

Tang Huat Restaurant Online Food Ordering System

Yang Jeng Foong¹, Mohd Zainuri Saringat^{1*}

¹Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, Malaysia

DOI: <https://doi.org/10.30880/aitcs.2024.05.01.053>

Received 24 June 2023; Accepted 18 June 2024; Available online 30 August 2024

Abstract: Online Food ordering system is a web-based system that allows the consumer to order food from the Internet using a web browser. TANG HUAT RESTAURANT is a food-selling business that sells a variety of food using WhatsApp and Telegram. The problems faced by the owner are difficulty tracking back orders, the need to calculate the monthly sales manually and not having a proper online menu. The objective of this system is to design an online food ordering system using an object-oriented approach, to develop the system and to test system. The system was developed based on the Iterative methodology. The system was launched on the web server and live on the Internet so that the customer can order food using their web browser. This system is expected to help customers to save time when ordering food and owner or staff to manage customers' orders efficiently.

Keywords: Food Ordering System, Web-based System

1. Introduction

Since the Covid-19 pandemic, e-commerce food and drink have become famous among consumers [1]. Online food ordering involves a customer searching for a restaurant on the web using electronic devices and redirecting the customer to a website where the customer can place food orders [3]. The online food ordering system usually provides a food menu so customers can easily place an order.

Tang Huat Restaurant is a food selling business that located in Muar, Johor. However, the physical shop is closed due to the pandemic and the owner has now switched the business to social media platforms like WhatsApp and Telegram. Now, Customers can contact the owner using WhatsApp or make phone calls to place orders. There are some problems that are faced by the owner which are difficult to track back orders because the owner record order inside a book using pen, needs to calculate the monthly sales manually using calculator and do not have a proper online menu hence the owner need to send the food image every time the customer requested for it.

The objective of this project is to design an online food ordering system using an object-oriented approach, develop the system using web-based approach and test the system using alpha testing. The targeted user of the system are the admin, staff and customer. This system also consists of several

*Corresponding author: zainuri@uthm.edu.my

| This is an open access article under the CC BY-NC-SA 4.0 license.

module which are register, login, manage order, make payment, manage food menu, manage user account, manage account details and generate report.

2. Related Work

2.1 Online Ordering System

Rapid growth in Internet use and associated web technologies have led many businesses to adapt to web applications that allow a broader target consumer[4]. This trend also applies to the Food and Beverage (F&B) Industry, as many Food delivery services have transitioned from telephone to digital ordering to satisfy customer needs and allow consumers to order products online [6]. This is made achievable through the use of a payment gateway. The customer can make payment through a credit card or debit card. Hence, everyone can order goods from anywhere on the Internet and deliver the products to the customers' houses.

An online ordering system is a tool that allows the consumer to order preferred food from the Internet using a web browser such as Google Chrome, Brave, Mozilla Firefox, and many more., from a consumer's electronic device like a laptop or mobile phone. The online ordering system can speed up the customers' food ordering process and provide an online platform that generates new opportunities for the food industry to attract more consumers.

2.2 System Comparison

In this study three existing system has been choose and used as a reference in developing the proposed system. The three existing system are Foodpanda, KFC Malaysia Online Food Ordering Website and GloriaFood. Table 1 shows comparison between the three existing systems with the proposed system.

Table 1: Comparison Table between Existing System and Proposed System

| Function/System | Foodpanda | KFC Malaysia Online Food Ordering Website | GloriaFood | Tang Huat Restaurant Online Food Ordering System |
|--------------------------|--------------------------------|---|--------------------------------|--|
| Register and Login | Yes | Yes | Yes | Yes |
| Menu Management | Yes | Not Available | Yes | Yes |
| Report Generation | Yes | Not Available | Yes | Yes |
| Tracking System | Yes | Yes | No | No |
| Display Order Status | No | No | Yes | Yes |
| Make Payment | Online, cash on delivery | Online, cash on delivery | Online, cash on delivery | Online, cash on delivery |
| Manage User | Yes | Not Available | Yes | Yes |
| Mange Account Details | Yes | Yes | Yes | Yes |

3. Methodology

SDLC is the process of developing a system with proper requirements analysis, design, implementation, testing, debugging, and maintenance [2]. The methodology used to develop this system is Iterative model. The Iterative Model is a model that divides the system into smaller chunks and works on an iterative basis in the software development process and drives the process in cycles such that each phase is repeated after another [5].

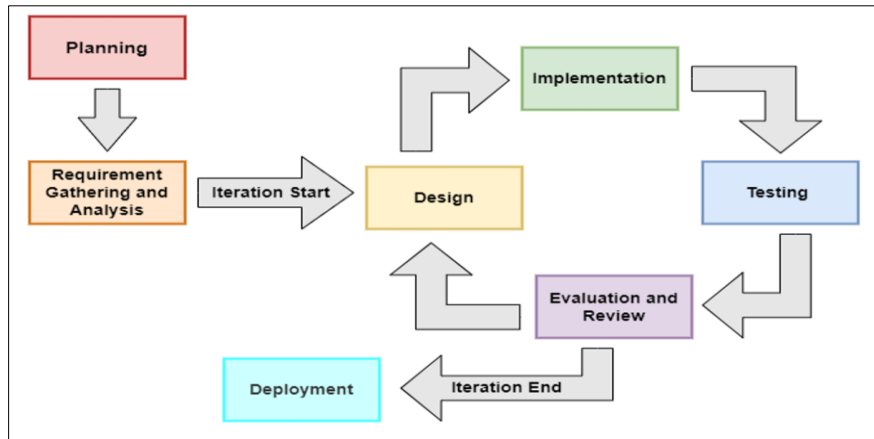


Figure 1: Iterative Model

3.1 Iterative Model Phases

Table 2 shows the list of activities that is done in each phase of the Iterative model for the Tang Huat Restaurant Online Ordering System.

Table 2: Activities in each Iterative model phases

| Phase | Task/Activities |
|------------------------------------|---|
| Planning | <ul style="list-style-type: none"> • Initial meeting and conversation with the owner • Identify problems, objectives, scopes, project significant and expected output. • Identify relevant stakeholders • Create a project workflow and timeline • Feasibility study • Document current business process and system |
| Requirement Gathering and Analysis | <ul style="list-style-type: none"> • Prepare Interview Questions • Interview session with owner • Requirement gathering • Analysis requirements • Document requirements |
| Iteration Start | |
| Design | <ul style="list-style-type: none"> • Design the UML diagrams for the proposed system • Identify the suitable software and hardware requirements • Design database structure • Design system’s user interface • System Design validation with owner • Document software design |
| Implementation /Development | <ul style="list-style-type: none"> • Develop the system module based on the SRS and SDD document (in each iteration) • Develop system’s database • Initial testing and debugging |

Table 2: (cont)

| | |
|----------------------|---|
| Testing | <ul style="list-style-type: none"> • Create test case based on SRS and SDD document • Test system’s module developed based on the test case created (in each iteration) • Document Test Result • Debugging system’s errors • Final Testing (after all module complete developed) |
| Evaluation / Review | <ul style="list-style-type: none"> • System validation by owner • Record Changes (if any) • Document Validation Result |
| Iteration End | |
| Deployment | <ul style="list-style-type: none"> • Deploy system to a web hosting server |

3.2 Use Case Diagram

Use Case diagram is a diagram that summarizes the details of the system's users, also known as actors, and their interactions with the system modules [7]. Figure 2 shows the use case diagram of the Tang Huat Online Food Ordering System. Based on the use case diagram, there are three types of users which are Admin, Staff, and Customer.

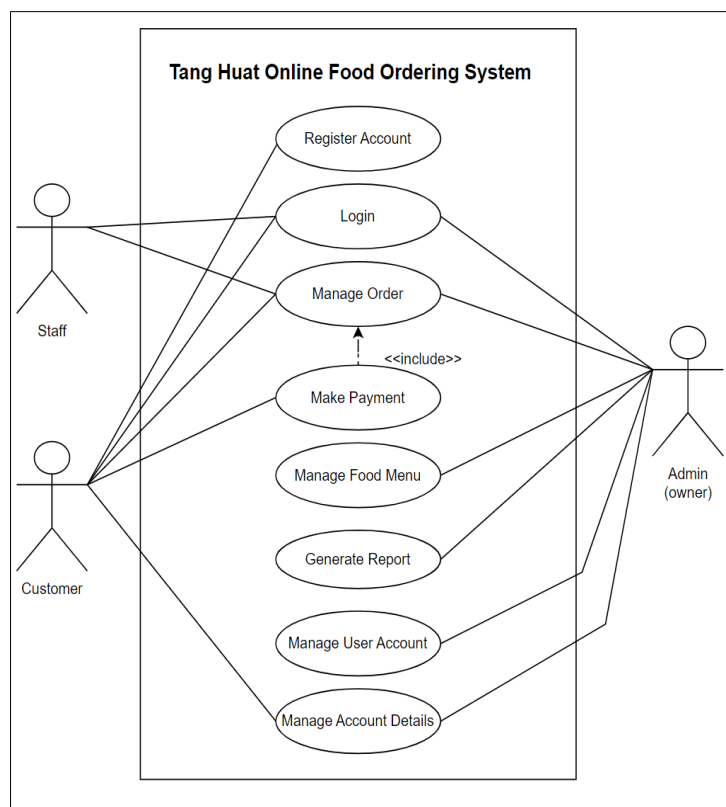


Figure 2: Use Case Diagram of Tang Huat Restaurant Online Food Ordering System

3.3 System Functional Requirements

Functional requirements are features or functions of a system or components that must be implemented to enable the users to fulfill their tasks or business process. Table 3 discussed and explained the functionality of each system module.

Table 3: Functional Requirements of the Tang Huat Restaurant Online Food Ordering System

| Module | Functionalities |
|------------------------|---|
| Register | 1. The system should allow new customer to register into the system by using their email address |
| Login | 1. The system should allow admin, staff and customer to log into the system using their email and password |
| Manage Order | 1. The system should allow the customer to add, remove or edit quantity of the food items in their cart. 2. The system should allow staff and admin to update the order status. 3. The system should allow the customer to view their order status. |
| Make Payment | 1. The system should allow the customer to check out the food items in the cart 2. The system should allow the customer to choose the payment methods |
| Manage Food Menu | 1. The system should allow admin to add, edit or remove food category 2. The system should allow admin to add, delete or edit food items. |
| Generate Report | 1. The system should display the sale report based on the dates input by admin |
| Manage User Account | 1. The system should allow admin to register new staff or delete existing staff account 2. The system should allow admin to update customer account status |
| Manage Account Details | 1. The system should allow the customer to change their account password 2. The system should allow the customer to edit their profile details 3. The system should allow admin to change the account password |

3.4 Non-Functional Requirement

Non-functional requirements, also known as quality requirements describe how a system must behave and its quality attribute.

Table 3: Non-Functional Requirements of the Tang Huat Restaurant Online Food Ordering System

| Requirement | Description |
|--------------------|---|
| Operational | 1. The system should be easy to use 2. The system should be able to work any web browser such as Google Chrome, Brave or Internet Explorer |
| Performance | 1. The system should be accessible at all time 2. Any interaction between user and the system should be longer than 3 second |
| Security | 1. The system shall deny access of any user if the email and password input is incorrect. 2. The staff can only allow to manage order 3. Customer must log into their account before allow to start ordering food |
| Usability | 1. The general appearance and the flow of the system are easily understood by the user |
| Integrity | 1. The password store in the database should be encrypted |

3.5 Database

In this section, the Tang Huat Restaurant Online Food Ordering System's database design is illustrated and discussed using a class diagram. Figure 3 shows the class diagram of the Tang Huat Restaurant Online Food Ordering System. The class diagram shows the entity inside the database and their relationship.

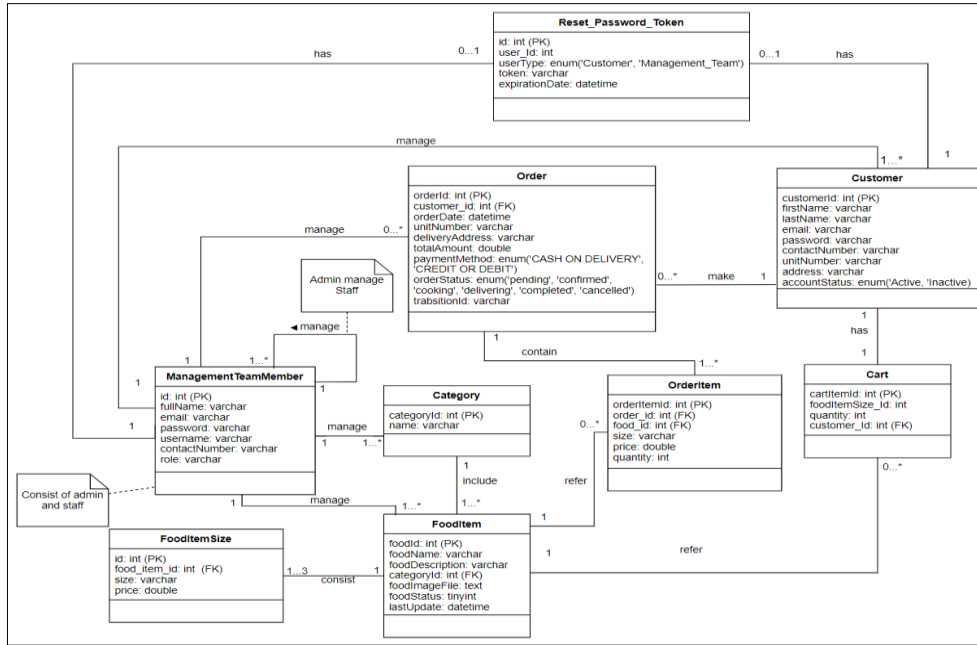


Figure 3: Class Diagram of Tang Huat Restaurant Online Food Ordering System

3.6 General System Architecture

Figure 4 shows the system architecture of the Tang Huat Restaurant Online Food Ordering System. The system is a web-based system that will be launched on the Internet. The admin can log in to his account without registering a new one since the admin already has a pre-set-up account inside the system database. The admin can use the system to manage the food menu, order, user account, and account details, register account for the staff and generate a sale report. Staff consist of the helper for the admin(owner). Hence, the staff is only allowed to manage the customer orders. Meanwhile, customers can visit the proposed system to view the food menu. If the customers want to proceed with the ordering and payment, they must first log into their accounts. However, if is the first time visiting, then the customer will need to register an account.

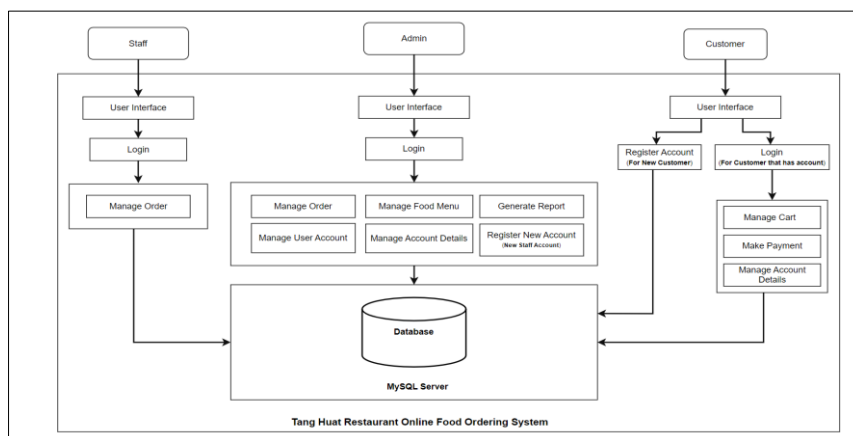


Figure 4: System Design Diagram for Tang Huat Restaurant Online Food Ordering System

3.7 User Interface

User Interface or UI is the point or place where human-computer interaction occurs. For this proposed system, the user will interact with the proposed system through these web pages.

3.7.1 Customer Interface

Figure 5 shows the customer register page. The customer that does not have an account must first create a new account at this page. Figure 6 shows the customer login page, customer must log onto the account before allow to start ordering foods.

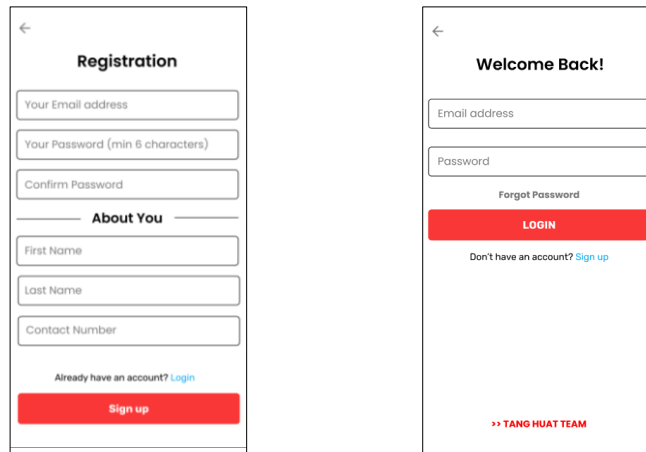


Figure 5: Customer Register Page Figure 6: Customer Login Page

Figure 7 shows the food item details page, this page is responsible to display the details of a food item such as size available and associated price. Customer can select the preferred size and quantity and click the add button to add the food item into the cart. Figure 8 shows the customer cart page which display the food items added into the cart. Figure 9 shows the customer checkout page where the customer can input the delivery address and select the payment methods.

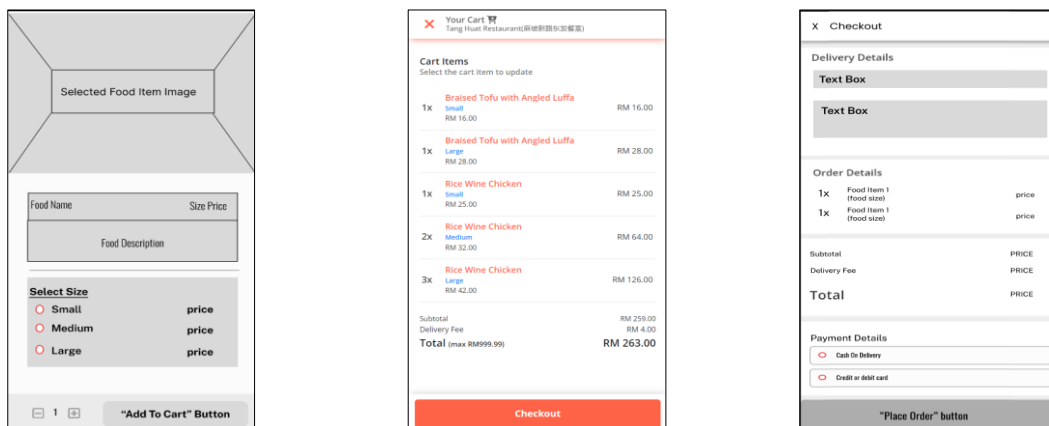


Figure 7: Food Item Details Page Figure 8: Customer Cart Page Figure 9: Checkout Page

Figure 10 shows the customer change profile page which allow the customer to change first name, last name, contact number and delivery address. Figure 11 shows the customer change password page, the customer needs to enter valid current password to reset the password.

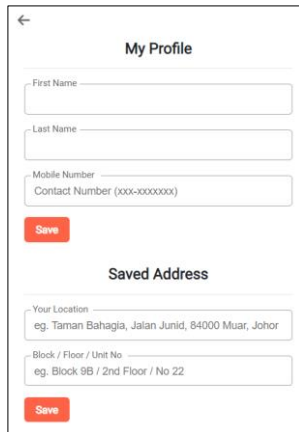


Figure 10: Change Profile Page

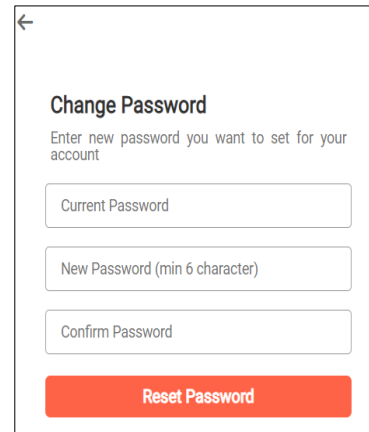


Figure 11: Change Password Page

3.8.2 Admin Interface

Figure 12 shows the admin and staff login page. Different with customer login page, this page is specifically for staff and admin login process. Figure 13 shows admin dashboard page, the system will redirect the admin to this page if the login process has succeeded.

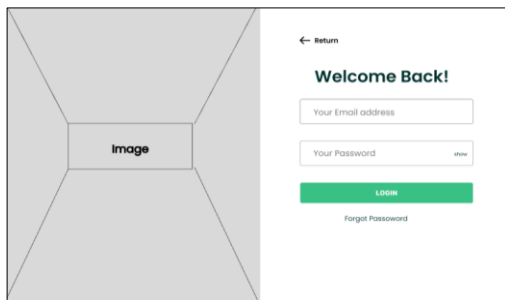


Figure 12: Admin and Staff Login Page

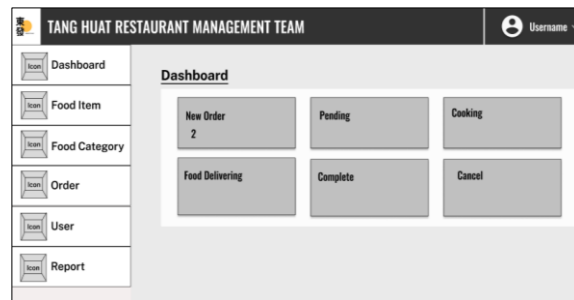


Figure 13: Admin Dashboard Page

Figure 14 shows food item page which the admin can manage the food items information such as add new food item, delete or edit existing food item. Figure 15 shows the food category page which the admin can manage the food category such as add, edit and delete.

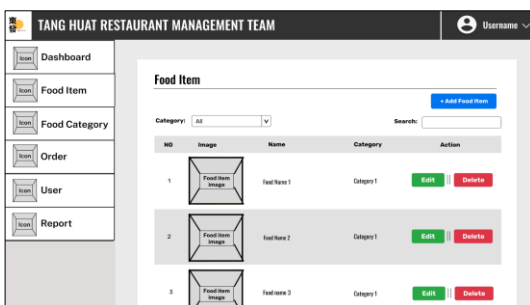


Figure 14: Food Item Page

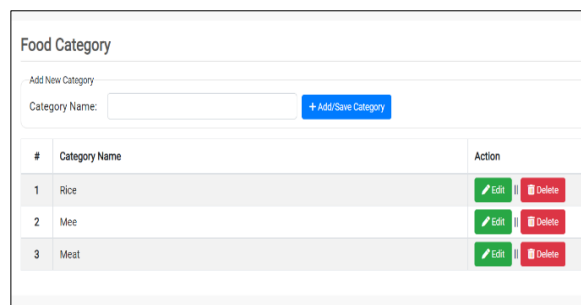


Figure 15: Food Category Page

Figure 16 shows the order page which display the orders placed by customers. Figure 17 shows the order details page where the information of an order is displayed.

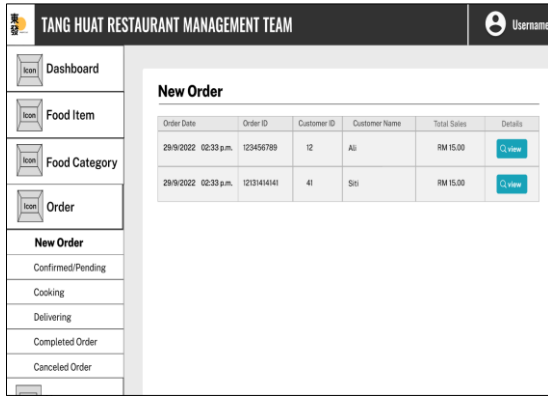


Figure 16: Order Page

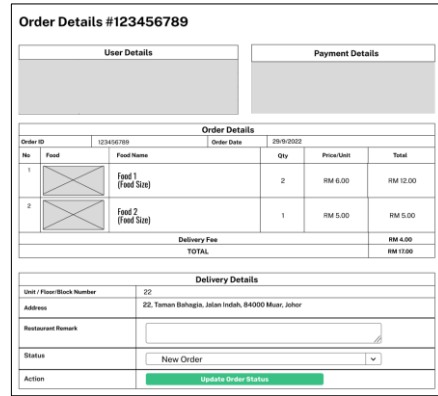


Figure 17: Order Details Page

Figure 18 shows the customer management page where admin can view and manage the customer account. Figure 20 shows the staff management page where admin can view and manage the staff account.

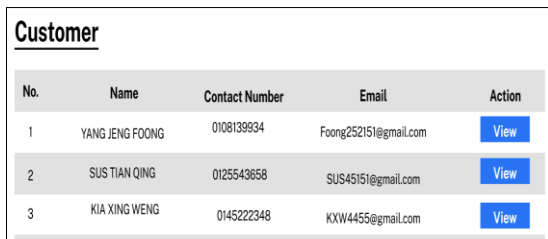


Figure 18: Customer Management Page

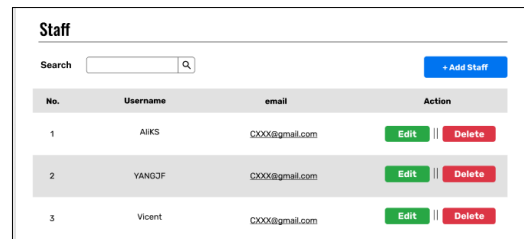


Figure 19: Staff Management Page

Figure 20 shows the sale report page where admin can input the start and end date to generate a sale report. Figure 21 shows the change password page where admin and staff can change the account password

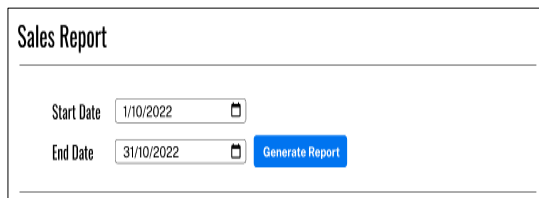


Figure 20: Sale Report Page

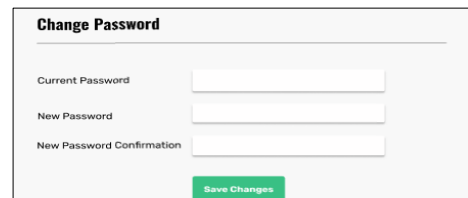


Figure 21: Change Password page

3.8.2 Staffs Interface

Figure 22 shows the staff dashboard page where the side navigation bar has only navigation item for manage order.

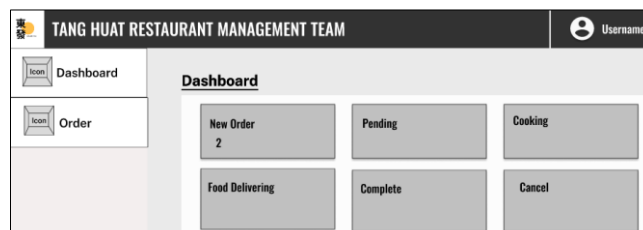


Figure 22: Staff Dashboard Page

4. Results and Discussion

The system was primarily developed using PHP, with frontend development done using HTML, CSS, and JavaScript. The most important development tools were Visual Studio Code for editing code, phpMyAdmin for managing MySQL databases, XAMPP for local testing.

4.1 Register Account Module

In this system, the register account module is responsible for allowing new customers who do not have an account in this system to register for a new account. Customers need to have an account to start ordering food. To register an account, the user needs to enter their email, password, confirm password, first name, last name, and contact number on the registration page. Figure 23 shows the interface of the register account page, where the user needs to fill in the text boxes. Figure 24 and Figure 25 show the code segments for the registration module.

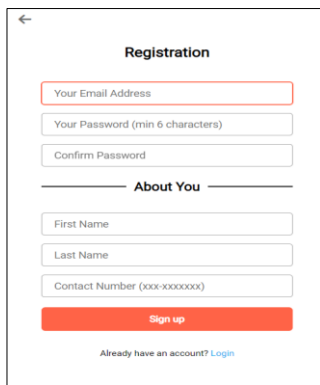


Figure 23: Register Account Interface



Figure 24: Register Module Code Segment (a)



Figure 25: Register Module Code Segment (b)

4.2 Login Module

In this system, the login module is responsible for authenticating the identity of users so that the system can restrict and authorize the website user from accessing modules of the websites. The customer needs to input valid email and password to log into their account. Figure 26 shows the interface for login page, where the user needs to enter their registered email and password to log into the account. Figure 27 shows the code segment for the login module.

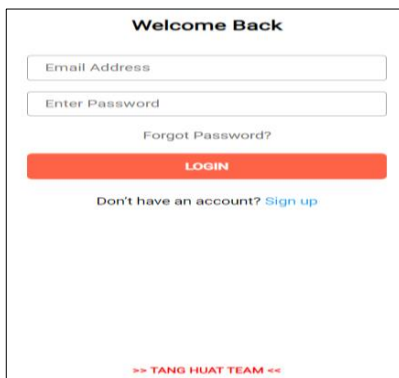


Figure 26: Login Interface

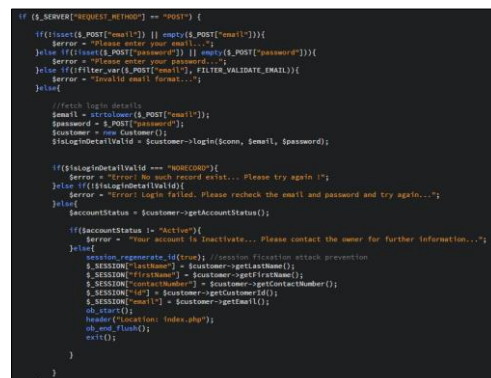


Figure 27: Login Module Code Segment

Figure 28 shows the admin and staff login interface. The admin and staff need to input valid email and password to log into their account. Figure 29 shows the code segment for admin and staff login module.

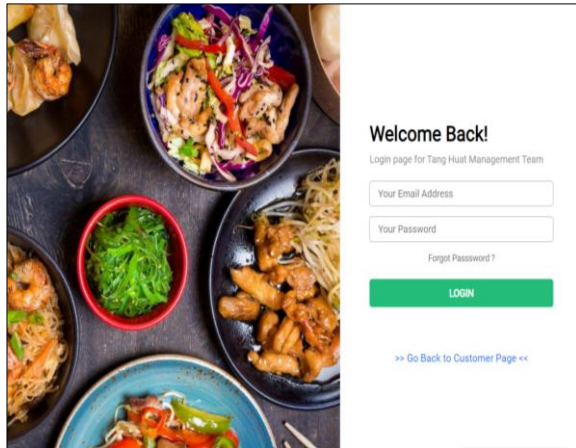


Figure 28: Admin and Staff Login Interface



Figure 29: Admin and Staff Login Module Code Segment

4.3 Manage Order Module

The manage order module can be divided into two different situations. One is customer manage the cart item in their cart like add, delete or edit the cart item details. Another situation is where the admin and staff can manage the order place by the customer by update the order status. Figure 30 shows the add food item to cart interface where the customer can select the preferred food size and quantity. After select the size and quantity the customer can click the Add to Cart button to add the cart item. Figure 31 shows the cart interface which display the cart items inside the customer’s cart. At the cart page, the customer can click the cart item to redirect to cart item details page for editing. Figure 32 shows the Edit Cart Item interface where the customer can edit the size or the quantity of the cart item. To delete or remove the cart item, the customer must decrement the quantity of the cart item to 0 and click the delete item to remove the item from the cart.

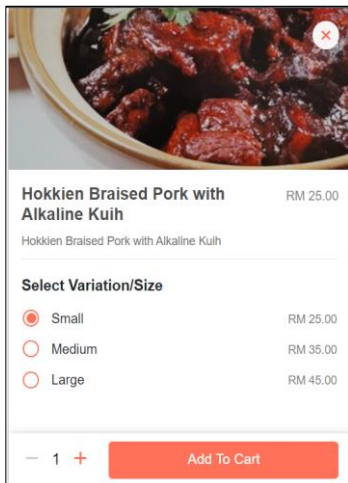


Figure 30: Add Food Item to Cart Interface

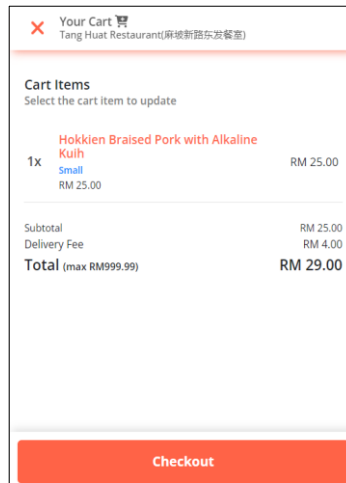


Figure 31: Cart Interface

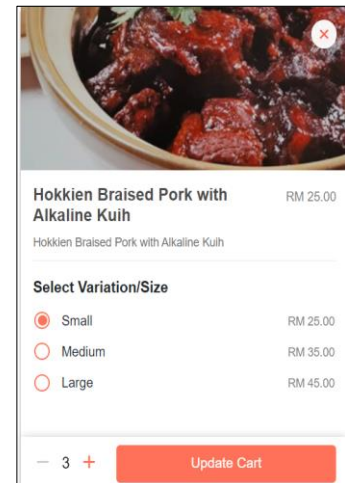


Figure 32: Edit Cart Item Interface

Figure 33 shows the code segment for adding food item into the cart. Figure 34 shows the code segment for remove or edit the cart item details which the PHP code will determine whether the quantity of the cart item is 0. If the quantity is 0, then the system will remove the item from the customer’s cart.

```

//if the user want to start order, the server check if there is a post request
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $cartItem = new Cart();
    $cartItem->setFoodItemSize_Id($_POST["foodItemSizePriceId"]);
    $cartItem->setQuantity($_POST["quantity"]);
    $cartItem->setCustomer_Id($_SESSION["id"]);

    //check if the current food item with associated size is exist in the cart
    if($cartItem->checkCartItemExist($conn)){
        //if exist then edit/update the quantity
        $cartItem->editCartItem($conn);
    }else{
        //if not exist add new cart item
        $cartItem->addCartItem($conn);
    }

    //return to the index page after put the item into the cart
    ob_start();
    header("Location: index.php");
    ob_end_flush();
    exit();
}
    
```

Figure 33: Add Food Item Code Segment

```

if($_SERVER["REQUEST_METHOD"] == "POST"){
    if(isset($_SESSION["id"])){
        ob_start();
        header("Location: CustomerLogin.php");
        ob_end_flush();
        exit();
    }

    if(isset($_GET["cartItemId"]) && filter_input(INPUT_GET, 'cartItemId', FILTER_SANITIZE_NUMBER_INT)){
        $cartItemId = $_GET["cartItemId"];
    }else{
        ob_start();
        header("Location: Cart.php");
        ob_end_flush();
        exit();
    }

    $cartItem = new Cart();
    $cartItem->setCartItemId($cartItemId);
    $cartItem->setFoodItemSize_Id($_POST["foodItemSizePriceId"]);
    $cartItem->setQuantity($_POST["quantity"]);
    $cartItem->setCustomer_Id($_SESSION["id"]);

    if($_POST["quantity"] == 0){
        $cartItem->deleteCartItem($conn);
    }else if($_POST["previousCartItemSizeId"] != $_POST["foodItemSizePriceId"]){
        //if there is change of the food size then delete the previous
        $cartItem->deleteCartItem($conn);
        if($cartItem->checkCartItemExist($conn)){
            //if the new food size we choose already exist
            $cartItem->editCartItem($conn);
        }else{
            //if the new food item size not exist before
            $cartItem->addCartItem($conn);
        }
    }else{
        $cartItem->editCartItem($conn);
    }

    ob_start();
    header("Location: Cart.php");
    ob_end_flush();
    exit();
}
    
```

Figure 34: Remove and Edit Cart Item Code Segment

For the second situation where admin and staff can update customer's order status. Figure 35 shows the order place by the customer and each order associated order status. The admin and staff can click the manage button to redirect to the order details page. Figure 36 shows the order details page, admin and staff will update the customer order status at this webpage. Figure 37 shows the code segment to handle the action to update the order status.

| Order Id | Order Date | Delivery Address | Total Amount | Status | Action |
|----------|-----------------------|--|--------------|-----------|--------|
| 00000006 | May 26, 2023 14:48 PM | Jalan Jati 9, Taman Bahagia, 84000 Muar, Johor | RM 279.00 | Pending | Manage |
| 00000005 | May 8, 2023 14:40 PM | Jalan Jati 9, Taman Bahagia, 84000 Muar, Johor | RM 29.00 | Confirmed | Manage |

Figure 35: Manage Customer Order Interface

Order Details #00000005

| Customer Details | | Payment Details | |
|------------------|-------|-----------------|-----------------|
| First Name | Guest | Payment Method | CREDIT OR DEBIT |
| Last Name | Stacy | Transaction Id | |
| Email | | Total Amount | RM 29.00 |
| Contact Number | | | |

| Order Details | | | | |
|---------------|-----------|---------------------------------|---------------------|-----------------|
| Order Id | #00000005 | Order Date | 05/09/2023 14:40 PM | |
| No | Image | Food Name | Size | Quantity |
| 1 | | Hokkien Buns with Alkaline Kuli | Small | 1 |
| | | Unit Price | RM 29.00 | Total |
| | | | | RM 29.00 |
| | | Delivery Fee | | RM 0.00 |
| | | Total | | RM 29.00 |

| Delivery Details | |
|-------------------------|--|
| Block / Floor / Unit No | No 22 |
| Delivery Address | Jalan Jati 9, Taman Bahagia, 84000 Muar, Johor |
| Order Status | Confirmed |

Actions: Update Return

Figure 36: Order Details Interface

```

if(isset($_GET["orderId"]) && !empty($_GET["orderId"]) && filter_input(INPUT_GET, 'orderId', FILTER_SANITIZE_NUMBER_INT)){
    $orderId = $_GET["orderId"];
}

if($_SERVER["REQUEST_METHOD"] == "POST"){
    ob_start();
    header("Location: AllOrder.php");
    ob_end_flush();
    exit();
}

$orderStatus = $_POST["orderStatus"];
$order = new Order();

$order->setOrderId($orderId);
$order->setOrderStatus($orderStatus);

$status = $order->updateOrderStatus($conn);
    
```

Figure 37: Update Order Status Code Segment

4.4 Make Payment Module

After the customer adding desired food item into the cart, the customer needs to make the payment to complete the order placement. Figures 38 and 39 show the checkout page. The customer must enter their delivery address, block/floor/unit number and select the payment method to complete the order placement. Figure 40 shows the code segment for the checkout process. It will determine the payment method choose by the customer.

Checkout
Tang Huat Restaurant (峇株新街东兴餐室)

Personal Details

First Name
JENG FOONG

Last Name
YANG

Mobile Number
010-8135577

Delivery Details

Your Location
eg. Taman Bahagia, Jalan Junid, 84000 Muar, Johor

Block / Floor / Unit No
eg. Block 9B / 2nd Floor / No 22

Default Address

Order Details

1x Hokkien Braised Pork with Alkaline Kuih - Small RM 25.00

Subtotal RM 25.00
Delivery Fee RM 4.00
Total (max RM999.99) RM 29.00

Figure 38: Checkout Interface (a)

Order Details

1x Hokkien Braised Pork with Alkaline Kuih - Small RM 25.00

Subtotal RM 25.00
Delivery Fee RM 4.00
Total (max RM999.99) RM 29.00

Payment

Cash on Delivery

Pay by cash on delivery
Simply pay the driver, when he delivers the food to your doorstep.

Credit or debit card

Place Order

Figure 39: Checkout Interface (b)

```

<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $_SESSION["deliveryAddress"] = $_POST["deliveryAddress"];
    $_SESSION["unitAddress"] = $_POST["unitAddress"];

    if($_POST["paymentMethod"] == 1){
        ob_start();
        header("Location: PlaceOrder.php?payment_method=1");
        ob_end_flush();
        exit();
    }else{
        <script>
            const key = "<?php echo STRIPE_PUBLICKEY; ?>";
        </script>
        <script src='https://js.stripe.com/v3/'></script><script src='js/StripeScript.js'></script>
    </script>
    }
}
?>
    
```

Figure 40: Checkout Code Segment

4.5 Manage Food Menu Module

The Manage Food Menu module can be divided into two sub-modules that are manage food category and manage food item. Managing food categories involves adding, deleting, or editing food categories. Figure 41 displays the webpage for managing food categories, where the admin can add new categories, edit existing ones, or delete categories. Figures 42 and 43 show the code segments for adding and editing food categories, respectively. Figure 44 shows the code segment for deleting a food category.

Food Category

Add New Category

Category Name:

| # | Category Name | Action |
|---|---------------|---|
| 1 | Meat | <input type="button" value="Edit"/> <input type="button" value="Delete"/> |
| 2 | Noodle | <input type="button" value="Edit"/> <input type="button" value="Delete"/> |
| 3 | Rice | <input type="button" value="Edit"/> <input type="button" value="Delete"/> |

Figure 41: Manage Food Category Interface



Figure 55: Manage Staff Interface

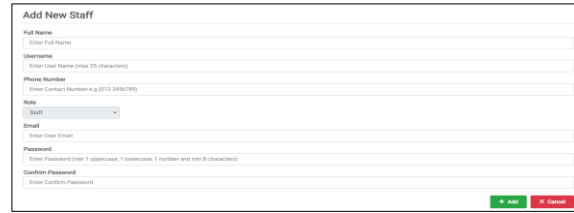


Figure 56: Add New Staff Interface

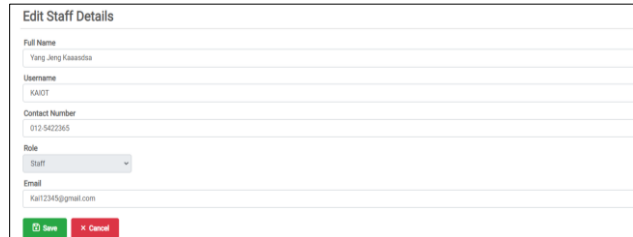


Figure 57: Edit Staff Details Interface

Figure 58 shows the code segment for adding new staff. Figure 59 shows the code segment for editing the staff details. Figure 60 shows the code segment to delete the existing staff account.

```

if($_SERVER["REQUEST_METHOD"] == "POST"){
    $user = new ManagementTeamMember();
    $user->setUsername(strtoupper(htmlspecialchars($_POST["username"])));
    $user->setEmail(htmlspecialchars($_POST["email"]));
    $user->setPassword($_POST["confirmPassword"]);
    $user->setRole($_POST["role"]);
    $user->setFullName(htmlspecialchars($_POST["fullName"]));
    $user->setContactNumber($_POST["contactNumber"]);

    if($user->checkSameUsername($conn)){
        $error[] = "499888; This username already exist...";
    }

    if($user->checkSameEmail($conn){
        $error[] = "499888; This email already exist...";
    }

    if (empty($error)) {
        if($user->addManagementTeamMember($conn){
            $status = true;
        }else{
            $status = false;
        }
    } else {
        //error array is not empty indicating got error
    }
}
    
```

Figure 58: Add New Staff Code Segment

```

if($_SERVER["REQUEST_METHOD"] == "POST"){
    $user = new ManagementTeamMember();
    $id = $_GET["id"];
    $user->setId($id);
    $user->setUsername(htmlspecialchars($_POST["username"]));
    $user->setEmail($_POST["email"]);
    $user->setFullName(htmlspecialchars($_POST["fullName"]));
    $user->setContactNumber($_POST["contactNumber"]);

    if($user->checkSameUsernameExceptOwn($conn){
        $error[] = "499888; This username already exist in other account...";
    }

    if($user->checkSameEmailExceptOwn($conn){
        $error[] = "499888; This email already exist in other account...";
    }

    if (empty($error)) {
        if($user->editStaffDetails($conn){
            $status = true;
        }else{
            $status = false;
        }
    } else {
        //error array is not empty indicating got error
    }
}
    
```

Figure 59: Edit Staff Details Code Segment

```

<?php
require '../includes/init.php';
require '../includes/db.php';
require '../class/FoodItem.php';

if($_SERVER["REQUEST_METHOD"] == "POST"){
    $idToDelete = $_POST["id"];
    $foodItem = new FoodItem();
    $foodItem->setFoodId($idToDelete);
    if($foodItem->deleteFoodItem($conn){
        echo true;
    }else{
        echo false;
    }
}
    
```

Figure 60: Delete Staff Code Segment

4.8 Manage Account Details Module

The manage account details module can be divided into two different scenarios. The first scenario involves customers being able to change their account profiles and passwords. The second scenario is where the admin and staff can change their account passwords. Figure 61 displays the interface for customers to change their passwords. The system allows the customer to reset the account password by entering the correct current password, as well as a valid new password and confirmation of the new password. Figure 62 shows the interface for customers to change their account profiles. Figures 63 and 64 display the corresponding source code for these interfaces.

Figure 61: Customer Change Password Interface

Figure 62: Customer Change Profile Interface

```
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $newPassword = $_POST["newPassword"];
    $confirmPassword = $_POST["confirmPassword"];
    $currentPassword = $_POST["currentPassword"];

    $customerId = $_SESSION["id"];

    $customer = new Customer();
    $customer->setCustomerId($customerId);
    if($customer->checkCurrentPassword($conn, $currentPassword)){
        if(!($newPassword == $confirmPassword)){
            $error = "*The password does not match";
        }else{
            if($customer->resetPassword($conn, $newPassword){
                $changeStatus = true;
            }else{
                $error = "*Error! Password cannot be changed...
                Please contact the owner";
            }
        }
    }else{
        $error = "*Incorrect Current Password";
    }
}
```

Figure 63: Customer Change Password Code Segment

```
if($_SERVER["REQUEST_METHOD"] == "POST"){
    if(isset($_POST["profileButton"])){
        $firstName = $_POST["firstName"];
        $lastName = $_POST["lastName"];
        $contactNumber = $_POST["contactNumber"];

        $customer->setFirstName($firstName);
        $customer->setLastName($lastName);
        $customer->setContactNumber($contactNumber);

        if(!$customer->updateProfile($conn){
            $error=1;
        }else{
            $status = 1;
        }
    }else if(isset($_POST["addressButton"])){
        $unitNumber = $_POST["unitNumber"];
        $address = $_POST["deliveryAddress"];

        $customer->setUnitNumber($unitNumber);
        $customer->setAddress($address);

        if(!$customer->updateAddress($conn){
            $error = 2;
        }else{
            $status = 2;
        }
    }
}
```

Figure 64: Customer Change Profile Code Segment

Figure 65 showcases the password change interface, which is specifically designed for administrators and staff members. When an admin or staff member intends to reset their password, they are required to input their valid current password and choose a new password that adheres to the correct format. In contrast, Figure 66 presents the corresponding source code associated with the change password module.

Figure 65: Admin and Staff Change Password Interface

```
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $currentPassword = $_POST["currentPassword"];
    $newPassword = $_POST["newPassword"];
    $confirmPassword = $_POST["confirmPassword"];
    $pattern = "/^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$%^&*])(?=[^&quot;'\s]).{12,}$/";

    $SESSION["mt_id"] = 1;
    $managementMemberId = $_SESSION["mt_id"];

    if(filter_var($managementMemberId, FILTER_VALIDATE_INT) && preg_match($pattern, $newPassword)){
        $managementTeamMember = new ManagementTeamMember();
        $managementTeamMember->setId($managementMemberId);

        //Admin12345678
        if($managementTeamMember->checkCurrentPassword($conn, $currentPassword){
            $error = "Incorrect Current Password";
        }else if($newPassword == $confirmPassword){
            $error = "New Password and Confirm Password Do Not Match";
        }else{
            if($managementTeamMember->resetPassword($conn, $newPassword){
                $status = true;
            }else{
                $error = "Error, Password cannot be change. Please Contact the developer";
            }
        }
    }else{
        $error = "Invalid New Password Format";
    }
}
```

Figure 66: Admin and Staff Change Password Code Segment

4.2 System Testing

System testing is conducted to ensure that the developed system meets both the functional and non-functional requirements. The completion of system testing is determined by the proper functionality of all modules, aligned with the specified functional and non-functional requirements. Table 4 shows the

result of system testing. There are a total of 8 modules with 54 test cases that has been carried out to test the system. The overall test case results are show in the Table 5 below.

Table 5: Overall Test Result of System Testing

| Test Cases | Total Test Cases | Total Success | Total Failed |
|---------------|------------------|---------------|--------------|
| TC_100 | 6 | 6 | - |
| TC_200 | 5 | 5 | - |
| TC_300 | 9 | 9 | - |
| TC_400 | 5 | 5 | - |
| TC_500 | 10 | 10 | - |
| TC_600 | 4 | 4 | - |
| TC_700 | 9 | 9 | - |
| TC_800 | 6 | 6 | - |
| Total: | 54 | 54 | 0 |

4.4 User Acceptance Testing

The main objective of user acceptance testing is to collect feedback and insights from the individuals who will be utilizing the system to perform designated tasks. In this project, the restaurant owner and customer were chosen as the testers for user acceptance testing. To facilitate the testing process, a straightforward form was created for them to complete after testing the system. Figure 67 and 68 shows the outcome of the results of the user acceptance testing done by the owner and customer.

User Acceptance Testing (UAT)
Tang Huat Restaurant Online Food Ordering System

Name: NG WEI LIANG
Phone Number: 012-2520703
Position: Owner

| Acceptance Criteria | Test Results | | Note |
|--|--------------|------|------|
| | Pass | Fail | |
| Login | / | | |
| a) Display Login page | / | | |
| b) Able to login into the account by entering valid email and password | / | | |
| Manage Order | / | | |
| a) Display Manage Order page | / | | |
| b) Display the Order Details page | / | | |
| c) Provide edit/update order status | / | | |
| Manage Food Menu | / | | |
| a) Display Food Item page | / | | |
| b) Able to add new food item | / | | |
| c) Able to edit food item details | / | | |
| d) Able to delete food item | / | | |
| e) Display Food Category page | / | | |
| f) Able to add new food category | / | | |
| g) Able to edit food category | / | | |
| h) Able to delete food category | / | | |
| Generate Report | / | | |
| a) Display Sale Report page | / | | |
| b) Able to see the five top most sale food items | / | | |
| c) Display report based on the date selected | / | | |
| Manage User Account | / | | |
| a) Display Manage Customer page | / | | |
| b) Able to edit/update customer account status | / | | |
| c) Display Manage Staff page | / | | |
| d) Able to add new staff account | / | | |
| e) Able to edit staff account details | / | | |
| f) Able to delete staff account | / | | |
| Manage Account Details | / | | |
| a) Display Change Password page | / | | |
| b) Able to change account password | / | | |

I, NG WEI LIANG hereby declare that the information provided is true and correct.

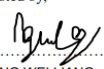
Tested by,

(NG WEI LIANG)
Date: 4/6/2023

Figure 67: User Acceptance Testing (Owner)

User Acceptance Testing (UAT)
Tang Huat Restaurant Online Food Ordering System

Name: Nurain Hadirah binti Ramli
Phone Number: 0137046151
Position: Customer

| Acceptance Criteria | Test Results | | Note |
|--|--------------|------|------|
| | Pass | Fail | |
| Register Account | | | |
| a) Display Registration page | / | | |
| b) Able to register a new account | / | | |
| Login | | | |
| a) Display Login page | / | | |
| b) Able to login into the account by entering valid email and password | / | | |
| Manage Order | | | |
| a) Display Food Menu page | / | | |
| b) Display Food Item page | / | | |
| c) Display Cart page | / | | |
| d) Able to add food item into cart | / | | |
| e) Able to edit cart item size and quantity | / | | |
| f) Able to remove cart item | / | | |
| Make Payment | | | |
| a) Display Checkout page | / | | |
| b) Able to input delivery details | / | | |
| c) Able to select payment method | / | | |
| d) Able to make online payment in stripe checkout page | / | | |
| e) Display Order Details page | / | | |
| Manage Account Details | | | |
| a) Display Profile page | / | | |
| b) Able change profile details | / | | |
| c) Display Change Password page | / | | |
| d) Able to change password | / | | |

I, Nurain Hadirah binti Ramli hereby declare that the information provided is true and correct.

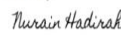
Tested by,

.....
(NURAIN HADIRAH BINTI RAMLI)
Date: 8/6/2023

Figure 68: User Acceptance Testing (Customer)

4. Conclusion

In conclusion, Tang Huat Restaurant Online Food Ordering System was developed to solve problems faced by admin, such as hard-to-track back orders, the need to calculate monthly sales manually and not having a proper online menu. Three types of users will be using this system which are admin, staff and customers. Admin can use this system to manage food menu, customer order, staff, customer account, and account details. Meanwhile, the system only allows staff to manage customer orders. Customers can use this system to place orders and make payments online. The system was launched on the web server and live on the Internet so that the customer can order food using their web browser. This system is expected to help customers to save time when ordering food and owner or staff to manage customers' orders efficiently.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia for its support.

References

- [1] S. Aday and M. S. Aday, "Impact of covid-19 on the Food Supply Chain," *Food Quality and Safety*, vol. 4, no. 4, pp. 167–180, 2020. [Accessed: 20-Dec-2022].
- [2] Chai, L. T., & Yat, D. N. C. "Online food delivery services: Making food delivery the new normal". *Journal of Marketing advances and Practices*. 2019. vol 1, no. 1. pp 62-77. [Accessed: 20-Dec-2022].
- [3] Deepa, T., & Selvamani, P. "Online Food Ordering System". *International Journal of Emerging Technologies and Innovative Research*. 2018. vol 5, no. 12, pp. 143-148. [Accessed: 20-Dec-2022].
- [4] Ramesh, R., Prabhu, S. V., Sasikumar, B., Devi, B. K., Prasath, P., & Kamala, S. P. R. "An empirical study of online food delivery services from applications perspective". *Materials Today: Proceedings*. 2021. Vol. 69, part 4. [Accessed: 10-Dec-2022].
- [5] A. Adel and B. Abdullah, "Waterfall model, V-model, spiral model, and other SE models," *Systems Engineering Models*, pp. 129–138, 2019. [Accessed: 27-Nov-2022].
- [6] Gharajeh, M. S. "Waterative model: An integration of the waterfall and iterative software development paradigms". *Database Syst. J.* 2019. vol 1, no. 10, pp 75-81. [Accessed: 20-Dec-2022].
- [7] Seidl, M., Scholz, M., Huemer, C., & Kappel, G. "The Use Case Diagram". In *UML@ Classroom*. 2015. vol 1. Springer, Cham. pp. 23 – 47. [Accessed: 18-Dec-2022].