



*Value of incorporating  
Web page for  
The Precinct Café*

Rushit Patel  
Student Id: 2019002045

## Abstract

The following guide is meant to assist readers to understand what important paperwork was created in order to construct a fully functional website for a client. This final project report combines all of the system analysis and design phases considered, including project details, project analysis, design, and testing details. As the reader progresses through the sections of this report, he will learn what functional and non-functional requirements were taken, as well as the various design models created, beginning with wireframes, UI design, ERD's, and many more, and followed by screenshots of manual testing conducted on the website via Katalon studio.

The end of the project includes a conclusion of the complete project as well as possible future works that can be integrated into this project.

# Table of Contents

Acknowledgments.....	4
Aim and Introduction.....	7
Requirement Specifications .....	9
1. Scope.....	9
1.1 Users .....	9
1.2 Location .....	9
1.3 Responsibilities .....	9
2. Functional requirements.....	10
2.1 Functional requirements for Login .....	10
2.2 Functional requirements for Ordering Food .....	10
2.1 Functional requirements for Admin Page .....	11
2.2 Other functional requirements .....	12
3. Non-functional Requirements.....	12
3.1 Usability.....	12
3.2 Reliability.....	12
3.3 Performance .....	12
3.4 Security .....	13
Analysis .....	14
1. Use Case Diagram.....	14
2. Domain Class Modelling .....	16
3. Use Case Description.....	18
Design .....	19
1. Design for Internal Deployment .....	19
2. Activity Diagram .....	20
3. Sequence Diagram .....	21
4. Entity Relationship Diagram.....	22
5. Entity Relationship Model .....	22
6. WIREFRAMES .....	23
Implementation .....	38
References.....	45

## List of Tables

Table 1 Functional requirements for login.....	10
Table 2 Functional requirements for Ordering Food .....	11
Table 3 Functional Requirements for Admin page.....	12
Table 4 Other functional requirements .....	12
Table 5 Non-functional requirement for Usability.....	12
Table 6 Non-functional requirement for Reliability .....	12
Table 7 Non-functional requirement for Performance.....	13
Table 8 Non-functional requirement for Security.....	13
Table 9 Use Case Table .....	14
Table 10 Domain Class Table.....	16
Table 11 Use Case Description.....	18

## List of Figures

Figure 1 Use Case Diagram .....	15
Figure 2 Domain Class Modelling.....	17
Figure 3 three-tier architecture .....	19
Figure 4 three-tier architecture for The Precinct Cafe's website .....	19
Figure 5 Activity Diagram.....	20
Figure 6 Sequence Diagram .....	21
Figure 7 Entity Relationship Diagram.....	22
Figure 8 Relational Model.....	22
Figure 9 Wireframe: Home Page.....	23
Figure 10 Wireframe: Home Page [highlighting About Us and Gallery] .....	23
Figure 11 Wireframe: Home Page [highlighting Location of the cafe and Contact us form] .....	24
Figure 12 Wireframe: Menu Page.....	24
Figure 13 Wireframe: Cart Page.....	25
Figure 14 Wireframe: Checkout Page[highlighting Personal details form] .....	25
Figure 15 Wireframe: Checkout Page[highlighting types of payment methods accepted].....	26
Figure 16 Wireframe: Checkout Page[highlighting Pay by Card form] .....	26
Figure 17 Wireframe: User Login Page .....	27
Figure 18 Wireframe: Navigation Bar Once a user successfully logs in .....	27
Figure 19 Wireframe: Admin Login Page .....	28
Figure 20 Wireframe: Admin's Dashboard Page.....	28
Figure 21 Wireframe: Update/Delete Food Page .....	29
Figure 22 Update/Delete Page[highlighting updates the price] .....	29
Figure 23 Wireframe: Add Food Page .....	30
Figure 24 Wireframe: View Orders Page [highlighting customers who has successfully placed an order] .....	30
Figure 25 Wireframe: View Orders Page [highlighting all the orders of an individual customer].....	31
Figure 26 Wireframe: Customer Inquiries Page.....	31
Figure 27 Test Case for Main navigation tabs.....	32
Figure 28 Test Case for User Login.....	32
Figure 29 Test Case for User Registration.....	33

Figure 30 Test Case for Sending Inquiries.....	34
Figure 31 Test Case for Admin Logon .....	35
Figure 32 Test Case for Adding a new food item.....	35
Figure 33 Test Case for Updating the price of an food item.....	36
Figure 34 Test Case for Deleting an food item.....	36
Figure 35 Test Case for viewing orders from customers .....	37
Figure 36 Test Case for viewing customer inquiries .....	37

## Acknowledgments

The happiness and euphoria that accompanies the successful completion of any task would be insufficient if the people who made it possible, whose constant assistance and support benefited me in the completion of my project, were not mentioned.

First and foremost, I would like to convey my heartfelt gratitude to my supervisor, Shahzad Khan, for his patience, insightful remarks, helpful information, and practical suggestions that have been invaluable throughout the project. His professional skills in the field of Software Development are one of the reasons I was able to effectively complete this project.

I would like to express my deepest appreciation to The Precinct Café for graciously agreeing to be my client for this project. Furthermore, without the café's manager, Sabrina, who has become a very good friend of mine, it would have been difficult to gather information about the café, so thanks to her for supplying me with the relevant resources whenever I needed them.

By submitting this project, I declare that the work I have completed is my knowledge and research.

# Aim and Introduction

## Background of my Client

The Precinct Café is a generous food service company located at 99 Cameron Street, Whangarei. The 3.400 square foot café is constructed in the heart of Whangarei CBD.

There are eight staff members employed in this café operating with four employees every day and the store is handled by one manager, Sabrina, for a year.

Facebook is the sole internet platform via which they regularly promote their café.

Through an interview with Sabrina, I found that they were having a lot of trouble promoting this café to those living on the outskirts in order to accommodate more customers into the café. As a result, the café decided to build its website to address this issue.

### This project aims to:

- Identify and understand what SDLC is.
- Create a completely functional website through system analysis and design processes.

The primary goal of this project is to assist my client in overcoming their problems by developing a website through the software analysis and design process. This procedure will result in a website that is considerably more user-friendly and capable of meeting the expectations of my sponsor.

This research will also assist me to understand what SDLC is and why it is important to use while designing websites.

### What is SDLC?

SDLC, or Software Development Life Cycle, is a process that creates software with the highest quality and lowest cost in the shortest amount of time. SDLC provides a well-structured sequence of stages that enables an organization to quickly deliver high-quality software that has been thoroughly tested and is ready for production use. This methodology focuses on the following phases of Software Development:

#### ***Identify Problems, Planning and Analysis, Design, Build, Coding, Testing, and Deployment.***

The following SDLC methods and/or stages were considered to ensure that the process runs smoothly, efficiently, and productively. (Altvater, 2021)

- Identify Problems

*“What are the Current Problems in the organization?”* (Altvater, 2021)

Gathering information from stakeholders and learning about the café's strengths and limitations are part of this step of the SDLC. Kick-off meetings and stakeholder interviews were used to identify problems.

- Planning and Analysis (Altvater, 2021)

*"What do we want?"*

At this stage of the SDLC, this project will determine the resources required to implement the analyzed requirements. This section will outline the scope of the project, the website responsibilities, the categories of users who would benefit from the website, and at last identifying functional and non-functional requirements. System Analysis, on the other hand, will include tasks such as Use Case Diagram, Domain Class Modelling, and Use Case Description.

- Design

*"How will we get what we want?" (Altvater, 2021)*

This stage of the SDLC begins with converting the software specifications into a design plan known as the Design Specification. The developed plan was reviewed by the stakeholder, who further provided comments and suggestions. This phase incorporates activities such as creating ERDs, UML, Sequence, Activity Diagrams, and wireframes.

- Build

*"Let's create what we want?" (Altvater, 2021)*

This is the stage at which the product is programmed. During this phase, the project would convert the high-level perspective of the roadmap into a tactical set of timelines and day-to-day work schedules. (ProductPlan, 2021)

- Code Test

*"Did we get what we want?" (Altvater, 2021)*

The website is thoroughly tested with all components of the software to find any flaws, bugs, or other issues, in other words, testing to see if the code met my client's objectives. (ProductPlan, 2021)

- Deployment (Altvater, 2021)

*"Let's start using what we got"*

The purpose of this step is to launch the newly produced product. This phase varies according to the organization's preferences. For instance, my client wished to experiment with the developed product for some time before releasing it to the market. This allowed me to identify final errors before the wider community comes into contact with the product.

# Requirement Specifications

This is a section of the Requirements Specifications for a new web-based ordering system for The Precinct Café. This section outlines the scope, objectives, and goals of the new system, as well as the functional and non-functional requirements.

## 1. Scope

The scope for this project is a web-based system that supports the marketing of The Precinct Café directly to the customers especially those living in the outskirts. This website will be an additional marketing tool alongside their Facebook page through which were usually promoting their café. (*Requirements Document Example*, n.d.)

### 1.1 Users

Those who will primarily benefit from the new system include:

#### **Customers:**

Once the new system is in place, customers will find it easier to browse through the food items and order them online. Customers will also be able to access contact information, as well as the café's address and exact Google Maps location, on the site.

#### **Café's Manager:**

Sabrina will be able to manage their product information right away using the admin panel. She will be able to adjust the prices of the products, add a new product, and view the orders once a consumer puts an order.

### 1.2 Location

Any potential customer who has access to the Internet will be able to use the system. The manager of the Precinct café will also be able to access the system from any location and will be able to access restricted portions of the site via a password-protection method.

### 1.3 Responsibilities

The primary responsibilities of the new system are as follows:

- Provide customers with direct access to the up-to-date and accurate product information on which they can decide to buy
- Allow customers to place an order through the website
- Send order information directly to the admin's account
- Allow the café manager to maintain the product's information directly
- Give customers accurate contact information as well as the exact Google Maps location of the café.
- Allow any user to register if they are new to the website and to log in if they have previously registered.

Other anticipated elements of the new system include:

- A consistent "look and feel" throughout the website
- A password protection method for both public and admin
- A footer on each web page that shows a link to their Facebook profile

## 2. Functional requirements

### 2.1 Functional requirements for Login

Requirement_ID	Requirement Definition
LI_1	The system must generate a username and password for each user, as well as an email address and password for an admin.
LI_2	Every User and Admin must be able to enter their login information into the system.
LI_3	The respected users and the admin must be routed to their dashboard after clicking the "Log in" button.
LI_4	When a user/admin attempts to log into the system, he or she must be authenticated.
LI_5	Invalid credentials dialog

Table 1 Functional requirements for login

### 2.2 Functional requirements for Ordering Food

Requirement_ID	Requirement Definition
OR_1	Each user must be able to select their preferred food item by clicking the "Add to cart" button and inputting the required quantity.
OR_2	When a user selects an item, a cart must be created automatically.
OR_3	The cart must provide all the relevant information such as the item's name, quantity, and total cost based on the number of food items selected.
OR_4	When a user clicks the "Checkout" button from the cart page, they must be directed onto the billing page.
OR_5	Wherever possible, all customers must be permitted to enter their personal information.
OR_6	The system must display two payment options: PayPal and Card.

OR_7	When a user selects their preferred payment option, the system must redirect them to the appropriate page.
OR_8	All users must be able to enter their credit card information.
OR_9	When the user clicks the “Purchase” button on the Pay by card page, the system must accept the transaction And, for all users, an Order ID should be generated automatically, and they should be redirected to the cart's page onto a successful purchase.
OR_11	User must be prompted to log in or register first once they try to checkout from the cart page

Table 2 Functional requirements for Ordering Food

## 2.1 Functional requirements for Admin Page

Requirement_ID	Requirement Definition
AP_1	Before accessing the admin's dashboard, the system must ask for the administrator's email and password.
AP_2	Once an administrator has successfully logged in, they must be led to the dashboard page.
AP_3	When an administrator clicks on one of the navigation buttons in the main navigation bar, he or she must be routed to that page.
AP_4	Admin must be able to enter the product's name, price, and select an image from the Add Food page.
AP_5	The system must successfully add a product to the Menu page when an administrator presses the "Add" button from the Add Food page.
AP_6	Once the administrator clicks on the "Update/ Delete Food" button, he or she must be successfully directed to the Edit Food page.
AP_7	Admin should be directed onto the Update Food's page once he/she clicks onto the "Update" button from the Edit Food's page
AP_8	When the administrator enters the new price and hits the "update" button on the Update The Price page, the system must update the pricing of the product.
AP_9	The system must show all the customer orders once an Admin clicks on the “View Orders” button

AP_10	The system must show all the customer inquiries once an Admin clicks on the "Customer Inquiries" button
-------	---

Table 3 Functional Requirements for the Admin page

## 2.2 Other functional requirements

Requirement_ID	Requirement Definition
OFR_1	Regardless of whether or not a customer is logged in, all users are permitted to fill out the Contact Form from the Home page.

Table 4 Other functional requirements

## 3. Non-functional Requirements

### 3.1 Usability

Requirement_ID	Requirement Definition
U_NFR_1	A seamless Ordering process
U_NFR_2	The system must be simple to use for both novices and experienced users.
U_NFR_2	When pushing a button, the system should respond in less than 2 seconds.

Table 5 Non-functional requirement for Usability

### 3.2 Reliability

Requirement_ID	Requirement Definition
R_NFR_1	All product information on the website, as well as the café's contact information, must be correct and up to date.
R_NFR_2	The defect rate of the system shall be fewer than one failure per 1000 hours of operation.
R_NFR_3	If the system goes down due to a problem, it should be recoverable in less than 8 seconds.

Table 6 Non-functional requirement for Reliability

### 3.3 Performance

Requirement_ID	Requirement Definition
P_NFR_1	The webpage should load completely in 3 seconds.
P_NFR_2	Each request must be completed in 4 seconds or less.
P_NFR_3	Over an LTE connection, the website must serve 5,000 users per hour and have a response

	time of 5 seconds or less in a Chrome desktop browser, including text and image rendering.
--	--

*Table 7 Non-functional requirement for Performance*

### 3.4 Security

<b>Requirement_ID</b>	<b>Requirement Definition</b>
S_NFR_1	All Users and administrators must be authenticated before they may access any of their or others' personal data.

*Table 8 Non-functional requirement for Security*

## Analysis

### 1. Use Case Diagram

<i>Use Case</i>	<i>Description</i>	<i>Actors</i>
Register	To order food, users must first register by inputting their personal information into the registration form.	New User
Login	User must enter his/her username and password to log into the account	Registered User
Make Inquiries	Customer can send their inquiries by filling “Contact Us” form	Registered User New User
Choose a food	Choose a food item by clicking the “Add to cart” button	Registered User New User
Manage Cart	A user can either delete the chosen food item or add another item from the Cart	Registered User New User
View Cart	User can view the cart by clicking the “cart” button from the navigation bar	Registered User New User
Checkout item	User can check out the chosen food items by clicking the “checkout” button from the Cart page	Registered User
Make Payment	The checkout process includes the Payment process for the chosen food items.  Payment extends two options, Pay by Card and Pay by PayPal	Registered User Registered User
Make Order	A customer makes an order by a successful payment transaction.  When a successful order is placed, an Order ID is automatically generated for each customer.	Registered User System
View Orders	Can see all of the information of the customer orders including their payment status	Admin
Manage Products	Maintain the products by adding a new product and by deleting as well as updating the existing product	Admin

Table 9 Use Case Table

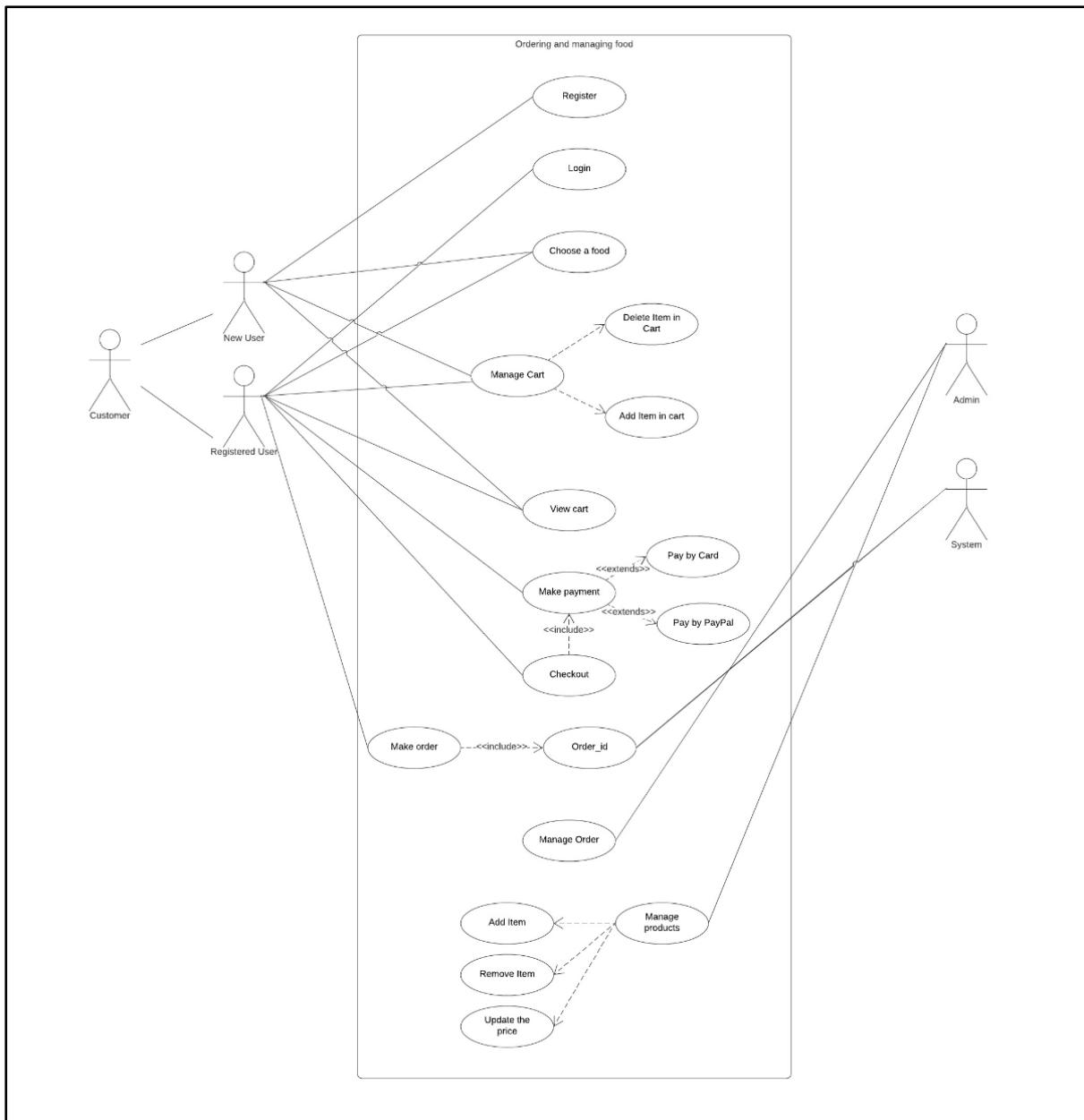


Figure 1 Use Case Diagram

## 2. Domain Class Modelling

<i>Classes</i>	<i>Attributes</i>
<i>Customer</i>	<i>Name, Address, Phononenumber, Email</i>
<i>User</i>	<i>Username, Email, password</i>
<i>Admin</i>	<i>Admin_email, admin_password, Name, Phononenumber</i>
<i>Inquiries</i>	<i>Cus_name, Cus_email, subject, message</i>
<i>Food item</i>	<i>Food_name, food_price, food_image</i>
<i>Cart</i>	<i>Food_name, food_price, food_quantity, total_price</i>
<i>Food Order</i>	<i>Order_id, food_name, food_price, food_quantity, food_total, cus_name, cus_email, cus_address, cus_phonenum</i>
<i>Payment</i>	<i>pay_by_card, pay_by_PayPal</i>
<i>Pay By Card</i>	<i>Name_on_card, Card_number, Expiry_date, CVV</i>
<i>Pay by PayPal</i>	

Table 10 Domain Class Table

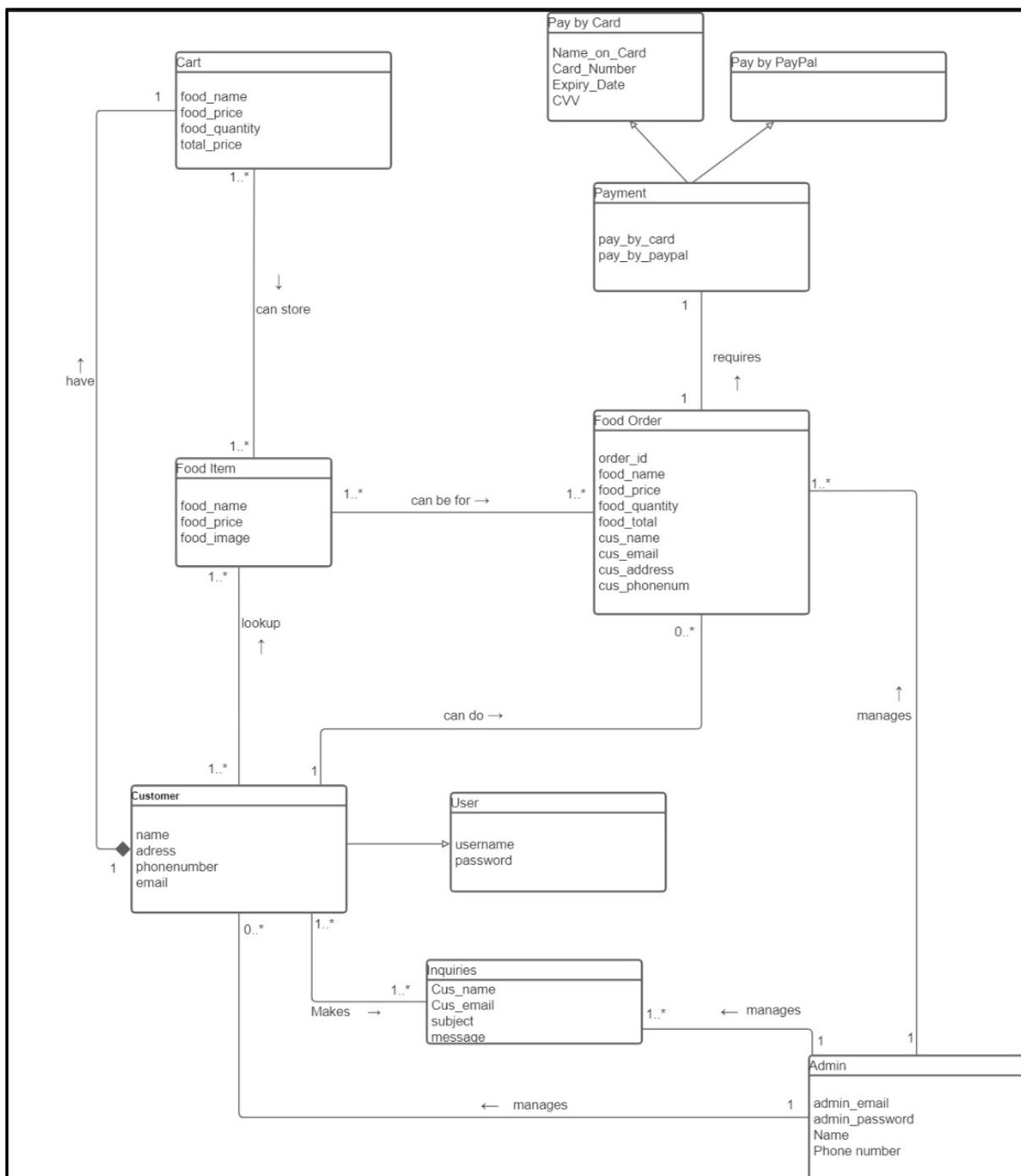


Figure 2 Domain Class Modelling

### 3. Use Case Description

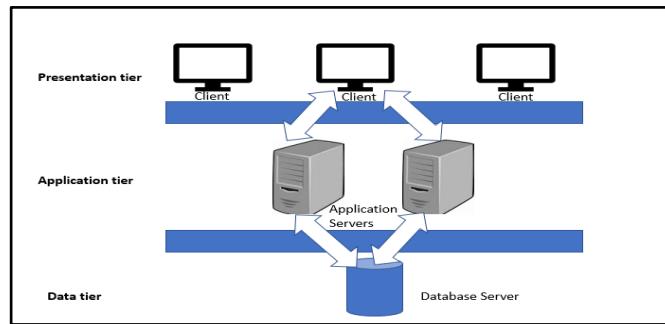
Use Case Name:	Ordering Process																					
Scenario:	Ordering Food Online																					
Triggering event:	Customers want to order food online																					
Brief Description:	<p>Customers browse through the advertised food items on the menu page. In order to place an order, customers must first register or login themselves. After that, they select their desired meal by clicking the "Add to cart" button. And the selected product is automatically put to the cart; from there, the customer clicks on the "Checkout" button, fills out their personal information, and selects their preferred payment option. When a payment transaction is completed successfully. The order will be processed.</p>																					
Actors:	Customers																					
Stakeholders:	Manager																					
Preconditions:	Customers must log in before checking out																					
Post Conditions:	<ul style="list-style-type: none"> <li>- Customers must receive an Order ID on every successful order placement</li> </ul>																					
The flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th><th>System</th></tr> </thead> <tbody> <tr> <td>1. Click on “Login Page”</td><td>3.1 Verifies the entered credentials with the database</td></tr> <tr> <td>2. Enter the valid username and password</td><td>3.2 Authenticate users</td></tr> <tr> <td>3. Click on “Login” button</td><td>3.3 Successfully logs in the user</td></tr> <tr> <td>4. Browse through the advertised food items from the menu page</td><td></td></tr> <tr> <td>5. If interested in a specific food, enter the quantity needed and then click the "Add to cart" button.</td><td></td></tr> <tr> <td>6. Click on the “Checkout” button from the cart page</td><td>5.1 Adds the selected products, together with their quantities, to the cart automatically.</td></tr> <tr> <td>7. Enter your personal information and select your desired payment option.</td><td>5.2 The total payment amount is calculated based on the price of each product and the quantity.</td></tr> <tr> <td>8. Fill in the appropriate bank details as required and formatted</td><td>9.1 Automatically Generate Order ID and redirect users onto the Cart Page</td></tr> <tr> <td>9. Click on the “Purchase” button to successfully order the chosen food items</td><td>6.2 Sends the Customer information as well as the information on the food items purchased to the Admin Panel automatically.</td></tr> </tbody> </table>	Actor	System	1. Click on “Login Page”	3.1 Verifies the entered credentials with the database	2. Enter the valid username and password	3.2 Authenticate users	3. Click on “Login” button	3.3 Successfully logs in the user	4. Browse through the advertised food items from the menu page		5. If interested in a specific food, enter the quantity needed and then click the "Add to cart" button.		6. Click on the “Checkout” button from the cart page	5.1 Adds the selected products, together with their quantities, to the cart automatically.	7. Enter your personal information and select your desired payment option.	5.2 The total payment amount is calculated based on the price of each product and the quantity.	8. Fill in the appropriate bank details as required and formatted	9.1 Automatically Generate Order ID and redirect users onto the Cart Page	9. Click on the “Purchase” button to successfully order the chosen food items	6.2 Sends the Customer information as well as the information on the food items purchased to the Admin Panel automatically.	
Actor	System																					
1. Click on “Login Page”	3.1 Verifies the entered credentials with the database																					
2. Enter the valid username and password	3.2 Authenticate users																					
3. Click on “Login” button	3.3 Successfully logs in the user																					
4. Browse through the advertised food items from the menu page																						
5. If interested in a specific food, enter the quantity needed and then click the "Add to cart" button.																						
6. Click on the “Checkout” button from the cart page	5.1 Adds the selected products, together with their quantities, to the cart automatically.																					
7. Enter your personal information and select your desired payment option.	5.2 The total payment amount is calculated based on the price of each product and the quantity.																					
8. Fill in the appropriate bank details as required and formatted	9.1 Automatically Generate Order ID and redirect users onto the Cart Page																					
9. Click on the “Purchase” button to successfully order the chosen food items	6.2 Sends the Customer information as well as the information on the food items purchased to the Admin Panel automatically.																					
Exception Condition:	- Invalid login credentials																					

Table 11 Use Case Description

# Design

## 1. Design for Internal Deployment

A three-tier client/server architecture is a sort of multi-layered computer architecture in which a complete application is distributed across three processing tiers. It divides the presentation, application logic and data processing layers across client and server devices. (Techopedia, 2013)

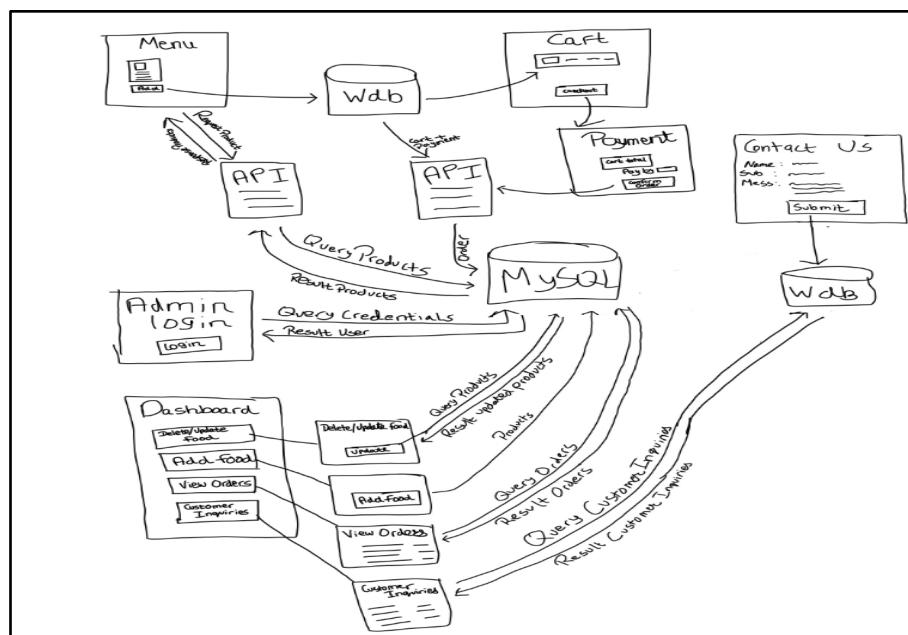


*Figure 3 three-tier architecture* (Raj, n.d.)

The application software is divided into three tiers:

- Presentation tier: Also known as the view layer, this tier accepts all user inputs and organizes them before displaying the processing results.
  - Application tier: implements business processing rules and procedures.
  - Data tier: maintains stored data, which is often stored in one or more databases.

This project will use this architecture, and the diagram below depicts the Client-Server communication for this project's website.



*Figure 4 three-tier architecture for The Precinct Cafe's website*

## 2. Activity Diagram

Typically, an event requires the completion of some procedures, particularly when the operation is intended to accomplish a variety of different goals that require coordination. The image below represents the set of activities performed by a customer on the site when he or she wishes to order a food item.

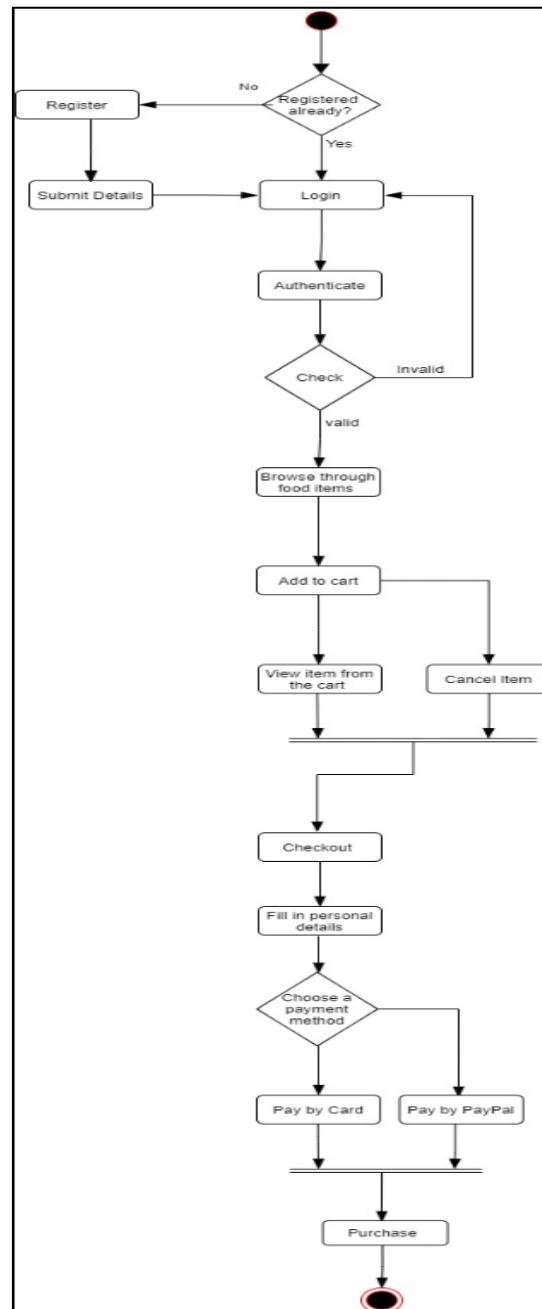


Figure 5 Activity Diagram

### 3. Sequence Diagram

The figure below depicts object interactions in chronological order. It displays the objects involved when a consumer attempts to order a food item, as well as the sequence of messages sent between the objects required to carry out the scenario's functionality.

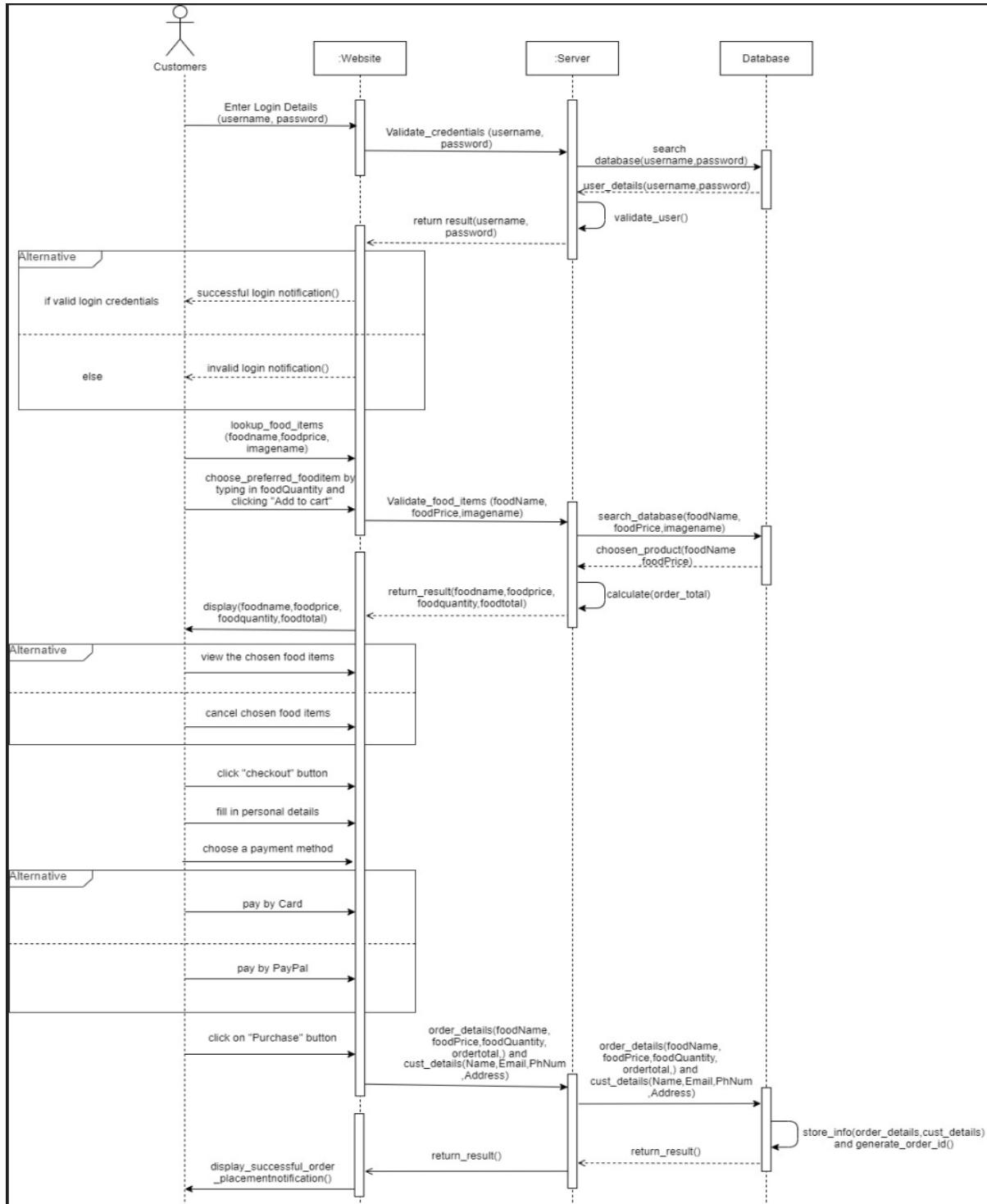


Figure 6 Sequence Diagram

#### 4. Entity Relationship Diagram

The diagram below depicts the relationships among entity sets in a database. These entities have attributes that specify their properties as well.

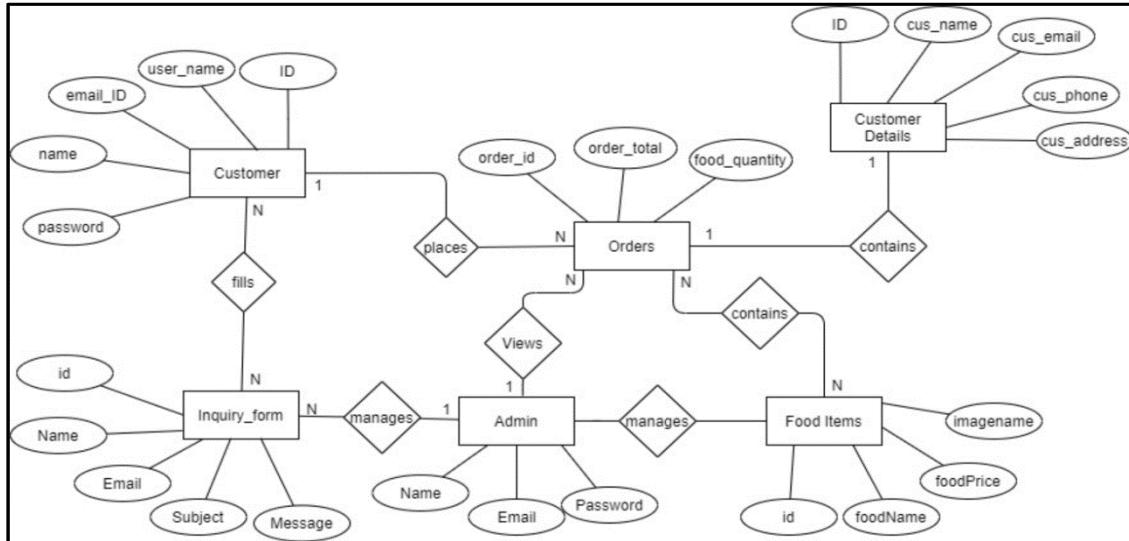


Figure 7 Entity Relationship Diagram

#### 5. Relational Model

The diagram below shows the entities along with their primary keys, foreign keys, cardinalities and its attributes.

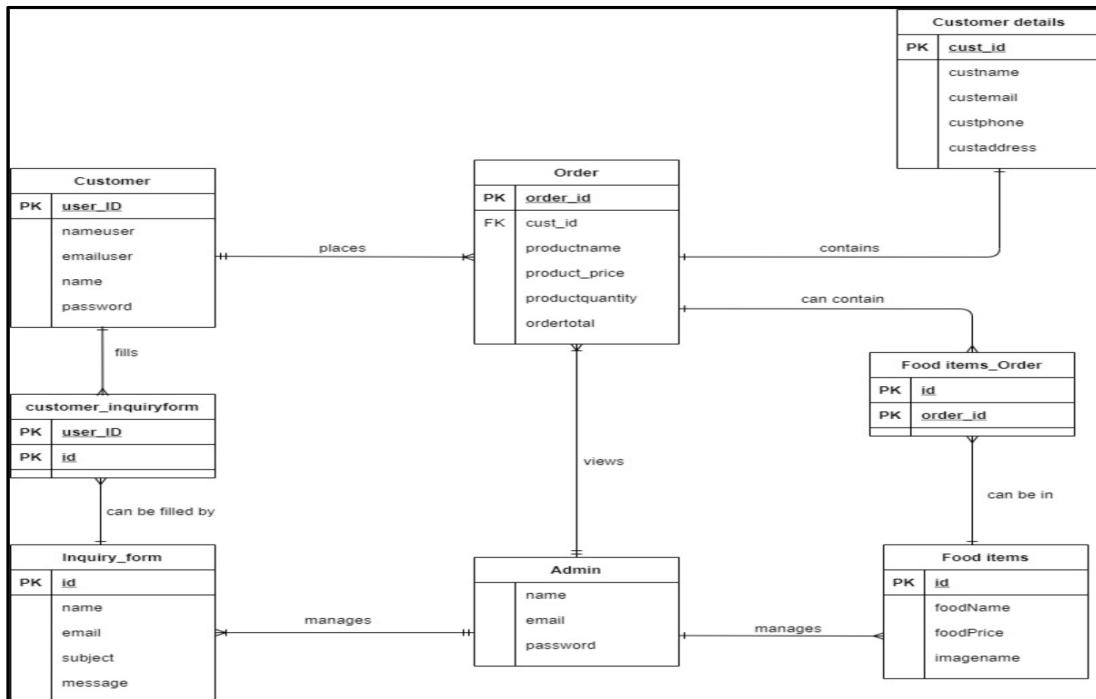


Figure 8 Relational Model

## 6. WIREFRAMES

- Home Page

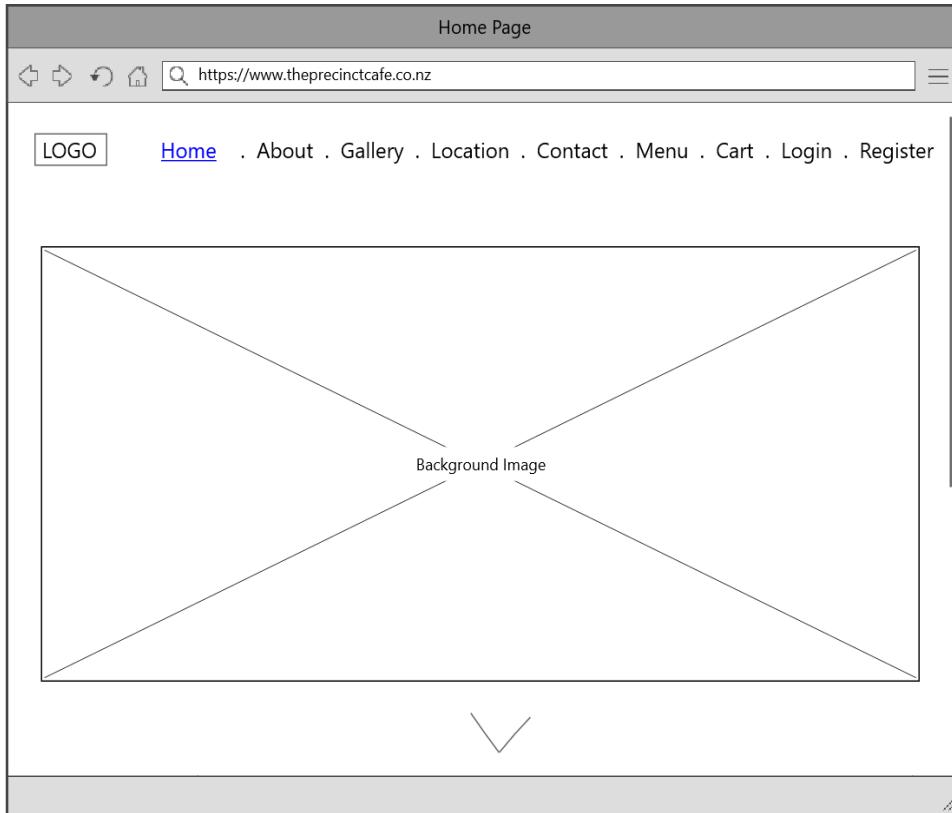


Figure 9 Wireframe: Home Page

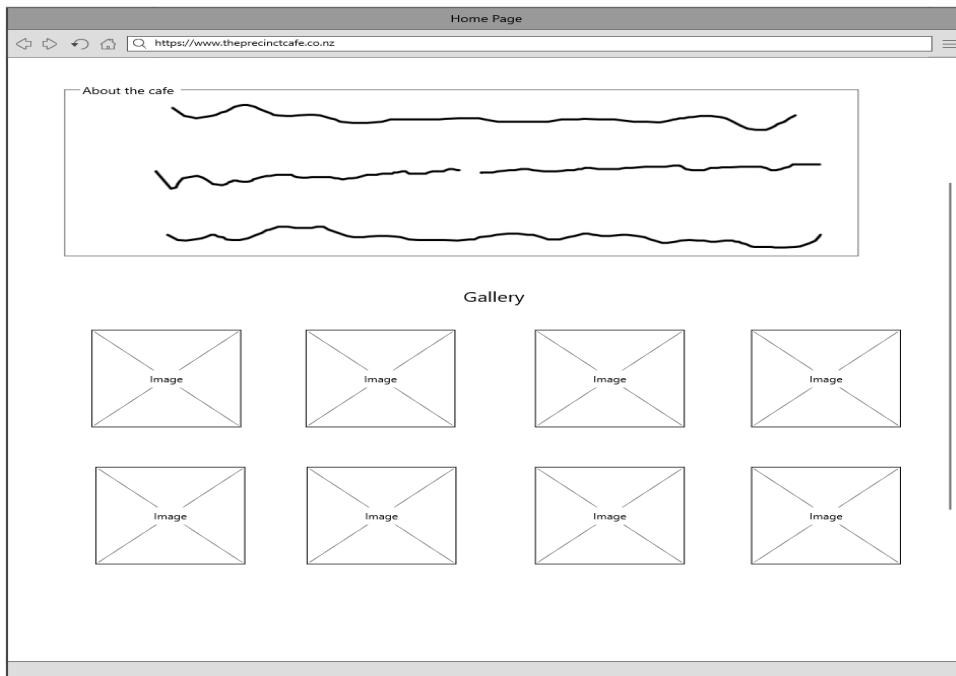


Figure 10 Wireframe: Home Page [highlighting About Us and Gallery]

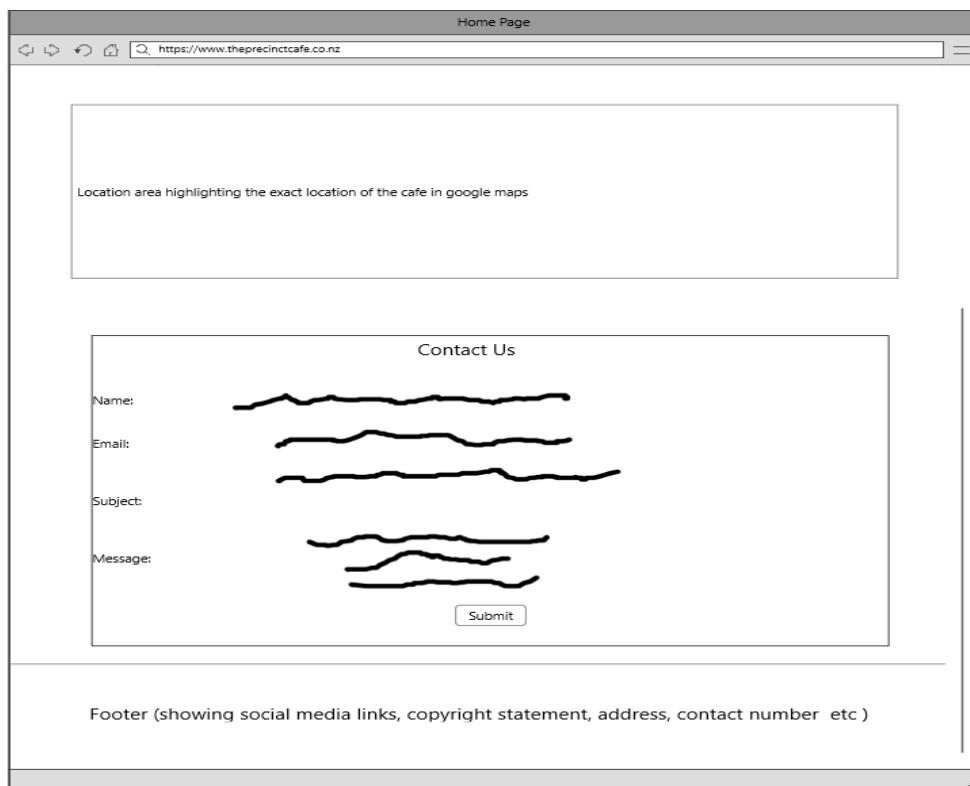


Figure 11 Wireframe: Home Page [highlighting Location of the cafe and Contact us form]

- **Menu Page**

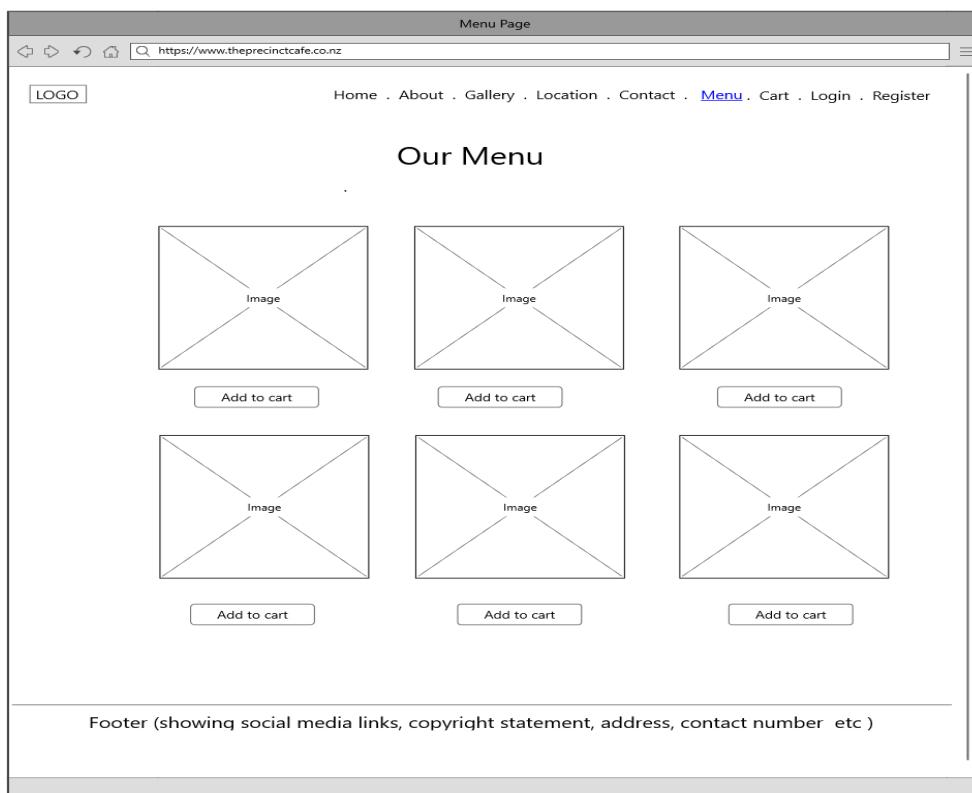


Figure 12 Wireframe: Menu Page

- Cart Page

The wireframe shows a browser window titled "Cart Page". The URL bar contains "https://www.theprecinctcafe.co.nz". The header includes a "LOGO" button and navigation links: Home . About . Gallery . Location . Contact . Menu . [Cart](#) . Login . Register. The main content area is titled "Cart" and contains a table:

Name	Quantity	Price	Total
Cappuccino	3	\$20	\$60

Below the table is a "Purchase" button and a "Total Amount:" field containing a wavy line. At the bottom, there is a footer section with the text "Footer (showing social media links, copyright statement, address, contact number etc)".

Figure 13 Wireframe: Cart Page

- Checkout Page

The wireframe shows a browser window titled "Checkout Page". The URL bar contains "https://www.theprecinctcafe.co.nz". The header includes a "LOGO" button and navigation links: Home . About . Gallery . Location . Contact . Menu . [Cart](#) . Login . Register. The main content area is titled "Checkout" and contains a large rectangular form labeled "Personal details" with four input fields:

- Full Name
- Email
- Phone Number
- Address

Figure 14 Wireframe: Checkout Page[highlighting Personal details form]

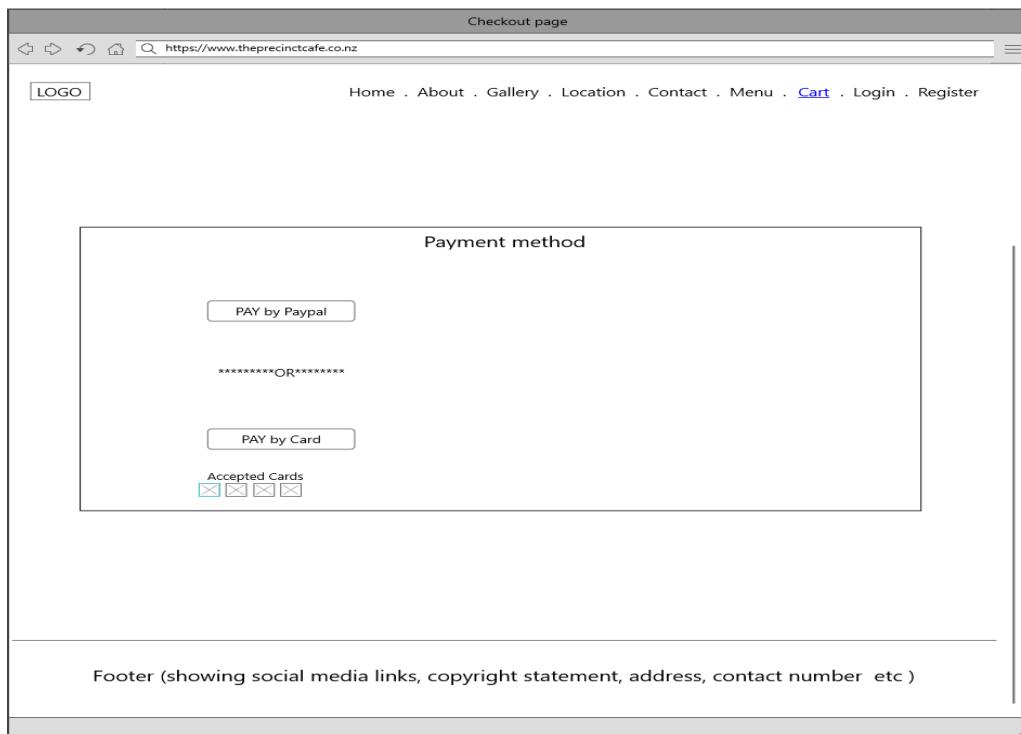


Figure 15 Wireframe: Checkout Page[highlighting types of payment methods accepted]

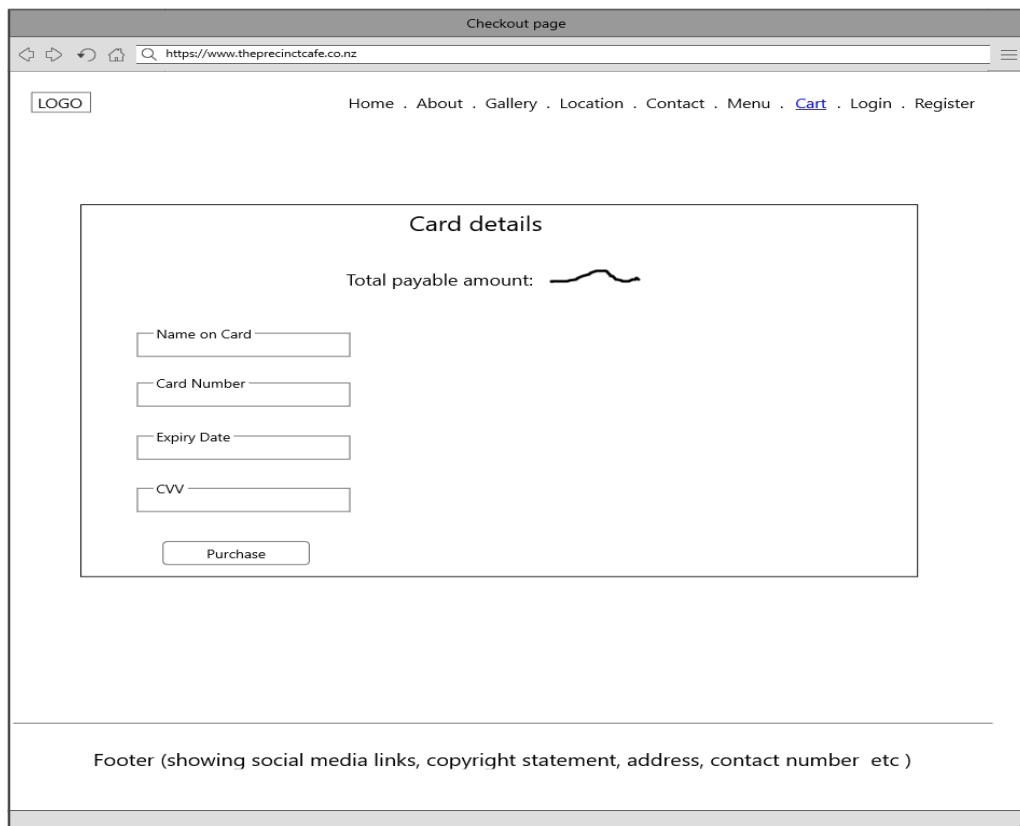
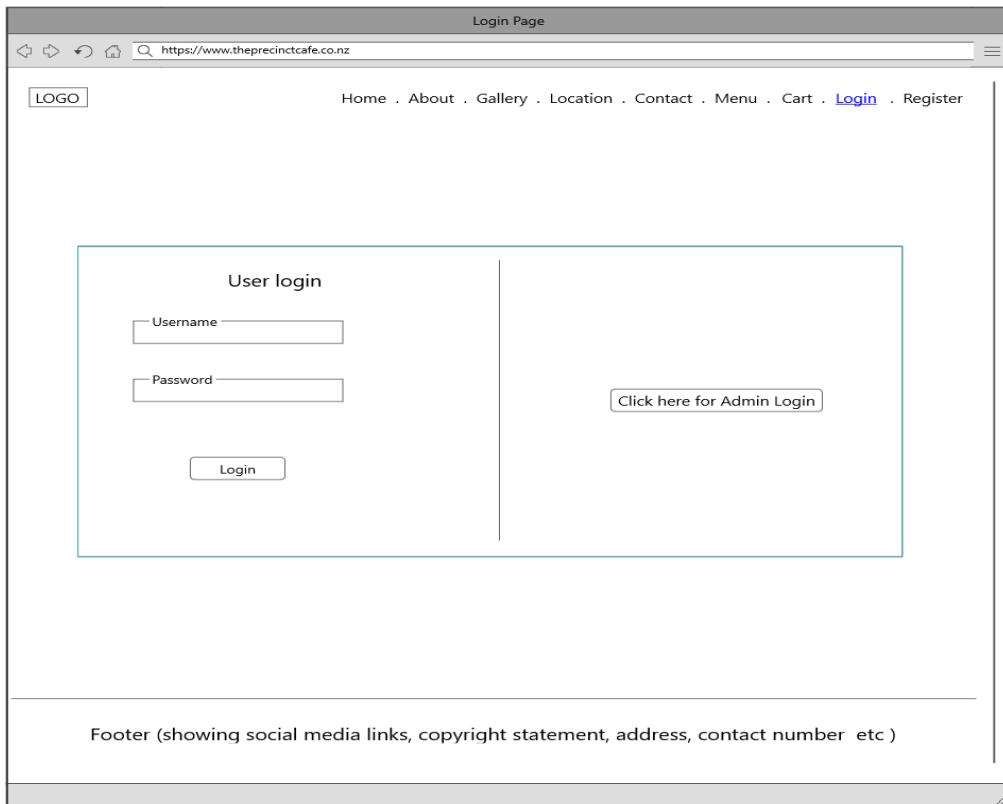


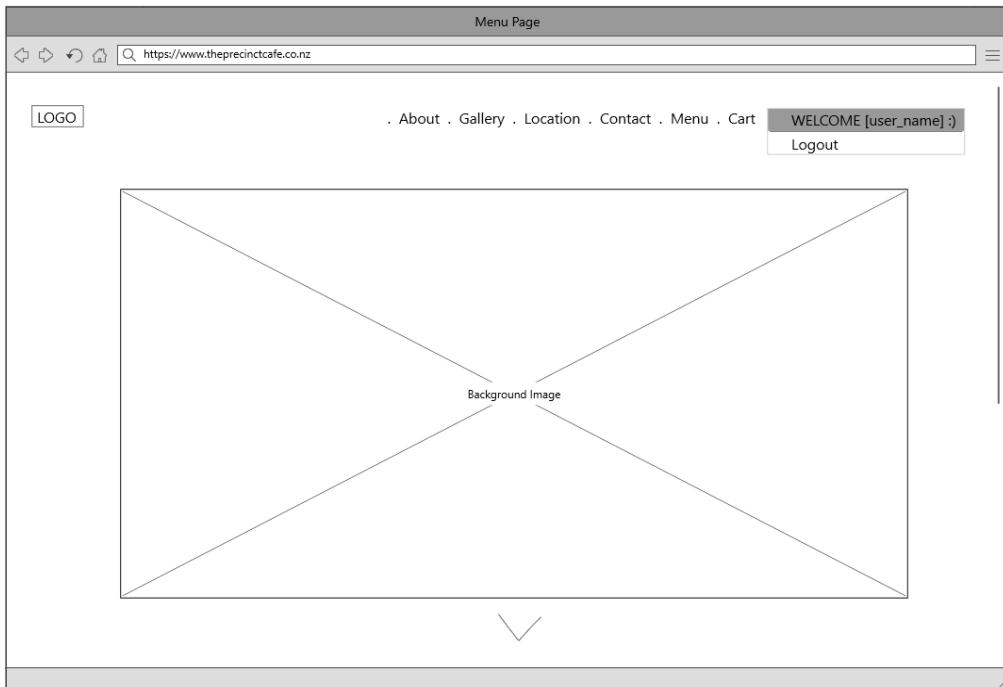
Figure 16 Wireframe: Checkout Page[highlighting Pay by Card form]

- User Login Page



*Figure 17 Wireframe: User Login Page*

- The diagram below shows the scenario once a user successfully logs in



*Figure 18 Wireframe: Navigation Bar Once a user successfully logs in*

- Admin's login page

The wireframe shows a browser window titled "Login Page". In the address bar, the URL is https://www.theprecinctcafe.co.nz. The main content area contains a "LOGO" placeholder and a "Home" link. A large central box is labeled "Admin login" and contains fields for "Email" and "Password", along with a "Login" button. At the bottom of the page is a footer section labeled "Footer (showing copyright statement)".

Figure 19 Wireframe: Admin Login Page

- Admin's Dashboard

The wireframe shows a browser window titled "Dashboard Page". In the address bar, the URL is https://www.theprecinctcafe.co.nz. The main content area contains a "LOGO" placeholder and a navigation menu with links for "Home", ". Main Site .", and a dropdown menu. The dropdown menu is currently open, showing options: "Menu", "Update/Delete", "Food", "Add food", "ViewOrders", and "Customer inquiries". To the right of the menu, a welcome message ". WELCOME [admin\_email] :)" is displayed. At the bottom of the page is a footer section labeled "Footer (showing copyright statement)".

Figure 20 Wireframe: Admin's Dashboard Page

- Update / Delete Food Page

The wireframe shows a web browser window titled "Edit food page". The URL bar contains "https://www.theprecinctcafe.co.nz". The header includes a "LOGO" button, navigation links for "Home", "Main Site", "Menu", and a welcome message "WELCOME [admin\_email] :)". Below the header is the main content area titled "Edit food". It displays a table with four columns: "Name", "Price", "Image", and "Actions". There are four rows of data, each representing a food item. The "Name" column lists "Cappuccino" three times and "Latte" once. The "Price" column shows "\$19" for all items. The "Image" column contains four placeholder "Image" boxes. The "Actions" column for each row contains two buttons: "Update" and "Delete". At the bottom of the content area is a footer section labeled "Footer (showing copyright statement)".

Figure 21 Wireframe: Update/Delete Food Page

The wireframe shows a web browser window titled "Update price page". The URL bar contains "https://www.theprecinctcafe.co.nz". The header includes a "LOGO" button, navigation links for "Home", "Main Site", "Menu", and a welcome message "WELCOME [admin\_email] :)". Below the header is the main content area titled "Update the price". It displays a form with fields for "Name" (set to "Cappuccino"), "Image" (a placeholder "Image" box), and "New Price" (a text input box containing "19"). At the bottom of the form are "Cancel" and "Submit" buttons. At the very bottom of the page is a footer section labeled "Footer (showing copyright statement)".

Figure 22 Update/Delete Page[highlighting updates the price]

- Add food Page

The wireframe shows a web browser window titled "Add food Page". The URL bar contains "https://www.theprecinctcafe.co.nz". The page header includes a "LOGO" icon, navigation links "Home . Main Site . Menu . WELCOME [admin\_email] ;)", and a title "Add food". Below the title are three input fields: "Title of food", "Price", and "Choose a file". A central "Add" button is positioned between the second and third fields. At the bottom of the page is a horizontal line followed by the text "Footer (showing copyright statement)".

Figure 23 Wireframe: Add Food Page

- View Orders

The wireframe shows a web browser window titled "View Orders Page". The URL bar contains "https://www.theprecinctcafe.co.nz". The page header includes a "LOGO" icon, navigation links "Home . Main Site . Menu . WELCOME [admin\_email] ;)", and a title "Customer Details". Below the title is a table with six columns: Customer ID, Customer Name, Customer Email, Customer Phone, Customer Address, and Actions. The first row has a Customer ID of "1" and a Customer Name of "Harry". The "Actions" column for this row contains three "View Order" buttons. The table has four rows in total. At the bottom of the page is a horizontal line followed by the text "Footer (showing copyright statement)".

Customer ID	Customer Name	Customer Email	Customer Phone	Customer Address	Actions
1	Harry	harry@gmail.com	0225714690	11/10 Avenue,NZ	<button>View Order</button> <button>View Order</button> <button>View Order</button>

Figure 24 Wireframe: View Orders Page [highlighting customers who has successfully placed an order]

View Orders page					
 Home . Main Site . Menu . WELCOME [admin_email] ;)					
Order details					
Order ID	Product name	Product price	Product quantity	Product total	Payment Status
1	Cappuccino	\$20	3	\$60	Paid

---

Footer (showing copyright statement)

Figure 25 Wireframe: View Orders Page [highlighting all the orders of an individual customer]

- Customer Inquiry Page

Customer inquiry page			
 Home . Main Site . Menu . WELCOME [admin_email] ;)			
Customer Inquiries			
Customer name	Customer Email	Subject	Message
Rushit	rushit12all@gmail.com	Location	Having difficulty searching for your cafe

---

Footer (showing copyright statement)

Figure 26 Wireframe: Customer Inquiries Page

# Test

## 1. Test Case for Main navigations tabs

The screenshot shows a test execution interface with a table of test steps and a detailed log view.

**Test Case Table:**

Item	Object	Input	Output	Description
-> 1 - Open Browser		""		
-> 2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
-> 3 - Click	a_About			
-> 4 - Click	a_Gallery			
-> 5 - Click	a_Location			
-> 6 - Click	a_Contact			
-> 7 - Click	a_Menu			
-> 8 - Click	a_Login			
-> 9 - Click	a_Register			

**Test Log:**

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

06-29-2021 04:31:14 PM Test Cases/nav\_buttons  
Elapsed time: 12.433s  
Test Cases/nav\_buttons

Test steps listed in the log:

- > 1 - openBrowser("") (4.004s)
- > 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.090s)
- > 3 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_About")) (0.801s)
- > 4 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Gallery")) (0.607s)
- > 5 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Location")) (1.033s)
- > 6 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Contact")) (0.592s)
- > 7 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Menu")) (0.826s)
- > 8 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Login")) (0.962s)
- > 9 - click(findTestObject("Object Repository/Page\_User and Admin Login/a\_Register")) (0.887s)

Figure 27 Test Case for Main navigation tabs

## 2. Test Case for User Login

The screenshot shows a test execution interface with a table of test steps and a detailed log view.

**Test Case Table:**

Item	Object	Input	Output	Description
-> 1 - Open Browser		""		
-> 2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
-> 3 - Click	a_Login (2)			
-> 4 - Set Text	input_Username_user (1)	"rp120300"		
-> 5 - Set Encrypted Text	input_Password_password (1)	"rHrA8Gqu/PiOMrMKSAnNNA=="		
-> 6 - Click	button_Login (1)			
-> 7 - Click	button_WELCOME rp120300 )			
-> 8 - Click	a_LOGOUT (1)			

**Test Log:**

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

06-29-2021 05:20:24 PM Test Cases/user\_registration  
Elapsed time: 11.893s  
Test Cases/user\_registration

Test steps listed in the log:

- > 1 - openBrowser("") (3.604s)
- > 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.370s)
- > 3 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Register")) (0.983s)
- > 4 - setText(findTestObject("Object Repository/Page\_User Registration/input\_Full Name\_nameuser"), "Zayn")
- > 5 - setText(findTestObject("Object Repository/Page\_User Registration/input\_Email\_emailuser"), "zayn@gr...
- > 6 - setText(findTestObject("Object Repository/Page\_User Registration/input\_Username\_user"), "zayn1234")
- > 7 - setEncryptedText(findTestObject("Object Repository/Page\_User Registration/input\_Password\_password")) (0.858s)
- > 8 - click(findTestObject("Object Repository/Page\_User Registration/button\_Register")) (0.858s)

Figure 28 Test Case for User Login

### 3. Test Case for User Registration

The screenshot shows a QTP test case for user registration. At the top, there's a table with columns for Item, Object, Input, Output, and Description. Below the table is a toolbar with various icons and tabs like 'Manual', 'Script', 'Variables', 'Integration', and 'Properties'. Underneath the toolbar, there's a status bar showing 'Runs: 1/1', 'Passes: 1', 'Failures: 0', 'Errors: 0', and 'Skips: 0'. The main area contains a tree view of test steps and their execution times, followed by a detailed log viewer.

Item	Object	Input	Output	Description
1 - Open Browser		"		
2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
3 - Click	a_Register			
4 - Set Text	input_Full_Name_nameuser	"Zayn Malik"		
5 - Set Text	input_Email_emailuser	"zayn@gmail.com"		
6 - Set Text	input_Username_user	"zayn1234"		
7 - Set Encrypted Text	input_Password_password	"wnf1HxwdOZ9yqbNMd+xEYA=="		
8 - Click	button_Register			

Test Cases/user\_registration (11.893s)

- 1 - openBrowser("") (3.604s)
- 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.370s)
- 3 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Register")) (0.983s)
- 4 - setText(findTestObject("Object Repository/Page\_User Registration/input\_Full Name\_nameuser"), "Zayn Malik") (0.001s)
- 5 - setText(findTestObject("Object Repository/Page\_User Registration/input\_Email\_emailuser"), "zayn@gmail.com") (0.001s)
- 6 - setText(findTestObject("Object Repository/Page\_User Registration/input\_Username\_user"), "zayn1234") (0.001s)
- 7 - setEncryptedText(findTestObject("Object Repository/Page\_User Registration/input\_Password\_password"), "wnf1HxwdOZ9yqbNMd+xEYA==") (0.858s)
- 8 - click(findTestObject("Object Repository/Page\_User Registration/button\_Register")) (0.858s)

Elapsed time: 11.893s

Test Cases/user\_registration

Figure 29 Test Case for User Registration

### 4. Test Case for Ordering Food

<b>Test Case ID</b>	OF_001	<b>Test Case Description</b>	Test the ordering process		
<b>Created By</b>	Rushit	<b>Reviewed By</b>	Shahzad	<b>Version</b>	1.0
<b>Tester's Name</b>	Mark	<b>Date Tested</b>	29-06-2021	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass

S #	Prerequisites:
1	User must login first

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to http://localhost:88/theprecinctcafe/home.php	Site should open	As Expected	Pass
2	Click on Login button	Login Page should open	As Expected	Pass
3	Enter username	Username must be entered,"rp120300"	As Expected	Pass
4	Enter password	Password must be entered,"@rp120300"	As Expected	Pass
5	Click "Login"	Customer must be logged in and must be directed onto the home page	As Expected	Pass
6	Click "Menu" button	Menu Page must open	As Expected	Pass
7	Hover over the preferred food item, For instance "cappuccino"	Options to enter quantity and add to cart button must be displayed	As Expected	Pass
8	Enter quantity	Credentials must be entered, for instance "3"	As Expected	Pass
9	Click "Add to cart" button	Chosen food items must be entered	As Expected	Pass
10	Click "Cart" button	The cart page must appear, displaying all of the selected food items as	As Expected	Pass

		well as the total price to be paid.		
11	Click on "Checkout" button	Checkout must open displaying all the fields required to be filled by the customer	As Expected	Pass
12	Enter Full Name	Credentials must be entered, "Zayn Malik"	As Expected	Pass
13	Enter Email	Credentials must be entered,"zayn@gmail.com"	As Expected	Pass
14	Enter Phone Number	Credentials must be entered, for instance "0225714599"	As Expected	Pass
15	Choose a Payment Method, Click "Pay by Card" button	Pay by Card must open	As Expected	Pass
16	Enter Name on Card	Credentials must be entered, for instance "Zayn Malik"	As Expected	Pass
17	Enter Card Number	Credentials must be entered, for instance "1234123412341234"	As Expected	Pass
18	Enter Expiry Date	Credentials must be entered, for instance "12/21"	As Expected	Pass
19	Enter CVV	Credentials must be entered, for instance "310"	As Expected	Pass
20	Click "Purchase"	Food must be successfully ordered and user must be re-directed onto the Cart Page	As Expected	Pass

Table 12 Test Case for Ordering Food

## 5. Test Case for Sending Inquiries

The screenshot shows a test automation interface with the following components:

- Table of Test Steps:**

Item	Object	Input	Output	Description
1 - Open Browser		""		
2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
3 - Click	a_Contact (1)			
4 - Set Text	input_Phone_name	"Zayn Malik"		
5 - Set Text	input_Phone_email	"zayn@gmail.com"		
6 - Set Text	input_Phone_subject	"This is test"		
7 - Set Text	textarea_Phone_message	"I hope it is successful"		
8 - Click	button_Submit			
- Script Editor:**

```

Manual <> Script <> Variables <> Variables (Script mode) ■■■ Integration ■ Properties
Problems Event Log Console Log Viewer Self-healing Insights
Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

```
- Test Results Summary:**

06-29-2021 06:44:59 PM Test Cases/send\_inquiry  
Elapsed time: 13.070s  
Test Cases/send\_inquiry
- Test Log:**
  - Test Cases/send\_inquiry (13.070s)
    - 1 - openBrowser("") (3.906s)
    - 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.187s)
    - 3 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Contact (1)")) (0.799s)
    - 4 - setText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Phone\_name"), "Zayn Malik") (0.799s)
    - 5 - setText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Phone\_email"), "zayn@gmail.com") (0.799s)
    - 6 - setText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Phone\_subject"), "This is test") (0.799s)
    - 7 - setText(findTestObject("Object Repository/Page\_The Precinct Caf/textarea\_Phone\_message"), "I hope it is successful") (0.799s)
    - 8 - click(findTestObject("Object Repository/Page\_The Precinct Caf/button\_Submit")) (0.764s)

Figure 30 Test Case for Sending Inquiries

## 6. Test Case for Admin Logon

Item	Object	Input	Output	Description
-> 1 - Open Browser		""		
-> 2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
-> 3 - Click	a_Login (6)			
-> 4 - Click	button_Click here for Admin Login (1)			
-> 5 - Set Text	input_Email_user (1)	"tiger@gmail.com"		
-> 6 - Set Encrypted Triput_Password_password (1)		"rHrA8Gqu/PiOMrMKSAnNNA="		
-> 7 - Click	button_Login (1)			
-> 8 - Click	button_WELCOME tiger@gmail.com			
-> 9 - Click	a_LOGOUT (1)			

Manual </> Script </> Variables </> Variables (Script mode) ■ Integration ■ Properties

Problems Event Log Console Log Viewer Self-healing Insights

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

```
06-29-2021 05:46:25 PM Test Cases/admin_logon
Elapsed time: 11.869s
Test Cases/admin_logon
```

- Test Cases/admin\_logon (11.869s)
  - 1 - openBrowser("") (4.030s)
  - 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.089s)
  - 3 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Login (6)")) (0.901s)
  - 4 - click(findTestObject("Object Repository/Page\_User and Admin Login/button\_Click here fo"))
  - 5 - setText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Email\_user (1)"), "tig")
  - 6 - setEncryptedText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Password"))
  - 7 - click(findTestObject("Object Repository/Page\_The Precinct Caf/button\_Login (1)")) (0.647s)
  - 8 - click(findTestObject("Object Repository/Page\_Admin Page/button\_WELCOME tiger@gmail.com"))
  - 9 - click(findTestObject("Object Repository/Page\_Admin Page/a\_LOGOUT (1)")) (0.974s)

Figure 31 Test Case for Admin Logon

## 7. Test Case for Adding a new Food item

Item	Object	Input	Output	Description
-> 1 - Open Browser		""		
-> 2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
-> 3 - Click	a_Login (9)			
-> 4 - Click	button_Click here for Admin Login (4)			
-> 5 - Set Text	input_Email_user (4)	"tiger@gmail.com"		
-> 6 - Set Encrypted Triput_Password_password (4)		"rHrA8Gqu/PiOMrMKSAnNNA="		
-> 7 - Click	button_Login (4)			
-> 8 - Click	button_MENU (2)			
-> 9 - Click	a_Add food (2)			
-> 10 - Click	input_Add Food_foodName (2)			
-> 11 - Set Text	input_Add Food_foodName (2)	"Chocolate Brownie"		
-> 12 - Set Text	input_Add Food_foodPrice (2)	"4.50"		
-> 13 - Click	input_Add Food_upload (2)			

Manual </> Script </> Variables </> Variables (Script mode) ■ Integration ■ Properties

Problems Event Log Console Log Viewer Self-healing Insights

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

```
06-29-2021 06:02:32 PM Test Cases/adding_new_product
Elapsed time: 15.361s
Test Cases/adding_new_product
```

- Test Cases/adding\_new\_product (15.361s)
  - 1 - openBrowser("") (3.731s)
  - 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.190s)
  - 3 - click(findTestObject("Object Repository/Page\_The Precinct Caf/a\_Login (9)")) (0.905s)
  - 4 - click(findTestObject("Object Repository/Page\_User and Admin Login/button\_Click here fo"))
  - 5 - setText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Email\_user (4)"), "tig")
  - 6 - setEncryptedText(findTestObject("Object Repository/Page\_The Precinct Caf/input\_Password"))
  - 7 - click(findTestObject("Object Repository/Page\_The Precinct Caf/button\_Login (4)")) (0.816s)
  - 8 - click(findTestObject("Object Repository/Page\_Admin Page/button\_MENU (2)")) (0.683s)
  - 9 - click(findTestObject("Object Repository/Page\_Admin Page/a\_Add food (2)")) (0.859s)
  - 10 - click(findTestObject("Object Repository/Page\_Admin Page/input\_Add Food\_foodName"))
  - 11 - setText(findTestObject("Object Repository/Page\_Admin Page/input\_Add Food\_foodName"), "Choc")
  - 12 - setText(findTestObject("Object Repository/Page\_Admin Page/input\_Add Food\_foodPrice"), "4.50")
  - 13 - click(findTestObject("Object Repository/Page\_Admin Page/input\_Add Food\_upload (2)"))

Figure 32 Test Case for Adding a new food item

## 8. Test Case for Updating the Price of a Food Item

Item	Object	Input	Output	Description
» 1 - Open Browser		""		
» 2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
» 3 - Click	a_Login (12)			
» 4 - Click	button_Click here for Admin Login (7)			
» 5 - Set Text	input_Email_user (7)	"tiger@gmail.com"		
» 6 - Set Encrypted Text	input_Password_password (7)	"rHrA8Gqu/PiOMrMKSAnNNA=="		
» 7 - Click	button_Login (7)			
» 8 - Click	button_MENU (5)			
» 9 - Click	a_DeleteUpdate Food (3)			
» 10 - Click	button_Update (2)			
» 11 - Set Text	input_Price_foodPrice (2)	"5.50"		
» 12 - Click	input_Cancel_update_btn (2)			
» 13 - Click	button_MENU (5)			
» 14 - Click	a_DeleteUpdate Food (3)			
» 15 - Close Browser				

Manual <> Script <> Variables <> Variables (Script mode) ■■■ Integration Properties

Problems Event Log Console Log Viewer Self-healing Insights

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

06-29-2021 06:17:02 PM Test Cases/update\_the\_price  
Elapsed time: 18.379s  
Test Cases/update\_the\_price

```

    > 1 - openBrowser("") (4.054s)
    > 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.269s)
    > 3 - click(findTestObject("Object Repository/Page_The Precinct Caf/a_Login (12)") (0.898s)
    > 4 - click(findTestObject("Object Repository/Page_User and Admin Login/button_Click here fo
    > 5 - setText(findTestObject("Object Repository/Page_The Precinct Caf/input_Email_user (7)"), "t
    > 6 - setEncryptedText(findTestObject("Object Repository/Page_The Precinct Caf/input_Password_password (7)"))
    > 7 - click(findTestObject("Object Repository/Page_The Precinct Caf/button_Login (7)") (0.884s
    > 8 - click(findTestObject("Object Repository/Page_Admin Page/button_MENU (5)") (0.575s
    > 9 - click(findTestObject("Object Repository/Page_Admin Page/a_DeleteUpdate Food (3)") (0.
    > 10 - click(findTestObject("Object Repository/Page_Admin Page/button_Update (2)") (0.969s
    > 11 - setText(findTestObject("Object Repository/Page_Admin Page/input_Price_foodPrice (2)"),
    > 12 - click(findTestObject("Object Repository/Page_Admin Page/input_Cancel_update_btn (2)"))
    > 13 - click(findTestObject("Object Repository/Page_Admin Page/button_MENU (5)") (0.599s
    > 14 - click(findTestObject("Object Repository/Page_Admin Page/a_DeleteUpdate Food (3)") (0.
    > 15 - closeBrowser() (0.876s)
  
```

Figure 33 Test Case for Updating the price of a food item

## 9. Test Case for Deleting a Food Item

Item	Object	Input	Output	Description
» 1 - Open Browser		""		
» 2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
» 3 - Click	a_Login (13)			
» 4 - Click	button_Click here for Admin Login (8)			
» 5 - Set Text	input_Email_user (8)	"tiger@gmail.com"		
» 6 - Set Encrypted Text	input_Password_password (8)	"rHrA8Gqu/PiOMrMKSAnNNA=="		
» 7 - Click	button_Login (8)			
» 8 - Click	button_MENU (6)			
» 9 - Click	a_DeleteUpdate Food (4)			
» 10 - Click	a_Delete			
» 11 - Close Browser				

Manual <> Script <> Variables <> Variables (Script mode) ■■■ Integration Properties

Problems Event Log Console Log Viewer Self-healing Insights

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

06-29-2021 06:22:57 PM Test Cases/deleting\_food\_item  
Elapsed time: 14.186s  
Test Cases/deleting\_food\_item

```

    > 1 - openBrowser("") (3.821s)
    > 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.253s)
    > 3 - click(findTestObject("Object Repository/Page_The Precinct Caf/a_Login (13)") (0.907s
    > 4 - click(findTestObject("Object Repository/Page_User and Admin Login/button_Click here fo
    > 5 - setText(findTestObject("Object Repository/Page_The Precinct Caf/input_Email_user (8)"), "t
    > 6 - setEncryptedText(findTestObject("Object Repository/Page_The Precinct Caf/input_Password_password (8)"))
    > 7 - click(findTestObject("Object Repository/Page_The Precinct Caf/button_Login (8)") (0.700s
    > 8 - click(findTestObject("Object Repository/Page_Admin Page/button_MENU (6)") (0.619s
    > 9 - click(findTestObject("Object Repository/Page_Admin Page/a_DeleteUpdate Food (4)") (1.
    > 10 - click(findTestObject("Object Repository/Page_Admin Page/a_Delete") (0.852s
    > 11 - closeBrowser() (0.896s)
  
```

Figure 34 Test Case for Deleting a food item

## 10. Test Case for Viewing orders from customers

Item	Object	Input	Output	Description
1 - Open Browser		""		
2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
3 - Click	a_Login (14)			
4 - Click	button_Click here for Admin Login (9)			
5 - Set Text	input_Email_user (9)	"tiger@gmail.com"		
6 - Set Encrypted Text	input_Password_password (9)	"rHrA8Gqu/PiOMrMKSAnNNA=="		
7 - Click	button_Login (9)			
8 - Click	button_MENU (7)			
9 - Click	button_MENU (7)			
10 - Click	a_View orders			
11 - Click	a_ViewOrder			
12 - Close Browser				

Manual Script Variables Variables (Script mode) Integration Properties

Problems Event Log Console Log Viewer Self-healing Insights

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

```

06-29-2021 06:29:55 PM Test Cases/view_orders
Elapsed time: 14.328s
Test Cases/view_orders

Test Cases/view_orders (14.328s)
> 1 - openBrowser("") (4.111s)
> 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.229s)
> 3 - click(findTestObject("Object Repository/Page_The Precinct Caf/a_Login (14)") (0.981s))
> 4 - click(findTestObject("Object Repository/Page_User and Admin Login/button_Click here fo"))
> 5 - setText(findTestObject("Object Repository/Page_The Precinct Caf/input_Email_user (9)"), "t")
> 6 - setEncryptedText(findTestObject("Object Repository/Page_The Precinct Caf/input_Password_"))
> 7 - click(findTestObject("Object Repository/Page_The Precinct Caf/button_Login (9)") (0.615s))
> 8 - click(findTestObject("Object Repository/Page_Admin Page/button_MENU (7)") (0.500s))
> 9 - click(findTestObject("Object Repository/Page_Admin Page/button_MENU (7)") (0.500s))
> 10 - click(findTestObject("Object Repository/Page_Admin Page/a_View orders") (0.837s))
> 11 - click(findTestObject("Object Repository/Page_Admin Page/a_ViewOrder") (0.753s))
> 12 - closeBrowser() (0.803s)

```

Figure 35 Test Case for viewing orders from customers

## 11. Test Case for Viewing customer inquiries

Item	Object	Input	Output	Description
1 - Open Browser		""		
2 - Navigate To Url		"http://localhost:88/theprecinctcafe/home.php"		
3 - Click	a_Login (15)			
4 - Click	button_Click here for Admin Login (10)			
5 - Set Text	input_Email_user (10)	"tiger@gmail.com"		
6 - Set Encrypted Text	input_Password_password (10)	"rHrA8Gqu/PiOMrMKSAnNNA=="		
7 - Click	button_Login (10)			
8 - Click	button_MENU (8)			
9 - Click	a_Customer inquiries	""		
10 - Open Browser				
11 - Close Browser				

Manual Script Variables Variables (Script mode) Integration Properties

Problems Event Log Console Log Viewer Self-healing Insights

Runs: 1/1 Passes: 1 Failures: 0 Errors: 0 Skips: 0

```

06-29-2021 06:55:17 PM Test Cases/viewing_customer_inquiries
Elapsed time: 17.069s
Test Cases/viewing_customer_inquiries

Test Cases/viewing_customer_inquiries (17.069s)
> 1 - openBrowser("") (3.891s)
> 2 - navigateToUrl("http://localhost:88/theprecinctcafe/home.php") (1.369s)
> 3 - click(findTestObject("Object Repository/Page_The Precinct Caf/a_Login (15)") (0.945s))
> 4 - click(findTestObject("Object Repository/Page_User and Admin Login/button_Click here fo"))
> 5 - setText(findTestObject("Object Repository/Page_The Precinct Caf/input_Email_user (10)"), "t")
> 6 - setEncryptedText(findTestObject("Object Repository/Page_The Precinct Caf/input_Password_"))
> 7 - click(findTestObject("Object Repository/Page_The Precinct Caf/button_Login (10)") (0.800s))
> 8 - click(findTestObject("Object Repository/Page_Admin Page/button_MENU (8)") (0.660s))
> 9 - click(findTestObject("Object Repository/Page_Admin Page/a_Customer inquiries") (0.998s))
> 10 - openBrowser("") (2.835s)
> 11 - closeBrowser() (1.256s)

```

Figure 36 Test Case for viewing customer inquiries

# Implementation

## Home Page

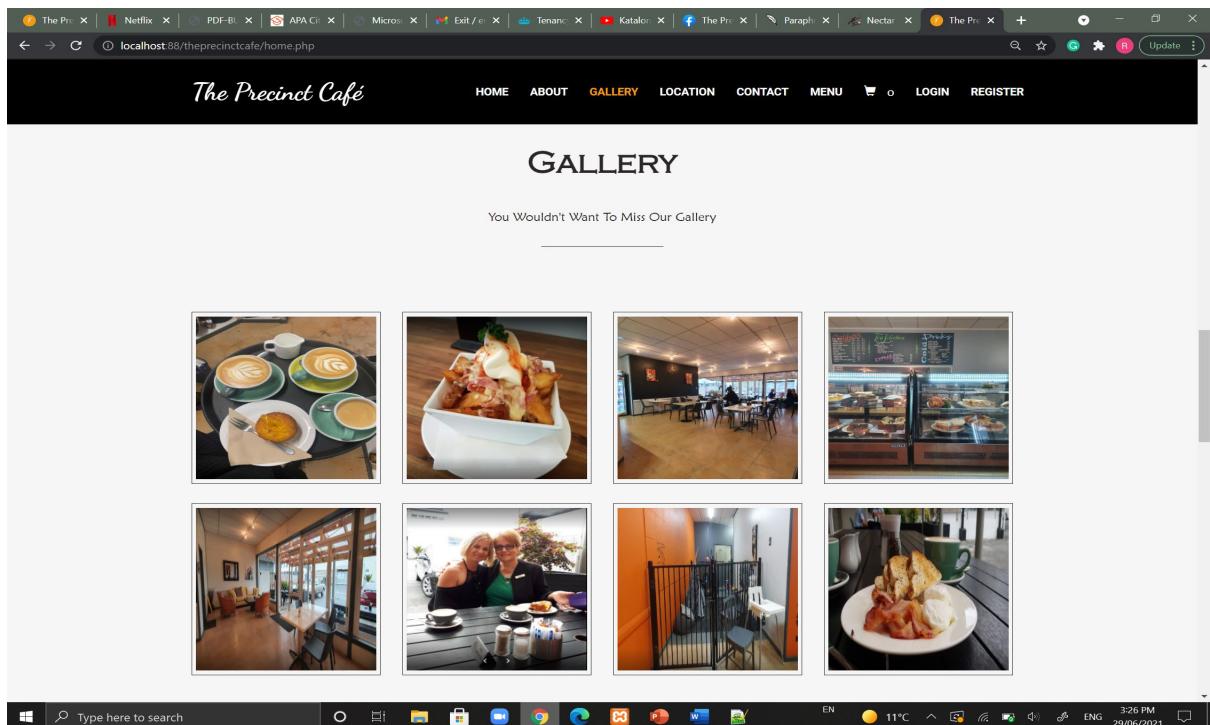
- *Main Page*



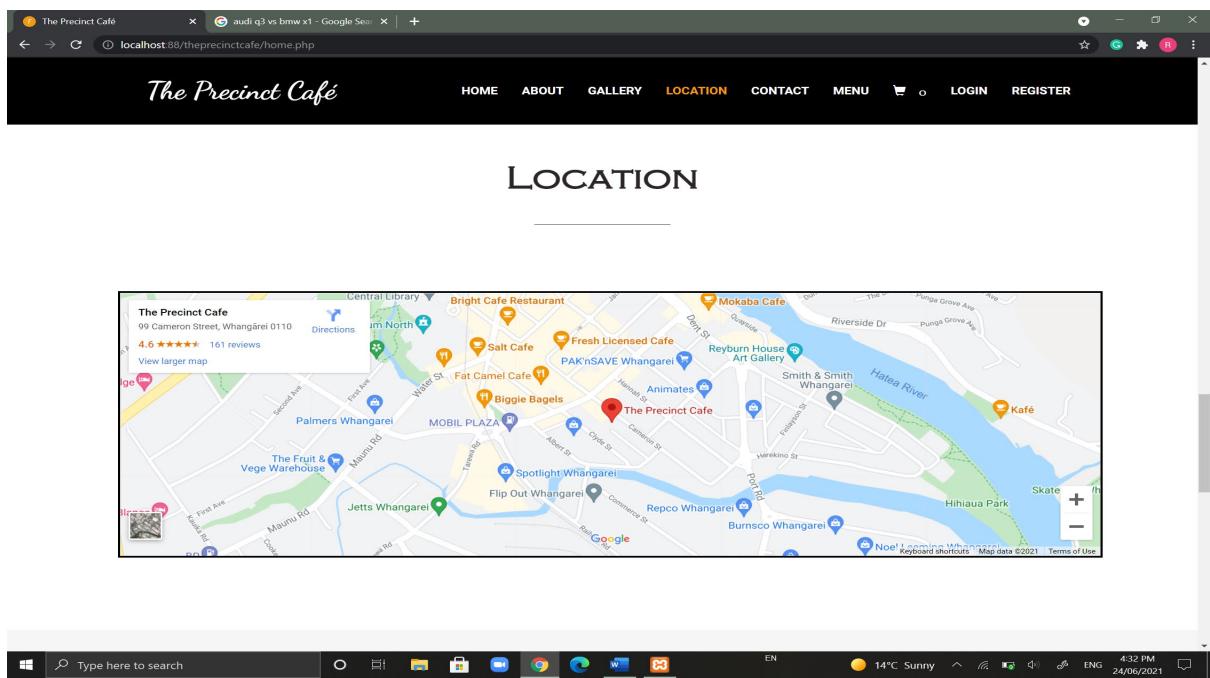
- *About us Page*

A screenshot of a web browser displaying the 'ABOUT US' page of 'The Precinct Café'. The title bar shows 'The Precinct Café' and the URL 'localhost:88/theprecinctcafe/home.php'. The page has a dark header with the 'The Precinct Café' logo and a navigation menu. Below the header, there is a section titled 'ABOUT US' with text about the cafe's atmosphere and ingredients. Further down, there is a section titled 'OPENING HOURS' with text indicating they are open from Monday to Friday from 7:00AM to 2:00PM. The bottom of the page shows a standard Windows taskbar with various icons.

- *Gallery Page*



- *Location*



- *Contact us Page*

The Precinct Café

HOME ABOUT GALLERY LOCATION CONTACT MENU ⚡ LOGIN REGISTER

## CONTACT US

Do you have any questions about the café? Please complete the quick form and we will be in touch with you.  
It's that easy!

---

Name

Email

Subject

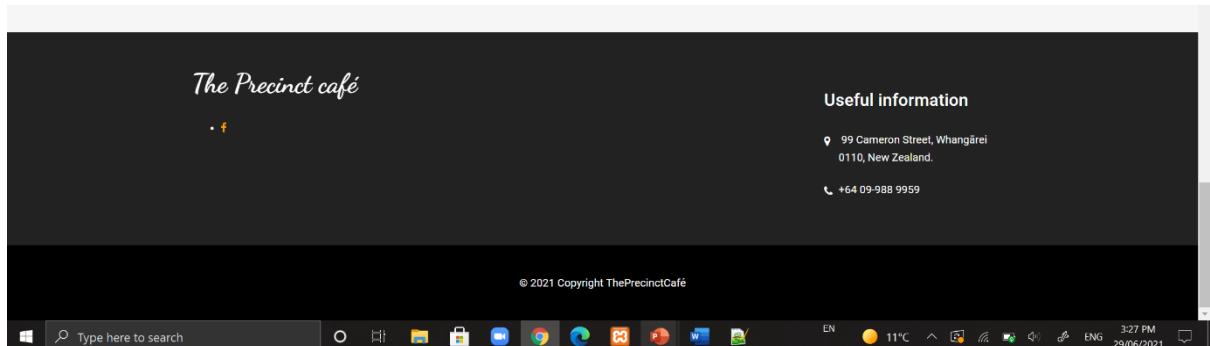
Message

**SUBMIT**

**Address**  
99 Cameron Street,  
Whangārei 0110, New Zealand.

**Phone**  
+64 09-988 9959

## Footer page



## User Login Page

The Precinct Café

HOME ABOUT GALLERY LOCATION CONTACT MENU ⚡ LOGIN REGISTER

**USER LOGIN**

Username

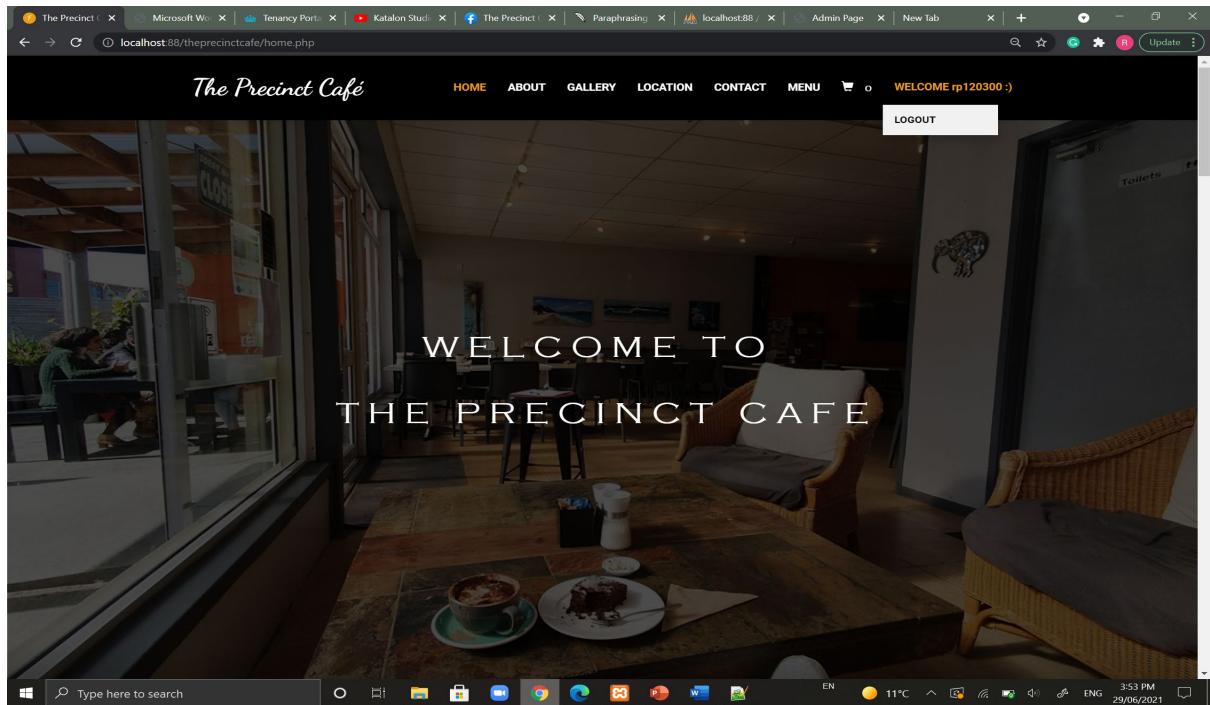
Password

**LOGIN**

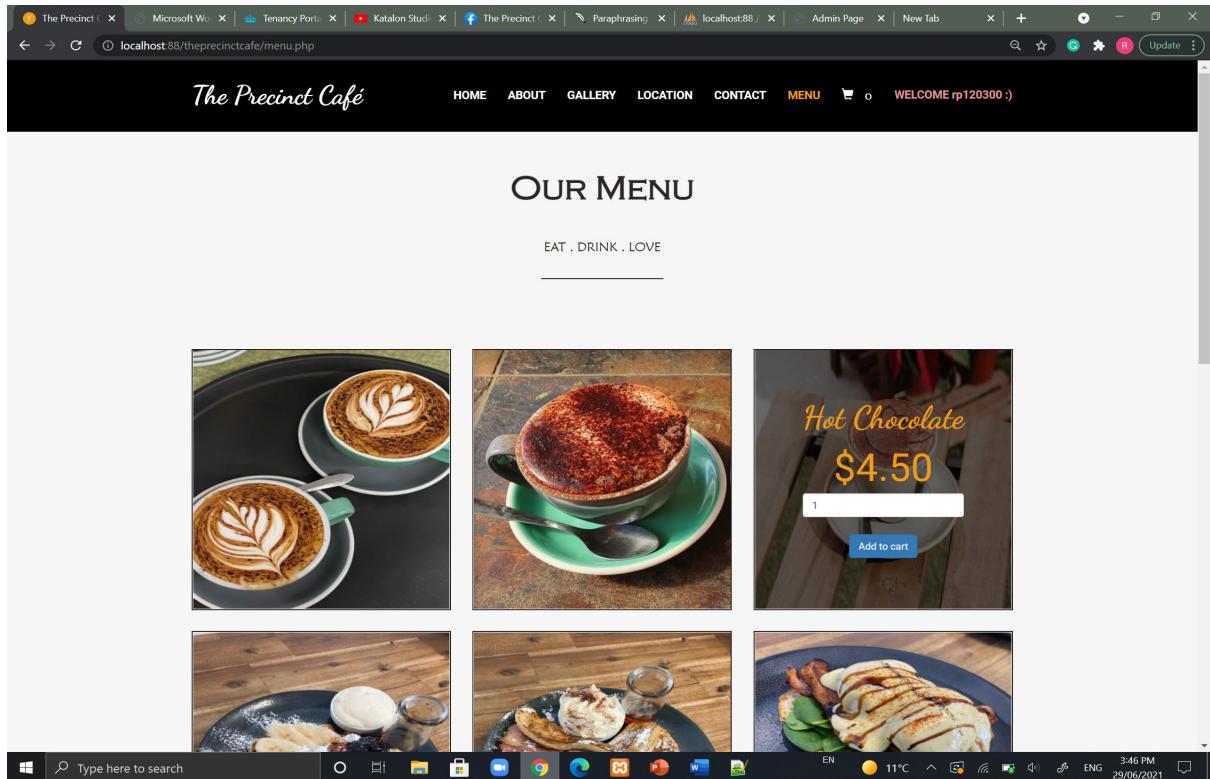
Do not have an account? Please [REGISTER](#)

**CLICK HERE FOR ADMIN LOGIN**

## Scenario when a user successfully logs in



## Menu Page



## Cart Page

The screenshot shows the Cart page of a website. At the top, there is a navigation bar with links for HOME, ABOUT, GALLERY, LOCATION, CONTACT, MENU, LOGIN, and REGISTER. The main content area has a title 'CART' and a table showing items in the cart:

Name	Quantity	Price	Total
Hot Chocolate	1	\$4.50	4.50
Eggs and Bacon Benedict	1	\$14.50	14.50

A total amount of \$19.00 is displayed at the bottom. A 'CHECKOUT' button is centered below the table.

## Checkout Page

- Personal Details Form

The screenshot shows the Checkout page. The navigation bar includes a welcome message 'WELCOME rp120300 :)'. The main content area has a title 'CHECKOUT' and a section titled 'PERSONAL DETAILS' containing fields for Full Name, Email, Phone Number, and Address. The phone number field contains '(022)-1111111'.

- *Payment Method*

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is 'localhost:88/theprecinctcafe/checkout.php'. The page content is titled 'PAYMENT METHOD' and contains two buttons: 'Pay by Paypal' and 'Pay by Card'. Below these buttons, it says 'Accepted Cards' and shows icons for Visa, MasterCard, American Express, and Maestro. At the bottom of the page, there is a section titled 'Useful information' with a list of contact details: address (99 Cameron Street, Whangārei 0110, New Zealand), phone number (+64 09-988 9959), and a small orange icon.

- *PayPal Payment Page*

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is 'paypal.com/webapps/shoppingcart?mfid=1624939458872\_6ac246a82d701&flowlogging\_id=6ac246a82d701#/checkout/shoppingCart'. The page title is 'Your shopping cart'. It displays a table with one item: 'The Precinct cafe' at \$32.00 quantity 2, totaling \$64.00. To the right is an 'Order summary' table showing 'Item total \$64.00' and 'Total \$64.00 NZD'. Below the tables are two large buttons: a yellow 'PayPal Check Out' button and a grey 'Check Out' button. A note below the buttons says 'Pay without a PayPal account'. At the bottom of the page is a 'Powered by PayPal' logo and a small footer with links to 'Terms', 'User Agreement', 'Privacy', 'Feedback', and copyright information. The status bar at the bottom of the screen shows the date and time as 29/06/2021 4:04 PM.

- Pay by Card Page

The Precinct Café

HOME ABOUT GALLERY LOCATION CONTACT MENU WELCOME rp120300 :)

**CARD DETAILS**

TOTAL PAYABLE AMOUNT:\$ 19

Name on Card  
John More Doe

Card Number  
1111-2222-3333-4444

Expiry Date  
month/year(12/28)

CVV  
312

Purchase

## Admin's Login Page

The Precinct Café

HOME

**ADMIN LOGIN**

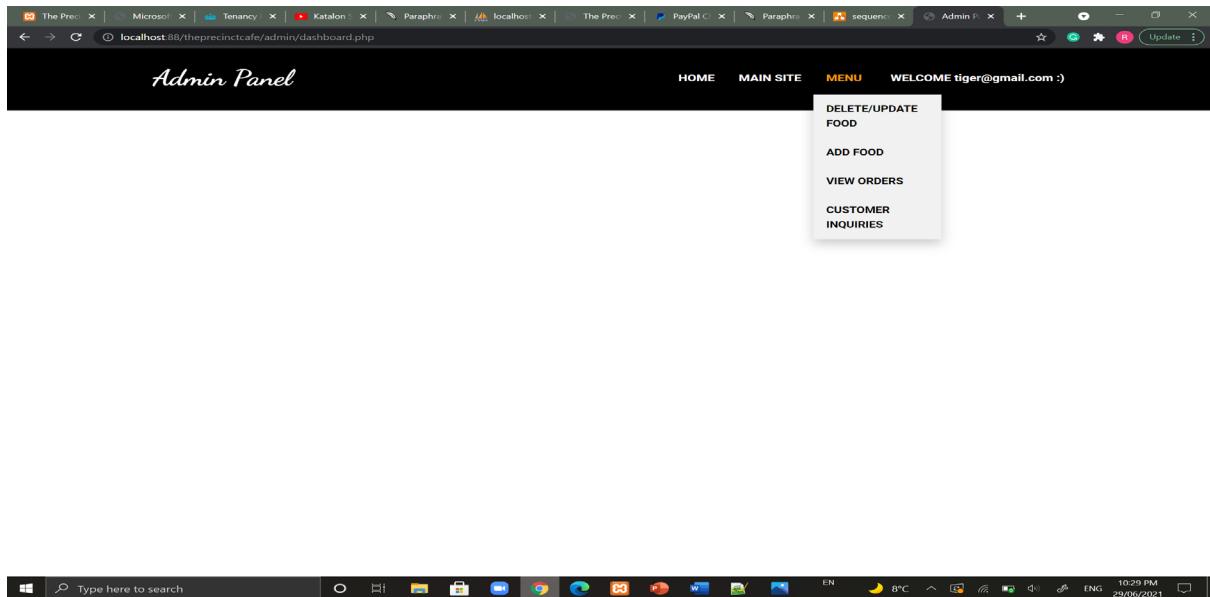
Email

Password

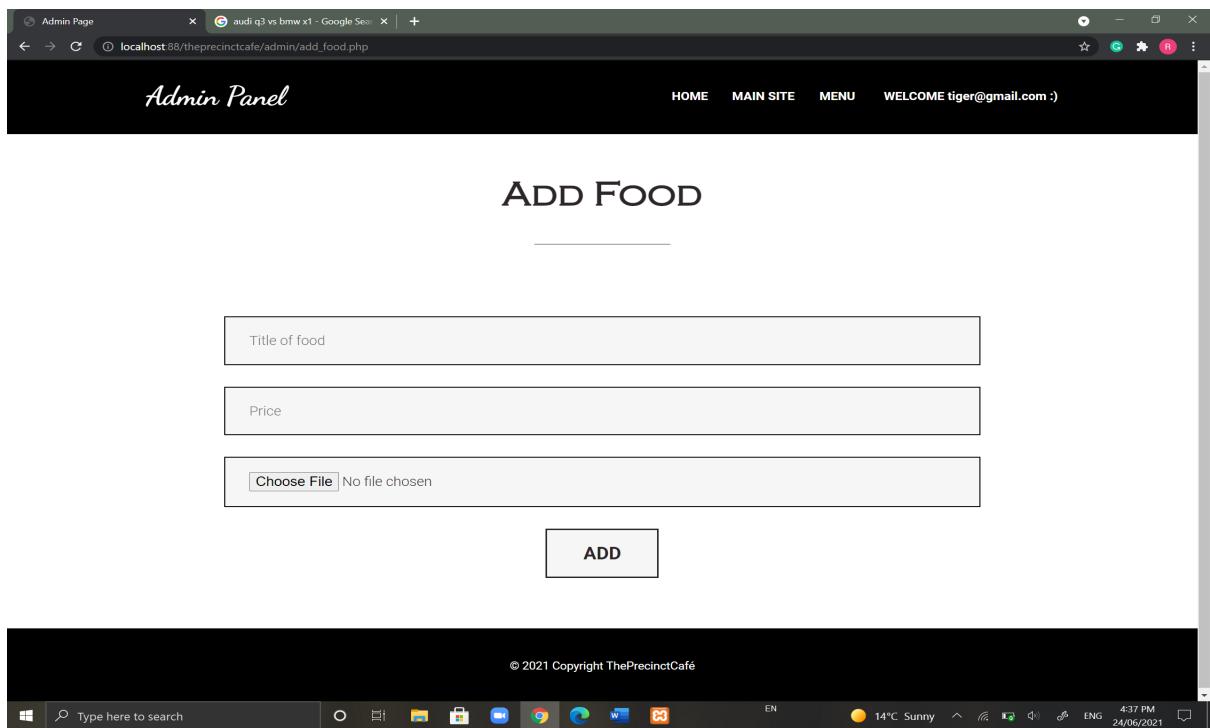
Login

© 2021 Copyright ThePrecinctCafé

## Admin's Dashboard Page



## Add Food Page



## Update/Delete Food Page

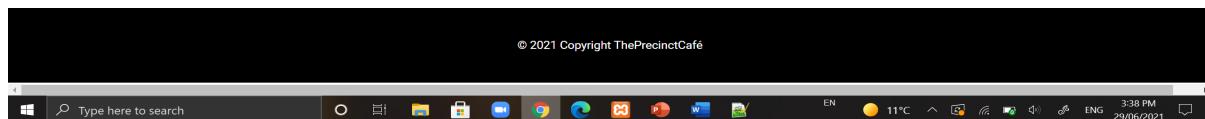
- *Products details Page*

Name	Price	Image	Actions
Cappuccino	\$4.50		<button>UPDATE</button> <button>DELETE</button>
Mochaccino	\$4.50		<button>UPDATE</button> <button>DELETE</button>
Hot Chocolate	\$4.50		<button>UPDATE</button> <button>DELETE</button>

- *Update the Price Page*

Name	Cappuccino
Current Image	
Price	<input type="text" value="4.50"/>

Cancel SUBMIT



## View Orders Page

- *Customer Details Page*

The screenshot shows a web browser window titled "localhost:88/theprecinctcafe/admin/manage\_food.php". The page has a header "Admin Panel" and navigation links "HOME", "MAIN SITE", "MENU", and "WELCOME tiger@gmail.com :)". The main content is titled "CUSTOMER DETAILS" and displays a table of customer information:

Customer ID	Customer Name	Customer Email	Customer Phone	Customer Address	Actions
33	Nucho	nucho@gmail.com	1234562344	wdcdwdefd	<button>VIEW ORDER</button>
34	Zed	zed@gmail.com	225714700	qdnfdewfd	<button>VIEW ORDER</button>
35	Zayn	zayn@gmail.com	225613400	17 BK Street, OR,NZ	<button>VIEW ORDER</button>

- *Customer order details page*

The screenshot shows a web browser window titled "localhost:88/theprecinctcafe/admin/view.php?id=35". The page has a header "Admin Panel" and navigation links "HOME", "MAIN SITE", "MENU", and "WELCOME tiger@gmail.com :)". The main content is titled "VIEW ORDERS" and displays a table of order details:

Order ID	Product Name	Product Price	Product Quantity	Product Total	Payment
22	Hot Chocolate	4.50	1	4.5	Paid
23	Eggs and Bacon Benedict	14.50	1	14.5	Paid
24	Mochaccino	4.50	3	13.5	Paid

## Customer Inquiries Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is 'localhost:88/theprecinctcafe/admin/contact.php'. The page title is 'Admin Panel' and the main content is 'CUSTOMER INQUIRIES'. Below this is a table with four columns: 'Customer Name', 'Customer Email', 'Subject', and 'Message'. The table contains three rows of data:

Customer Name	Customer Email	Subject	Message
Rushit	rushit12all@gmail.com	Cmon	this is test
Zed	rushit12all@gmail.com	test	closing time pls
Zayn Malik	zayn@gmail.com	This is test	I hope it is successfull

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar with 'Type here to search', and icons for various applications like File Explorer, Edge, and File History. System status icons include battery level (8%), temperature (8°C), signal strength, and date/time (10:31 PM, 29/06/2021).

## Conclusion

The project objectives have been met. The project successfully constructed a client/server application for a food ordering system using several programming tools such as HTML, CSS, JavaScript, PHP, and others. It also gives a far more convenient and accurate manner for manager-related tasks such as viewing orders, adding food items, altering prices, and so on. This website was designed in such a way that all of the issues that customers and clients were experiencing were remedied.

### Future Works

The system will be much more developed in terms of UI, with plenty more animations on the home page that are much smoother, the ability to edit personal information from the account itself once successfully logged in, and an order tracking system built into the website that will allow users to view the exact details of where their ordered food items are in the kitchen. For improved mobility and user convenience, an Android and iOS app for the Precinct Café will be developed. Managers will also have sophisticated capabilities in their Admin Panel, such as automated sales counting, a graphical representation of daily sales in terms of online orders, and a more user-friendly interface.

## References

- ProductPlan. (2021, June 15). *What is the Software Development Lifecycle? | Definition and Overview.* <https://www.productplan.com/learn/software-development-lifecycle/>
- Altvater, A. (2021, March 30). *What Is SDLC? Understand the Software Development Life Cycle.* Stackify. <https://stackify.com/what-is-sdlc/>
- Requirements Document Example. (n.d.). Web.Cse. Retrieved June 19, 2021, from [http://web.cse.ohio-state.edu/%7Ebair.41/616/Project/Example\\_Document/Req\\_Doc\\_Example.html](http://web.cse.ohio-state.edu/%7Ebair.41/616/Project/Example_Document/Req_Doc_Example.html)
- Techopedia. (2013, January 26). *Three-Tier Client/Server.* Techopedia.Com. <https://www.techopedia.com/definition/23813/three-tier-clientserver>
- Raj, P. (n.d.). *Architectural Patterns.* O'Reilly Online Learning. Retrieved June 21, 2021, from <https://www.oreilly.com/library/view/architectural-patterns/9781787287495/df0e98f1-0190-42e2-a9b1-69f050a03a4e.xhtml>

