Hubert Bukowski & Jibril Abdulazeez

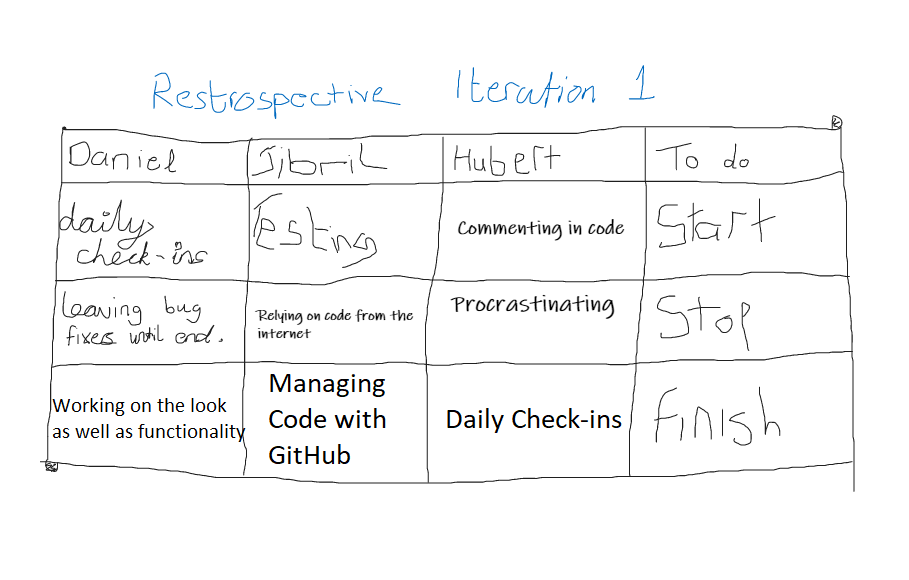
Table of Contents

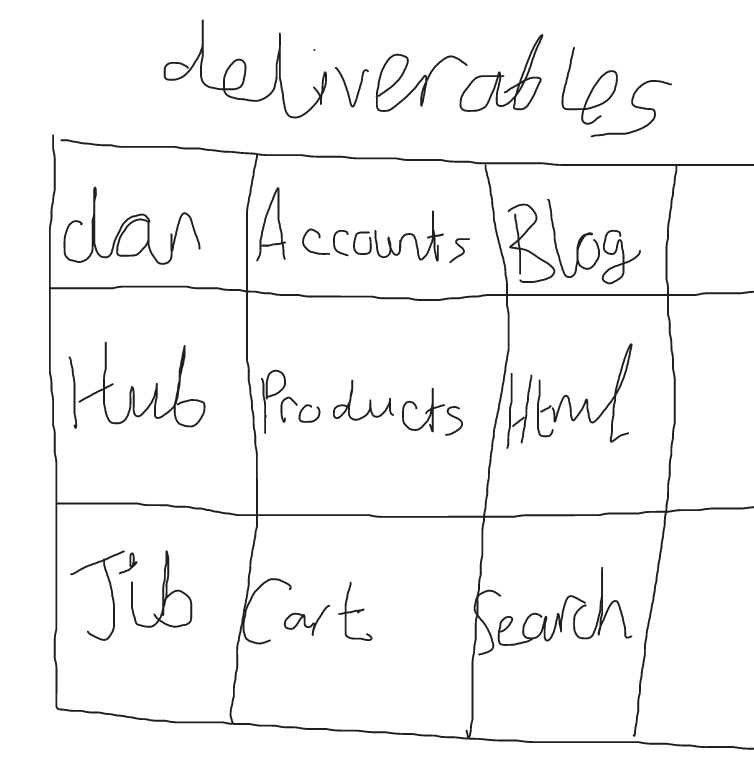
1. Planning
2. Planning
3. Planning
4. Design
5. Design & Tests
6. Tests
7. Tests
8. Tests
9. Tests, Issues & Solutions

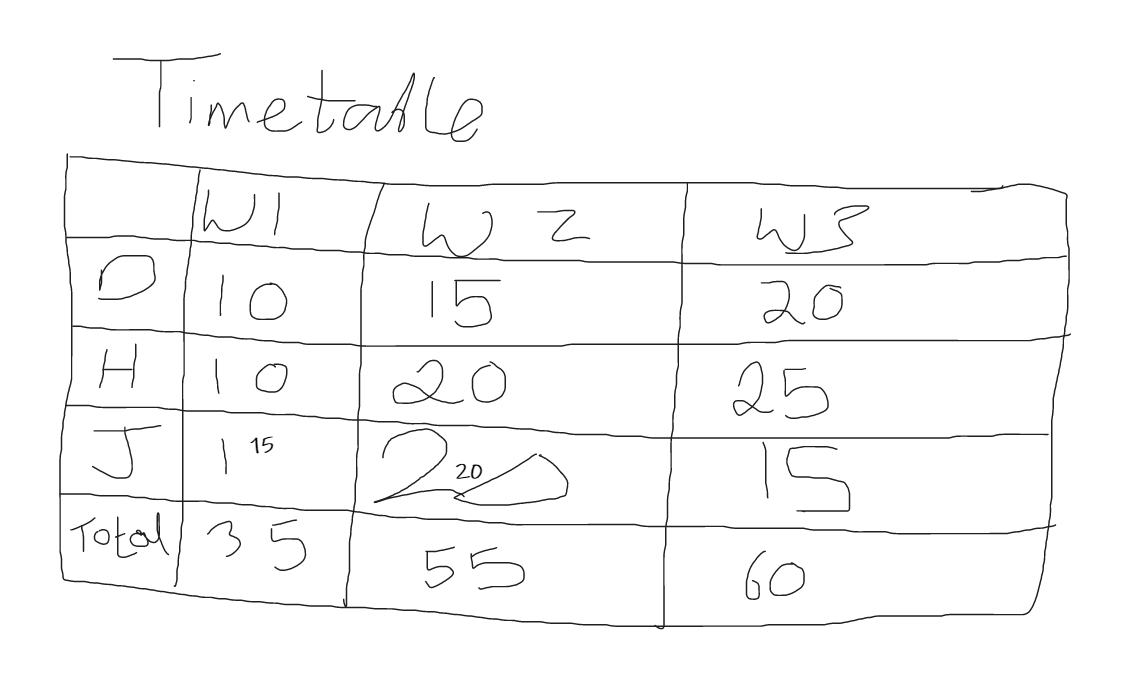
Iteration 1 – Prototype Report – Plant Oasis

Planning:

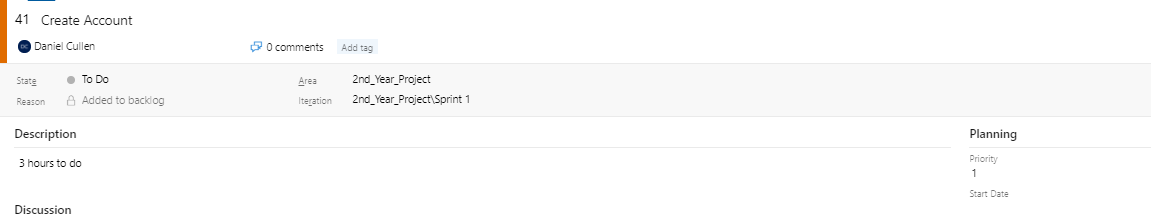
We all joined the teams meeting workshop that was on Tuesday 16th of February, we spoke about the retrospective for iteration 1 which can be seen below. We wrote what we thought we needed to start doing more, what we needed to stop doing and what we believed we should continue with. We had a daily check-in which we spent talking about what we intended to do for the day. During the meeting we also decided on how we would split up the work that needed to be done towards the first iteration. Daniel was assigned with doing the accounts and the blog, Jibril was assigned with doing the cart and search function and Hubert was assigned with doing the products and the html of the website. We also spoke about the maximum number of hours that we believed we could put towards the iteration weekly; we were unable to achieve these hours but we are satisfied with the amount of work and time that we put into the website.

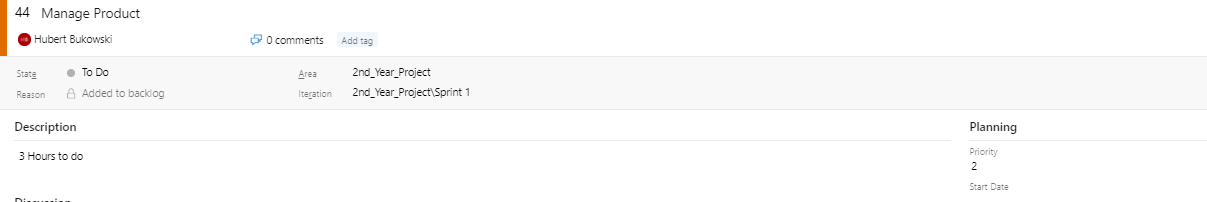






We used azure devops to manage our progress on the website as we pushed and pulled from the repo. We would notify the members of the team when we pushed a new feature to the repo so that the other members could pull it and update their features to work with it. At the planning phase of iteration 1 we prioritized the features that we would be implementing into the website so that we had a path to follow on what we believed were the essential components of the website. This allowed us to focus on the important features before attempting to implement extra features that we thought would add to iteration 1. We used the boards feature of azure devops to categorize these features by priority, an example of this can be seen below displaying the 2 priorities we set on the features.





Design:

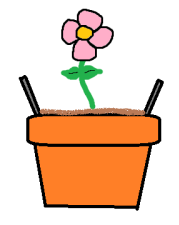
Before Hubert began on the html of the website, we worked on a design for the base html which we all agreed on. We also worked on some logos for the website before we decided on the product which we were happy with. These prototype designs can be seen below which are also compared to what we have finished with.

Diagram

Description automatically generated







We thought it would look more professional if we used a high-quality logo instead of drawing our own, so we used an online logo generator that helped us create a logo that we believed would suit the websites aesthetic and products.

Tests:

We decided to split our tests up between the team based on the features that the team member worked on so Daniel did tests on account and blog, Hubert did tests on the products and html and Jibril tested the cart and search functions. An image of the test cases can be seen below showing the results of each of our tests. The tests were conducted through the use of selenium ide and selenium web driver which we believed was an efficient and robust way of testing the features through different programs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Matrix** | | | | | |
| **Test Case No** | **Test Case Name** | **Input** | **Expected Outcome** | **Actual Outcome** | **Result (Pass/Fail)** |
| 1 | signup | Customer01, 20, [customer@gmail.com](mailto:customer@gmail.com), jinglebells, jinglebells | Taken to the sign in page after account is successfully created | Taken to the sign in page after account is successfully created | Pass |
| 2 | signin | Customer01, jinglebells | Taken to the homepage logged into the account | Taken to the homepage logged into the account | Pass |
| 3 | signout | Click profile button then click signout | Taken to the homepage logged out of the account | Taken to the homepage logged out of the account | Pass |
| 4 | profileView | Click profile name then click profile button | Taken to profile information page | Taken to profile information page | Pass |
| 5 | updateProfile | Click profile name then click profile button, increase age, name or email then click update | Taken back to profile information page displaying updated information | Taken back to profile information page displaying updated information | Pass |
| 6 | deleteProfile | Click profile name then click profile button then click delete button and click yes you want to delete account | Taken back to signup page since account is deleted | Taken back to signup page since account is deleted | Pass |
| 7 | addBlog | Click add new blog, new blog as title, choose blog slug, choose username, input blog content, choose to publish the blog, click add. | Taken back to blog list page, blog will be added to the viewable blogs | Taken back to blog list page, blog will be added to the viewable blogs | Pass |
| 8 | updateBlog | Click edit blog, change the title, change the content and change the slug, click update | Taken back to blog list page, blog will be updated in the viewable blogs | Taken back to blog list page, blog will be updated in the viewable blogs | Pass |
| 9 | deleteBlog | Click delete blog, click confirm | Taken back to the blog list page and the blog will not be present | Taken back to the blog list page and the blog will not be present | Pass |
| 10 | addComment | Click to read a blog, enter name, email and comment content, click comment | A message will appear saying that the comment is awaiting moderation, once moderated it will appear on the blog | A message will appear saying that the comment is awaiting moderation, once moderated it will appear on the blog | Pass |
| 11 | navbarNavigation | Click blog, click cart, click admin, click add item, click username, click edit profile, click home, click logo | All links in the navigation bar work as they should and redirect as appropriate | All links in the navigation bar work as they should and redirect as appropriate | Pass |
| 12 | categoryNavigation | Click Indoor Plants, Click outdoor plant, Click plant pots, click tools | All category links work as they should and redirect to their respective product lists | All category links work as they should and redirect to their respective product lists | Pass |
| 13 | productNavigation | Click product 1, click home, click product 2, click outdoor plants, click product 3, click plant pots, click product 4 | All the product links redirect to their respective product detail views | All the product links redirect to their respective product detail views | Pass |
| 14 | addProduct | Click admin, click add product, enter name, enter alt name, enter description, enter price, enter stock, click submit, select category, click submit | A new product with the specified fields is created and the user is redirected to the home page | A new product with the specified fields is created and the user is redirected to the home page | Pass |
| 15 | editProduct | Click product, click edit product, set stock to 0, click back to product button | The product stock has been reduced to 0 and the product page is displayed appropriately | The product stock has been reduced to 0 and the product page is displayed appropriately | Pass |
| 16 | RemoveProduct | Click Product, click delete product, click cancel, click product, click delete product, click confirm | The product is removed from the database and is no longer displayed on the home page | The product is removed from the database and is no longer displayed on the home page | Pass |
| 17 | searchProduct | Click searchbar, input “Bonsai Arts”, press enter to search | The product “Bonsai Arts” is displayed on page with the title “Bonsai Arts: Search Results” | Search results page is returned | Pass |
| 18 | searchProducts | Click searchbar, input “a”, press enter to search | The page should return all products containing “a” in their product name | Search results page returns every product with an “a” listed in the product name. | Pass |
| 19 | cartProduct | Click indoor plants, click product of choice, click add to cart | Product should add to cart, redirect the user to the cart page and display a shopping cart icon with the label (1) | User is redirected to shopping cart page with the product added, listed and the shopping cart icon appears to reflect the number of items in cart | Pass |
| 20 | increaseQuantity | Click shopping cart, click + icon below product item name | Cart should update price and quantity to reflect the increased number of products added to cart | Cart pricing and quantity updates to reflect the new number of products in cart | Pass |
| 21 | decreaseQuantity | Click shopping cart, click – icon below product item name | Cart should update price and quantity to reflect the decreased number of products added to cart | Cart pricing and quantity updates to reflect the new number of products in cart | Pass |
| 22 | emptyCart | Click shopping cart, click x icon next to the subtotal | Cart should empty and a new title labeled “Your shopping cart is empty” should appear, alongside the removal of the numbered label beside the shopping cart icon in the navbar | User page now displays “Your shopping cart is empty”, cart is emptied and a number label is no longer displayed alongside the shopping cart icon in the navbar | Pass |
| 23 | checkStock | Click product that is out of stock | User should be unable to add product to cart as it is out of stock | User is unable to cart the out of stock product | Pass |

Issues:

* A major issue that we came across was that the slugs for categories and products would not work which caused the dispatcher to be unable to distinguish between urls in the shop, blog and cart apps.
* Another issue that we came across was that the url for blog and cart would crash the website when accessed from any part of the site other than the homepage. This was caused by the url name being appended to the end of the url instead of the url changing.

Solutions:

* The first issue was fixed by Hubert as he added the app name before the slug in each of the urls to help distinguish them. An example of the difference of the urls can be seen here path('<slug: slug>/', views.allProdCat, name='products\_by\_category') compared to path('shop/<slug:slug>/', views.allProdCat, name='products\_by\_category').
* We fixed the second issue by changing the url from \cart to {% ‘cart\_detail’ %} and the same was done for the blog app.

An extra feature we plan on adding in iteration 2 is creating a new customuser called manager that will be used instead of admin for the front end editing and deleting.