

# Click & Eat Application Specification Document



30<sup>th</sup> April 2021

Bachelor Of Science (Honours) Software Development



(Student) - Ana Griga  
(Student Number) - C00231441  
(Tutor) - Dr. Chris Meudec

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE of  
TECHNOLOGY  
CARLOW

At the Heart of South Leinster

# Declaration

I declare that this specification document titled “Click & Eat” has been written by me under the supervision of Dr. Christophe Meudec.

This work was not presented in any previous research for the award of bachelor degree to the best of my knowledge.

The work is entirely mine and I accept the full responsibility for any errors that might be found in the work, while the reference to publish materials have been duly acknowledged.

I have provided a complete table of reference of all works and sources used in the preparation of this document.

I understand that failure to conform with the Institute’s regulations governing plagiarism represent a serious offence.

Signature: *Ana Griga*

Ana Griga (Student)

C00231441 (Student Number)

Date: 25.04.2021

## **Abstract**

For decades, the restaurant sector has undergone technological advancement to sustain its standard of service. Studies also suggested that the introduction of technology would have beneficial effects on the efficiency of operations.

The current study was carried out to design the Click & Eat application for the restaurants and for the customers and to identify and describe the functional and non functional specifications related to the proposed application.

The Click & Eat system mentioned in this document is designed to fill a particular void in the market by offering restaurants the opportunity to provide an online ordering and payment service to their customers with a less human interaction beneficial in these times of pandemic.

The customizable system allows restaurant staff to check the content of customers orders and payments. The administrators can also see the orders, payments, registered customers details and also they can update the menu items through a very intuitive graphical interface.

The aim of this specification document is to provide in-depth explanation and description of all available features of the Click & Eat application mainly through the understanding of the system's use cases.

# Table of Contents

<b>1. Introduction</b>	<b>7</b>
2. Application Description	<b>8</b>
2.1 User Application Structure	8
2.2 Application Context	9
<b>3. Functional Specifications</b>	<b>10</b>
3.1 Main Functionalities	10
3.2 Use Cases	10
3.2.1 Users	12
3.2.2 Detailed Use Cases	12
3.2.2.1 Order Detailed Use Case	13
3.2.2.2 Pay Detailed Use Case	14
3.2.2.3 Create Account Detailed Use Case	15
3.2.2.4 Send Feedback Use Case	16
3.2.2.6 Login Detailed Use Case	19
3.2.2.7 View Order Detailed Use Case	20
3.2.2.8 CRUD Menu Detailed Use Case	21
3.2.2.9 View Payment Detailed Use Case	23
3.2.2.10 View Customer Detailed Use Case	24
3.2.2.11 View Feedback Detailed Use Case	25
<b>4. Non-Functional Specifications (FURPS+)</b>	<b>26</b>
4.1 Functionality	26
4.2 Usability	26

4.3 Reliability	27
4.4 Performance	27
4.5 Supportability	27
4.6 Security	28
<b>5. Success Measurement</b>	<b>28</b>
<b>6. Testing</b>	<b>29</b>
6.1 Functionality Testing	29
6.2 Usability Testing	30
6.3 Compatibility Testing	30
6.4 Performance Testing	30
6.5 Security Testing	30
6.6 Interface Testing	30
<b>7. Conclusion</b>	<b>31</b>

## Table of Figures

Figure 1 - Click & Eat Context Diagram.....	9
Figure 2 - Click & Eat Application Structure.....	9
Figure 3 - Click & Eat Use Case Diagram for the Customers Module.....	11
Figure 4 - Click & Eat Use Case Diagram for the Restaurant Module.....	12

## Table of Tables

Table 1 - Order Use Case.....	13
Table 2 - Pay Use Case.....	14
Table 3 - Create Account Use Case.....	15
Table 4 - Send Feedback Use Case.....	16
Table 5 - CRUD Account Use Case.....	18
Table 6 - Login Use Case.....	19
Table 7 - View Order Use Case.....	20
Table 8 - CRUD Menu Use Case.....	22
Table 9 - View Payment Use Case.....	23
Table 10 - View Customer Use Case.....	24
Table 11 - View Feedback Use Case.....	25

# 1. Introduction

Current pandemic conditions have had a dramatic, negative financial impact on the restaurant industry. For several causes like unemployment, and general uncertainty of what the future holds, the panic of the pandemic, restaurant customer behavior habits have been considerably altered, and restaurateurs would need to look at different strategies to establish and recover clients. The face of technological advancement is constantly evolving. In introducing technological innovations, the restaurant sector is taking a constructive stand by constantly trying to create standards of service quality and guest loyalty.

Cost and time savings are necessary in order to operate a restaurant effectively. For each table, reducing time by a few seconds will speed up order, increase productivity and increase income. The proposed applications aim to improve table turn-times and reduce the cost of staff which will provide more time enabling great hospitality.

Click & Eat will greatly simplify the ordering and payment of the meals for both customers and the restaurant.

Given the COVID19 restrictions and social distancing, this project intends to develop a system that offers the customers a safer dinner experience with less contact with the staff. With this application, the guest will be able to browse the menu, place their order and pay for their meal using their device. There will be no menus, bills or pens to touch and no waiting for a waiter.

The system presents an interactive menu with all available options in an easy to use manner. Customers can choose one or more items to place an order which will land in the Cart. Customers can view all the order details in the cart before checking out. In the end, the customer gets the order confirmation details and the payment option. Restaurant staff will easily go through the orders as they are received and with minimum delays and uncertainty, process all orders efficiently and effectively.

This document describes the application, the application' s structure and specifies in detail the functions that the Click & Eat application will perform.



## **2. Application Description**

Click & Eat is a web application designed for the restaurant industry which allows customers to order and pay for their meal from their devices and which brings the benefit of cost staff reduction and improved table turn times for the restaurant.

With this solution, the customers can browse the menu, order and pay using their device without the need of downloading an app or registering for an account.

This application enables customers to take the ordering and payment process into their own hands. They will be presented with a QR code placed on each table which will be scanned using their mobile phone camera to access a digital menu. They will browse the menu to craft their order, submit their order and pay for their meal all while maintaining minimum customer-staff contact.

This application will also help restaurants in their ability to offer customers the experience of dining out with less interaction, due to the pandemic restrictions and everyone's safety.

The restaurants can see the benefits of the increased spend through easy upsells, the ability to send special offer notifications to their customers once they leave, reduced staff cost as the customers self-serve by ordering and paying on their own and increase the number of diners as the experience is safer due to reduced interaction.

### **2.1 User Application Structure**

From the usage of the application perspective, the structure of the system can be divided into two main logical components as shown in Figure 1. The first component resides in the restaurant part of the application and will provide retrieving of orders and payments, login and menu management for the staff which include the waiters and the administrator. The menu management includes the CRUD operations for the menu items which will allow the restaurant to monitor what can be ordered by customers.

The second component is the ordering and payment part of the application which will be used by the customers and which will provide the functionality to view the food items, place their order, pay for their order, register if they wish and provide feedback.

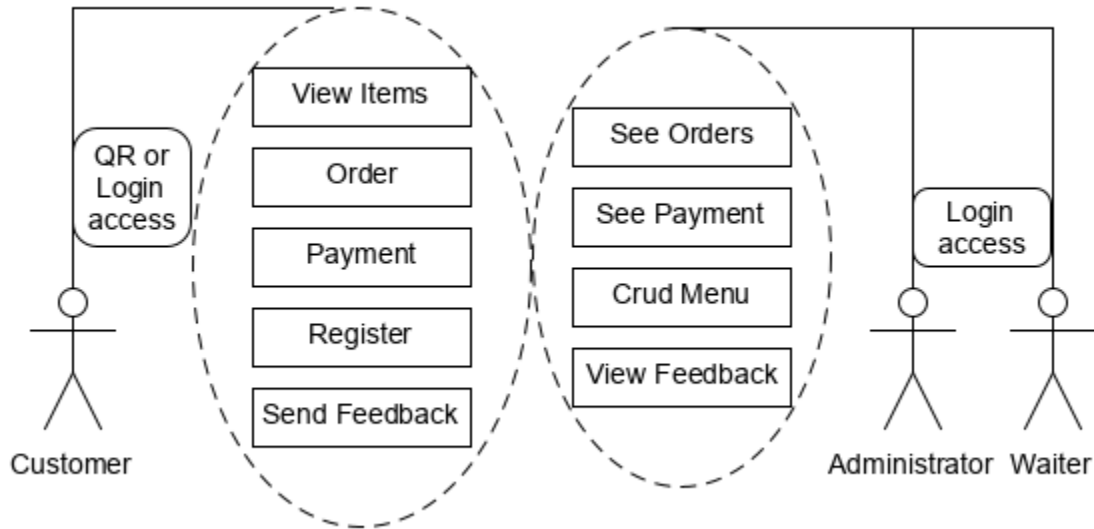


Figure 1 - User Application Structure

## 2.2 Application Context

The figure below shows the context diagram of the system which represents the high level of the application. It defines the boundary between the system and its environment and it shows the entities that the application interacts with. For Click & Eat the entities are the customers, the waiters, the administrator and a payment service in the backend.

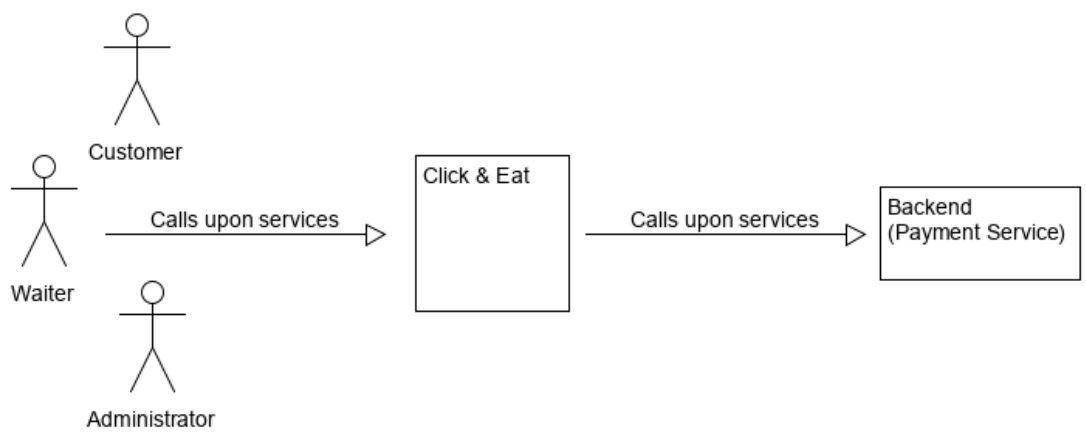


Figure 2 - Click & Eat Context Diagram

## **3. Functional Specifications**

### **3.1 Main Functionalities**

There are specific requirements the proposed application must fulfil to meet the objectives of the project. The requirements to be achieved:

The main functionalities for the customer's application module are as follows:

- Place the order
- Review the order
- Pay for the order
- Create account
- Send feedback

The module for the Restaurant's staff is designed to provide the following functions:

- Staff Login
- Retrieve the orders
- CRUD menu items
- Retrieve payments
- View customers
- View customer's feedback

### **3.2 Use Cases**

A use case is a series of scenarios that characterize a user's interaction with a device. The relationship between actors and use cases is shown in a use case diagram. Use cases and actors are the principal components of a use case diagram.

Functional requirements are listed depending on their relationship to the overall system, customers, waiters, administrators. Using a natural language description applying UML Analysis Models, the functional requirements will be specified.

The figure below presents the use case diagrams for the Click & Eat application. The key interactions between the users and the system itself are represented.

The use cases are presented in two diagrams, one for the customer module and the second one for the restaurant staff module which include waiter and administrator actors.

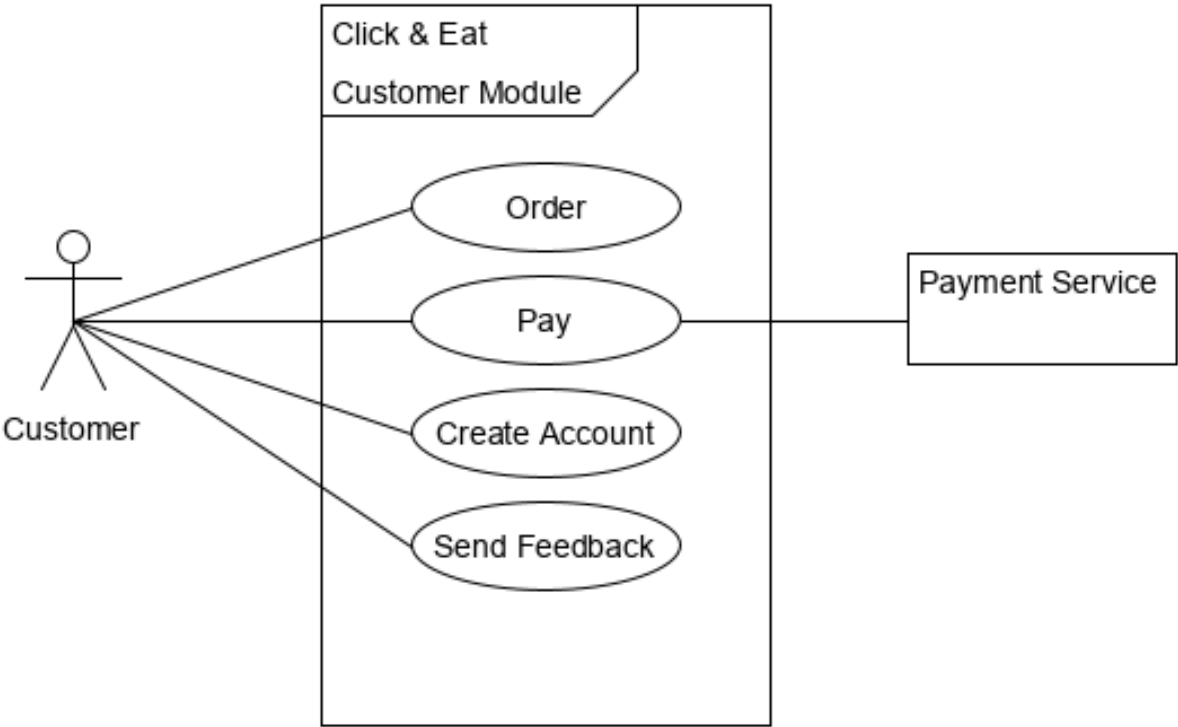


Figure 3 - Click & Eat Use Case Diagram for the Customers Module

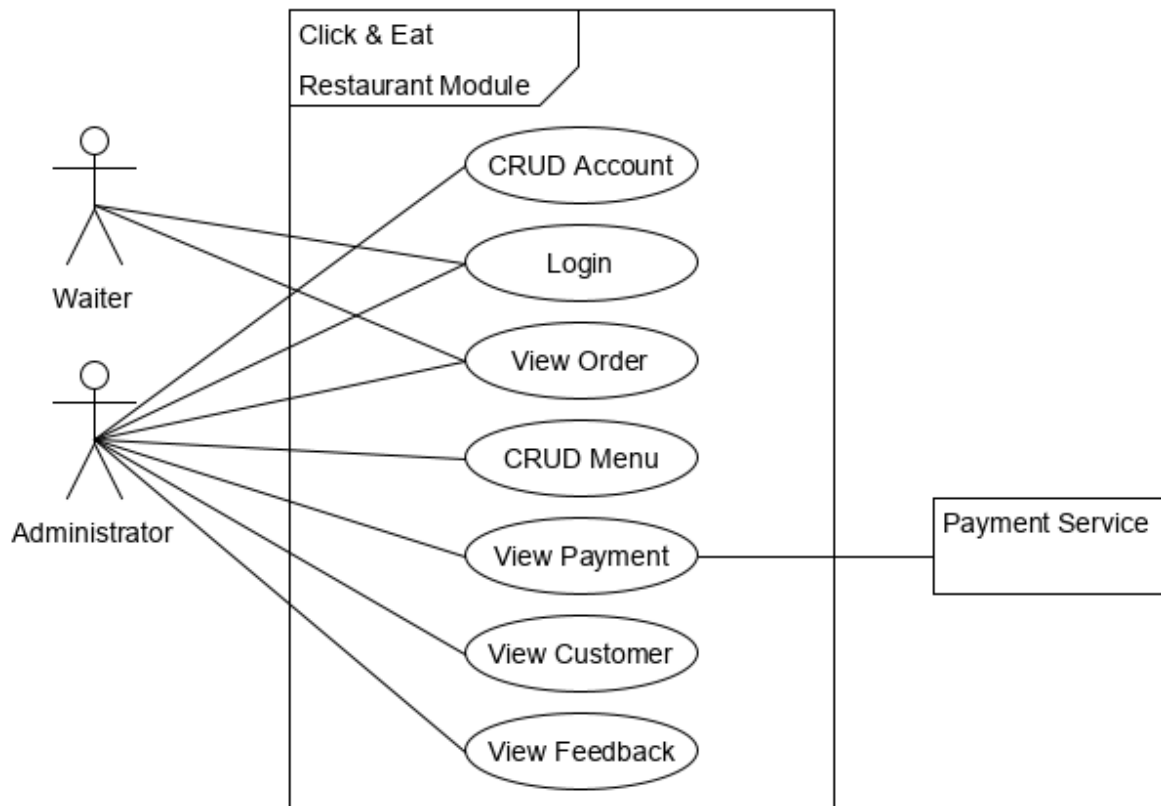


Figure 4 - Click & Eat Use Case Diagram for the Restaurant Module

### 3.2.1 Users

The Click & Eat application has two main target users to which it must satisfy, restaurant's customers and restaurant's staff (waiters and administrator/restaurant manager or restaurant owner), both of which have roles that are mutually exclusive from one another, although some of their functionality will cross over.

### 3.2.2 Detailed Use Cases

The detailed uses cases will describe how all the users will perform tasks on the Click & Eat application. It will outline the application behaviour, from the user's perspective and how it responds to a request. Each use case is represented below as a sequence of simple steps and will contain the actors, a description of the use case, postconditions, preconditions and alternatives.

### 3.2.2.1 Order Detailed Use Case

<b>Name</b>	<b>Order</b>
<b>Actors</b>	Customer
<b>Description</b>	This use case describes the process of placing the order by the customer. This is done by browsing the restaurant's menu, selecting the items they wish to purchase and submitting the order.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The customer must scan the QR code provided by the restaurant or Login</li> <li>2. The QR or the Login must reveal the restaurant menu page</li> <li>3. The restaurant's menu page should present all the menu items</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The customer browse the menu and selects the menu items they wish to purchase</li> <li>2. The customer adds the selected menu items into the order cart</li> <li>3. The application presents to the customer a preview order option.</li> <li>4. The customer confirms the order by selecting the Proceed to checkout button</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The order has been made and sent to the restaurant module of the application where the restaurant staff can see it and place it for processing</li> <li>2. When the order has been filled, the restaurant staff can checked it as Served</li> <li>3. The application is ready for another order at any time if the customer wants to place another order</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>4a. The customer wishes to add more to the current order <ol style="list-style-type: none"> <li>1. The customer is presented with a Place Order &amp; Pay Later option</li> <li>2. The customer selects the add more option</li> <li>3. The application shows the menu</li> <li>4. The customer selects the new item they want to add to the current order</li> <li>5. The customer submits the new order</li> </ol> </li> <li>4b. The customer added unwanted items to the order <ol style="list-style-type: none"> <li>1. The customer is presented with a current order</li> <li>2. The customer can decrease the quantity of any of the items</li> <li>3. The customer submit the edited order</li> </ol> </li> </ol>

Table 1 - Order Use Case

### 3.2.2.2 Pay Detailed Use Case

<b>Name</b>	<b>Pay</b>
<b>Actors</b>	Customer
<b>Description</b>	This use case describes the process of paying for the order placed by the customer and it starts when the order is completed and ready for payment.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The order has been confirmed and the total bill has been displayed to the customer</li> <li>2. The application presents to the customer the pay for order option</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The customer enters the details for payment</li> <li>2. The application presents the add tip option</li> <li>3. The customer adds a tip if they wish too</li> <li>4. The application recalculates the total to pay</li> <li>5. The customer confirm the total and press pay now option</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The payments was successful</li> <li>2. The application display a thank you message for payment completion</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>3a. The customer wishes to modify the amount for the tip <ol style="list-style-type: none"> <li>1. The customer is presented with a review and pay order option</li> <li>2. The customer selects the edit option</li> <li>3. The application displays the pay order in an editable mode</li> <li>4. The customer edits the tip amount</li> <li>5. The customer confirm the total and press pay now option</li> </ol> </li> </ol>

Table 2 - Pay Use Case

### 3.2.2.3 Create Account Detailed Use Case

<b>Name</b>	<b>Create Account</b>
<b>Actors</b>	Customer
<b>Description</b>	This use case begins when the customer has already made the payment for the order and is presented with the create account option which is non-compulsory.
<b>Preconditions</b>	1. The customer has successfully completed the payment of the order
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The create account form asks the customer for a username, email, password</li> <li>2. The create account asks to re enter the chosen password</li> <li>3. The customer enters the required details</li> <li>4. The form will display any errors</li> <li>5. The customer will click create account for submission</li> <li>6. The application will store the customer's information for future activity</li> </ol>
<b>Postconditions</b>	1. The user has successfully registered their information with the application
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a. The customer enters an invalid email address <ol style="list-style-type: none"> <li>1. The customer is prompted to enter a valid email</li> </ol> </li> <li>1b. The customer enters a password that is too short <ol style="list-style-type: none"> <li>1. The customer is prompted to enter a password of sufficient length</li> </ol> </li> <li>2a. The passwords entered do not match <ol style="list-style-type: none"> <li>1. The customer is asked to enter matching passwords</li> </ol> </li> </ol>

Table 3 - Create Account Use Case



### 3.2.2.4 Send Feedback Use Case

<b>Name</b>	<b>Send Feedback</b>
<b>Actors</b>	Customer
<b>Description</b>	This use case occurs when the customer wishes to comment on the overall experience. This option will be available to the customer after the payment of the order was submitted.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The customer has successfully completed the payment of the order</li> <li>2. The restaurant staff checks the order as Served, when the order is filled</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The customer is presented with the feedback form</li> <li>2. The form requires the customer's name</li> <li>3. The customer fills in their name</li> <li>4. The customer writes the feedback in the presented box</li> <li>5. The customer submits the feedback</li> <li>6. The application saves the feedback</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The customer sends successfully a feedback</li> <li>2. The feedback is saved and the restaurant staff and other customers can see it</li> </ol>
<b>Alternatives</b>	N/A

Table 4 - Send Feedback Use Case

### 3.2.2.5 CRUD Account Detailed Use Case

<b>Name</b>	<b>CRUD Account</b>
<b>Actors</b>	Administrator, Waiter
<b>Description</b>	This use case begins when the administrator needs to create a new account or view, edit or delete an existing account. The accounts created will be for the waiters.
<b>Preconditions</b>	<p><b>Create Account</b></p> <ol style="list-style-type: none"> <li>1. To create a new account the administrator must create their own account and log in</li> <li>2. To create a new account, the account holder needs to be present to get their information details</li> </ol> <p><b>View Account, Edit Account, Delete Account</b></p> <ol style="list-style-type: none"> <li>1. To view, edit, delete an existing account the administrator must be logged in</li> </ol>
<b>Activity</b>	<p><b>Create Account</b></p> <ol style="list-style-type: none"> <li>1. For creating a new account the application will ask for the account holder information (waiter): username, email, password</li> <li>2. The waiter will enter their name, email, password</li> <li>3. The create account form asks to re enter the chosen password</li> <li>4. The waiter re-enters the password</li> <li>5. The form will display any fields errors if any</li> <li>6. The waiter will click create account for submission</li> <li>7. The application will store the new account information for future login</li> <li>8. The application will display a create account confirmation message</li> </ol> <p><b>View Account</b></p> <ol style="list-style-type: none"> <li>1. The application will display all accounts</li> <li>2. The administrator will click on view account for the account they want to view</li> <li>3. The application will display the account details</li> </ol> <p><b>Edit Account</b></p> <ol style="list-style-type: none"> <li>1. The application will display all accounts</li> <li>2. The administrator will choose the account they wish to edit</li> <li>3. The application will display the account details</li> <li>4. The application will display an edit account option</li> <li>5. The administrator will choose edit account option</li> <li>6. The administrator will edit the specific fields of the account</li> <li>7. The administrator will click submit changes</li> </ol>

	<ol style="list-style-type: none"> <li>8. The application will save the changes</li> <li>9. The application will display an account edit confirmation message</li> </ol> <p><b>Delete Account</b></p> <ol style="list-style-type: none"> <li>1. The application will display all accounts</li> <li>2. The administrator will choose the account they wish to delete</li> <li>3. The application will display a delete account option</li> <li>4. The administrator will click delete option</li> <li>5. The application will ask to confirm deletion of the account</li> <li>6. The administrator will confirm the deletion of the account</li> <li>7. The application will delete the account</li> <li>8. The application will display a delete confirmation message</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The new account was successfully created</li> <li>2. An existing account was viewed</li> <li>3. An existing account was successfully edited</li> <li>4. An existing account was successfully deleted</li> </ol>
<b>Alternatives</b>	<p><b>Create Account</b></p> <ol style="list-style-type: none"> <li>2a. The waiter enters an invalid email address       <ol style="list-style-type: none"> <li>1. The waiter is prompted to enter a valid email</li> </ol> </li> <li>2b. The customer enters a password that is too short       <ol style="list-style-type: none"> <li>1. The customer is prompted to enter a password of sufficient length</li> </ol> </li> <li>4a. The passwords entered do not match       <ol style="list-style-type: none"> <li>1. The customer is asked to enter matching passwords</li> </ol> </li> </ol>

Table 5 - CRUD Account Use Case

### 3.2.2.6 Login Detailed Use Case

<b>Name</b>	<b>Login</b>
<b>Actors</b>	Administrator, Waiter
<b>Description</b>	This use case begins when the administrator or the waiter opens the application and wants to login so they can access all the features of the application.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The application will display a login option</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The application will ask for their username and password</li> <li>2. The administrator or waiter will enter their username and password</li> <li>3. The administrator or waiter will submit the login details</li> <li>4. The application will check and validate the details</li> <li>5. The application will redirect the administrator or the waiter to the home screen</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The administrator or the waiter logged in successfully</li> <li>2. The administrator or waiter can access all features of the application</li> </ol>
<b>Alternatives</b>	<p>2a. The administrator or the waiter enters their username or password incorrectly</p> <ol style="list-style-type: none"> <li>1. The application will ask to re enter username and password</li> </ol> <p>2b. The administrator or the waiter enters their username or password incorrectly 3 times</p> <ol style="list-style-type: none"> <li>1. The administrator or the waiter will be locked out of their account temporarily for a certain amount of time or until they verify their details via email.</li> </ol>

Table 6 - Login Use Case

### 3.2.2.7 View Order Detailed Use Case

<b>Name</b>	<b>View Order</b>
<b>Actors</b>	Administrator, Waiter
<b>Description</b>	This use case describes the actions the waiter or administrator needs to do in order to view the details of a placed order.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The administrator or waiter needs to be logged in</li> <li>2. The application will have an order option in the Management menu</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The administrator or waiter selects order option</li> <li>2. The application will display the orders on the screen</li> <li>3. The administrator or waiter can see all the details of the order</li> <li>4. The administrator or waiter can mark the order as Served, when the order is completed</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The administrator or waiter can access the view order feature of the application</li> </ol>
<b>Alternatives</b>	N/A

Table 7 - View Order Use Case

### 3.2.2.8 CRUD Menu Detailed Use Case

<b>Name</b>	<b>CRUD Menu</b>
<b>Actors</b>	Administrator
<b>Description</b>	This use case begins when the administrator needs to create a new menu item or menu category or view, edit or delete an existing one.
<b>Preconditions</b>	<p><b>Create Menu Item/Category</b></p> <ol style="list-style-type: none"> <li>1. The administrator must be logged in</li> <li>2. The administrator must access the menu page of the application</li> </ol> <p><b>View, Edit, Delete Menu Item/Category</b></p> <ol style="list-style-type: none"> <li>1. The administrator must be logged in</li> <li>2. The administrator must access the menu page of the application</li> <li>3. A menu item/category already exists in the application in some particular format</li> </ol>
<b>Activity</b>	<p><b>Create New Menu Item/Category</b></p> <ol style="list-style-type: none"> <li>1. The application will display add new item or add new category option</li> <li>2. The administrator will select the desired option</li> <li>3. Based on the selected option the application will display a new form to be filled in</li> <li>4. The administrator starts filling in the details for the new menu item or menu category</li> <li>5. The administrator will submit the details for the newly created menu item or menu category</li> <li>6. The application will submit and save the details</li> </ol> <p><b>View Menu Item</b></p> <ol style="list-style-type: none"> <li>1. The application will display all the menu items/categories</li> <li>2. The administrator will select a menu item/category they wish to view</li> <li>3. The application will display the selected menu item/category details</li> <li>4. The administrator is able to see the selected menu item/category</li> </ol> <p><b>Edit Menu Item</b></p> <ol style="list-style-type: none"> <li>1. The application will display all the menu items/categories</li> <li>2. The administrator will select a menu item/category they wish to edit</li> <li>3. The application will display the selected menu item/category details</li> <li>4. The administrator will make the desired changes</li> <li>5. The administrator will submit the changes</li> <li>6. The application will save the changes</li> </ol> <p><b>Delete Menu Item</b></p> <ol style="list-style-type: none"> <li>1. The application will display all the menu items/categories</li> </ol>

	<ol style="list-style-type: none"> <li>2. The administrator will select a menu item/category they wish to delete</li> <li>3. The application will display the selected menu item/category details</li> <li>4. The administrator will choose the delete option for the selected menu item/category</li> <li>5. The application will delete the menu item/category</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The administrator is able to create a new menu item/category</li> <li>2. The administrator can view, edit and delete an existing menu item/category</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>4a. Administrator selects Cancel option <ol style="list-style-type: none"> <li>1. The application does not add the menu item to the item list.</li> <li>2. It is redirected to previous screen</li> </ol> </li> <li>5a. The application detects a letter in price data field <ol style="list-style-type: none"> <li>1. The application notifies the administrator that data field is invalid or missing</li> </ol> </li> <li>5b. The application detects a data field is empty <ol style="list-style-type: none"> <li>1. The application notifies the administrator that data field is invalid or missing</li> </ol> </li> </ol>

Table 8 - CRUD Menu Use Case

### 3.2.2.9 View Payment Detailed Use Case

<b>Name</b>	<b>View Payment</b>
<b>Actors</b>	Administrator, Waiter
<b>Description</b>	This use case describes the actions the administrator or the waiter needs to do in order to view the details of a payment.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. The administrator or waiter needs to be logged in</li><li>2. The application will display the Order option</li></ol>
<b>Activity</b>	<ol style="list-style-type: none"><li>1. The administrator or waiter will open the Order option</li><li>2. The application will display the order details including the payment</li><li>3. The administrator or waiter can see the if the order is paid or not</li></ol>
<b>Postconditions</b>	<ol style="list-style-type: none"><li>1. The administrator and waiter are able to see the details of an order if is paid or not</li></ol>
<b>Alternatives</b>	N/A

Table 9 - View Payment Use Case



### 3.2.2.10 View Customer Detailed Use Case

<b>Name</b>	<b>View Customer</b>
<b>Actors</b>	Administrator
<b>Description</b>	This use case describes the actions the administrator needs to do in order to view the details of a customer.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The administrator needs to be logged in</li> <li>2. The application will display an option to view customer</li> <li>3. The customer must create an account previously - an option which is not mandatory</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The administrator will open the view customer option</li> <li>2. The application will display all the customers saved in the application</li> <li>4. The application will display the customer's details</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>2. The administrator is able to see the details of a registered customer</li> </ol>
<b>Alternatives</b>	N/A

Table 10 - View Customer Use Case

### 3.2.2.11 View Feedback Detailed Use Case

<b>Name</b>	<b>View Feedback</b>
<b>Actors</b>	Administrator, Waiter
<b>Description</b>	This use case describes the actions the administrator and the waiter needs to do in order to view a feedback sent by a customer.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The administrator needs to be logged in</li> <li>2. The application will display all the menu items</li> <li>3. The customer must submit a feedback</li> </ol>
<b>Activity</b>	<ol style="list-style-type: none"> <li>1. The administrator or waiter will click on the star button on any of the items</li> <li>2. The application will display the feedback left by the customers for any of the menu items</li> <li>3. The application will display the feedback</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>3. The administrator and the waiter is able to see a customer's feedback</li> </ol>
<b>Alternatives</b>	N/A

Table 11 - View Feedback Use Case

## 4. Non-Functional Specifications (FURPS+)

This section presents the identified non-functional requirements for the proposed application. The subcategories of non-functional requirements given are functionality, usability, reliability, performance, supportability.

### 4.1 Functionality

The core functionality of Click & Eat represents the main features of the application which were enumerated and described above. There are also secondary features such as:

- QR Code presented to the customer on the table which will send the customer to the restaurant's menu page for placing the order

### 4.2 Usability

Usability refers to the aesthetics, consistency, responsiveness, accessibility of the application. The usability of the Click & Eat application will be evaluated by possible users with every iteration of the development process. In the development process of Click & Eat application there will be considered the following usability principles.

- The user interface design and aesthetics is essential for providing an experience that engages the user
- The application must be easily accessible and crossable on a range of devices and browsers such as: android, IOS, desktops using Google Chrome, Microsoft Edge, Firefox. The application was tested on the devices and browsers enumerated above.
- The user should be able to find and navigate between pages within 10 seconds
- The user should be able to create an account within 2 minutes maximum, 90 percent of the time
- The user should be able to login within 5 seconds, 90 percent of the time
- A customer must be able to place and order in under 3 minutes after the customer has decided on what the order will be

- An administrator must be able to create a new menu item/category in less than 2 minutes

### **4.3 Reliability**

Reliability means the expectation of the application's up-time, what it is accepted as the application's failure, how fast can the application recover from a failure and what should be the mean time between failures.

Any failures should be recoverable 95% of the time. The application should load without issues 99% of the time.

### **4.4 Performance**

Performance relates to the application response time, transmission and retrieval of data speed, the overall application capacity to handle multiple users simultaneously and scale for new users. Data transmission and retrieval should take approximately 2 seconds maximum 95% of the time. 100 customers should be able to place an order at the same time.

### **4.5 Supportability**

Supportability refers to the ability of the software to be easily modified to accommodate enhancements and repairs. Comments and well-structured code are necessary, and it will be added in for assisting in the maintainability and in adding any new functionality.

To achieve manageable maintenance for the application, the code logic will be separated into interconnected elements that overlap in functionality as little as possible. Many programming languages, through the usage of design patterns, make it possible to separate code by dividing the structure into data processing, presentation, content, actions.

The application should be able to accept updates without suffering crashes.

## 4.6 Security

For the security requirements of the application, the following will be considered:

- Using the main Web services security requirements which are authentication, authorization, data protection
- Using external libraries and databases, always a current version will be used
- Secure payment feature
- For every action or retrieval of data, it will always be checked for access rights
- Ensuring that debug output and error messages do not leak sensitive information
- Ensuring that databases and servers are not directly reachable from the outside
- Considering to block old browsers from using the application

## 5. Success Measurement

This section explains the evaluation of the application's success as follows:

The application should run on desktop, laptop, mobile devices and should permit the following actions for each of the users:

Administrator:

- The administrator should be able to login
- The administrator should be able to create accounts for the restaurant staff
- The administrator should be able to CRUD menu
- The administrator should be able to view customer's orders details, special instructions and payment status
- The administrator should be able to mark the order as being Served
- The administrator should be able to see daily, weekly, monthly, yearly sales report
- The administrator should be able to see the menu items feedback from the customers
- The administrator should be able to safely logout

Waiter:

- The waiter should be able to login
- The waiter should be able to view customer's order details, special instructions and payment status
- The waiter should be able to mark the order as being Served
- The waiter should be able to see daily, weekly, monthly, yearly sales report
- The waiter should be able to see the menu items feedback from the customers
- The waiter should be able to safely logout

Customer:

- The customer should be able to scan the QR code
- The customer should be able to see all menu with all the details of the menu items after scanning the QR code
- The customer should be able to add menu items to their order
- The customer should be able to view their order, add more items, modify the quantity and request special instructions for each item individually and for the full order
- The customer should be able to proceed to check out when satisfied with their order
- The customer should be able to introduce their payment details, safely
- The customer should be able to view their current order
- The customer should be able to review any menu items after the order was served
- The customer should be able to register with an account for future visits
- The customer should be able to login to their account
- The customer should be able to safely logout

## **6. Testing**

Since Click & Eat is a web application, a series of techniques specific to applications hosted on the web will be used to test the interfaces and any other functionalities.

### **6.1 Functionality Testing**

- Verification of invalid redirects of the pages or pages not loading correctly

- Testing for validating input fields and data integrity
- Testing the workflow of the application

## **6.2 Usability Testing**

- Verifying the intuitiveness and ease of use of the application
- Testing the navigation and controls
- Checking the content correctness

## **6.3 Compatibility Testing**

- Checking browser compatibility
- Checking operating system compatibility
- Testing compatibility on various devices

## **6.4 Performance Testing**

- Testing the response time for operations
- Testing the capacity of the system when utilized by multiple users

## **6.5 Security Testing**

- Testing for Authentication and Authorization
- Testing for sensitive data exposure

## **6.6 Interface Testing**

- Testing the logical functionality of each feature of the application
- Simulating end user interactions with the UI

## **7. Conclusion**

The aim of this application is to meet the customer and restaurant requirements. This specification document outlined all the functional and supplementary specifications required for this application to perform as intended.

There were presented use cases that showed the actions from the user perspective, the essential features of the application and the non-essential features that are not at the core of the application but are equally important.