Final Application Design and Implementation

by

Team SnowBonk

Version 1.1

March 2nd, 2018

Contents

Document Introduction

Purpose

This Document aims to give an overview and reflection on how the group (SnowBonk) approached and implemented the final product that was to be delivered at the expo dated 29/03/2018. The document will highlight all areas of the group project including;

- Overview of the System
- Detailed Design of the Final System
- Final Interface Design
- Implementation Methodology
- Final System Testing
- Documentation and User Guide

The document will give a detailed and in depth overview of the design and implementation of the final stage of this system.
The Document itself is intended to be read by;

- Our Group Manager Chris Fensch and will be assessing this document.
- Rob Petrick who is leading the Project and will be assessing this document.

Overview

During the creation of this document all group members have had input and all were present at the writing of this document to make the reflection accurate and honest for the readers and invigilators of the project.

The Document will be split into the following sections;

- Overview of the System
- Detailed Design of the Final System
- Final Interface Design
- Implementation Methodology
- Final System Testing
- Documentation and User Guide
- Appendix

Overview of the System

This section will discuss a high level look at the system discussing technologies and components used. We will discuss design choices for each part of the system, how they are beneficial and why we used them.

Detailed Design of the Final System

This section will discuss factors that include how the group designed and implemented the Application from Mock-Ups detailed in the Stage One Document to how they transferred into the final prototype. This will showcase our implementation schedule of when we would meet to complete areas of functionality and design by showcasing our Software Engineering approach. We will then discuss what was successful and what was not using examples as reference. The section will also showcase some of the tools and technology we encompassed to create the project.

Final Interface Design

This section will highlight the UI we have chosen for each part of the system and what perstiencies we have kept across each page in the three different sections. This part is heavily based on screenshots of our final product.

Implementation Methodology

This section will discuss the methodologies used in designing our system. We will discuss what our iterations consisted off and how we approached Stage 3.

Final System Testing

This section discusses the testing that undertook the final parts of our system to ensure robustness.

Documentation and User Guide

This section briefly covers the system as a whole and a mini user guide.

Appendix

This section will contain any material relevant to the current document to emphasise certain factors and to make reference to any claims that are made by the group in the creation of this document.

Websites

Please find a list of websites for our system:

- DineBro's User : www2.macs.hw.ac.uk/~ajg2/snowbonk/
- DineBro's Staff : www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/
- Company : www2.macs.hw.ac.uk/~pm31/SoftwareDevelopment/

Notes

All group members contributed to this section of the report and every opinion regarding the product and project as a whole was discussed and appropriately issued into the report.

Overview of the System

General

When building the system we have aimed to have a robust and simple application to ensure maximum productivity when using the service. We have built the project to be easily adaptable, using a modular design to make implementing, removing and modifying sections per customer request easy. We have aimed to complete most of the core functional requirements in the customer specification to a high level of quality, considering security and malicious intent using best practices.
The system currently runs on Heriot-Watt's MACS development servers. This server is running CentOS 6, using,

- Apache hhtpd: Version 2.2.15
- PHP: Version 5.6.18
- phpMyAdmin: Version 4.6.6
- MySQL: Version 5.6.39

For frontend development we have used a number of technologies to ensure simple yet robust functionality and attractive aesthetics. We have used,

- JavaScript
- AJAX
- JQuery: Version 3.3.1
- Cookies
- CSS3
- HTML5

A mix of all the technologies mentioned above has helped us produce our project DineBro's to a high and satisfactory quality.

User Side

The user side is considerably the most important side of the project. This is what attracts consumers to the website and restaurant so has to be simple to use, attractive and just work. We have done this by incorporating a number of technologies to ensure a smooth flow through the product. We have incorporated the famous unofficial Three-Click rule (Appendix 1.1) which is popular when designing websites to ensure this smooth flow. We have used all the specified frontend technologies and backend technologies running on the MACS server to support the wide range of functionality required.

We built the user side in a very modular approach. An example of this is the navigation bar displayed on every page for the user. The navigation bar's script is stored in a single file and imported to the top of every page on the frontend for consistency (Figure 1.1). Furthermore, this approach is used throughout the entirety of the user side to make updating, maintaining and modifying a simple.

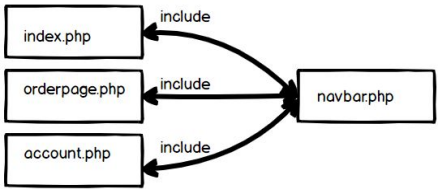


Figure 1.1 : 'Modular navigation file'

Cookies is another important addition towards developing our front end. This allowed for consistency when switching amongst pages ensuring the user never had to repeat a task. We used an open source solution to make the use of cookies easy. This was provided by Iamkate.com (Appendix 1.2) and was used for storing the users order choices and quantity of that item.

A final notable implementation was our use of AJAX (Figure 1.2) for the user side. This allowed us to perform query operations on the database without having constant refreshes of the page. This allowed the webpage to flow well and improved a variety of operations such as menu selection, restaurant filtering and table booking by loading dynamically.

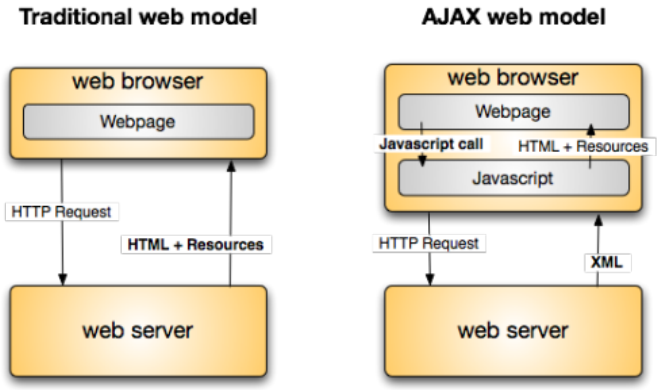


Figure 1.2 : 'AJAX Model'

Waiter Side

The Waiter System is goes to be used by multiple staff members all trying to achieve fairly similar tasks e.g take orders, view available tables, process orders etc. For this reason it was decided that the system design must be simple and easy to use. No time is wasted trying to get around the waiter system we have designed it in such a way that a staff member is no more than 2 button presses away from where they need to be.

Various technologies where used in the Waiter System implementation, these include PHP, Javascript, JQuery, AJAX and JSON. PHP was mainly used to for the communication between the system and the backend database. AJAX was used to POST html data from various web pages to specific PHP scripts which made it easy to tailor specific queries needed to pull information from the database. Javascript and JQuery where used a lot when it came to manipulating certain html elements on an individual page, most of the order transaction calculations are calculated using mostly javascript and JQuery functions.

Kitchen Side

The Kitchen Side was created with ease of use, and with minimal interaction in mind, everything needed for the kitchen side is accessed by a staff login page identical to the login page required by the waiter side, but once you've logged in at the start of the day, everything can be accessed from next page.

Cookies were again used, created on login to determine which restaurant to populate the tables orders from. Our final approach to the kitchen side revolved around the use of an AJAX request set to go off every 5 Seconds (Figure 1.2), this is to continue with our usability theme so you don't have to refresh the

page to receive a new order.

The vast majority of the actual work is done with a PHP script which gets does a huge query into the database and constructs a table to form the data, in the later stages we actually moved from a single section to multiple sections including a “Waiting area” of sorts, the kitchen had the ability to Selectively accept orders rather than have a prank order, or an order too large for their stock to go through immediately to the prep stage.

Detailed Design of Final System

General

Key Tasks

The Key tasks involved in this stage were;

- Designing a semi-permanent layout
- Making the Interfaces Responsive
- Making the Interfaces Available on all devices
- Creating forms that pass correct and valuable data

Site Maps

After the first stage of the Project and moving into the second stage we spent the first iteration making a solid overview of how the Web Application would function for both the User and the Staff(Waiter/Waitress) Systems. After the Stage Two demonstration we refined the navigation of the Customer Application even further.
Below are the current Pages we are working towards implementing and give a full abstract overview of how the User and Staff will navigate the Application. It is also important to know that the User can reach different parts of the Map from the Navigation bar that is implemented within our Application.

User Site Map

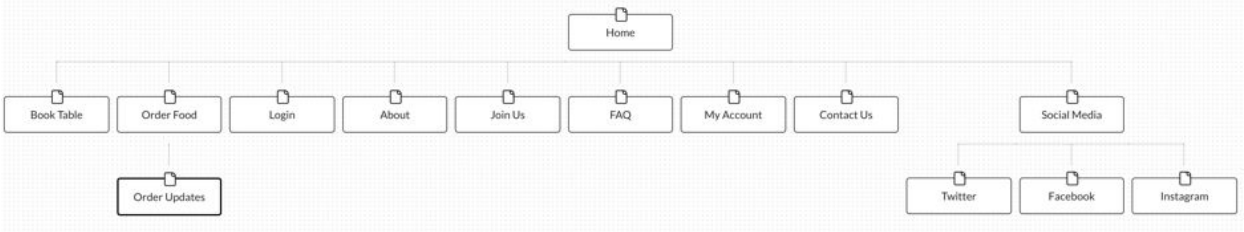


Figure 2.1 : 'Site Map'

Notes

When refining the Customer Site we believed that we should more closely adopt the “three-click” rule and make any process the User aimed to complete as easy and efficient as possible to complete.
Due to this we concluded that the homepage would be used to search and filter restaurants that the customer may choose to purchase from and that by doing this when the customer chose to follow through with ordering food or booking a table they were only ever a click away from completing that task.

Staff Site Map

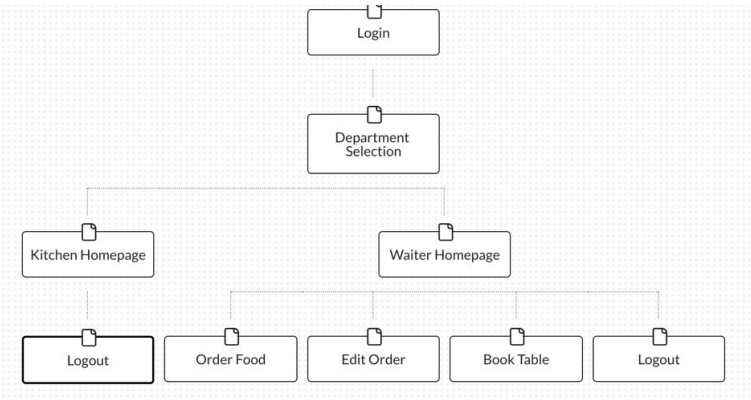


Figure 2.2 : 'Staff Site Map'

Notes

For the Staff Interface which encompasses both the Waiter and Kitchen system we again refined our implementation from the Stage Two design and re-evaluated that the site map shown in fig 2.2 was the best model of implementation. By doing this we can condense the amount of navigating the user of the system would have to do to complete a task.

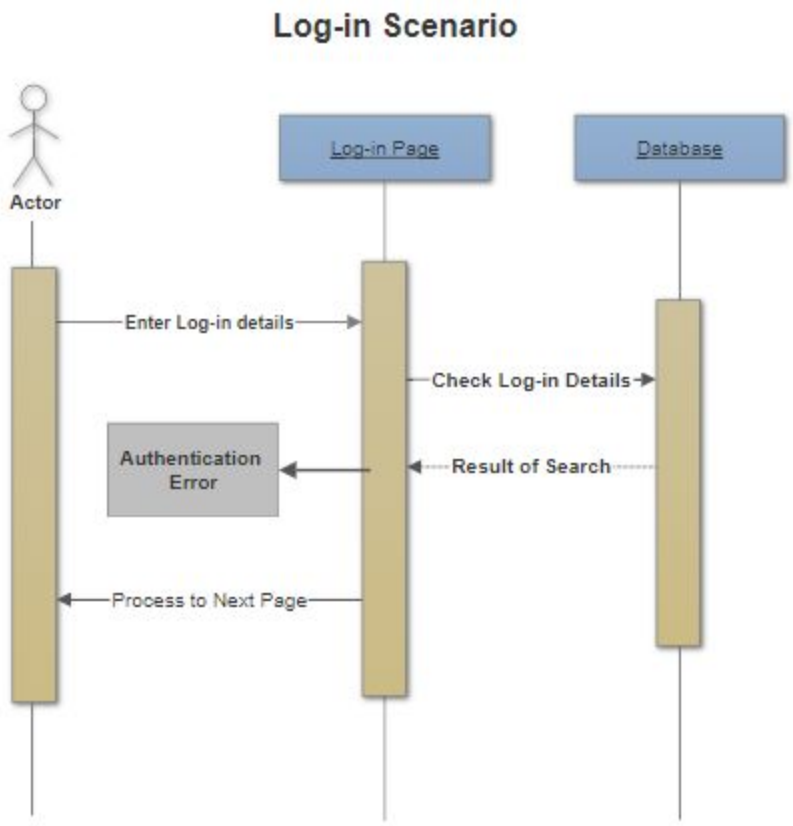
This is particularly important when you consider that staff situated in the kitchen wouldn't have the time to navigate so many pages and thus having everything on one page is a much better design choice than the previous version.

For the waiter this also makes it particularly easy to move from the correct interface he/she needs to use to complete a process aiding in making the user more efficient when it comes to competing tasks.

User Side

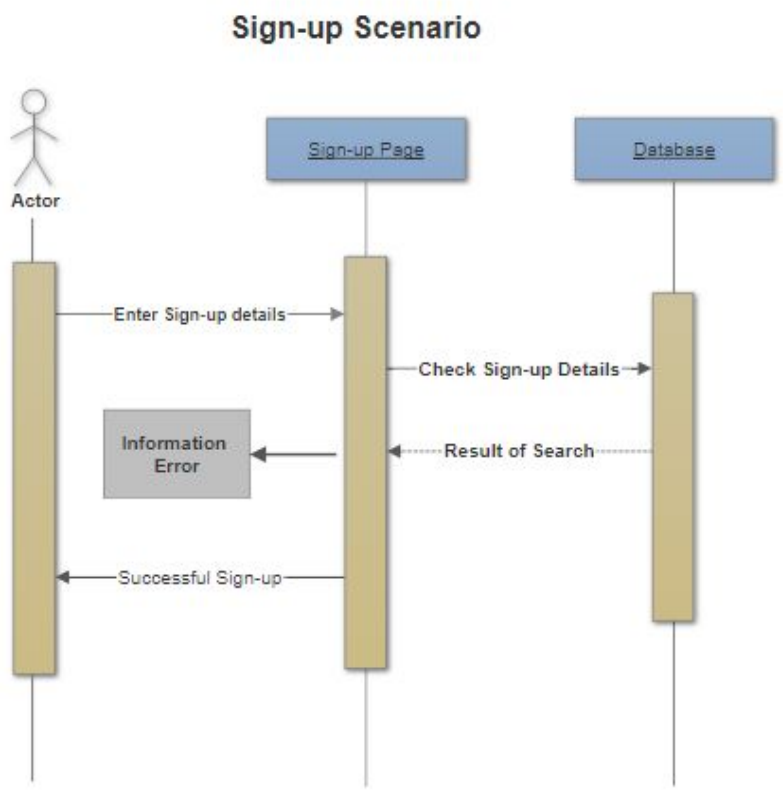
The user side has been focused on making sure that users can't break the system easily and as such there has been validation checks been put into place. Some examples of this have been that the user can't access the food ordering page without first selecting a restaurant, or that the user can't order anything unless they are also logged into the website. This kind of validation is how SnowBonk manages control of what the users can and cannot do on Dine Bro's. This can been seen in other places on the frontend such as the table reservation system.

Log-in Scenario



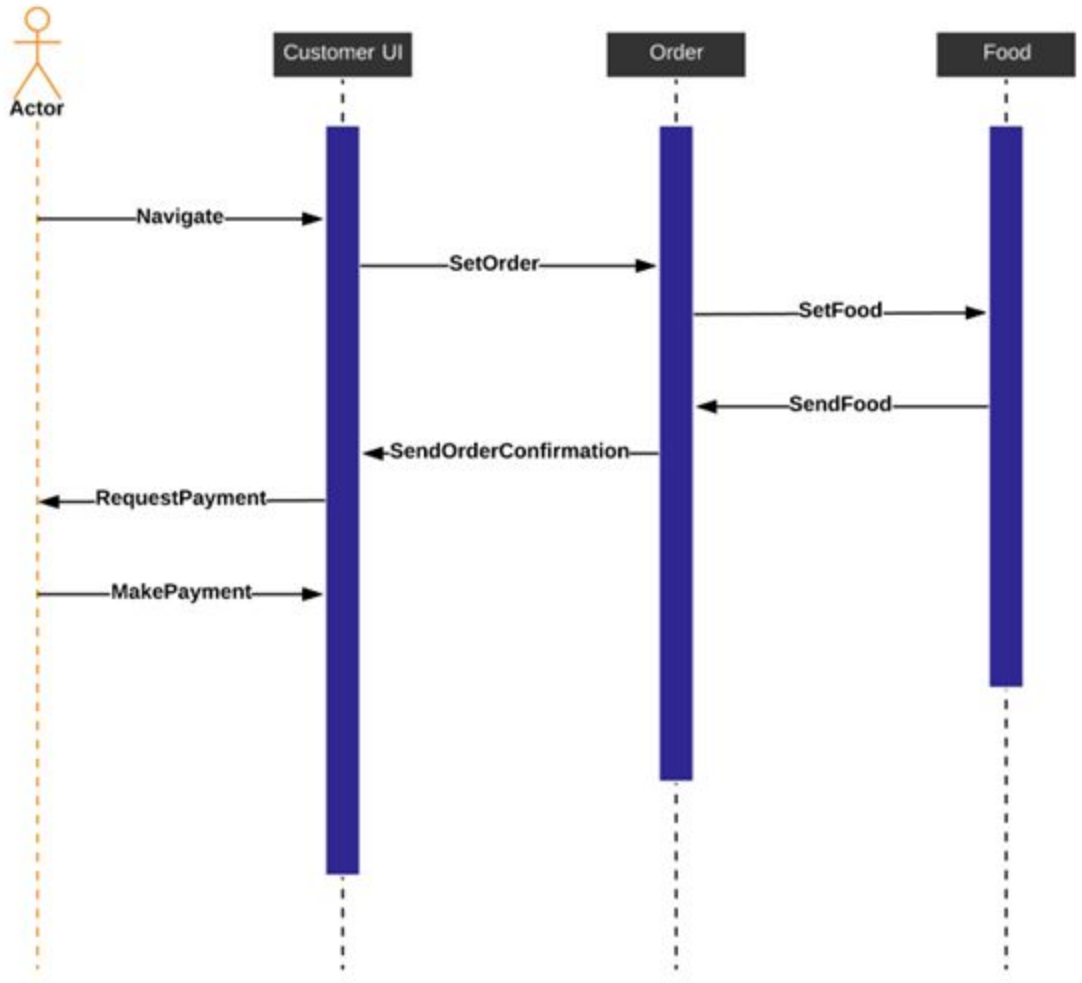
When the user wants to make an order, they will have to login to the system through the use of a unique username and password combination. The username will be based upon the email they used when signing up. When they have done this the system will attempt to verify that this is indeed correct and if the verification is successful, the system will allow the customer to progress with creating an order. If the verification fails however the user will be given an authentication error and will have to try and input the correct login details again if they wish to proceed with creating an order.

Sign-up Scenario



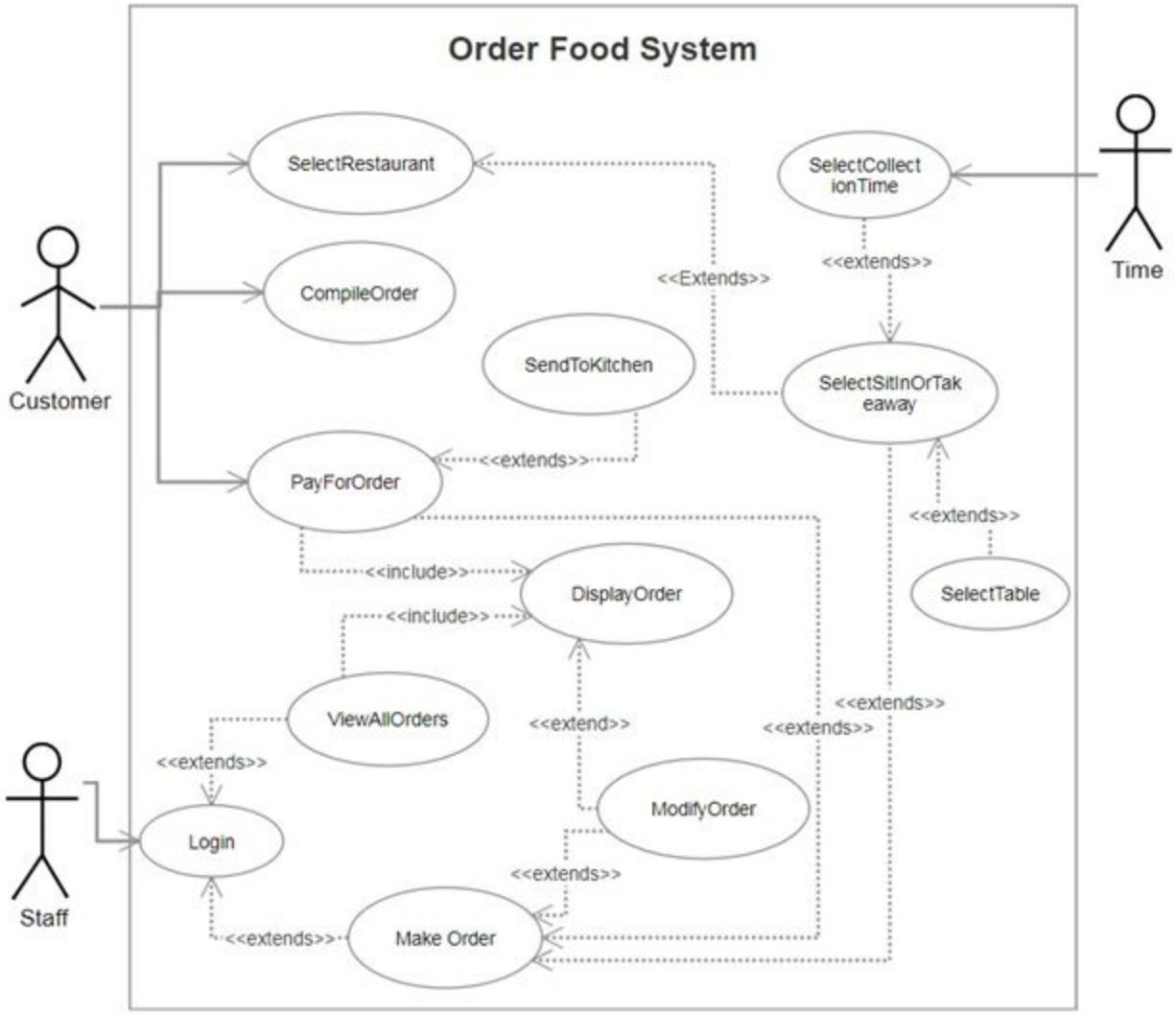
When the user wants to make an order, they will have to login to the system through the use of a unique username and password combination. If the user has not created an account previously then they will have to click on the sign-up button. This will prompt the user for several pieces of information including name, address, phone number and age. On top of this they will also be prompted to enter an email and password. The email will be used as the username when logging into the system. When they have done this the system will attempt to verify that the details are correct with no duplicates for the email as this acts as a username and must remain unique. If the verification is successful, the system will inform the customer of a successful sign-up and allow the customer to progress to the login page so they can sign-in and create an order. However, If the verification fails then the user is redirected to the sign-up page and must attempt to sign-up again.

CUSTOMER ORDER SCENARIO



When the user wants to make an order, they must first begin by navigating to the main web application, otherwise known as the Customer User Interface. From here they can directly order food from there desired restaurant. After this they can begin ordering. Whether they want to collect or have it delivered, it's up to them. Once the customer has chosen their food and completed their order having paid for it, it is then sent off to the kitchen. There order information will then be displayed along with the order status.

UML Diagram: Order Food System



There is a description below of the complex use cases in the above diagram. **Case Descriptions**

1. **SelectRestaurant -** This is where the customer can select which restaurant they want to dine from. This brings up a list of restaurants based on location.
2. **SelectSitInOrTakeaway -** This extends from **SelectRestaurant** and gives the user the option to collect their order or sit at a table in the restaurant.
3. **CompileOrder -** This is where the user can add menu items to their order. This includes adding, editing and modifying menu items.
4. **PayForOrder -** This allows the order to be paid for. This gives the option of digital payment method or cash. For cash payments the order is then placed as pending state. Other payments paid immediately records the order on the system and sends order to via **SendToKitchen** case**.** After paying for the order, the order is displayed via **DisplayOrder** case that it includes.
5. **MakeOrder -** This is where the staff member can take orders of customers who do not wish to use the app. They can link the order to an account of a customer however to track progress of the order.

USE CASE: Compile Order.

ID: 1

Brief Description: Ordering food sitting at a table in a restaurant.

Primary Actors: Customer.

Secondary Actors: None.

Preconditions:

1. Menu items are in stock.

Main Success Scenario:

1. Select Restaurant ordering from.
2. Select sit in or take away.
3. Create order by compiling all the menu items.
4. Pay for the order for customer.
5. Order is displayed. **Postconditions:**

    1. The order is recorded in the system.
    2. The order is sent to the kitchen.
    3. Customer has paid. **Alternative Flows:**

1. User modifies order.
2. User collects food.

Alternative Flow: User modifies order.

ID: 1.1
Brief Description: The user wants to modify their order.
Primary Actors: Customer.
Secondary Actors: None
Preconditions:

1. User has already paid for order.

Alternative Flow:

1. Begins after step 5 of main flow.
2. User presses Edit button.
3. User modifies order.
4. User confirms modification.

Postconditions:

1. User is refunded/charged the difference.

Alternative Flow: User collects food.
ID: 1.2
Brief Description: The user wants to order food for collection.
Primary Actors: Customer.
Secondary Actors: Time
Preconditions:

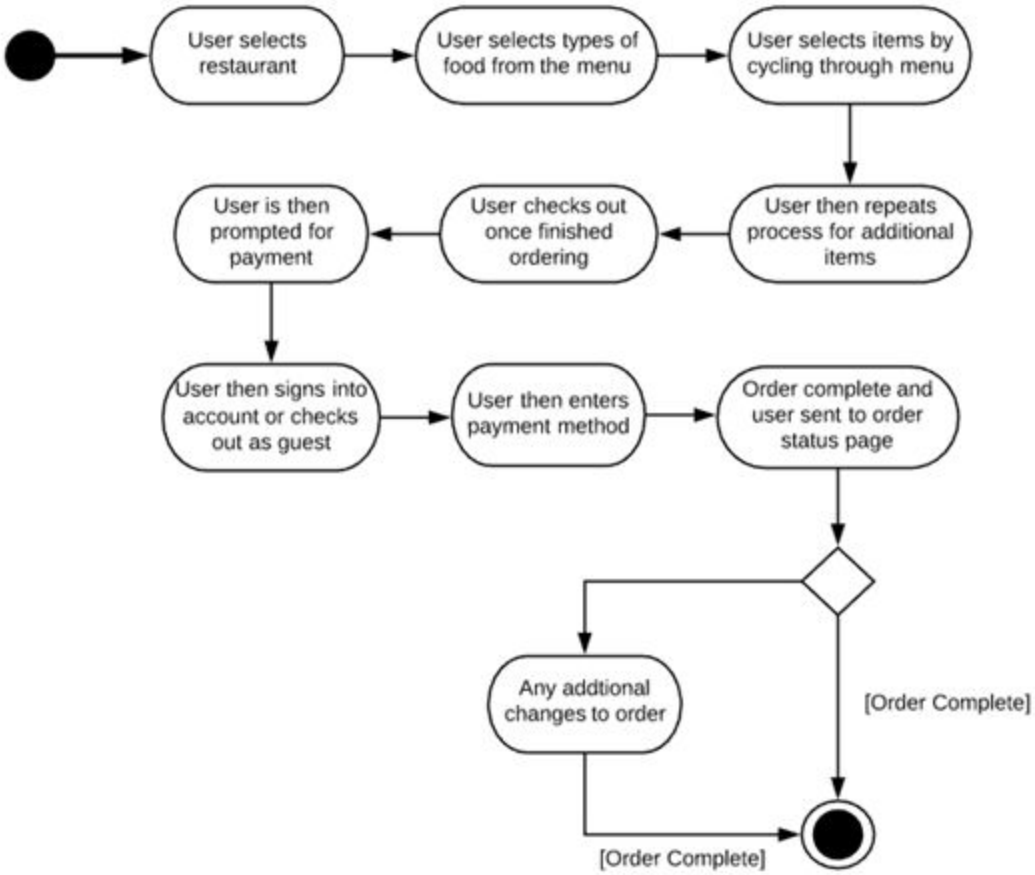
1. Completed steps of main flow up to 2.

Alternative Flow:

1. Customer selects collect option.
2. Compiles order.
3. Chooses a time for collection.
4. Pays for order.
5. Order is displayed for customer. **Postconditions:**

1. The order is recorded on the system.
2. The order is sent to the kitchen near time of collection.
3. Order is displayed for customer.

User Order Activity Diagram



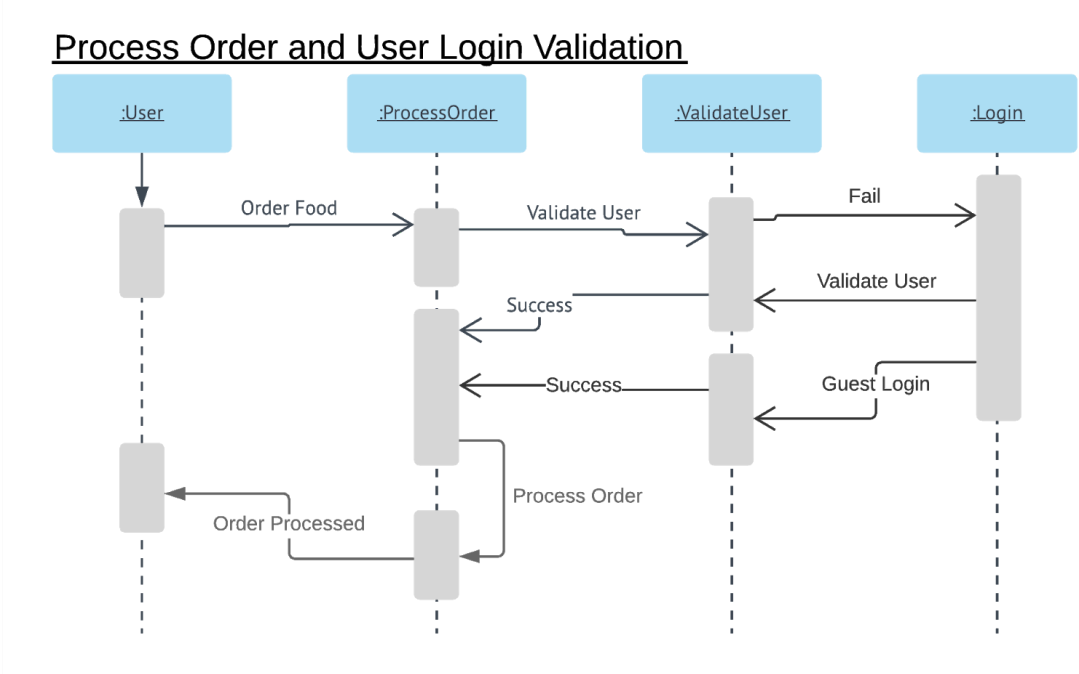


Figure 2.3 : 'Sequence Diagram for Login Validation on Orders'

Figure 2.3 demonstrates the basic steps to validate a user to order food. When attempting to process the the order, the user first has a validation check. If the user is not logged in to their account or a guest account, the user validation fails and requires the user to login. This then takes them to the login to their own account or a guest account. If they choose to login to their own account the details or then validated again and either return success or fail. For the guest login, an account is created and signed into on the spot, as such this action will always be a success. Once the user is successfully logged in, the order can be processed and feedback is returned to the user.

This input validation prevents users attempting to make an order without being logged in to an account. We designed this system to make the entire process of ordering food robust, easy and secure.

Order Page without Selecting a Restaurant

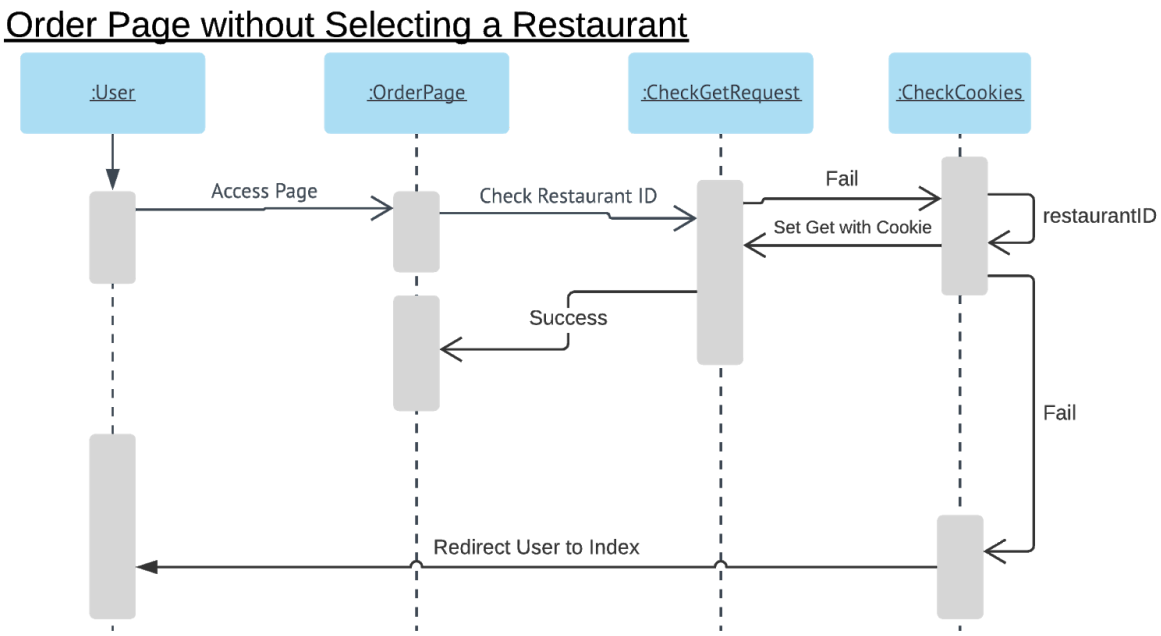


Figure 2.4 : 'Order Page Without Selecting a Restaurant'

Figure 2.4 demonstrates the checks that are undergone when the user tries to access the order page. If no restaurant ID is present as a Get request method in the URL, the page first checks to see if there is a restaurant ID stored in the user's cookies. If not, then the user is redirected to the index page. If so, the page is reloaded with the restaurant ID from cookies. Then on reload, the process is repeated to now return success when checking if the ID is in the Get request method.
This method ensures persistency and convenience for the user as well as robustness in the form of input validation. If the user leaves the page and revisits later straight to the order page, the last restaurant they were ordering for will be loaded and they can carry on from where they left off as their basket information will also be saved in cookies.

These subtle touches ensure the users flow of the website is smooth and not frustrating. SnowBonk pride ourselves on these design choices as we think about the user's experience with the product.

RESTAURANT SELECTION AND FOOD ORDERING

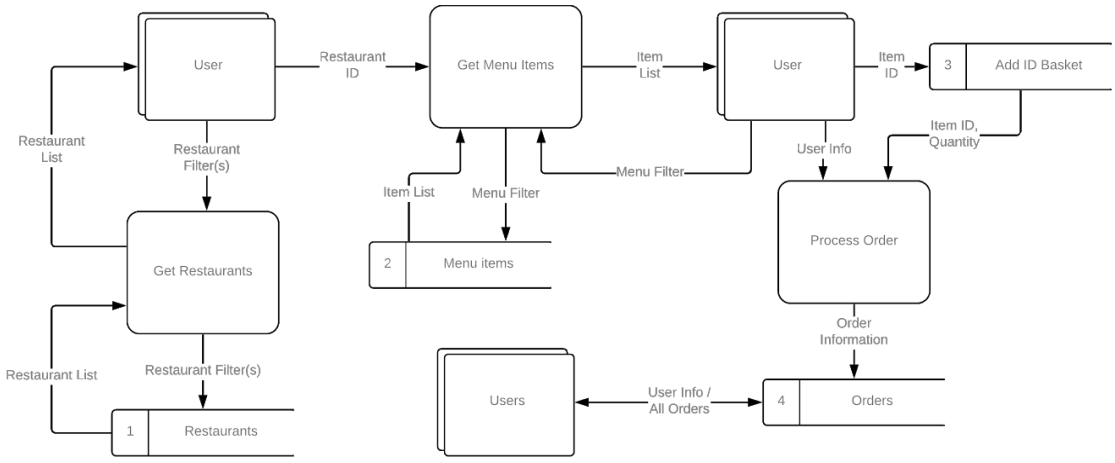


Figure 2.5 : 'Restaurant Selection and Food Ordering Data Flow'

Figure 2.5 represents how an already logged in user would proceed with ordering food through Dine Bro's. Before making a selection the user must firstly their restaurant. These can be done with a wide range of filters to sort by dish, alphabetical order or review rating system.

The restaurant ID would then be sent to get menu items based on the restaurant selected. From here users can then filter by menu category such as Starts, Mains and so on. Each time the user adds an item to their basket the ID is stored into cookies. There is also a separate quantity amount for each item.

Once the user finishes their item selection they send their user information to be processed by the server, which also gets sent the ID and quantities of each menu item ordered. This processes the information and stores the order in the database.

The user can then view all their current and past orders stored in the database.

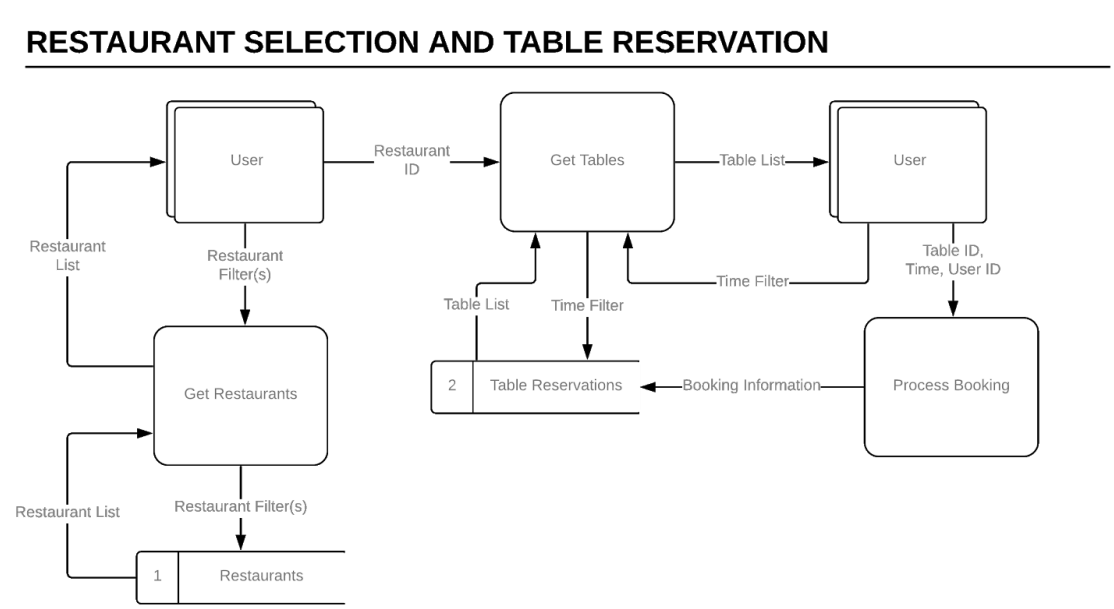


Figure 2.6 : 'Restaurant Selection and Table Reservations Data Flow'

Figure 2.6 shows data flow similar to Figure 2.5. This example is with the user already logged into the website. In terms of restaurant selection and retrieving available tables. The 'Get Tables' process on first load gets all available table reservation times and displays them back to the user. The user selects time filters from a dropdown time selector which displays available booking times in 15 minute intervals. Once the user has selected a table, their user ID, table ID and booking time is processed. This then adds their reservations into the table reservations. From here an updated table list is then sent back to the user with a popup message saying their table number has been booked for the specified time.

User Signup

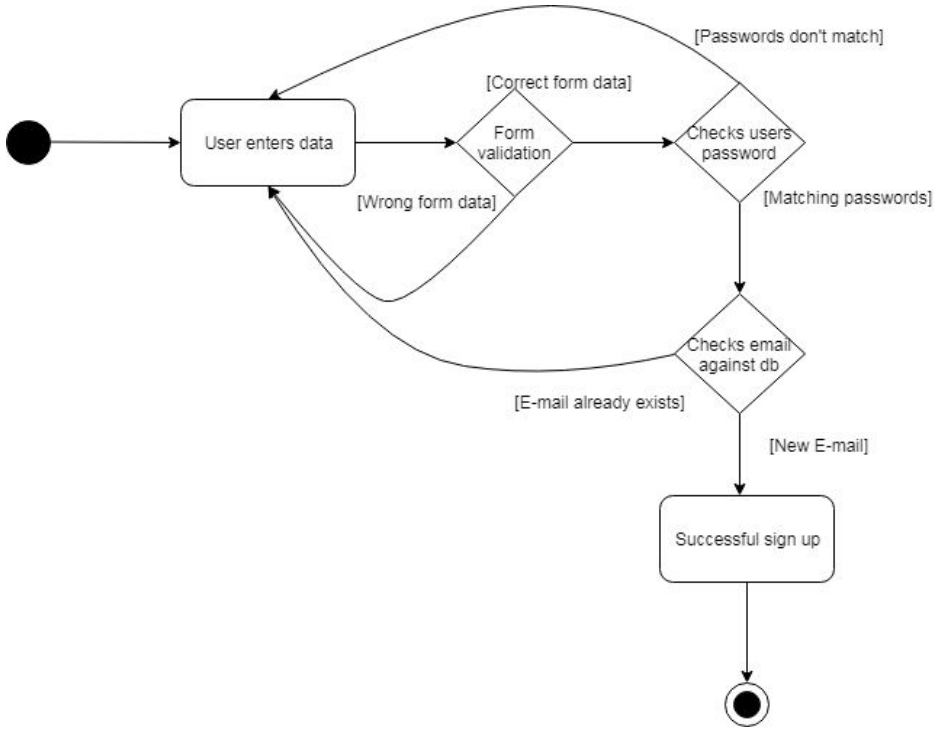


Figure 2.7 shows how a new user would sign up to the application. They would begin first with entering all their personal information into the boxes in the sign up form. When the user then clicks “Get Started” at the bottom of the form, the information will be validated to check that it has been entered correctly (emails contain @, telephone contains numbers etc.). If there is an issue with the data entered, the user will be prompted to correct the data which was invalid. The data will then move on to the next step of validation, checking the password and resubmitted passwords match. If they do match they will, the email entered will then be checked against the database to ensure the same person hasn't signed up multiple times using the same email address. If all of this is successful, a new account will have been created.

User Login

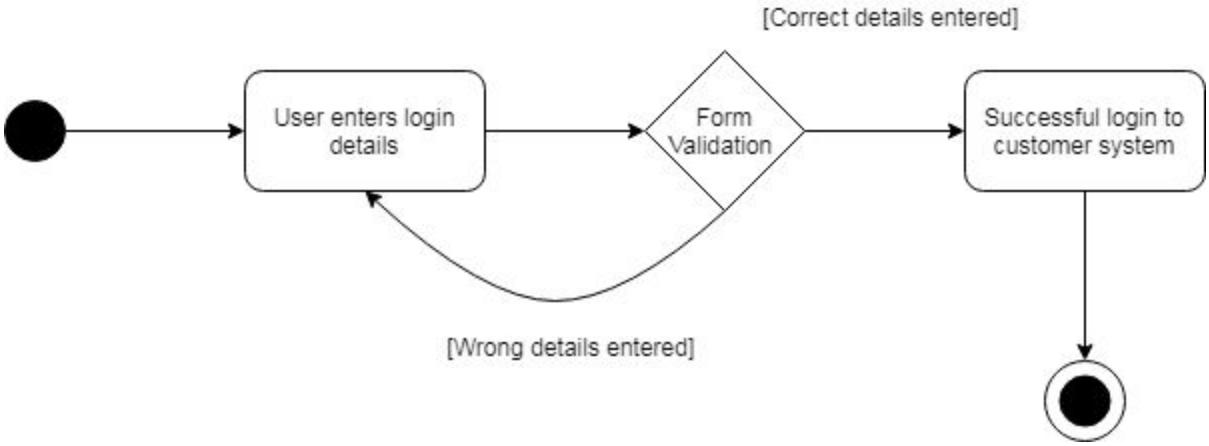
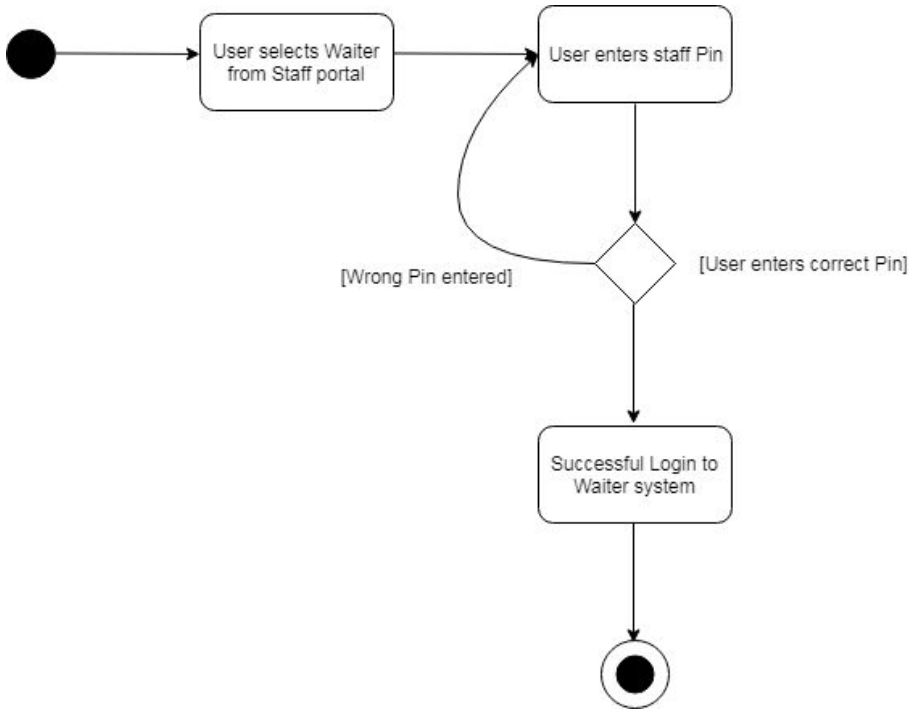


Figure 2.8 shows how an existing user would login to the system. They would begin by entering their details into the form, this information is then checked against the database, if what they have entered is correct, they will have successfully logged in. Should they enter the wrong information, they will be directed to enter their information again.

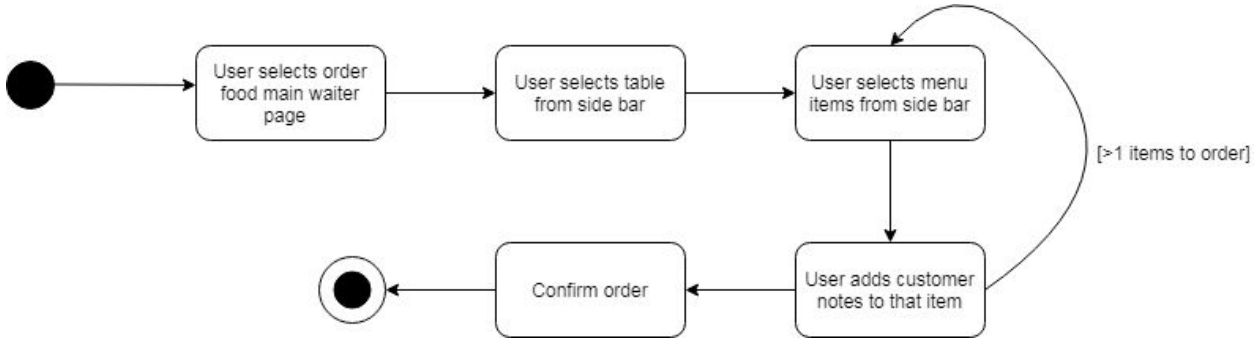
Waiter Side

Staff login



As you can see from this figure, the user will start by selecting the Waiter system from the staff portal. If they are not already logged in, they will be presented with a login form that will require their staff pin. If their pin matches what is in the database, they will be granted access to the waiter system. If their pin does not match anything in the database, they will be prompted re-enter their pin.

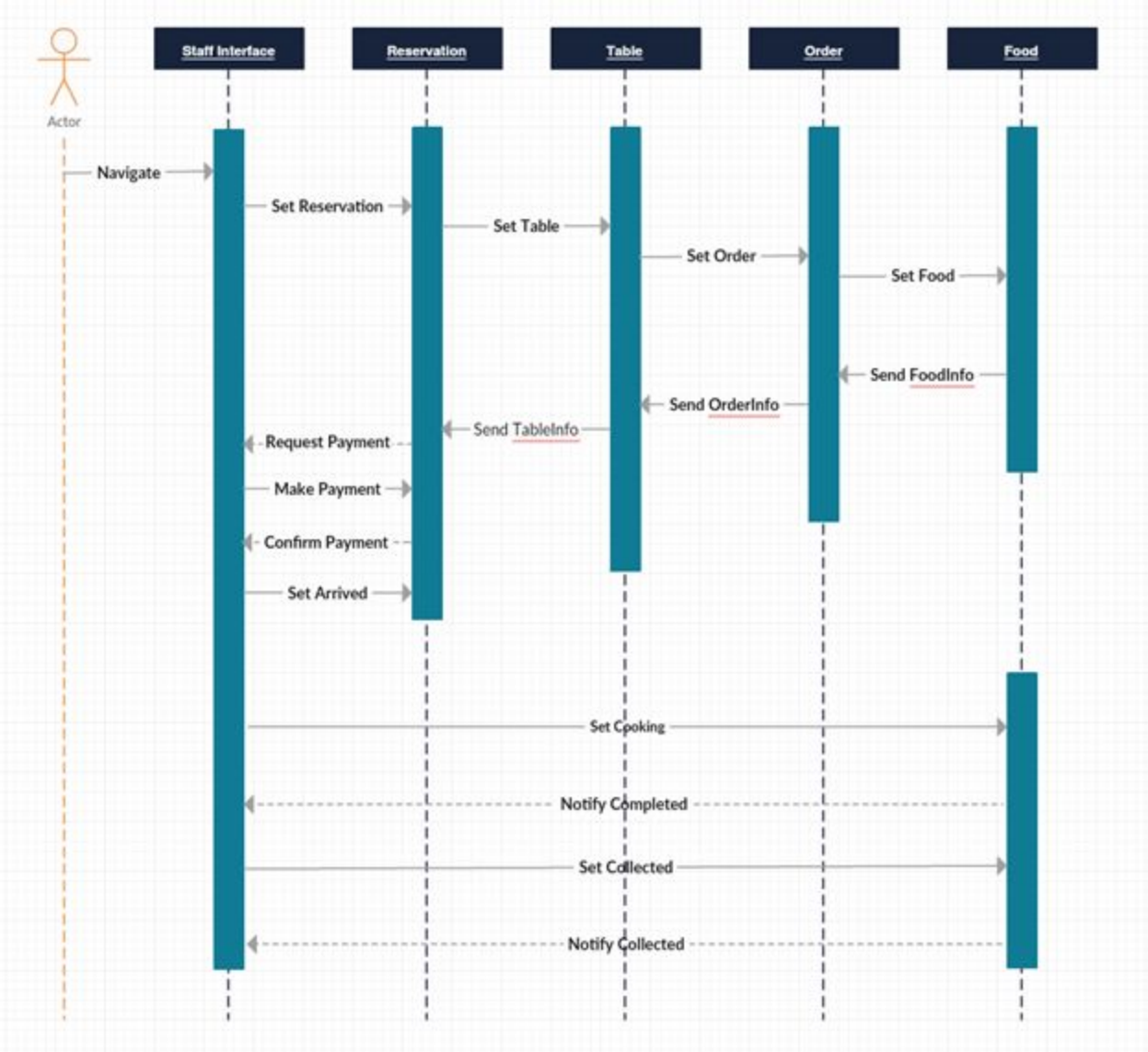
Order food



As you can see from this figure, the user will start off by selecting the order food button from the main menu of the waiter system. To make an order, they must first select the table they are serving form the

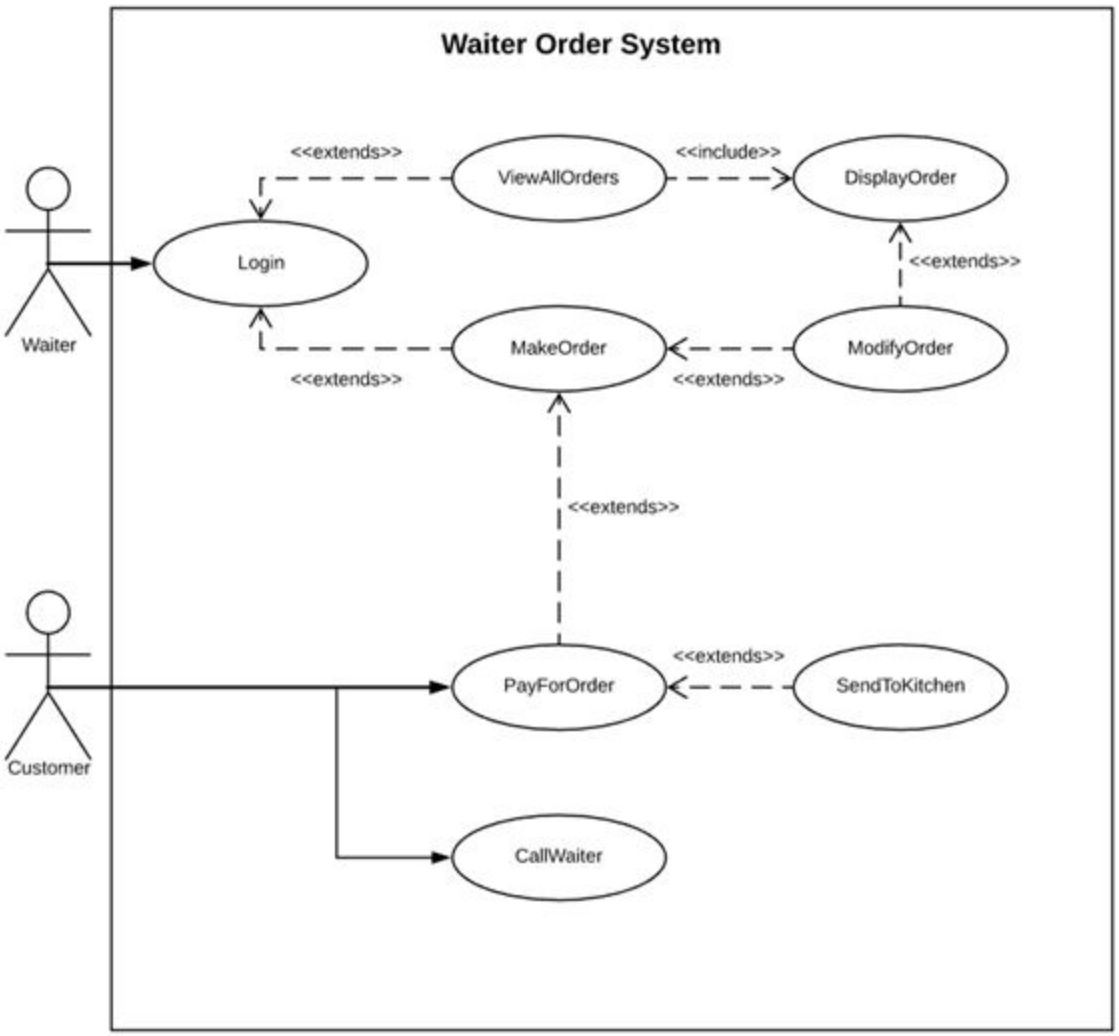
left panel. They will then have the option to choose items off the menu, again from the left panel. Once they have made a selection, they will have to add customer notes and confirm that order. This process can be repeated a number of times. Once the final order has been taken, the staff member will confirm the order, and they will be notified that the order has been recorded.

STAFF ORDER SCENARIO



The Staff will take orders from the customers in the usual way though restaurants utilizing our system will be using tablets to do order taking with our application loaded. Much in the same way as a customer makes an order, the waiter will set and reservation for a table making it temporarily unavailable on the system. They will then take the customer to the table and take their order details and their food order. They will then send this information back which will lead to a payment prompt from the customer which will hopefully lead to a successful payment. Once this occurs the order will be sent to the kitchen and the food will have begun the cooking process. The customer will then receive their order which will pass as a complete order.

UML Diagram : Waiter Order System



There is a description below of the complex use cases in the above diagram. **Case Descriptions**

1. **ViewAllOrders -** This is where the waiter can view all current orders. This also leads to the **DisplayOrder** case for individual detailed order viewing.
2. **MakeOrder -** This is where the waiter can add menu items to the order. This includes adding, editing and modifying menu items.
3. **PayForOrder -** This allows the order to be paid for. This gives the option of digital payment method or cash. For cash payments the order is then placed as pending state. Other payments paid immediately records the order on the system and sends order to via **SendToKitchen** case**.** After paying for the order, the order is displayed via **DisplayOrder** case that it includes.
4. **CallWaiter -** This is where the customer can call over the waiter either in person or through the system. This will send a notification to the waiter that a customer has requested them.
5. **ModifyOrder –** This is where the waiter can modify current orders at the user's request. This involves adding, removing, modifying and canceling orders.

USE CASE: Make Order.

ID: 1

Brief Description: Ordering the food in a restaurant via the waiter.

Primary Actors: Waiter.

Secondary Actors: Customer.

Preconditions:

1. Menu items are in stock

Main Success Scenario:

1. Create order by compiling all the menu items.
2. Confirm compiled items with customer.
3. Request payment from customer.

Postconditions:

1. The order is recorded in the system.
2. The order is sent to the kitchen.
3. Customer has paid.

Alternative Flows:

1. User modifies order.
2. User receives food.

Alternative Flow: User requests modification for order.

ID: 1.1
Brief Description: The user wants to modify their order.
Primary Actors: Customer.
Secondary Actors: Waiter.
Preconditions:

1. User has already paid for order.

Alternative Flow:

1. Begins after step 5 of main flow.
2. User requests modifications.
3. Waiter makes modifications.
4. User confirms modification with waiter.

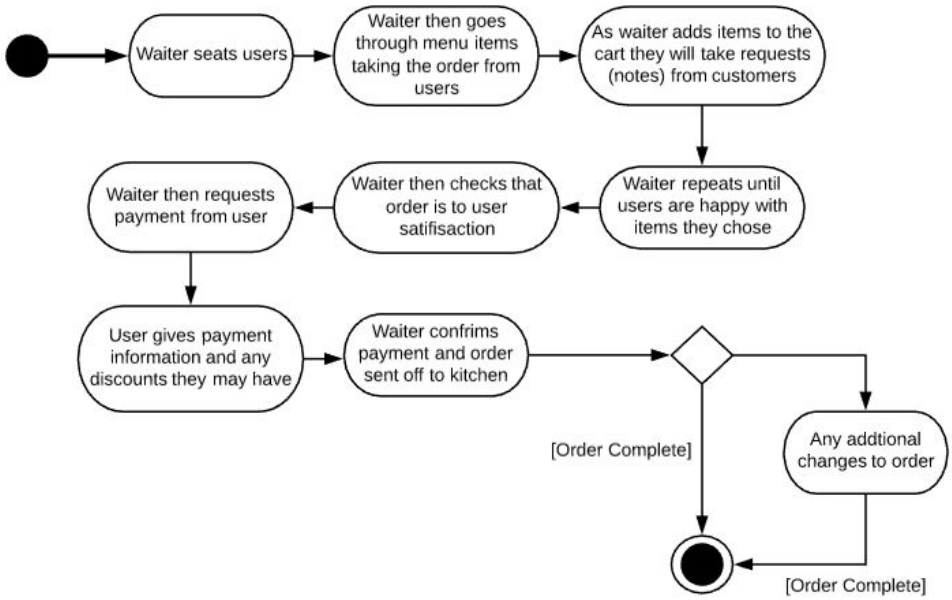
Postconditions:

1. User is refunded/charged the difference.
2. Order modifications sent to kitchen.

Alternative Flows:

1. None

Waiter Order Activity Diagram



USE CASE: Staff View/Modify Order.
ID: 2

Brief Description: Staff can view/modify current active orders.

Primary Actors: Staff.

Secondary Actors: None.

Preconditions:

1. There is current active orders in the system.

Main Success Scenario:

1. Staff member logins in.
2. Selects an active order from all list of active orders.
3. Modifies the specific order.
4. Confirms modification.

Postconditions:

1. The order update is recorded in the system.
2. Additional charges/refunds are due.

USE CASE: Staff Makes Order.
ID: 3

Brief Description: Staff makes order.

Primary Actors: Staff.

Secondary Actors: Customer.

Preconditions:

1. Staff is communicating with customer.
2. Staff member logs in.

Main Success Scenario:

1. Staff member logs in.
2. Staff selects table customer is at.
3. Staff compiles their order.
4. Staff takes payment of order.

Postconditions:

1. The order is recorded on the system.
2. The order is sent to the kitchen.
3. Order is displayed to staff.

Alternative Flows:

1. Staff Makes Collection Order.

Alternative Flow: Staff Makes Collection Order.

ID: 3.1
Brief Description: The customer wants to order food by phone for collection.
Primary Actors: Staff.
SecondaryActors:
Customer, Time.
Preconditions:

1. Completed steps of main flow up to 1.

Alternative Flow:

1. Staff selects collect option.
2. Staff compiles order.
3. Staff chooses a time for collection that customer requests.
4. Staff sets payment as pending for order.

Postconditions:

The order is recorded on the system as pending.

Kitchen Side

The kitchen side has been built to be not particularly complex, so security of design wasn't a problem, the whole staff section is presided over by a login checker (Figure 2.8), if you aren't logged in then you get kicked out to the login screen. the benefit for the kitchen section is that it's all contained in one page so there is less chance of a hole in security.

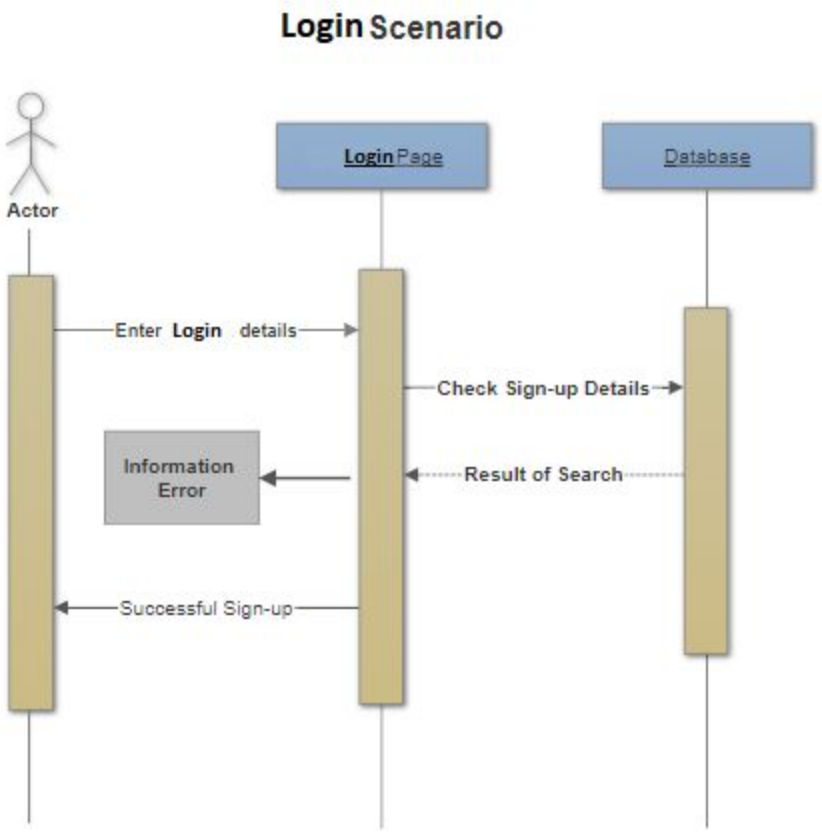
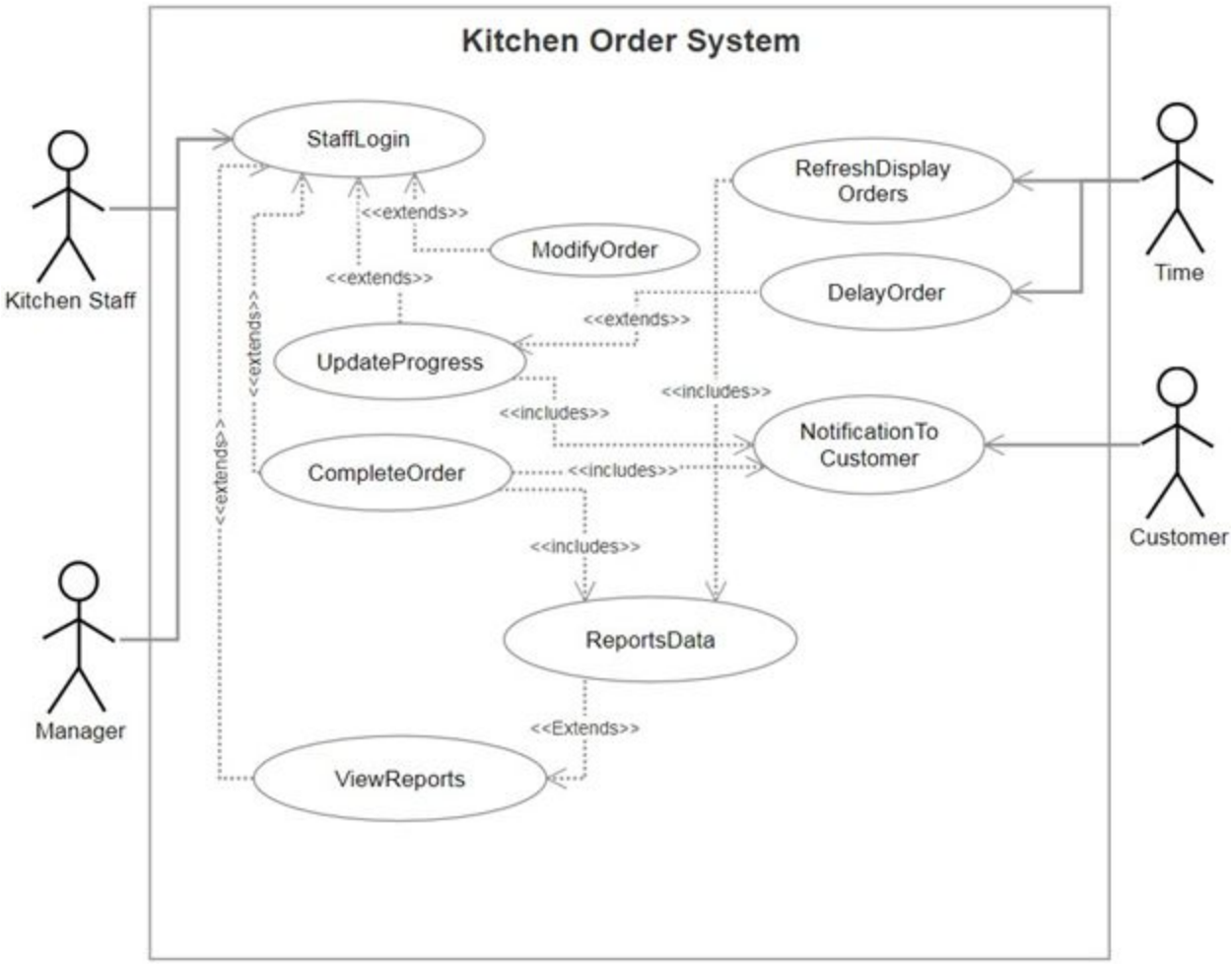


Figure 2.8 Login Process for the Staff

The rest of the kitchen processes are all very simple for the user to use, the edit button simply creates a form which will mimic a change in the database, and the accept button will immediately change the database from not accepted to preperaring

UML Diagram : Kitchen Order System



There is a description below of the complex use cases in the above diagram. **Case Descriptions**

1. **UpdateProgress -** This is where kitchen staff can set the progress of an order until it's completion. This sends updated progress to the customer via **NotificationToCustomer** case.
2. **RefreshDisplayOrders -** This is when every few seconds the kitchens display system is refreshed to alert the kitchen staff of new active orders.
3. **DelayOrder** - This allows kitchen staff to delay orders which may take longer than expected. They have a choice to multi-select orders or do a mass delay order which will delay every order.
4. **ReportsData** - This is where certain data is added to reports for the restaurant. **CompleteOrder** and **RefreshDisplayOrders** include this case to update data from these specific cases.

USE CASE: Kitchen Order.

ID: 1

Brief Description: Where kitchen staff complete an order.

Primary Actors: Kitchen Staff.

Secondary Actors: Customer, Time.

Preconditions:

1. Kitchen display has updated to the latest active orders.
2. There is an active order.
3. Kitchen staff are logged into the system.

Main Success Scenario:

1. Kitchen staff expand active order.
2. Kitchen staff check off parts of the order as they are completed.
3. Customer is notified of updated progress.
4. Kitchen staff complete order.

Postconditions:

1. Data is sent to reports.
2. Customers are updated of order completion.

Alternative Flows:

1. Delay Order

Alternative Flow: Delay Order.

ID: 1.1

Brief Description: Kitchen Staff need to delay an order.

Primary Actors: Staff.

SecondaryActors:

Customer,Time.

Preconditions:

1. Completed steps of main flow up to

Alternative Flow:

1. Kitchen staff updates order that there is a delay.

Postconditions:

1. The customer is notified of the progress update.

USECASE:

KitchenReports.

ID: 2

Brief Description: Manager views reports.

Primary Actors: Manager.

Secondary Actors: None.

Preconditions:

1. Manager is logged in to system.

Main Success Scenario:

1. Manager presses the 'view reports' button.

Postconditions:

1. Reports are displayed

Alternative Flows:

1. None.

Final Interface Design

Research into Design

During the initial design stage we looked for inspiration within the area we were developing for to see and inhabit some of the qualities which makes professional applications in the field so successful.
One successful food ordering system is the “hungry house” application which offers a simplistic design and uses an image with bold text to bring the users attention towards the center.

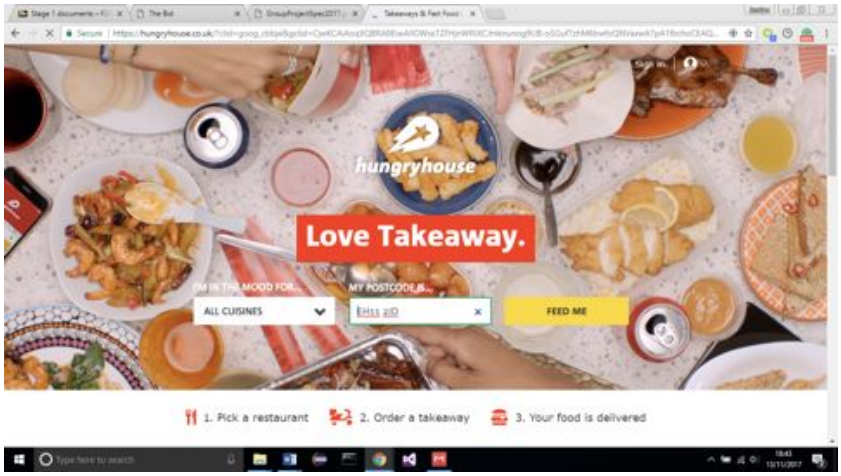


Figure 3.1 : 'Hungry House Website’

Fundamentally we believed the best approach to creating software for a food ordering service is to look at other successful food ordering systems to gain inspiration and design ideas of successful products. However, doing this we can also critic on what we would want to improve on and add these to our systems.
Another successful ordering system is “Just-Eat”. Comparing the two there is a large emphasis on the colour red promoting promotions, quality and to highlight the different areas to the user.

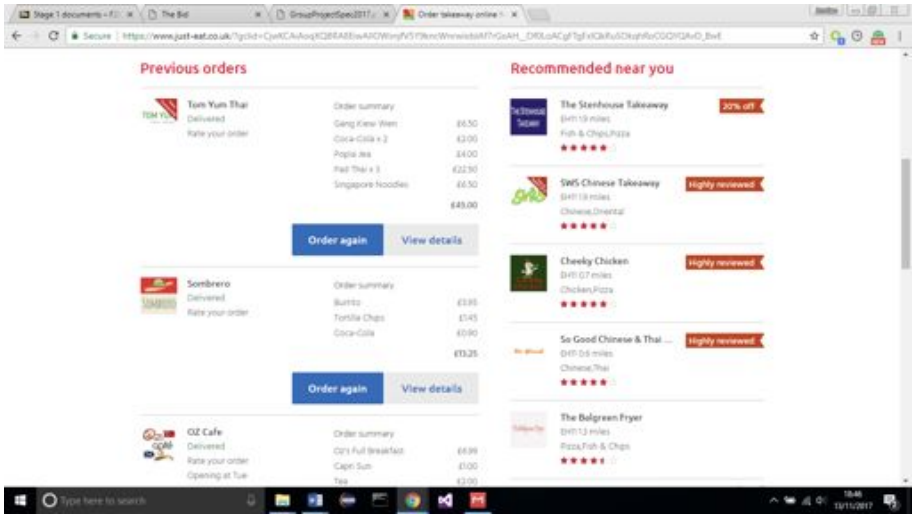


Figure 3.2 : 'Just Eat Website'

Finding inspiration for the “Waiter” and “Kitchen” systems was quite difficult as many of the designs we found were dull and lifeless which is something we wished to avoid when creating our applications. This decision was made as a group and we as a whole believe that good design leads to better functionality and user experience.

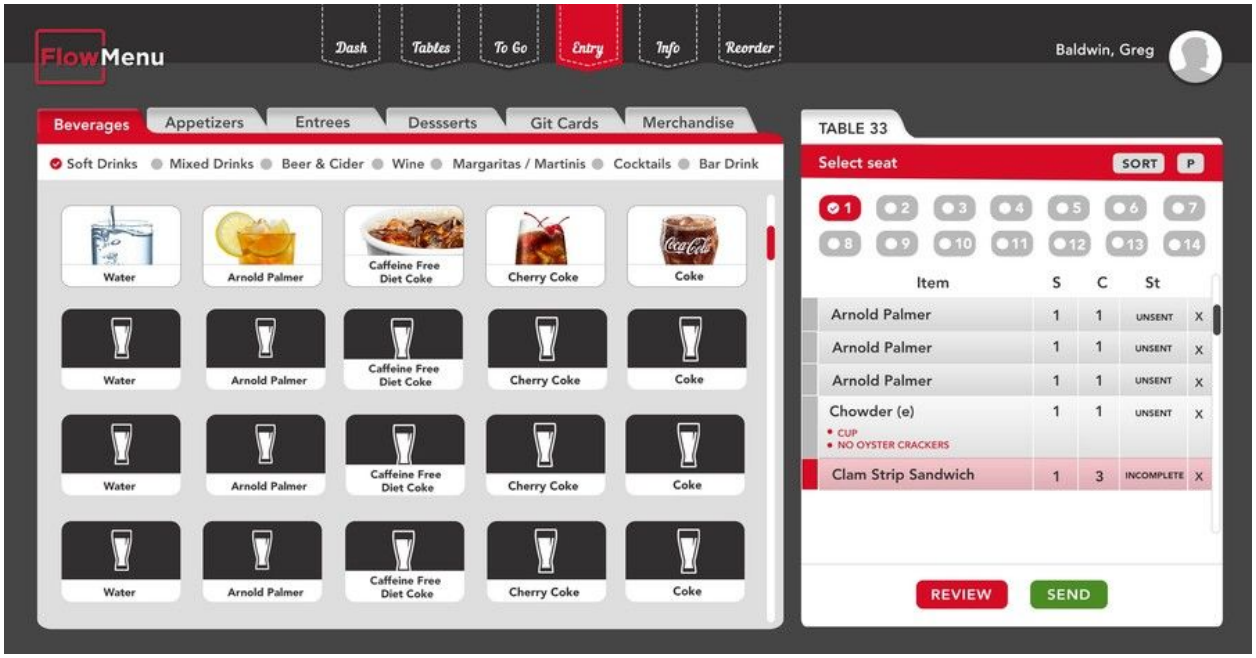


Figure 3.3 : 'Flow Menu Interface Design'

The flow menu pictured above showcases a “fun” and aesthetically pleasing design for the waiter system and was used as a reference when generating the initial design for the “Waiter” and “Kitchen” sides of the system.

However, the group believed that there was too much going on within the screen itself and due to the specification stating that the “Waiter” and “Kitchen” systems were predominantly going to be used on tablets we chose to limit the amount of icons that the user would see.

Designing interfaces encompasses the challenge that one could try and add every possible feature the user could do and furthermore itemise the entirety of the selections available. However, by doing this the problem arises that the interface could become more confusing and irritating rather than intuitive and simple to use.

What we aimed to design was a series of systems that could easily be viewed once and the user instinctively knows how to operate. By doing this we would be creating an advantage in the market by reducing the training time and the added cost the time brings upon the business.

Colour Scheme

Colour is a vastly important factor to get right in the designing of any product regardless of it being physical or not. Because of the importance of this factor, we as a group considered what the best colours would be to use throughout the applications of both the Customer and Staff systems. Due to the nature of the project and because it was in regards to “food” we believed that the best option was to go for a strong “red” that would be used sparingly but weighted enough that it made an impact. The choice for red came from research into similar products on the market and research into colour selections themselves. From the research we found that red was a colour that indicated appetite and would entice the user to feel hungry thus the other colours used were used in highlighting the refined uses of the colour.

Customer

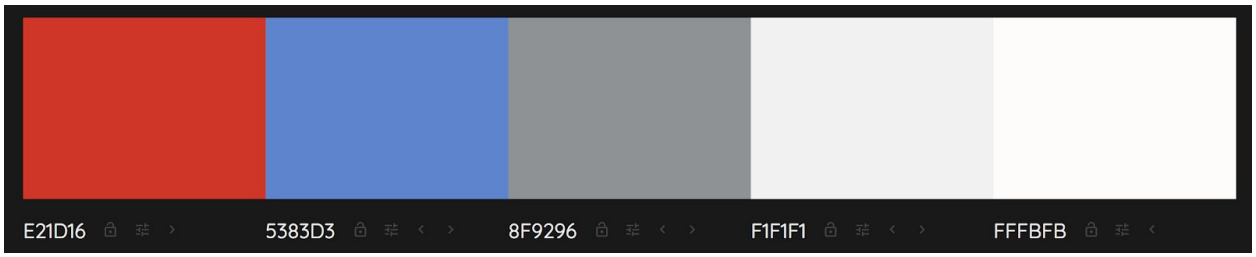


Figure 3.4 : 'Customer Colour Scheme'

To contrast the red undertones and the dominant white and grey backgrounds we decided to go for a light blue that would both contrast but also “work” with the developed system design. The blue itself was heavily used for buttons and other actions the user could interact with. By keeping this theme running throughout the design we believed it would help the user feel familiar on any page on the application. The different shades of grey were used to give a more formal background from the pure white background sections of the pages. This allowed us to highlight the different areas of the user focus but not create so much contrast that it would cause the user headaches when viewing the site for longer periods of time.

Staff

Figure 3.6 : 'Staff Colour Scheme'

The staff colour scheme was designed with the same priorities as the customer side of the interface however in contrast to the “clean” view of the light grey and white backgrounds we chose to use a darker contrast that we believed would highlight the buttons and operations the user would need to complete tasks.

Design Iterations

Throughout the project we have made many changes to the design of the system, this is due to;

- Usability testing carried out in Stage One with the initial design of the mockups
- Modifications to best fit the functionality we were trying to implement
- improvements suggested by members of the group, peers and supervisors of the project.

Although this has hindered our final implementation we believe that we created a truly unique and professionally aesthetic product compared to other solutions in completing the given specification. Throughout the designing and implementation we as a group have held the value that design helps with user experience and usability.

Because of the values we held in presenting a professional product the group continuously evaluated the current status of the design.

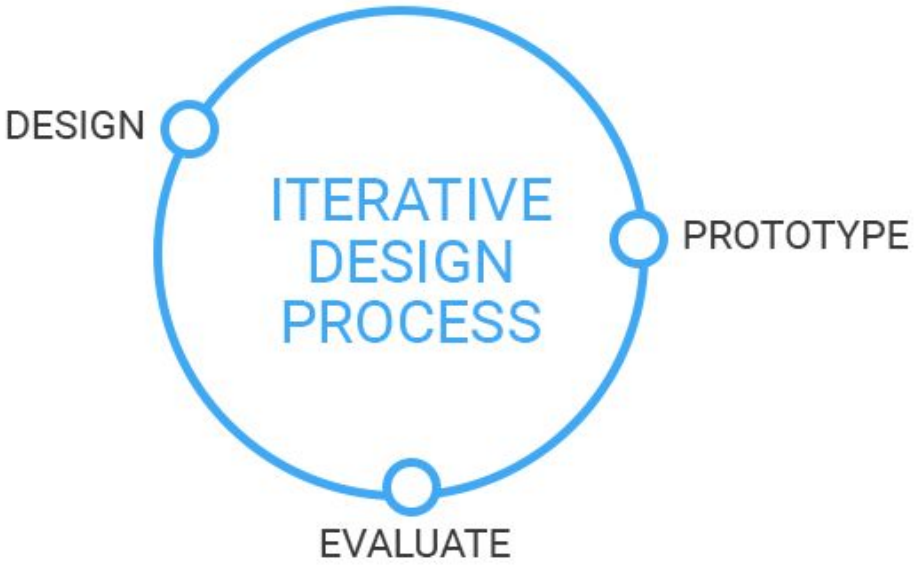


Figure 3.7 : 'Design Iteration Process Diagram'

Due to the nature of how our group developed our application (Agile Methodology) whenever a functionality or element was added to the project we could then go about making it look visually appealing and then refining the process to make it as user friendly as possible.

User Side

HomePage

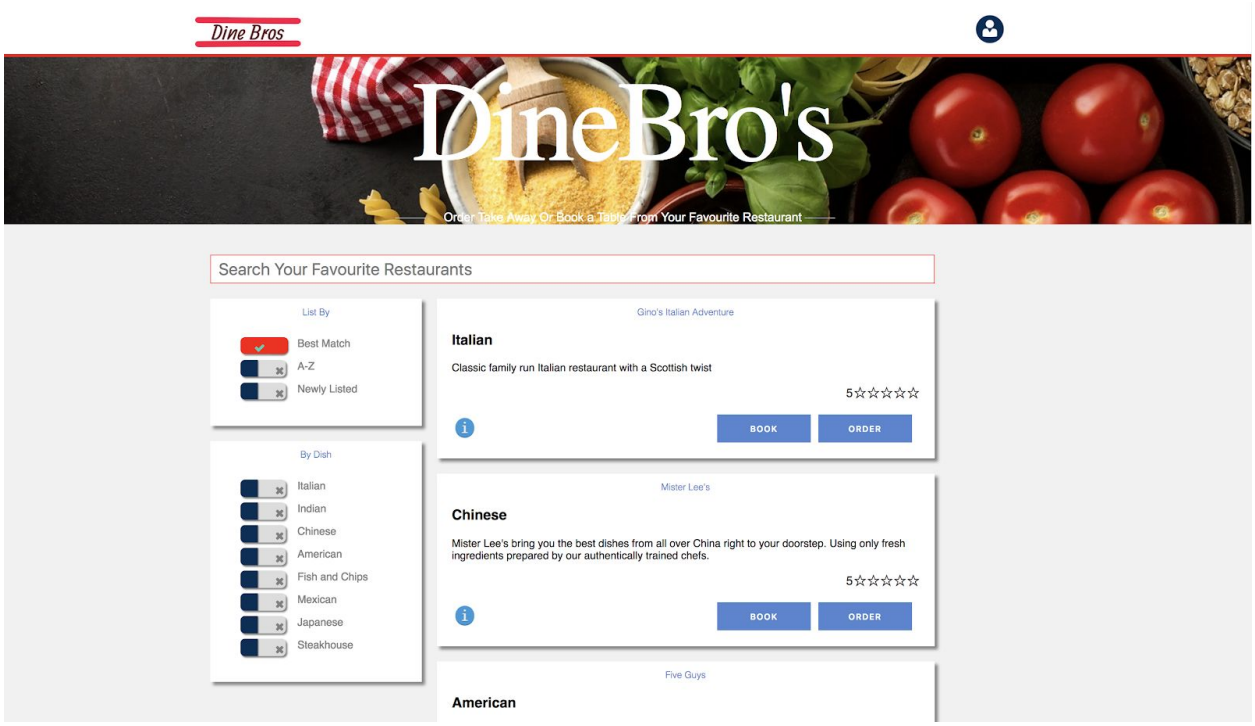


Figure 3.8 : 'Customer Homepage'

This is the first page the customer will see when they access the Dinebro's application. By using a "T" style designed where the navigation and banner carry the whole width of the page it then draws the users attention to the centrally focused search bar, listing features and restaurant profile.

Each restaurant has its own description that showcases a little piece of information about the restaurant and also has a rating figure. We chose this to highlight alongside the description and above the "order" and "book" buttons so that the user can easily gauge on how good the restaurant is so they have more information upon creating an order with a particular restaurant.

To help the user we also added an icon on each restaurant description that when clicked would show if the restaurant was vegan and or vegetarian friendly. This was to help the usability and user satisfaction of the customer and to allow those with any initial food requirements can easily see if the restaurant will cater for their needs.

Header and Footer

Both aspects of the header and footer were designed to be simple and not too distracting from the main focus of the user. To highlight these factors we used a completely white background with a red font and line colour to highlight the navigation aspects that were available to the user.

Figure 3.9 : 'Customer Site Header Navigation'

The navigation bar was designed to be simple. Like many sites now do the icon on the top right acts as a hover dropdown menu and when the user scrolls over it the option to navigate to different pages is available to the user. The logo placed on the left side of the screen is again a standard of most websites and applications. By having the button here we aimed in achieving easy navigation back to the initial homepage from anywhere in the site as the header is included throughout the entire application.

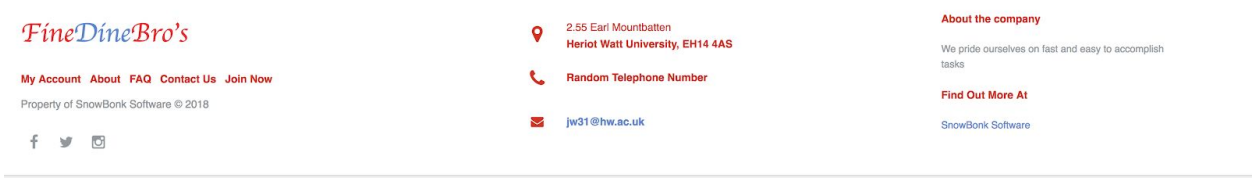


Figure 3.10 : 'Customer Footer'

The footer for the site was designed with the idea of it being “fun” yet informative enough that the user without any prompt could quickly navigate its links and also have quick access to contact information and further information about the group itself.

The footer also has working links to our groups social media accounts which was a design choice that satisfied another level of professional image and also aided in our marketing campaign.

General Pages

General pages throughout the website encompassed the same running theme and were centrally focused to draw the users attention to that particular area.

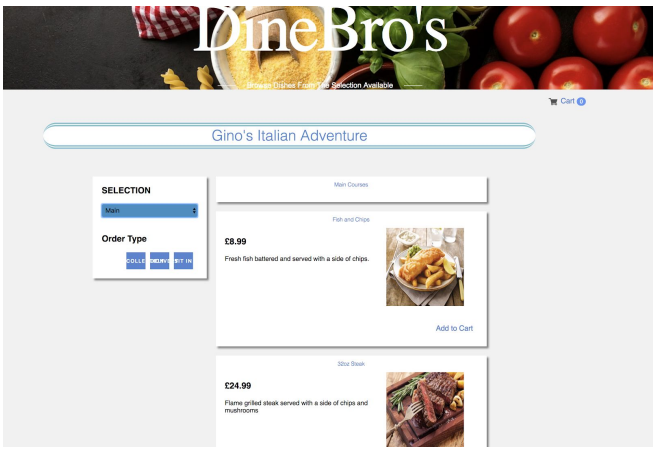


Figure 3.11 : 'Customer Menu Item Selection'

Every restaurant upon “Order” selection had its own title at the top under the banner that was a constant throughout the application and also brought attention to the cart that was situated to the top-right corner of it. Again we kept the layout concurrent through the application and allowed the user to easily navigate and filter the menu items available at that particular restaurant.

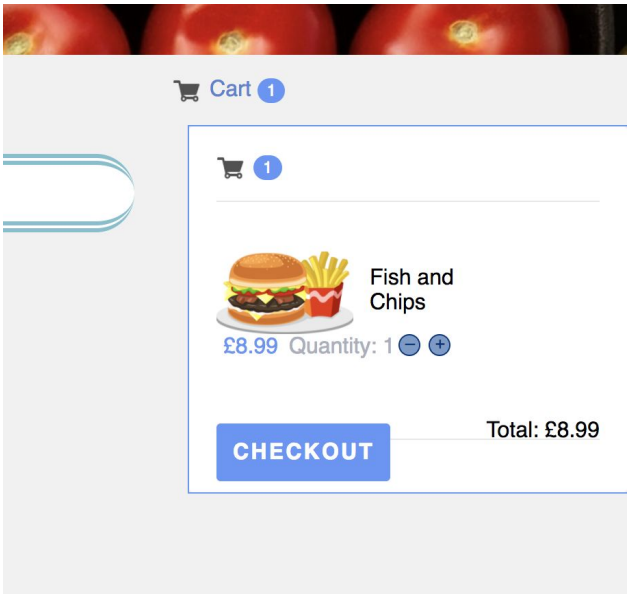


Figure 3.12 : 'Customer Basket'

The cart itself when expanded would show the items that are in the basket with additional information such as the cost of individual items and also the total cost of the order and the progression button that

would be used to complete the order. Again the colour scheme was used here to allow users to feel familiar with how they should proceed to complete an order.

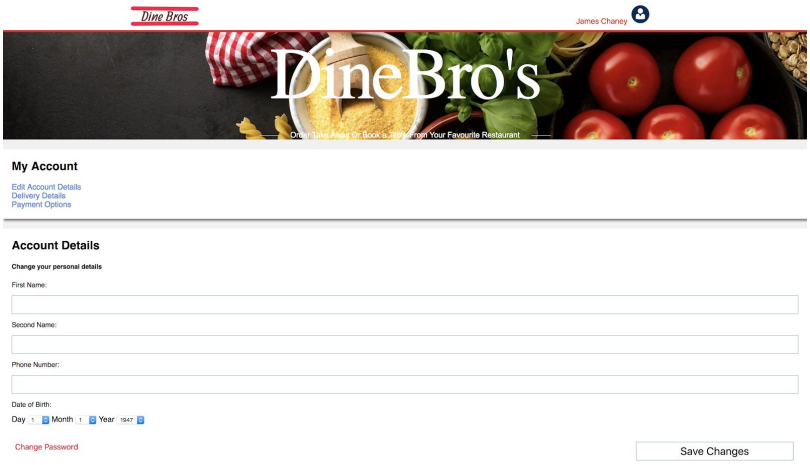


Figure 3.13 : 'Customer Update information form'

Updating the users account was left to be easy as possible for the user thus we created a simple form that the user could enter the new up-to date information into and then easily click the “save changes” button to commit. The button itself was left white to signify a more dramatic change from the rest of the buttons that run throughout the web application itself. At the top of the page there is also page navigation options to allow the user to easily find the particular area they are looking for.

Figure 3.14 : 'Customer Order Review'

The order review page was created again to be simplistic and easy for the user to gauge the progress of their order. This was achieved by using a circular progress bar to achieve this effect. The colour however differentiates from the normal colour scheme. This was done to highlight that it was an important area hence why it is also centred.
The page also consists of being broken up into different sections showing current and previous orders allowing the user to see their order history as an extra feature.

User Interface Prompts

To highlight some of the areas the user could interact with and to also showcase when they had completed an action we designed message boxes that would darken the background of the current page to focus the user upon the action they have committed.

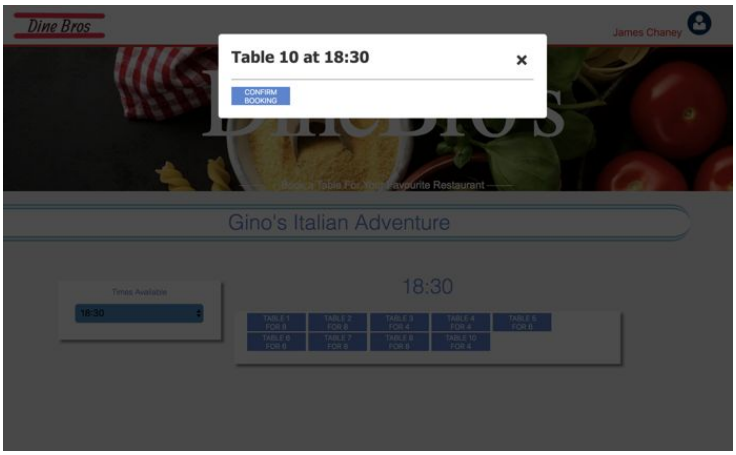


Figure 3.15 : 'Table Booking Confirmation Prompt'

The above prompt shows what would display when the customer had booked a table for themselves. It would show the user the table number and time of booking and then contain a button that would allow the user to confirm the booking incase it was selected by accident.

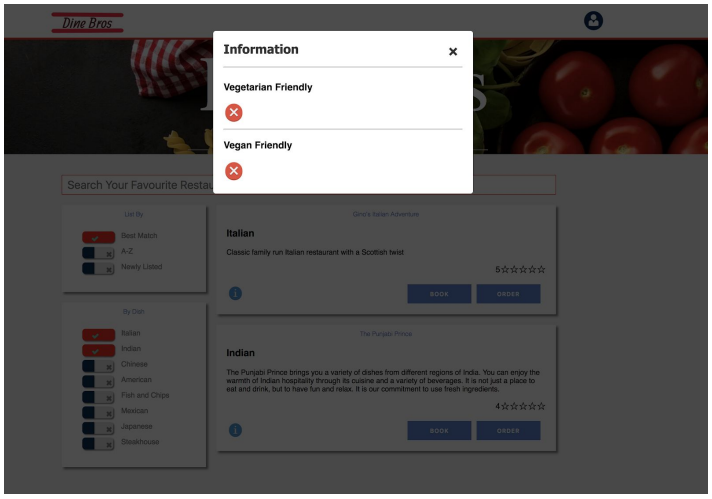


Figure 3.16 : 'Restaurant Information Prompt'

Another prompt that would show if the restaurant catered for particular groups needs in a stylish manner that reflected the rest of stylings in the web application.

FORM VALIDATION

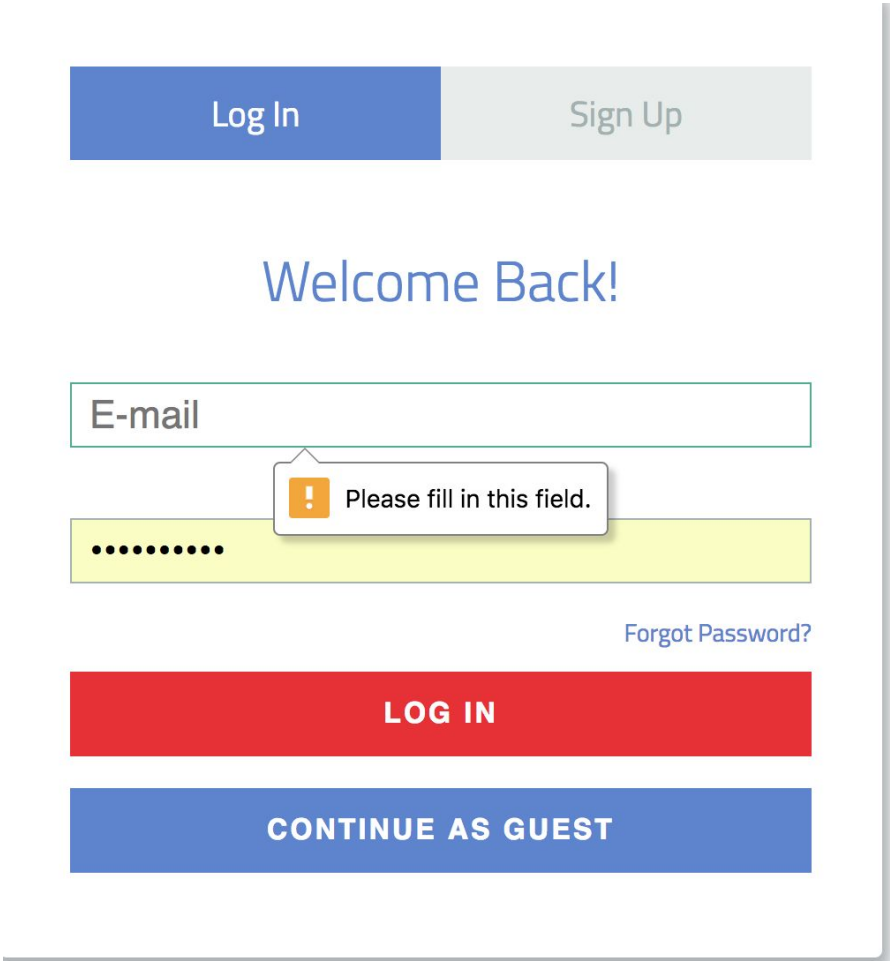


Figure 3.17 : 'User Form Validation'

To make sure that data being passed between the customer and the database that would be used by the staff applications was correct all forms were created with form validation. This allowed us to make sure the information was genuine and reliable.

All form validation responses were created with friendly prompt messages that gave a professional yet formal response to the customer.

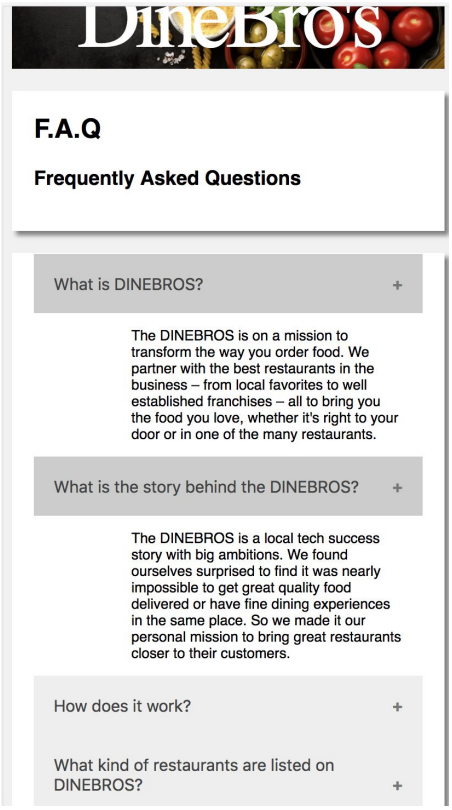
The design of all these forms ran concurrent with the rest of the design for the customer interface.

RESPONSIVE DESIGN

One of the functional requirements of the specification was that the system would be usable on different screen types and sizes. This meant that for the User we had to design for all screen types as a customer could potentially use any device to make an order or book a table.

Figure 3.18 : 'Mobile Header Navigation'

As seen in the above figure when the user was on a mobile device then certain icons would change and adapt to better the usability of the application on smaller screened platforms. As a bonus they still contained all the relevant functionality expected from the larger screen versions but was included in dropdowns better suited for mobiles and tablets.



For general pages the content was still centered to make full use of the users mobile device.

This worked similarly for tablet devices however not as streamlined as it was for the mobile view.

Figure 3.19 : 'Mobile Content'

Waiter Side

Staff Login Screen:

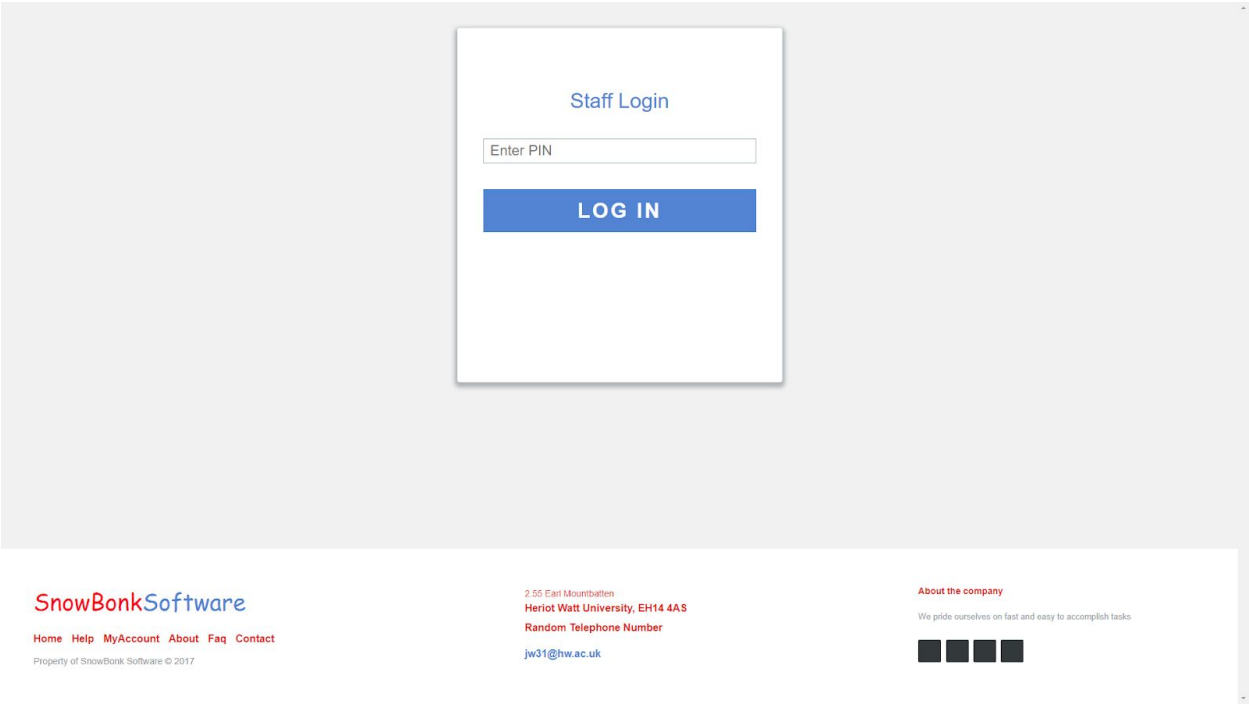


Figure 4.1 - Waiter System staff login screen.

This is the first screen a staff member trying to access the waiter system will be met with. It is a simple login system which only requires a single piece of information in order to pass, the staff members personal PIN number. Each staff member will have their own unique PIN number which when entered correctly will grant them access to the waiter system's dashboard of their specific restaurant providing that also have one of the following job titles: Waiter, Owner or Manager. If the staff member does not have one of these titles, if they are part of the kitchen staff for instance then they will not be allowed to progress forward to the waiter system dashboard and instead will be redirected back to the login page.

Waiter System Dashboard

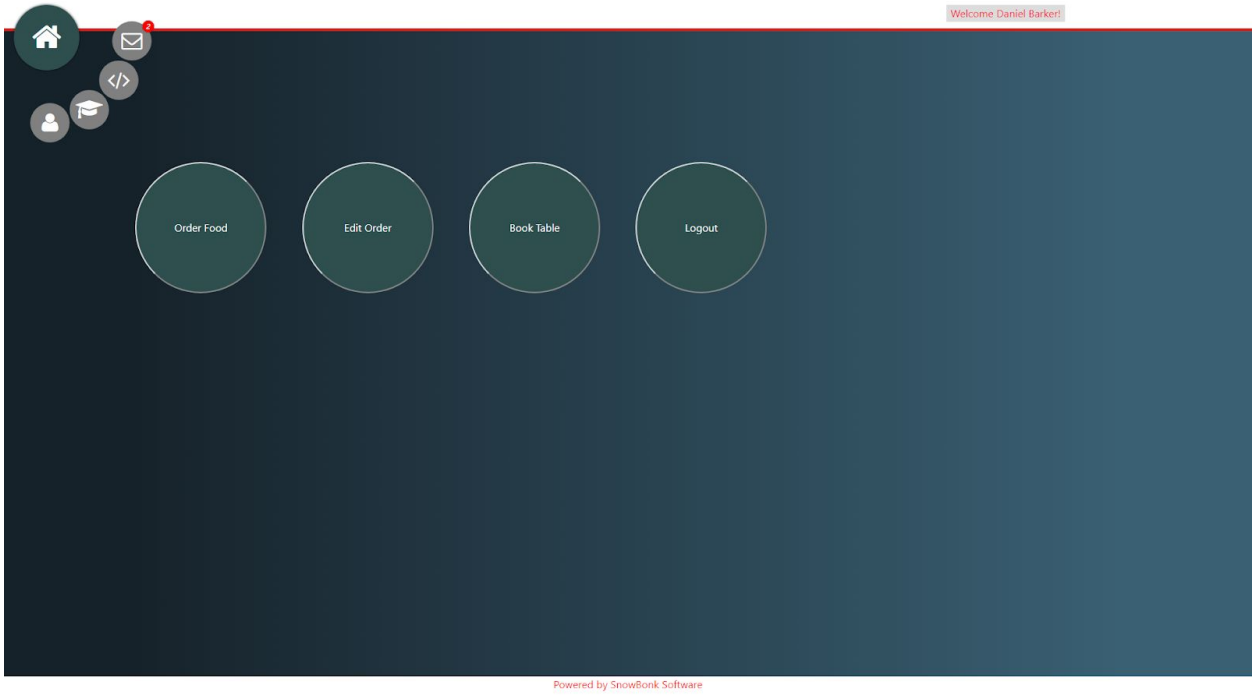


Figure 4.2 - Waiter System Dashboard.

http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/welcome.php

The waiter system dashboard is the first page a member of staff will be greeted with upon a successful login. Here a member of staff will see 4 buttons, from left to right the buttons are as follows: the first button labeled “Order Food” will take the staff member to another page which will display all of the food items offered by the restaurant which the staff member works at. Here they will also be able to add items to existing orders. The second button “ Edit Orders” will take the staff member to another page where they will be able to see a list of tables which have an existing order, from here they will be able to select a table and view the order details related to it. The third button “Book Tables” will take the staff member to another page where they will be able to look at all currently available tables in their restaurant, they will also have the ability to book specific tables when required for example if a group of customers where to come into the restaurant for lunch. The fourth button “Logout” simply destroys the staff members session and redirects them back to the staff login page.

In regards to the design of this page we chose to display the staff members main points of navigation as large easy to click buttons. The reason for this was due to assumption that the restaurants waiters would predominantly use some form of tablet therefore the large buttons would make the task of navigating through the system on the go or at tableside much simpler.

Some features that have been implemented on the waiter dashboard include a waiter notification panel which can be seen in the top left of the image. When a customer who is currently dining in the restaurant requests a waiter a notification will be sent out to each staff member logged into the system. The number of notifications is displayed in the red circle and once clicked will open a popup box detailing which table requires assistance, once read the notification can be discarded by the waiter. This feature makes the restaurant seem more user friendly and welcoming towards customers. Another feature that has been implemented on the waiters dashboard is the name displayed in the top right hand corner of the image. This shows which staff member is currently logged into the system allowing for ease of communication between staff members.

Waiter System Order Items

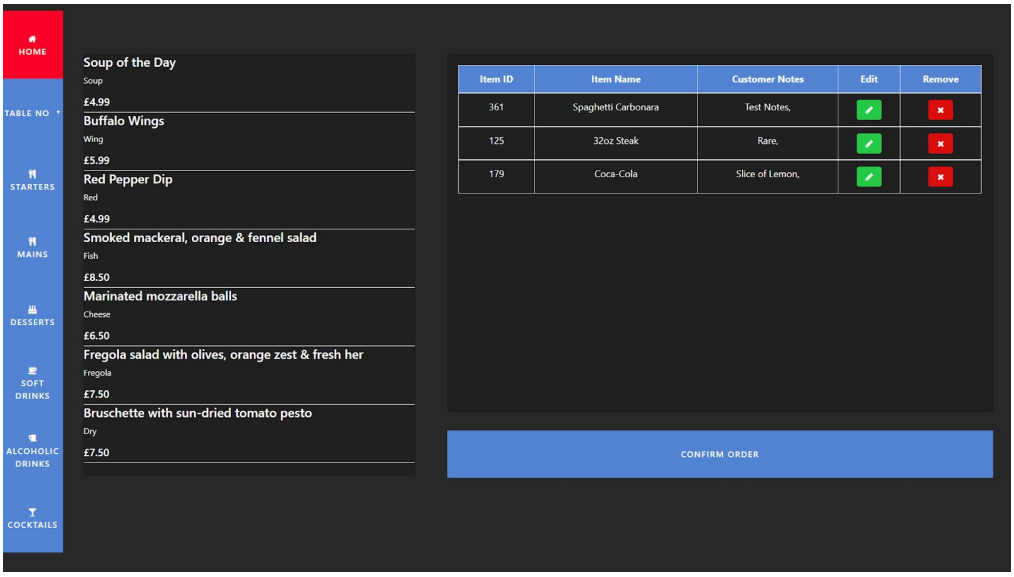


Figure 4.2 - Waiter System Order Items Screen.

http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/Menu.php

On this page a staff member will be able to view and their restaurants menu items as well as append to existing table orders. To make things easier on the staff member we have implemented a visually pleasing, easy to navigate sidebar which will provide the staff member with all of the tools they will need in order to complete their task. Among these tools lies a home navigation button which will redirect the member of staff back to the waiter system dashboard when clicked therefore making it easier for staff members to jump between certain jobs. Another tool located on the sidebar is the “Table Select” dropdown menu, here the member of staff will be able to see what tables already have existing orders in place which makes it easier for them to select the table they wish to add items to. The remaining buttons located on the side bar are various menu categories which when clicked display the restaurant items of that specific type in the menu box just to the right of the sidebar. Having these menu categories makes it easier for

the member of staff to navigate their way around the restaurants multitude of items in a quick and easy manner.

When an item is displayed in the menu box the member of staff is then provided with the opportunity to click on it if they so wished. If they did decide to click on it then a new screen would appear over the existing one. Details about this part of the design can be found in figure 4.4 and figure 4.5.

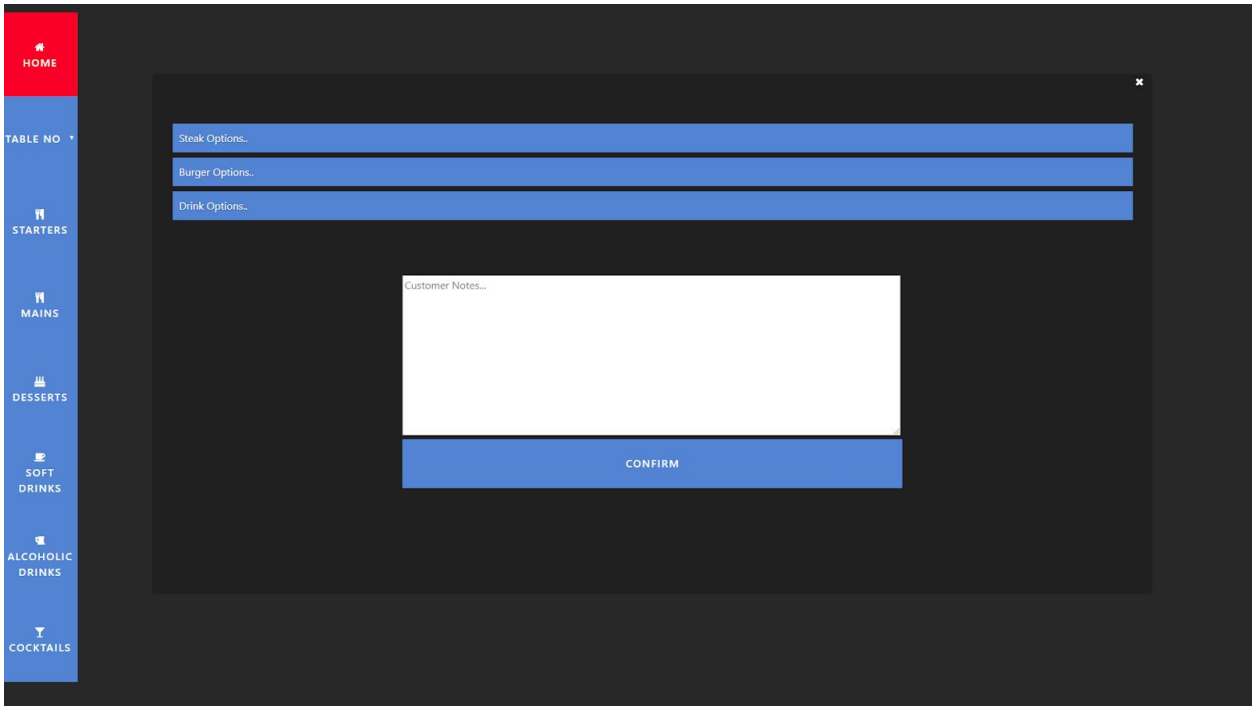


Figure 4.3 - Waiter System Order Items Notes.

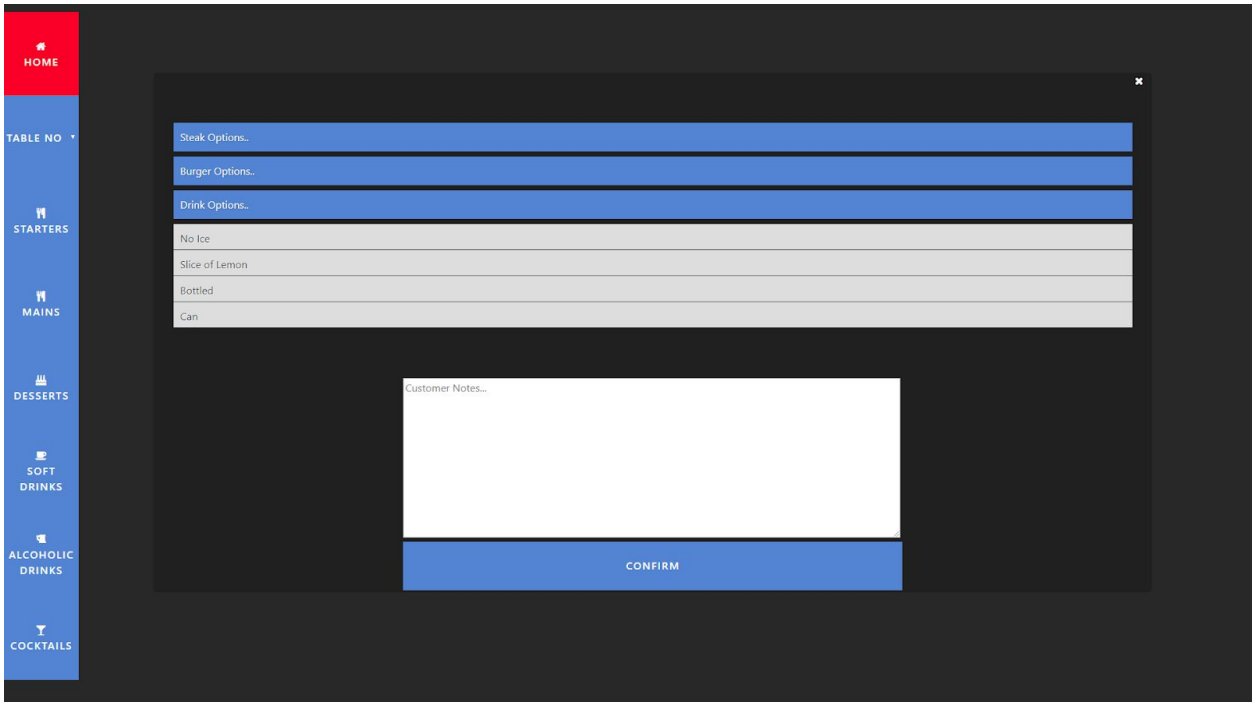


Figure 4.4 - Waiter System Order Items Notes.

This new screen would hold predefined customer note options for popular items such as steak, burgers or drinks. The screen would also provide the member of staff with the option to input any custom notes the customer may have in a text box. Once the member of staff is satisfied with the notes they can then press the “confirm button” which will take them back to the previous screen and populate the “orders table” on the right with the selected item and corresponding note.

Looking back at figure 4.2 the “orders table” provides 2 buttons with each item that is added to it. The green button is the “update notes” button, if a staff member where to click on this button they would be able to edit the notes of the item held within that table row. The red button is the “remove item” button, if a staff member where to click on this button then the table row holding that item would be removed. These two buttons provide members of staff with a way to error check orders before actually confirming the order.

Once the order is complete it is up to the staff member to confirm the order by clicking on the big “confirm button” at the bottom of the screen. Once this button has been clicked each order item is given an expiry time equal to the current time + 10 minutes and then it is added to the database. The expiry time given to each order item provides a 10 minute window where that items notes can be updated or be completely removed from the order, after this time period expires the order will be final. Once and order is in the database the kitchen system can begin work on it.

Waiter System View Orders

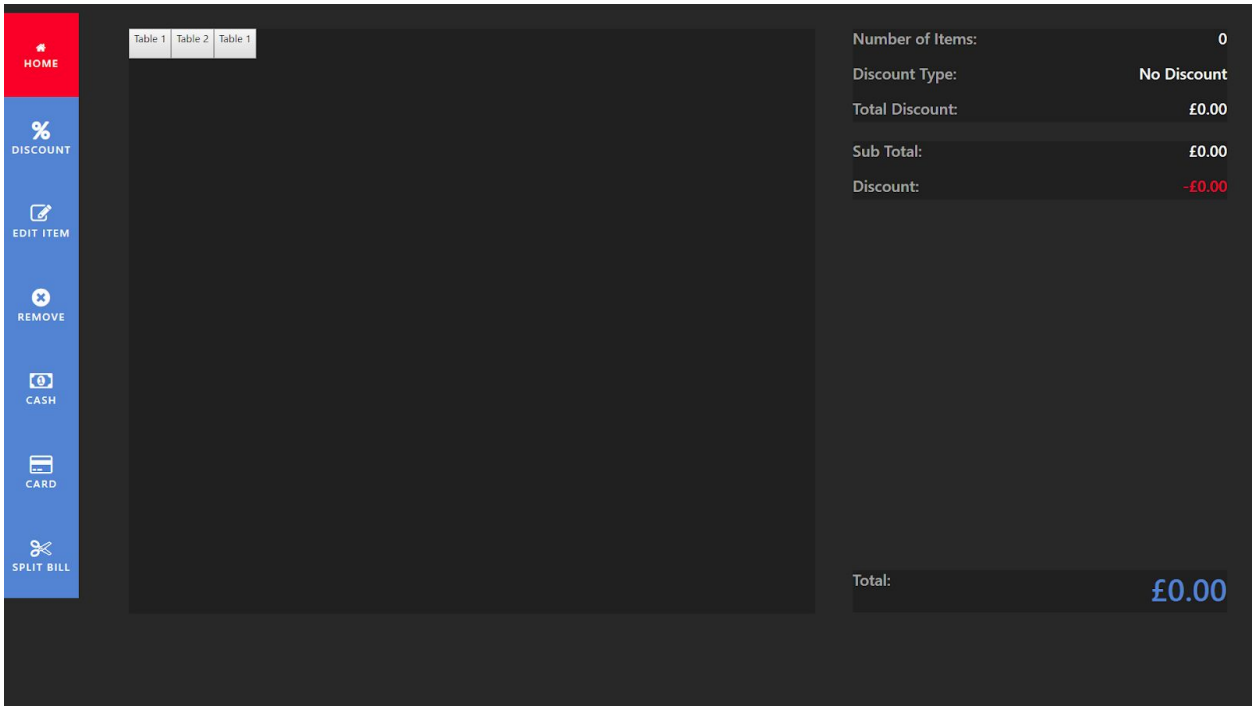


Figure 4.5 Waiter System View Orders

http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/editOrders.php

On this page a staff member is able to see a list of buttons each of which correspond to a table within the restaurant which has an order in place. To the right of the list container are three containers which provides various pieces of information based on the order currently selected, since no order is currently selected these containers will not display anything other than their default placeholders.Once a table is selected a new page will appear, more details about this can be seen in figure 1.6. All sidebar options except the home button are disabled on this page.

Waiter System View/Edit Orders

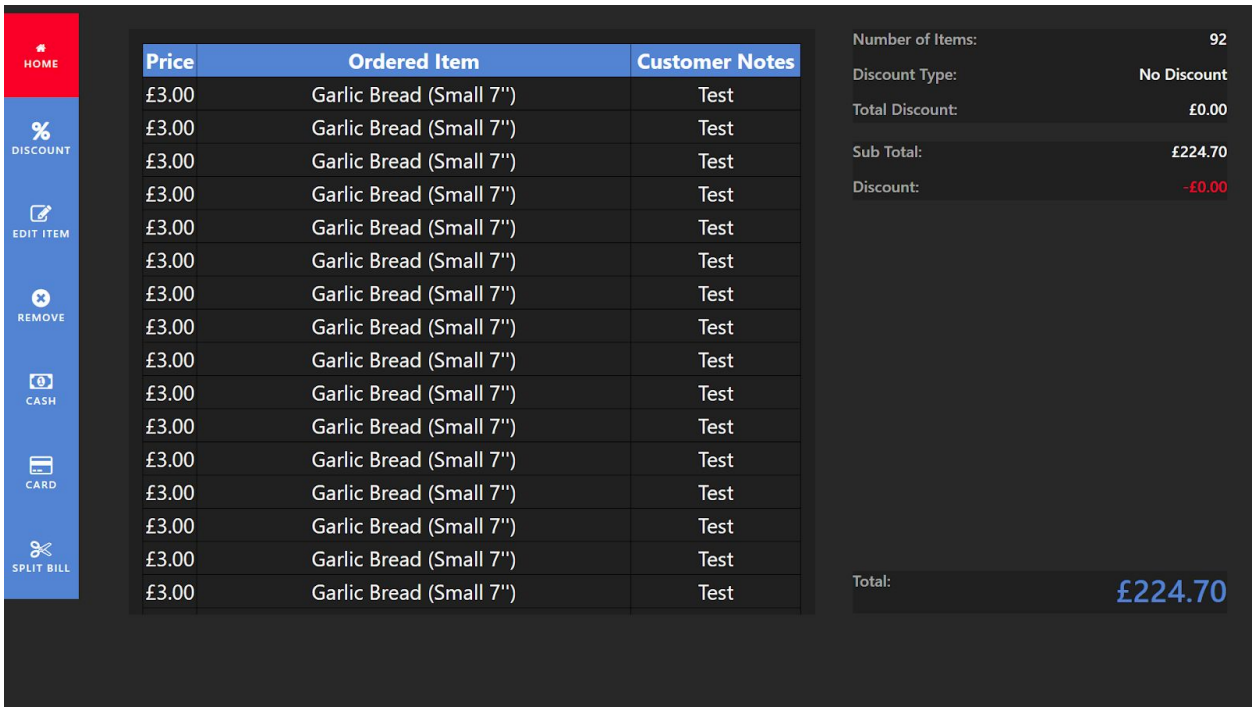


Figure 4.6 Waiter System View/Edit Orders

This is the page that appears once a member of staff has selected a table whose order they wish to view/edit. Now the containers on the right have partially filled up with information about the order and all of the sidebar buttons are now active. This page follows a similar design to the "Order Items" page the side bar holds all the necessary tools need to complete any task the staff member may need to perform therefore keeping everything neat and tidy as well as simple.

Some of the buttons located on the sidebar include the "Update" and "Remove" buttons. If a member of staff where to click on a row in the "Orders Table" that table row would now be selected and turn red (this can be done to multiple rows). Once the staff member has finished selecting the table rows if they then click on the "Update" button they will be able to update the items notes of those rows(if the items expiry time has not already expired). Similarly if the staff member where to click on the "Remove" button once they had finished selecting their desired table rows then those items/item will be removed from the order (as long as the items expiry time has not already expired).

Another button that can be located in the sidebar is the "discount button". When clicked the "discount button" will open a new screen on the same page which will display a list three possible discounts that may be given to n order. The first one is called "The Family and Friends Discount" and will provide customers with 10% off of their order. The second discount option is called "Staff Discount" which can be used to give staff members 30% off their orders. The final discount option is called "Refund", this one can be used to give 100% off of a customer's order. Once one of these discounts options has been clicked a percentage of the order total will be subtracted from the original total to give a new total. For

more details about this process see figures 1.7 and 1.8. The purpose of implementing these discounts was to make it easier for restaurants to incorporate meal deals for example a “happy hour” event where certain menu items have a reduced price for a certain time during the day. They were also implemented to help resolve any issues that might occur between customers and the restaurant itself where perhaps the customers food wasn’t entirely to their satisfactory, a good discount or refund here may be what's needed in order to help maintain a high overall customer satisfaction rating.

Discounts

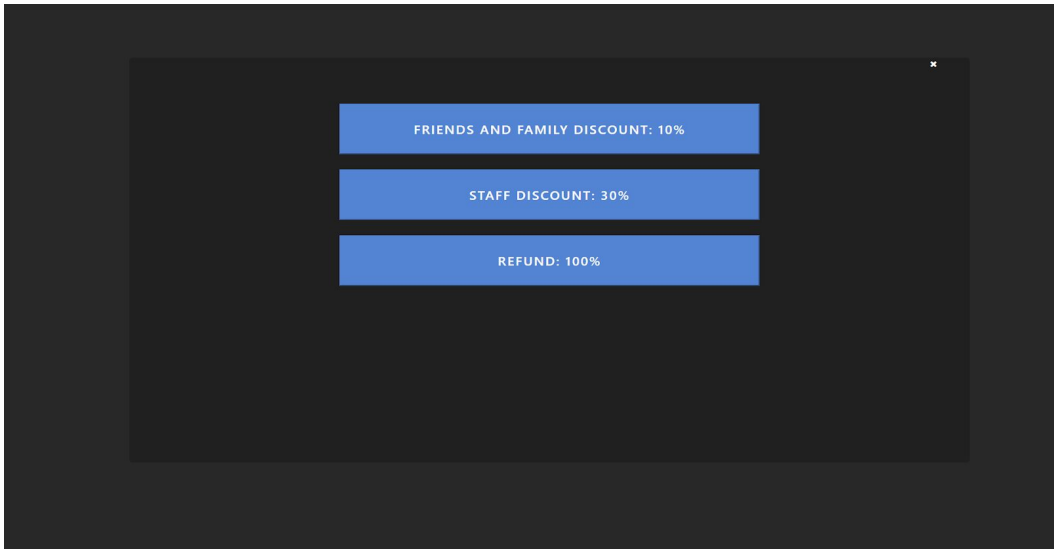


Figure 4.7 discounts screen

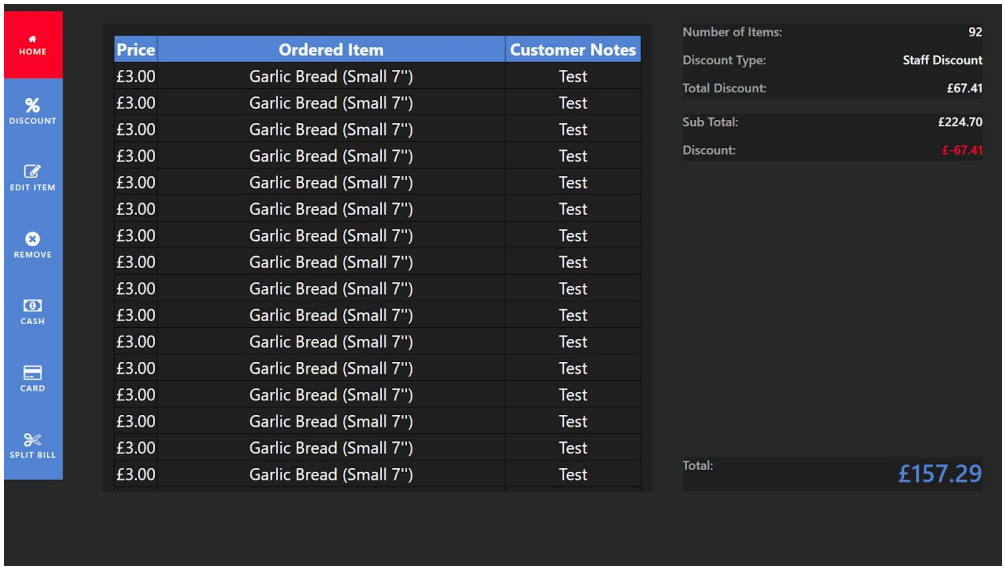


Figure 4.8 Order with “Staff Discount” applied.

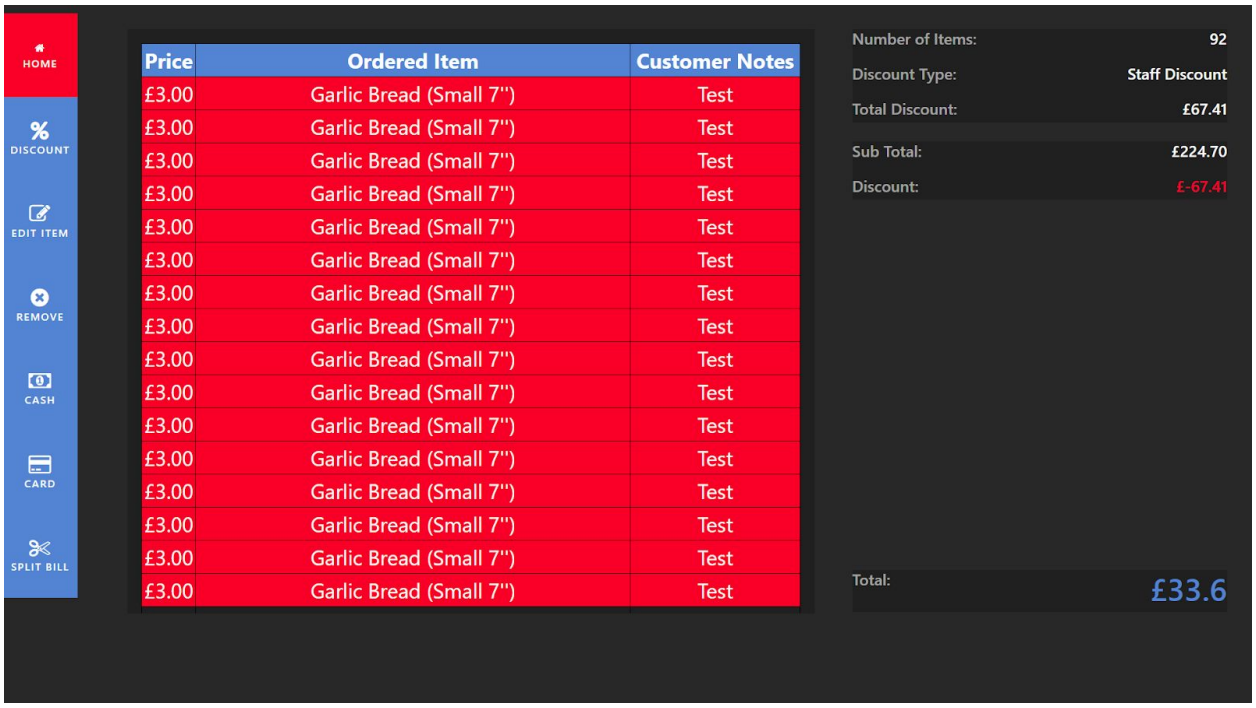


Figure 4.9 Split Order Selected Items

Similarly to selecting tables rows to remove from the order a staff member can also select the table rows which they wish to split into a seperate bill. Once the tables rows have been selected if the staff member clicks on the "Split Bill" button located on the sidebar then the total value of those items will be calculated and replace the previous total. If a discount was active before the split then the split items total value will also be discounted.

Once the bill has been split it is now time for the customer to pay. If the customer is paying with cash then we would navigate over to the sidebar and click on the "Cash" button. Once we have clicked on this a new screen will pop up on the page, on it will be the price the customer needs to pay, a set of predefined cash values and a keypad (see figure 4.9).

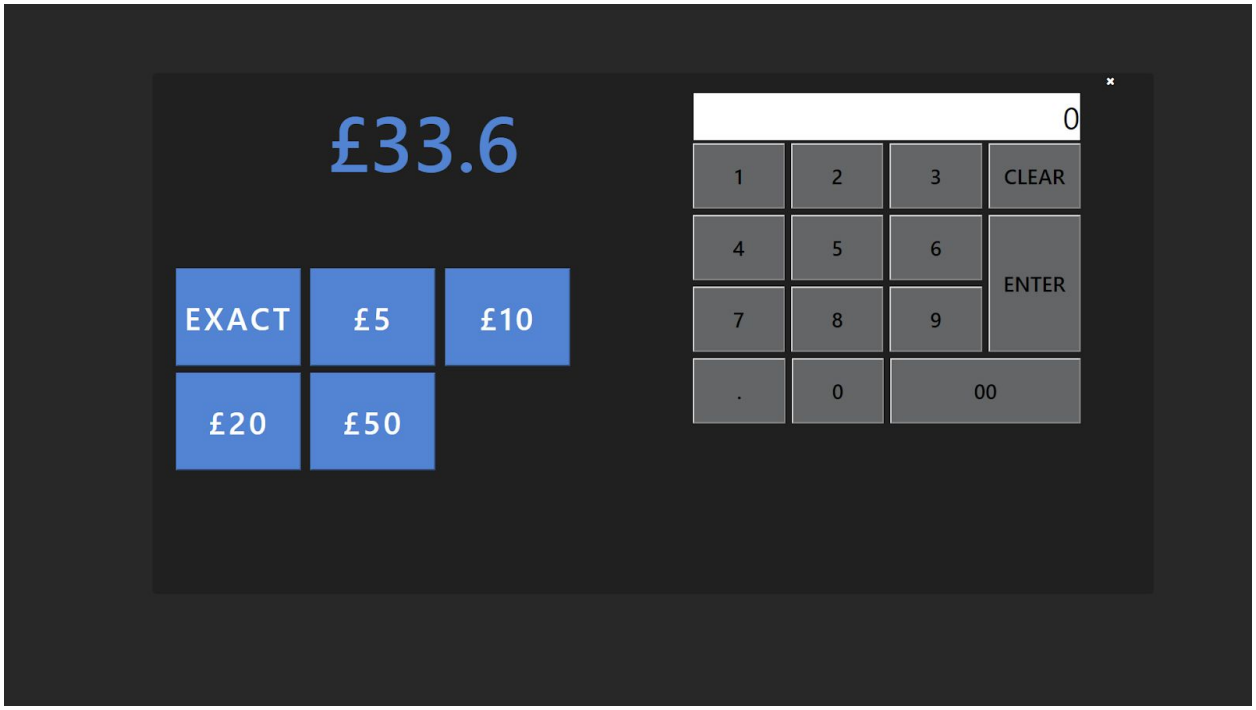


Figure 4.9 Cash screen

Hitting any of the predefined buttons on the left will deduct their value from the total value, except for the "EXACT" which will deduct the full amount simulating that the customer has paid in full and requires no change. On the right hand side is a keypad which can be used to input the exact amount of money handed to you. Once the total value reaches zero or below then a "confirm button" will appear (see figure 4.10), when clicked a new screen will appear which will display the total amount of change(if any)(see figure 4.11) the member of staff owes the customer. After being on the change screen for a few seconds the member of staff will be redirected back to the original screen where they selected the items to split from the table. Once they arrive back here the page will refresh and the items that have been paid for will no longer be visible in the table. This process is repeated until there are no items left in the table, once this happens the completed order will be added to the "order_history" table in the database and dropped from the original "orders" table. The table the customer was sat at will also become available again ready for any new customers that may walk into the restaurant.

The "Split Bill" option was implemented to provide for certain groups of customers who wish to pay separately rather than all together.

The "Cash Screen" was implemented to provide the staff with a quick and easy way of dealing with transactions.

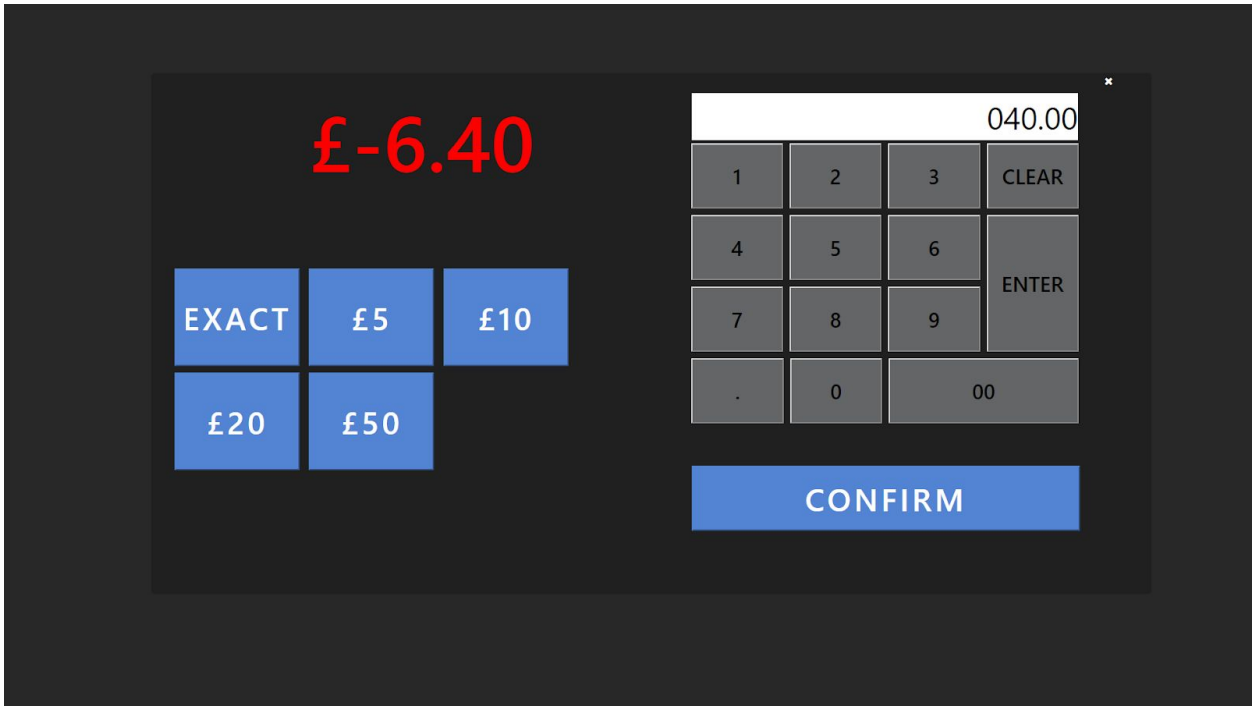
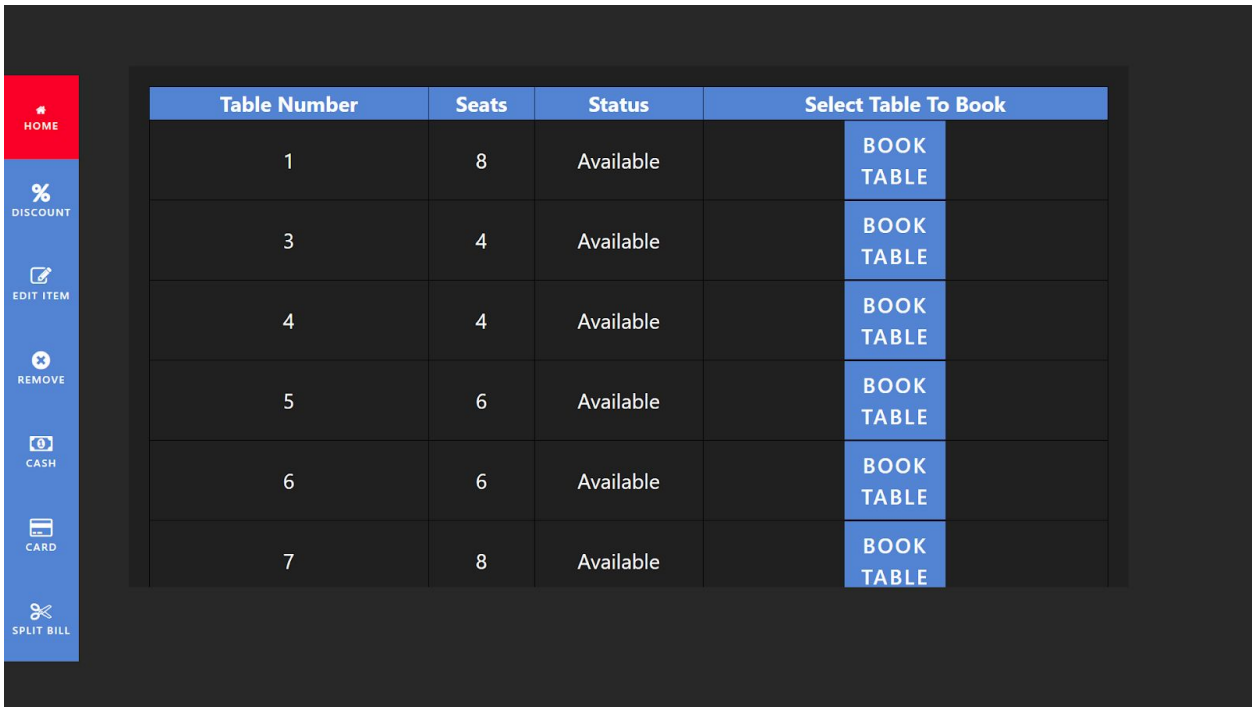


Figure 4.10 Cash screen with confirm button

Figure 1.11 Change Screen

BOOK TABLES



4.12 Available Tables

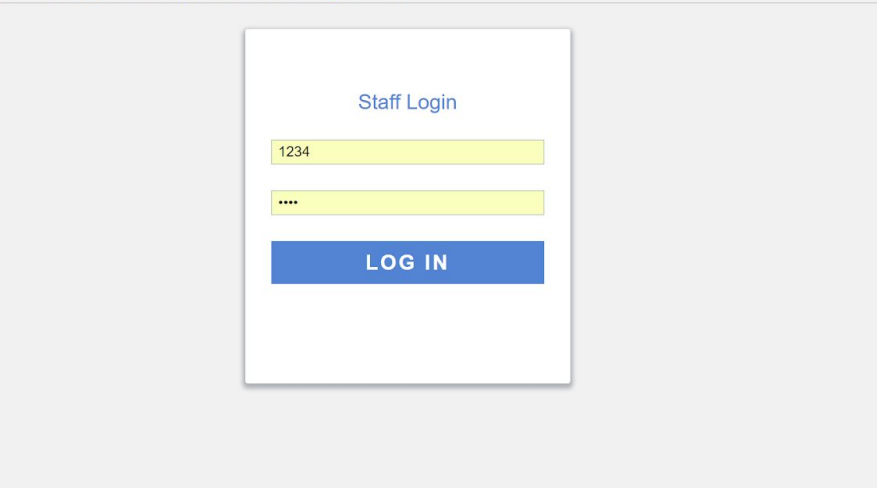
http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/availableTablestest.php

On this page a staff member is able to see what tables are currently available in the restaurant. If a customer was to come in the staff member could view the list of available tables see which one is most appropriate for the customer. Once a table has been chosen the staff member can click on the "Book Table" button which will update the database changing the tables status from "Available" to "Unavailable". Once a table is made unavailable it also also creates an active order which will allow staff staff to add items to it.

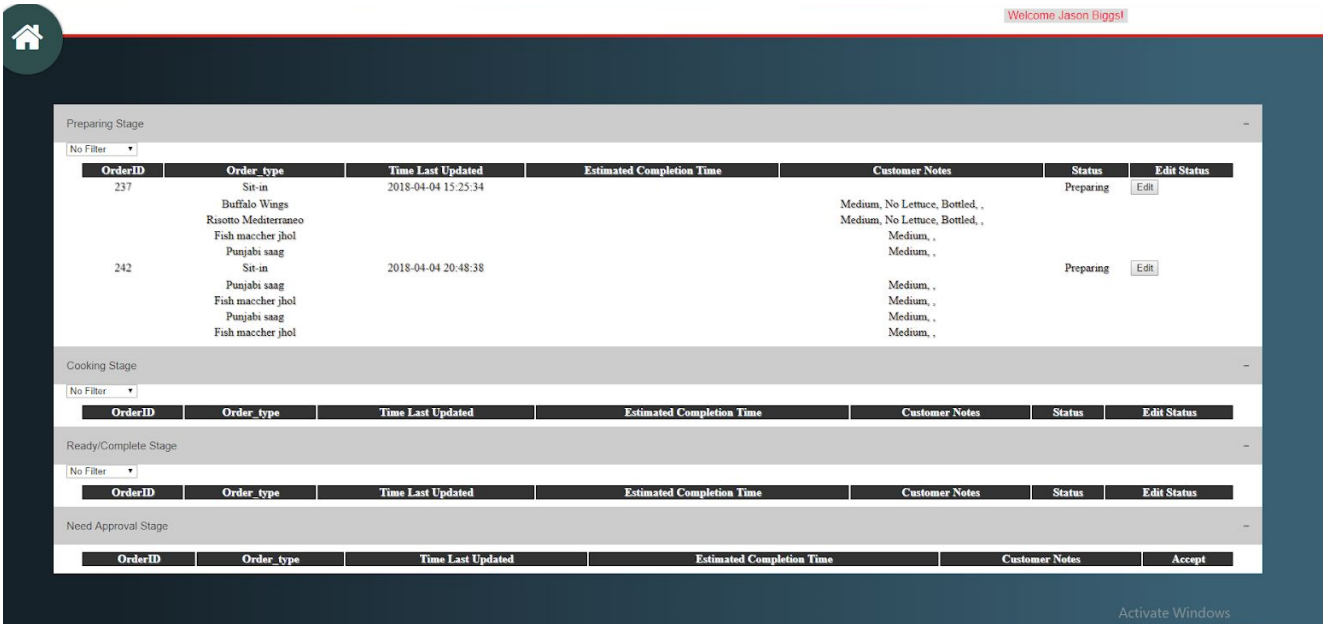
On this page a staff member will be able to view and their restaurants menu items as well as append to existing table orders. To make things easier on the staff member we have implemented a visually pleasing, easy to navigate sidebar which will provide the staff member with all of the tools they will need in order to complete their task. Among these tools lies a home navigation button which will redirect the member of staff back to the waiter system dashboard when clicked therefore making it easier for staff members to jump between certain jobs. Another tool located on the sidebar is the "Table Select" dropdown menu, here the member of staff will be able to see what tables already have existing orders in place which makes it easier for them to select the table they wish to add items to. The remaining buttons located on the side bar are various menu categories which when clicked display the restaurant items of that specific type in the menu box just to the right of the sidebar. Having these menu categories makes it easier for the member of staff to navigate their way around the restaurants multitude of items in a quick and easy manner.

When an item is displayed in the menu box the member of staff is then provided with the opportunity to click on it if they so wished. If they did decide to click on it then a new screen would appear over the existing one. Details about this part of the design can be found in figure 4.4 and figure 4.5.

Kitchen Side



This simple but effective staff login page is less flashy than the standard customer login page, it allows for usernames and passwords rather than requiring an email, this means a kitchen could have a generalised account that can't be hacked into with social hacking methods such as password changes linked to emails. In addition it means that a whole kitchen might only have to memorise the one account, or if different levels of permissions are desired they can have multiple levels of access.



We decided to go with a number of dividers for the kitchen section as the singular table approach ended up being clunky with too many orders. Each divider holds a different category of order, Preparing, Cooking, Complete, and Needs Approval. With the exception of “Needs Approval” all the tables have the ability to manually change the status of an order, this allows for a great deal of control at the touch of a button. The needs accepted bracket is slightly different, in that it doesn’t simply edit an order but accepts the order and moves it to the preparing section so that the corresponding chef can begin work on the order.

In the case of different stations such as a chef who only works on the Starters, each divider is accompanied by a filter in the top left of the divider, this filter can filter down to starter, main, and desert, so when a lot of orders come in, the lsit can be shortened down to only the items your station works on, this can be done separately per list, so a head chef can also filter the not accepted list to see how much of each dish will be required and as such decide which orders to cancel.

Implementation Methodology

Snowbonk based the development model of the project by an incremental approach. This allowed us to interleave our specification, design, implementation and testing. Interleaving our development allowed us to be iterative and revise our plan during certain changes of the final development stage. Some of the principles behind the Agile Manifesto corresponded with how Snowbonk will worked on the project.

Incremental Process Benefits Reflection

Some benefits stated in the Stage One report that aided in our development are,

1. ***"The cost of accommodating changing customer requirements is reduced."***
   For the circumstances we faced during our development of the system, this was the truest benefit. We had to drastically change our development plan throughout stage 3 and with the incremental process this made it easy to manage. We could easily review what we had already achieved and decided how to proceed against what our next iteration plan was.
2. ***"More rapid delivery and deployment of useful software to the customer is possible as a quicker working prototype is produced."***
   We had a basic working prototype available for the system by the second week in January. This helped us grap very quickly where the project was heading and how we were going to approach new tasks and features.
3. ***"It's easier to get customer feedback on the development of the software and see how much has been implemented.***"
   This was evident at our manager demo where we demonstrated a vast range of functionality.

Incremental Process Problems Reflection

Some problems stated in the Stage One report to this approach are,

1. ***"The process development isn't visible."***
   To the development team this was not the case, with weekly or bi-weekly system reviews we could measure our progress of the system. However, without delivering weekly deliverables to our managers they were left quite in the dark the state of development.
2. ***"System structure tends to degrades as new increments are added."***
   We believed this was an issue that was faced during development when we had started on additional functionality in Stage 3 that was to be disregarded. We had factor features into the ordering system to remove stock as orders came through, however, with cutting out this feature we had to reiterate over the code and remove unnecessary clunky code.

For the Incremental delivery methodology we believe we had great success overall, helping to boost the project through rough development periods and keeping us on track with what we were accomplishing. Not every iteration was managed to be kept to 100% but allowed us to reiterate to see what needed to be completed. Some parts of development were a more loosely applied agile approach, especially around the times of the strikes where the team would work a bit freely on what they deemed necessary.

Iterations Overview

Generally for every iteration we had managed to keep to the what was planned. We discuss more in depth about what occurred during our iterations in the Project Evaluation document.

Database Implementation:

For the implementation of the Database, we followed an Iteration model as outlined in the Stage 2 documentation. For the most part we were able to follow the iteration cycles for the database implementation and more often that not finished ahead of schedule, we didn't once have to push development back a cycle. For the database, Iterations 4, 5 and 6 were focused on the main implementation, while later Iterations like 9 and 15 were making small alterations due to design change as we progressed with the application.

Data Dictionary

Customer:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
customer_id	The unique ID given to all customers.	int (7)	No	Yes	Yes
guest_id	Random number given to users if they chose guest	int(64)	No	No	No
name	Derivative of customer's firstname and surname.	varchar (100)	No	No	No
fname	Customer's firstname.	varchar (50)	No	No	No
sname	Customers surname.	varchar (50)	No	No	No
email	Customers email address.	varchar (100)	No	No	No
password	Customer's password.	varchar (50)	No	No	No
address	Customers address.	varchar (100)	No	No	No
addressL1	Customers address line 1.	varchar (50)	No	No	No
addressL2	Customers address line 2.	varchar (50)	No	No	No

city	Customer's city.	varchar (50)	No	No	No
postcode	Customers postcode.	varchar (7)	No	No	No
telephone_no	Customers telephone number.	varchar (11)	Yes	No	No
dob	Date of birth of customer	date	No	No	No
age	Customer's age derived from DOB.	int (3)	No	No	No

BILL:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
bill_id	The unique ID given to a bill.	int (7)	No	Yes	Yes
amount	Total cost of the order.	decimal (4,2)	No	No	No
date	Date of order.	timestamp	No	No	No
status	Current status of order payment. Either "Paid" or "To be Paid"	enum ('Paid', 'To be Paid')	No	No	No
order_id	Unique ID given to each order.	int (7)	No	No	Yes
restaurant_id_FK	Reference to restaurant	int(7)	No	No	Yes

RESTAURANT:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
restaurant_id	The unique ID given to all restaurants.	int (7)	No	Yes	Yes

rname	Restaurant name.	varchar (64)	No	No	No
rtype	Restaurant type.	varchar (64)	No	No	No
rrating	Restaurant rating	enum(1,2,3,4,5)	No	No	No
Description	What type of restaurant it is	varchar(500)	No	No	No
image_veggie	Vegetarian image	varchar(500)	No	No	No
image_vegan	Vegan Image	varchar(500)	No	No	No

ITEM:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
item_id	The unique ID given to each menu item.	int (6)	No	Yes	Yes
name	The name of the item.	varchar (50)	No	No	No
type	The type of menu item e.g. Main Course, Soft Drink, Alcoholic Drink etc.	enum ('Starter','Main Course','Dessert','Soft Drink','Alcoholic Drink','Cocktail')	No	No	No
description	Description of the item.	varchar (300)	No	No	No
price	Price of the item.	decimal (4,2)	No	No	No
restaurant_id_FK	Unique ID given to each restaurant.	int (7)	No	No	Yes
Image_Source	Image	varchar(250)	No	No	No

ORDER_DETAILS:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
orders	The unique ID given to order details.	int (7)	No	Yes	No

customer_notes	Comments from the customer.	varchar (500)	No	No	No
edit_expiry_time	Time for orders	time	No	No	No
order_id_FK	Unique ID to represent each order.	int (7)	No	No	Yes
item_id	The unique ID given to each menu item.	int (7)	No	No	Yes

ORDER_STATUS:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
status_id	Unique ID to represent the status of the order.	int (1)	No	Yes	Yes
status	Status is used to keep the customer informed of the status of their order.	enum ('Preparing','Cooking','Ready for Collection','On Route','Ready','Delivered')	No	No	No

STOCK:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
Stock_id	The unique ID given to all stock items.	int (11)	No	Yes	No
item_id_FK	The unique ID given to each menu item.	int (6)	No	No	Yes
Quantity	Quantity represents the number of items in stock.	int (6)	No	No	No

ORDER_HISTORY:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
history_id	Unique ID to represent the order history.	int (7)	No	Yes	No
order_id_history	Unique ID to represent each order.	int (7)	No	No	No
order_type_history	Unique ID to represent each order type.	enum ('Sit_In', 'Delivery', 'Collection')	No	No	No
status_id_history	Unique ID to represent the status of each order.	int (1)	No	No	No
customer_id_history	Unique ID to represent each customer.	int (7)	No	No	No
table_id_history	The unique ID given to all tables.	int (7)	No	No	No
restaurant_id_history	The unique ID given to all restaurants.	int (7)	No	No	No
date_time_history	Used to identify the time the order was made.	datetime	No	No	No

ORDERS:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
order_id	Unique ID to represent each order.	int (11)	No	Yes	Yes
order_type	Order type is used to identify what type of order the customer wants.	enum ('Sit-in', 'Delivery', 'Collection')	No	No	No
date_time	Used to identify the time the order was made.	timestamp	No	No	No
Order_completion	Order Status	datetime(6)	No	No	No
status_id_FK	Unique ID to represent the status of the order.	int (1)	No	No	Yes
customer_id_FK	The unique ID given to all customers.	int (7)	Yes	No	Yes
table_id_FK	The unique ID given to all tables.	int (7)	Yes	No	Yes
restaurant_id_FK	The unique ID given to all restaurants.	int (7)	No	No	Yes

RESERVATIONS:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
reservation_id	Unique ID for each customer reservation.	int (7)	No	Yes	Yes
date_time	Used to identify the time the order was made.	datetime	No	No	No
table_id_FK	The unique ID given to all tables.	int (7)	No	No	Yes
customer_id_FK	The unique ID given to all customers.	int (7)	No	No	Yes

TABLES:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
table_id	The unique ID given to all tables.	int (7)	No	Yes	Yes
table_no	Each table has a unique number for each restaurant.	int (7)	No	No	No
no_of_seats	Number of seats available at each table.	int (2)	No	No	No
status	Table availability.	enum (‘Available’,’Unavailable’)	No	No	No
restaurant_id_FK	The unique ID given to all restaurants.	int (7)	No	No	Yes

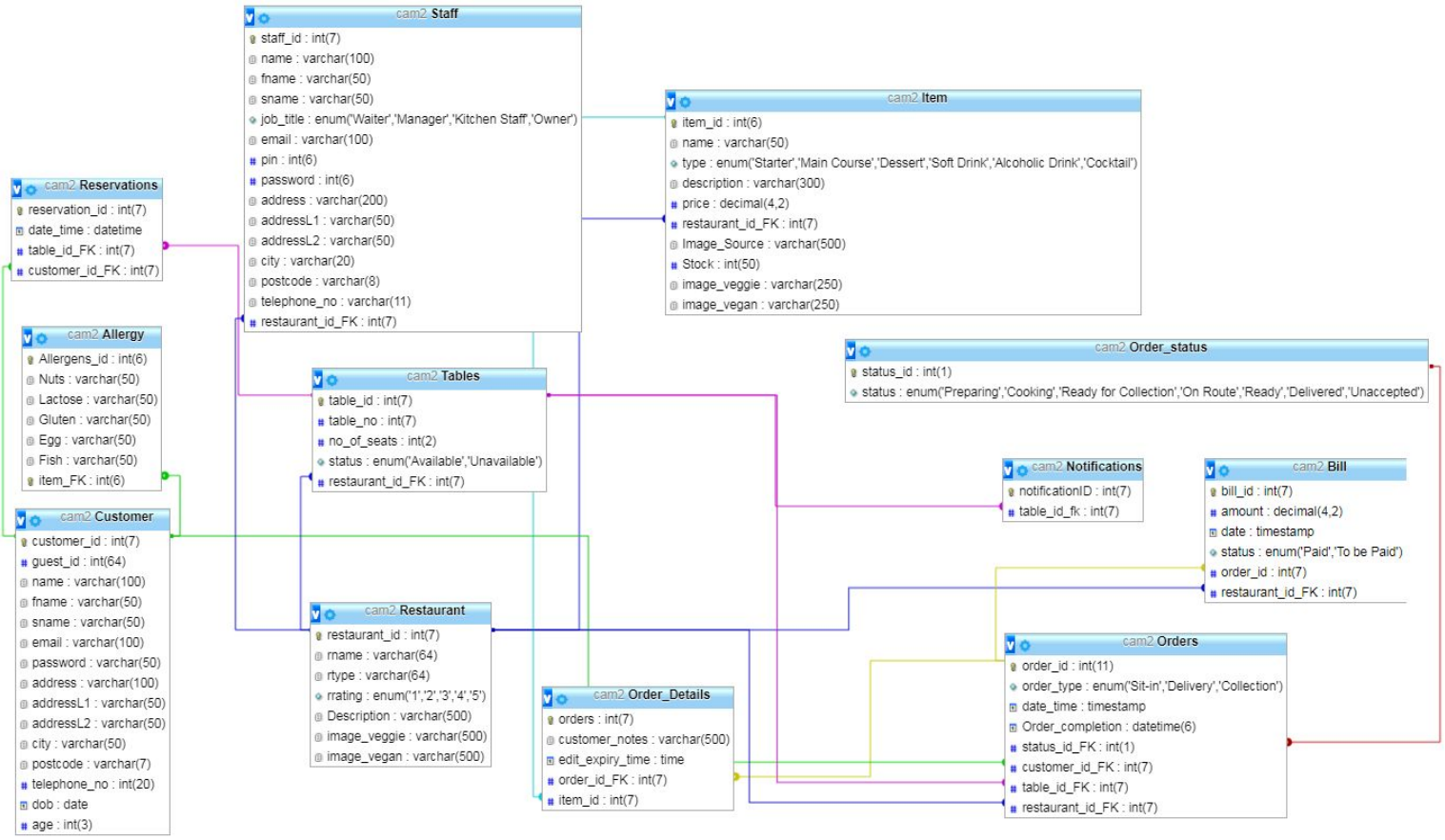
STAFF:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
staff_id	The unique ID given to Staff.	int (7)	No	Yes	Yes
name	Derivative of Staff firstname and surname.	varchar (100)	No	No	No
fname	Staff firstname.	varchar (50)	No	No	No
sname	Staff surname.	varchar (50)	No	No	No
job_title	Staff position.	enum ('Waiter','Manager' ,'Kitchen Staff', 'Owner')	No	No	No
email	Staff email address.	varchar (100)	No	No	No
pin	6 digit pin code	int(6)	No	No	No
password	Staff password.	varchar (50)	No	No	No
address	Staff address.	varchar (100)	No	No	No
addressL1	Staff address line 1.	varchar (50)	No	No	No
addressL2	Staff address line 2.	varchar (50)	No	No	No
city	Staff city.	varchar (50)	No	No	No
postcode	Staff postcode.	varchar (7)	No	No	No
telephone_no	Staff telephone number.	varchar (11)	Yes	No	No
restaurant_id_FK	Staff age	int (3)	No	No	No

NOTIFICATIONS:

Attribute	Description	Domain	Nullable	Primary Key	Foreign Key
notificationID	Unique ID for each notification	int (7)	No	Yes	Yes
table_id_fk	Used to identify the time the order was made.	int(7)	No	No	No

ER DIAGRAM



Final System Testing

Device Type

To test the Applications and how they interacted with different browser types we installed all popular browsers and ran our application. This was done for every page we have to make sure the testing was sufficient.

User

	Yes / No
Computer/Laptop	✓
Tablet	✓
Mobile Screen	✓

Staff

	Yes / No
Computer/Laptop	✓
Tablet	✓

As seen from these tests currently the Application for both staff and user are working as expected to. This means we can continue forward in development in the aesthetics of the Application.

Browsers

User

	Yes / No
Mozilla FireFox	✓
Chrome	✓
Internet Explorer	✓
Safari	✓

Staff

	Yes / No
Mozilla FireFox	✓
Chrome	✓
Internet Explorer	✓
Safari	✓

Forms

To test our forms we tested them in stages.
The first stage was used to test if you could enter empty data into the form and submit it. Because of the Scripts we use no empty data could be sent.
The Second stage was used for evaluating if they could only enter valid data such as email address, names, date of birth etc. Again all these tests were passed by the Scripts being ran on the forms.
The Third stage was making sure the form sent the relevant data. We used this in combination with the back end and all data was sent successfully.

	Yes / No
Log In / Sign Up	✓
Edit Details	✓
Payment	✓

Waiter Side Testing

Testing was spread out over multiple browsers, specifically Chrome, Firefox and Edge. We did this to make sure that our system was versatile and not just tailored specifically to one browser. The general approach to the system testing was to provide the test case with the data it required to run and then comparing the expected result with the actual result. If the results where the same then the test was successful however if things didn't go as planned a note was taken which would give a description of the problem. Any problems found where looked into and fixed as quickly as possible. Failed tests where always tested again after troubleshooting to make sure the problem had been fully resolved.

A description of each group of test cases can be found below:

Logging In With Different Job Titles

Appendix 2.1 - These group of tests checked to make sure that only authorised staff members were allowed to log in to the Waiter System and view its contents. Authorised Staff would include: Waiters, Managers and The Owner. Unauthorised staff would include: Kitchen Staff.

Accessing Waiter System Webpages Without Being Logged In

Appendix 2.2 - This group of tests made sure that nobody could bypass the Waiter System login page by just typing in a Waiter System page URL. All Waiter System Pages are wrapped in SESSION variables which can only be obtained via a successful login.

Applying Discounts to Table1's Order

Appendix 2.3 - Apply Discounts to Table 1s Order
This group of tests focused on testing various discounts on the same order to make sure that the correct values where being obtained before a customer finalised their transaction.

Apply Multiple Discounts to Table 1s Order

Appendix 2.4 - This group of tests were used to make sure that no more than one discount could be in place on an order at the same.

Splitting Table2's Order

Appendix 2.5 - This group of tests focused on selecting 2 specific items from table 2s order and creating a separate bill from them.

Adding Discount Then Splitting Table2's Order

Appendix 2.6 - This group of tests where use to check and see if discounts would still display correct values even if the splitting of bills takes place afterwards.

Transaction With Split Items

Appendix 2.7 - This group of tests checked to see if customers could actually pay for a group of split items individually.

Complete Transaction

Appendix 2.8 - This group of tests checked to see that once a customer had fully paid for there order that their order would be removed from the database and the table that they were sitting at status would change from “Unavailable” to “Available”.

Using Predefined Cash Buttons

Appendix 2.9 - This group of tests to make sure that the predefined buttons located in the “Cash model” would subtract their total from the already existing order total

Using Keypad

Appendix 2.10 - This group of tests checked to make sure that whatever was entered on the keypad in the “Cash Model” that it would be subtracted from the existing order total.

Adding Items to Table1

Appendix 2.11 - This group of tests focused on added items to already existing orders.

Editing Item Notes

Appendix 2.12 - This group of tests focused on whether or not an Items notes could be updated based on the time of edit. If the time of edit was after the Items expiry time then the notes should not be updated however if the time of edit is before the items expiry time then the notes should update.

Removing Items from Order

Appendix 2.13 - This group of tests focused on removing Items that are already in an order based off of the current time and the Items expiry time. If the current time of attempted removal if before the Items expiry time then the Item should be removed from the order, if not then it should remain.

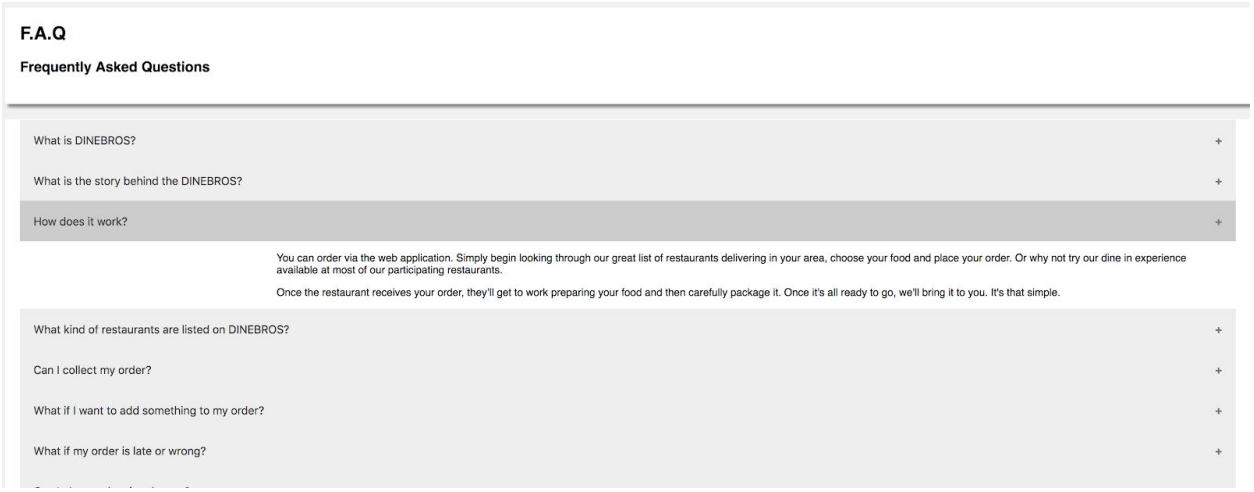
Waiter Notifications

Appendix 2.14 - This group of tests check to make sure that any notifications that the waiter had should be displayed in a list in order for them to see which tables required assistance.

Documentation and User Guide

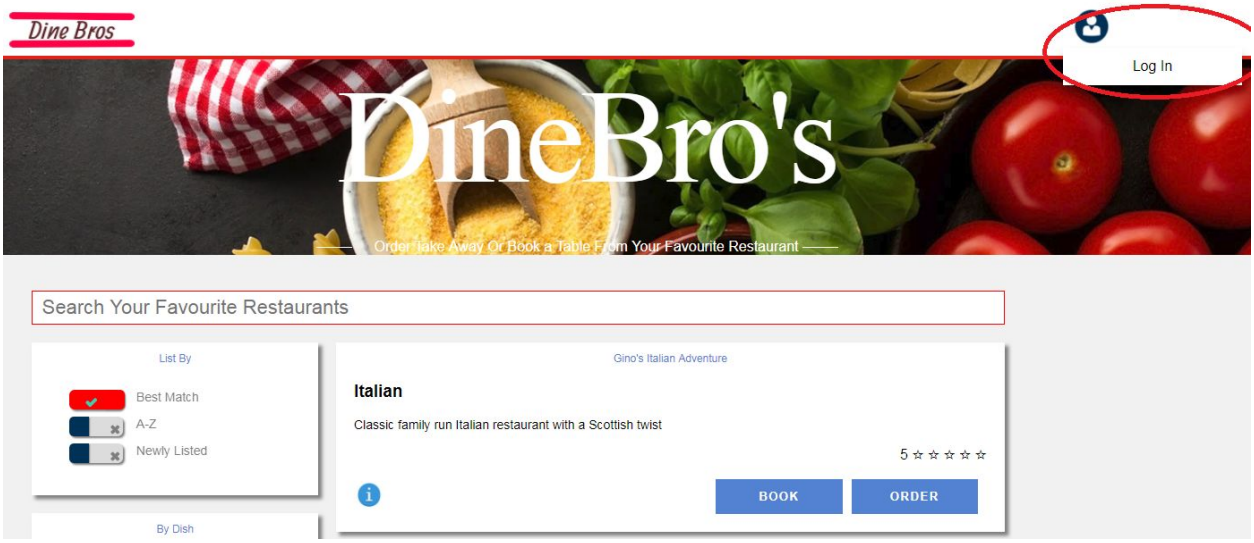
FAQ

For the Customer User System we have a small FAQ section which details problems and questions customers may ask when using the web application. This is quite vague however but has the potential to have easily adaptable further work carried out to answer problems the general user would have.

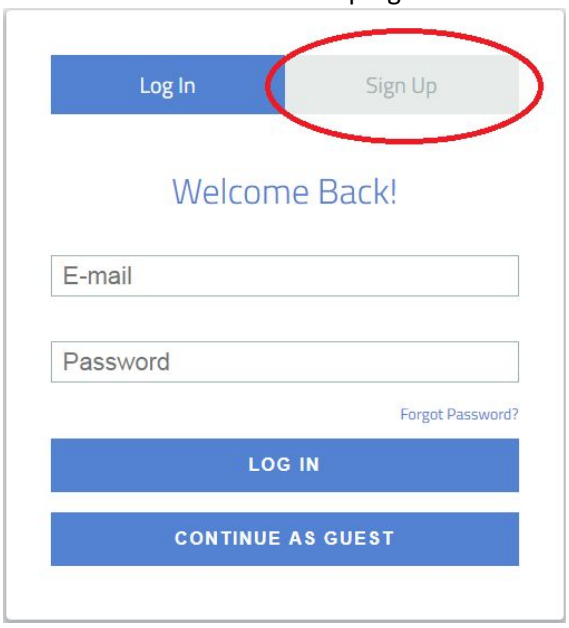


Customer User Guide

Sign-up/Login

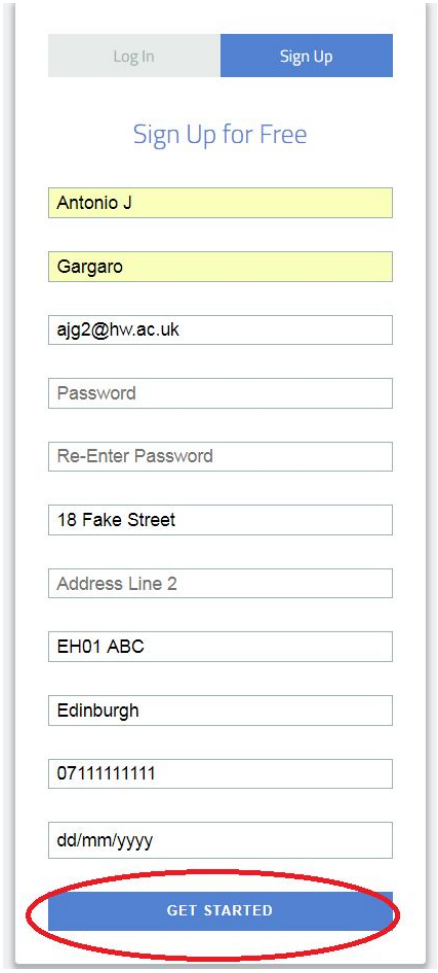


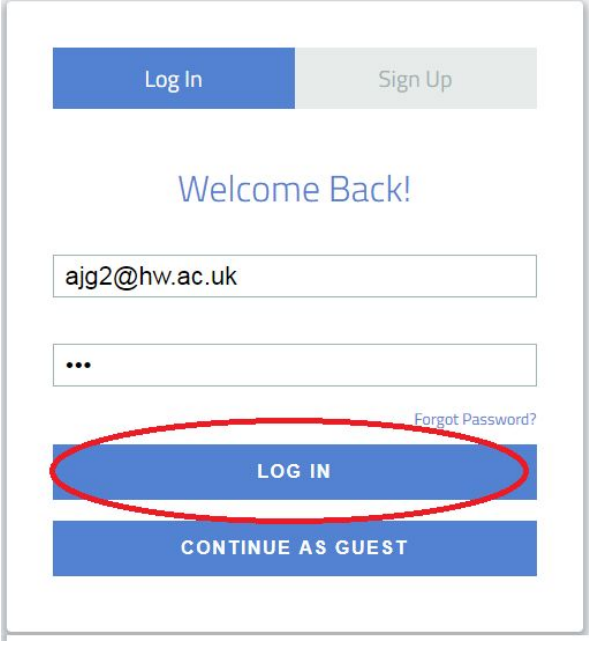
1. Hover your mouse over the account icon in the top right.



2. Press the sign-up tab.

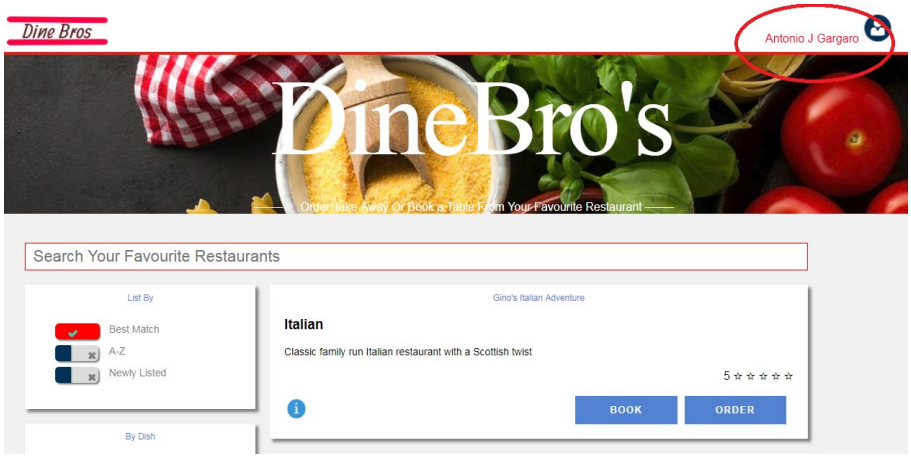
3. Enter your details in the fields and press 'Get Started'.





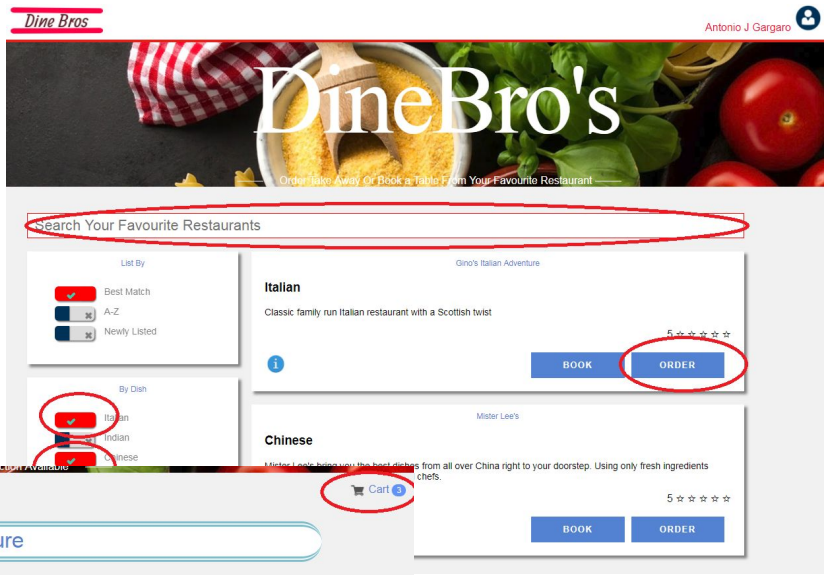
4. Then login with the details you entered.

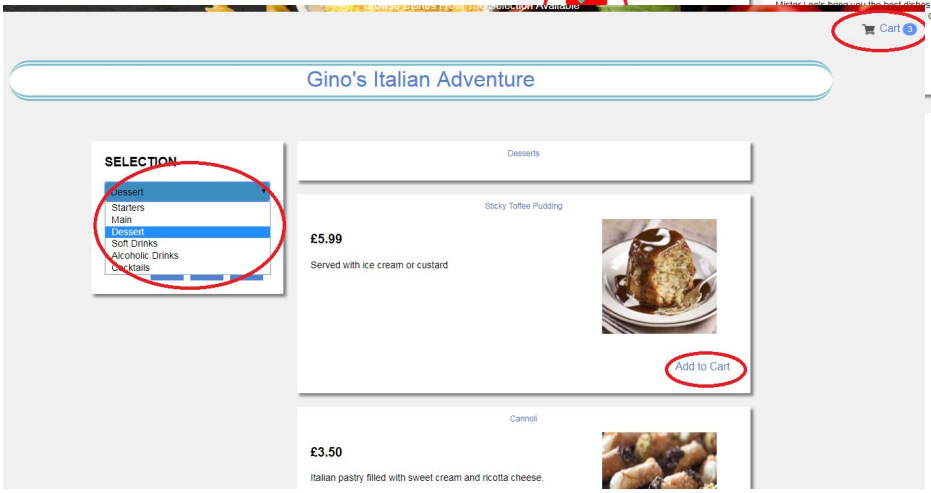
5. Then voila, you're signed in!



Making your first order

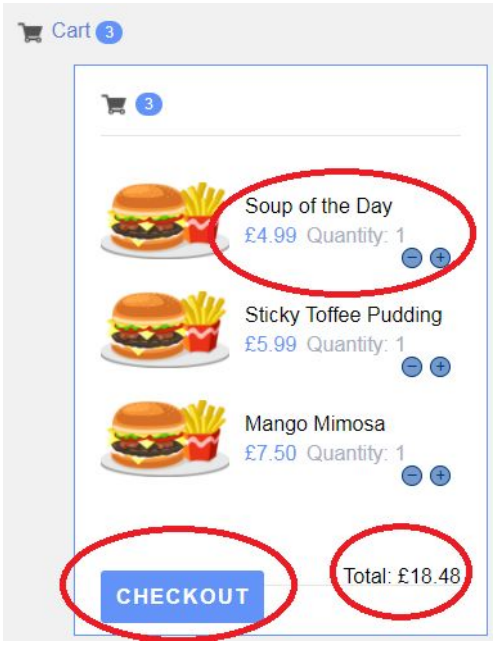
1. Select or filter your favourite restaurant from the options to the left and top of the body of content. Then select the order button on your chosen restaurant.



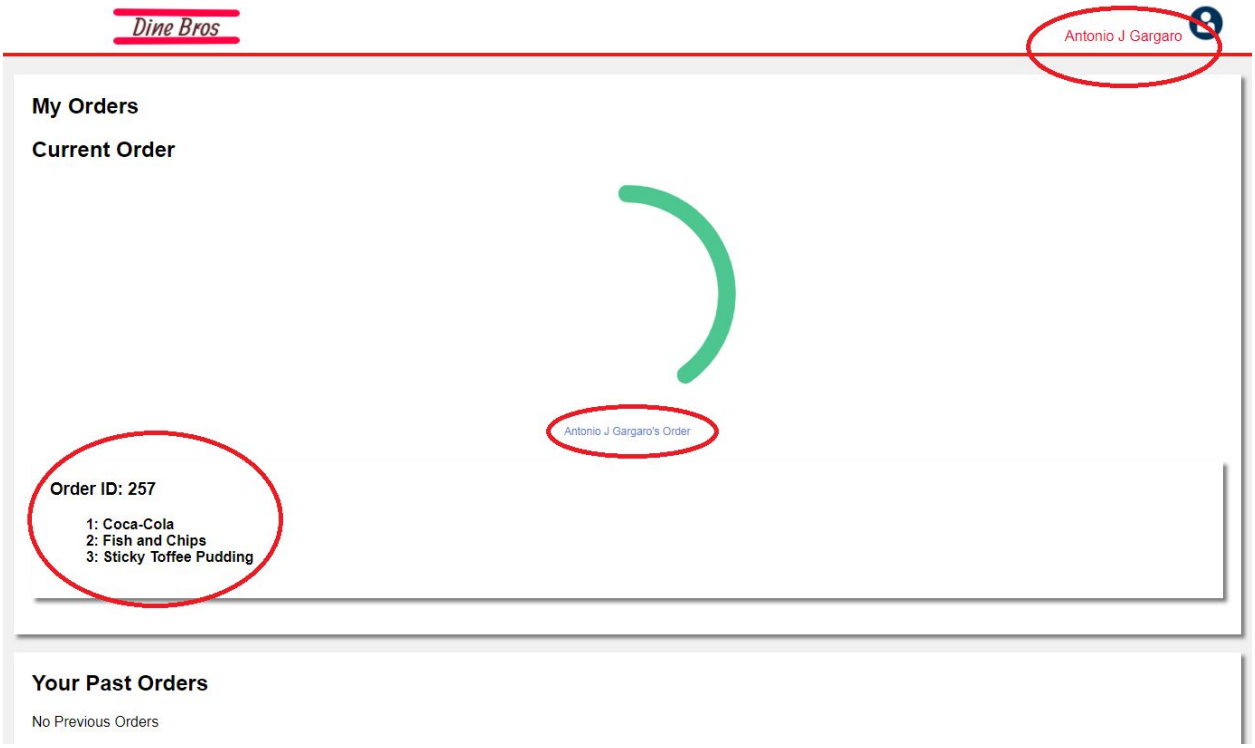


2. Filter your food menu choices with the dropdown selector on the left and add to your cart with the 'Add to Cart' button on the food item cards. Once chosen all your food items, click the cart icon in the top right of the body of content.

3. After selecting all your items you can review and edit your cart options, review your total amount and then checkout.

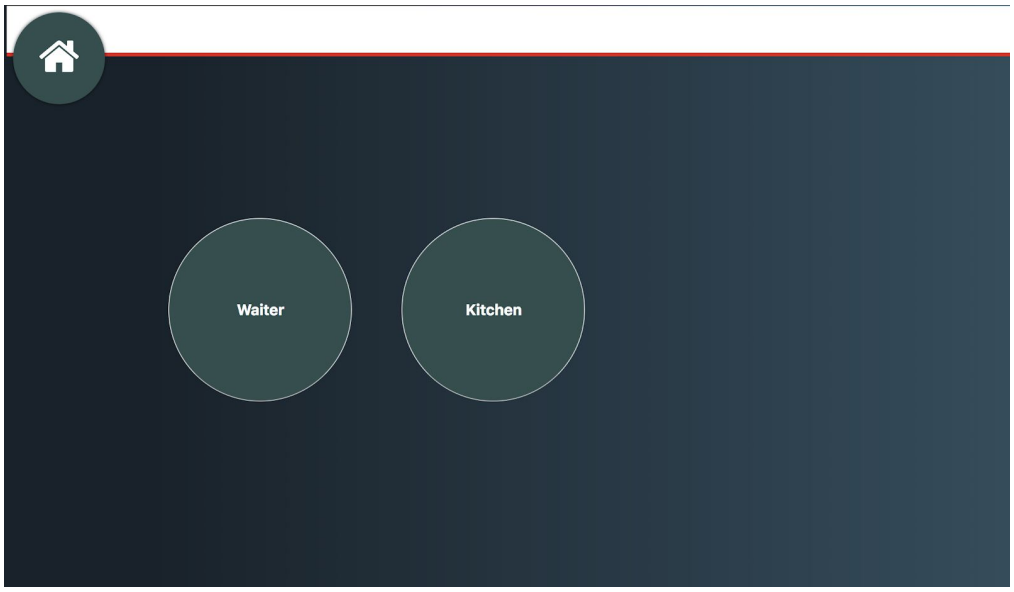


4. Then voila, you're order has been completed!

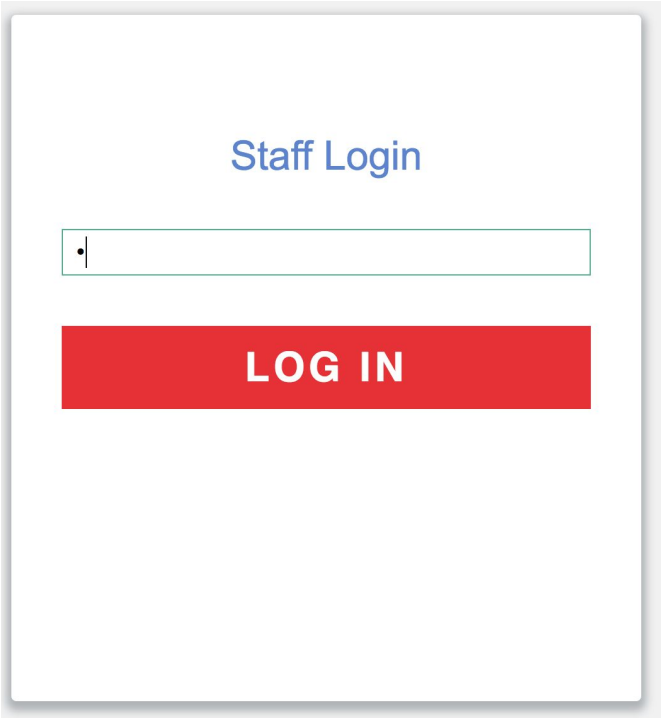


Staff User Guide

From this screen the staff member can choose which department they need to use i.e. either the kitchen or the waiter department.

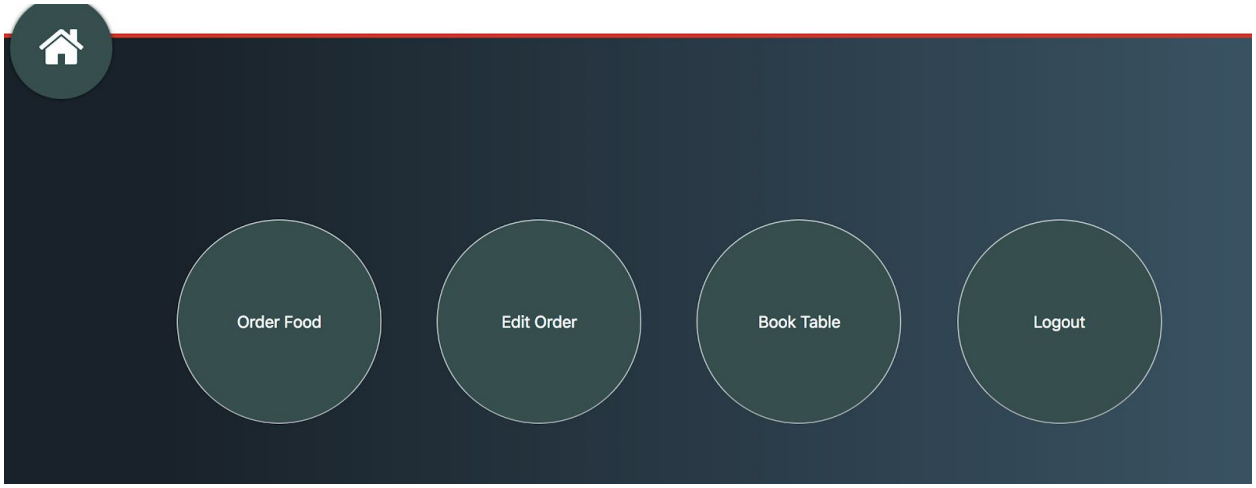


Waiter



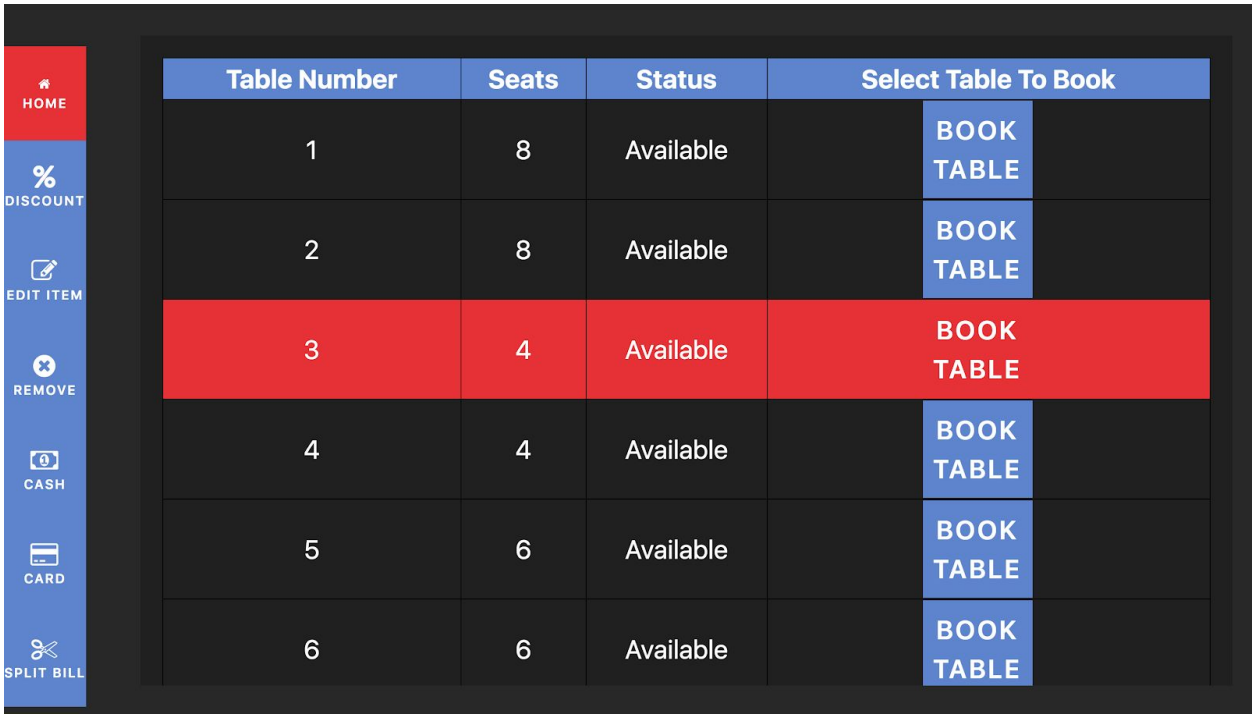
From here the staff member need only enter their pin and they will be directed to the waiter area.

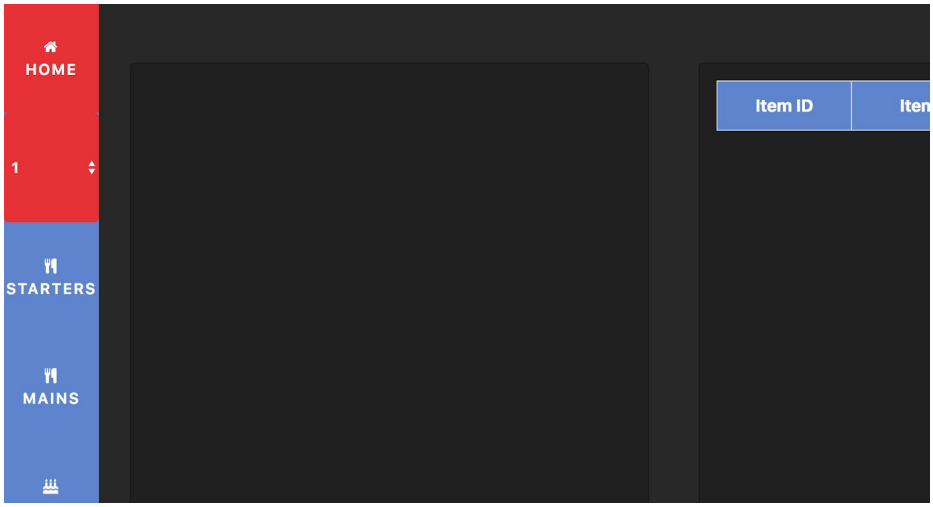
Here the staff member can easily choose which option they want to proceed with completing.



To create a booking, the user first the staff member needs to select the book table option from the panel.

From here the staff member just has to click on the book table button from the available tables and confirm the booking.

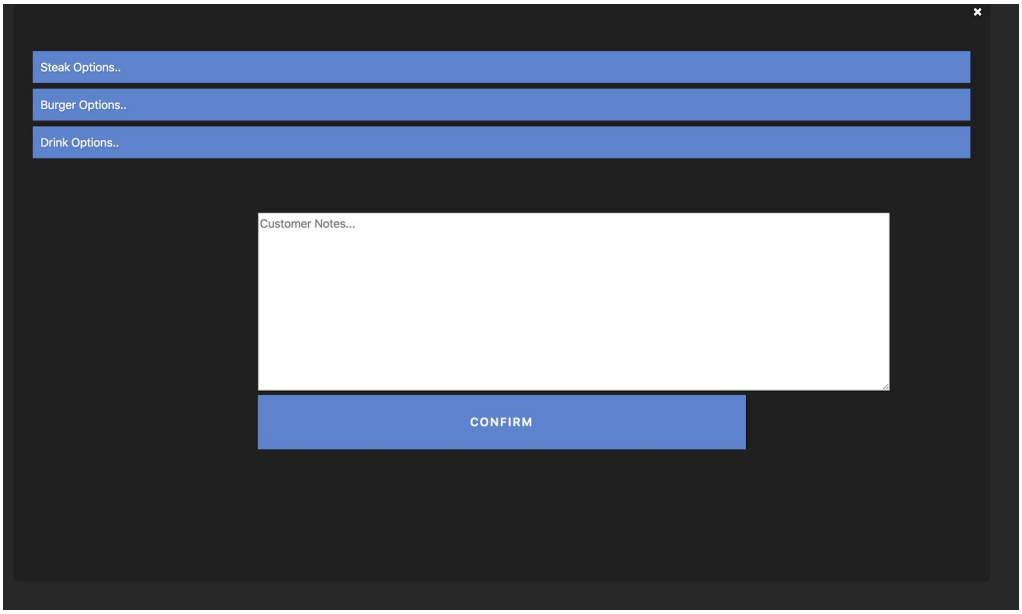




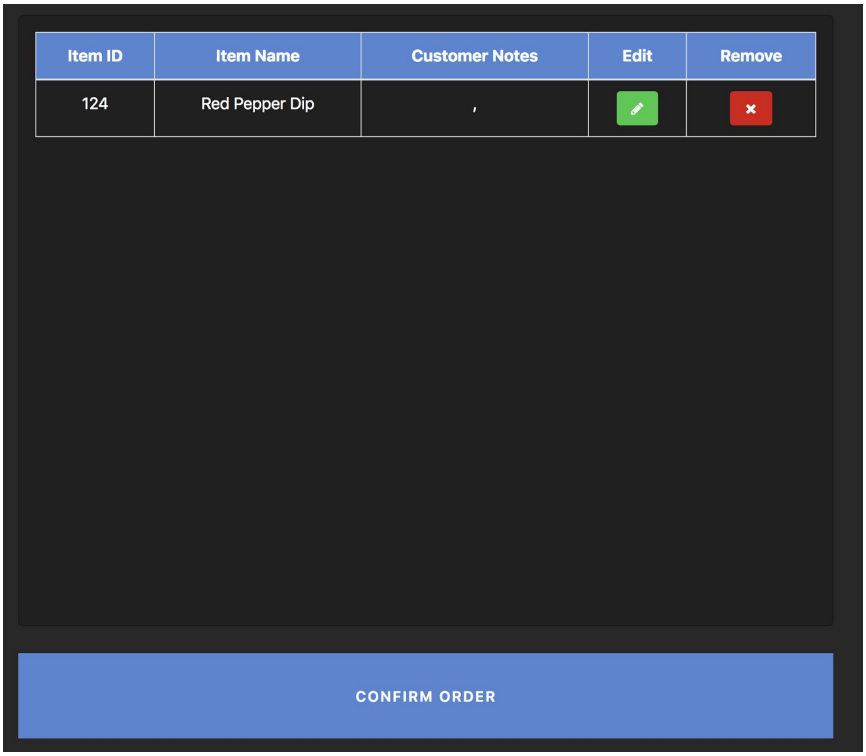
To create an order for the customer the staff member just needs to return to the user panel and click on the “Order Food” panel and they will be directed to the order page.

From here the user just needs to click on the button under the home button and select a table.

Once that is done the staff member can select any of the menu section buttons to bring up a list of menu items that are produced by that restaurant including soft drinks, alcoholic beverages and cocktails.

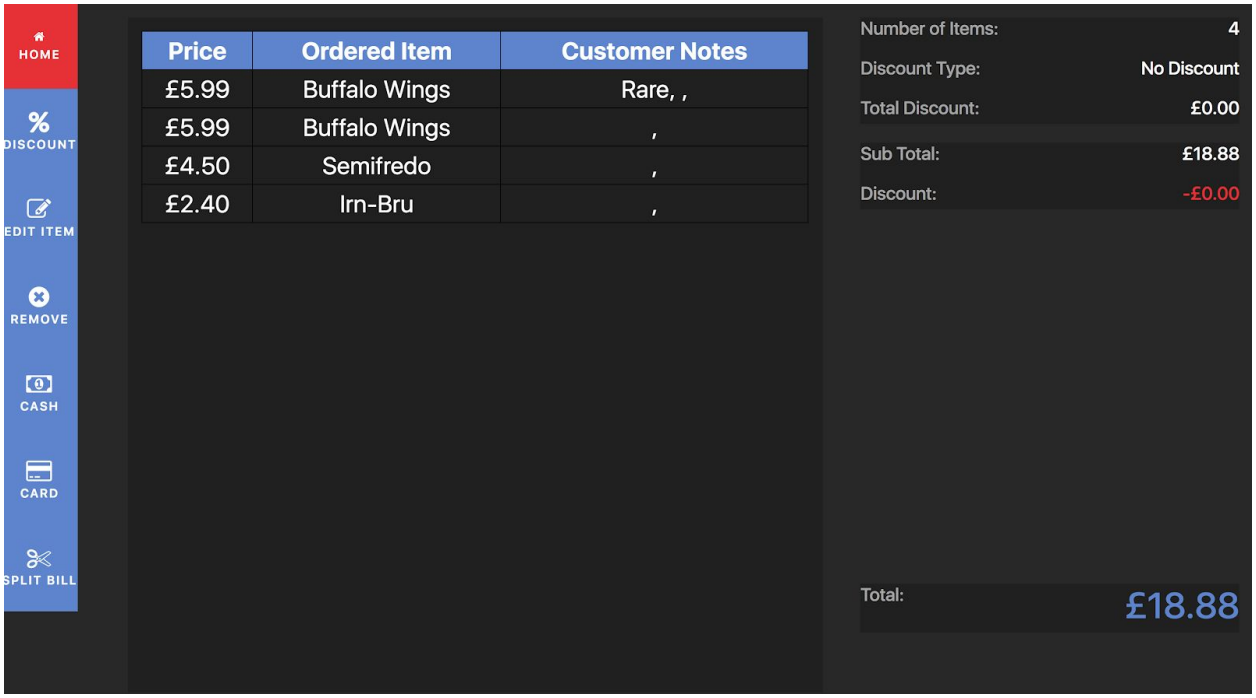


Once an item is selected a window will appear allowing the staff member to easily add any options and notes to suit the customers needs.



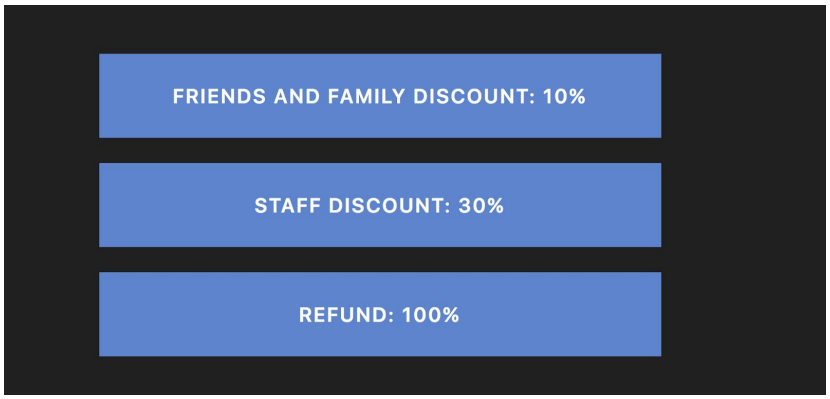
To make further adjustments or remove an item completely from the order all the staff member has to do is press one of the chosen icons in the table and once finalised complete the order.

To finalise or edit a current order all the user has to do is return to the main panel and the select the “edit order” button.

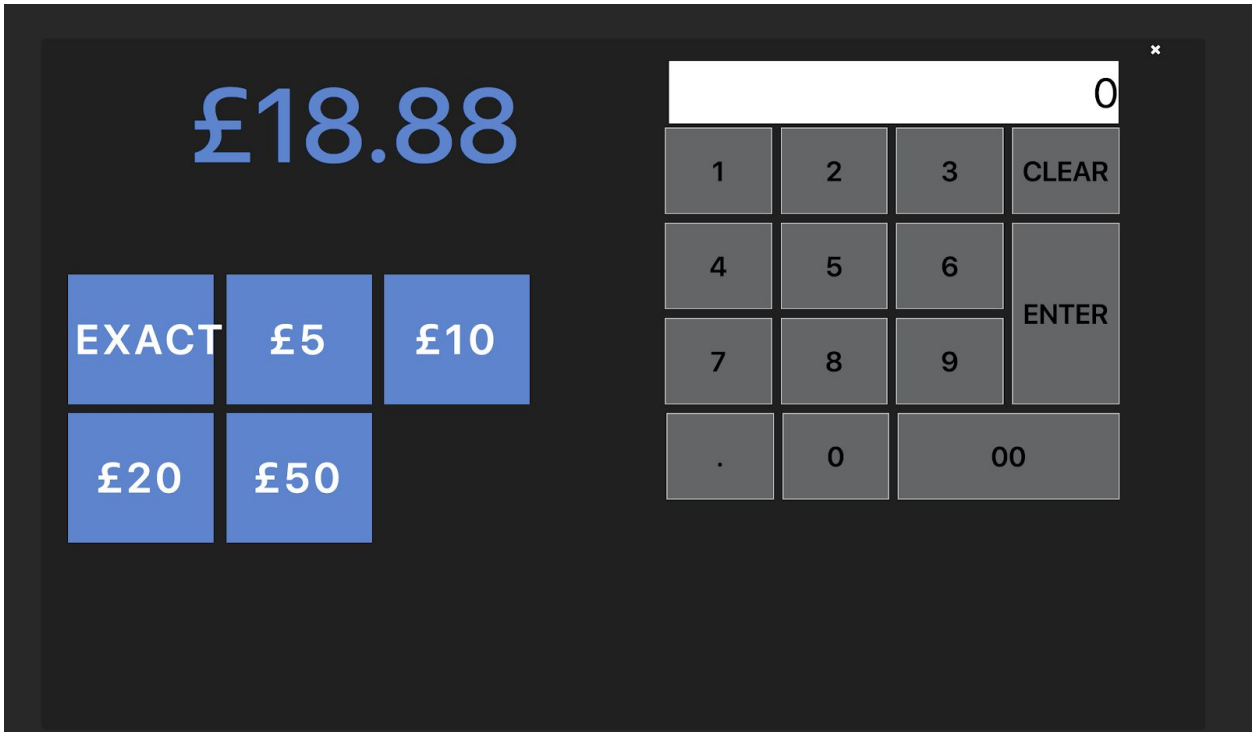


If the order is still needing to be accepted or in its prepping stage then the waiter can make changes to the notes, remove items and still add items to the menu from the menu on the left hand side.

To finalise an order the staff member has the ability to add a discount before hand to the overall order.



Here the staff member choose from a variety of discounts before making the final payment.

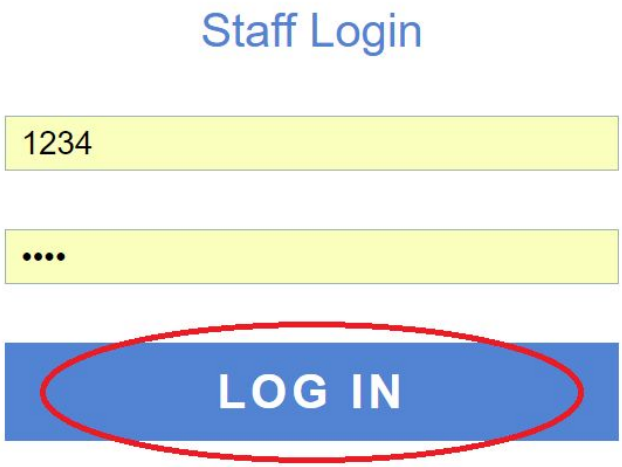


To complete the order the user simply must choose either a cash or card payment and if the customer is choosing cash a hand calculator is implemented that will show how much change the staff member must give the customer on transaction completion.

Kitchen

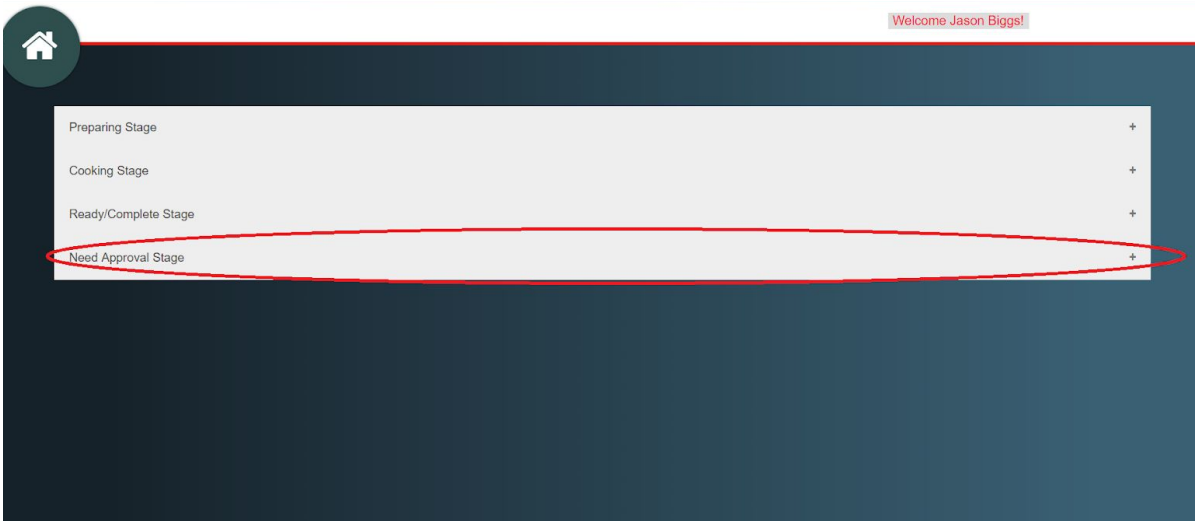
1. To get started we'll need to login!

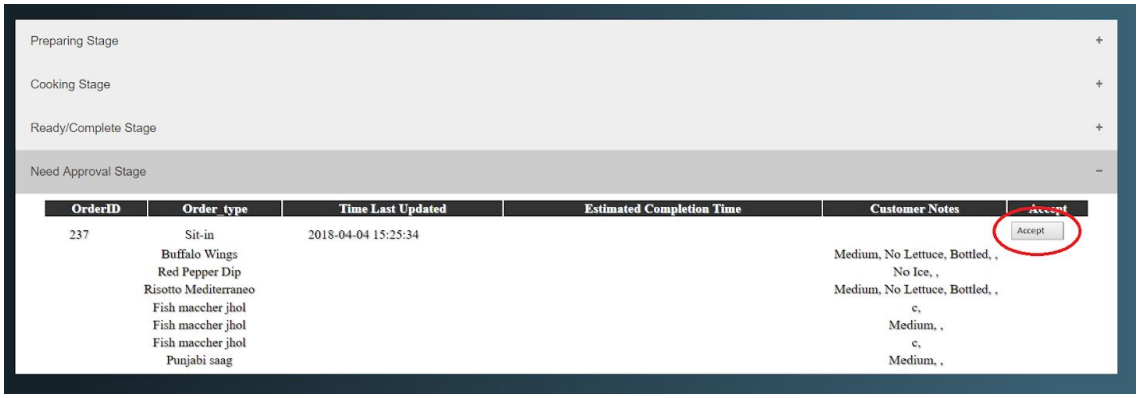
Simply input the username and password you were given and login to the kitchen side, if you has already logged into the waiter side, you won't need to login as the session should carry over.



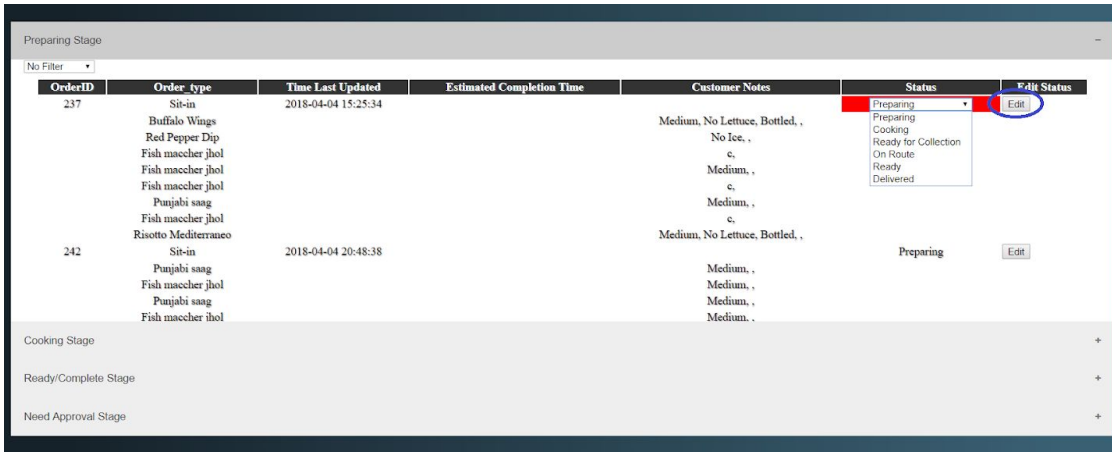
2. Now that we've logged in it's time to see if there are any orders pending for us.

Expand the needs approval divider to see if there are any unapproved orders





3. You should now see a screen not dissimilar to what is pictured above, from here you can click the accept button to accept the order, which will move it into the preparation section which is similar but with a different button.



4. The above picture is the preparation section which has had the edit button clicked which changes the regular status section into a drop down selection box for all other statuses. By selecting from this dropdown box you can change the status of the order in the database to reflect what's happening in the kitchen.

Appendix

Links - 1

1.1- https://en.wikipedia.org/wiki/Three-click_rule/
1.2- http://code.iamkate.com/javascript/using-cookies/
1.3- https://www.just-eat.co.uk/
1.4- https://hungryhouse.co.uk/?utm_source=afuk&utm_medium=affiliate&utm_campaign=aff_qmee-ltd
1.5- https://webflow.com/blog/10-essential-ui-design-tips

Tables - 2

2.1-

: Logging In With Different Job Titles

Test Case	Browser	Type of Test	User	Job Title	Expected Output	Actual Output	Notes
1	Chrome	Standard Input	Daniel Barker	Owner	Success	Success	PASS
2	FireFox	Standard Input	Daniel Barker	Owner	Success	Success	PASS
3	Edge	Standard Input	Daniel Barker	Owner	Success	Success	PASS
4	Chrome	Standard Input	Jason Biggs	Manager	Success	Success	PASS
5	FireFox	Standard Input	Jason Biggs	Manager	Success	Success	PASS
6	Edge	Standard Input	Jason Biggs	Manager	Success	Success	PASS
7	Chrome	Standard Input	Joe Bloggs	Waiter	Success	Success	PASS
8	FireFox	Standard Input	Joe Bloggs	Waiter	Success	Success	PASS
9	Edge	Standard Input	Joe Bloggs	Waiter	Success	Success	PASS
10	Chrome	Standard Input	Paul Potts	Kitchen Staff	Fail	Fail	PASS
11	FireFox	Standard Input	Paul Potts	Kitchen Staff	Fail	Fail	PASS
12	Edge	Standard Input	Paul Potts	Kitchen Staff	Fail	Fail	PASS

2.2-

Group #2: Accessing Waiter System Web Pages Without Logging In

Test Case	Browser	Page Trying to Access	Expected Output	Actual Output	Notes
1	Chrome	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/welcome.php	Redirect to Login	Redirect to Login	PASS
2	FireFox	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/welcome.php	Redirect to Login	Redirect to Login	PASS
3	Edge	http://www2.ma...m/newMenu.php bonk/staff/WaiterSystem/welcome.php	Redirect to Login	Redirect to Login	PASS
4	Chrome	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/newMenu.php	Redirect to Login	Redirect to Login	PASS
5	FireFox	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/newMenu.php	Redirect to Login	Redirect to Login	PASS
6	Edge	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/newMenu.php	Redirect to Login	Redirect to Login	PASS
7	Chrome	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/editOrders.php	Redirect to Login	Redirect to Login	PASS
8	FireFox	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/editOrders.php	Redirect to Login	Redirect to Login	PASS
9	Edge	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/editOrders.php	Redirect to Login	Redirect to Login	PASS
10	Chrome	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/temp.php	Redirect to Login	Redirect to Login	PASS
11	FireFox	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/temp.php	Redirect to Login	Redirect to Login	PASS
12	Edge	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/temp.php	Redirect to Login	Redirect to Login	PASS
13	Chrome	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/availableTablestest.php	Redirect to Login	Redirect to Login	PASS
14	FireFox	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/availableTablestest.php	Redirect to Login	Redirect to Login	PASS
15	Edge	http://www2.macs.hw.ac.uk/~ajg2/snowbonk/staff/WaiterSystem/availableTablestest.php	Redirect to Login	Redirect to Login	PASS

2.3-

Group #3: Applying Discounts to Table1s Order

	Test Case	Browser	Discou	Beginning Order Value	Expected End Order Value	Actual End Order Value	Notes
	1	Chrome	10%	£187.95	£169.16	£169.16	PASS
	2	FireFox	10%	£187.95	£169.16	£169.16	PASS
	3	Edge	10%	£187.95	£169.16	£169.16	PASS
	4	Chrome	30%	£187.95	£131.57	£131.57	PASS
	5	FireFox	30%	£187.95	£131.57	£131.57	PASS
	6	Edge	30%	£187.95	£131.57	£131.57	PASS
	7	Chrome	100%	£187.95	£0.00	£0.00	PASS
	8	FireFox	100%	£187.95	£0.00	£0.00	PASS
	9	Edge	100%	£187.95	£0.00	£0.00	PASS

2.4-

Group #4: Applying Multiple Discounts to Table1s Order

	Test Case	Browser	Discou	Beginning Order Value	Expected End Order Value	Actual End Order Value	Expected Output	Actual Ouput	Notes
	1	Chrome	10%	£187.95	£169.16	£169.16	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	2	FireFox	10%	£187.95	£169.16	£169.16	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	3	Edge	10%	£187.95	£169.16	£169.16	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	4	Chrome	30%	£187.95	£131.57	£131.57	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	5	FireFox	30%	£187.95	£131.57	£131.57	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	6	Edge	30%	£187.95	£131.57	£131.57	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	7	Chrome	100%	£187.95	£0.00	£0.00	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	8	FireFox	100%	£187.95	£0.00	£0.00	Discount Doesn't Stack	Discount Doesn't Stack	PASS
	9	Edge	100%	£187.95	£0.00	£0.00	Discount Doesn't Stack	Discount Doesn't Stack	PASS

2.5-

Group #5: Splitting Table2s Order

	Test Case	Browser	Items t		Expected End Order Value	Actual End Order Value	Notes
	1	Chrome	Soup of	£11.28	£6.29	£6.29	PASS
	2	FireFox	Soup of	£11.28	£6.29	£6.29	PASS
	3	Edge	Soup of	£11.28	£6.29	£6.29	PASS

2.6-

Group #6:Adding Discount Then Splitting Table2s Order

	Test Case	Browser	Items to Split	Discount	Beginning Order Value	Expected End Order Value	Actual End Order Value	Notes
	1	Chrome	Soup of the Da	10%	£11.28	£5.66	£5.66	PASS
	2	FireFox	Soup of the Da	10%	£11.28	£5.66	£5.66	PASS
	3	Edge	Soup of the Da	10%	£11.28	£5.66	£5.66	PASS
	4	Chrome	Soup of the Da	30%	£11.28	£4.40	£4.4403000	Didn't fix to 2 decimal place
	5	FireFox	Soup of the Da	30%	£11.28	£4.40	£4.4403000	Didn't fix to 2 decimal place
	6	Edge	Soup of the Da	30%	£11.28	£4.40	£4.4403000	Didn't fix to 2 decimal place
	7	Chrome	Soup of the Da	100%	£11.28	£0.00	£0.00	PASS
	8	FireFox	Soup of the Da	100%	£11.28	£0.00	£0.00	PASS
	9	Edge	Soup of the Da	100%	£11.28	£0.00	£0.00	PASS

2.7-

Group #7:Transaction With Split Items Table 2

	Test Case	Browser	Items to Split	Beginning Order Value	Expected Output	Actual Output	Notes
	1	Chrome	Soup of the Day, Irn-Bru	£11.28	Split Items Removed From Table, OrderTotal = £4.99	Split Items Removed From Table, OrderTotal = £4.99	PASS
	2	FireFox	Soup of the Day, Irn-Bru	£11.28	Split Items Removed From Table, OrderTotal = £4.99	Split Items Removed From Table, OrderTotal = £4.99	PASS
	3	Edge	Soup of the Day, Irn-Bru	£11.28	Split Items Removed From Table, OrderTotal = £4.99	Split Items Removed From Table, OrderTotal = £4.99	PASS

2.8-

Group #8:Complete Transaction Table 2

	Test Case	Browser	Beginning Order Value	Expected Output	Actual Output	Notes
	1	Chrome	£11.28	Order Removed from database, Table made Availiable	Order Removed from database, Table made Availiable	PASS
	2	FireFox	£11.28	Order Removed from database, Table made Availiable	Order Removed from database, Table made Availiable	PASS
	3	Edge	£11.28	Order Removed from database, Table made Availiable	Order Removed from database, Table made Availiable	PASS

2.9-

Group #9:Use Predefined Cash Buttons On Table 1 Order

Test Case	Browser	Button Used	Order Value	Expected Output	Actual Output	Notes
1	Chrome	EXACT	£192.94	Change = £0.00	Change = £0.00	PASS
2	FireFox	EXACT	£192.94	Change = £0.00	Change = £0.00	PASS
3	Edge	EXACT	£192.94	Change = £0.00	Change = £0.00	PASS
4	Chrome	£5	£192.94	Total = £187.94	Total = £187.94	PASS
5	FireFox	£5	£192.94	Total = £187.94	Total = £187.94	PASS
6	Edge	£5	£192.94	Total = £187.94	Total = £187.94	PASS
7	Chrome	£10	£192.94	Total = £182.94	Total = £182.94	PASS
8	FireFox	£10	£192.94	Total = £182.94	Total = £182.94	PASS
9	Edge	£10	£192.94	Total = £182.94	Total = £182.94	PASS
10	Chrome	£20	£192.94	Total = £172.94	Total = £172.94	PASS
11	FireFox	£20	£192.94	Total = £172.94	Total = £172.94	PASS
12	Edge	£20	£192.94	Total = £172.94	Total = £172.94	PASS
13	Chrome	£50	£192.94	Total = £137.94	Total = £137.94	PASS
14	FireFox	£50	£192.94	Total = £137.94	Total = £137.94	PASS
15	Edge	£50	£192.94	Total = £137.94	Total = £137.94	PASS

2.10-

Group #10:Use Keypad On Table 1 Order

Test Case	Browser	Order Value	Keypad Input	Expected Output	Actual Output	Notes
1	Chrome	£192.94	100	Total = £92.94	Total = £92.94	PASS
2	FireFox	£192.94	100	Total = £92.94	Total = £92.94	PASS
3	Edge	£192.94	100	Total = £92.94	Total = £92.94	PASS

2.11-

Group #11:Add Item to Table1 Order Using Various Notes

Test Case	Browser	Item	Notes	Expected Output	Actual Output	Notes
1	Chrome	32oz Steak	Rare	Success	Success	PASS
2	FireFox	32oz Steak	Rare	Success	Success	PASS
3	Edge	32oz Steak	Rare	Success	Success	PASS
4	Chrome	Irn-Bru	Can	Success	Success	PASS
5	FireFox	Irn-Bru	Can	Success	Success	PASS
6	Edge	Irn-Bru	Can	Success	Success	PASS
7	Chrome	Sticky Toffee F	Random Notes	Success	Success	PASS
8	FireFox	Sticky Toffee F	Random Notes	Success	Success	PASS
9	Edge	Sticky Toffee F	Random Notes	Success	Success	PASS

2.12-

Group #12:Edit Notes of Items

Test Case	Browser	Item	Notes	New Notes	Expiry Time	Current Time	Expected Output	Actual Output	Notes
1	Chrome	32oz Steak	Rare	Test	19.53.12	19.48.43	Success	Success	PASS
2	FireFox	32oz Steak	Rare	Test	19.53.12	19.48.43	Success	Success	PASS
3	Edge	32oz Steak	Rare	Test	19.53.12	19.48.43	Success	Success	PASS
4	Chrome	Irn-Bru	Can	Test	19.53.12	20:12:24	Fail	Fail	PASS
5	FireFox	Irn-Bru	Can	Test	19.53.12	20:12:24	Fail	Fail	PASS
6	Edge	Irn-Bru	Can	Test	19.53.12	20:12:24	Fail	Fail	PASS
7	Chrome	Sticky Toffee Pudding	Random Notes	Test	19.53.12	20:12:24	Fail	Fail	PASS
8	FireFox	Sticky Toffee Pudding	Random Notes	Test	19.53.12	20:12:24	Fail	Fail	PASS
9	Edge	Sticky Toffee Pudding	Random Notes	Test	19.53.12	20:12:24	Fail	Fail	PASS

2.13-

Group #13:Removing Items From Orders

Test Case	Browser	Item	Expiry Time	Current Time	Expected Output	Actual Output	Notes
1	Chrome	32oz Steak	19.53.12	19.48.43	Success	Success	PASS
2	FireFox	32oz Steak	19.53.12	19.48.43	Success	Success	PASS
3	Edge	32oz Steak	19.53.12	19.48.43	Success	Success	PASS
4	Chrome	Irn-Bru	19.53.12	20:12:24	Fail	Fail	PASS
5	FireFox	Irn-Bru	19.53.12	20:12:24	Fail	Fail	PASS
6	Edge	Irn-Bru	19.53.12	20:12:24	Fail	Fail	PASS
7	Chrome	Sticky Toffee Pudding	19.53.12	20:12:24	Fail	Fail	PASS
8	FireFox	Sticky Toffee Pudding	19.53.12	20:12:24	Fail	Fail	PASS
9	Edge	Sticky Toffee Pudding	19.53.12	20:12:24	Fail	Fail	PASS

2.14-

Group #14:Getting Waiter Notifications

Test Case	Browser	No of Requests	Expected Output	Actual Output	Notes
1	Chrome	2	Requests Shown: 2	Infinite Loop of Requests	FAIL---Requests stuck in infinite loop
2	FireFox	2	Requests Shown: 2	Infinite Loop of Requests	FAIL---Requests stuck in infinite loop
3	Edge	2	Requests Shown: 2	Infinite Loop of Requests	FAIL---Requests stuck in infinite loop

Marketing Report
By
Team SnowBonk

CONTENTS

Executive Summary

Summary

Marketing is a vital component of any software development process. Usually occurring during the latter stages of any new project, it is used to garner the attention of the target demographic, through doing this they help to grow their brand and increase sales. The aim of this marketing analysis is to comprehensively gain an understanding of the prevailing market conditions in which DineBros is about to enter and outline the main marketing objectives of this campaign. This will be done by conducting a number of research methods such as SWOT, PESTEL and the marketing mix. These can then be used to help achieve these marketing targets.

Background

The DineBros system has been created by a group of developers from Heriot Watt university in Edinburgh, called SnowBonk. The team was made up of students from a range of computing disciplines and skill sets. The system has been developed over the course of 2 university semesters and been designed to assist restaurants to manage their food orders and allow customers to take control of their reservations and food orders. In theory, helping to streamline the ordering process and make it more efficient whilst making it more personal to the consumer.

Market Analysis

The market analysis can help a company understand where it stands in relation to the rest of the marketplace. There are already an abundance of food ordering systems in the market at present, this analysis will help establish what the system does similarly to other competitors and help to highlight what it does that others don't. The main advantage of this being that it should help set out a number of objectives that should be achieved by the end of the process.

Location

The company decided to promote and market the app in the most populous cities in the UK as that is where most restaurants will be located. As the development team are based in Edinburgh (pop 507,170) a special focus will initially take place in this area due to increased local knowledge and connections, as well as bustling food scene in the city, home to some of the uk's best restaurants. Edinburgh itself is a city with a rich dining heritage and the Leith area is renowned for its culinary options. Small independent restaurants mingle effortlessly with the bigger chains. Once established in Edinburgh there will be a view to expand the system across the UK with marketing capitalising on the various cultural food traditions across the country.

Target Market

In order to get a complete overview of the sector in which a new product is being entered into the market, it is important to find out where and what the current marketplace is. The food app business is very competitive. Most new delivery players allow consumers to compare offerings and order meals from a group of restaurants through a single website or app, it will be the responsibility of the marketing team to highlight the systems strengths and to showcase why DineBros will be the new revolution in dining.

The restaurant ordering market in the UK is steadily riding as students, young professionals and families are wanting a more refined dining experience. From the usability study carried out in phase 1 of the project we found out that 77% of people were most likely to use their mobile device to order food. In order to capitalise on this, this therefore will be the platform focus going forward.

Customers

From a 2018 survey carried out by research company Statista (Appendix 1) it is evident 18-29 year olds are most likely to use food delivery apps, with 58% of respondents claiming that they use platforms such as deliveroo and JustEat. The older the age of respondent the more likely it would be that they would not use any kind of online food delivery service, with 46% of respondents having not used such a service previously. There is a possibility there may perhaps be resistance to change from some of those older customers. Therefore the software must be intuitive and easy to use in order to win over this sector of the market.

The 16-34 age group will be the one which we will mostly target as we feel that this will be the group will the greatest possibility of revenue generation. This is not to say that the marketing strategy will

completely ignore the older demographic of customer especially due to the rise of the so called ‘silver surfers’, those older people who are active internet users.

Restaurants

The system will be targeted mainly at small chain or independent restaurants. As explored further in the report, many of the larger restaurant chains such as McDonald’s and Weatherspoon’s have their own proprietary ordering systems and are unlikely to be interested in an off the shelf solution such as this.

Competitive Analysis

Summary

The market in which Dine-bros is operating is one which is still emerging as restaurants and dining places continue to balance the need to invest in new technology while using the financial resources available to them. At present, Dine Bros is in a unique position to capitalise in this market as it provides an all in one technology solution for customers and restaurants alike.

By far, the most popular food ordering system is Just Eat with 85-95% market share in the UK. Its’ reach and dominance considerably increased with the acquisition of competitor Hungry House for £200 million in 2016.

There are a number of other smaller, but similar food ordering app’s such as deliveroo and Uber Eats. Bar chain Weatherspoons have a table service food ordering app for pub’s and restaurants however the scope for their system is restricted to their own business. In terms of restaurant reservation booking, the market leader is OpenTable.

Other restaurants will have their own separate waiter and kitchen system of indeed still be using paper based note taking. These may be restaurant who may not see the need or benefit of such a system

A Comparison of Competitor Food Ordering Systems

Weatherspoons

The weatherspoons food and drink app is largely useful for ordering food and drink to your table. A customer selects the pub they are at on a map, then thay select their table number and finally choose their option of food and drink, paying for this on the app means that they don’t have to leave their seat and can avoid standing at a busy queue at the bar. One main disadvantage of this is that there is always the chance that bar staff will focus on serving customers at the bar, leaving customers on the App waiting longer for their orders.

McDonalds

McDonalds, the world's largest fast food chain, has in recent years, started to introduce self service ordering systems for its customers. This is done through large touch screens placed near the entrance of the restaurant. Once the user pays for their order, they get a order number which they then use to collect their food when ready.

Touch Bistro

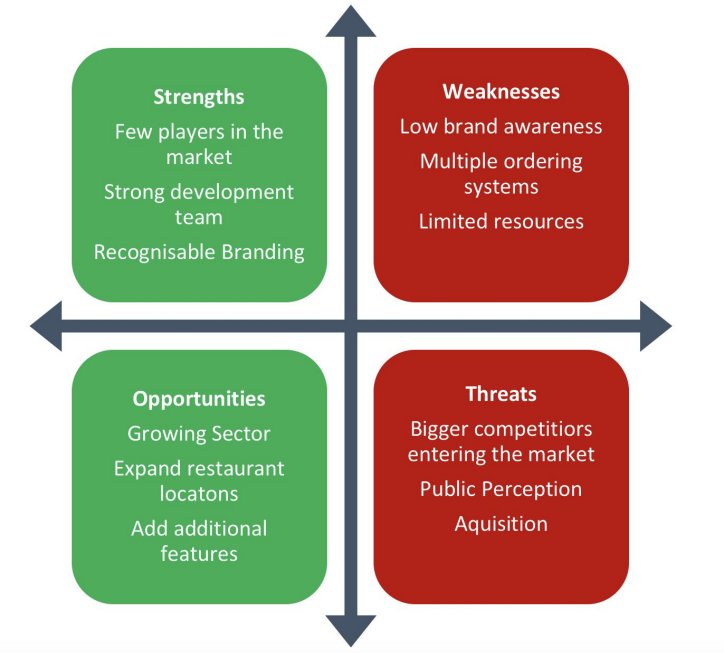
There is a wider market of off the shelf pieces of software for handling restaurant orders. One of the most popular systems is Touch Bistro, it is the number 1 Grossing food & drink app in over 37 countries. It's main advantage is the flexibility with which it's tablet based system allows staff take orders and take payments at the diners table. It's easy to use interface mean it can be used by waiting staff though to the kitchen and back office.

SWOT Analysis

Purpose

By performing a SWOT analysis a comprehensive analysis can take place of both the positives and negatives of launching a new product into market, This is done in the form is analysing the strengths, weaknesses, opportunities and threats to the business. This can help focus the business on what it is doing right and what it can potentially improve upon.

SWOT Diagram



This diagram explaining the strengths, weaknesses, opportunities and threats is explained in detail below

Strengths

As the food delivery market is still in its emergent stages, there are still relatively few players in the market, especially for a hybrid in-restaurant and takeaway dining experience.
The system has been developed over the course of two semesters by a team with over 20 years experience between them. The concept is strong and should be easily sellable to restaurateurs as it is an all in one system which directly links orders to the restaurants kitchen. The branding is quite recognisable, with a strong tagline in 'redefining dining'.

Weaknesses

DineBros will be a new entrant to the market, this therefore may mean that it will be difficult to achieve penetration. For businesses, it may take much persuading in order to get them to use our system. On the customer side, they won't want to use the system if the restaurants they wish to order from aren't on the system. As snowbonk are a start-up company, one of the main weaknesses will be a lack of brand awareness coupled with the company having little reputation or client base from which to showcase our previous works.
Many restaurants who work with other food orders services like Just Eat and uber Eats may be reluctant to change provider or add another platform to their work environment as this could prove bothersome to both customers and restaurant staff alike.
Snowbonk is made up of quite a small team of developers so it may take a little while longer to fix any bugs or errors in the system, this could throw up doubts about the reliability of the system

Opportunities

There are great opportunities for businesses to make efficiency gains, saving money while attracting more customers as they will no longer need to pay for two separate systems. Once the application has established itself in the initial roll-out cities there are multiple chances for expansion in town's and cities all over Scotland and the UK. This has a snowball effect as the more restaurants use the system, the more consumers who will too.
As the company enters into its initial launch, the work on the system will not stop there. In response to feedback from customers and restaurant owners, snowbank will make functionality improvements to the system making it easier to use and adding features, this for instance could mean adding a gps feature in time.

Threats

While this a unique concept, it is not to say that many established food ordering companies may adjust their business model to accommodate systems similar to DineBros. This means the company should always be on its toes, improving the system, ensuring that it is the best in its field.
If the new company is successful there is the chance that a bigger company could come in and acquire the company or absorb it into their own.
The company will be starting off with a user base of Zero, if the company does not quickly establish itself in the market place, it could become obsolete or unsustainable in a short period of time.

Customers may prefer the traditional dining experience rather than using technology to order. Some may see it as more impersonal. Older people for instance may not have the resources or capability to use the system, perhaps leaving them feeling alienated or isolated.

Marketing Mix

Purpose

The marketing mix refers to the set of actions, or tactics, that a company uses to promote its brand or product in the market. The 4Ps make up a typical marketing mix - Price, Product, Promotion and Place.

Product

The app ordering system will mainly consist of restaurants, offering a more upmarket experience to consumers compared to other restaurant rival who focus on traditional fast food outlets.

Price

Restaurant who use the overall system will pay a license fee to use the system, this will include their placement on the ordering website/app. For customers ordering food, there will be no charge to use the system,

Place

DineBros is an food ordering/restaurant management system, with applications available on ios/android. The app will primarily feature restaurants in the Midlothian region initially, but it hoped that it will extend across the UK and potentially even further with content available in a number of languages

Promotion

Promotion would be aimed at those within the industry as this is who the system is aimed it. Conversely, the customers who use the app will also be targeted.

Revenue Model

Summary

Unlike other food ordering apps DineBros will not charge customers a service charge for ordering food on the system, instead restaurants will pay a license fee to run the food ordering system in their workplace. In return for their license fee the restaurant will will be hosted for free on the website and app. Pricing will be dependent on the size of the restaurant or the number of restaurants in operation

PESTLE Analysis

Purpose

A PESTEL analysis is a framework or tool used by marketers to analyse and monitor the macro-environmental (external marketing environment) factors that have an impact on an organisation. The result of which is used to identify threats and weaknesses which is used in a SWOT analysis. The acronym stands for Political, Economical, Socio-cultural, Technological, Legal and Ecological.

Political

BREXIT is currently something which could possibly cause instability as the UK's future relationship with Europe is unclear at present. This means that there may be no guarantee over such things such as data protection laws such as the GDPR due to be introduced in May 2018. A new sugary drinks tax in being introduced in the UK as well as minimum pricing on alcoholic drinks in Scotland. Any associated price increases may hit consumers in the wallet meaning that they are less likely to dine out, along with the possibility down the line that further taxes could be introduced to impact unhealthy foods.

Economical

There are a number of macro and micro economic factors which could affect DineBros entry into the marketplace. The UK economy is currently undergoing a dip and especially the restaurant sector which in the past has expanded in a unsustainable rate, This has recently resulted in well-known chains such as Prezzo and Jamies Italian having to restructure and close restaurants. According to PWC, household income is expected to be lower in 2018. As an SME they may be some government grants available to help the business promote themselves.

Socio-Cultural

Most people in the UK can now access some form or other of app based content on their mobiles/tablets through wi-fi or 3/4g. This gives us a great opportunity to capitalize on this and reach as wide an audience as possible. __ In 2017, 90% of households in Great Britain had internet access, leaving the company with a wide range of potential customers. People in the UK dine out on average x more often than they previously did

People are also living busier lives as they now work longer hours (source)

Technological

App based food ordering systems have gained a foothold in the uk market and consumers are comfortable ordering food this way meaning that the system should be familiar to users. Technology will enable us to spot where the greatest usage of the app is and see where potential expansion opportunities lie. Targeted adverts will allow clients to advertise to the customers they want.

Legal

The greatest legal concern will be that surrounding data protection. As the app will handle large amounts of personal data it is imperative that correct measures are in place to account for this. In light of the new EU General Data Protection Regulations (GDPR), this is especially important as harsh financial penalties could occur if any of these are broke.

Ecological

As software based company there should be little paper based wastage. There is an onus on companies to aim towards being carbon neutral. As the system handles orders through a web/mobile/tablet interface there should be little need for paper based systems.
Many companies operate an carbon offset scheme where they will plant trees to account for their companies fossil fuel usage. Snowbonk themselves as a company should aim to promote ecologically friendly schemes such as recycling, ride sharing and promoting cycle to work schemes

Unique Selling Point

When developing the DineBros application, SnowBonk development wished to find ways to differentiate the system from other existing players in the marketplace. While most existing players work on a basis on ordering food for delivery or collection in-store there was no multi restaurant equivalent for booking places for dining at a restaurant which was integrated with such a system.

This all in one solution will have its own unique benefits for both consumers and restaurateurs. The product will be marketed at both a business and customers, both in slightly different ways. This two pronged approach will aim to highlight the benefits of the system specific to each stakeholders needs.

Marketing Strategy

The tagline for the system is 'Redefining Dining'. This simply explains to the customer what the system is and brings in some of the essence of who the product caters to, those who want to move away from the traditional dining experience of long queues and waiting to be seated. For the marketing strategy of the system a large social media element will be included. This will include the biggest social platforms such as Facebook, Instagram, Twitter and Snapchat. These will be used to firstly increase brand awareness, especially to our target audience who are most likely to use App's and websites such as these. Once this has been established these accounts will showcase the various restaurants and foods available through the app.

The marketing will try to highlight that companies using the platform offer a more sophisticated experience than would perhaps be found on other food ordering apps such as Just Eat and Scoffable.

Whilst the booking system will be available as both an App and web experience, users will be encouraged to download the app as for long term and repeated use it is better to have a permanent presence on their phone allowing them to call up the system as they please. At the product Expo this was demonstrated using various marketing materials (Appendix 4). To demonstrate this the group purchased a snapchat filter(Appendix 7). This covered the area of the Earl Mountbatten Building and the university canteen. This was purchased for 3 hours, costing seven pounds and proved to be a highly efficient way of targeting specific individuals or areas.

If people elect to use the fun filter it also put the brand front and centre to the snapchat user who receives a message.

Date	Swipes	Uses	Views	Reach	Use Rate	Swipe Time
2018-03-29	99	23	575	402	23.2%	11.2

This analysis of the snapchat filter shows that 575 people viewed the filter. This works out at 1.71p per person, this showcases how cost effective this marketing method is. Extrapolating this method of advertising to specific areas and town/city centres could prove as equally fruitful.

As the company is not well established, guerrilla advertising may perhaps be another method of encouraging new users. Also known as ambush marketing, attack marketing is a form of marketing that incorporates a series of creative and strategic techniques used to build and maintain public awareness surrounding a person, place, product, or event. At the expo, various 'snapcodes' were posted on the wall and distributed throughout the room (Appendix 3). These invited users to follow the company on snapchat, allowing them to find out more about the company. This complemented the use of the snapchat filter as part of the marketing showcase.

Stakeholders

Summary

By launching the DIneBros system onto the market, this will affect a number of stakeholders on the project, who will need engagement. The success of the product will depend on the successful and productive relationship with a number of these.

SnowBonk – As the developers of the system snowbank will obviously have a part to play in the systems development

Customers – Those who will use the system to order food and spread the word about the company and product to their friends and family,

Restaurants - The restaurants who will use the system should be able to provide feedback on what they want from the system and possibly anything which has been overlooked in the design of the system.

Ledios – This was the company who set the spec of how the system should look. As a company they will want to see that the system spec has been closely followed

(City of Edinburgh Council) – Product success could be dependent upon the availability of restaurants to invest in a new system, this ability could be proportional to council tax and rental costs.

Heriot Watt University – The university were involved in the initial part of the project setting up the software team and setting the project for the client

Competitors – Competitors will be closely watching the success and or failure of the system along with monitoring aspects of it which they themselves may wish to integrate into their platforms

Branding

Red is one of the main primary colours used in the design of the system and is included in the colour of the logo of DineBros. This helps to create and reinforce a strong, recognisable and confident brand which can sit at ease amongst the main players in the market.
The DineBros logo was created first and foremost to establish the name of the system in people's minds, hence it features prominently. The red two horizontal lines above and below this help to create the image of a burger, the Dinebros being the meat in the middle, reflecting on the systems food background (Appendix 7).

Bibliography

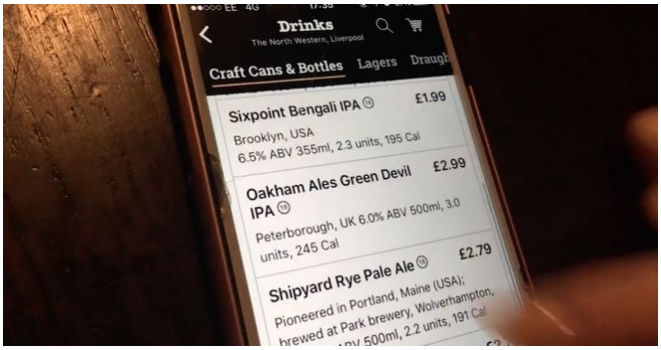
Academy, P. (2018). *Marketing Theories - PESTEL Analysis*. [online] Professionalacademy.com. Available at: https://www.professionalacademy.com/blogs-and-advice/marketing-theories---pestel-analysis [Accessed 5 Apr. 2018].

PwC. (2018). *Restaurants 2017: Food for Thought*. [online] Available at: https://www.pwc.co.uk/services/business-recovery/insights/restructuring-trends/restaurants-2017-food-for-thought.html [Accessed 5 Apr. 2018].

Appendix

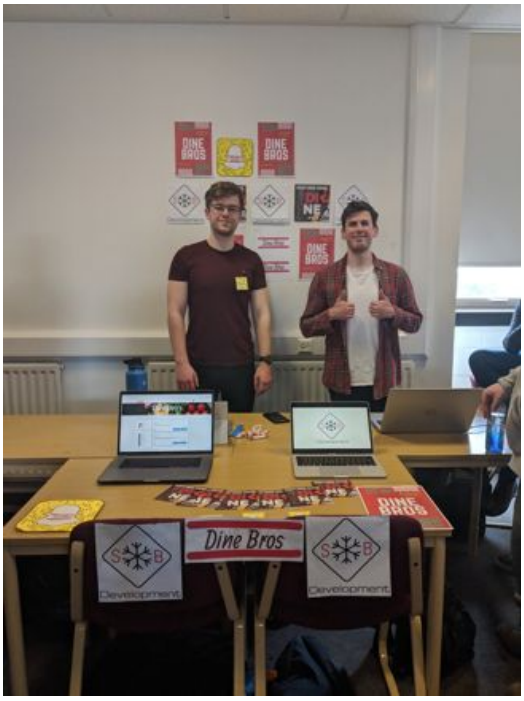
	18-29 years	30-59 years	60 years and older
Food delivery app	58%	31%	4%
Call restaurant	42%	52%	37%
Order online at the restaurants website	40%	31%	12%
Use online food delivery service site	55%	33%	12%
Don't use these kinds of services	5%	12%	46%
Other	1%	1%	0

Appendix 1 Market research detailing differing age group habits for ordering food

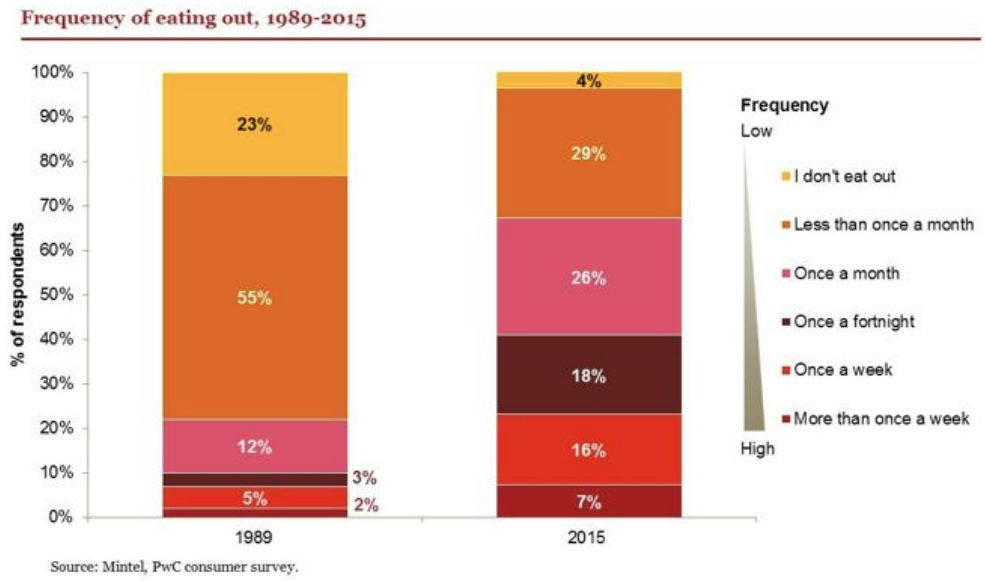


Appendix 2 The Weatherspoons food & drink ordering app

Appendix 3 An example of guerilla advertising showcased at the Expo



Appendix 4 Expo Day



Appendix 5 A table showing the UK's changing dining habits since 1989 .

Accessed from:https://www.pwc.co.uk/services/business-recovery/insights/restructuring-trends/restaurants-2017-food-for-thought.html

Appendix 6 Marketing materials for the DineBros system

Appendix 7 Snapchat Filter purchased for Expo

Project Evaluation

by

Team SnowBonk

Version 1.1

March 29, 2018

Table of Contents

Document Introduction

Purpose

This Document aims to give an overview and reflection on how the group (SnowBonk) approached and implemented the final product that was to be delivered at the expo dated 29/03/2018. The document will highlight all areas of the group project including;

- Organization of the Group
- Implementation of the Project
- Final Product Evaluation
- Conclusions and review of the Project

This document then will showcase the strengths and weaknesses of the group, however, will not highlight any issues with individual group members which is to be assessed in a separate "Peer Review" document.

The Document itself is intended to be read by;

- Our Group Manager Chris Fensch and will be assessing this document.
- Rob Petrick who is leading the Project and will be assessing this document.

Overview

During the creation of this document all group members have had input and all were present at the writing of this document to make the reflection accurate and honest for the readers and invigilators of the project.

The Document will be split into the following sections;

- Organization of the Group
- Implementation of the Project
- Final Product Evaluation
- Conclusions and review of the Project
- Appendix

Organization of the Group

This section will discuss factors that include how the group was organised and if the approach chosen was beneficial to SnowBonk or if it hindered our production and efficiency. The section will also discuss how SnowBonk would collaborate throughout the project and if this accounted for any problems we may have had during the project. Finally the section will contrast the realistic timings of the group during the project opposed to the original plan detailed in Stage One.

Implementation of the Project

This section will discuss factors that include how the group designed and implemented the Application from Mock-Ups detailed in the Stage One Document to how they transferred into the final prototype. This will showcase our implementation schedule of when we would meet to complete areas of

functionality and design by showcasing our Software Engineering approach. We will then discuss what was successful and what was not using examples as reference. The section will also showcase some of the tools and technology we encompassed to create the project.

Final Product Evaluation

This section will highlight the functionality we failed or succeeded in creating at the final implementation of the prototype application. The section will also highlight what we believe is a unique selling point of our application and any extra features we have included upon completion. We will also discuss the robustness of the system and highlight any known issues with the product and give a comparison as to how usable the application truly is.

Conclusions and Review of the Project

This section will aim to showcase how we felt about the project as a whole and the final application we have provided as a solution to the specification that was asked of us at the beginning of the project and any contrasts that may appear. There will also be a summary showcasing how we believe we could of improved the product having had more time, resources etc.

Appendix

This section will contain any material relevant to the current document to emphasise certain factors and to make reference to any claims that are made by the group in the creation of this document.

Notes

All group members contributed to this section of the report and every opinion regarding the product and project as a whole was discussed and appropriately issued into the report.

Organisation

Group Structure

Group Layout

For this project we aimed to have a less formal and more democratically based structure with members of the group choosing which roles they would like to work on based on their comfortability with that particular section. We felt this approach greatly improved productivity as each member felt responsible and proud of their own area they were strong in. The work produced in each area was of high quality and provided a strong initial development of the product.

Informal Management

We did have to have an informal “leader” of the project of which that role fell mostly towards Toni. This setup was mostly to keep the team organised and focused on tasks but the group was still entirely democratic. Following on from this we also organized with ourselves that Daniel would be the liaison coordinator for the group and would organize meetings with our group manager, Chris Fensch. Having a sole corresponder kept the communication between our manager consistent and professional.

Responsibility of the Group

The team was divided to work on different aspects of the project where each member would be in charge and have total control of this area. This was a loose structuring of the group which has since differed to that of the stage one assignment of roles. Some members would hotswap between departments however the rough division by stage 3 was as follows;

- Database
  - Calum and Daniel.
- Backend
  - Toni focused on the user side's functionality.
  - Owen focused on the kitchen side's functionality.
  - Craig focused on the waiter side's functionality.
- Frontend
  - Jordan focused on the CSS and design characteristics of each side of this project.
  - Phil focused on marketing of this project and the company website.

This setup original setup in stage one produced impressive results from the get go with the database team completing the production of the database well before the estimated completion time. This freed up members to work on different departments spreading out workload. This is why, compared to stage one, Craig has been placed on the backend functionality instead of database design to give extra focus on the development of this area. Furthermore, as informal leader roles switched from Daniel being the project lead to Toni being the project lead, Daniel switched to the database department.

Group collaboration

Group collaboration is an important factor in the success of any project and being able to be access work or members of the team anytime is crucial. For our group, we used a variety of tools and techniques to aid in the development of this project.

Collaboration Tools

- **Communication**
  We established our main method of communication as Facebook messenger. Since every one of our members had a smartphone and a Facebook account this was an obvious choice to setup a central hub of communication. The rich features of Facebook messenger allowed us to share valuable data such as files, links and meeting times while also enabling us to hastily retrieve a team member when a problem arose. This allowed reaching any member at any point instantly which helped in the flow of communication our team proved to had.
- **Documentation**
  Google Drive and Google Docs were essential tools for developing the documentation of this project. This allowed us to collaborate in real time on the same documents and accessing all our files in the same location. This also meant all our documentation was up-to-date and being supplied to each team member, removing the chance of someone working on old documents. We had three different folders for the three different stages of this project. This allowed us to keep ourselves organised and files relevant depending on the directory we were working on. Conclusions to our group meeting discussions were organised in their own separate directory and labelled by date for easy reference. This tool greatly aided in development of documents required.
- **Project Files**
  When developing our project we used Heriot Watt's webserver and Toni's local directory to work on the project. We enabled FTP privileges on this directory to allow all members of the team to work from this location. This kept our project's files in the same place and allowed live updating of the website rather than waiting for approval needed to execute file updates. This benefitted team collaboration greatly as all members would be working on the projects up-to-date files and removed implementation conflicts.
- **Meetings**
  At the start of our project we aimed for a minimum of one meeting a week however this was not entirely kept to. The start of development of the project was where each member focused on the initial development of their own sections and didn't require much collaboration. Closer to the stage 2 deadline is where meetings became more frequent to bring the first of the system together. At these stages we communicated informally on tasks to be completed for the next meeting, which were roughly a week apart. This helped demonstrate to other team members where each part of this project was at. Where a big concern that arose for our group was during the lecturer strikes. Group productivity overall took a turn for the worst and entire development of the project. This will be discussed in depth later in the project evaluation.

Project Timings

The actual timeline of this project compared to the original plan has been tightly matched or even completed in advance from expected completion times for the database tasks. This allowed us to have a strong initial development on this project which allowed us to focus in other areas. During the end of stage two however it took the team longer than expected to complete the tasks planned for the frontend which led to delays. This can be seen in more depth in the Implementation Schedule section of this report.

For stage 3 of this project, we firstly had to revise the iteration schedule as this stage was left quite vague from when it was planned in stage 1. The team came together to review the current state of the system and planned what the next few iterations would focus on. After creating our plan we set to work and kept to the timings of the iterations closely. As will be discussed in the Implementation Schedule we faced massive delays in iteration 12 to 14. This is discussed in depth in that section. Ultimately towards the end of stage 3 the team didn't complete everything specified in the iteration schedule and felt we could of produced more if we hadn't face such big delays. The team does feel we produced a satisfactory product for the expo however and are very proud of what we have completed.

In total, we had a very strong stage 2 and an initially strong stage 3, however we fell short of what we expected for the product towards the end of stage 3. We feel the incremental development process we had used was very beneficial to how we developed our project and kept us on track for the most part.

Implementation

Implementation Schedule

Stage 2 Revised Iterations

Below is the revised iteration schedule reflecting the changes mentioned in stage 2's document.

Stage 2 Revised Iterations 1 - 8 – Colour Key

Type of Task	Colour
Iterations	
Frontend	
Backend	
Database	
Distractions	
General	

Stage 2 Revised Iterations 1 - 8 - Timeline

Task Name	Priority	Status	Start	Finish	Who
Prototype Development Stage 2	High	Complete	27/11/17	01/29/18	Everyone
Iteration 1	Medium	Complete	27/11/17	01/12/17	Everyone
Basic User UI and Frontend Implementation	High	Complete	27/11/17	01/12/17	Jordan, Phil
Server Setup and Working	Medium	Complete	27/11/17	28/11/17	Toni, Owen
Database Structure Design	High	Delayed	27/11/17	29/11/17	Calum, Craig, Daniel
Intialise Backend	Medium	Complete	29/11/17	01/12/17	Toni, Owen
Setup SQL Database	Medium	Delayed	30/11/17	01/12/17	Calum, Craig, Daniel
Exams			12/04/17	15/12/17	Everyone

This iteration was for the most part kept to the original iteration plan. We mainly focused on setting up what was needed for starting the creation of our project. This involved starting to familiarise ourselves with how to use server side Tomcat servlets supplied by the university running basic 'hello world' servlet equivalents. Foundation of the basic user UI was also developed and database was set up and ready for implementation. Later on we would change from the use of TomCat but at this current iteration we were still aiming to use it.
Database design was postponed on this iteration until iteration 4. This didn't require us to to modify our iteration plan as we managed to construct the database design in iteration 4 swiftly.
The total work done on iteration 1 was minimal as revision for exams was starting to take place but the plan had already accommodated for it.

Iteration 2	Medium	Complete	04/12/17	08/12/17	Everyone
Iteration 3	Low	Complete	11/12/17	15/12/17	Everyone
Iteration 4	Low	Complete	18/12/17	22/12/17	Everyone
Design and Finalise Rough Version of DB	Medium	Complete	18/12/17	22/12/17	Calum, Craig, Daniel
Design and Finalise Rough Backend Functionality	Medium	Complete	18/12/17	22/12/17	Toni, Owen
Basic User UI and Frontend Implementation	Medium	Complete	18/12/17	22/12/17	Jordan, Phil
Create Documentation	High	Not Complete	21/12/17	22/12/17	Everyone
CHRISTMAS/NEW YEAR			24/12/17	01/01/18	Everyone

There was no tasks planned during iterations 2 and 3.

Iteration 4 saw the start of the database design. This was completed swiftly and a finalised rough design of the database was agreed upon. The backend design and implementation was completed this iteration as well as the frontend UI for the user side seeing the first basic implementations.
Documentation at this stage was not started on per the gantt chart however at this stage we felt it didn't affect the iteration schedule too badly.
The work done on this part saw the completion of the rough design of different parts of the software and this first lines of code for implementation of the software.

Iteration 5	High	Complete	02/01/18	05/01/18	Everyone
Implement, Test and Revise Database	Medium	Complete	02/01/18	05/01/18	Calum, Craig, Daniel
Implement, Test and Revise Backend Functionality	High	Complete	02/01/18	05/01/18	Toni, Owen
Connect Server to DB	High	Complete	04/01/18	05/01/18	Toni, Owen
User UI and Frontend Testing and Implementation	Medium	Complete	02/01/18	05/01/18	Jordan, Phil
Connect Frontend To Server	Medium	Complete	04/01/18	05/01/18	Jordan, Phil
Create Documentation	Medium	Not Complete	04/01/18	05/01/18	Everyone

The implementation of the database was constructed at this stage and imperfections were ironed out. Specific server side requirements began to be implemented and saw the connectivity to the database and frontend. At this stage was when we decided to change from TomCat server and Java Servlet setup to just PHP. to The basic UI was starting to come together for the frontend.
Again documentation as this stage was not started and we decided it would need to be addressed soon.
The work done on this iteration saw the real bulk of the work started. The database and frontend being linked to server side scripts. This allowed basic functionality to come to fruition.

Iteration 6	High	Complete	08/01/18	12/01/18	Everyone
Start Order System	Medium	Complete	08/01/18	10/01/18	Jordan, Phil
Fill DB with Menu Items	Medium	Complete	08/01/18	08/01/18	Calum, Craig, Daniel
Implent and Test Server's Frontend Operaions	Medium	Delayed	08/01/18	10/01/18	Toni, Owen
Design DB Queries	Medium	Complete	08/01/18	08/01/18	Calum, Craig, Daniel
Finalise and Test DB Queries	Medium	Complete	09/01/18	9/01/18	Calum, Craig, Daniel
Implement and Test Server's DB Operations	High	Delayed	11/01/18	12/01/18	Toni, Owen
Display DB Menu Items Frontend	Medium	Complete	11/01/18	12/01/18	Jordan, Phil

With server side scripting connecting the front and back end the order system was started. This took longer than originally planned and spanned on to iteration 8. This pushed our testing back for iteration 8 but we allocated a generous amount of time during iteration 8 for testing and didn't affect our plan too much. Extra UI was needed to display certain information or allow certain operations as well as extra functionality required to be constructed.

The addition of data into the database was completed on time and as we all the construction of database queries. The database tasks for this iteration was completed early which allowed us to catch up on some of the documentation.

The work done in this iteration saw our first real deviation from the Iteration schedule. This required us to pushback our order implementation tasks onto week 8 and as a result shrink our testing time for the application. The database tasks were completed early however which allowed us to assign team members to other tasks. This helped out a lot which ultimately led us to completing our demo version software on time.

Iteration 7	Medium	Complete	15/01/18	19/01/18	Everyone
Order Implementation	High	Complete	15/01/18	19/01/18	Jordan, Toni, Craig, Owen
Finalise Early User Frontend	High	Delayed	18/01/18	19/01/18	
Finalise Early Database	High	Complete	15/01/18	16/01/18	Calum, Craig, Daniel
Finalise Early Backend	High	Delayed	15/01/18	19/01/18	
Finalise Early Documentation	Medium	Delayed	15/01/18	19/01/18	

During this iteration the main focus was completing the order system. The team was split into different parts working on the waiter and customer implementations of the system. We had one member work on the kitchen side but it wasn't a top priority for the manager demo.

This pushed back the finalisation of the server side, UI and documentation until the second day ofteration 8.

Iteration 8	High	Complete	22/01/18	26/01/18	Everyone
Finalise Early Backend	High	Complete	22/01/18	24/01/18	Toni, Owen,Craig
Finalise Early User Frontend	High	Complete	22/01/18	24/01/18	Jordan, Daniel, Calum
Finalise Early Documentation	High	Complete	22/01/18	24/01/18	Everyone
Test Application and Refine	High	Complete	24/01/18	25/01/18	Everyone
Prepare for Demo	High	Complete	25/01/18	26/01/18	Everyone
Manager Demo	High	Complete	29/01/18	29/01/18	Everyone + Manager

With previous tasks pushing into iteration 8's tasks we decided to shrink testing of the application by a day and push the start of this task to Wednesday. We had to push back our demo of the product however with our manager being unavailable which allowed us extra time to prepare for the demo. This was a big benefit for us.

Stage 3 Unforeseen Problems

Stage one's implementation schedule differentiates entirely to the actual course of events throughout stage 3. The team was hit with a massive unforseen disruption to our whole course and in turn this project due to UCU strikes. The disruption of this will be discussed here and the impact it had on the project. The strike dates took place between the February the 22nd to March the 16th. This was 4 weeks of disruption and affected our Iterations 12, 13 and 14. This was due to late released stage 3 specification as well as lack of motivation to go into university to work on the project as there was no lectures or lecturers but during these iterations there was still work done. No new features were added however bug fixes and refinements were carried out.

Stage 3 Revised Iterations

Below is a revised iteration schedule of what we completed for stage three. This has differentiated from stage one's estimated timeline significantly due to issues discussed above as well at changes made during our system review meetings.

Stage 3 Revised Iterations 9 - 17 – Colour Key

Type of Task	Colour
Iterations	
Frontend	
Backend	
Database	
Distractions	
General	

Stage 3 Revised Iterations 9 - 17 - Timeline

Task Name	Priority	Status	Start	Finish	Who
Complete Application Stage 3	High	Complete	29/01/18	03/30/18	Everyone
Iteration 9	Medium	Complete	29/01/18	02/02/18	Everyone
System Review and Plan	High	Complete	29/01/18	29/01/18	Everyone
Database Revised for Extra Functionality	Medium	Complete	29/01/18	02/02/18	Calum, Daniel
User Side Restaurant Selection	Medium	Complete	29/01/18	02/02/18	Toni, Jordan
Staff Side Seating System	Medium	Complete	29/01/18	02/02/18	Craig
Kitchen Side Refinement	Medium	Complete	29/01/18	02/02/18	Owen
Iteration 10	Medium	Part Complete	05/02/18	09/02/18	Everyone
User Side Restaurant Filters	Medium	Complete	05/02/18	07/02/18	Toni
User Side Restaurant Reviews	Low	Part Complete	08/02/18	09/02/18	Toni
User Side Look	Medium	Complete	05/02/18	09/02/18	Jordan
Staff Side Order System Revision	Medium	Complete	05/02/18	09/02/18	Craig
User Side Account Edit Pages	Medium	Complete	05/02/18	09/02/18	Calum, Daniel

Kitchen Side Order Separation	Medium	Complete	05/02/18	09/02/18	Owen
System Review and Plan	High	Complete	09/02/18	09/02/18	Everyone
Iteration 11	Medium	Part Complete	12/02/18	16/02/18	Everyone
User Side Guest Login	Medium	Complete	12/02/18	16/02/18	Calum, Daniel
User Side Look Refinement	Medium	Complete	12/02/18	14/02/18	Jordan
Staff Side Look	Medium	Part Complete	15/02/18	16/02/18	Jordan
User Side Restaurant Select Page	Medium	Complete	12/02/18	16/02/18	Toni
Staff Side Extra Features for Orders	Medium	Complete	12/02/18	16/02/18	Craig
Kitchen Side Improvements	Medium	Not Complete	12/02/18	16/02/18	Owen
Iteration 12	Medium	Not Complete	19/02/18	23/02/18	Everyone
Iteration 13	Medium	Not Complete	26/02/18	02/03/18	Everyone
Iteration 14	Medium	Not Complete	05/03/18	09/03/18	Everyone
Iteration 15	Medium	Complete	12/03/18	16/03/18	Everyone
System Review and Plan	High	Complete	12/03/18	12/03/18	Everyone
User Side Add Table Bookings	Medium	Complete	12/03/18	14/03/18	Toni
User Side Table Selection	Medium	Complete	15/03/18	16/03/18	Toni
Kitchen Side Look	Medium	Complete	12/03/18	14/03/18	Jordan
Staff Side Look Revisit	Medium	Complete	15/03/18	16/03/18	Jordan
Staff Side Basic Functionality	Medium	Complete	12/03/18	16/03/18	Craig
Kitchen Side Basic Functionality	Medium	Complete	12/03/18	16/03/18	Owen
Database Revision for Extra Features	Medium	Complete	12/03/18	16/03/18	Daniel, Calum
Iteration 16	Medium	Part Complete	19/03/18	23/03/18	Everyone
Staff Side Add Extra Features	Medium	Complete	19/03/18	23/03/18	Craig
User Side Order Page	Medium	Part Complete	19/03/18	23/03/18	Toni
Staff Side Look Overhaul	Medium	Complete	19/03/18	22/03/18	Jordan
UI Extras	Medium	Complete	23/03/18	23/03/18	Jordan
Manager Report Side	Medium	Not Complete	19/03/18	21/03/18	Calum, Daniel
Kitchen Side Expected Order Completion	Medium	Not Complete	19/03/18	23/03/18	Owen
System Review and Plan	High	Complete	23/03/18	23/03/18	Everyone
Iteration 17	Medium	Complete	26/03/18	30/03/18	Everyone
Refine Project and Prepare for Expo	High	Complete	26/03/18	29/03/18	Everyone
Demo at Expo	High	Complete	29/03/18	29/03/18	Everyone
Finalise Documentation	High	Complete	26/03/18	05/04/18	Everyone
Coursework Hand In	High	Working on it	05/04/18	05/04/18	Everyone

This revised iteration shows how we approached this project through iterations to complete the project. This varied dramatically from what was estimated in our stage one report but still managed to complete the project to a high quality. We never managed to complete every requirement but this will be discussed later.

Reflections on Implementation Approach

Software Process Used

From the stage one document we set out to approach the development of this project incrementally. Each stage we focused on weekly iterations to complete certain tasks by. During each system review we had an opportunity to revise our iteration plan for the coming week. Up to before the strikes this was kept to, completing many iterations on time or in advance. We found overall the incremental process an effective, easy and productive methodology. It ensured the team kept to its targets and what we had to complete during the development. Not every iteration was managed to be kept to 100% but allowed us to reiterate to see what needed to be completed. Some parts of development were a more loosely applied agile approach, especially around the times of the strikes where the team would work a bit freely on what they deemed necessary.

Experiences

During Iteration 10, we managed to complete a lot of our projects planned functionality for that iteration and decided to carry out a project review on the Friday 09/02/18. At this stage we looked back at what we never managed to complete on our system during Iterations 9 and 10 and made a revised plan for the next iteration. We carried out the extra tasks on that day and into Iteration 11 which allowed us to tick off some of the tasks. This session and planned tasks via this iteration really helped us analyse our current state of the project and evaluate our next steps.

During Iterations 12 to 14 the incremental stage plan wasn't kept to. There was confusion on how to proceed with stage 3 and lack of motivation. Some attempt to reunify the team occured but only in a sense to work on the project at home. This allowed for minimal amounts of documentation during these iterations as to actually what happened in terms of development. This stage more entailed pre-existing components being worked on and improved. This interruption required a total revision of the projects plan and the system to be reviewed at the start of iteration 15 to assess where we were at with the project. By looking back over our previous iteration plan and the systems current status we managed to prioritise certains parts of the system such as completing core functionality. The methodology helped get the group back on track for development and allowed us to set out goals to be completed.

Technologies Used Throughout Implementation

Server Side

When we first set out to work on the project we were originally going to use Java Servlets and a TomCat server to effectively handle all our backend operations. However, during actual implementation we found that PHP was a much more effective tool for what we wanted to do with this project. We then made an agreed decision to switch entirely to PHP. There required no additional setup or coding in a programming language which reduced production time on the server side. The support for PHP was vast, allowing easy connection and querying to our database, creating sessions frontend to make the flow of

the website better by storing data and the flexibility to embed it with frontend scripts. The development of the server side became simple by switching to PHP making it the most suitable server language for our project.

Front End

Our frontend scripts were built entirely from JavaScript, using the JQuery library and AJAX. This made frontend implementation simple as there was wide support for the scripting language and helped aid in the needs of the project. JQuery was a great aid to development keeping our frontend scripts short and tidy. The vast functionality provided JQuery and JavaScript as a whole allowed us to complete advanced features which PHP or HTML couldn't support by themselves. Furthermore, we used Cookies to store persistent data for users baskets to improve flow of the website. We could of implemented the users baskets being stored by PHP but we thought this process would be horribly clunky and messy. The open source Cookie scripts provided by Iamkate.com (Appendix 1.1) aided in development. HTML was the base of our project mixed with CSS and animations to improve the overall design of the site.

Database

MySQL was the database language used to query the database and PhpMyAdmin UI to make it easier to access and modify.

Overall Implementation

SnowBonk built this project from the ground up, only using the JQuery library and an open source shopping basket addon from ATN-Solutions (Appendix 1.2) to aid in development. In this respect we're very proud to have used our planning, implementing and development skills to produce our product DineBro's. The techniques instated from Software Engineering classes explaining methodologies pushed our development team to a productive and forward thinking team. Using the iteration system helped us easily set targets and goals to have parts of the project completed by and was one of the biggest drives to complete this project.

Product

Requirements Met

Functional Requirements

Requirement	Description	Status
F-UR1	Customer Ordering System	Yes/Partially/No
F-UR1-1	The system shall display a summary of the customer's order.	Yes
F-UR1-2	Shall display current order status	Yes
F-UR1-3	System shall be web based	Yes
F-UR1-4	The system shall include a feedback system.	Partially
F-UR1-5	The system should allow a customer to request a waiter.	Partially
F-UR1-6	The system shall have a low data option	No
F-UR1-7	The system shall allow the user to order from home and pickup in store.	Yes
F-UR1-8	Intelligent recommendation system	No

Our Customer Ordering System fully met 4/8 and partially met 6/8 functional requirements we set for the system. As we were developing this system, we prioritised what requirements were more essential than others and developed them first and would take the others as an added 'bonus'. Ideally we would have been able to implement F-UR1-8, but given time constraints, we weren't able to do so.

Requirement	Description	Status
F-UR2	Waiter Display System	Yes/Partially/No
F-UR2-1	Must be able to take orders for at least 4 people	Yes
F-UR2-2	Must allow waiter to modify menu items	Yes
F-UR2-3	Must allow corrections to order one order has been confirmed	Yes

Our Waiter Display System meets all of the requirements we set out for it originally, this system is by far the most complete of the 3.

Requirement	Description	Status
F-UR3	Kitchen Display System	Yes/Partially/No
F-UR3-1	Orders from the waiter system must relays orders in an unambiguous fashion to the kitchen	Yes
F-UR3-2	Customer requirements/modifications	Yes
F-UR3-3	Able to provide estimates for orders	Partially
F-UR3-4	Able to update estimates	Partially
F-UR3-5	Orders processed in FIFO manner	Yes
F-UR3-6	Orders could be read out when they come in.	Yes
F-UR3-7	Order time limit (expiration)	No
F-UR3-8	As orders come in, the system should remove ingredients from the database.	No
F-UR3-9	Able to update all orders at once	No

Currently our Kitchen System doesn't live up to what we originally planned to produce at the start of the project. This is for a number of reasons, mainly time related; we felt as the project went on that we would rather have 2/3 systems mostly completed (Customer and Waiter systems) rather than all 3 systems partially completed. As a result, the Kitchen Display System was left to the end. Given more time, we feel that most of the functionally requirements that were left unimplemented, could be completed.

Constraint Requirements

Requirement	Description	Status
NF-UR1	Security	Yes/Partially/No
NF-UR1-1	Managers have access to entire system	No
NF-UR1-2	Employees have limited access to system	Yes
NF-UR1-3	Customers have no access to backend	Yes
NF-UR1-4	Employees and customers require password	Yes
NF-UR1-5	System shall be secure	Yes

We met all the security requirements for the system with the exception of managers having access to the entire system. At the moment, Managers are on the same level as employees in the terms of permissions, but this can easily be implemented.

Requirement	Description	Status
NF-UR2	Usability	Yes/Partially/No
NF-UR2-1	System shall be user friendly and intuitive	Yes
NF-UR2-2	The system shall be easy to learn and use	Yes
NF-UR2-3	A manual will be produced	No
NF-UR2-4	Employees and customers require password	Yes
NF-UR2-5	System shall have a neutral but appealing colour scheme	Yes

We met all the requirements for security with exception to NF-UR2-3. If required, a manual can easily be drawn up.

Requirement	Description	Status
NF-UR3	Legal	Yes/Partially/No
NF-UR3-1	Customer details comply with Data Protection Act	Yes
NF-UR3-2	Customer Payments will be held in accordance with PCI DSS	No
NF-UR3-3	All data held will be stored securely	Yes

We pass all Legal requirements with exception to NF-UR3-2. At the moment we don't take customer payments so this requirement is not applicable.

Requirement	Description	Status
NF-UR4	Software/Hardware	Yes/Partially/No
NF-UR4-1	System is usable on common web browsers	Yes
NF-UR4-2	The system will be run on major mobile OS	Yes
NF-UR4-3	System shall allow concurrent access	Yes
NF-UR4-4	System shall take into account disabilities	No
NF-UR4-5	Data shall be stored using MySQL DB	Yes

The system passes all requirements with exception to NF-UR4-4. At the moment, there is no disability support for the application, but in the future this can be implemented.

Requirement	Description	Status
NF-UR5	Robustness	Yes/Partially/No
NF-UR5-1	If the system were to fail, it should restart and be functional again quickly	Yes
NF-UR5-2	System should be stable	Yes
NF-UR5-3	Low chance of data corruption	Yes

The application meets all the requirements for robustness.

Requirement	Description	Status
NF-UR6	Reliability	Yes/Partially/No
NF-UR6-1	The system will not be able to record any new order if it is not connected to the database.	Yes
NF-UR6-2	Should be bug free as money interactions and customer satisfaction are at stake.	Yes

The system meets all the requirements for reliability.

Requirement	Description	Status
NF-UR7	Cost	Yes/Partially/No
NF-UR7-1	The cost of the system should be in line with the cost provided in the Project Costing document.	Yes

The system meets all the requirements for cost.

Requirement	Description	Status
NF-UR8	Data	Yes/Partially/No
NF-UR8-1	The system shall not allow invalid data to be entered.	Yes
NF-UR8-2	The system shall allow an unlimited number of orders	Yes

The system meets all the requirements for data.

Requirement	Description	Status
NF-UR9	Performance	Yes/Partially/No
NF-UR9-1	Orders should be processed within 5 seconds	Yes
NF-UR9-2	The system shall allow for 100 transactions a minute.	Yes

The system meets all the requirements for performance.

Requirement	Description	Status
NF-UR10	Time	Yes/Partially/No
NF-UR10-1	The system shall be completed by week 13 – Semester 2	Yes

The system meets all the requirements for time.

Requirement	Description	Status
NF-UR11	Maintainability	Yes/Partially/No
NF-UR11-1	Website should be updated/edited day-to-day by managers and owners to reflect on new menus or out of stocks.	N/A

Throughout the development process we made sure to keep our original ideas in mind for each of the individual systems, coupling this with our continued reference to our requirements, both functional and non-functional we were able to keep a clear idea of what we wanted the systems to end up like. Setting out a number of clear goals throughout the development process also ensured we would meet our ambitions and deadlines and not fall behind.

Because we were able to set out clear goals for our development we were able to achieve 75% of our functional requirements or have them in advanced stages of development. Furthermore we were able to implement 86% of our non-functional requirements or have them in an advanced stage of development. We also went the extra mile and introduced several of the desirable features which were outside our list of functional and non-functional requirements.

This high rate of completion for our requirements and the successful implementation of desirable features have ensured we produced an aesthetically pleasing yet functional application that has great potential.

WHAT MADE US SPECIAL

USER SIDE ADDITIONAL FEATURES

This section highlights some of our key features that have been implemented in our system including those that are additional to our requirements. Along with each feature will be a brief discussion and the benefits it brings our customers.

At the end of our development process and after viewing our competitors systems it was clear that our product stood out amongst our competitors. A number of notable design choices and unique features really made our system stand out whilst making it more approachable and usable. Our design choices from the beginning gave us a winning start. Our simple User Interface(UI) and neutral colour scheme gave us a great template to work with and build our system on. This also ensures clients have a neutral platform for there service to be advertised on as opposed to a one sided system that suits some but looks wrong for others.

The ability for users to use our system without having to go through a lengthy sign up process or even have to go through the trouble of remembering what password you used really struck home with us which is why we opted for a guest login system so at the click of a button the user could have a tasty meal on it's way. The guest login has all the same functionality as a user account.

Our wholly unique filter system was implemented to give our user maximum control in our system. The filter ensured the user could browse our system by type ensuring they can see all the restaurants serving their favourite types of food. The ability for our user to also search for their favourite restaurant quickly via a search bar was another important feature for us as this boosted usability for the user and limited time they would have to spend searching on our site. The search bar coupled with our filter system really gave us a unique search system.

Due to our range of clientele, from fast food to fine dining we had to implement a table booking feature for the fine dining restaurants on our system. This was too ensure full functionality for the user allowing them to book a table in there favourite fine dine restaurant all at the click of a button. Another important feature to us is when the user makes an order with their account. That order will become part of there order history which they can view at anytime. This allows users to easily remember what that great takeaway and what great items they ordered a month ago.

The system is currently only a web based application. Our system works on any device with any major browser. This is true for a range of devices, whether it is mobile, tablet or desktop. We wanted to ensure that users can access our system from any device. We have a plan to launch on both iOS and Android in the future.

Waiter Side Additional Features

This section highlights some of our key features that have been implemented in our system including those that are additional to our requirements. Along with each feature will be a brief discussion and the benefits it brings our staff.

Splitting the bill is common practice nowadays and we decided to implement a feature in the waiter system to allow the staff to quickly generate separate bills and reduces the chance of them making any errors.

In a restaurant, there may be discounts for staff and friends and family etc. We have implemented an additional feature to allow set percentages to be removed from the final bill. This could be particularly helpful for staff because it is far easier to press a button and calculate a percentage instantly rather than off the fly, which could lead to mistakes given that the atmosphere in a restaurant can be stressful when busy.

Another valuable feature which was added was customer notes. These notes can be sent to the kitchen and can inform the kitchen staff of any requests that the customer might have, this could be dietary requirements, food alterations etc. In addition to this, predefined notes have been added for items such as steak where there is commonly a choice of how it should be cooked. This saves the waiter's time as they can select the 5 or so options quickly.

Robustness of System

The system that we created was both aesthetically pleasing and functional. All parts of the current systems we have implemented are fully functional ensuring a user friendly experience. We have also taken care to ensure beta functions remain offline until they have been vigorously tested to ensure that users still have a bug free experience when using our system. This, plus our testing strategies ensures there is a low chance of system failure and data corruption. And on the slim chance of a system failure it will require a simple reload of the page.

Our security has been a high priority ensuring that the system remains functional and that user data is protected. If we were to lose the database or the important information on the database it could be disastrous for the systems functionality and our users experience. We have implemented several methods to protect our users data; encryption on their passwords and preventing malicious data from being entered into the database. All this comes together to make a robust and secure system, it works in a range of conditions and can withstand malicious attacks.

How Useable Was the System

Following the Usability study we conducted the System(s) were highly usable and gave a positive user experience from the feedback we received. The main highlight of the feedback was the general aesthetics of the systems were done to a high standard which resulted in heightened usability and user experience.

For full analysis please view our usability study.

Conclusion

Review of Project

We believe the project has had its ups and down but was overall a generally positive experience. From beginning as a group who didn't know each other into a fully social and hardworking team has been fun. By grouping individuals with different strengths and backgrounds has helped shape us into a fully capable development team. By pushing through late nights to early mornings we have shown ourselves how committed we all are to bringing out our full potential. Team Snowbonk are really proud of what we have produced during this two semester project and really think has prepared us for industry.

Reflecting on our project we believe we managed to create some unique features such as implementing a guest login system to make the product flow well and not stop the user on a tedious sign up page. We also implemented a filter feature to pick desired dining options with ease.

We believe we could have achieved more functionality and extra features if we hadn't missed out on iterations during our stage 3 but believe this is a minor setback on our overall project. We never hit all the requirements in the specification but were in progress implementing,

- The waiter call system from user side
- Intelligent Recommendation System
- Stock level controls
- Time expectation from kitchen side
- Delivery and Collection

The project overall was a great success which really introduced the team into industrial level development methodologies. We gained first hand experience diving into working with an unknown team, making project plans and iterations, democratically deciding the direction of the project and ultimately producing something unique.

Future Improvements

The only future improvements team SnowBonk would like to see is a more fair and consistent marking system comparing amongst different group. Some groups are advantaged by having no IS members which means less opportunity to lose marks. Other groups have managers who aren't from technical backgrounds and as such wouldn't grade fairly as some who are.

Appendix

Links - 1

1.1. http://code.iamkate.com/javascript/using-cookies/

1.2. http://www.atnsolutions.com/atn-simple-cart/

VERSION 1.2
APRIL 5, 2018

FINAL USABILITY STUDY & REPORT

BY

TEAM SNOWBONK

Presented by: *Jordan Walker, Owen Welch, Philip McQuaid, Antonio Gargaro, Craig Quint,Danial Barker, Calum McLean*
SNOWBONK
HERIOT WATT UNIVERSITY, EDINBURGH

Contents

DOCUMENT INTRODUCTION

PURPOSE

The Purpose of this Document is to show the transition of the project from the requirements specification and "first drafts" of the implementation we created for the group project to the final Alpha implementation we displayed at the Expo on Thursday 29th/03rd /2018. This final Alpha implementation is the product we will be using to run this current usability study which will give our group relevant and un-biased feedback on the project we have completed.
The document shall show the creation of usability testing plans, mock up designs, overview of design features and results from subjects who took part in the tests. From this we will be able to see any design flaws or problems we as a group did not realize initially.
The Document itself is intended to be read by;

- Our Group Manager Chris Fensch
- Rob Petrick who is leading the Project and will be assessing this document.

SCOPE

The Scope of this document is to understand and test Users that are not in any way affiliated with the group members as to provide an unbiased and fair review of the implementation we have created. To do this we will aim to gain information from around 6-7 individuals who will agree to take part in the Usability Study.

From this we can highlight areas that we could of improved upon and made more clear for the general user of one of our three systems. Alongside this it will also give us the opportunity to see what we have succeeded in implementing that makes the system as a whole and individually pleasing to use.

The Scope of the project is to create three different systems that will be able to communicate data between each system. I.E a User may place an order to a Waiter/Waitress which they can then modify and then forward this information onto the kitchen system who can then forward information about the status of the customer's order to the original user.
The Web App must be available on all devices thus including;

- iOS
- Android
- Windows

Due to this factor the application must be responsive to screen size and there is no limitation to the user not being able to use the application based on his/her device.
This also means that the Web application needs to work on the following browsers;

- Chrome
- Safari
- FireFox
- Internet Explorer

Data regarding the Waiter/ Kitchen Systems will be kept in a MySQL database which will be Meals, Sides, Cost of Product, Quantity, Order Number, Staff ID etc.
User data will also be kept in a MySQL and will refer to their First and Last Name, Contact Telephone, Email Address, Address and Password.

OVERVIEW

The Document will be split into the following sections;

- Mock-Up Designs and Choice of Design to Be Carried Forward
- Test Planning
- Results of the Test Subjects
- Conclusion with note to any changes
- Appendix

The Mock-Up Designs will showcase the designs we have created and how we vision the web application too look. There will be multiple mock-ups, so we can take the best points from each Mock Up and then apply them into two final Mock-Ups which we will create small prototypes for to move towards testing the application itself.

The Test planning will outline how we believe is the best way to gain useful information about our web application. This will show the strategies we will implement to test the user to gain qualitative and quantitative information useful to the project.
The Results of the Tests Subject will show the data we have acquired from running our tests along with useful feedback that may illustrate changes that need to be made to the Application(s).

The Conclusion shall outline our summary of what we found from running our tests and any changes that the group feels need to be modified to better the success of our Application(s).
The Appendix will contain User Consent Forms, Questionnaires, Examples of Mock-Up Designs and all other referenced materials.

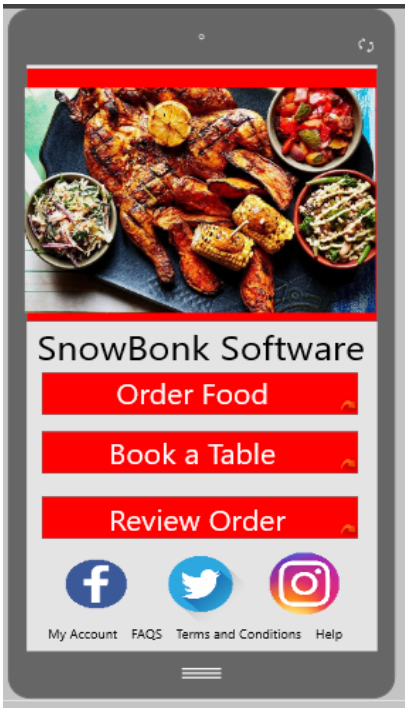
NOTES

- The Mock Ups shown are those brought forward into the development stage of the project.
- The initial questionnaire that shows the demographic of our participants has been adapted from the groups Stage One questionnaire and has had the questions regarding Gender and Age removed in replacement of new questions under review from Chris Fensch.
- The Consent Form, Testing Protocol and Post Questionnaire documents have been modified to reflect the testing purposes of the Stage 3 implementation of the project.

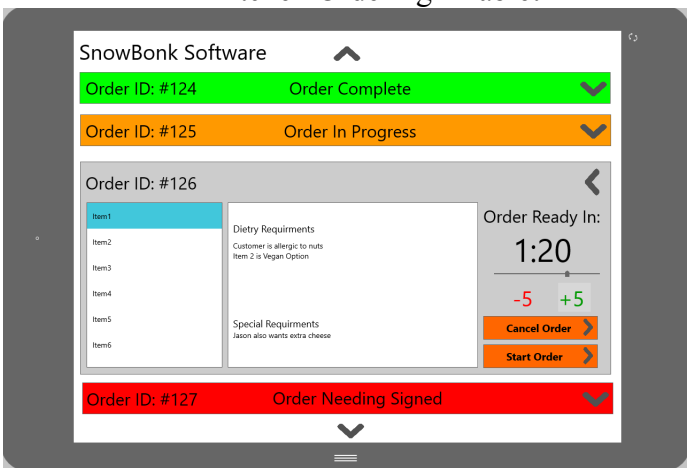
MOCK-UP DESIGNS BROUGHT FORWARD

For Stage One we were tasked with developing initial "mock-up" designs to bring forward towards the initial stages of implementation. The mock ups created are shown following;

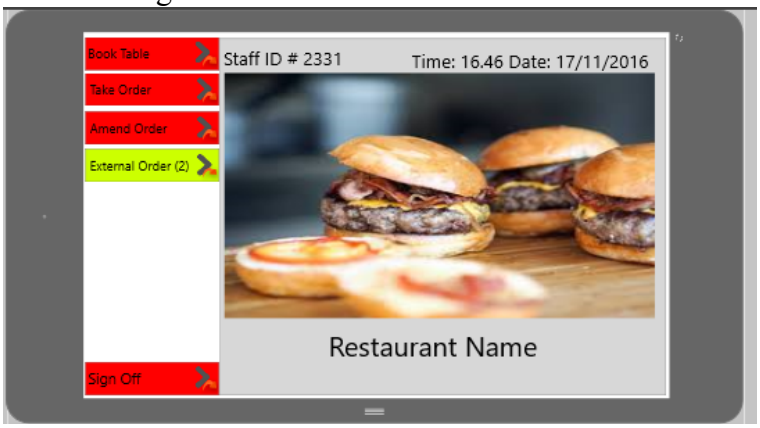
User – Phone



Kitchen Ordering - Tablet



Waiter Ordering - Tablet

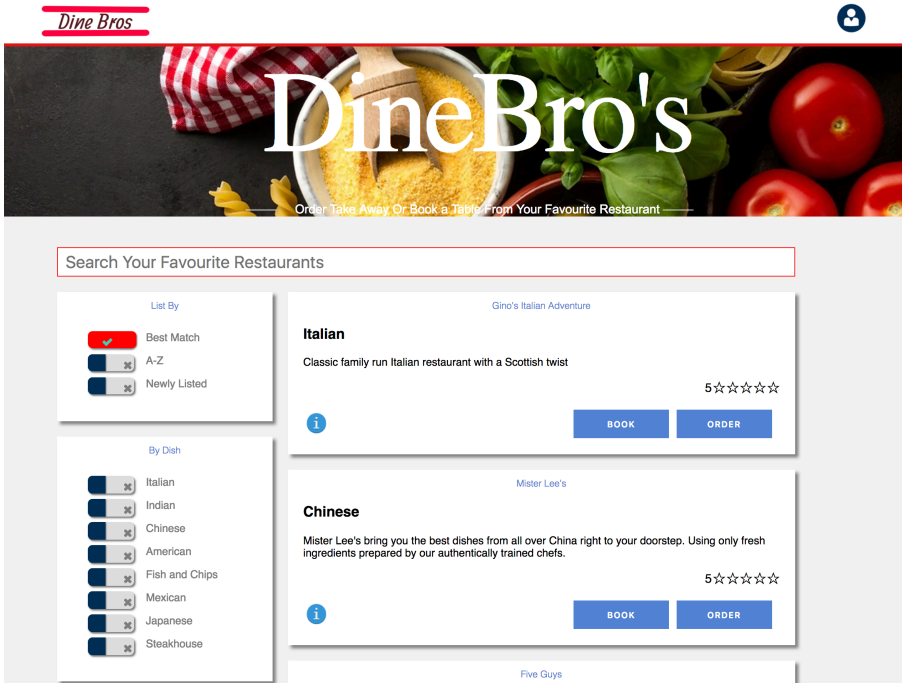


Upon reflection the final implemented design of our project held vaguely true to the design and implementation we wanted as a group to create. A number of alterations have been made however to make what we believe a more usable and friendly system for the User.

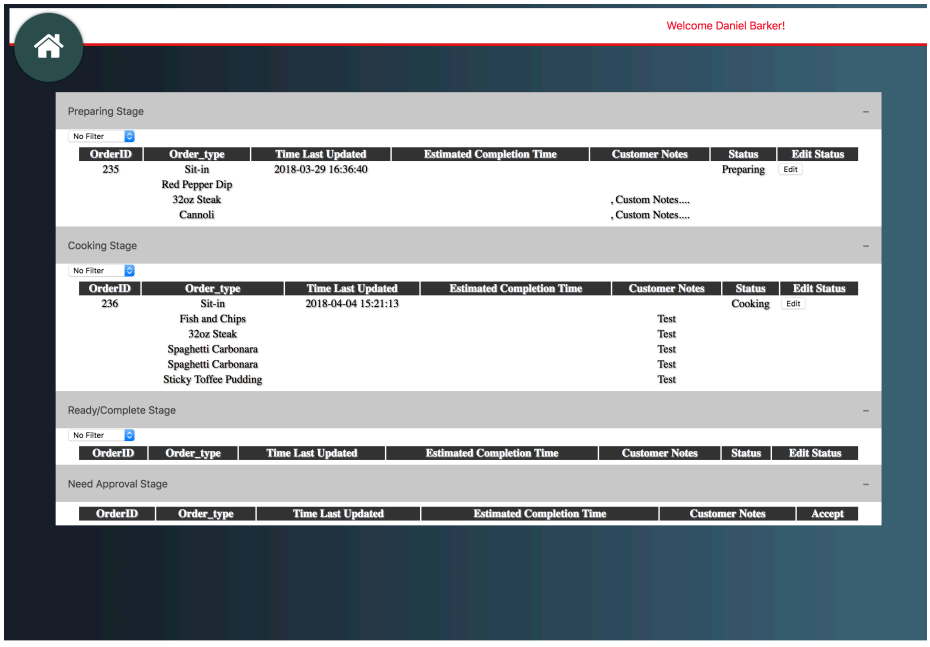
Initially when we tested these designs that we created in "Balsamiq" and used them for our first Usability study we gained positive feedback and a larger understanding of what we were trying to implement in the end product.

FINAL DESIGN OF THE SYSTEM

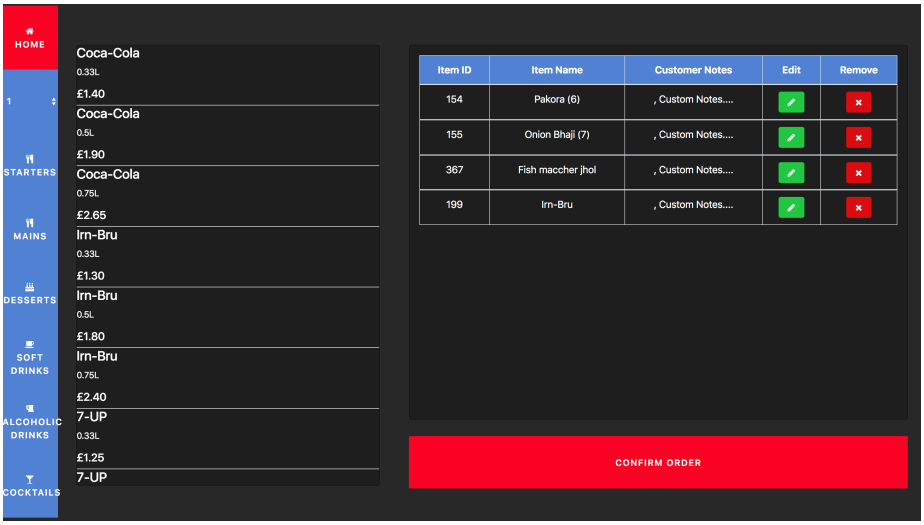
CUSTOMER USER INTERFACE – LAPTOP VIEW



KITCHEN USER INTERFACE – LAPTOP VIEW



WAITER USER INTERFACE – LAPTOP VIEW



COMPARISON

During this stage of the Usability study we can now use our Alpha prototype product on the Users. It is important to note that the design of the interfaces we brought forward in Stage 1 have dramatically changed in certain areas for various reasons including;

- Feedback from the Stage One Usability Report
- Feedback from our Group Manager Chris Fensch
- General Feedback from peers in the University
- Group Decisions in meetings.

Although not fully complete we believe our design was highly intuitive and unique compared to other applications that have been made to solve this certain specification. We as a group believe that the design brought forward to the Alpha Implementation is one that both satisfies the user experience and also aids in the usability of the product itself.

Given more time we believe we could of created a more rigorous testing environment for what we think is an ambitious and unique solution to the problem we faced and hopefully the usability study will showcase that the DineBros Web Application created for Customers, Waiters and Kitchen Staff is successful in validating what we believe in our application.

TEST PLAN

AIMS

In this section, the group will outline the plan we are going to use to test the usability of our systems. The Test aims are as follows:

- Test that the different systems convey the appropriate information to the user.
- Check that the systems meet the functional requirements proposed by the group.
- Control that the system has a positive user experience as a baseline that can then be developed upon.
- Test that the system does not have any user performance or experience problems that might prevent a successful implementation of the project.
- Collect relative qualitative and quantitative data from the user who participate in testing.

METHODOLOGY

LOCATION

The Systems are designed to be used in a food ordering service. Providing three different systems for the client, waiter/waitress and the kitchen staff. The systems shall be tested on participants from the public and will not be affiliated with any of the group members in any way to prevent bias in the test results.

Due to the fact that these tests will be conducted in a location that is suitable for both testers and participants potential external variables may appear that cannot be controlled.

PARTICIPANTS

For this Usability Test, we will collect data from around 6 – 12 participants that are not in any way affiliated with any group members or indeed connected with the University of Heriot Watt. The participants can be of any age but for legal concerns they must be above the age of 16.

From our participants, we expect a series of tasks to be completed as easily as possible using our final implemented systems and then answer a series of questions after completing the given tasks. The participants of this usability case study are responsible for giving honest and reliable information regarding the usability of the systems.

TRAINING

Before the participant starts the assessment of our system they will be given a briefed description of the purpose of this project and what each system has been created for. They will however not receive formal training on how to use the systems, so the group can assess the intuitiveness of the systems themselves to allow arrangements for any further developments in both usability and functionality.

PROCEDURE

The participants of the study will be tested in suitable locations that are convenient to themselves. The Group Member(s) overseeing the test participants and the study will first ask them to fill out a Consent Form and remind the participant that it is the systems we are testing and not the individual. During the process, there will be no digital recordings on any device.

They will then be asked to complete a pre-questionnaire that gives a small amount of detail regarding the participants knowledge of kitchen, waiter systems and their own food ordering habits.

After this the participant will be shown a variety of low-fidelity prototypes of our system on a laptop which will mimic the groups idea of how the functionality of the systems will work. By completing the tasks set on these proto-type systems we aim to find any underlying errors that may appear within the systems. These tasks are derived from what we believe will be the most popular actions that will be completed by the end users of these systems and every participant will be asked to complete the same tasks.

During these tasks, an invigilator will be overseeing the progress and will be making notes throughout in order to correspond to error finding, and questions that are being asked to the participant. The invigilator will not give any immediate help to participates if it seems they might be struggling. However, if it becomes the case that the participant is confused about a certain part of the system the invigilator will help guide them towards the next stage of the test and will make note of this event.

After the tasks are finished the participant will then be asked to complete a final questionnaire that contains a series of questions designed to evaluate the systems and certain functionality of the systems.

REQUIRMENTS TESTING IN USABILITY STUDY

All the tasks that are being asked of the participants to complete are relevant to the requirements we have identified in our Stage 1 Requirements Specification. For the study we created testing that highlighted the fully complete areas of the system.

Figuratively we have chosen to mainly focus on the “Must Have” requirements that have been described in this document.

USABILITY TEST GOALS

This section describes the Usability test goals we are setting for the systems created by SnowBonk.

COMPLETION RATE

For the Usability Test we want a completion rate of 100%, this is calculated by the amount of people who participate within the usability study and complete the tasks given to them fully.

ERROR FREE RATE

The error free rate we aim for is 90%., this is calculated by the amount of people who complete the tasks given without any errors either critical or non-critical.

PROBLEM SEVERITY

To correctly find where the errors lie and how detrimental they are within our systems, we will categorize them to a suitable severity category based on the issues they hold.
The impact of a problem is classified according to the impact it has on successful task completion.

- high: if it causes a critical error
- moderate: if the problem results in a non-critical error
- low: minor problems that do not significantly affect the task completion (non-critical error)

We will also measure the frequency the problems by the percentage of how many of the test participants encounter the problem.
The Frequency will be classified according to frequency.

- High: 30% or more of the participants faced the challenge.
- Moderate: 11-29% of the participants encountered the problem.
- Low: 10% or less of the participants encountered the problem.

According to these two factors, problem severity can be categorized into four groups:

1. An error with a high impact and high frequency.

   These errors must be dealt with to reduce development and training costs and improve user satisfaction.
2. An error with a moderate or high impact and a moderate or low frequency.

   Solving this the problem will usually reduce training costs and time-on-task.
3. An error with a moderate or low impact but a high frequency.

   Solutions to these problems will reflect in a reduce time-on-task and increased user satisfaction.
4. An error with low impact and moderate or low frequency.

   This will result in increased user satisfaction.

Once the tests are completed a full overview of the severity of the systems will be given as part of the Usability Report.

TESTING PROTOCOL

SNOWBONK WEBSITE TESTING PROTOCOL

Invigilator Name:

Date:_ _ _ _ _ _ _ _ _ _ _ _ _ **Time**: _ _ _ _ _ _ _ _ _ _ _ _ _

Candidate Number:

_____

Purpose

What we aim to achieve in this scenario is to see how the user interprets the system design we have created for a food ordering system. The system is called “DineBro’s” and we aim to:

- Make ordering food, booking tables and reviewing orders easier for the user.
- Create a simple and effective ordering, booking and submitting system for the waiter/waitress.
- Develop an easy-to-use system for kitchen staff that allows them to review open orders as well as any possible requirements of the users themselves in order to give them an expected time for when their food will be ready.

Brief

Today I will ask you to describe our systems to us from what you see on the screen. I shall also ask you to complete a variety of tasks on all three of these systems. Throughout this I will be taking notes from your descriptions and from you completing the tasks. All these notes will be anonymous and are for internal uses only.

After the tests, I will ask you to fill out a small questionnaire to help us develop our systems and make them as effective and accessible/supportive as possible. You are more than welcome to stop the test at any time and if so all information regarding notes will be destroyed.

TESTER QUESTIONS

User

1. **Customer Web App**

1.1 Describe the page.

1.2 Can you filter the restaurants to show those that are both "Chinese" and "Italian"?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.3 Can you create a food order?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.4 Can you now go to the order tracking page?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.5 **Describe the page**

1.6 Can you create an account?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.7 Can you go to "MyAccount"?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.8 Can you now Book a table?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.9 Can you now edit your own details?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

1.10 Can you now find out more information about the website?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

2 Kitchen System

2.1 How would you describe this page?

2.2 Is the page clear in showcasing where the orders are?

2.3 Can you list all the orders that are in the "Preparing Stage"?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

2.4 Can you filter the orders in the "Cooking Stage" to only show "Mains"

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

2.5 Can you accept an Order needing approval?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

2.6 Can you change an Order from the "Preparing Stage" to the "Cooking Stage"

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

3 Waiter/ Waitress Order System

3.1 How would you describe this page?

3.2 Can you book a Table for a customer?

Completion Rate:

0. Couldn't Complete *1. Completed with minor difficulty* *2. Completed with ease.*

3.3 Can you create an order for the booked table?

Completion Rate:

0. Couldn't Complete *1. Completed with minor difficulty* *2. Completed with ease.*

3.4 Can you edit the order you created for the customer?

Completion Rate:

0. Couldn't Complete *1. Completed with minor difficulty* *2. Completed with ease.*

3.5 <u>Can you complete a cash order for the table with a "friends and family discount"?</u>

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

3.6 <u>Can you go back to the start screen and then amend an order?</u>

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

3.7 <u>Can you Describe the page?</u>

3.8 <u>Can you now log out the system?</u>

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

Debrief:

Thank you for taking part and completing our short test. You did fantastic! What I ask now is that you complete a short questionnaire/survey that is intended to gather more information and feedback on the systems you have seen today. By doing so you are helping us to create a variety of systems that are user friendly and will be more beneficial *(or more accessible)* for the end users.

The questionnaire will take around 5 to 10 minutes to answer and is very straightforward. We ask you to write an honest opinion about the systems so that we can do a precise evaluation about possible improvements. Once again thank you for your time, it is greatly appreciated by us all at snowbank.

RESULTS OF THE USER
TEST AND QUESTIONNAIRE FINDINGS

PRE-QUESTIONARRE

1) Currently what is your position?

Currently what is your position?	
Professional	1
Student	6
Other	0
Total	7

2) Do you rent or own your own home?

Do you rent or own your own home?	
Yes	7
No	0
Total	7

3) Have you ever worked as a waiter or in a kitchen?

Have you ever worked as a waiter or in a kitchen?	
Yes	3
No	4
Total	7

4) How often do you order food?

How often do you order food?	
Rarely (1-2 Times a month)	1
Often (1 – 2 Times every two weeks)	5
Regularly (1 – 2 Times a week)	1
Total	7

5) What days are you most likely to order food?

What days are you most likely to order food?	
Weekdays(Monday- Friday)	1
Weekend (Friday- Sunday)	6
Total	7

6) Around What Time are you most likely to order food?

Around What Time are you most likely to order food?	
Morning (8a.m – 12p.m)	0
Early Afternoon (12p.m – 5p.m)	0
Evening (5p.m – 9p.m)	5
Late Night (9p.m. 12a.m)	2
Early Morning (12a.m – 8a.m)	0
Total	7

7) How Often do you reserve a table at a restaurant?

How Often do you reserve a table at a restaurant?	
Rarely (1-2 Times a month)	6
Often (1 – 2 Times every two weeks)	1
Regularly (1 – 2 Times a week)	0
total	7

8) What days are you most likely to book a table?

What days are you most likely to book a table?	
Weekdays (Monday- Thursday)	0
Weekend (Friday- Sunday)	7
Total	7

9) When ordering food which website/app do you use most?

When ordering food which website/app do you use most?	
Just-Eat	5
Hungry House	2
Deliveroo	0
Uber Eats	0
Other	0
Total	7

10) Which Browser do you use most?

Which Browser do you use most?	
Google Chrome	5
Mozilla FireFox	2
Safari	0
Internet Explorer	0
Opera	0
Other	0
Total	7

11) Which Device are you most likely to order food from?

Which Device are you most likely to order food from?	
Mobile Phone	4
Laptop	2
Ipad	1
Desktop Computer	0
Other	0
Total	7

12) If using a mobile device, do you prefer ordering through an app or the mobile website?

If using a mobile device, do you prefer ordering through an app or the mobile website?	
App	1
Website	5
No Difference	1
Total	7

13) Are you willing to try new restaurants based on previous customer satisfactory?

Are you willing to try new restaurants based on previous customer satisfactory?	
Yes	4
No	0
Maybe	3
Total	7

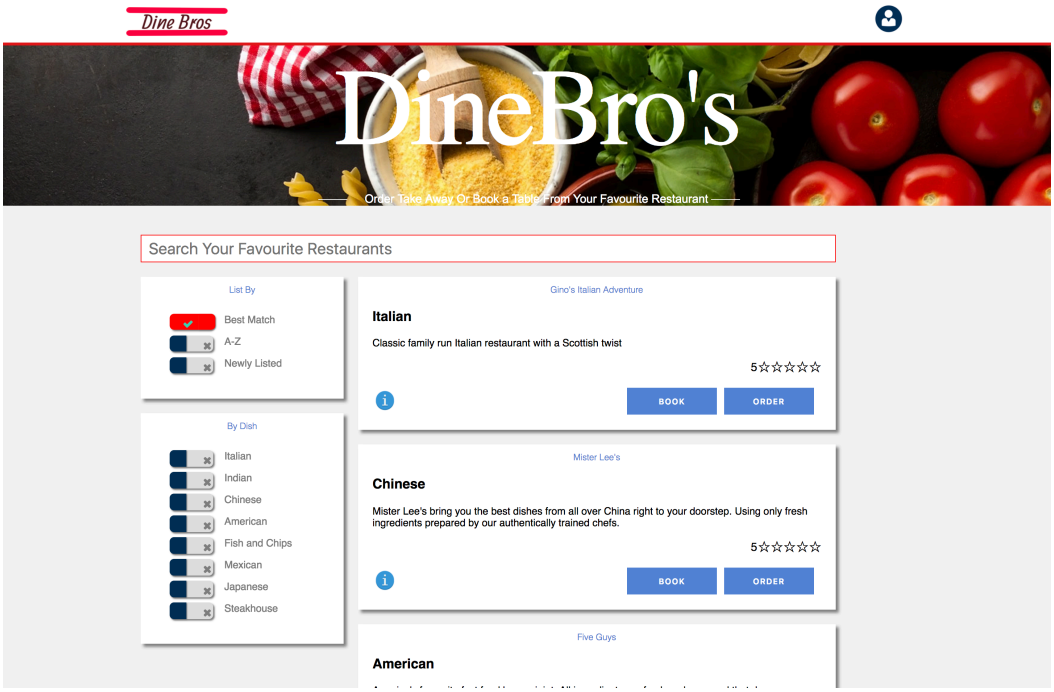
14) I would sign-up for a newsletter in order to hear about special offers and recommendations

I would sign-up for a newsletter in order to hear about special offers and recommendations	
Yes	0
No	6
Maybe	1
Total	7

TEST STUDY

Customer Web App

1.1 Describe the page.



All users correctly identified the page as the “Homepage” of the customer interface.

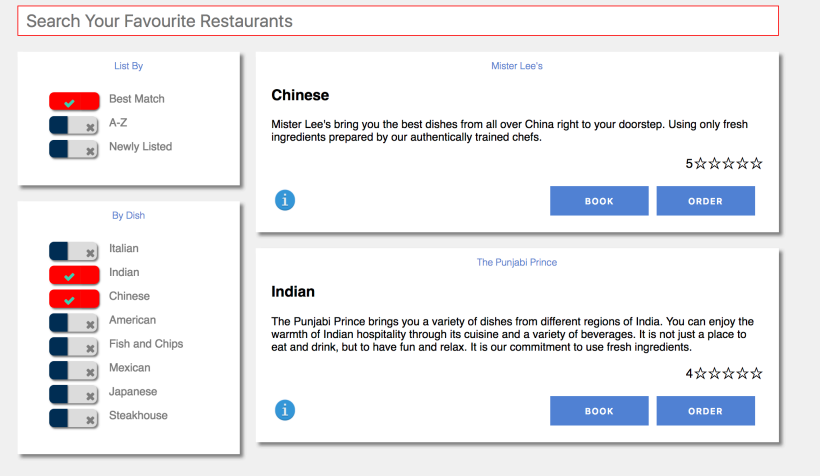
Users were quick to point out what they believed each button would be used for.

Users also understood the icon for the restaurant upon clicking that it would specify if the restaurant was vegan or vegetarian friendly.

1.2 Can you filter the restaurants to show those that are both “Chinese” and “Italian”?

Completion Rate:

0. Couldn’t Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn’t Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

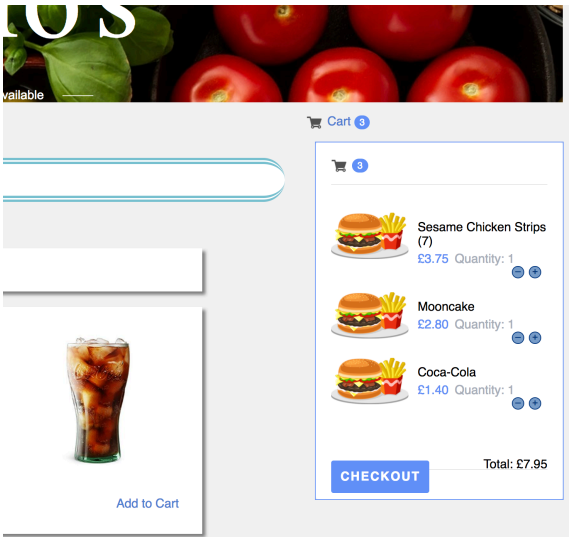
All users successfully completed the filtering of the restaurants to show only the “Chinese” and “Indian” restaurants.

Users were quick to identify to filter they could use the checkboxes on the left of the page.

1.3 Can you create a food order?

Completion Rate:

0. Couldn't Complete *1. Completed with minor difficulty* *2. Completed with ease.*



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

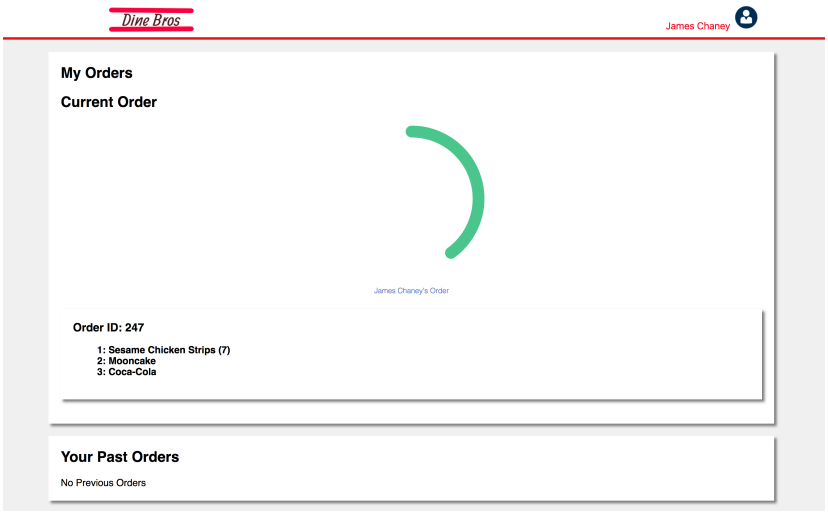
All Users quickly created a food order from any of the restaurants they chose.

The users also quickly identified how to remove duplicates of items if they wanted to edit their order in the initial stages.

1.4 Can you now go to the order tracking page?

Completion Rate:

0. Couldn't Complete *1. Completed with minor difficulty* *2. Completed with ease.*



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

Users quickly nevaigated to the order tracking page and commented on how they liked the idea of using a circle to indicate how far along the order was.

One user also expressed the liking of how orders were split into "current orders" and "previous orders".

1.5 Describe the Page

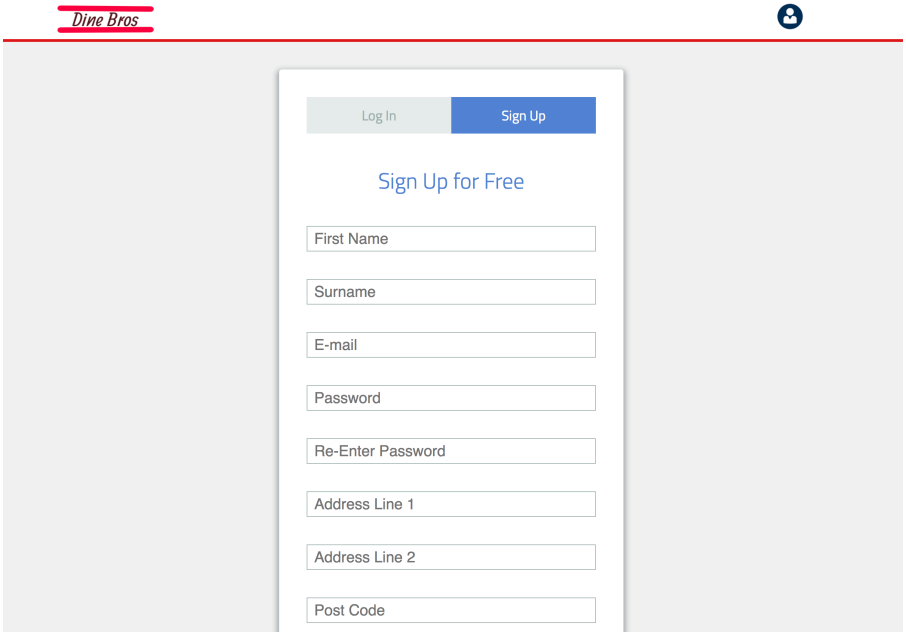
Users all identified that the page showcased how far along the order was (Although the page is not fully functional due to the timings not being passed from the kitchen to the user upon the testing of the usability study).

Users also described why there was different sections for the previous and current orders.

1.6 Can you create an account?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

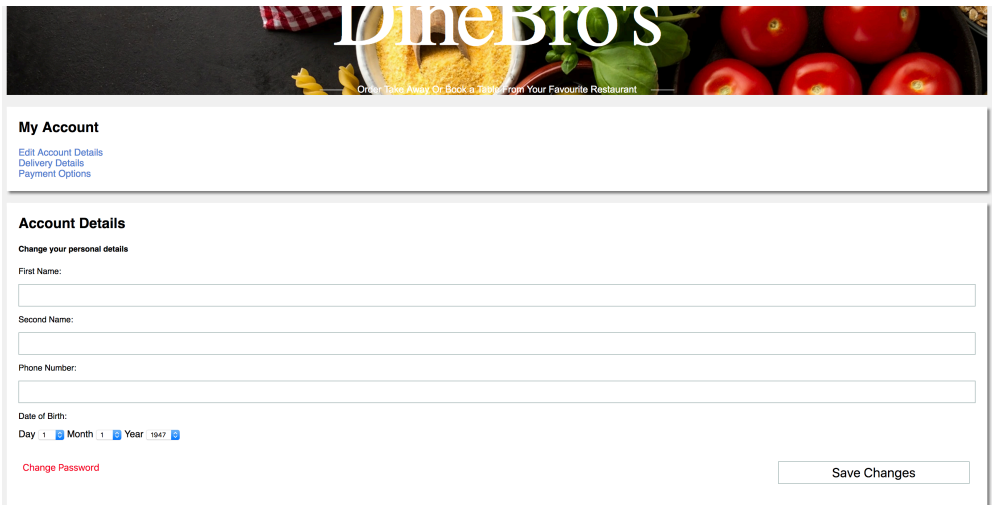
All Users completed the task assigned and with ease.

Users quickly moved the cursor to the top right corner onto the "MyAccount" icon and quickly set themselves up with an account.

1.7 Can you go to "MyAccount"?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

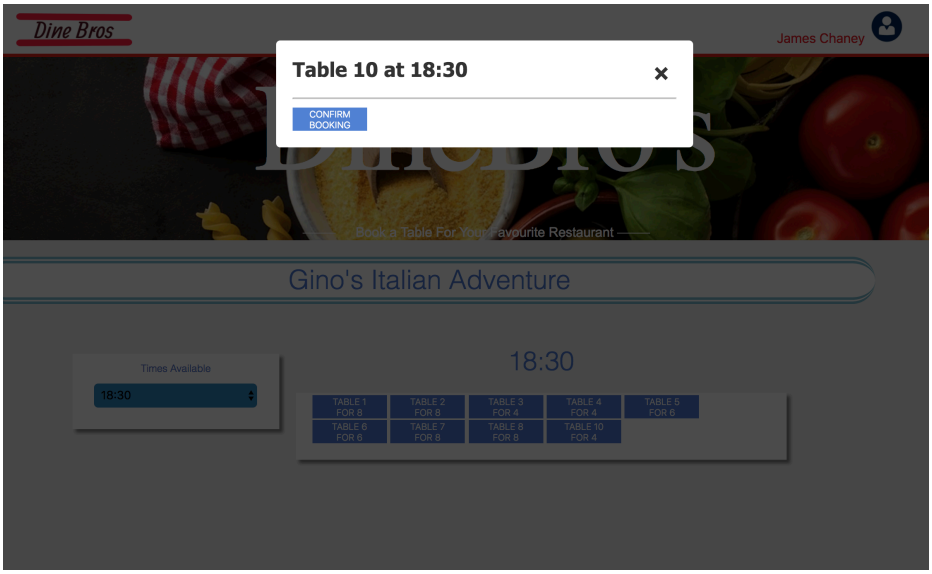
Upon completing a sign up form the users were then asked to go the "MyAccount" page.

All users correctly navigated to the page with no hassle.

1.8 Can you now Book a table?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

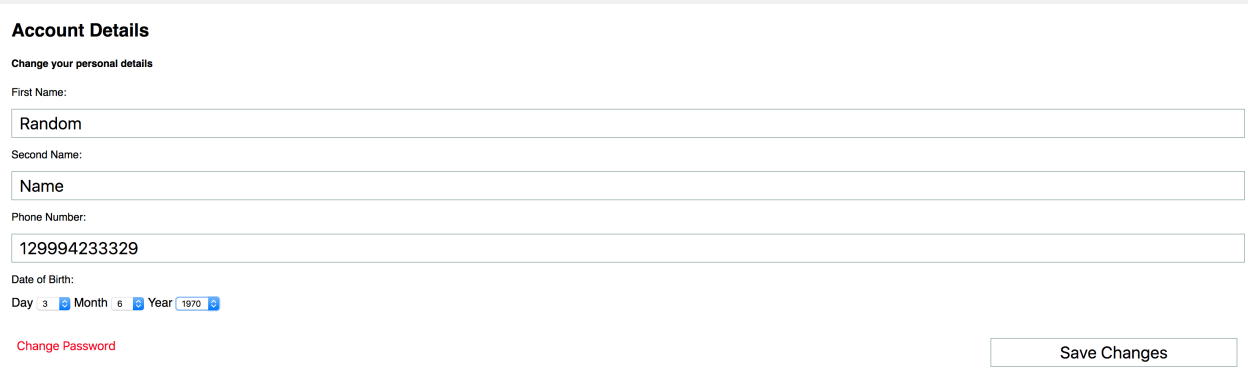
All Users completed the task assigned and with ease.

Users were very quick at completing this process.

1.9 Can you now edit your own details?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

All Users completed the task assigned and with ease.

There was a bit of hindrance however with users waiting for a prompt to change their details with one making a suggestion that we should have place holders in each of the text boxes.

1.10 Can you now find out more information about the website?

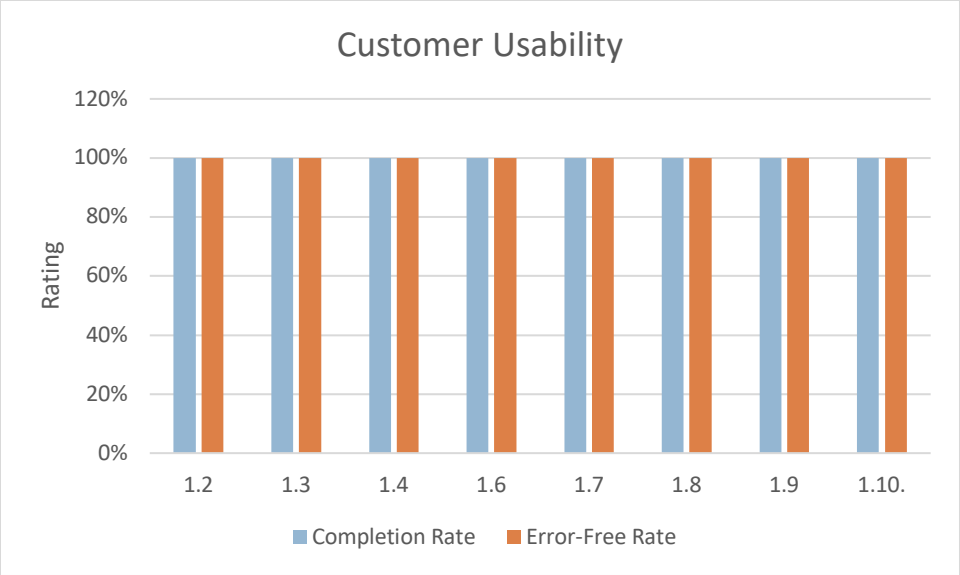
Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.

Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**0**
Completion Rate	**100%**
Error-Free Rate	**100%**

Users were quick at navigating to the footer of the webpage and choosing from a range of extra information regarding "DineBros".

The question prompted to go to the FAQ page which all users did.



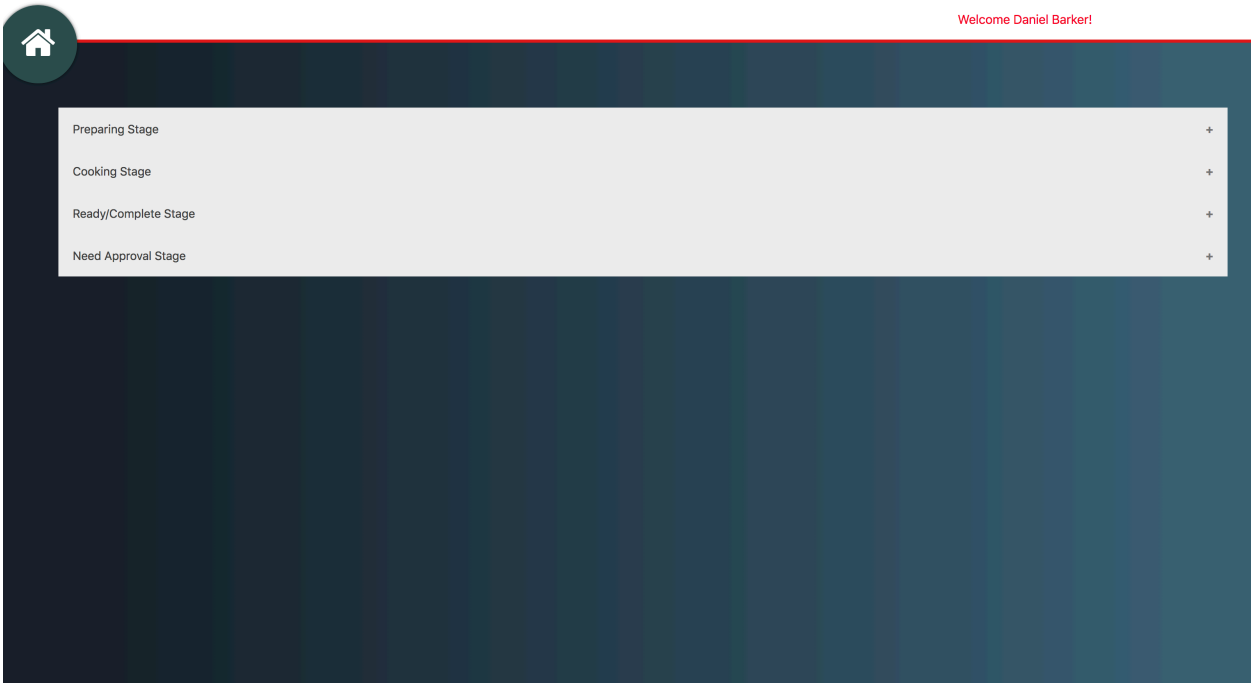
AREAS THAT NEED CHANGING

Although every test came back with a zero error rating and a 100% completion rate there is still areas for improvement if the project was to be carried on.

This mainly applies to overlooked areas such as making sure all forms have place holders and other similar details.

2 Kitchen System

2.1 How would you describe this page?



There was some initial confusion upon describing the page as it was designed to only have one view due to the group believing the "user" shouldn't have to navigate different pages to get to a certain order.

Once users had a minute to gain familiarity with the system they all correctly described what each section and dropdown was for.

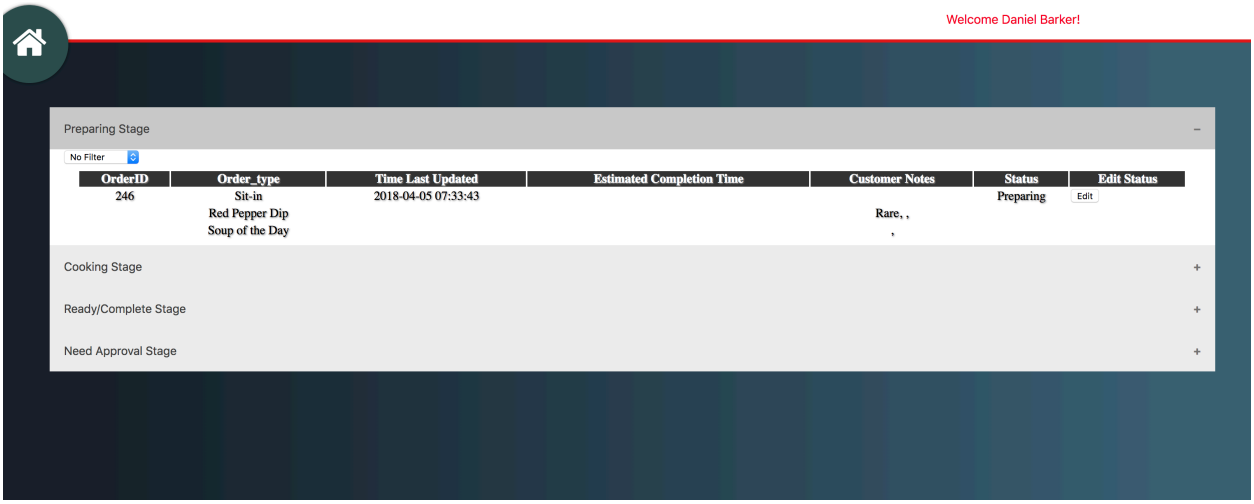
2.2 Is the page clear in showcasing where the orders are?

All Users correctly showcased which section was for which orders and which dropdown they would navigate to view orders in that current process.

2.3 Can you list all the orders that are in the "Preparing Stage"?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



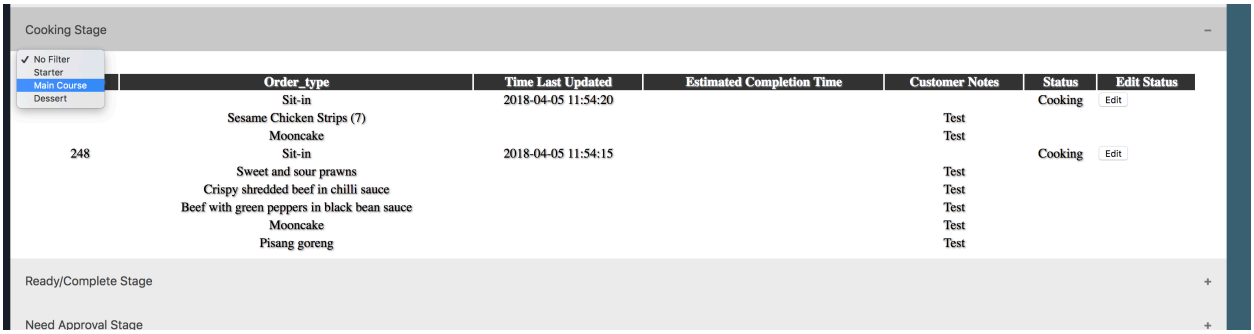
Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

All Users completed the task assigned and with ease with no issues.

2.4 Can you filter the orders in the "Cooking Stage" to only show "Mains"

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



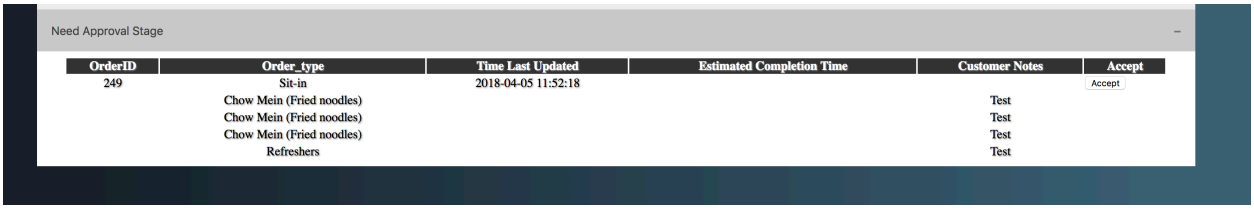
Couldn't Complete	0
Completed with minor difficulty	**1**
Completed with ease	**6**
Completion Rate	**100%**
Error-Free Rate	**85.7%**

All Users completed the task however one user was unsure of using the drop down selection inside the particular order process to filter orders.

2.5 Can you accept an Order needing approval?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



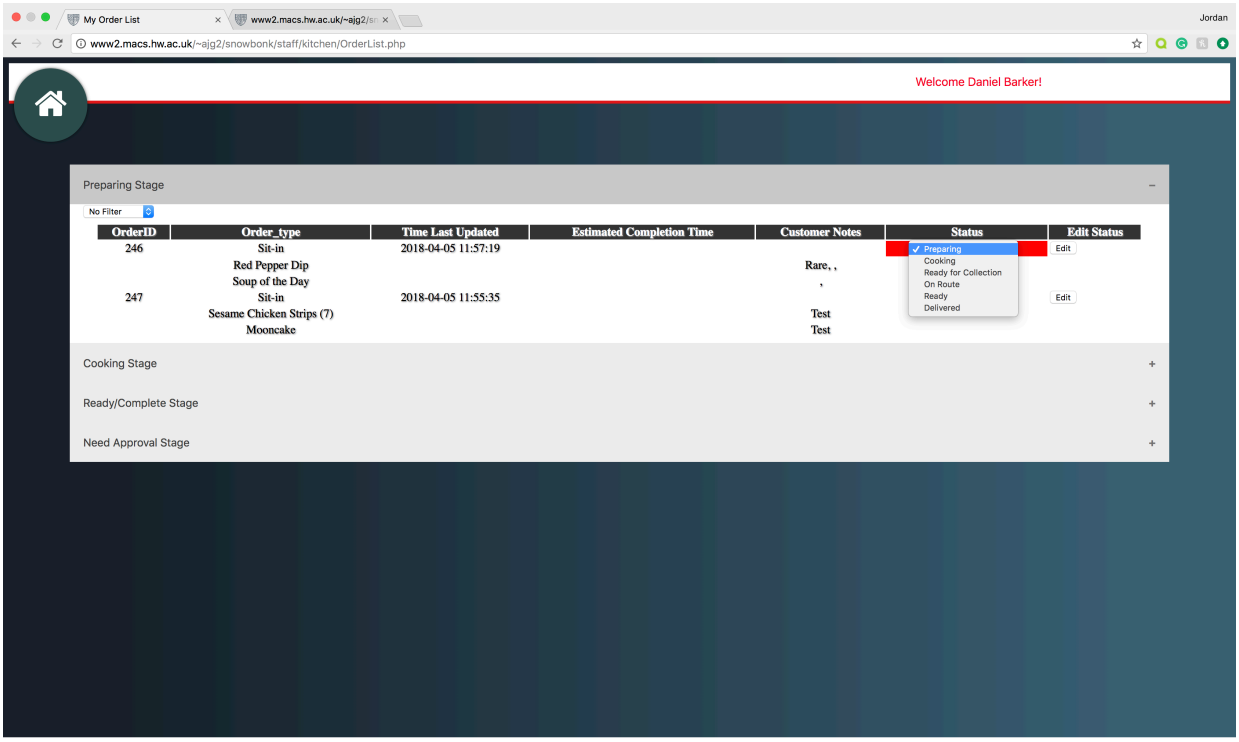
Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

All Users completed the task assigned and with ease and completed this task very quickly.

2.6 Can you change an Order from the "Preparing Stage" to the "Cooking Stage"

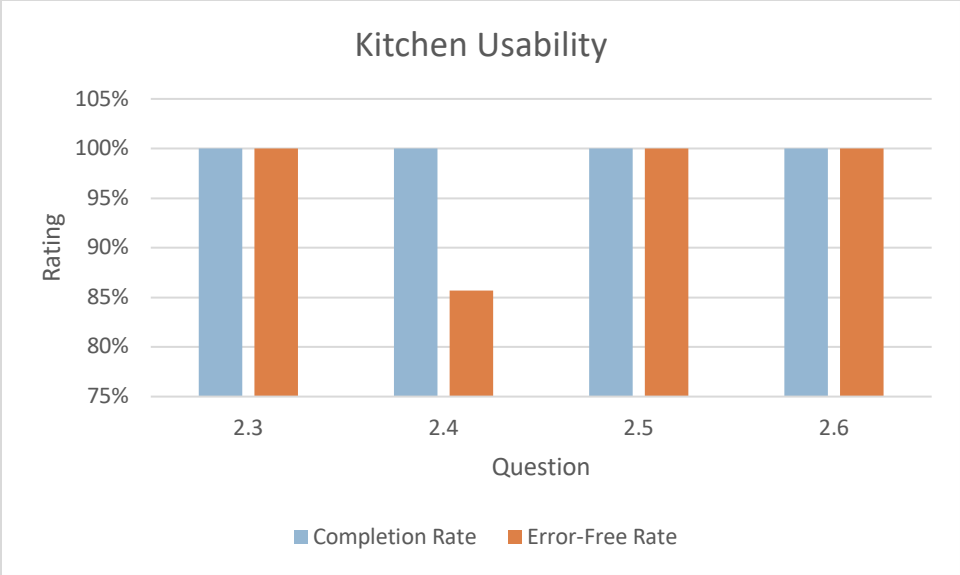
Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

All Users completed the task quickly.

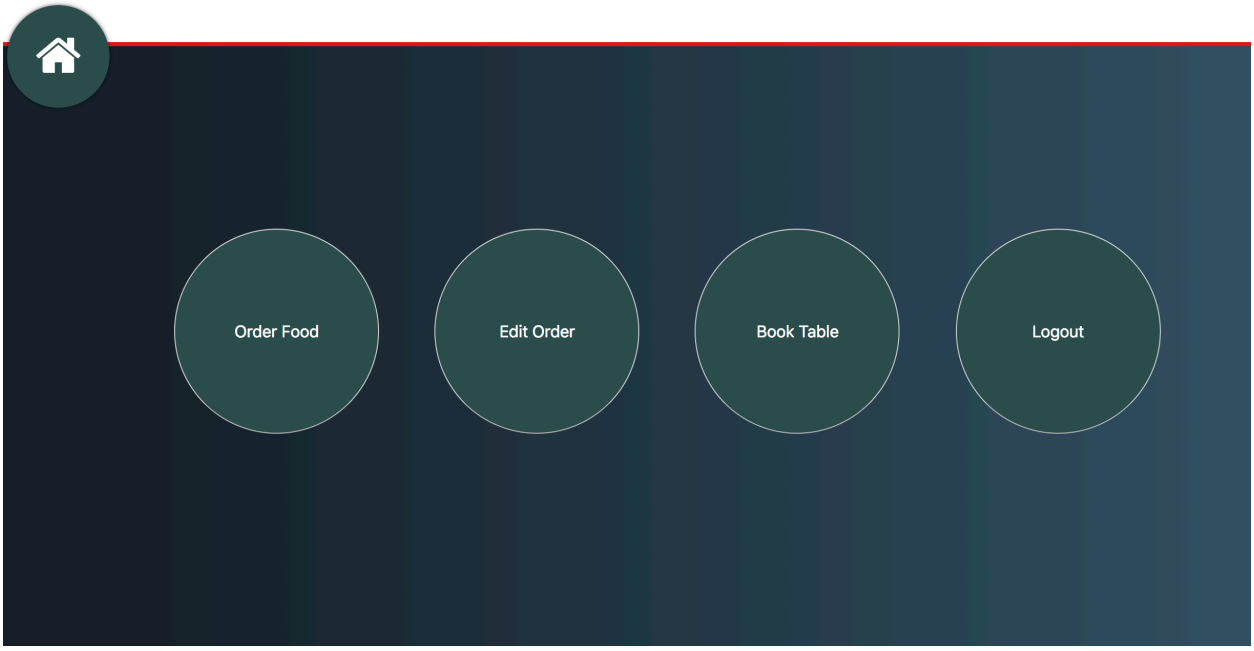


AREAS THAT NEED CHANGING

It became apparent that the design of this system was not up to the full specification of what we were trying to achieve. Moving forward from this if we were carrying forward this project we would change some of the design features of this system such as the filtering option to make it easier for the user to quickly access this section.

3 Waiter/ Waitress Order System

3.1 How would you describe this page?

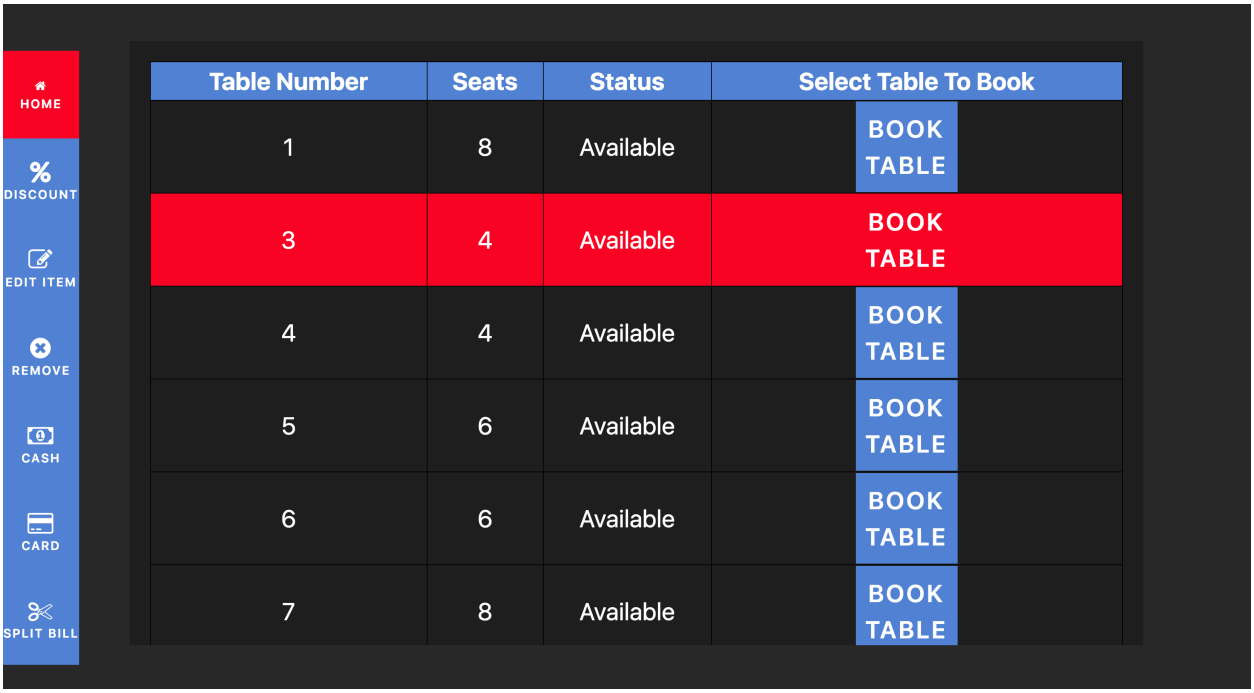


All Users identified this page as the selection for the waiter to access the different areas of the waiter system.

3.2 Can you book a Table for a customer?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

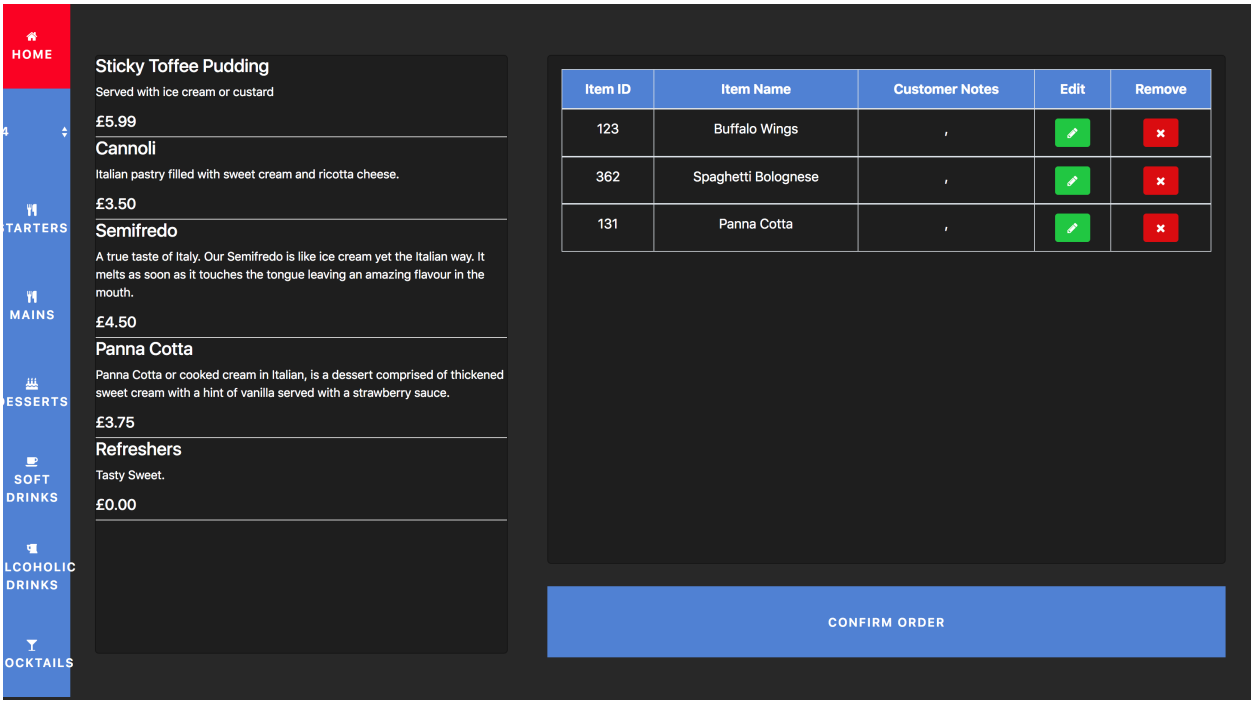
All Users completed the task assigned.

Users completed this task very quickly.

3.3 Can you create an order for the booked table?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



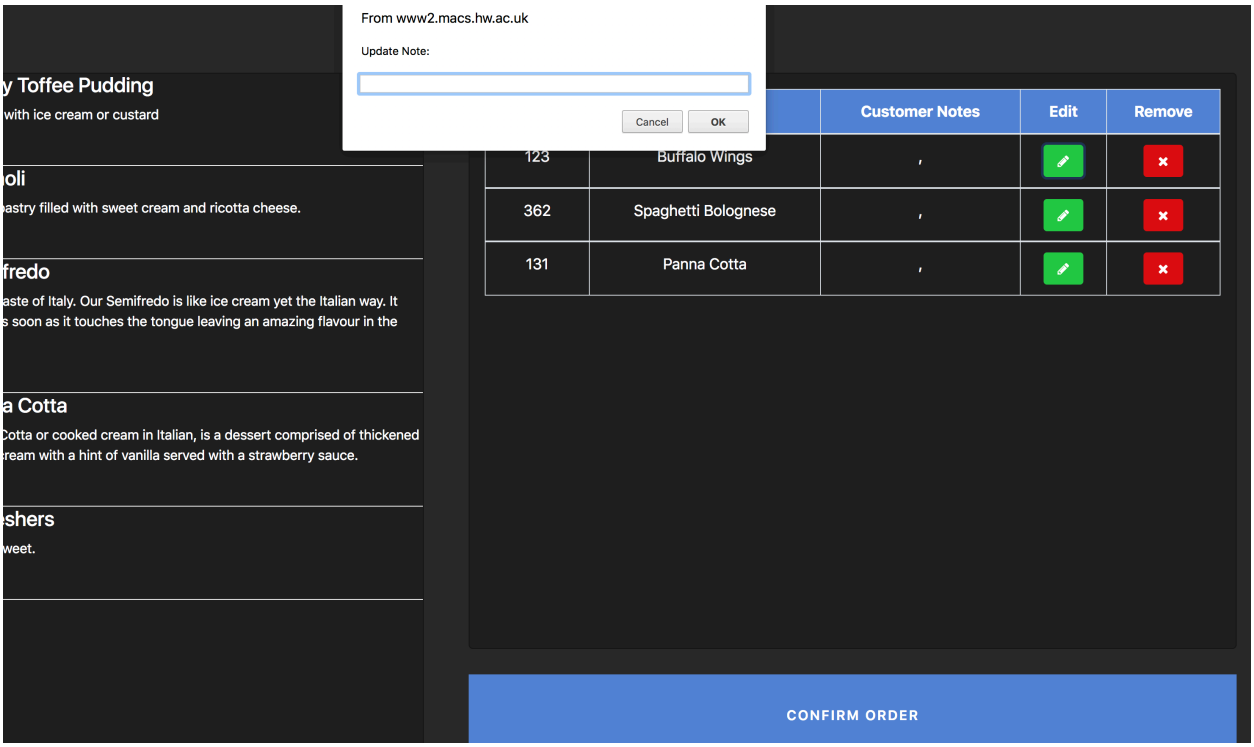
Couldn't Complete	0
Completed with minor difficulty	**1**
Completed with ease	**6**
Completion Rate	**100%**
Error-Free Rate	**85.7%**

All users completed the task however one was unsure of how to add items to the order and asked if there was a button to add the item itself.

3.4 Can you edit the order you created for the customer?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



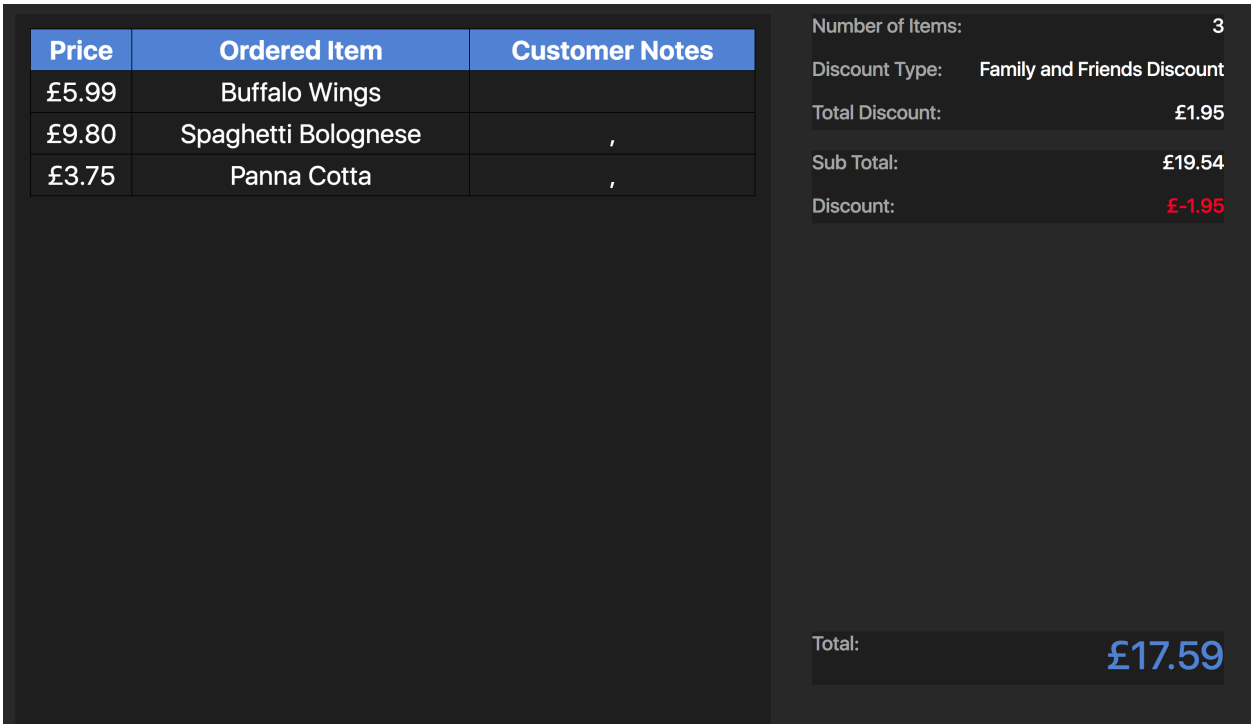
Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

Users quickly completed this task after adding an item to an order.

3.5 Can you complete a cash order for the table with a "friends and family discount"?

Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Price	Ordered Item	Customer Notes
£5.99	Buffalo Wings	
£9.80	Spaghetti Bolognese	,
£3.75	Panna Cotta	,

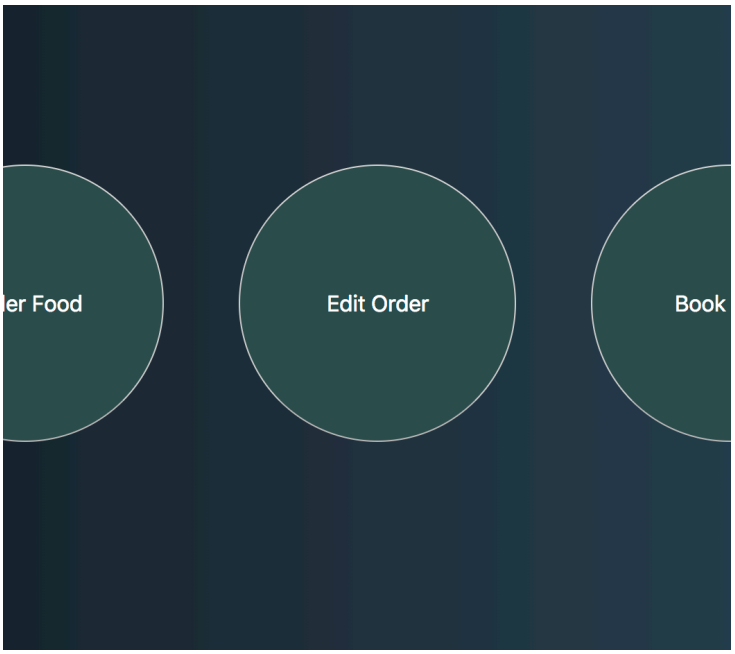
Couldn't Complete	0
Completed with minor difficulty	**2**
Completed with ease	**5**
Completion Rate	**100%**
Error-Free Rate	**71.4%**

Two users were unsure of how to create a payment for the order an looked for an option of payment.

3.6 Can you go back to the start screen and then amend an order?

Completion Rate:

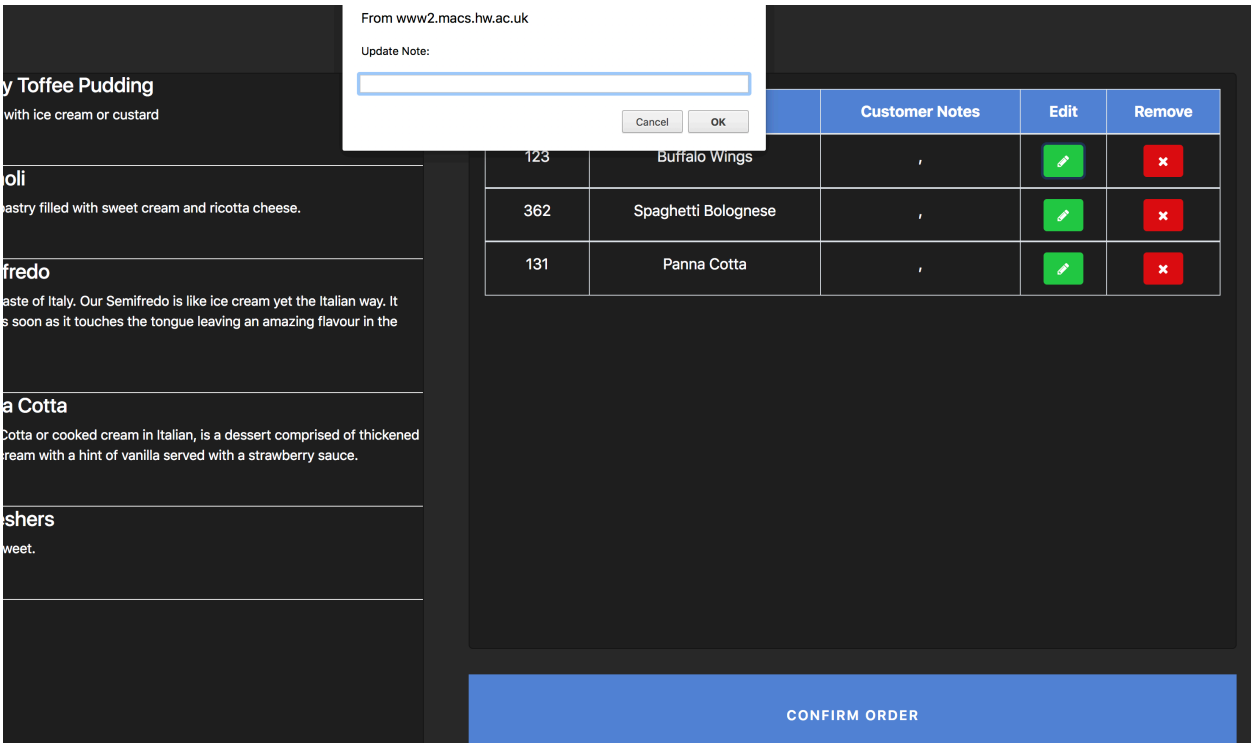
0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**3**
Completed with ease	**4**
Completion Rate	**100%**
Error-Free Rate	**57.1%**

This task posed some difficulty with the users and assistance had to be given by the invigilators of the test. This was mainly due to the user not knowing to continue with the prompt "Update Order" and thought they were drastically changing the order itself.

3.7 Can you Describe the page?

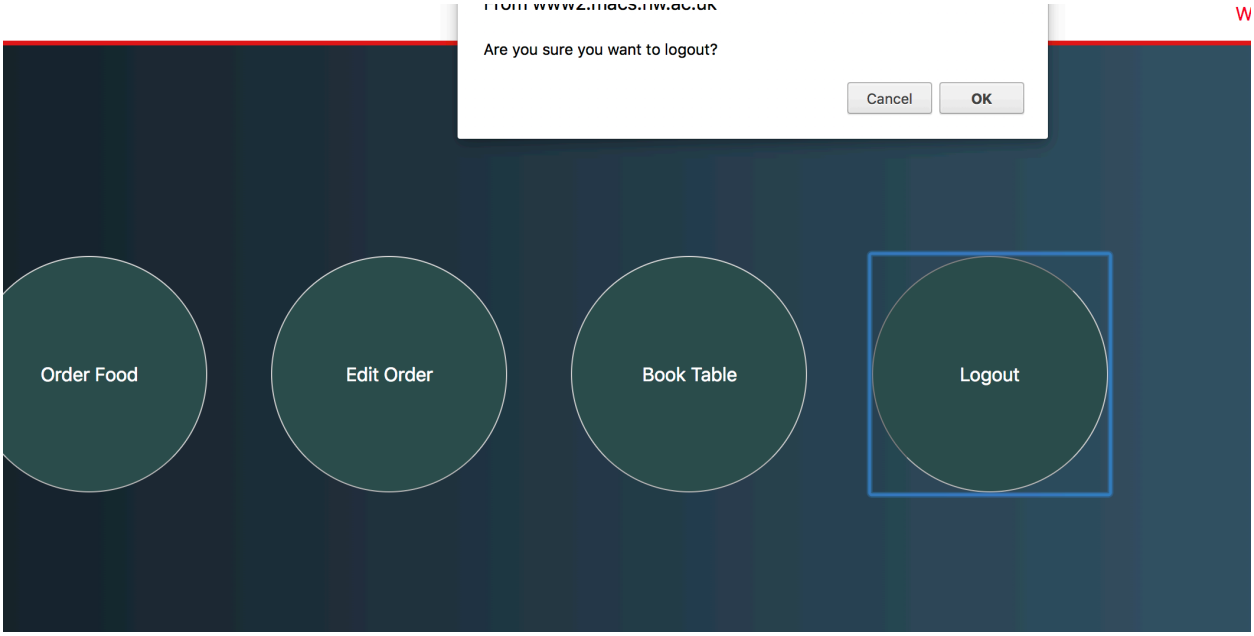


All users correctly identified this as the ordering page of the waiter application and correctly identified what each of the buttons did for the staff.

3.8 Can you now log out the system?

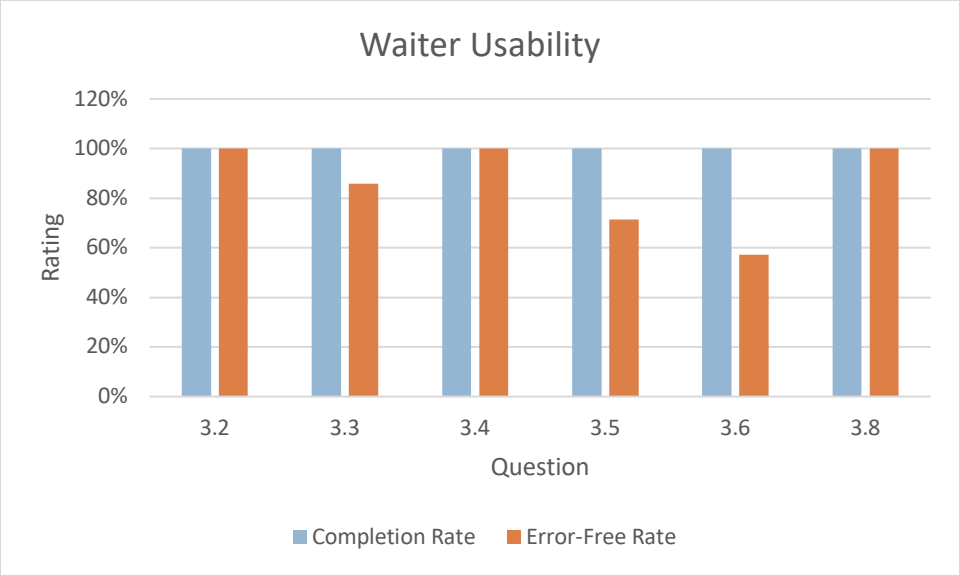
Completion Rate:

0. Couldn't Complete 1. Completed with minor difficulty 2. Completed with ease.



Couldn't Complete	0
Completed with minor difficulty	**0**
Completed with ease	**7**
Completion Rate	**100%**
Error-Free Rate	**100%**

All users quickly completed the task of logging out of the system.

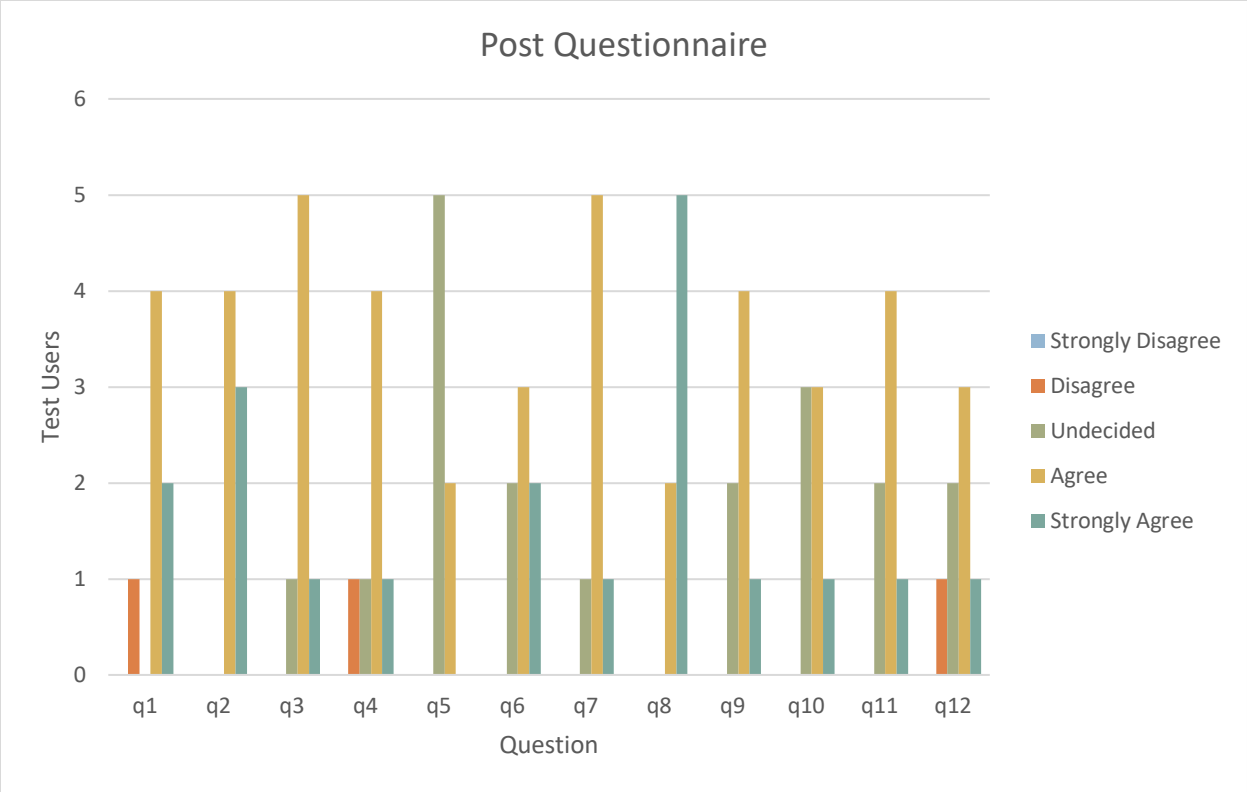


AREAS THAT NEED CHANGING

Certain features of the Waiter application would be changed if the project was to be continued. This primarily resolves around how users navigate the different areas of functionality especially regarding making an edit to an order after it has been confirmed.

POST- QUESTIONNAIRE

	Strongly Disagree	**Disagree**	**Undecided**	**Agree**	**Strongly Agree**
Q1) The System was easy to use	0	1	0	4	2
Q2) The design and layout of the system was clear and understandable	0	0	0	4	3
Q3) The Systems were User Friendly	0	0	1	5	1
Q4) Completing tasks was logical and straightforward	0	1	1	4	1
Q5) Information on pages was suitable and easily understandable	0	0	5	2	0
Q6) Icons and Buttons were well placed	0	0	2	3	2
Q7) The navigation through the systems was logical	0	0	1	5	1
Q8) The Systems were well presented and designed	0	0	0	2	5
Q9) I was satisfied with the system and its services	0	0	2	4	1
Q10) I would recommend the system to other users	0	0	3	3	1
Q11) I would use this system to order food	0	0	2	4	1
Q12) I would use this system to book a table	0	1	2	3	1



Q13) What features did you find best about the systems?

Feedback for the final usability study was generally positive regarding what users found best about the systems. Most users cited the design as one of the best parts of the application saying that it gave a familiar feel especially when concerning the Customer Application.

Other feedback highlighted the waiter system as being very realistic and usable even considering the mild challenges some faced with completing tasks.

Q14) What features would you change about the Systems?

This section was fairly vague from the feedback responses but highlighted issues that were apparent in the Usability Test in the kitchen side such as making editing orders after completion easier.

One comment however stated was;

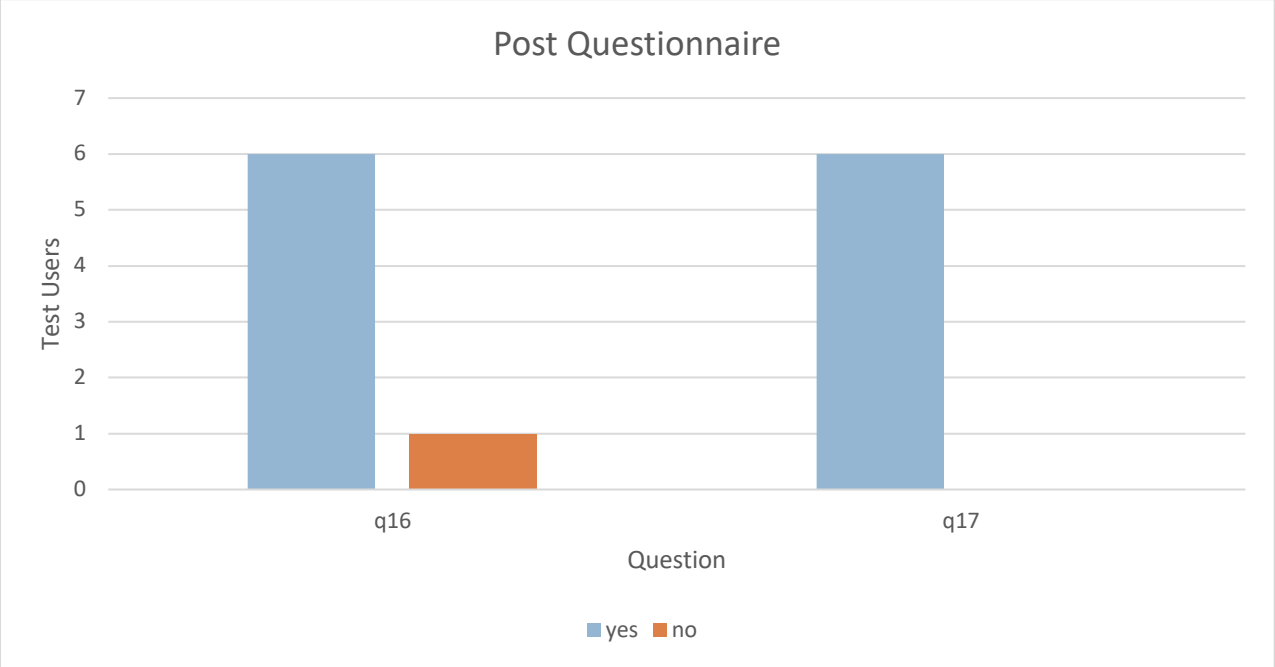
"Better design on kitchen system"

Q15) Do you have any more details about the Systems?

There was little details given back in this question but a varied of comments which included;

"Fully function application would be nice"

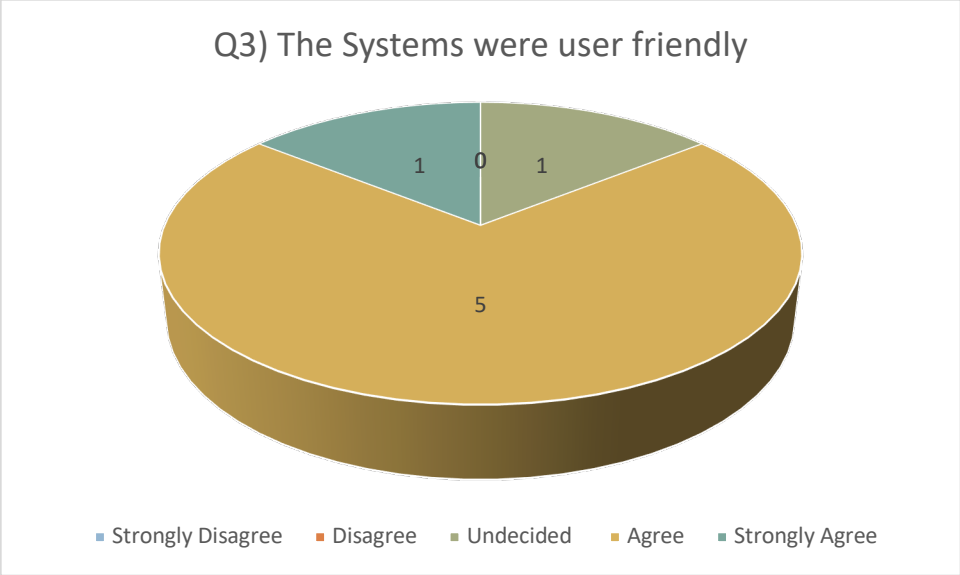
"Information on particular dishes and their ingredients for vegetarians/vegans would be nice"



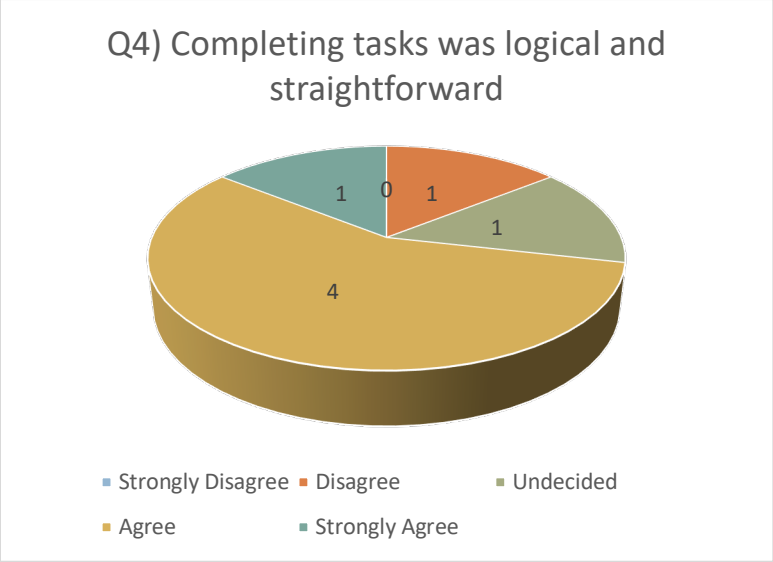
	Yes	No
Q16) Would you take part in another Usability Study?	6	1
Q17) would you mind testing on a further range of devices such as Tablets, PC's and Mobile?	6	0

CONCLUSION

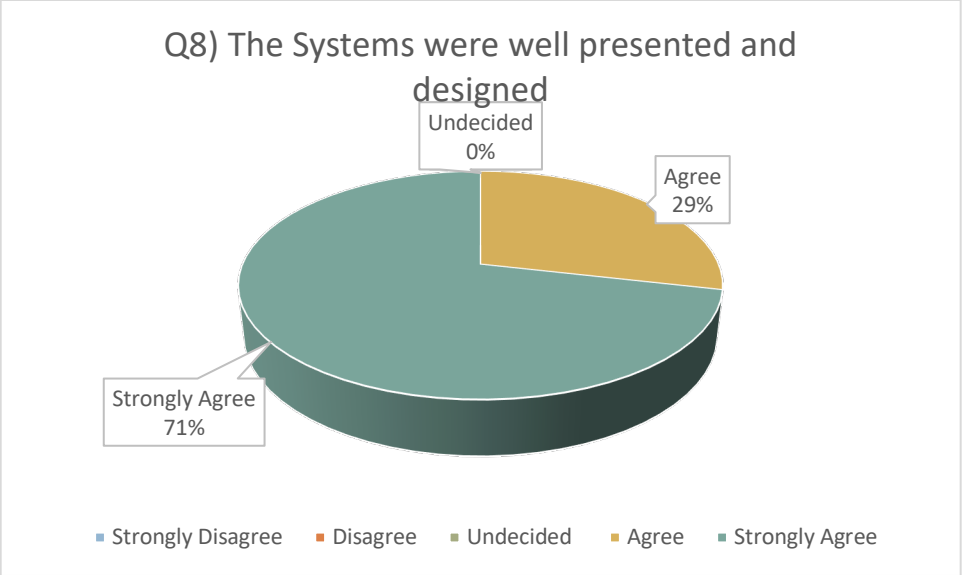
Generally in terms of a Usability study we received the responses we wanted from our test participants. This was further verified by having three users who have worked in a kitchen or as a waiter as they have had experience working with interfaces that would complete similar tasks to what we aimed to achieve during the course of this project.



One of the most anticipated portions of the usability study was to gauge at just how user friendly our application was. As the graph above shows the majority believed that the systems were user friendly which allows us to say we successfully implemented a working project that achieved one of its primary goals.



The graph on the left however shows that there was partial disagreement to how we approached the completion of tasks. As seen from the User Tests we could of improved some of the areas of design and functionality in the kitchen and waiter systems. However, the majority of users believed that the systems as a whole were created in a logical manner allowing for good user experience and usability.



As seen from the above graph when asked if the systems were well presented and designed the feedback from all users was over the "agree" option with the majority signalling that they "strongly believed" that the Systems were well presented and designed.
This is one of the highlights of the usability study and a great reinforcement that it was worthwhile spending time on both factors of design and functionality.

AREAS OF IMPROVEMENT

No project comes without its criticism and throughout the Usability study it is obvious that we have areas we could develop on. This mainly effects the waiter and kitchen system but as a whole the project suffered in some functionality and design by the time constraints posed on us from other factors.

Given more time many of these areas could have been developed more thoroughly and refined to make a more usable and more aesthetically pleasing application. If this project was to continue we would use this study like we did with the Stage One usability study to highlight areas that users may have problems with. However, as a final usability study on a project that is in its alpha state of prototype we as a group believe that we have ran a successful usability study and have created an efficient and practical solution to the specification asked of us.

APPENDIX

BIBLOGRAPHY

USABILITY REPORT
In-text: (Usability Report)
Your Bibliography: Usability Report. (n.d.). [ebook] Mandala Software Engineering. Available at: https://vision.hw.ac.uk/bbcswebdav/pid-2574211-dt-content-rid-2582381_2/courses/F29SO_2017-2018/GroupProjectExamplesEDDubai/Group7.pdf [Accessed 10 Nov. 2017].

CHUMBE, S.
Interactive Systems Principles for Apps
In-text: (Chumbe, 2015)
Your Bibliography: Chumbe, S. (2015). Interactive Systems Principles for Apps.
at: http://www.icbl.hw.ac.uk/santiago/teaching/F27IS/2015_2016/F27IS_lecture_3.pdf

Hastie, H.
Exploring Data
In-text: (Hastie, 2016)
Your Bibliography: Hastie, H. (2016). Exploring Data
at: https://vision.hw.ac.uk/bbcswebdav/pid-2335307-dt-content-rid-2180047_2/courses/F28IN_2016-2017/ED64Vision.pdf

Surveygizmo.com. (2018). What is a Likert Scale Survey Question & How to Use It. [online] Available at: https://www.surveygizmo.com/resources/blog/likert-scale-what-is-it-how-to-analyze-it-and-when-to-use-it/ [Accessed 4 Apr. 2018].

CONSENT FORM

Usability Test Consent Form for "SnowBonks" User Web App

Principal Investigators Include: *Jordan Walker, Owen Welch, Philip McQuaid, Antonio Gargaro, Craig Quint, Danial Barker, Calum McLean*

Consent to participate in an experimental study

The Purpose of this test is to measure and evaluate the usability of this web applications functionality.

Please read and sign this form.

In this usability test:

- You will be asked to complete a small number of questionnaire's involving questions around your age, occupation, preferences to food ordering applications, etc.
- You will also be asked to complete a short number of tasks using an abstract version of a food ordering application which will be assessed to gain information about our system.

Participation in this usability study is voluntary and all information will remain strictly confidential. The descriptions and findings may be used to help improve the "DineBros" application. However, at no time will your name or any other identification be used. You can withdraw your consent to the experiment and stop participation at any time.

If you have any questions after today, please contact Jordan Walker at jw31@hw.ac.uk.

I have read and understood the information on this form and had all my questions answered

_____ _____

Subject's Signature Researcher Signature

_____

Date

QUESTIONNAIRES

PRE-QUESTIONNAIRE

1)Currently what is your position?

- Professional ☐
- Student ☐
- Other ☐

2)Do you rent or own your own home?

- Yes ☐
- No ☐

3)Have you ever worked as a waiter or in a kitchen?

- Yes ☐
- No ☐

4)How often do you order food?

- Rarely (1-2 Times a month) ☐
- Often (1 – 2 Times every two weeks) ☐
- Regularly (1 – 2 Times a week) ☐

5)What days are you most likely to order food?

- Weekdays (Monday- Thursday) ☐
- Weekend (Friday- Sunday) ☐

6)Around What Time are you most likely to order food?

- Morning (8a.m – 12p.m) ☐
- Early Afternoon (12p.m – 5p.m) ☐
- Evening (5p.m – 9p.m) ☐
- Late Night (9p.m. 12a.m) ☐
- Early Morning (12a.m – 8a.m) ☐

7)How Often do you reserve a table at a restaurant?

- Rarely (1-2 Times a month) ☐
- Often (1 – 2 Times every two weeks) ☐
- Regularly (1 – 2 Times a week) ☐

8)What days are you most likely to book a table?

- Weekdays (Monday- Thursday) ☐
- Weekend (Friday- Sunday) ☐

9)When ordering food which device do you use most?

- Just-Eat ☐
- Hungry House ☐
- Deliveroo ☐
- Uber Eats ☐
- Other (Please Specify) ☐

10)Which Browser do you use most?

- Google Chrome ☐
- Mozilla FireFox ☐
- Safari ☐
- Internet Explorer ☐
- Opera ☐
- Other (Please Specify) ☐

11)Which Device are you most likely to order food from?

- Mobile Phone ☐
- Laptop ☐
- Ipad ☐
- Desktop Computer ☐
- Other (Please Specify) ☐

12) If using a mobile device, do you prefer ordering through an app or the mobile website?

- Weekdays (Monday- Thursday) ☐
- Weekend (Friday- Sunday) ☐

13) I would sign-up for a newsletter in order to hear about special offers and recommendations of the restaurants available on this system/website

- Yes ☐
- No ☐

14) Are you willing to try new restaurants based on previous customer satisfactory?

- Yes ☐
- No ☐

POST SYSTEM QUESTIONNAIRES

Participant Number

5	4	3	2	1
Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree

Q1_The Systems were easy to use

Q2_The design and layout of the systems was clear and understandable

Q3_The Systems were User Friendly

Q4_Completing tasks was logical and straightforward

Q5_Information on pages was suitable and easily understandable

Q6_Icons and Buttons were well placed

Q7_The navigation through the systems was logical

Q8_The Systems were well presented and designed

Q9_I was satisfied with the system and its services

Q10_I would recommend the system to other users

Q11_I would use this system to order food

Q12_I would use this system to book a table

Q13 What features did you find best about the systems?

Q14 What features would you change about the Systems?

Q15 Do you have any more details about the Systems?

Q16 Would you take part in another Usability Study?

Yes ☐
No ☐

Q17 If Yes would you mind testing on a further range of devices such as Tablets, PC's and Mobile?

Yes ☐
No ☐