

National College of Ireland

BSc (Honours) in Computing

Software Development

2021/2022

Eunkyoung Kim

X18150365

X18150365@student.ncirl.ie

My POS

Technical Report

Contents

Executive Summary	3
1.0 Introduction	3
1.1. Background	3
1.2. Aims.....	3
1.3. Technology.....	4
1.4. Structure	4
2.0 System.....	4
2.1. Requirements.....	4
2.1.1. Functional Requirements.....	5
2.1.1.1. Requirement 1: User Registration	6
2.1.1.1.1 Description & Priority.....	7
2.1.1.1.2 Use Case	7
2.1.1.2. Requirement 2: Log in.....	8
2.1.1.2.1 Description & Priority.....	8
2.1.1.2.2 Use Case	8
2.1.1.3. Requirement 3: Add products.....	10
2.1.1.3.1 Description & Priority.....	10
2.1.1.3.2 Use Case	10
2.1.1.4. Requirement 4: Generate QR code.....	12
2.1.1.4.1 Description & Priority.....	12
2.1.1.4.2 Use Case	12
2.1.1.5. Requirement 5: View order.....	14
2.1.1.5.1 Description & Priority.....	14
2.1.1.5.2 Use Case	14
2.1.1.6. Requirement 6: Manage staff	16
2.1.1.6.1 Description & Priority.....	16
2.1.1.6.2 Use Case	16
2.1.1.7. Requirement 7: View Sales report	18
2.1.1.7.1 Description & Priority.....	18
2.1.1.7.2 Use Case	18
2.1.1.8. Requirement 8: Make table side order	19
2.1.1.8.1 Description & Priority.....	19

2.1.1.8.2	Use Case	19
2.1.1.9.	Requirement 9: Order Payment.....	21
2.1.1.9.1	Description & Priority.....	21
2.1.1.9.2	Use Case	21
2.1.1.10.	Requirement 10: Logout	23
2.1.1.10.1	Description & Priority.....	23
2.1.1.10.2	Use Case	23
2.1.2.	User Requirements	25
2.1.3.	Environmental Requirements	25
2.1.4.	Usability Requirements.....	25
2.2	Design & Architecture	25
2.3	Implementation	26
2.3.1	Login/Logout	26
2.3.2	Add product/ upload image	27
2.3.3	Display QR code	27
2.3.4	Sales Report	28
2.3.5	Change order status.....	30
2.3.6	Change calculation	32
2.4	Graphical User Interface (GUI).....	33
2.5	Testing.....	39
2.6	Evaluation	39
3	Conclusions	39
4	Further Development or Research	39
5	References	40
6	Appendices.....	40
7	Objectives.....	42
8	Background	43
9	State of the Art.....	43
10	Technical Approach.....	43
11	Technical Details	44
12	Special Resources Required	44
13	Project Plan	44
14	Testing.....	45
14.1	Project Plan	46
14.2	Reflective Journals	47
14.3	Other materials used	49

Executive Summary

My POS is a web-based Point of Sale application for small size restaurants and cafeterias which is available for iPad and mobile devices. It allows a table-side ordering system with a QR code which drives a good user experience. By scanning a QR code, customers can order at a table conveniently. It can also reduce the contact with staff. When customers scan the QR code, they will get a notification that will lead them to the order section. The POS application provides register, login and logout using staff id and password. On Staff page, staff account can be managed. On the Product section, new items can be added with its name, image, description, price, category of the products. The POS section allows for managing order. It shows the whole table orders and the status of each table order item whether each order item is ready or not. MyPOS also provides a sales report section, which demonstrates daily, monthly sale report in a graph. It will allow the owner to identify the most popular menus and busy time which helps scheduling staffs effectively. QR code generation section will generate QR codes. It also has a payment section that customers pay by card or cash.

1.0 Introduction

1.1. Background

During the Covid 19, social distancing and contactless becomes our new lifestyle. It leads some people to feel comfortable and safe to use online for ordering foods. Moreover, these days, it is found that many restaurants are struggling to fill staff vacancies. These made me think of how I can merge the ideas. MyPOS allows for customer-cantered order with QR code table-side ordering service, which make customers feel comfortable in a restaurant/cafeteria. The POS application is easy to use for users/staffs. Moreover, purchasing an expensive POS device can be a burden for small size businesses. This is because there are continuous additional costs for licenses and maintenance. My POS will address these problems and provide restaurant/cafeteria owners and customers with safe and convenient ordering service.

1.2. Aims

The POS application will provide the following functionalities. Login, logout, register staff, manage staff account, manage table order by table order status, view orders, add new products, sales report, QR code generation. The owner of a restaurant can login and logout using staff number and password, register, and manage the staff account. It allows for adding new products with its name, image, description, price, and category of the product. The owner can view orders as a whole and can manage orders by ticking the checkbox of each item when it is ready and done. It also shows the reporting metrics in a graph which allows the owner to figure out the busy time and slower time during times and days so that it helps over schedule and under schedule. It will provide daily, monthly sale reports. The owner can identify the most popular menus. Downloadable QR codes is provided. It can be also printed to put on the table. When customer scan the QR code located on the table, it leads them to

the table order page. Once the customer completes placing order, the information of order is send to the POS. Extra order can be also added by staff manually.

1.3. Technology

The main technologies used are Visual Studio Code IDE, PHP, JavaScript,(Ajax), jQuery, MySQL, DuckDns, Chart.js, HTML5, Bootstrap, and CSS. Visual Studio Code is the IDE that I used to write code to develop the application. PHP is used to send data to database and query to retrieve data from the database. JavaScript is used for few button operations and calculation of changes for cash payment and load pages with data and display alert. Mainly jQuery is used for my interactive web application. For example, food shopping(ordering) cart, loading cart, saving cart, adding items to cart, displaying cart, file uploading from browser, order list check box in Order status section, displaying modal (QR code) on button click, and loading next action and changing CSS of certain item, etc. Ajax is used to communicate with PHP script which deals with database query and interaction. The phpMyAdmin is used to manage MySQL database which I am familiar with as I have been used it for my previous projects. GitHub is used for version control. The important approach and big challenge was generating of QR code and connecting it to various tables and integration of sending the ordered data from mobile to POS system. PHP Session is used to for logout, to allow only authorised (logged in) user to access and use the application, and is used to store the table number variable to be used across multiple PHP pages including URL embedded in QR code. Chart.js is used to visualize the sales report page. Google font icon svg is used for navbar icons. Traylor Brands tool is used to create the logo image(favicon) for my application.

1.4. Structure

This report includes an Executive Summary of my POS web application and its functionalities. The introduction section will address the background of my project idea, its aim, and the technology I will use to develop the POS application. In System section, it will describe the functional requirements of the application. In other words, what the system must accomplish and how it can be done by using Use Case and Use Case Diagram. It will also present the design & architecture of the application, testing plan, and evaluation.

2.0 System

2.1. Requirements

To use this application, a user should be a staff member of the store and already registered in the application. The UI is easy to use and user-friendly. All the functionalities are straightforward. I believe a few days of training is enough to get familiar with this application. Users will be trained on how the application is working and learn all the functionalities of the application.

2.1.1. Functional Requirements

Functional requirements describe what the system must accomplish and how the system accomplishes its functional requirements. In this section, I will describe the functional requirements of my POS web application.

1. Register

The user must register and have an account in the application. It is done by clicking the register button after typing a name, staff id, email, and uploading photo, and setting their password. While registering, the user can cancel the registration process by clicking the Cancel button. It will direct user to the login page. This registration process will be done by PHP MySQL user authentication.

2. Login

The user can log in once they have registered for the application. The user enters the staff id number and the password that they set when they registered. User will click the login button and then the user will be logged in to the My POS application. If the user is a new staff, a registration button is available for new registration.

3. Add products

The user can add products of the store by clicking Add products button, then it will show pop up that takes inputs of product name, image, description, price, and category of the product. They can upload images by clicking the browse button. When they set all the information of a product and click the add button, the product will be displayed with the image and its information on the product list page. While adding a new product, the process can be cancelled by clicking the cancel button.

4. QR code generation

The user clicks the 'Generate QR code' button, then it will display QR codes. The user can download the QR code in zip file and print these QR codes by clicking 'Download All QR code' and 'Print QR code' button.

5. Make table order page that notification leads to, and make confirm order page

When a user scans a QR code on the table with their mobile phone camera, there will be pop up notification. When the user clicks the notification, it will lead customers to the ordering site. On that page, user can select the food category tab to choose food item. User can add food items to cart, cancel items from cart, and clear the cart. Once 'Order now' button is clicked, the order is sent to MyPOS system.

6. View order

This page will show the entire table orders with each table number and corresponding table order lists. When a customer makes an order or a staff takes extra orders in POS, those orders lists will be added and displayed on this page. This page also shows the order status so that user(staffs) can set status of the completed order by clicking the check box and deleting the order status by clicking the x button. Once a food item is ready and clicks the check box of the item, then the corresponding food item name on the corresponding table will turn to bold green to show that item is ready and serve. Staff can add extra order and deal with payment on orderPayment page. This page can be reached by clicking the corresponding table box in View order page.

7. OrderPayment

This page is used for staff to take orders via POS and deal with the payment. The user enters the amount of money that was given from customer via the number key buttons and clicks the cash button. The change for the customer will be presented. Card payment is also acceptable.

8. Manage staff account

The user can see all staff lists with their name, staff id number and email address. It allows management staff by adding new staff, editing staff details, and deleting staff by clicking the delete button.

9. Sales report in graph

The user can view current date's sales amount, sales per weekdays which give user a insight which weekday has more sales. The user can view the busy time of a day, monthly sales, and the most popular menus.

10. Log out

Log out allows the user to log out of the POS application.

2.1.1.1. Requirement 1: User Registration

The requirement is User Registration. The user will click the registration button and enter name, id, email, and password.

2.1.1.1.1 Description & Priority

The requirement is User Registration, this allows the users to register for the application. This is the essential step to use the POS application because the only registered user can log in to the application.

2.1.1.1.2 Use Case

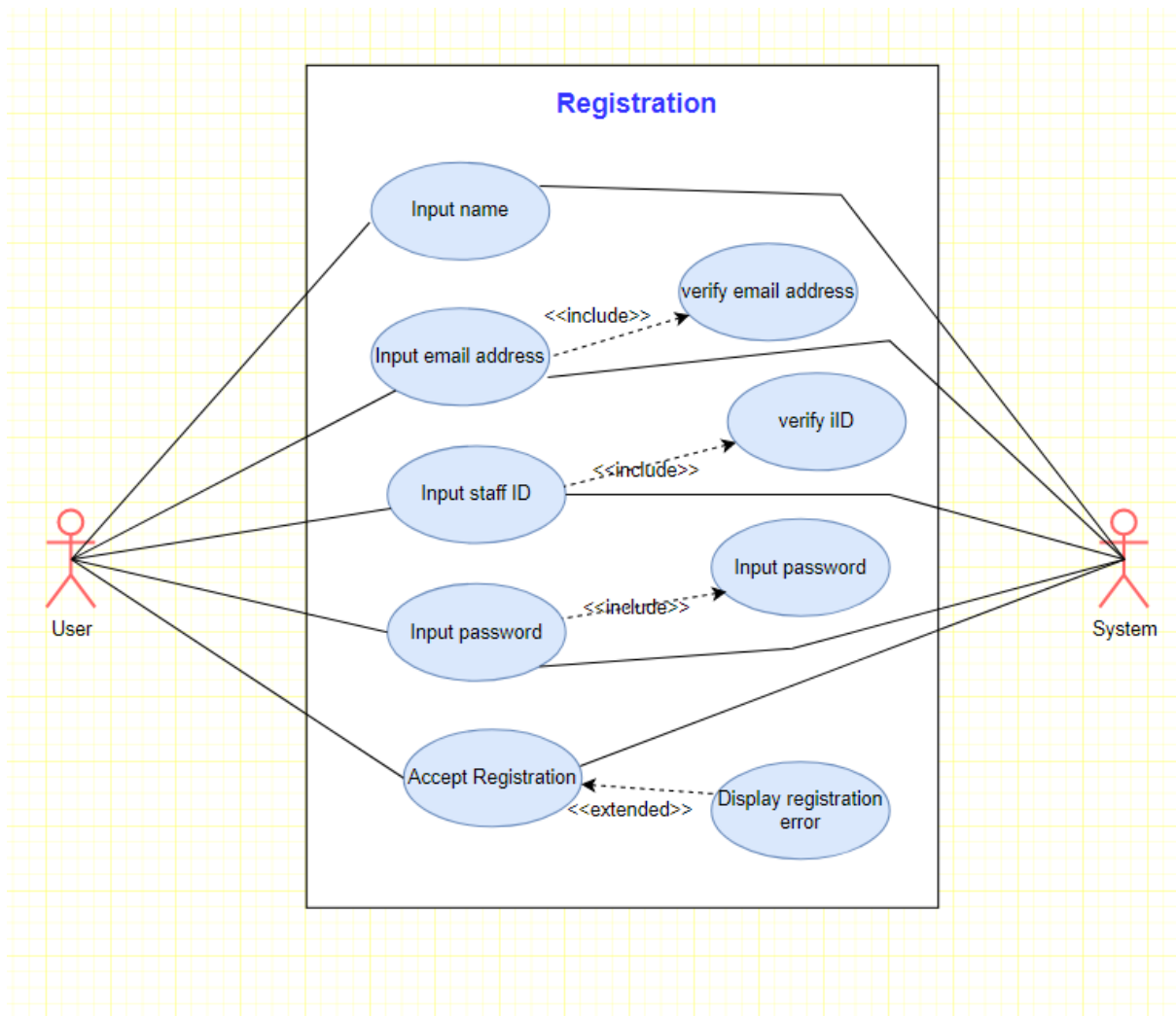
Scope

The scope of this use case is to allow the user to register for the system with a name, staff id number, email, and password.

Description

This use case describes the registration page of the application.

Use Case Diagram



Flow Description

Precondition

The system has started and is waiting for input from the user. The user must enter a valid email address which contains @ symbol and a password.

Activation

This use case starts when the user clicks the registration button in the first default page which is login page of the application.

Main flow

1. The system identifies the register button being clicked and leads the user to the registration page where the user must enter a name, staff id, email, password and upload photo.
2. Once the user details are entered the user press the registration button. (See E2)
3. The system verifies that input details are valid, if they are valid, then the user will be directed to login page for login. (See E1)

Exceptional flow

E1 : Invalid user details

1. The system verifies if input details are valid, if they are not valid, then the user will be informed of the problem and will be asked to try to register again. When any blank is empty, it will alert to user to fill all field.

E2 : click the cancel button

1. The user clicks the cancel button.
2. The system will not save the details and direct the user to the login page.

Termination

The system presents the user with the main page(login page) of the application.

Post condition

The system goes into a wait state.

2.1.1.2. Requirement 2: Log in

The requirement is to log in. The user will enter their staff id number and password they set on the process of registration.

2.1.1.2.1 Description & Priority

The requirement is Login, this allows the users to log in to the application. This is the essential requirement to use the POS application because the only logged-in user can access the application.

2.1.1.2.2 Use Case

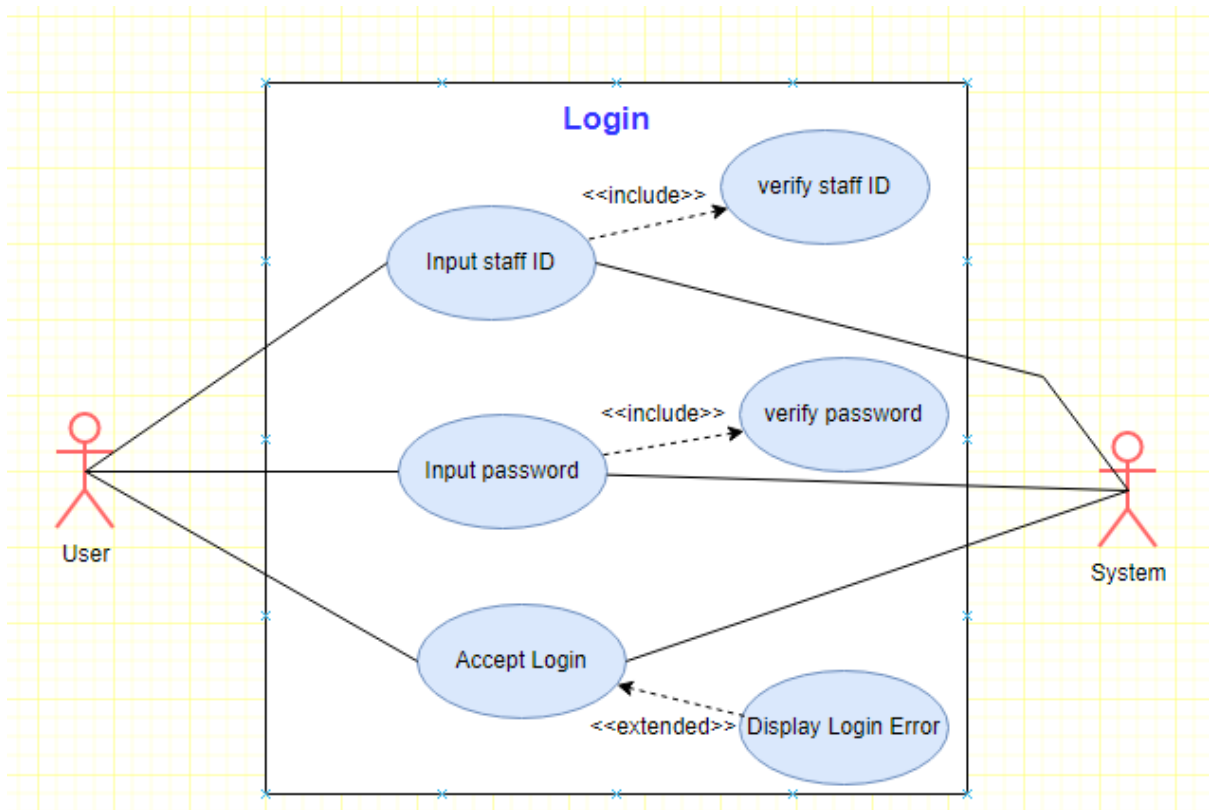
Scope

The scope of this use case is to allow the user to log in to the system with a staff id number and password.

Description

This use case describes the login page of the application.

Use Case Diagram



Flow Description

Precondition

The user must have registered for the application by using a name, valid email, staff id number, password and photo uploads.

Activation

This use case starts when the user clicks the login button and has not logged in or has logged out.

Main flow

1. The system identifies the user is logged in and loads the login page where the user must enter the staff id and password.
2. Once the user details are entered the user press the login button.
3. The system verifies that input login details are valid, if they are valid, then the user will be logged in.

Exceptional flow

E1 : Invalid user details

1. The system identifies the user is logged in and loads the login page where the user must enter the staff id and password.
2. Once the user details are entered the user press the login button.
3. The system verifies if input details are valid, if they are not valid, then the user will be informed of the problem and will try logging in again.

Termination

The system presents the user with the main page of the application.

Post condition

The system goes into a wait state.

2.1.1.3. Requirement 3: Add products

The requirement is Add products. The user will enter the details of the new product to add new items.

2.1.1.3.1 Description & Priority

The requirement is adding products, this allows the users to add new products to the application. This is the essential requirement because, without products, the POS application cannot deal with taking orders.

2.1.1.3.2 Use Case

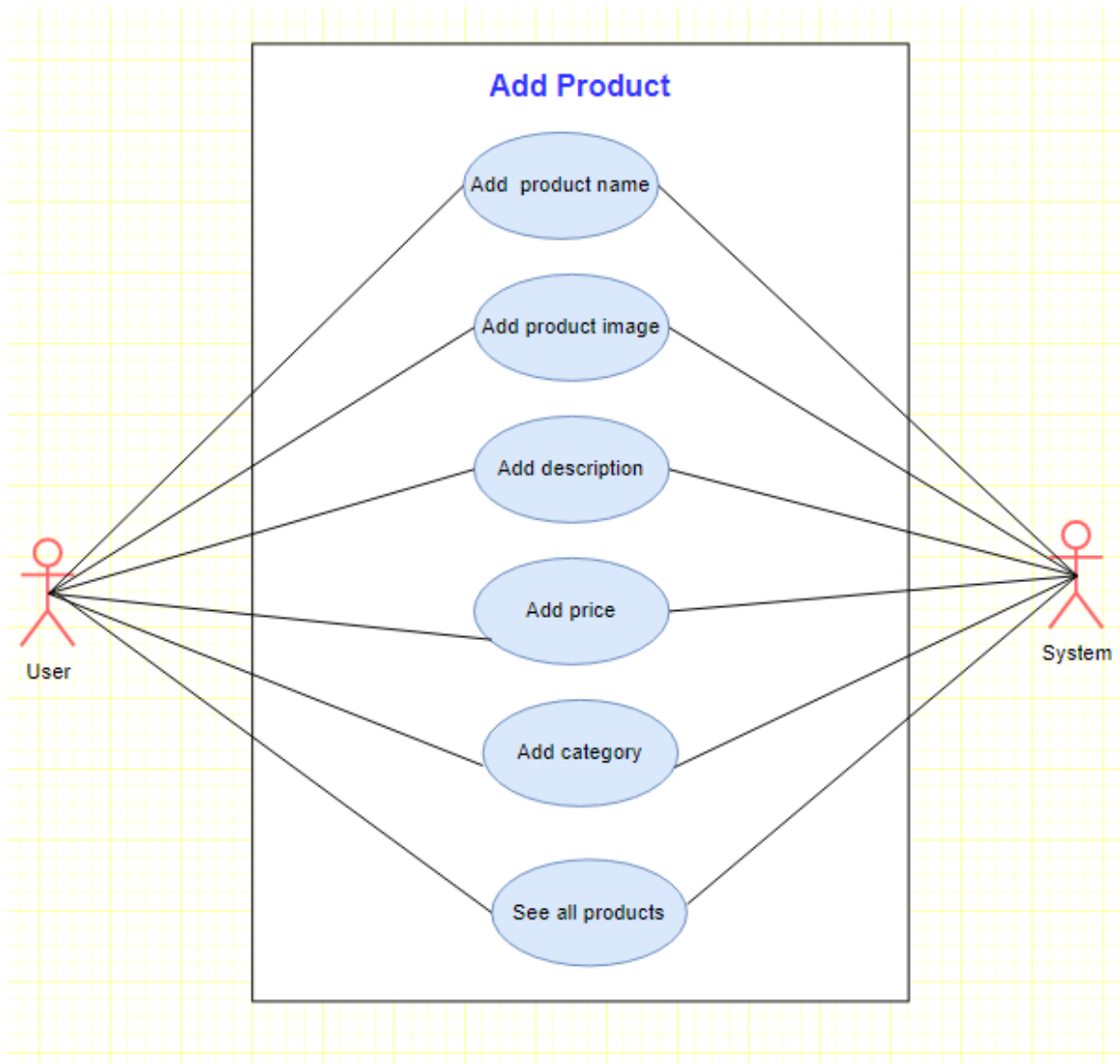
Scope

The scope of this use case is to allow the user to add a new product to the system by entering the product name, image, description, price, and category.

Description

This use case describes the Add product page of the application.

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application and have the image of a product to upload.

Activation

This use case starts when the user clicks the Product section in the navigation bar and clicks the Add products button.

Main flow

1. The system identifies the user clicking the Products section in the navigation bar.
2. The system displays the list of products that were already added.
3. Once the user clicked the Add Product button, the system pops up the form(modal) for adding products.

4. The user enters the product name, product price, description, category, and uploads the product image by clicking the brows button.
5. The user clicks the Add button. (See A1), (See E1)
6. The system saves details in a database.

Alternate flow

A1 : Reset the form

1. The user clicks the reset button to remove the typed detail and fill it out again.
2. The system removes all typed details from the input fields.
3. The use case continues at position 4 of the main flow.

Exceptional flow

E1 : click the close button

1. The user clicks the cancel button.
2. The system will not save the details and direct the user to the product list view page.

Termination

The system presents the user with the product list page of the application.

Post condition

The system goes into a wait state.

[2.1.1.4. Requirement 4: Generate QR code](#)

The requirement is to generate a QR code. The user will generate QR code, download and print them.

[2.1.1.4.1 Description & Priority](#)

The requirement is to Generate a QR code, this allows the users to generate, download and print the QR code. This is the essential requirement because this POS application accepts the table side order via QR code scanning.

[2.1.1.4.2 Use Case](#)

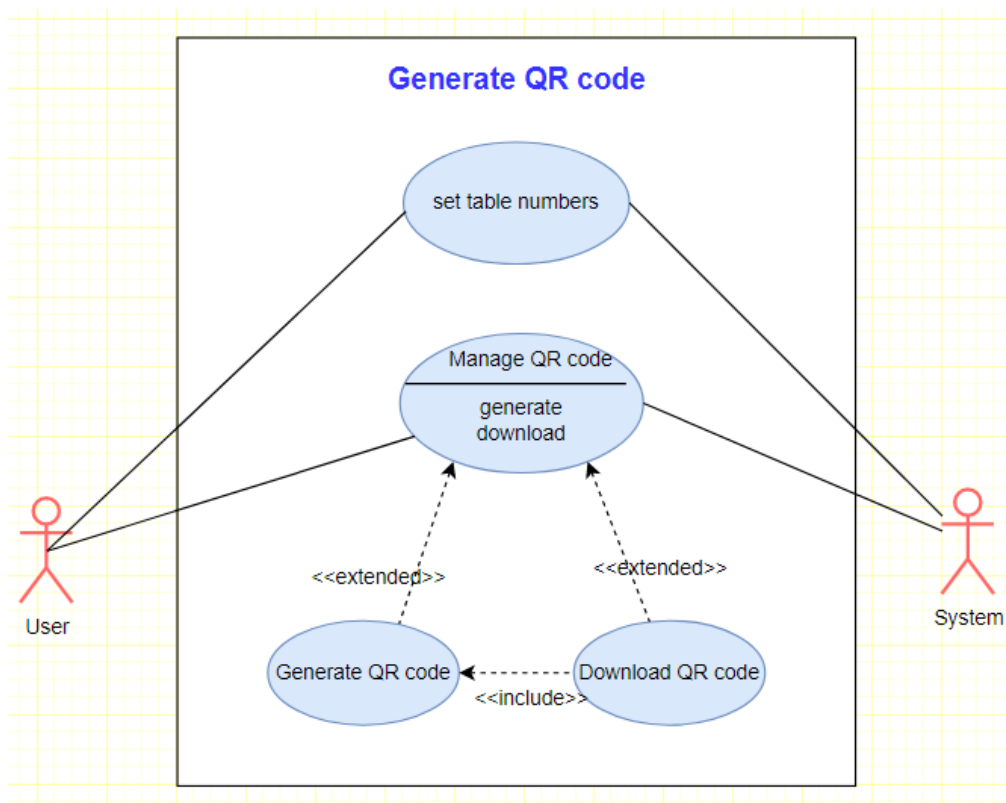
Scope

The scope of this use case is to allow the user to get QR code. The user can generate the code by clicking the Generate QR code button, and the QR code can be downloaded by clicking the Download all QR code button and printable by Print QR code button.

Description

This use case describes the Generate QR code, Download and Print QR code.

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application.

Activation

This use case starts when the user clicks the QR code section in the navigation bar.

Main flow

1. The system identifies the user clicking the QR code section in the navigation bar.
2. The system displays the QR code page.
3. The user clicks the Generate QR code button. (See A1). (See A2)
The system generates and displays the QR.

Alternate flow

A1 : Download QR code

1. The user clicks the Download QR code button.
2. The system downloads the zip file of the generated QR code images.

A2 : Print QR code

1. The user clicks the Print QR code button.
2. The system print thePDF file of the generated QR code.

Termination

The system presents the user with a QR code page of the application.

Post condition

The system goes into a wait state.

2.1.1.5. Requirement 5: View order

The requirement is a View order. The user can see the entire order with its table.

2.1.1.5.1 Description & Priority

The requirement is to view table orders, this allows the users to view table orders as whole and manage each table order. This is the essential requirement because the order placed by scanning QR code will be presented on this page.

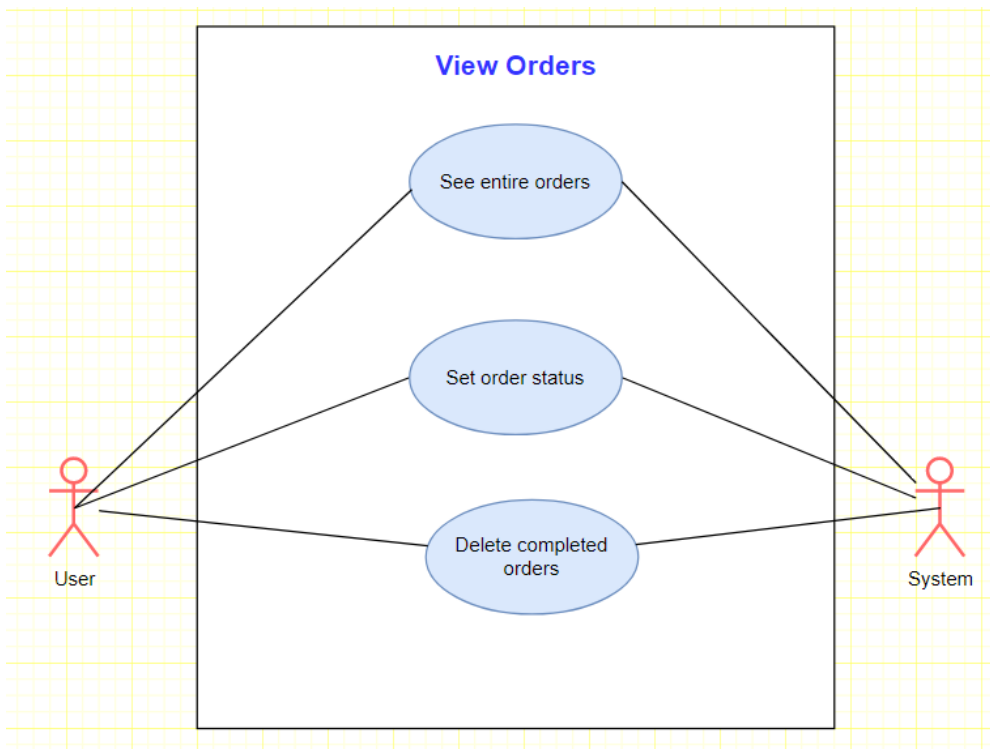
2.1.1.5.2 Use Case**Scope**

The scope of this use case is to allow the user to view all table orders and manage the orders by checking and updating each table order item status.

Description

This use case describes the View table orders

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application and there is at least one order.

Activation

This use case starts when the user clicks the POS view orders button in the navigation bar.

Main flow

1. The system identifies the user who clicks the POS view orders button in the navigation bar.
2. The system displays the view orders page.
3. The user can see the entire orders per table.
4. The user can set the status of the order.
5. The user ticks the check box of the list of orders to set the order status as 'served'.
6. The system crosses the list out by drawing a line through it.
7. The system changes the colour of the food item name to green in table order list box.
8. The user clicks the x to delete the completed orders.
9. The system deletes the display of the order.

Termination

The system presents the user with the current view orders page of the application.

Post condition

The system goes into a wait state.

2.1.1.6. Requirement 6: Manage staff

The requirement is to Manage staff. The user will view all the staff details, add new staff, edit, and delete staff details.

2.1.1.6.1 Description & Priority

The requirement is to Manage staff, this allows the users to view staff information, the list of staff with their photo, name, email address, and staff id. The user can add new staff through the registration process. The user also can update and delete staff details.

2.1.1.6.2 Use Case

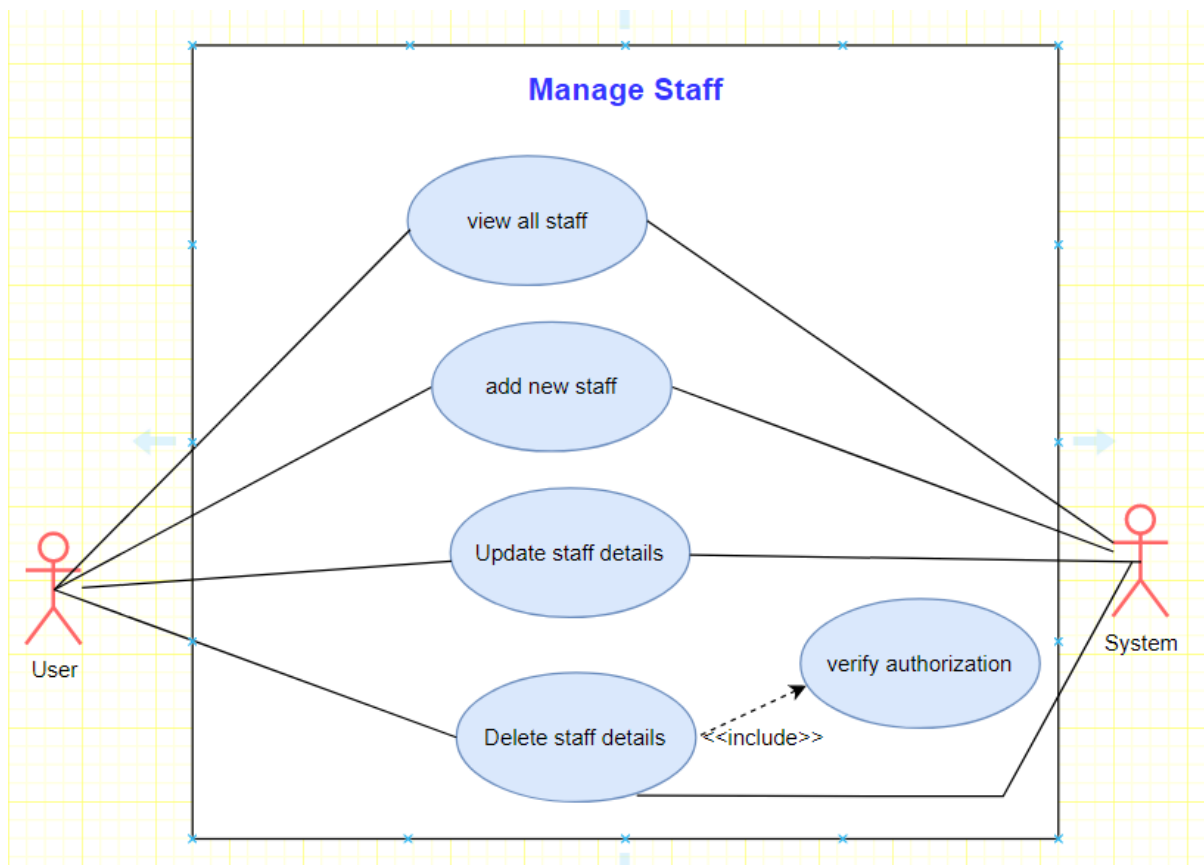
Scope

The scope of this use case is to allow the user to manage staff such as adding new staff, viewing the list of staff, update and deleting the staff detail.

Description

This use case describes the Manage staff.

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application. There is a registered user to update and delete.

Activation

This use case starts when the user clicks the staff section in the navigation bar.

Main flow

1. The system identifies the user clicking the Staff Page section in the navigation bar.
2. The system displays the staff list page.
3. The user views the details of staffs. (See A1, A2)

Alternate flow

A1 : Update staff

1. The system identifies the user who clicks the update button.
2. The user changes the name, email address, and password.
3. The system updates the modified staff details.

A2 : Delete staff

1. The system identifies the user clicking the delete button.
2. The system checks if the user has a right to deletion. (See E1)
3. If the user has a right, the system will delete the staff data.

Exceptional flow

E1 : Unauthorised user

1. The user is an unauthorized user for deletion.
2. The system will alert the user that the user has no right to delete.
3. The system will display the staff list page.

Termination

The system presents the user with the staff list page of the application.

Post condition

The system goes into a wait state.

2.1.1.7. Requirement 7: View Sales report

The requirement is a Sales report. The user will view weekday sales, monthly sales, busy time of a day view and the most popular menus in a graph form.

2.1.1.7.1 Description & Priority

The requirement is a Sales report, this allows the users to view the matrix of sales. The user can view weekday sales, monthly sales, busy time a day and identify what the most sold and popular menu is.

2.1.1.7.2 Use Case

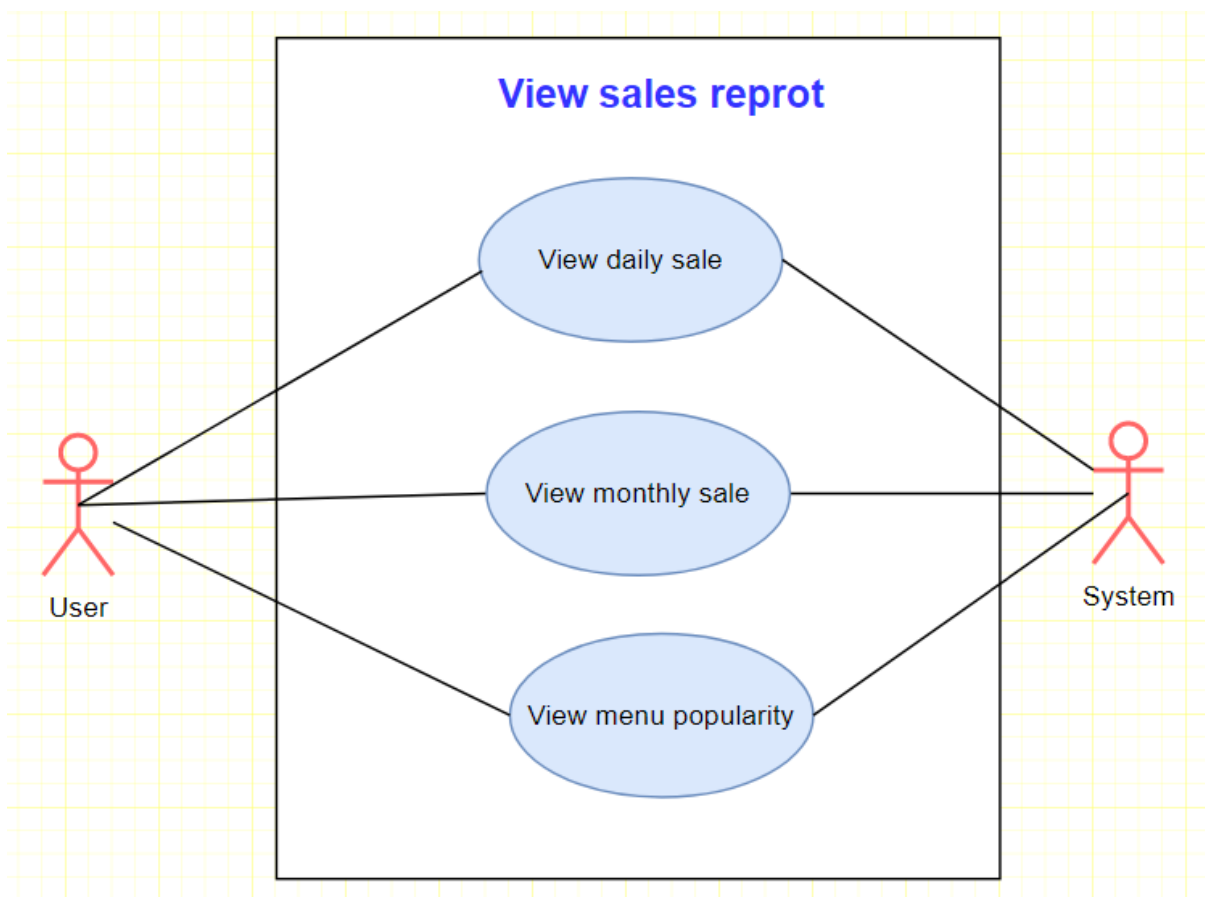
Scope

The scope of this use case is to allow the user to view sales reports on a weekday and monthly basis and to view the busy time of a day and popular menu.

Description

This use case describes the View sales report.

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application. There should be sales.

Activation

This use case starts when the user clicks the Sales report section in the navigation bar.

Main flow

1. The system identifies the user clicking the sales report section in the navigation bar.
2. The system displays the sales report page with the sales matrix in graphs.

Termination

The system presents the user with a sales report of the application.

Post condition

The system goes into a wait state.

[2.1.1.8. Requirement 8: Make table side order](#)

The requirement is to Make table side order. The user will be able to make an order at a table.

[2.1.1.8.1 Description & Priority](#)

The requirement is to Make table side order. This allows the users to make an order at a table by scanning the QR code. When the QR code is scanned, the user will get a notification that directs users to the menu ordering page. Once the user makes an order, the order details will be sent to the POS system.

[2.1.1.8.2 Use Case](#)

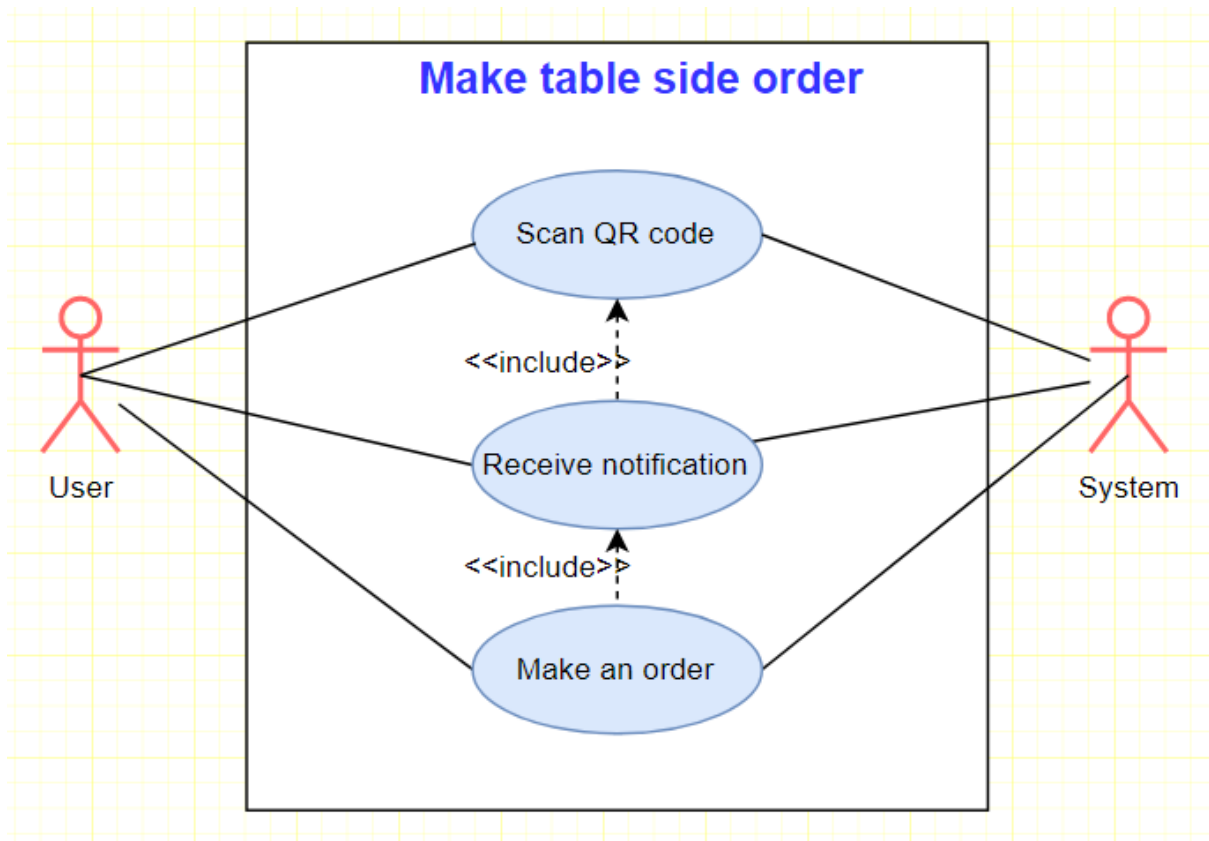
Scope

The scope of this use case is to allow the user to make table-side order by scanning the QR code on the table.

Description

This use case describes the Make table side order.

Use Case Diagram



Flow Description

Precondition

The QR code printed from the QR code generation section must be located on the table. The users have their mobile devices. The network connection is stable.

Activation

This use case starts when the user scans the QR code on the table.

Main flow

1. The user scans the QR code located on the table with a mobile device.
2. The device pops up a notification on user's mobile device screen.
3. The user clicks the notification.
4. The system directs the user to the order site.
5. The user making order via mobile device.
6. The order information will be sent to the POS system. (See E1)

Exceptional flow

E1 : Network connection error

1. The network connection is unstable.

2. The table-side order will not be transmitted to the POS system.
3. The user will be informed and try to order again.

Termination

The system presents the order menu page.

Post condition

The system goes into a wait state.

2.1.1.9. Requirement 9: Order Payment

The requirement is to Order Payment. The user can pay the bill by card or cash.

2.1.1.9.1 Description & Priority

The requirement is Order Payment, this allows the users to choose the payment method and pay the bill. This is the essential requirement because once the payment process is done, it reflects the sale report.

2.1.1.9.2 Use Case

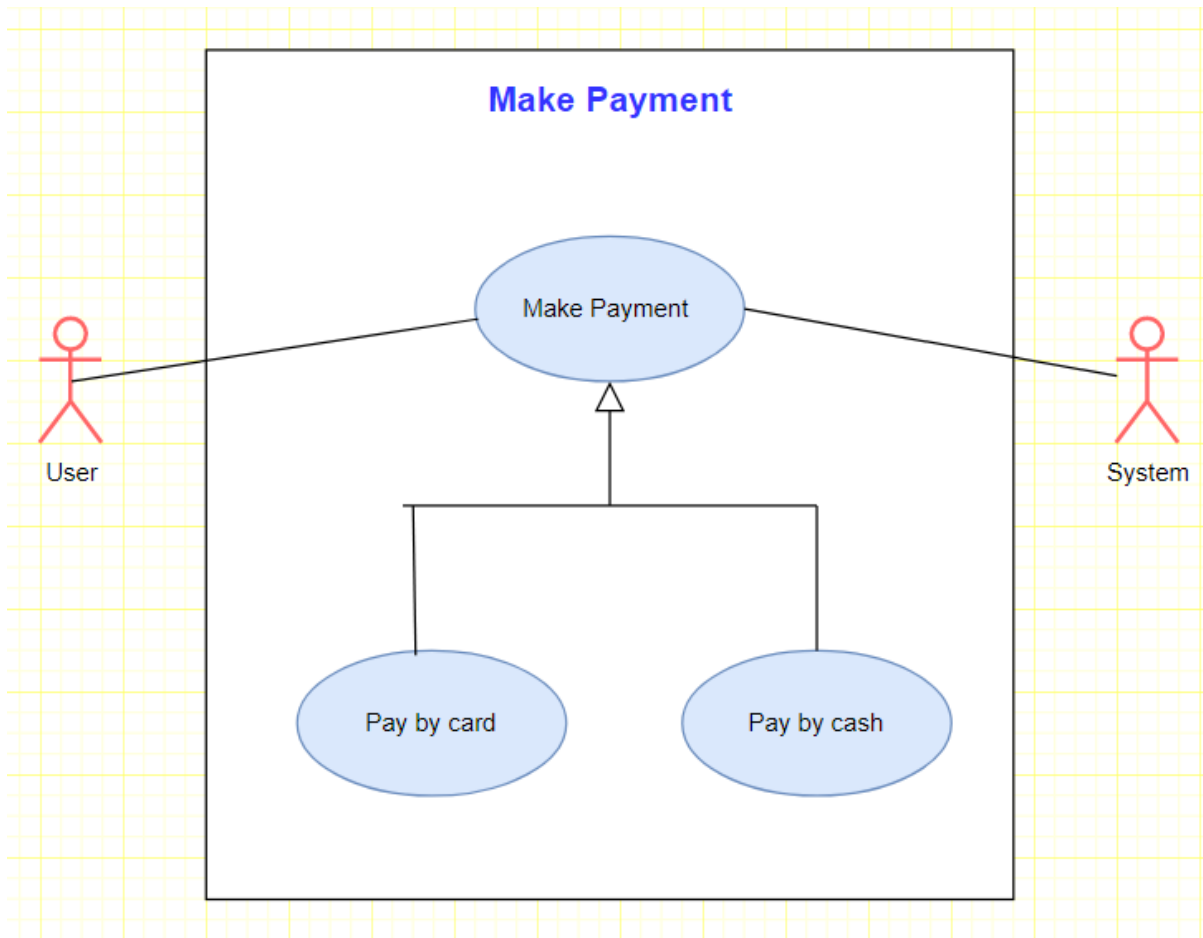
Scope

The scope of this use case is to allow the user to choose the payment type and pay the bill. Users can view the total cost of a specific table order and operate the payment.

Description

This use case describes the Payment.

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application and the customer has an order to pay.

Activation

This use case starts when the user clicks the POS section in the navigation bar and then click a specific table order box on View Order page (Requirement 5).

Main flow

1. The system identifies the user clicking a specific table order box on View Order page.
2. The system shows the total cost of the order.
3. The user clicks cash payment type. (See A1)
4. The user types the amount of money received from a customer.
5. The system displays the amount of changes.
6. The payment transaction takes place.

Alternate flow

A1 : Pay by card

1. The user chose to pay by card.
2. The system accepts the card.
3. The use case continues at position 6 of the main flow.

Termination

The system presents the user with the POS page of the application.

Post condition

The system goes into a wait state.

[2.1.1.10. Requirement 10: Logout](#)

The requirement is Logout. The logged in user can log out by clicking the logout section in the navigation bar.

[2.1.1.10.1 Description & Priority](#)

The requirement is Logout, this allows the users to log out of the application. The user can click the logout section and be logged out and the main page will be loaded. This page is important because it prevents other users from accessing the system.

[2.1.1.10.2 Use Case](#)

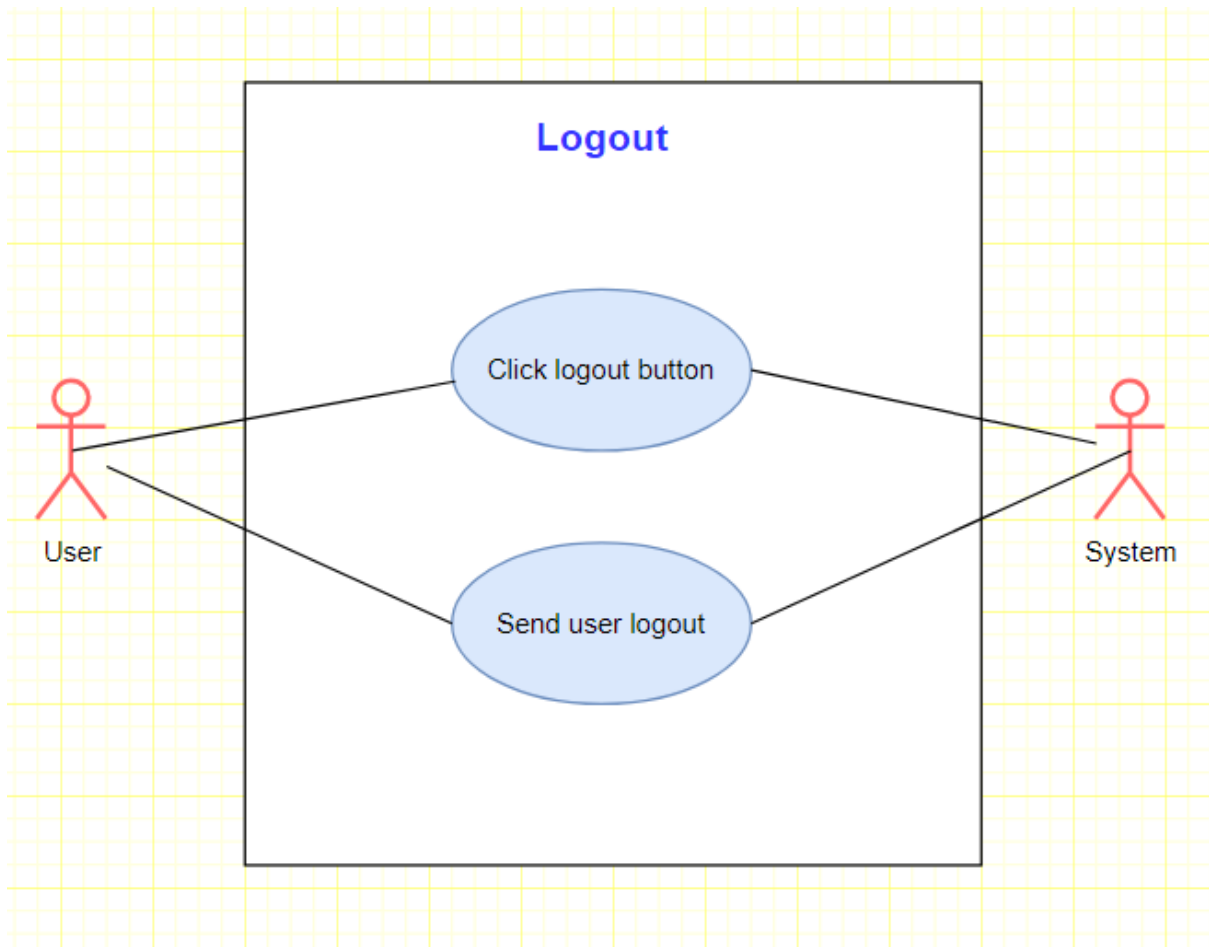
Scope

The scope of this use case is to allow the user to log out.

Description

This use case describes the Logout.

Use Case Diagram



Flow Description

Precondition

The user must have logged in to the application and clicked the logout button.

Activation

This use case starts when the user clicks the logout section in the navigation bar.

Main flow

1. The system identifies a user is logged in.
2. The user clicks the logout section.
3. The system checks that the logout section was clicked. (E1)
4. The logout process takes place.
5. The system loads the login page.

Exceptional flow

E1 : Logout section not clicked

1. The user did not click the logout section.
2. Logout action will not take place.
3. The system will let the user stay in the log-in state.

Termination

The system presents the user with the main page of the application.

Post condition

The system goes into a wait state.

2.1.2. User Requirements

There are few requirements for the POS users to use POS system and requirements for customer to use QR code ordering system.

POS User: Only authorised employ can access the POS system, so registration is required.

Device: This application is for table side order, so customer should have their own mobile device to scan the QR code and make order.

Internet Access: Internet connection should be stable to place order.

2.1.3. Environmental Requirements

MyPOS application requires following environmental requirements.

Visual Studio Code: I used Visual Studio Code to create and code my application. Each page is written with HTML, PHP, JavaScript, jQuery and designed using CSS and Bootstrap.

phpMyAdmin: I used phpMyAdmin to manage database.

XAMPP: I used XAMPP web server to run and test my application.

Windows Laptop: I used HP laptop with Visual Studio Code installed on it to develop my application.

Internet Access: The application requires internet access and for QR code ordering system.

2.1.4. Usability Requirements

Usability requirements is an important part for application development. POS application design should be easy to learn, understand and use. I used consistent navigation bar with icons located on the left side of each page of the application so that user can reach the desired page quickly. The navigation bar also highlights the section where the user is located currently.

2.2 Design & Architecture

My POS application starts from the customer. When customer scan the QR code located on the customer's table, it opens the table order page, and user take orders. Once the customer completes the order, the order confirmation page is loaded. The order data is

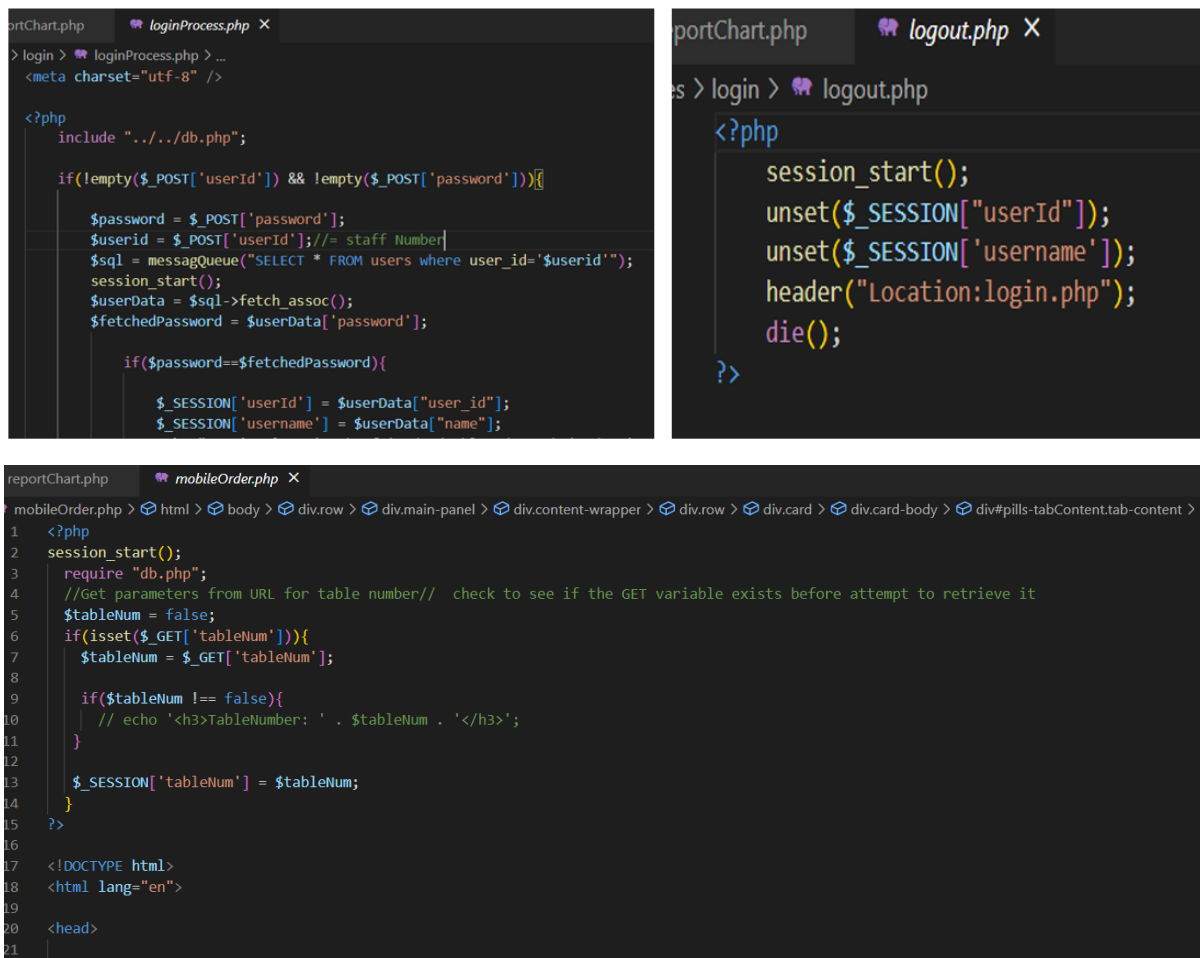
saved in database. The saved order data displays on POS with table name and ordered item names and amount. When staff changes the status of order items, the status is also saved on database. After the payment, the sales information is saved, and it will be reflected to sales report.

2.3 Implementation

I will present some key functions and code for the functions.

2.3.1 Login/Logout

In login/logout function, session is used to allow authorized user can access and use the application. On top of that, QR code should contain unique table number, so when I created QR code I added table number as a query parameter in URL. When the table order is done, mobile order page take the parameter as a session variable and the table order is saved in the database with the table number.



The image displays three screenshots of PHP code editors, likely VS Code, showing code for login, logout, and mobile order processing.

Top Left Screenshot (loginProcess.php): Shows the login logic. It includes a database connection, checks for user ID and password, queries the database for user data, and sets session variables for user ID and username upon successful login.

```
<?php
include "../db.php";

if(!empty($_POST['userId']) && !empty($_POST['password'])) {
    $password = $_POST['password'];
    $userId = $_POST['userId']; // staff Number
    $sql = mysqli_query($conn, "SELECT * FROM users where user_id='$userId'");
    session_start();
    $userData = mysqli_fetch_assoc($sql);
    $fetchedPassword = $userData['password'];

    if($password==$fetchedPassword){
        $_SESSION['userId'] = $userData["user_id"];
        $_SESSION['username'] = $userData["name"];
    }
}
```

Top Right Screenshot (logout.php): Shows the logout logic. It starts a session, unsets the user ID and username session variables, and sends a header to redirect to login.php before exiting.

```
<?php
session_start();
unset($_SESSION['userId']);
unset($_SESSION['username']);
header("Location:login.php");
die();
?>
```

Bottom Screenshot (mobileOrder.php): Shows the mobile order processing logic. It starts a session, requires the database connection, and checks for a table number in the URL. If found, it sets the table number as a session variable.

```
<?php
session_start();
require "db.php";
//Get parameters from URL for table number// check to see if the GET variable exists before attempt to retrieve it
$tableNum = false;
if(isset($_GET['tableNum'])){
    $tableNum = $_GET['tableNum'];

    if($tableNum != false){
        // echo '<h3>TableNumber: ' . $tableNum . '</h3>';
    }

    $_SESSION['tableNum'] = $tableNum;
}
?>
```

```

reportChart.php  mobileOrderToDatabase.php X
mobileOrderToDatabase.php > ...
1
2 <?php
3 session_start();
4 require "db.php";
5 $tableNum= $_SESSION['tableNum'];
6
7 $cartlist = $_POST['cartlist'];
8 foreach($cartlist as $row){
9
10     $productName = $row['name'];
11     $quantity = $row['count'];
12     $unitPrice = $row['price'];
13     $totalPrice = $row['total'];
14     $sql = messagQueue("INSERT INTO purchase_order (table_number,product_name,quantity,
15     VALUES('".$tableNum."','".$productName."','".$quantity."','".$unitPrice."','".$total
16
17 }
18

```

2.3.2 Add product/ upload image

When a new product is added, it should include the image of the product, so to upload images, the image was encoded with base 64 and saved as BLOB in database.

```

ortChart.php  addProducts.php X
> products > addProducts.php > ...
<?php
session_start();
require "../db.php";
if(!isset($_SESSION['userId'])){
header("location:login.php");
die();
}

$productName = $_POST['pname'];
$category = $_POST['category'];
$price = $_POST['price'];
$description = $_POST['description'];
//image
$filename = $_FILES["image"]["name"];
$tempname = $_FILES["image"]["tmp_name"];
$image = "../images/".$filename;
$imageContent = base64_encode(file_get_contents(addslashes($image)));

$sql = messagQueue("INSERT INTO product (product_name,description,category_name,product_unit_price,image) VALUES('".$productName."','".$description
?>

<meta charset="utf-8" />
<meta http-equiv="refresh" content="0 url=products.php">

```

2.3.3 Display QR code

The list of QR code is displayed when user click the 'generate QR code' button. To hide the list of QR code and show them after button clicked, I used hide/show toggle.

```

<div class="row">
  <div class="col-md-2"></div>
  <div class="col-md-3 div-toggle" data-target="div-toggle">
    <button type="button" class="btn btn-inverse-primary btn-fw" data-show=".display" id="showQrcode" >Generate QR code</button>
  </div>
  <div class="col-md-3">
    <a href="download.php"> <button type="button" class="btn btn-inverse-primary btn-fw">Download all QR code</button></a>
  </div>
  <div class="col-md-3">
    <a href="printQRcode.php"> <button type="button" class="btn btn-inverse-primary btn-fw" id="printQR">Print all QR code</button></a>
  </div>
  <div class="col-md-1"></div>
</div>
<br>

<div class="display grid grid-template-columns-2 hide" >
  <?php
    while ($row = $sql->fetch_assoc())
    {
      $image = $row['image'];
    }
  >

  <div class="item">
    <h3> <?php echo $row['table_number']; ?></h3>
    
  </div>

  <?php
}
?>

```

```

reportChart.php  qr.php  JS qrcode.js  X
> JS qrcode.js > ...
1  $(document).on('click', '.div-toggle', function() {
2    var target = $(this).data('target');
3    var show = $("button:focus", this).data('show');
4    $(target).children().addClass('hide');
5    $(show).removeClass('hide');
6  });
7  $(document).ready(function(){
8    $('.div-toggle').trigger('click');
9  });

```

2.3.4 Sales Report

I used PHP to retrieve sale report data (weekday sales, busy time, monthly sales and popular menu) from database and used ajax to display them in graphs. When I took the data from the database using PHP script, I encoded the data in JSON file using **echo json_encode(\$result)** like bellowing.

```

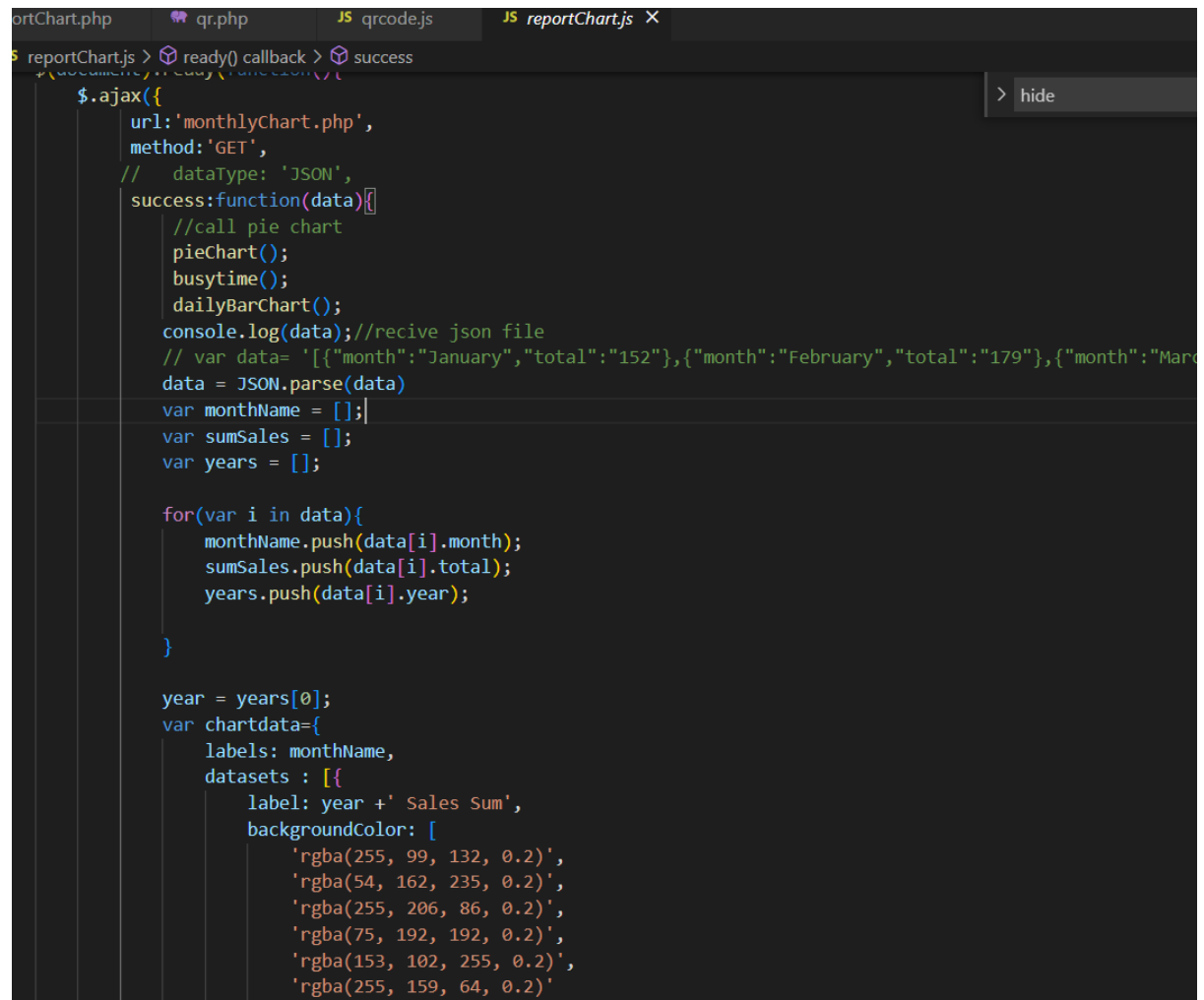
[{"product_name":"Fanta","totalQuantity":"70"},
{"product_name":"Americano","totalQuantity":"50"},..]

```

```
[{"total":"200","dayname":"1"}, {"total":"74","dayname":"2"},
{"total":"169","dayname":"3"}, {"total":"383","dayname":"4"},
{"total":"198","dayname":"5"}, {"total":"215","dayname":"6"},
{"total":"294","dayname":"7"}]
```

```
[{"year":"2022","month":"Jan","total":"152"},
{"year":"2022","month":"Feb","total":"179"},
{"year":"2022","month":"Mar","total":"229"},
{"year":"2022","month":"Apr","total":"267"},
{"year":"2022","month":"May","total":"706"}]
```

After retrieve the data as JSON format, I parsed the data and put the values into corresponding key array to display as the graph form using ajax like below. I used same approach for four sale report charts(two bar charts, one scatter chart and one pie chart).



```
reportChart.js > ready() callback > success
$.ajax({
  url: 'monthlyChart.php',
  method: 'GET',
  // dataType: 'JSON',
  success: function(data) {
    //call pie chart
    pieChart();
    busytime();
    dailyBarChart();
    console.log(data); //receive json file
    // var data= '[{"month": "January", "total": "152"}, {"month": "February", "total": "179"}, {"month": "March", "total": "229"}, {"month": "April", "total": "267"}, {"month": "May", "total": "706"}]';
    data = JSON.parse(data);
    var monthName = [];
    var sumSales = [];
    var years = [];

    for(var i in data){
      monthName.push(data[i].month);
      sumSales.push(data[i].total);
      years.push(data[i].year);
    }

    year = years[0];
    var chartdata = {
      labels: monthName,
      datasets: [{
        label: year + ' Sales Sum',
        backgroundColor: [
          'rgba(255, 99, 132, 0.2)',
          'rgba(54, 162, 235, 0.2)',
          'rgba(255, 206, 86, 0.2)',
          'rgba(75, 192, 192, 0.2)',
          'rgba(153, 102, 255, 0.2)',
          'rgba(255, 159, 64, 0.2)'
        ]
      }]
    };
  }
});
```

```

    data: samples
  }
  };

  var bar = $('#barChart');
  var barGraph = new Chart(bar, {
    type: 'bar',
    data: chartdata
  });
},
error: function(data){
  console.log(data);
}
});

```

2.3.5 Change order status

MyPOS provides Order Status section which allow staff to check the status of each item and change the status of order. Order Status page is divided into two parts. Order status is on the right side of page and whole tables box with list in is on the right side of page. When a food item is ready, and staff ticks the check box of the item, then the colour of the item name in corresponding table box is changed to green to show that the item is served and done. To do this, I used jQuery. Firstly, I check if the check box is check or not. When unchecked item is checked, I extracted the product name and table number and iterate through the tables orders section and find the same table number and find the same product name in that table and changed CSS of the item name.

```

//when checkbox is checked, the text in table box color is changed.
$(document).ready(function(){
    $(".checkboxbox").click(function(){

        if ($(this).attr('checked')) {
            $(this).removeAttr('checked');

            //table status section
            var label = $(this).parent();
            var pname = label.attr('pname');
            var tableNum = label.attr('tableNum');
            var product_name = pname;

            var li_Items = $('div.popover-static-demo div');
            li_Items.each(function(i, div) {
                var li_Item = $(div);
                var tbTableNum = li_Item.find('h3.popover-header').attr('tableNum');

                if(tbTableNum == tableNum){

                    var li_pname_items = li_Item.find('div.popover-body').find('li.pname')
                    li_pname_items.each(function(i, li) {

                        var li_pname_item = $(li);
                        var tbPname = li_pname_item.attr('productName');

                        if(tbPname == pname){

                            li_pname_item.css( "color", "#737F8B" );
                            li_pname_item.css( "font-weight", "normal" );
                            return false; // breaks
                        }
                    }
                }
            }
        )
    }
    )
}
)

```



```

}
else { // if not checked
    $(this).attr('checked', 'checked');

    //table status section
    var label = $(this).parent();
    var pname = label.attr('pname');
    var tableNum = label.attr('tableNum');
    var product_name = pname;

    var li_Items = $('div.popover-static-demo div');
    li_Items.each(function(i, div) {
        var li_Item = $(div);
        var tbTableNum = li_Item.find('h3.popover-header').attr('tableNum');

        if(tbTableNum == tableNum){

            var li_pname_items = li_Item.find('div.popover-body').find('li.pname')
            li_pname_items.each(function(i, li) {

                var li_pname_item = $(li);
                var tbPName = li_pname_item.attr('productName');

                if(tbPName == pname){

                    li_pname_item.css( "color", "#1E5631" );
                    li_pname_item.css( "font-weight", "bold" );
                    return false; // breaks
                }

            });
        }
    });
} //if
}

```

2.3.6 Change calculation

jQuery and JavaScript are used to display number that is pressed from number pad and calculate the change on the Payment page.

```

js > JS numberPad.js > ...
1  ✓ $( '.btn-number1Btn' ).on( 'click', function() {
2
3      var clickedNum = $(this).text();
4      var currtBtnVale = $(this).attr( 'id' );
5
6  ✓      if( clickedNum == currtBtnVale ) {
7          $( '#pressedNumDisplay' ).text( "€" );
8
9      }
10 ✓      else {
11          $( '#pressedNumDisplay' ).text( $( '#pressedNumDisplay' ).text() + ( clickedNum ) );
12      }
13
14  } )
15

```

```

<script>

function cashBtn() {
    var recivedCashText = document.getElementById( "pressedNumDisplay" ).textContent;
    var recivedCash = recivedCashText.replace( /[ (€)]/g, '' );
    var toTalPriceText = document.getElementById( "totalPrice" ).textContent;
    var toTalPrice = toTalPriceText.replace( /[ (€)]/g, '' );
    var change = recivedCash - toTalPrice;
    document.getElementById( "changes" ).innerHTML = "€ " + change;


    window.location.href = "tableOrders.php";
}

```

2.4 Graphical User Interface (GUI)

1. Register & Login

Register page take name, staff number, email address, password and photo as inputs. User can click submit button to register or come out of the section by clicking cancel button. Login pages take staff number and password as inputs. User can login by pressing login button. Unregistered user can register by pressing register button.




Name

Staff Number

Email address

Password

Staff Photo




Enter your ID

Enter Password

2. QR code generation

This page has three buttons for QR code generation, download and print. When download button is clicked, we can find that the zip file is downloaded. When print QR code button is clicked, the print window is open.



Staff : admin

- POS
- Staff Pages
- QR Code**
- Products
- Sale Report
- Logout

QR Code

1




2

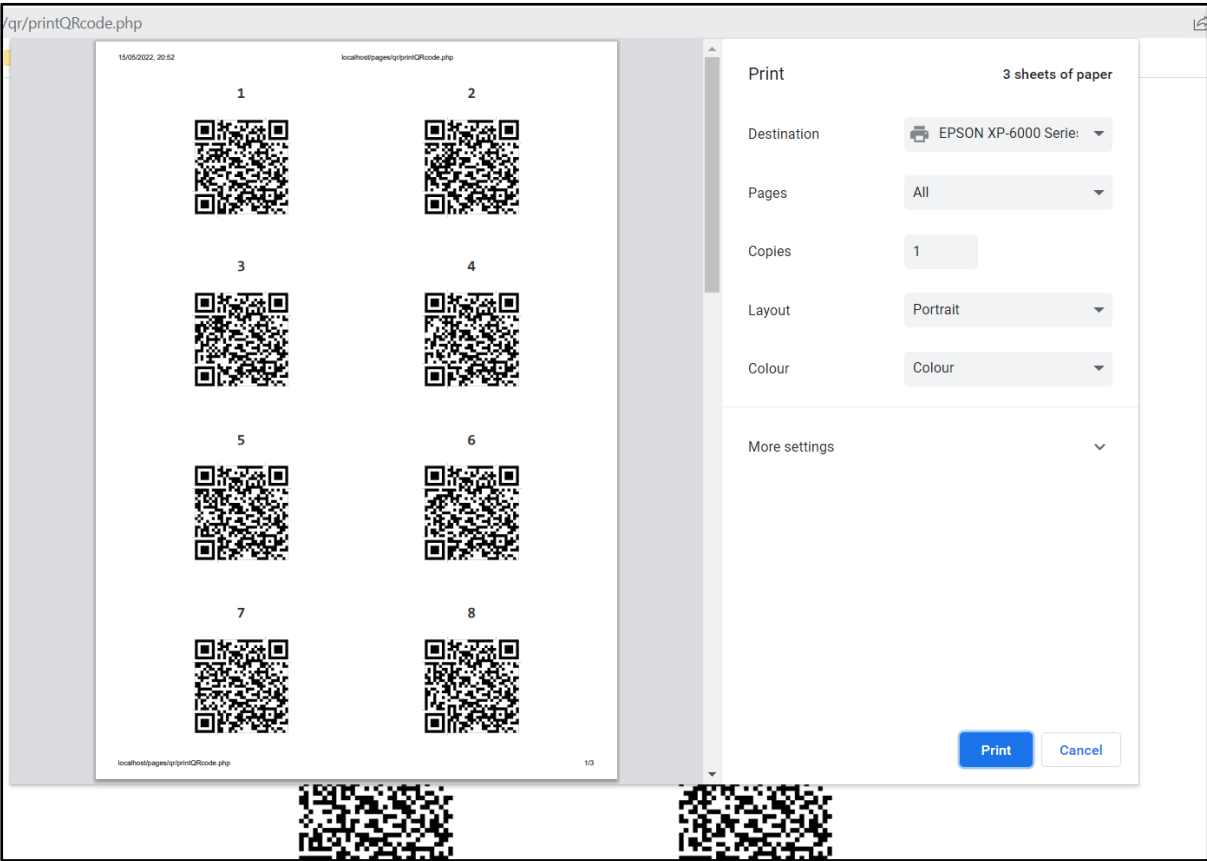
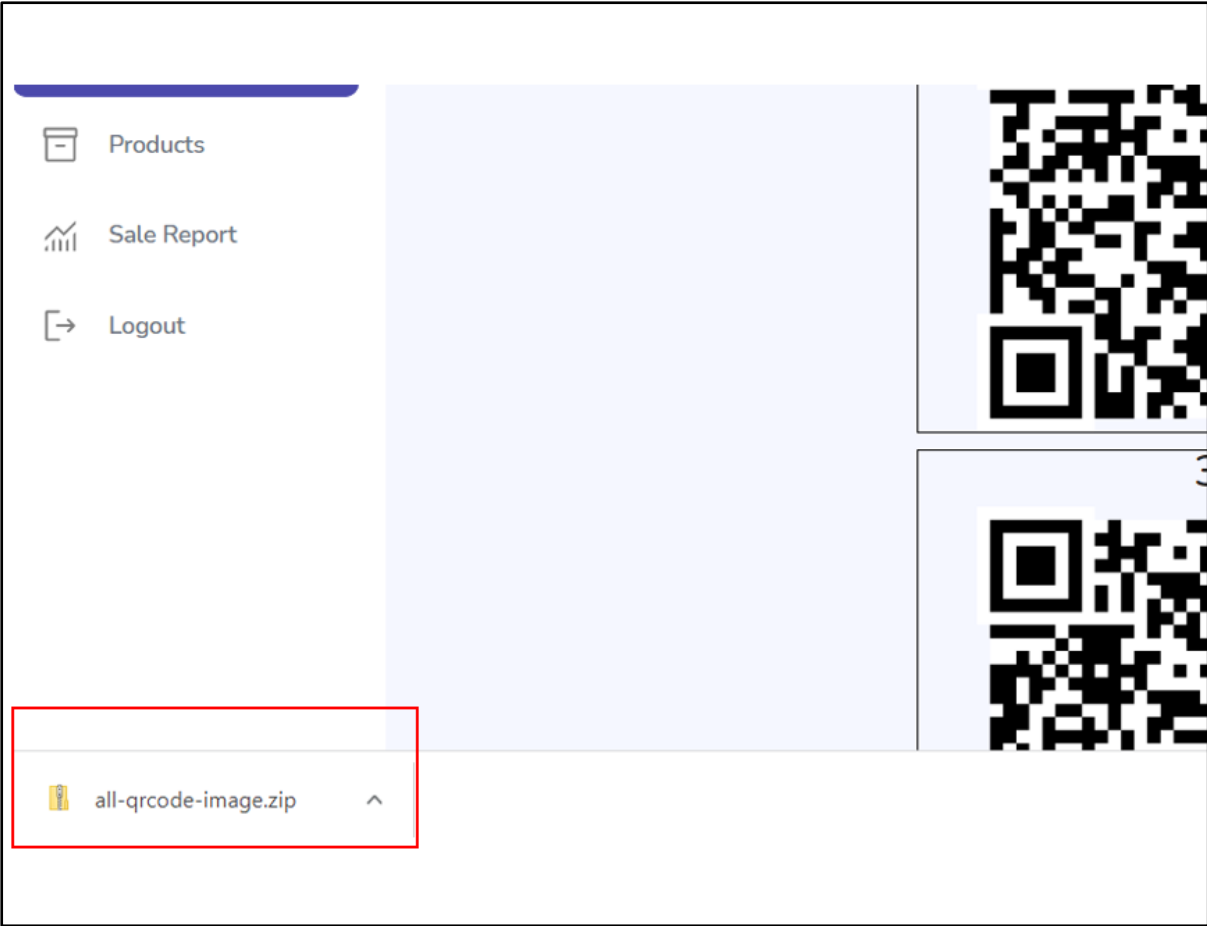


3



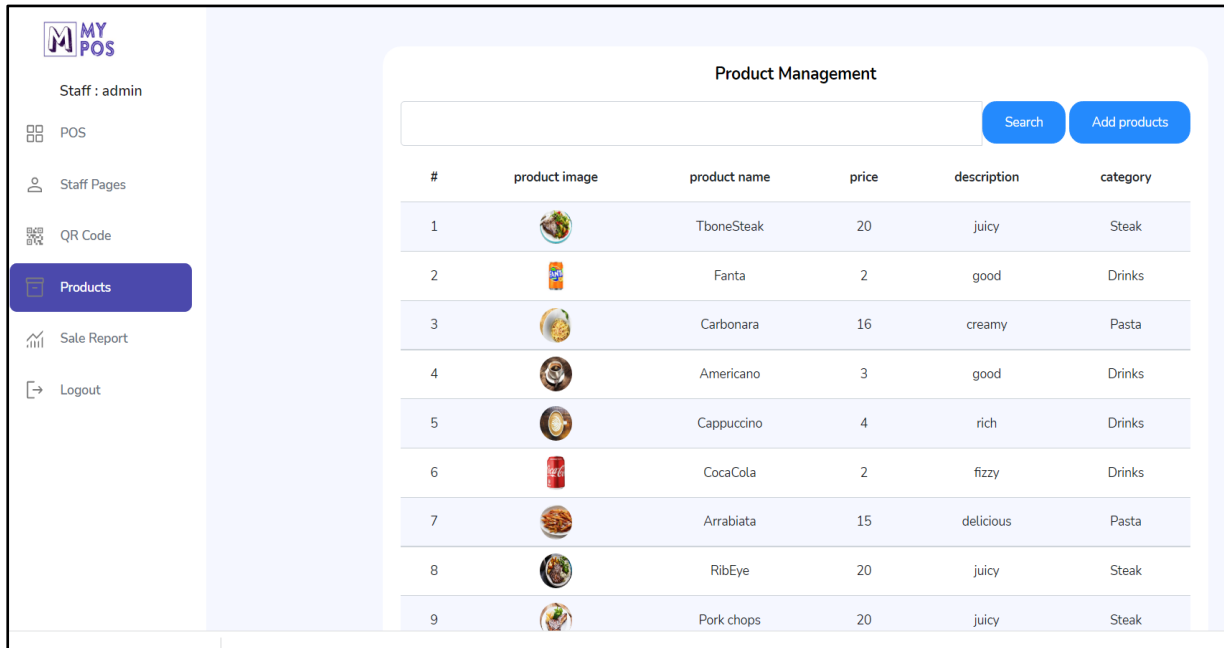
4














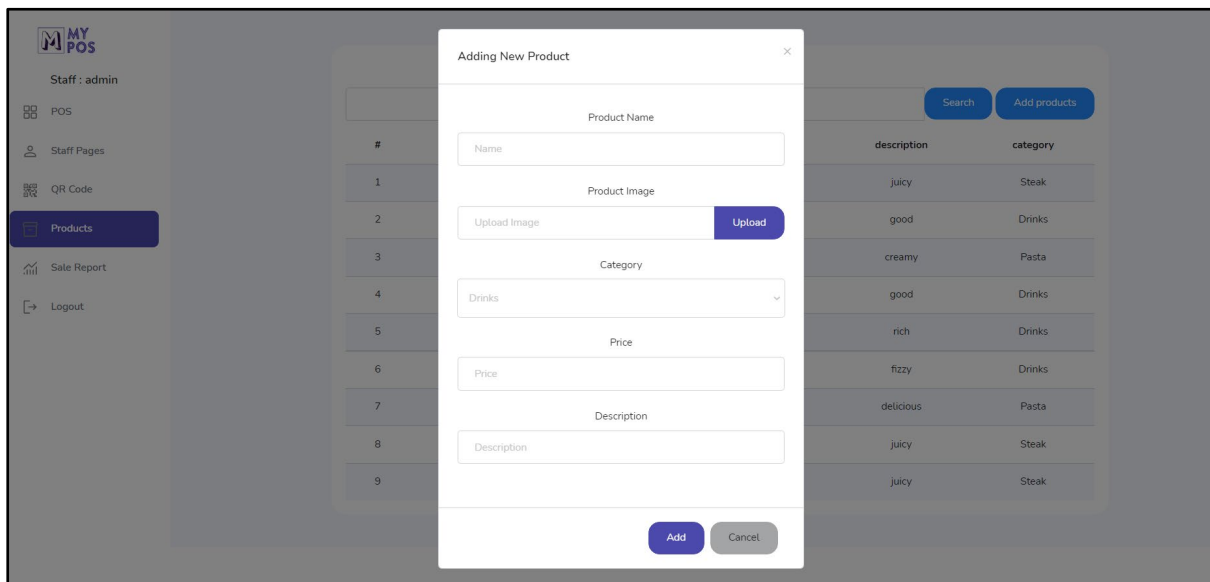
3. Add products

This page displays the list of the product with its name, image, unit price, description and category. User can add a new product by pressing the add products button.



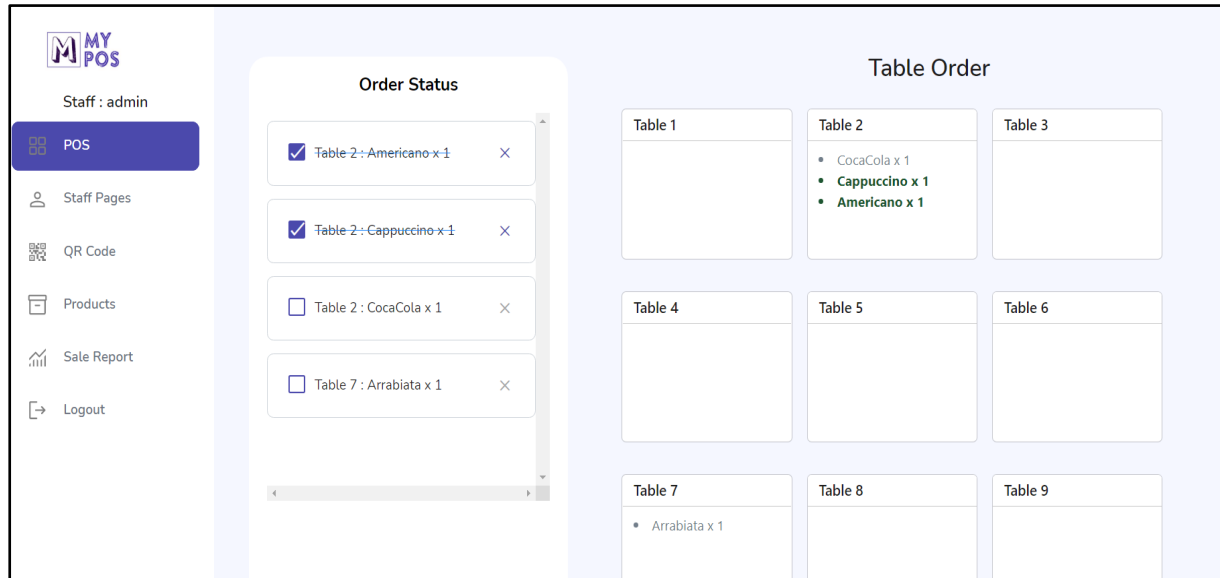
#	product image	product name	price	description	category
1		TboneSteak	20	juicy	Steak
2		Fanta	2	good	Drinks
3		Carbonara	16	creamy	Pasta
4		Americano	3	good	Drinks
5		Cappuccino	4	rich	Drinks
6		CocaCola	2	fizzy	Drinks
7		Arrabiata	15	delicious	Pasta
8		RibEye	20	juicy	Steak
9		Pork chops	20	juicy	Steak

The below screen shot is the pop-up modal to add new product. It takes product name, image, category, unit price and description as inputs. User can complete add operation by pressing Add button or close the modal to cancel the adding operation.



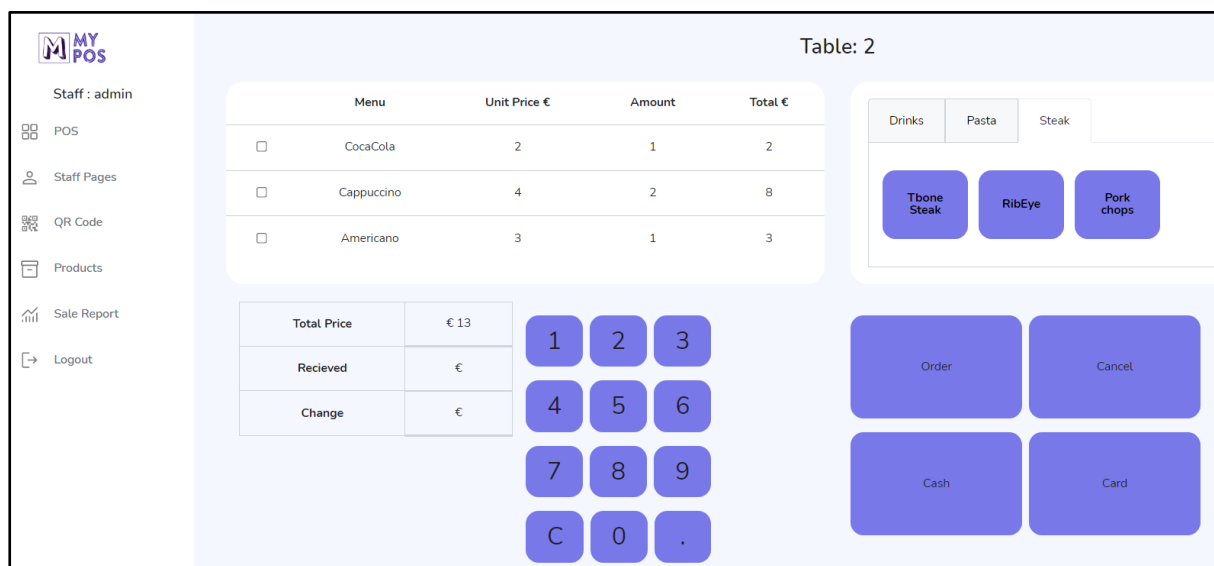
4. View order

This page shows the order status of each item with table number. First two items are ready and checked, so the items are marked as green in table 2. Staff can close the served item box by pressing the x icon.



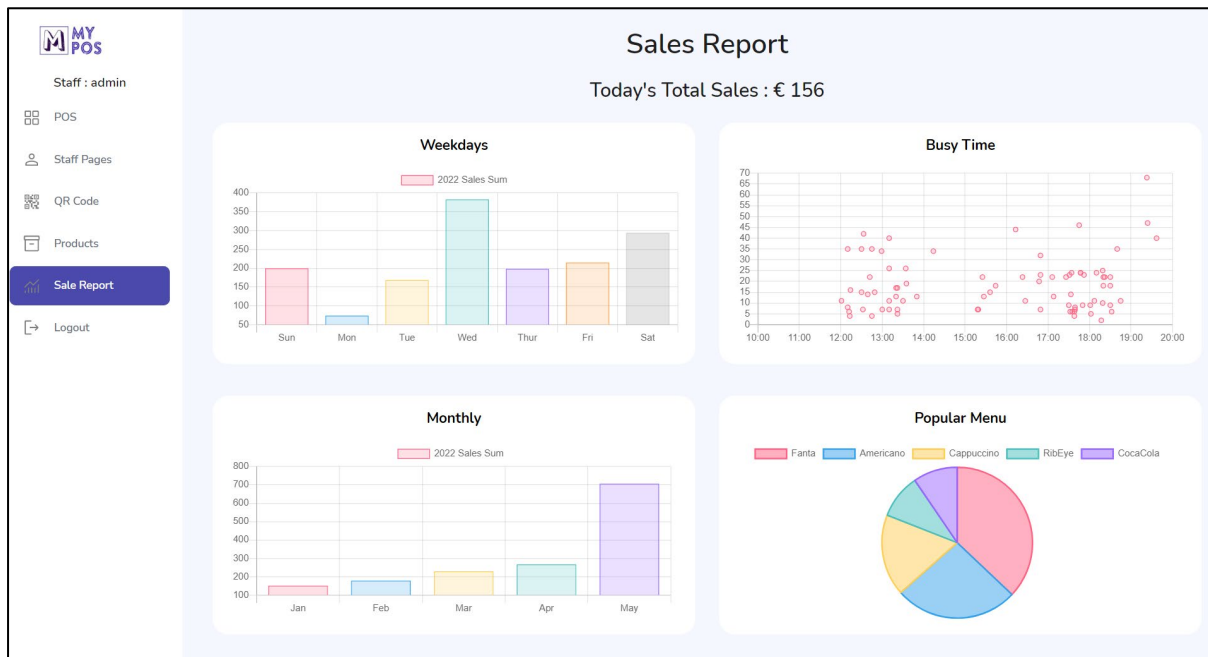
5. Order Payment

This page is reached by clicking the table box (table 2) on the above page. On this page, ordered items are listed and here extra item can be added by staff. I added one more Cappuccino, so the amount become 2 total cost is re calculated and displayed.



6. Sales report

This page shows the sales total of current date, weekday sales and monthly sale in bar chart, busy time of a day in scatter chart, and popular menu in pie chart.












7. Manage Staff

This page shows staff details with name, email address and staff number. The staff details can be edited and deleted by corresponding buttons.

MY POS
Staff : admin

POS
Staff Pages
QR Code
Products
Sale Report
Logout

User Profile

 Esther dnstpqhrt@naver.com Staff Number: 141516 Edit Delete	 Andy candy@hotmail.com Staff Number: 141566 Edit Delete	 Mark manitto@gmail.com Staff Number: 141565 Edit Delete
 Jina mimi@gmail.com Staff Number: 141525 Edit Delete	 Christina hahaha@gmail.com Staff Number: 141555 Edit Delete	 Filip filip20@gmail.com Staff Number: 141579 Edit Delete
 Jane happyj@gmail.com Staff Number: 141523 Edit Delete	 admin Ryan@hotmail.com Staff Number: admin Edit Delete	 June june@hotmail.com Staff Number: hoho Edit Delete

2.5 Testing

When I planned the project development, I planned to do the Unit test using PHPUnit to test functions. But when I developed the application, I didn't make any PHP functions or classes. I mainly used PHP to deal with data CRUD for the database and for some calculations. So, I was not able to do the PHPUnit testing. Instead, I continuously test manually if the requirement functions are working through the whole developing process. I ran the application on the browser. When I faced some errors, I fixed them right away. I was able to solve each error by many researching and learning. It was very beneficial to test a function on the development stage. Once one function is certainly working, and I moved to next function. I used alert with if statement to display the negative message for errors and positive message for correct operation to test if each function is working correctly as it is supposed to work. This way I tested all the requirement functions and database operation.

2.6 Evaluation

I continuously tested and evaluated my application while developing it. All functions are working correctly.

Login (V), Logout (V), display check box list for order statement (V), set and change the order status ready and done (V), display correct order data to each table (V), display staffs info details (V), edit and delete staff information (V), create QR code with unique URL for each table (V), QR code generation and display (V), QR code download (V), Print QR code (V), display products list (V), add new product with name, image, category, unit price and description (V), display sale report of weekdays sale (V), monthly sales (V), busy time (V), popular menu (V).

3 Conclusions

MyPOS is a web-based Point of Sale System and allows table-side order by scanning the QR code using the customer's own mobile device. It is very convenient for customers and requires less contact. It makes customer feel comfortable and less pressure to ready quickly, so this application gives good user experience. For the Staff and owner of a store, MyPOS allows them view orders as whole and manage the order status and gives insight for their sales by sales report section. Most of all, it is easy to learn and use. The store owner doesn't need to spend a lot of cost for licenses and maintenance. The limitation is that it needs further development to be used in real store.

4 Further Development or Research

For the further development, in terms of the QR code section, I would like to write code to take table numbers as input to generate QR code and display as many QR code as the table

numbers entered by user. For products section, I would like to change the basic table into the sortable table which has filter function, so that user can view specific category items. For sales report part, I would like to make show the popular menu per each category with three pie charts. For staff page, I would like to view more details of each staff by click staff photo and shows the details with the staff's schedule.

5 References

Achinga, C., 2021. *Shopping cart JS*. [Online]

Available at: <https://codepen.io/chrisachinga/pen/MWwrZLJ>

[Accessed 25 2 2022].

Bootstrapdash, n.d. *Bootstrapdash*. [Online]

Available at: <https://www.bootstrapdash.com/product/skydash-free/>

[Accessed 1 12 2021].

Chart.js, 2022. *Chart.js*. [Online]

Available at: <https://www.chartjs.org/docs/latest/>

[Accessed 25 04 2022].

Google, 2021. *Google Font*. [Online]

Available at: https://developers.google.com/fonts/docs/material_icons

[Accessed 09 05 2022].

sweetalert2, 2022. *sweetalert2*. [Online]

Available at: <https://sweetalert2.github.io/>

[Accessed 01 05 2022].

6 Appendices



National College of Ireland

Project Proposal

My POS

07 November 2021

BSc (Honours) in Computing

Software Development

2021/2022

Eunkyoung Kim

X18150365

X18150365@student.ncirl.ie

Contents

1.0	Objectives.....	42
2.0	Background	Error! Bookmark not defined.
3.0	State of the Art.....	43
4.0	Technical Approach.....	43
5.0	Technical Details	44
6.0	Special Resources Required	44
7.0	Project Plan	44
8.0	Testing.....	45

7 Objectives

The objective of this project is to create a web-based Point of Sale application for small size restaurants and cafeterias, which will be available for iPad and mobile devices. It allows a table-side ordering system with a QR code which drives a good user experience. Customers can sit at the table and make an order by scanning a QR code. When it is scanned, the notification will pop up and it leads to the menu site. When customers make an order, it will be sent to the POS app, so it reduces the staff contact with customers. And customers don't need to feel under pressure to decide quickly. The project will provide some functionalities such as login and logout, manage account, admin table, add products, view order, sales report(metrics), calculating sales, QR code generation, and payment by card and cash.

8 Background

Due to the Covid 19, social distancing and contactless becomes our new lifestyle. It leads some people to feel comfortable and safe to use online for ordering foods. Some people prefer to visit an offline store. So, it made me think of how I can merge the ideas. It allows customers to be in an offline restaurant and use table-side ordering service. This POS application is easy to use for users. Moreover, purchasing an expensive POS device can be a burden for small size businesses. This is because there are continuous additional costs for licenses and maintenance. My project idea will address these problems and provide restaurant owners and customers with safe and convenient ordering and payment services.

As I mentioned above about the functionalities of my application, the owner of a restaurant can **log in** and **log out** using a user id and password, **register**, and **manage the user(staff) account**. The owner can **set the table numbers** and admin tables which is draggable. It allows for a **setting category of products, add products**, its name, image, description, price, category of the product. The owner can see **view orders** as a whole and can manage orders via option buttons like 'The meal is done, canceled'. It also shows the **reporting metrics** in a graph which allows the owner to figure out the busy time and slower time during times and days so that it helps over schedule and under schedule. It will provide **daily, monthly sale reports**. The owner can identify the most popular menus and which menu makes customers come back. Downloadable QR codes will be generated according to the number of tables. In terms of payment, customers can pay by card and cash.

9 State of the Art

The POS system is essential for business and retail. There are various kinds of POS systems. It keeps evolving according to the trend of our lives, becoming easier to use and more efficient. I searched and studied some popular POS systems and the top 3 were Lightspeed, Square, and Clover. They provide sales and inventory management, vendor management, payment processing, customer, employee management, and analytics. They are well designed and increase the good user experience. My application will have the same features like them but the outstanding one of my applications is that it allows a table-side ordering system using QR code. Any POS system doesn't receive the order from QR code, in general, waiter/waitress goes to the table, directly receive customer request, register order on POS or take the online order for taking away, whereas my application is designed to allow customers to directly place orders using QR codes at their tables when they are ready, which minimize face to face contact. Therefore, at this point, I believe my application is outstanding.

10 Technical Approach

I will take an Agile development approach that can react fast to unpredictable change and minimize risk. I found that this approach is more effective than other approaches. This is because it increases customer satisfaction, and the project runs smoothly. Moreover, I chose to interview the expected use of my application. The user of my application is not only the owners of restaurants and stores but also customers who make orders. I already interviewed a person who is working in the restaurant sector and got his requirements, he also shared his experience as a customer in the restaurant. I am going to write the requirements specification based on that to identify requirements and will break down the expected functions into manageable sub-tasks. Whenever each function is completed, I will show it to the user and get feedback and revise it when it is needed according to the feedback.

11 Technical Details

The main technologies used are Visual Studio IDE, PHP, JavaScript, and firebase. Visual Studio Code is the IDE that I used to code to develop web applications, so I chose this for my application. PHP will be used for my dynamic and interactive web application. JavaScript will be used for my interactive web application. For the database, I will use MySQL that I am familiar with as I have been used it for my previous projects. To validate the functionalities and performance of my web application, I will do Web Testing which includes functional testing, visual testing, performance & load testing, and database testing. Postman will be used for testing. GitHub will be used for version control. The important approach and big challenge are generating of QR code and connecting it to various tables and integration of sending the ordered data from mobile to POS system.

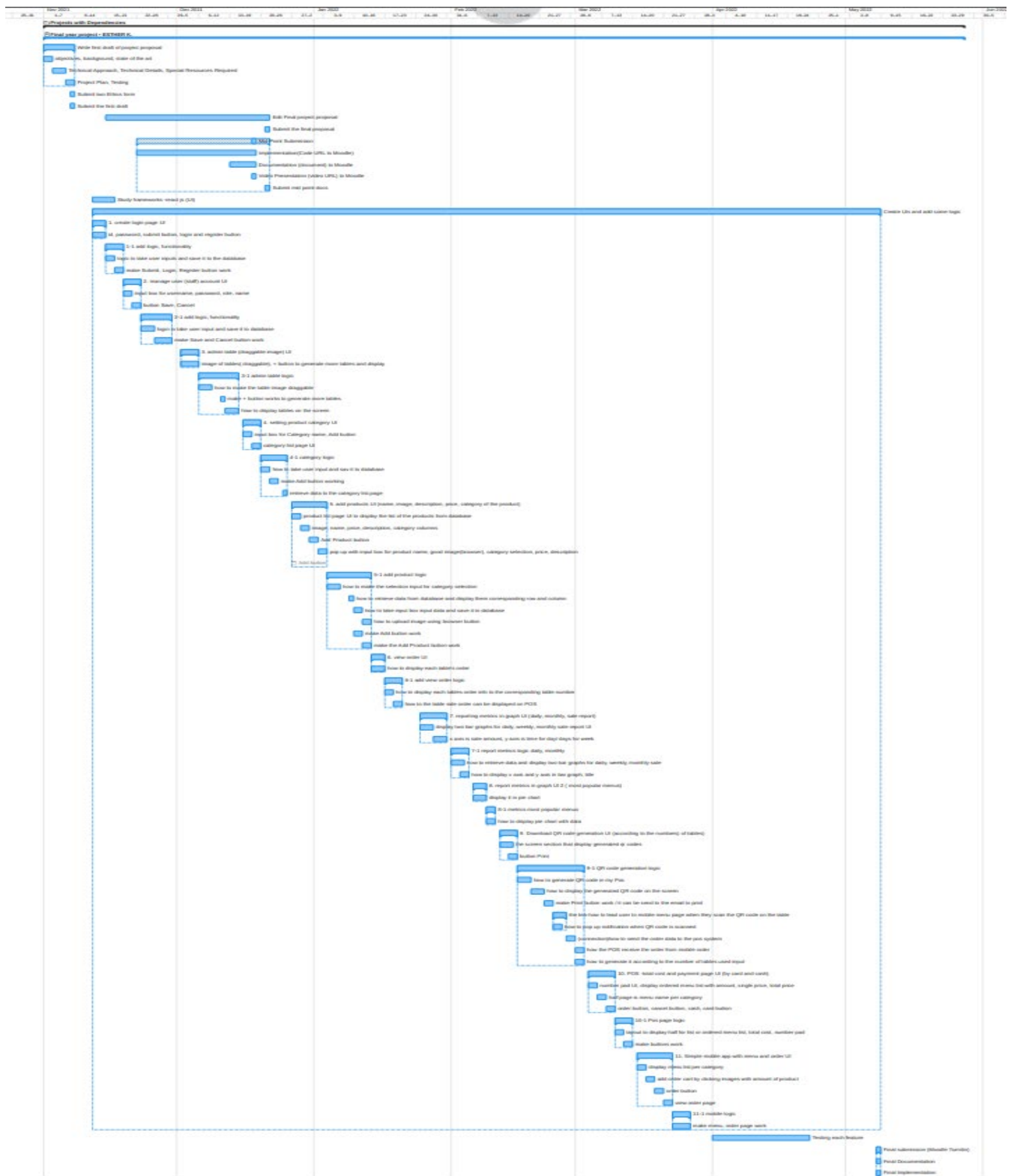
12 Special Resources Required

It requires only any devices which allow for QR code scanners because the customers should scan a QR code on the table, the QR code which was generated by the POS system.

13 Project Plan

The picture below is my project plan by timeline.

The link for my project plan file is https://studentncirl-my.sharepoint.com/:b:/g/personal/x18150365_student_ncirl_ie/ETXJ4WrgQQVMgVlodCboGUABYj0zhsxyME739r3VVHVabg?e=GCfjHc



14 Testing

To validate my web application, I will process Web testing which includes functional testing, usability & visual testing, performance & load testing, and database testing. End-user testing will be done. In terms of functional testing, I will check if the POS application is working as expected according to the defined specification(requirement). I will check whether it has any broken links, will check all the links or navigation are directed to the expected URLs. Input forms will be checked as well by testing if all

the input fields are accepted and see what would happen when any required fields don't fill out. I will test the workflow business scenario. The following workflow should be checked. A user register in the POS application, log in, add new staff, set a table number, admin table, add products, takes an order, view order, payment, check sales reporting, and logout.

In terms of usability & Visual Testing, I will check if the web application is consistent visually. I am also going to run the application on my device and test whether it works as intended and whether the design does not look broken. Especially, my POS application is available for iPad and mobile, so it will be checked if it has any breakpoints across from desktop, iPad, and mobile devices.

For the performance testing, the logic of each function of the application will be tested right after it is written to check each function is operating as it is supposed to work.

Moreover, my project requires communication between the mobile application and POS application, so an integration test will be done if the connection works properly. As I submitted in Ethic Application Form, end-users will run the app on their device and test it.

14.1 Project Plan

The picture below is my project plan by timeline. The link for my project plan file is

https://studentncirl-my.sharepoint.com/:b:/g/personal/x18150365_student_ncirl_ie/ETXJ4WrgQQVMgVlodCboGUABYj0zhsxyME739r3VVHVabg?e=GCfjHc

This month, I thought about my project idea and was able to decide what it would be. I decided to make a web-based POS system (Point of Sale) available for iPad and mobile, which allows customers a contactless ordering system with QR code (table-side ordering). When it is scanned, the notification will pop up and it leads to the menu site. When a customer makes an order, the order will be sent to the POS system that I will make. This system provides owners with an admin table, kitchen display, total reviews such as total visits, total sales, customer's favorite. It also shows the busy time and slower time. After I finalized my idea, I made a Project Pitch video and submitted it. It was reviewed by my supervisor. I also researched some other POS systems. I went through their functionalities, features, and UIs. This research was very helpful. It gave me more ideas of what I want to add and make difference in my project. I reviewed how to write a User Case.

My plan to follow

I will contact my supervisor to know if my project idea is approved. Once my idea is approved, I am going to start writing a project proposal. I need to refine some requirements and features for my idea. I need to figure out what tools and technical skills I need. To do so, I need further research to refine some requirements and will investigate what tools and technical skills I need to develop my application. I also list the works and break down those works into sub works and make a plan by timeline.

November Monthly Journal

My achievements

I researched for QR code generation service for my project and watched some tutorials for React JS and start to write code using React JS, and wrote a project proposal.

My plan to follow

I need to figure out how to use QR code generation in my application. I will start to develop UI.

December Monthly Journal

My achievements

I created a graphic image icon, favicon, for my application called 'My POS'. I developed UI of some functionalities such as register, login, QR code generation, add products, view order, and sales report.

My plan to follow

In terms of the UI, POS payment section, manage staff account section, logout, push notification section needs to be developed. After developing UI, I will start to deal with actual functionalities with JavaScript and PHP. Especially, I need to research to find the way of communication between order by QR code and POS system. Push notification should be researched.

January Monthly Journal

My achievements

I searched for QR code generation service for my project and watched some tutorials for React JS. I worked on the table order page for setting the table order status. I researched some tutorial how to use ajax and deal with CSS to change the colour of the served item by checking check box of item. I've done displaying correct order list on the corresponding table box.

My plan to follow

I should work on the link from order table status page to orderPayment to add extra food items and delete items and show the list of ordered item with product name, its unit price and amount ordered and total cost.

February Monthly Journal

My achievements

This month, I created table order webpages where users will reach when they scan the QR code. On that page, I created three tabs for each category of food, and in each category, I added images of food, name, price, and 'Add to cart button. The page contains the 'Cart' button and the 'Clear Cart' button. When the user clicks the 'add to cart' button the corresponding item is saved to 'Cart'. They can see view their order item by clicking the 'Cart' button. When the 'Clear Cart' button is clicked, all menu items will be removed. I created them using JavaScript.

I created QR codes for 15 tables using a QR code generator and integrated table order page URL. For the URL, to create a domain name, I used Duckdns.

My plan to follow

I need to work on the QRcode display page. Especially need to make buttons work. A user to view a set of QR codes by clicking the 'Generate QRcode' button, and by clicking the 'Download' button, users can download the QR codes. I need to work on button actions, figure out how the download is working and implement it.

14.3 Other materials used

To create a project plan, I used Wrike which is a project management tool.

To draw the Use Case Diagram, draw.io is used.