CS335 End Of Year Assignment

By: Josh Tyrrell (16340431) and Cormac Gilbert (17449782)

Retail

User stories: Josh Tyrrell

UML Diagrams: Cormac Gilbert

UI Mock Up: Josh Tyrrell

System Tests: Cormac Gilbert

Our client has come to us with a vague specification regarding a mobile app for selling clothing. They would like us to make an app which is easy to navigate, which will increase the chance of first time buyers. The clothing on the app will be split into categories which we have to find a way to allow the customer to easily navigate between. We decided to also have a blog feature, so users can engage with one another.

Furthermore, to increase chances of return customers and loyalty, we decided to use gamification by implementing a point system which works as follows: Every time a user purchases an item, they receive an amount of points based on the price of that item. These points can then be used to get discounts on any item for sale on the app.

Our Target demographic: Men and women aged 16-30. People who are more likely to use an app to purchase clothes over a physical store. These people would also be more susceptible to gamification and feel more rewarded receiving points for making purchases, thus making them more likely to reuse the app.

User Stories:

Before we decide exactly how to structure the app, it is important that we consider what the users of the app might want. This set of user stories to allow us to better understand the software requirements of the app:

Title: Opening app
User Story: As a user who wants to use the app. I want to be able to open the app.
Conversation: Once the user has installed the app from the app store, they must find the app icon and then press it.
Acceptance Criteria: When the app icon is pressed by the user, the app opens to the homepage.

Title: Shopping for items
User Story: As a user, I would like to be able to buy products
Conversation: Once the user is on the homepage, there will be a large button labelled “Clothing”. Once pressed the app will display all of the items currently on sale on the website. The user can then navigate the pages in some way to view all the items for sale.
Acceptance Criteria: When the Clothing button is pressed, the user will be shown all items currently for sale.

Title: Finding a specific type of product
User story: As a user, I want to be able to find a specific type of product easily
Conversation: Once the user is on the Clothing page, there will be a button labelled “Category”. Once clicked this will display a drop down menu listing the different types of clothing for sale on the app. Once the user clicks on one of these types, the app will display every item for sale on the app that fits this category. The user can then scroll down the page to view all said items.
Acceptance Criteria: When the category button is pressed, the drop down menu is displayed. The user can click one of the categories and then they will be shown all of the items for sale in that category.

Title: Finding wish list
User Story: As a user, I would like to be able to view my current wish list.
Conversation: Once the user is on the homepage, there will be a button labelled “Account”. Once pressed the app will display the account home page. There will be a button labelled “Wish list”. Once pressed the app will display all the items currently in the user’s Wish list.
Acceptance Criteria: Once the account button is pressed, and subsequently the wish list button, the user will be shown the items in his wish list

Title: Adding item to wish list
User Story: As a user, I would like to be able to add items to my wish list
Conversation: Once the user is browsing any of the items for sale on the app, in the top right corner of every item there will be a plus sign. Once pressed the item will be added to their wish list.
Acceptance Criteria: Once the user presses the plus button at the top right of any item for sale, that item will be added to their wish list.

Title: Finding out about app
User Story: As a user, I want to be able to find out more about the app.
Conversation: From the homepage, the user can click a large button labelled “About us”. From there the user will be taken to a page which explains app policy, how it got started etc.
Acceptance Criteria: Once the About us button is pressed, the user will be shown a description of the app.

Title:
Searching for a specific item

User Story:
As a user, I would like to be able to look for a specific item using a keyword.

Conversation:
Once the user is on the homepage, there will be an icon at the top of the screen clearly denoting a search function. Once pressed by the user, they will be shown a keyboard and they can type in the keyword of choice. Afterwards they can click search and they will be shown all of the items that match the keyword they chose.

Acceptance criteria:
Once the user presses the search icon and types in their keyword, then hits the search button, they will be shown all of the matching items.

Title:
Tracking an order

User story:
As a user, I would like to be able to track a placed order

Conversation:
When the user is on the homepage, there will be a button labelled “Account”. Once clicked they will be shown their account homepage and there will be a button labelled “Orders”. If they click this button they will be shown a list of their orders. If they click on a specific order, they will be shown the items in the order, along with the order status.

Acceptance criteria:
When an order is selected, the items in the order along with the order status should be displayed to the user.

Title:
Adding to cart

User Story:
As a user, I would like to be able to add to my shopping cart.

Conversation:
When the user is in either the clothing or wish list section of the app, if they click into a specific product they will be shown an image of the product, along with its name, price and description. There will be a button at the bottom of the page labelled “Add to cart”. Once pressed, the item will be added to the users shopping cart, and the customer will be notified.

Acceptance criteria:
When add to cart is pressed, item will be added to cart, and customer notified.

Title:
Placing an order

User Story:
As a user, I would like to be able to place an order

Conversation:
When the user is on any page on the app, there will be a shopping cart icon on the top right of the screen. Once clicked the user will be taken to their shopping cart. If their shopping cart is empty, they will be notified of this and shown a button labelled “Shop” which will take them to the shop if pressed. If there is anything in their shopping cart, they will be shown the contents and there will be a button labelled “Place Order”. When pressed, they will be taken to a page where they will have to enter their home address, shipping address and card details. Once complete, they will have the chance to review their order before pressing a button labelled “Complete order”, which will complete the order.

Acceptance Criteria:
When the shopping cart icon is pressed, the user will be taken to their shopping cart. From there they can choose to place an order if the shopping cart isn’t empty. When they fill out all their details, they will review order then complete when “Complete order” button is pressed

Title:
Accessing blog

User Story:
As a user, I would like to be able to access the app’s blog.

Conversation:
Once the user is on the home screen, there will be a button labelled “Blog”. Once clicked the user will be shown the app’s blog.

Acceptance criteria:
Once blog button is pressed, user is taken to blog

Title:
Posting on blog

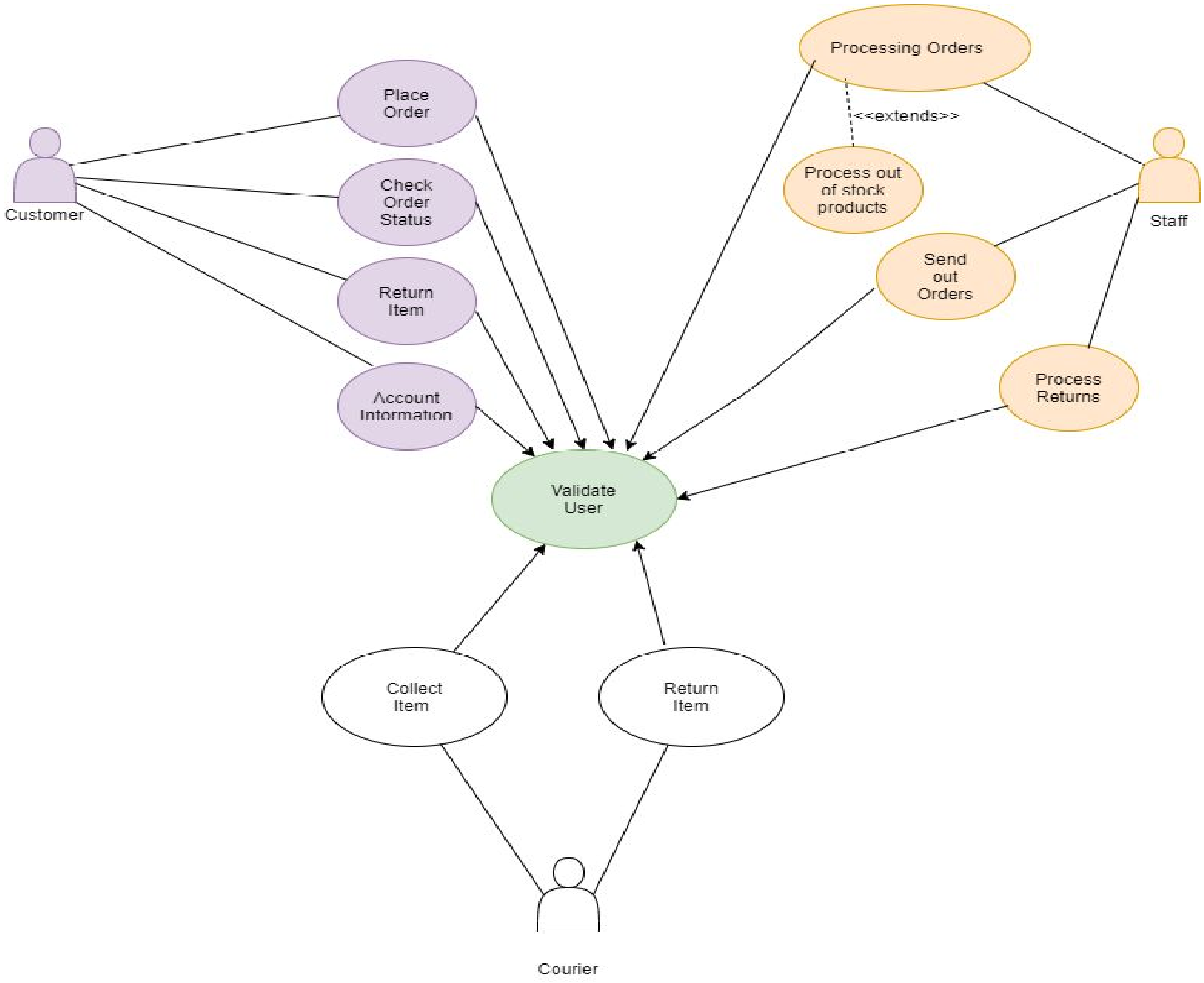
User story:
As a user, I would like to make a blog post

Conversation:
Once the user is on the blog screen, there will be a plus button in the top left of the screen, when clicked, a keyboard will appear and the user will be able to type up a post. Once they are finished, they can click a button labelled “Submit”, and their post will be submitted, and they will be notified.

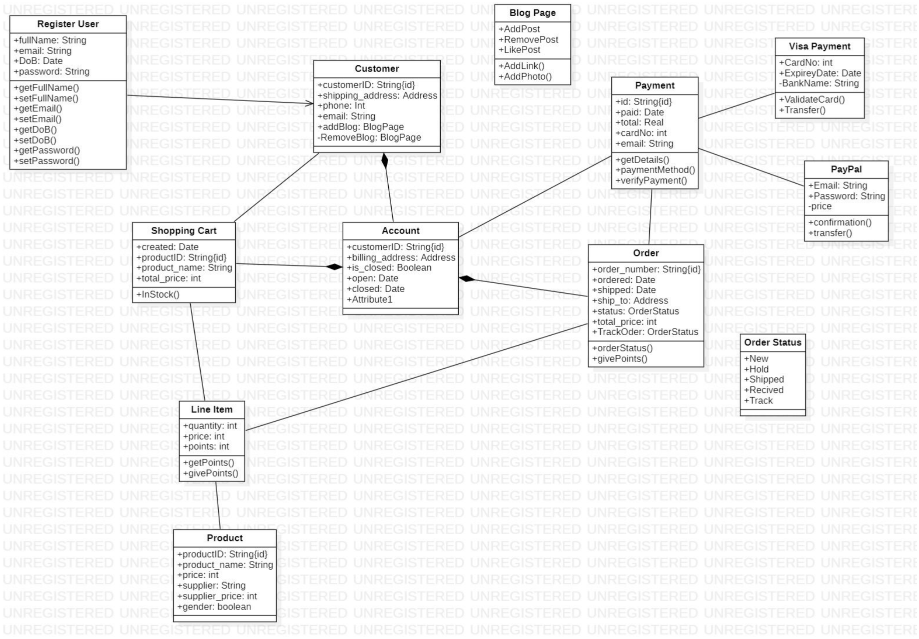
Acceptance criteria:
When the plus button is pressed, keyboard must pop up. When post is typed up, and submit button is pressed, post must be submitted on blog and user notified.

Formal Descriptions/UML Diagrams:

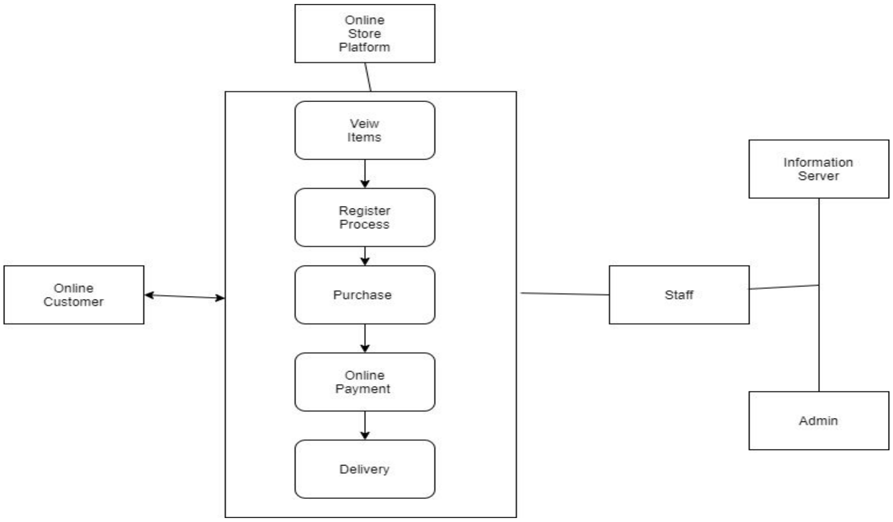
Use Case Diagram:
In this use case diagram we are only showing the functional requirements of each use case. Our top level use case is “Validate Customer”. After each operation that happens on the software, it checks to see if the user is allowed to do each action and then validates. It then relays this information to another user. As an example we would have a customer place an order and then have it validated. Once the user signs in and authenticates themself, we can relay the information onto the staff. This streamlines all operations in the software allowing for easy usability.



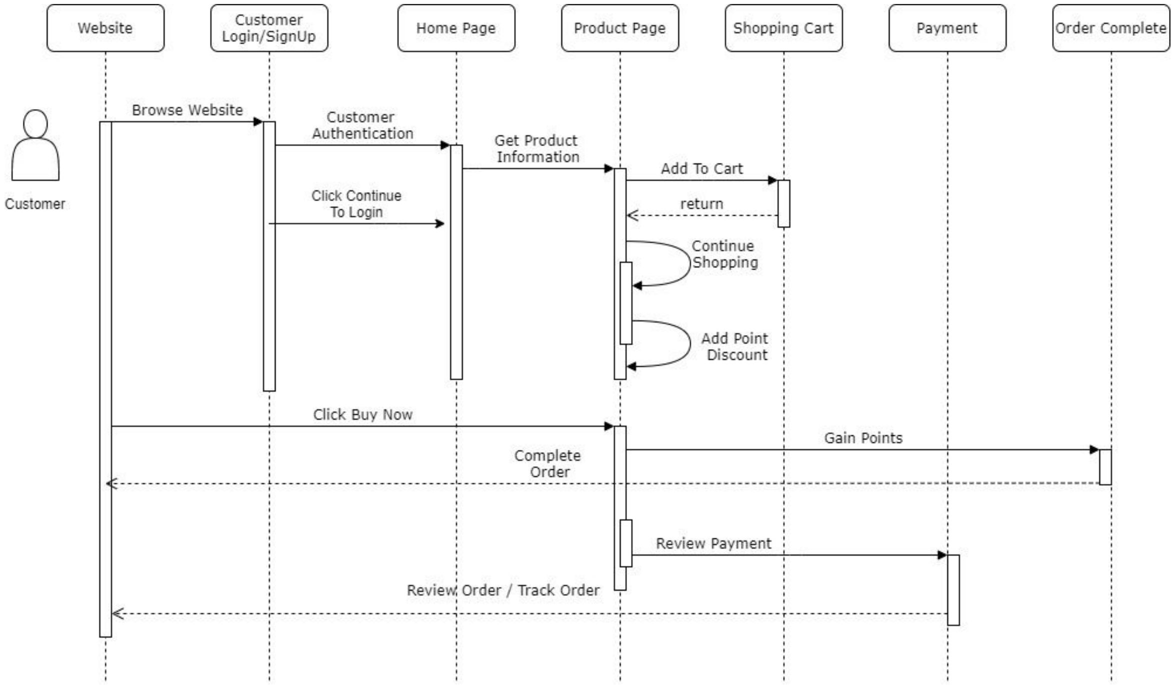
Class Diagram: In this class diagram we are showing each of the processes and their relationships with one another. It also allows us to find each attribute and operation that is needed in the development process. It gives us a better understanding of the schematics of each operation. Class diagrams are easy to understand and use basic language which is very helpful when showing it to any of the stakeholders.



Architecture diagram: An architecture diagram is a graphical representation of the basic set of concepts in the software system. It includes the main elements and components of the architecture.

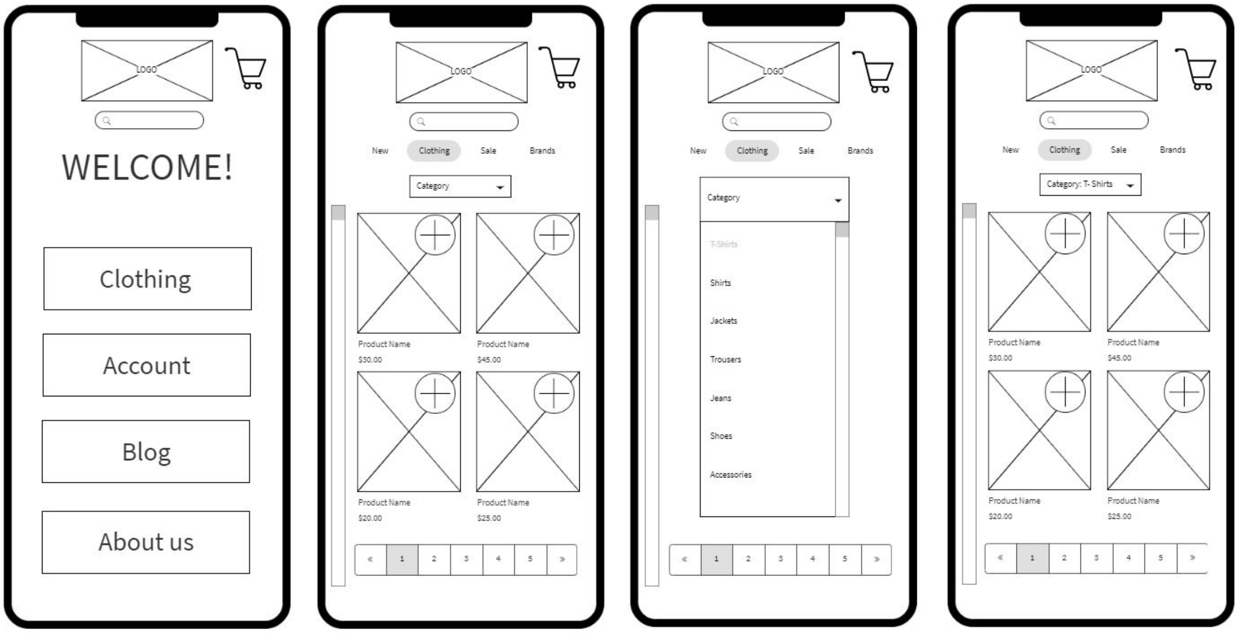


Sequence Diagram: In our sequence diagram we show objects interactions arranged in a time sequence. We breakdown each operation on our diagram and see the outputs being relayed into our software and each function that happens. When our customer is using the website we have them log in before being able to purchase any products. It allows us to understand the process that needs to happen when creating our software. Creating a good sequence diagram is extremely important for early stages of development as it allows us to extract requirements and to fully understand how our software would interact with customers.

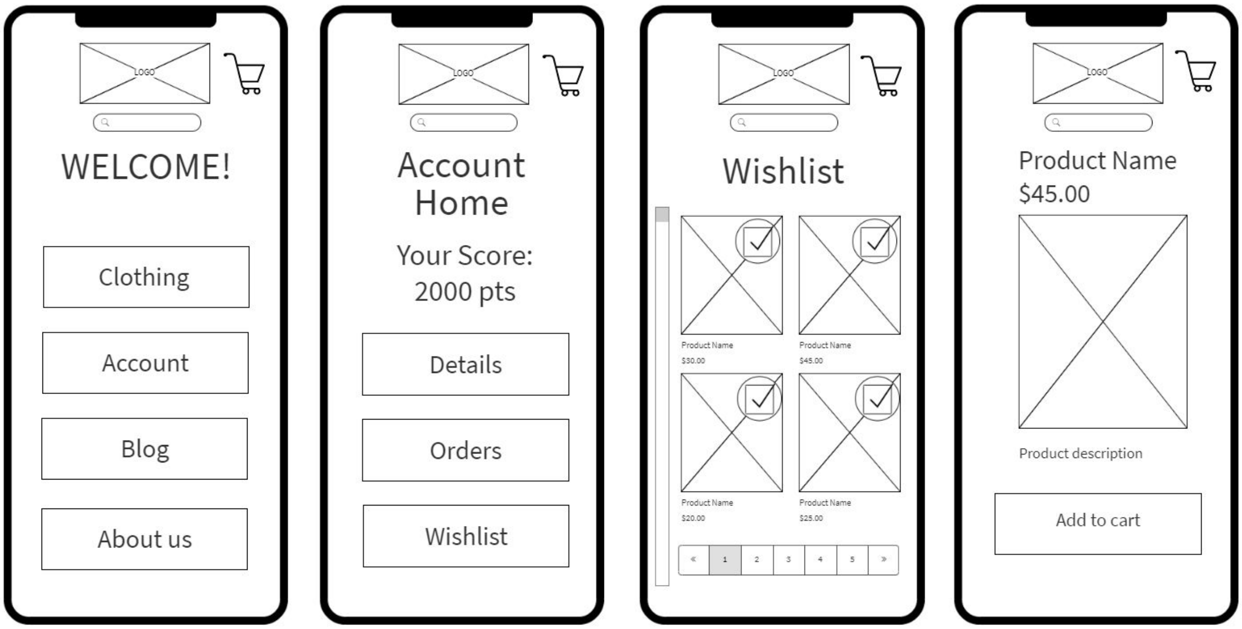


User Interface:

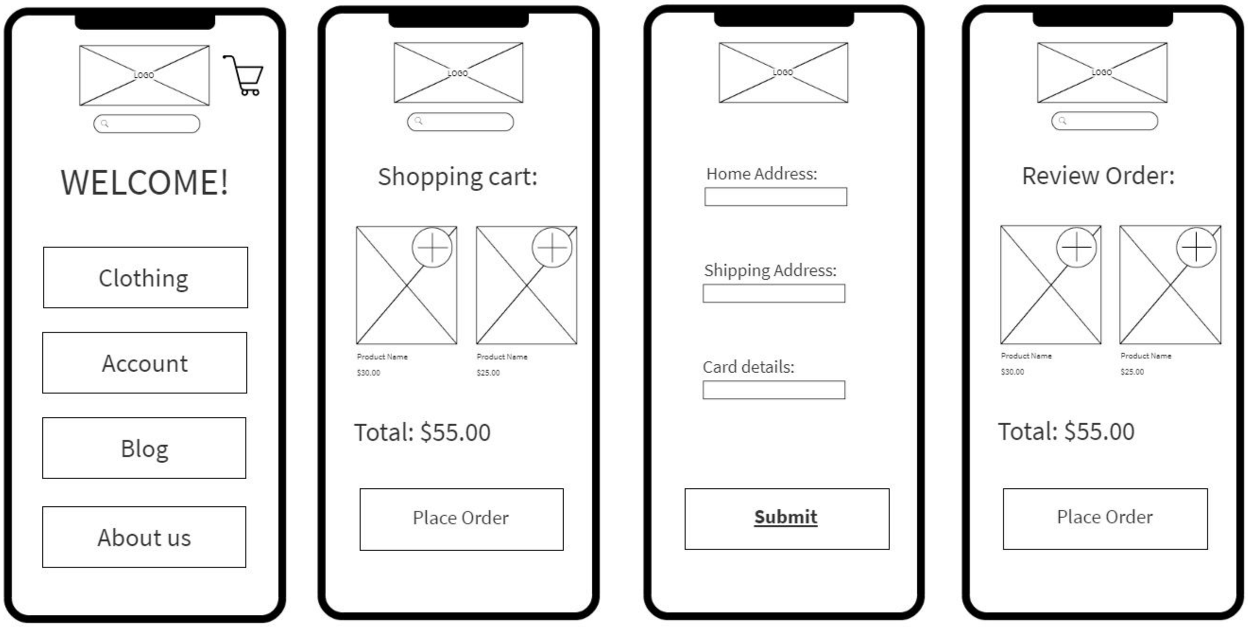
As mentioned earlier, it is critical that we make the user interface as practical and simplistic as possible, while still being able to carry out all of the functionality required. When designing a retail based app with as many features as this, it is critical that everything is simplistic and easy to find for the user. It is important not to bombard the user with information, so we decided that simple navigation menus with minimal options would be best. We decided to split the app into four distinct sections, Clothing (The items on sale on the app), Account, Blog, and About Us.
I (Josh Tyrrell), designed the Mock Up User Interface to satisfy these requirements.



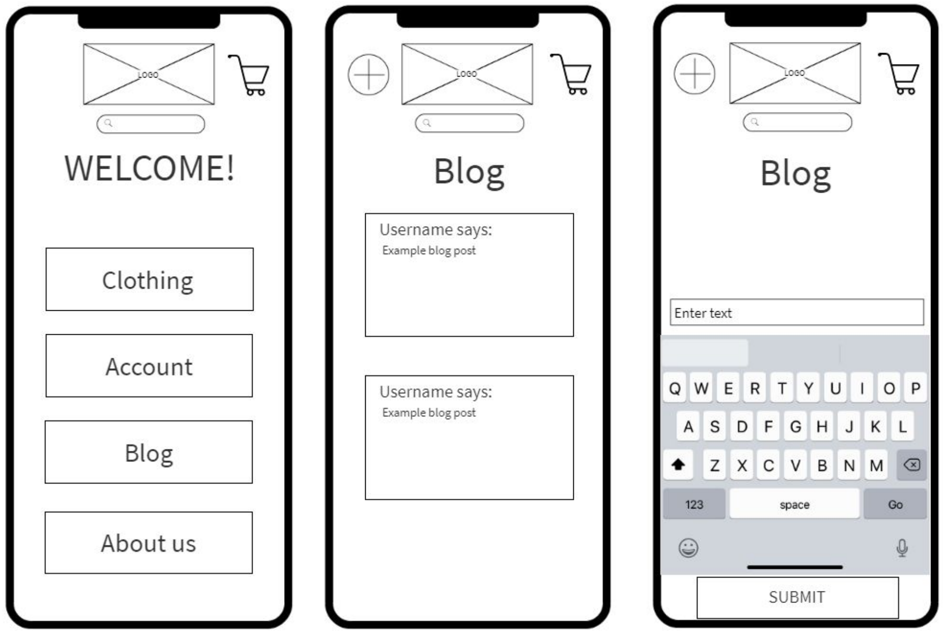
This first mock up shows how the user would go from the home page of the app into the shop portion of the app, and furthermore how to shop for specific categories of clothing. This way in one click, the user can be browsing all of the items on store, and in three, they can be browsing a specific type of item they desire. This no nonsense will ensure the user will not have to jump through hoops to get to the most important part of the app: the shop. They will also not struggle to find specific items as they can browse through different categories, along with using the search bar permanently displayed at the top of the app. They can also add items to their wish list easily using the plus button at the top right of each individual item.



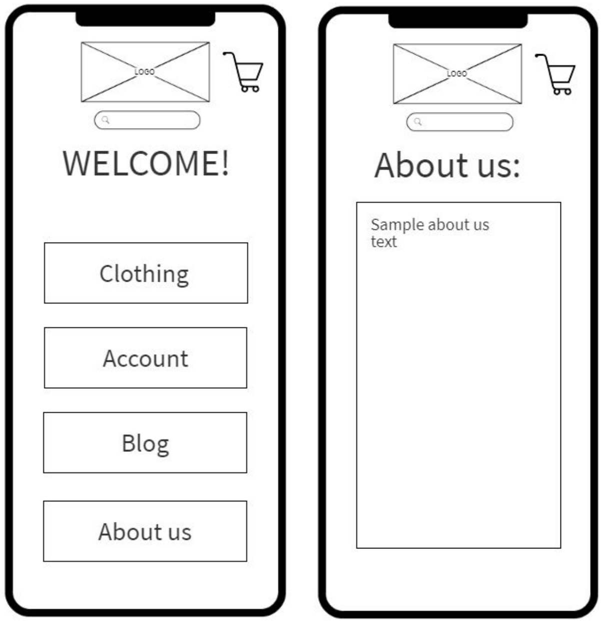
This mock up shows the functionality of the account section of the app. This shows how a user would get from the main page to a specific item they've added to their wish list. From the account page they will also be able to view their orders, past and present, and see the status of each order by clicking on the orders button. Furthermore they will be able to view and change their account details by clicking on the details button. If the user isn't logged in, clicking the account button will take them to a login/register page. The final image also shows what the product page looks like for each individual item on the site. The only difference is if the item wasn't in their wishlist, there would be the same plus sign in the right top corner allowing the user to easily add the item to their wish list. Once on the product page they can choose to add the item to their cart by pressing the add to cart button. The shopping cart is easily accessible to the user as there is a cart icon at the top right of the page at all times (unless the user is looking at their shopping cart). This ensures that it is as easy as possible for the user to make a purchase when they are finished browsing the items on the app.



As discussed briefly in the previous mock up description, we wanted the shopping cart to be as easily accessible as possible to increase the chances of orders being fulfilled. As such, when the user clicks the shopping cart icon at the top right of every page of the app, they will be taken straight to their shopping cart. They will be shown all of the items currently in their shopping cart, as well as the total cost. There will be a button labelled place order fixed at the bottom of the screen, allowing them to easily place an order. Once clicked they will be taken to a page where they can fill in their details. Once their details are submitted, they have the option to review their order. This gives the user the chance to realize if they don’t like an item, which will help the app retain users in the long run, as they won’t have the memory of being dissatisfied with an item they bought off the app.



This mock up shows how the user would go from the home page, to making a blog post. The user would have to be logged in to see the blog, so if they are not, clicking the blog button on the home page would take them to a login/register page. Once on the blog page they can click the plus sign at the top left of the page which will allow them to type up and post a blog post of their own.



The last mock up shows how a user would view the about us page. We wanted to make it as simple as possible to find out more about the client’s company.

System Tests:

The first type of system testing we would take advantage of would be usability testing. This would be one of our most important system tests as it would allow us to see if our website is easy to use and understand for our day-to-day customers. Having an easy to use website as it would greatly increase the number of customers that would be more comfortable with our software. It would also allow us to know how effective our system is. We want the system to be useful and add value to the target audience. We aim to have little navigation in order to reach the desired location.

Our method of testing will be 'Laboratory Usability Testing'. We found five volunteers in our target demographic (male and female aged 16-35). Research done in 1993 by Neilsen Landauer indicated that 5 users was enough to uncover 80% of usability problems.
We would sit them in a room with our system loaded onto phones and computers. We'd give each volunteer a different set of tasks to complete on said system. We then would silently observe and see how long it took each volunteer to complete each task. Once we'd have all the information in a report. We'd relay this back to everyone involved in the project such as the stakeholders and developers.

The reason we started with usability testing is because of its many benefits such as it allows us to get true feedback from our target audience that have used our system while testing. It allows us to not rely on 'opinions' from different stakeholders. It also helps improve end user satisfaction.

As with every system test its main disadvantage would be that it can be very expensive. It takes a lot of time and resources to set up a test laboratory and it is very hard to find people of the target demographic to do it for free. Using 5 people can be good but it would be better to use more.

Once our usability testing is finished and our developers have fixed most or all of the problems we would continue onto stress testing. Stress testing verifies the stability and reliability of the system. It measures the system on its robustness and error handling capabilities under extremely heavy load conditions.

We would do this to make sure our program would not crash in crunch situations where we would have a lot of incoming traffic. This may be during the launch of our system and during times of sales.

We would start by gathering data and setting our goals of the test. We would then write a script that would overload the system. This script would include having multiple users logged in at the same time and trying to check-out with the same item in the same size. We would then execute the script and analyze the results.

Once completed and our results have been gathered, we could send it to our developers that could tweak and change the code in order to optimize the code to meet the desired benchmark goal set out in the beginning.

The benefits of stress testing would be that we could see how our system would react in times of heavy traffic. It would also allow us to see if our system could quickly recuperate and get back up in time to not lose any sales.