Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 308-326
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Design and Development of Ferry E-Ticketing System with QR Code based on IoT Technology

Ong Yee Lin[1], Noryusliza Abdullah[1]*

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

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.019
Received 30 August 2022; Accepted 26 September 2022; Available online 30 June 2023

Abstract: The objective of developing an IoT-based ferry e-ticketing system with QR is to digitalize a public transportation company's ticket selling business by reducing the time spent purchasing, and attempting to reduce human errors. This system is critical to improving top-line performance at Bahtera Ferry Services Sdn. Bhd. Using this technology, passengers can obtain real-time ticket information by visiting the company's website rather than contacting customer service. It makes things easier for customers while also saving the company money on making the call and printing all of the tickets on a single sheet of paper. The e-ticket will be generated instantly and emailed to the user after purchase. This system also includes a management system extension that makes it easier to manage all ticket, vessel, and website information. A computerised system, when compared to manual labour, may improve job efficiency by eliminating redundant effort. On the contrary, there are numerous public transportation ticketing solutions available today, but their services are unable to connect with the back-end database server to provide passengers with on-time verification and validation during the boarding session. This project is equipped with Arduino controller and IoT technology, allowing the use of a QR code scanner to retrieve a passenger's e-ticketing boarding details and validation in a split second using the Arduino controller linking with the back-end phpMyAdmin database to verify a passenger's boarding particulars and details. In a nutshell, the Eclipse PHP IDE and phpMyAdmin as the web server and database server were used to create this web-based ferry ticketing system as an intermediate level ticketing and boarding system.

Keywords: E-Ticket, IoT, QR Code, Ferry, Ticket

1. Introduction

Tourism is one of the world's leading players, contributing more than USD 7.6 trillion to the global economy in 2017 (Biddulph & Scheyvens, 2018). A ferry vessel is a passenger vessel that transports passengers, automobiles, and other cargo from one ferry marine terminal to another (K.Talley, 2021). Ferries transport passengers over rivers, straits, and longer sea trips of up to 24 hours. The increasing economy goes hand in hand with the increased use of ferries for passenger transportation

*Corresponding author: yusliza@uthm.edu.my

(Yuzri, 2021). The ticketing process is an essential element of this ferry transportation. This project aims to develop an Internet of Things (IoT) based ferry electronic ticketing system with Quick Response (QR) code for Bahtera Ferry Services Sdn. Bhd. Bahtera Ferry Services Sdn. Bhd. exclusively offers walk-in and telephone services, including selling, booking, and cancelling tickets. All of the workloads in this business are done by hand, such as recording pre-booking tickets, passenger information, and vessel details.

Bahtera Ferry Services Sdn. Bhd. has faced several problems. Firstly, bookings are challenging to do (Zameer & Logeswaran, 2020). This issue arises when the lines are long, which is common during rush hour. According to Miniwatts Marketing Group (2021), automatic check-in is preferred by more than 62% of passengers. Secondly, all work is done manually, which might cause fatalities. For example, there have been numerous reports of passenger ferry overloading problems that occurred in Malaysia. Such an event would result in inadequate stability owing to overloading, resulting in the boat's incapacity to withstand the strong winds and waves to which it was exposed as one of the factors leading to the tragedy (Serap, Jabai, Kamu, Hassan, & Adam, 2017). Thirdly, a larger workforce. The more labor-intensive the process is, the more significant the workforce needs become. Fourthly, According to Intelligent Transport (2021), digitalization may save ticketing expenses by up to 85% as the need for paper tickets, ink, and ticket vending machines is no longer required. Fifthly, fare evasion and ticket forgery in public transport.

Based on the problem statement, an IoT-based ferry ticketing system with QR code is essential to be developed to digitalize the selling of tickets by reducing the time for purchasing and checking tickets and minimising human errors. Three objectives have been set to achieve the aim of this study, which are:

1. To analyse and design an IoT-based ferry e-ticketing system with QR code to increase company efficiency
2. To develop an efficient platform for users to get ferry ticket information in real-time.
3. To test and evaluate the performance of an IoT-based ferry e-ticketing system with QR code.

On the other hand, this system is targeted at people who travel by ferry and for Bahtera Ferry Services Sdn. Bhd. This system includes two types of users: customers, and administrators. Besides, this system includes six modules as shown in Table 1.

Table 1: Modules of the proposed system

No.	Module	Description
1	User management module	Allow users to register, log in, update profile, and log out.
2	Ticket reservation module	Allow customers to query for a ticket, view status (availability) of seat, and book a ticket in real-time.
3	Vessel module	Allow administrators perform CRUD on vessel details and vessel schedule
4	Sales record module	Allow administrators to generate a report that includes the number of sold ticket and the total profit monthly.
5	Payment module	Allow customers to see the quantities, and price of a ferry ticket that has been reserved for future usage. The customer needs to continue with the payment to finish the purchase process.
6	Boarding scanner module	A boarding scanner consists of an Arduino UNO component with two modes: check-in and check-out mode. The LED will emit green light only if the QR code is valid. Else, it will emit red light.

2. Related Work

2.1 IoT Based Ferry e-Ticketing System with QR

The proposed system, named "IoT Based Ferry e-Ticketing System with QR Code," is a web-based PHP system that provides users/passengers with a more convenient and sophisticated All-In-One ferry system. The system is ideal for island travelers who wish to travel by ferry. The user or passenger would simply log in to their account on the ferry e-ticketing website and buy the ferry ticket directly. The e-ticket with the QR code attached will be sent to their mailbox immediately. After that, the user/passenger can proceed to the boarding stage. There is a boarding gate equipped with a QR code sensor at the ferry terminal that allows users/passengers to get on board the ferry without a glitch. The QR code will be detected by the back-end database/administrator system automatically. In the meantime, after matching with the database, the QR code is verified, and the boarding gate equipped with an IoT sensor and light emitting diode (LED) light indicator (Green colors light up) will open up to let the authorized passenger or user go through. On the contrary, the QR code scanned will be rejected and the IoT sensor and the LED light indicator on the boarding gate will not open and light up in red, respectively. This is to ensure that the validation process of the user/passenger is up to date.

2.2 Comparison Between the Existing System and the Proposed System

A comparison table is displayed and compared between the existing system and the proposed system based on their specific features. The chosen existing system are the Langkawi Ferry Line, Keretapi Tanah Melayu Berhad (KTMB), and redBus. The comparison is diagnosed and verified by different aspects as shown in Table 2.

Table 2: Comparison of the existing system with the proposed system

Categories	Feature	Langkawi Ferry Line	KTMB	redBus	Proposed System
Operating System	Android-based	✓	✓	✓	×
	IOS-based	✓	✓	✓	×
	Web-based	✓	✓	✓	✓
User Experience Design (UX)	Clear navigation	×	✓	✓	✓
	Simple interface	✓	✓	×	✓
	Attractive user interface	✓	✓	✓	✓
User Interface Design (UI)	Navigation elements	×	✓	✓	✓
	Concise content	×	✓	×	✓
	Advertisement free	✓	×	✓	✓
	User manual	×	×	×	✓
Functions	Query	✓	✓	✓	✓
	Modify	✓	✓	✓	✓
	Status	✓	✓	✓	✓
	Payment	✓	✓	✓	✓
	Seat Selection	✓	✓	✓	✓
	Cancellation	✓	✓	✓	✓
	Mail	✓	✓	✓	✓
IoT	QR code	×	✓	×	✓
	LED indicator	×	×	×	✓
	E-ticketing system	✓	✓	×	✓
	Validation process	✓	✓	×	✓
	Involve manpower	✓	×	✓	✓

3. Methodology/Framework

A suitable model is chosen to ensure all the progress in completing this project is smooth and completed on time. The chosen model is Prototype Model.

3.1 Prototype Model

The underlying approach and logic are referred to as the methodology. It is necessary to examine and understand the research techniques that are commonly used in the area and the theories or ideas that drive them to choose the most appropriate approach for the aims. This approach offers certain advantages, such as minimising the amount of time and effort required to build the final system since the final method is introduced after all of the needs have been identified, and there is less danger that the final plan would be incorrect (Notepub, 2021). Involving more users in the prototyping process allows them to better understand and connect with the prototype, as well as offer more thorough and full critiques of the prototype. In order to determine which functionalities are missing in the system, this information will be useful. Figure 1 shows the prototyping process modelling.

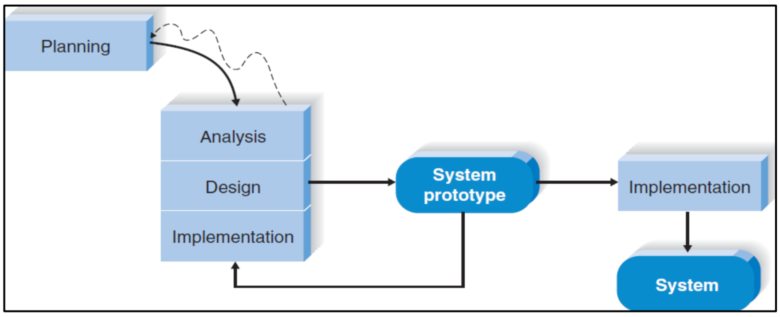


Figure 1: The Prototyping Process Model (Dennis, Wixom, & Roth, 2012)

3.2 System development workflow

The process of system development summarises all the main phases of prototyping, which includes planning, design, development, implementation, and testing. Each phase has its own activities to achieve deliverables. Table 3 illustrates the workflow for the development of the proposed system.

Table 3: Workflow for the development of the proposed system

Phase	Scope of Activity	Outcome
Planning	1. Identify issue statements, objectives, scope of the project, anticipated outcome, and significance of the project. 2. Create a work schedule. 3. Conduct an online search for resources and articles connected to the title. 4. Conduct a thorough examination of the features and operation of existing programmes.	1. Project proposal. 2. Gantt chart. 3. Literature review. 4. Comparison between existing web-based system and proposed system.
Analysis	1. Conduct an interview with the company's staff. 2. Perform a hardware and software requirements analysis. 3. Distinguish between functional and non-functional requirements.	1. Compile a list of user requirements for the interview session. 2. Requirements for hardware and software. 3. Requirements that are both functional and non-functional.
Design	1. Design system architecture diagram. 2. Design CD, and DFD 3. Design a wireframe 4. Design an ERD	1. System architecture diagram production. 2. CD, and DFD production. 3. Wireframe production. 4. ERD production.
Implementation	1. Build a complete module. 2. Complete the system's integration. 3. Establish a connection to the database.	1. Proposed system. 2. Error found and fixed.

	4. A rudimentary system prototype is initially created for testing purposes.	
Testing	1. Conduct system testing 2. Identify if there is any room to upgrade the system	1. Fix and improve the bugs. 2. Fix and ready to release the proposed system.

4. Results and Discussion

The results and discussion section presents data and analysis of the study. It includes system architecture diagrams, CD, DFD, and ERD. Next, the test plan of the system is explored and categorised by module.

4.1 System architecture diagram

An architecture diagram is a kind of system diagram that is used to abstract the general structure of a software system and to build constraints, relationships, and boundaries between components. It provides a full view of the software system's evolution plan's physical deployment (Freeman, 2021). The system architecture design of the proposed system is shown in Figure 2. It consists of user, administrator, user interface (web-based), register, login, user management, ticket reservation, ticket history, payment, ticket receipt with QR code, vessel management, QR code generator, email notification, scan QR code, QR code reader with gate barrier, LED, and database.

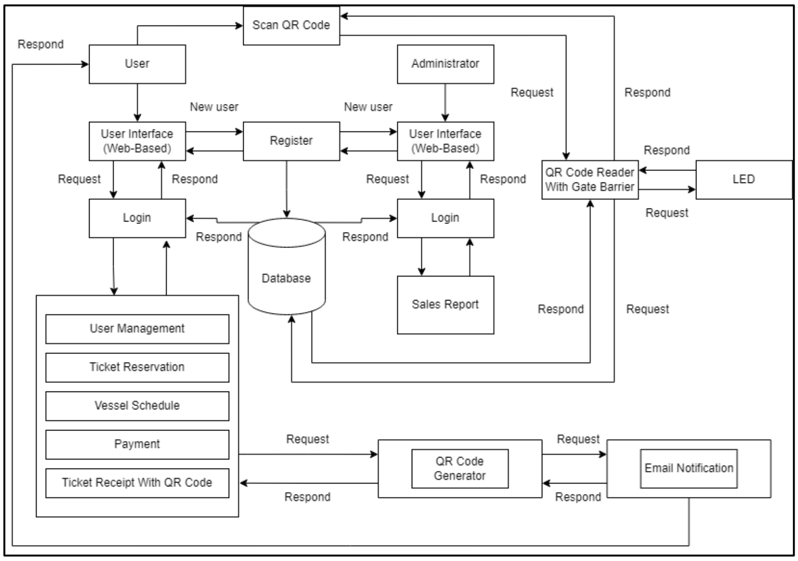


Figure 2: System architecture of the proposed system

4.2 System Design

System design includes CD and DFD. A CD is a diagram that represents the entire system. The purpose of this diagram is to show the expected inputs and outputs from the system. Figure 3 illustrates the CD of the proposed system. While DFD shows the way information flows via a process or system. It is used to explain the flow of data across the proposed system. Figure 4 depicts the DFD level 0 of the proposed system.

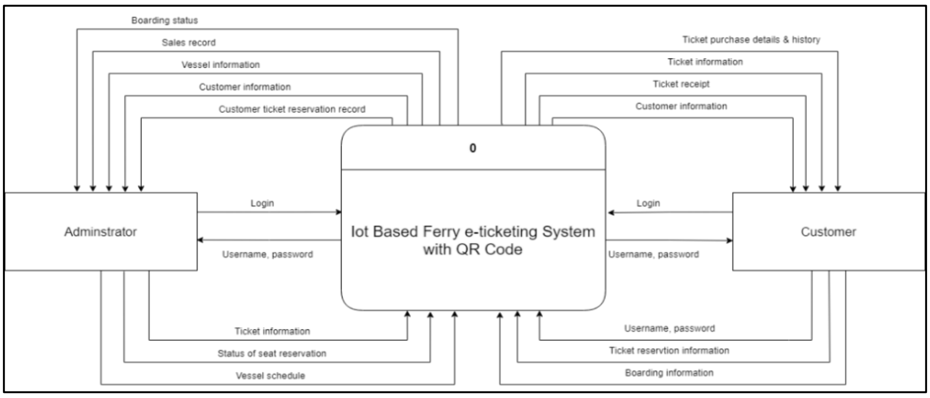


Figure 3: CD of the proposed system

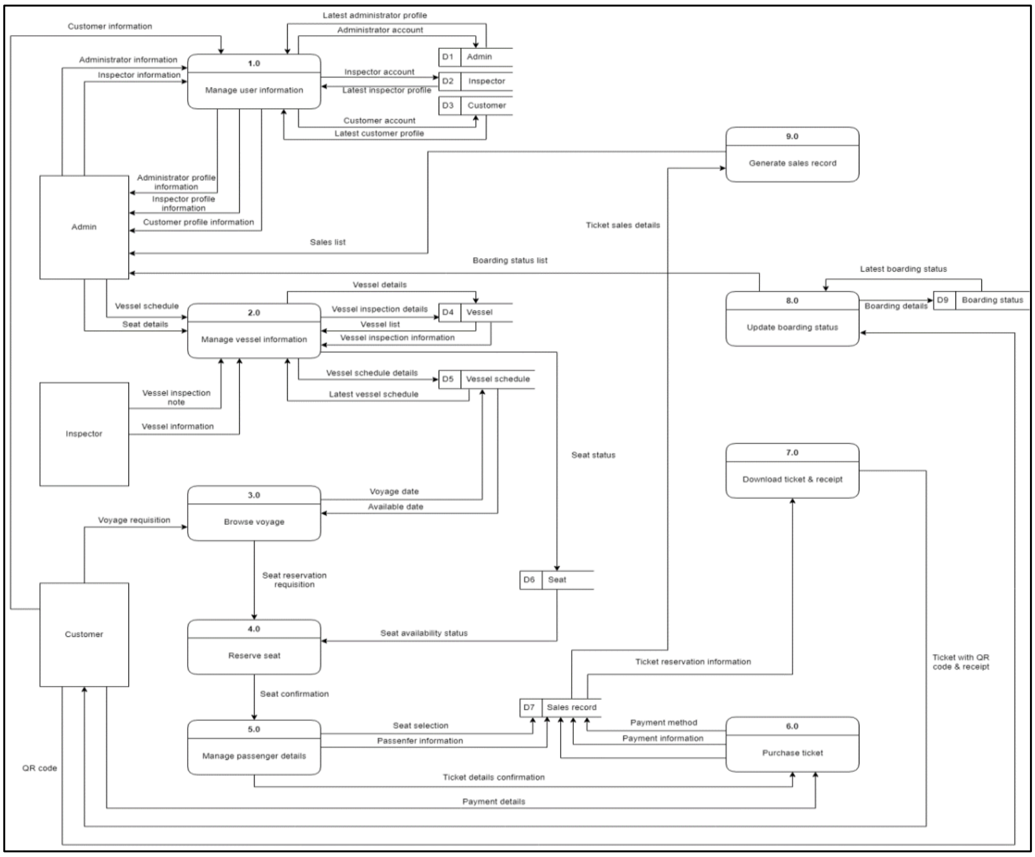


Figure 4: DFD level 0 of the proposed system

4.3 System Requirement Analysis Database design

The system analysis is to have full and detailed knowledge of the system requirement. Requirement analysis is a method in determining user requirements for an application to be developed or updated. It involves analysing, documenting, validating, and handling software or system requirements (NY, 2020). The functional and non-functional requirements of the proposed system are specified and presented in-depth at this stage. Table 4 shows the functional requirements of the proposed system. Table 6 depicts the non-functional requirements of the proposed system.

Table 4: Functional requirements of the proposed system

No.	Module	Description
1	User management module	- A user can register, log in, update profile, and log out. - An administrator can add, delete or update an account for an administrator.
2	Ticket reservation module	- Users can query for a ticket, view status (availability) of access, and book, a ticket in real-time. - At the same time, an administrator can perform CRUD (create, read, update and delete) action on a ticket's status.
3	Vessel module	- Administrators can CRUD vessel’s image, name, model, capacity, year of build, depth, breadth, length overall (LOA), gross register tonnage (GRT), net register tonnage (NRT), and status (active or inactive). - Administrators can arrange the schedule for the vessel. An administrator can add, update, and inactive a vessel schedule by the requirement.
4	Sales record module	- The administrator can generate a report that consist of the number of sold ticket and the total profit monthly.

Table 5: Functional requirements of the proposed system (continue)

No.	Module	Description
5	Payment module	- A user may see the quantities, and price of a ferry ticket that has been reserved for future usage. - When the user confirms the reservation procedure, he or she will continue with the payment to finish the purchase process by cash or PayPal payment gateway.
6	Boarding scanner module	- A boarding scanner consists of an Arduino UNO component with two modes: check-in and check-out mode. - For check-in and check-out mode, it can read and verify data from QR code. It also can update the check-in or check-out time (time when user boarding) at the back-end system. If the QR code is valid, the LED will emit green light, else the red light is emitted.

Table 6: Non-functional requirements of the proposed system

No.	Requirement	Description
1	Performance	- The system should not read QR code for more than one second.
2	Security	- The system should only log the user in if they have the correct login credentials.
3	Usability	- The system should provide a user-friendly interface to the user, easy to use and update the status correctly after the QR scanner reads the QR code.
4	Availability	- The system should be available 24/7, except for scheduled maintenance.
5	Operational	- The system should be available only if there is Internet connection.

4.4 Database design

Database design is a collection of techniques that facilitate the design, construction, execution, and maintenance of organizational data management systems. An ERD depicts the relationships between entity sets stored in a database. In this sense, an entity is an object, a data component. Figure 5 depicts the ERD of the proposed system.

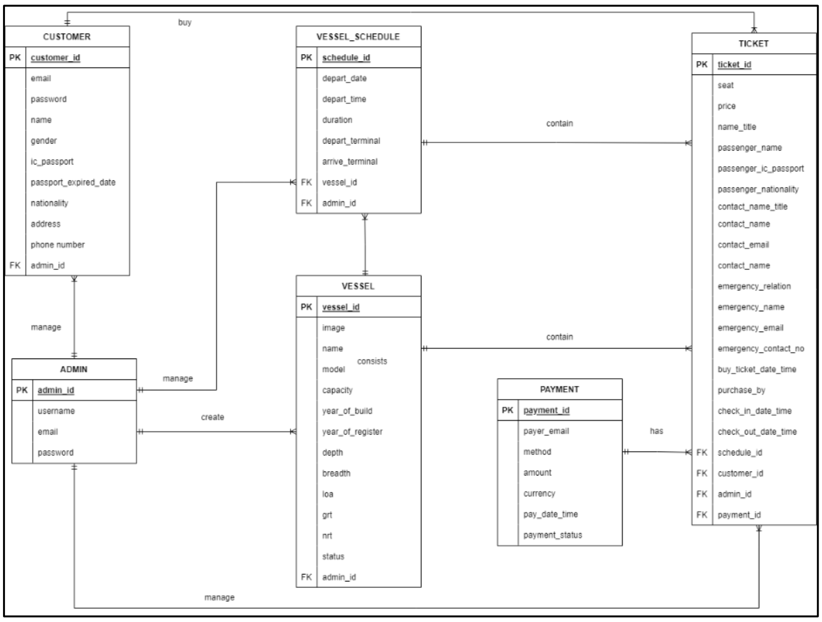


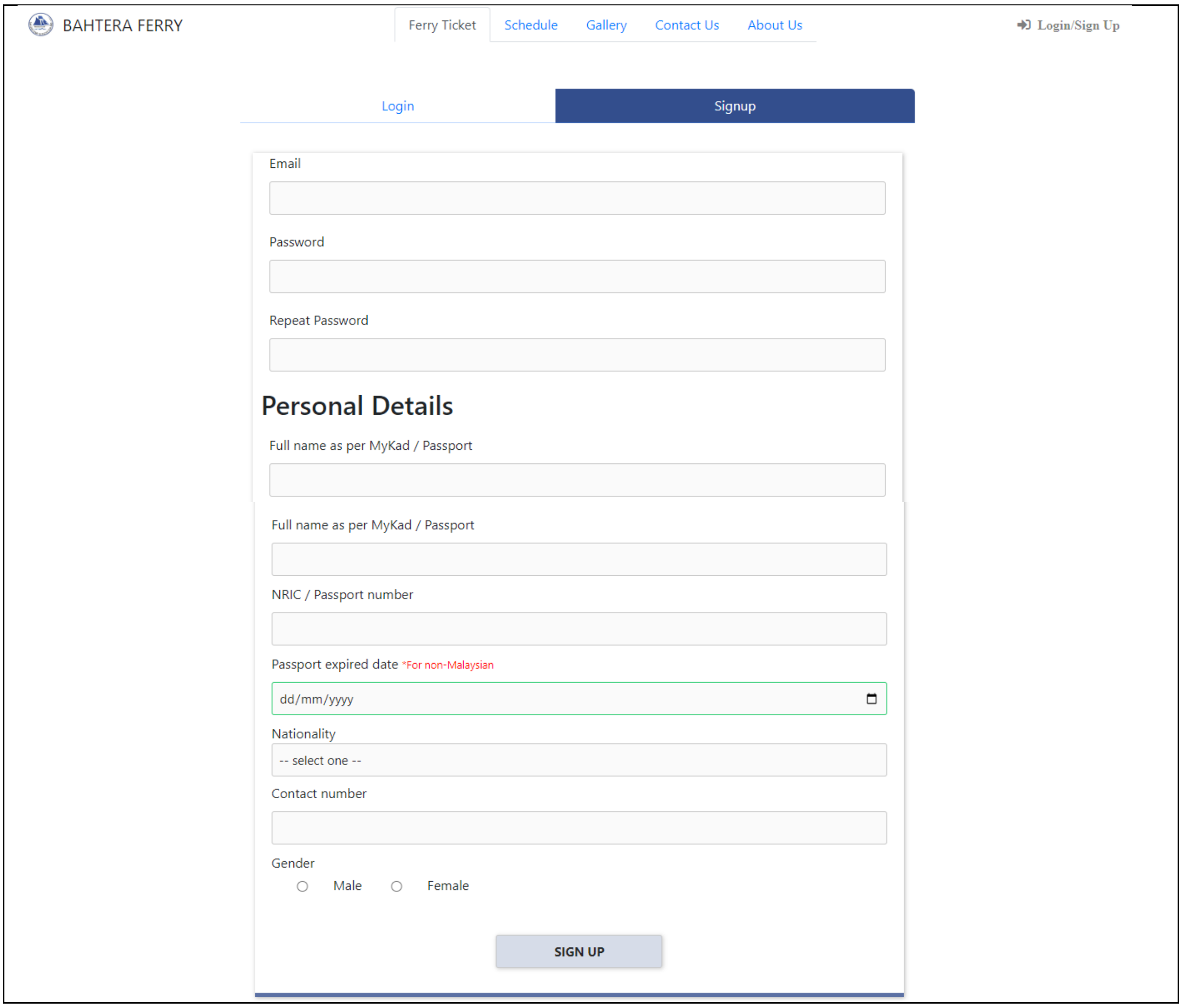
Figure 5: ERD of the proposed system

5. Implementation and Testing

The implementation process guarantees that the web application under development meets the established criteria, while the testing step ensures that the web application is free of errors.

5.1 Implementation

The IoT Based Ferry e-Ticketing System with QR Code is a web-based system constructed with PHP, HTML, CSS, and JavaScript programming languages. The proposed system, which includes fully defined security and verification methods, is configured using an Arduino Uno, QR Code scanner and MySQL database. The database and web server will be hosted on the phpMyAdmin database server. When passengers on-board the ferry for their journey, an LED indicator will be used in conjunction with the IoT sensor to begin the validation process. Figure 6 shows the customer registration user interface while Figure 7 shows the code segment for the customer registration process. Furthermore, Figure 10 depicts the administrator login user interface while Figure 11 depicts the code segment for administrator login process.



Email

Password

Repeat Password

Personal Details

Full name as per MyKad / Passport

Full name as per MyKad / Passport

NRIC / Passport number

Passport expired date *For non-Malaysian

dd/mm/yyyy

Nationality

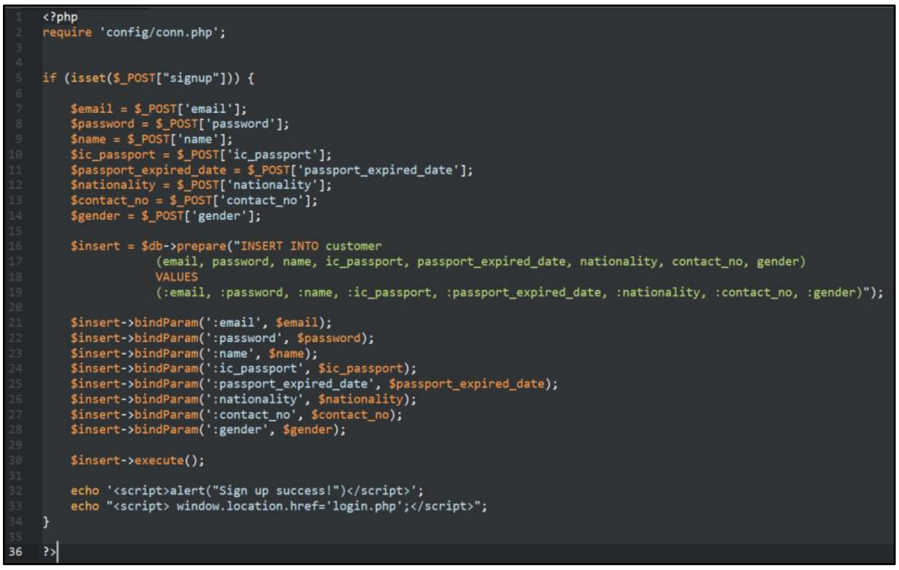
-- select one --

Contact number

Gender

○ Male ○ Female

Figure 6: Customer Registration Interface



```php
<?php
require 'config/conn.php';


if (isset($_POST["signup"])) {

    $email = $_POST['email'];
    $password = $_POST['password'];
    $name = $_POST['name'];
    $ic_passport = $_POST['ic_passport'];
    $passport_expired_date = $_POST['passport_expired_date'];
    $nationality = $_POST['nationality'];
    $contact_no = $_POST['contact_no'];
    $gender = $_POST['gender'];

    $insert = $db->prepare("INSERT INTO customer
                (email, password, name, ic_passport, passport_expired_date, nationality, contact_no, gender)
                VALUES
                (:email, :password, :name, :ic_passport, :passport_expired_date, :nationality, :contact_no, :gender)");

    $insert->bindParam(':email', $email);
    $insert->bindParam(':password', $password);
    $insert->bindParam(':name', $name);
    $insert->bindParam(':ic_passport', $ic_passport);
    $insert->bindParam(':passport_expired_date', $passport_expired_date);
    $insert->bindParam(':nationality', $nationality);
    $insert->bindParam(':contact_no', $contact_no);
    $insert->bindParam(':gender', $gender);

    $insert->execute();

    echo '<script>alert("Sign up success!")</script>';
    echo "<script> window.location.href='login.php';</script>";
}

?>
```

Figure 7: Code Segment for Customer Registration Process

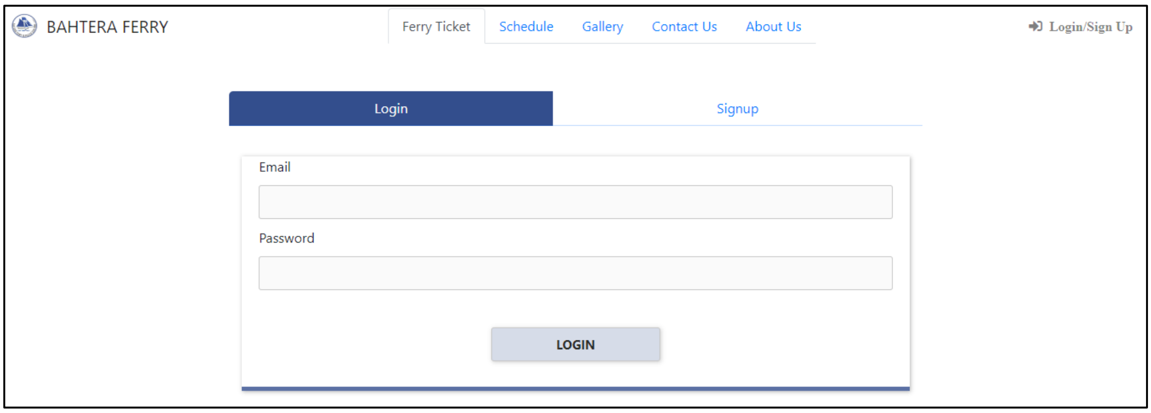
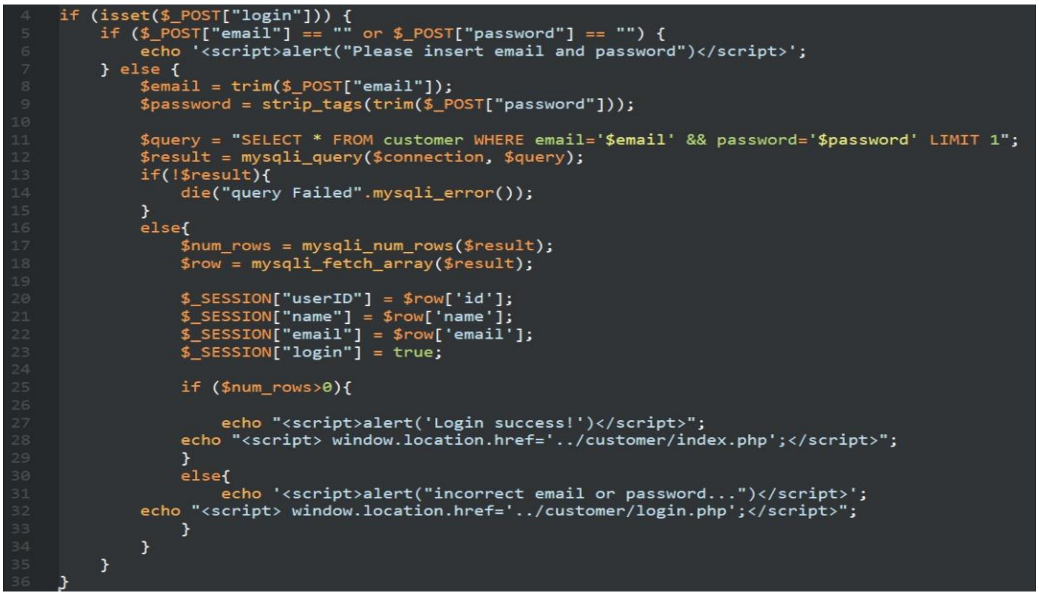


Figure 8: Customer Login



```
if (isset($_POST["login"])) {
    if ($_POST["email"] == "" or $_POST["password"] == "") {
        echo '<script>alert("Please insert email and password")</script>';
    } else {
        $email = trim($_POST["email"]);
        $password = strip_tags(trim($_POST["password"]));

        $query = "SELECT * FROM customer WHERE email='$email' && password='$password' LIMIT 1";
        $result = mysqli_query($connection, $query);
        if(!$result){
            die("query Failed".mysqli_error());
        }
        else{
            $num_rows = mysqli_num_rows($result);
            $row = mysqli_fetch_array($result);

            $_SESSION["userID"] = $row['id'];
            $_SESSION["name"] = $row['name'];
            $_SESSION["email"] = $row['email'];
            $_SESSION["login"] = true;

            if ($num_rows>0){

                echo "<script>alert('Login success!')</script>";
            echo "<script> window.location.href='../customer/index.php';</script>";
            }
            else{
                echo '<script>alert("incorrect email or password...")</script>';
        echo "<script> window.location.href='../customer/login.php';</script>";
            }
        }
    }
}
```

Figure 9: Code Segment for Customer Login Process

In addition, the login user interface for the web system application is shown in Figure 10. The administrator will use their email and password assigned by Super Admin 1 to log into the web system. Figure 11 depicts the code section for the administrator login process.

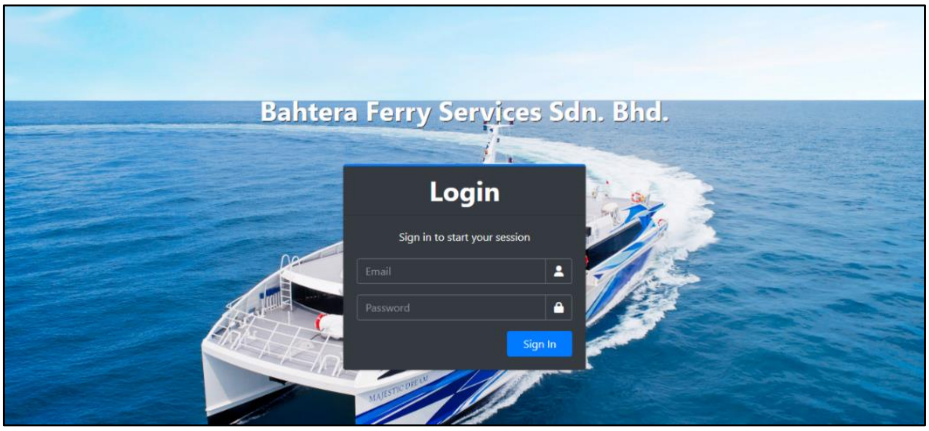
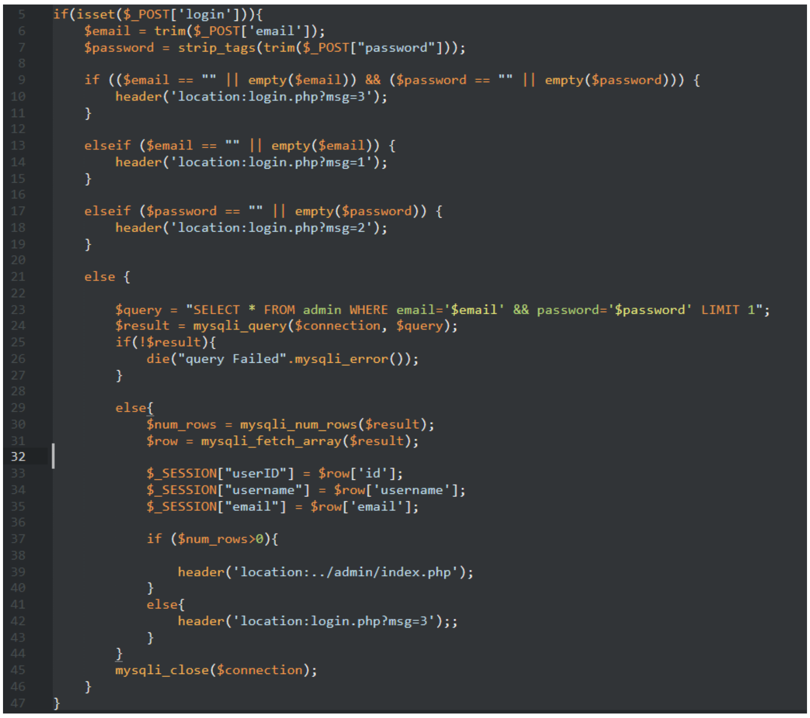

Bahtera Ferry Services Sdn. Bhd.
Login
Sign in to start your session
Email
Password
Sign In

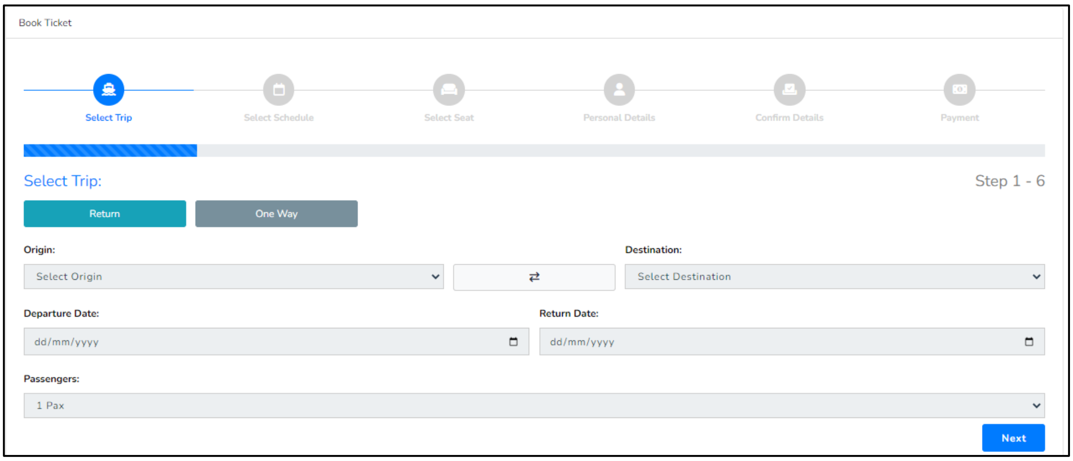
Figure 10: Administrator Login User Interface

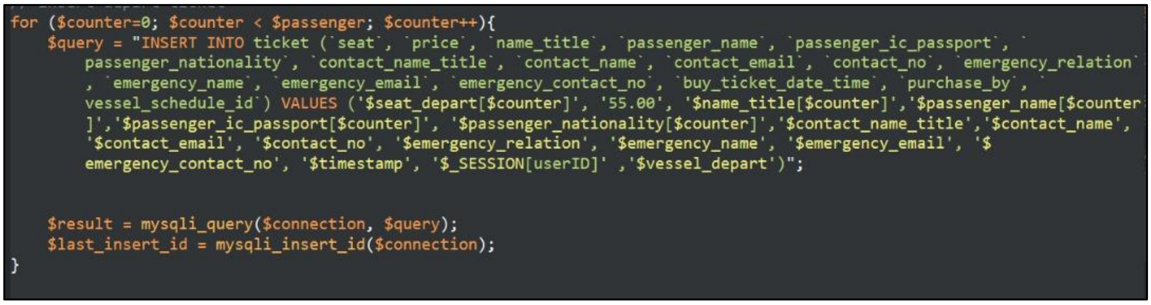


```
if(isset($_POST['login'])){
    $email = trim($_POST['email']);
    $password = strip_tags(trim($_POST["password"]));

    if (($email == "" || empty($email)) && ($password == "" || empty($password))) {
        header('location:login.php?msg=3');
    }

    elseif ($email == "" || empty($email)) {
        header('location:login.php?msg=1');
    }

    elseif ($password == "" || empty($password)) {
        header('location:login.php?msg=2');
    }

    else {

        $query = "SELECT * FROM admin WHERE email='$email' && password='$password' LIMIT 1";
        $result = mysqli_query($connection, $query);
        if(!$result){
            die("query Failed".mysqli_error());
        }

        else{
            $num_rows = mysqli_num_rows($result);
            $row = mysqli_fetch_array($result);

            $_SESSION["userID"] = $row['id'];
            $_SESSION["username"] = $row['username'];
            $_SESSION["email"] = $row['email'];

            if ($num_rows>0){

                header('location:../admin/index.php');
            }
            else{
                header('location:login.php?msg=3');;
            }
        }
        mysqli_close($connection);
    }
}
```

Figure 11: Code segment for administrator login process

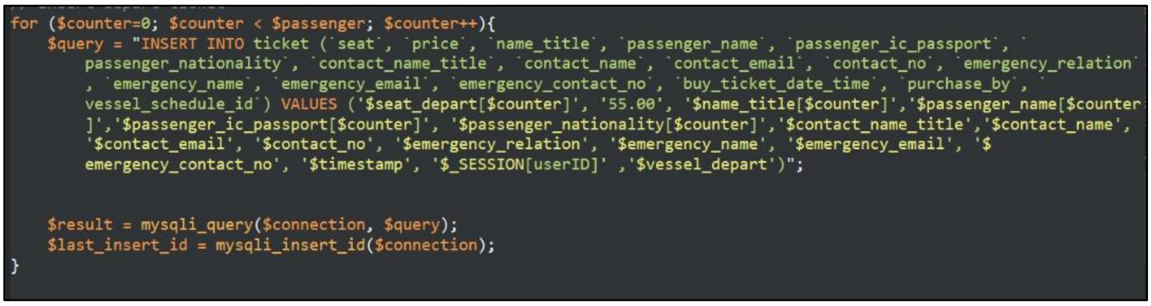
Ticket reservation module is for users (administrator or customer) to query for a ticket, view status (availability) of access, book, or cancel a ticket in real-time. At the same time, an administrator can perform CRUD (create, read, update and delete) action on a ticket's status. Figure 12 shows the ticket reservation module while Figure 13 illustrates the code segment for user query tickets.





```
for ($counter=0; $counter < $passenger; $counter++){
    $query = "INSERT INTO ticket (`seat`, `price`, `name_title`, `passenger_name`, `passenger_ic_passport`, `
        passenger_nationality`, `contact_name_title`, `contact_name`, `contact_email`, `contact_no`, `emergency_relation`
        , `emergency_name`, `emergency_email`, `emergency_contact_no`, `buy_ticket_date_time`, `purchase_by`, `
        vessel_schedule_id`) VALUES ('$seat_depart[$counter]', '55.00', '$name_title[$counter]','$passenger_name[$counter
        ]','$passenger_ic_passport[$counter]', '$passenger_nationality[$counter]','$contact_name_title','$contact_name',
        '$contact_email', '$contact_no', '$emergency_relation', '$emergency_name', '$emergency_email', '$
        emergency_contact_no', '$timestamp', '$_SESSION[userID]' ,'$vessel_depart')";


    $result = mysqli_query($connection, $query);
    $last_insert_id = mysqli_insert_id($connection);
}
```

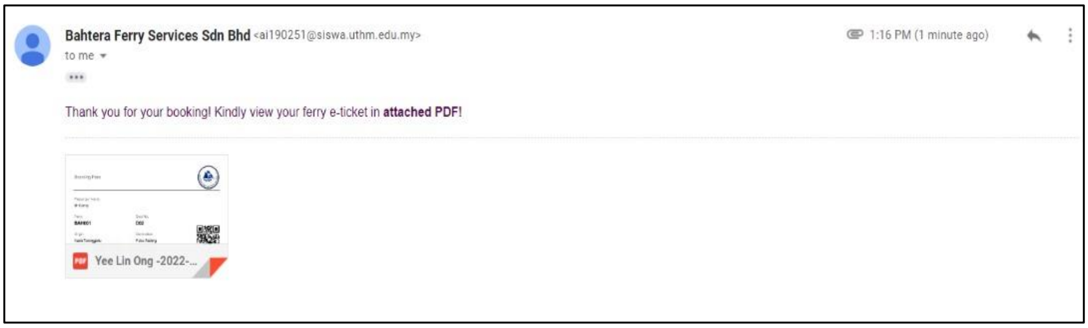
Figure 12: Ticket reservation module



```
for ($counter=0; $counter < $passenger; $counter++){
    $query = "INSERT INTO ticket (`seat`, `price`, `name_title`, `passenger_name`, `passenger_ic_passport`, `
        passenger_nationality`, `contact_name_title`, `contact_name`, `contact_email`, `contact_no`, `emergency_relation`
        , `emergency_name`, `emergency_email`, `emergency_contact_no`, `buy_ticket_date_time`, `purchase_by`, `
        vessel_schedule_id`) VALUES ('$seat_depart[$counter]', '55.00', '$name_title[$counter]','$passenger_name[$counter
        ]','$passenger_ic_passport[$counter]', '$passenger_nationality[$counter]','$contact_name_title','$contact_name',
        '$contact_email', '$contact_no', '$emergency_relation', '$emergency_name', '$emergency_email', '$
        emergency_contact_no', '$timestamp', '$_SESSION[userID]' ,'$vessel_depart')";


    $result = mysqli_query($connection, $query);
    $last_insert_id = mysqli_insert_id($connection);
}
```

Figure 13: Code segment for ticket reservation query

According to Figure 14, customers will receive e-ticket with QR Code via email filled in the ticket reservation module. Figure 15 depicts the E-ticket PDF with passenger's details, seat number, origin/destination, and boarding time. Figure 16 illustrates the code segment for generating pdf and e-ticket with QR Code.



Bahtera Ferry Services Sdn Bhd <ai190251@siswa.uthm.edu.my>

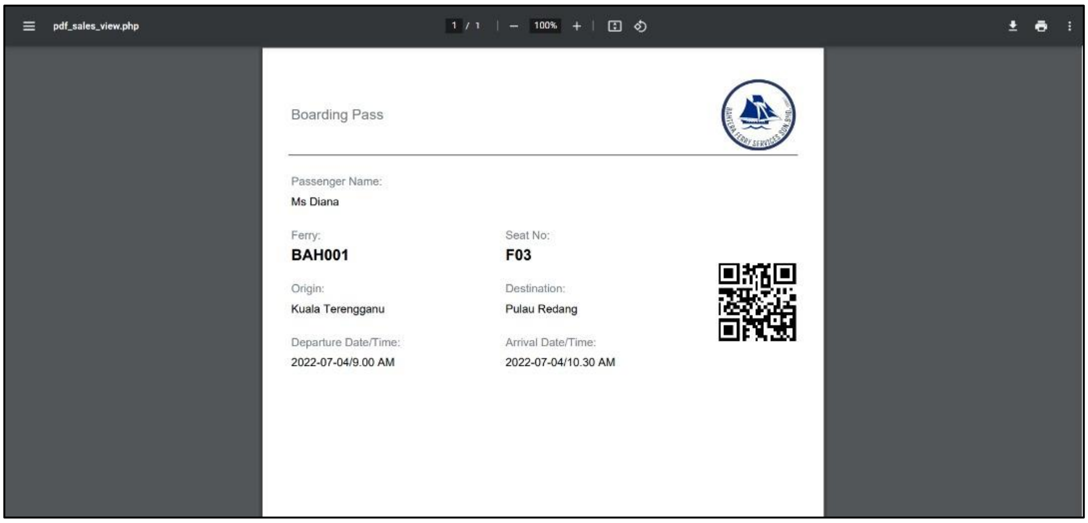
to me

1:16 PM (1 minute ago)

Thank you for your booking! Kindly view your ferry e-ticket in **attached PDF!**

Yee Lin Ong -2022-...

Figure 14: E-ticket with QR code via email



Boarding Pass

Passenger Name:
Ms Diana

Ferry:
BAH001

Seat No:
F03

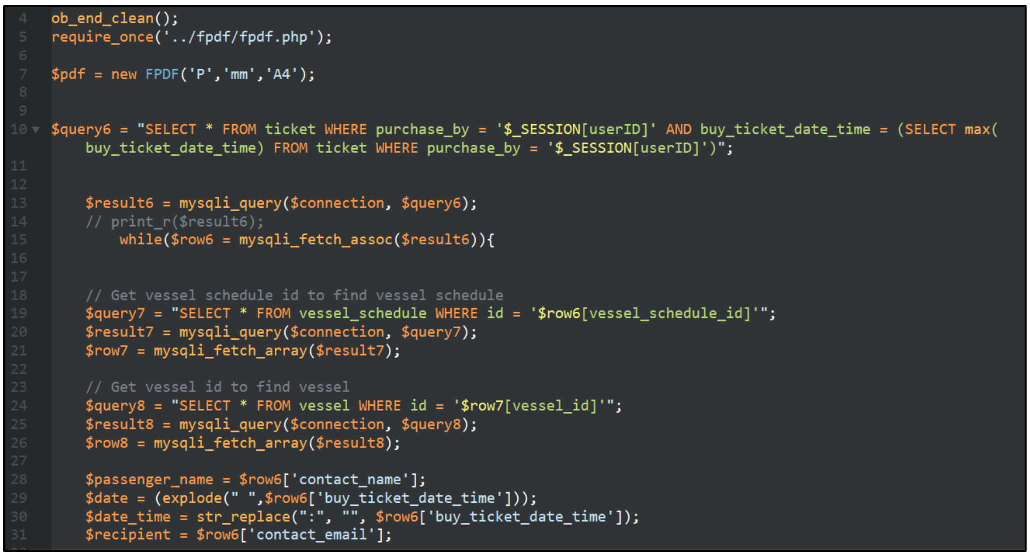
Origin:
Kuala Terengganu

Destination:
Pulau Redang

Departure Date/Time:
2022-07-04/9.00 AM

Arrival Date/Time:
2022-07-04/10.30 AM

Figure 15: E-ticket PDF



```php
ob_end_clean();
require_once('../fpdf/fpdf.php');

$pdf = new FPDF('P','mm','A4');


$query6 = "SELECT * FROM ticket WHERE purchase_by = '$_SESSION[userID]' AND buy_ticket_date_time = (SELECT max(
    buy_ticket_date_time) FROM ticket WHERE purchase_by = '$_SESSION[userID]')";


    $result6 = mysqli_query($connection, $query6);
    // print_r($result6);
        while($row6 = mysqli_fetch_assoc($result6)){


    // Get vessel schedule id to find vessel schedule
    $query7 = "SELECT * FROM vessel_schedule WHERE id = '$row6[vessel_schedule_id]'";
    $result7 = mysqli_query($connection, $query7);
    $row7 = mysqli_fetch_array($result7);

    // Get vessel id to find vessel
    $query8 = "SELECT * FROM vessel WHERE id = '$row7[vessel_id]'";
    $result8 = mysqli_query($connection, $query8);
    $row8 = mysqli_fetch_array($result8);

    $passenger_name = $row6['contact_name'];
    $date = (explode(" ",$row6['buy_ticket_date_time']));
    $date_time = str_replace(":", "", $row6['buy_ticket_date_time']);
    $recipient = $row6['contact_email'];
```

Figure 16: Code segment for generate pdf and e-ticket with QR Code

Figure 17 shows the dashboard page for administrator. Figure 19 illustrates the user management module, and Figure 18 portrays the vessel module.

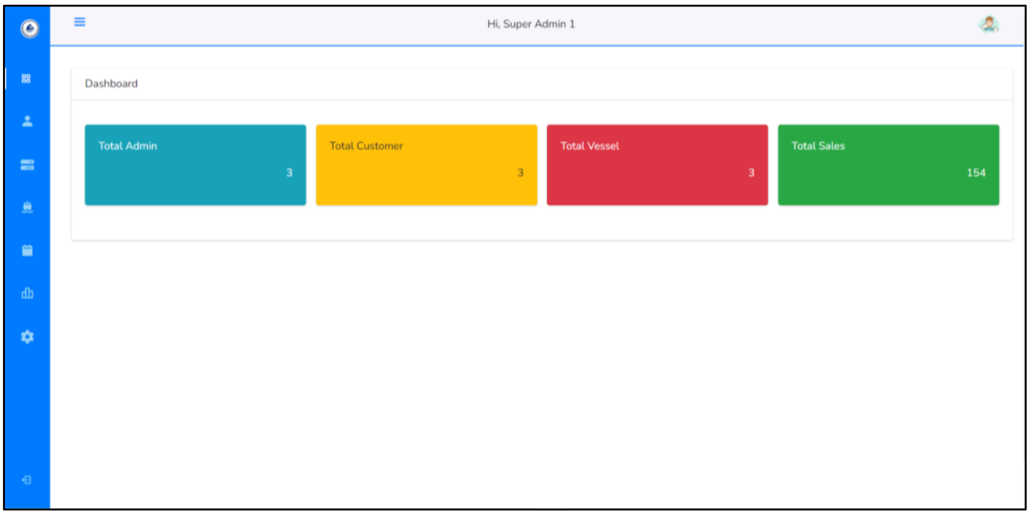
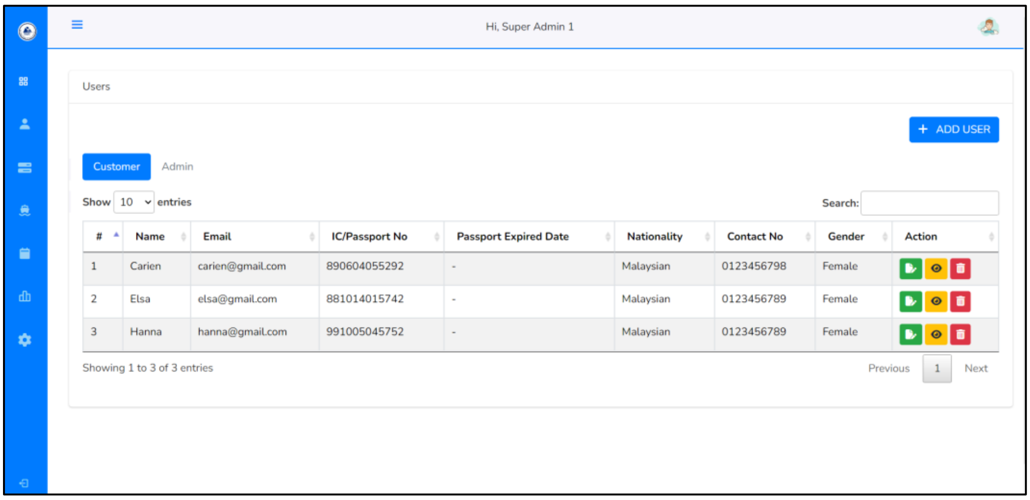


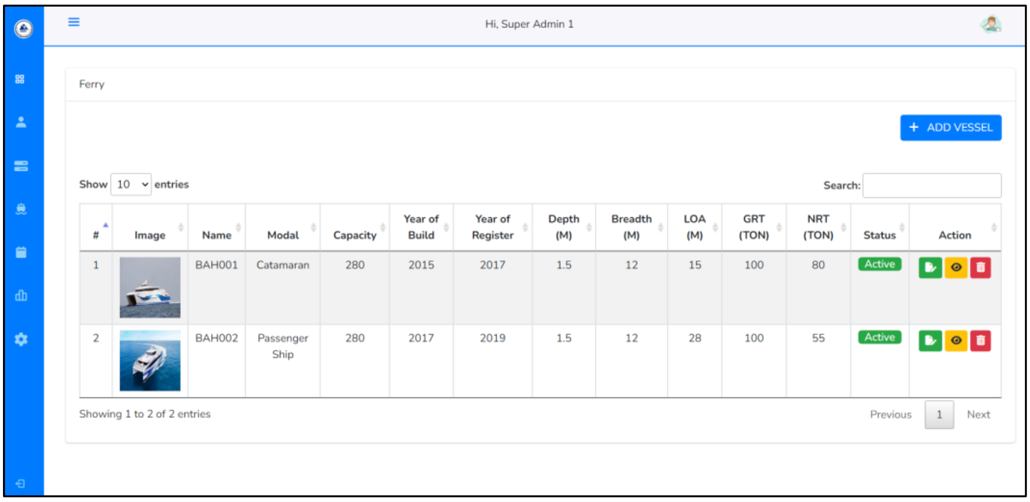
Figure 17: Admin dashboard



#	Name	Email	IC/Passport No	Passport Expired Date	Nationality	Contact No	Gender	Action
1	Carien	carien@gmail.com	890604055292	-	Malaysian	0123456798	Female	
2	Elsa	elsa@gmail.com	881014015742	-	Malaysian	0123456789	Female	
3	Hanna	hanna@gmail.com	991005045752	-	Malaysian	0123456789	Female	

Showing 1 to 3 of 3 entries

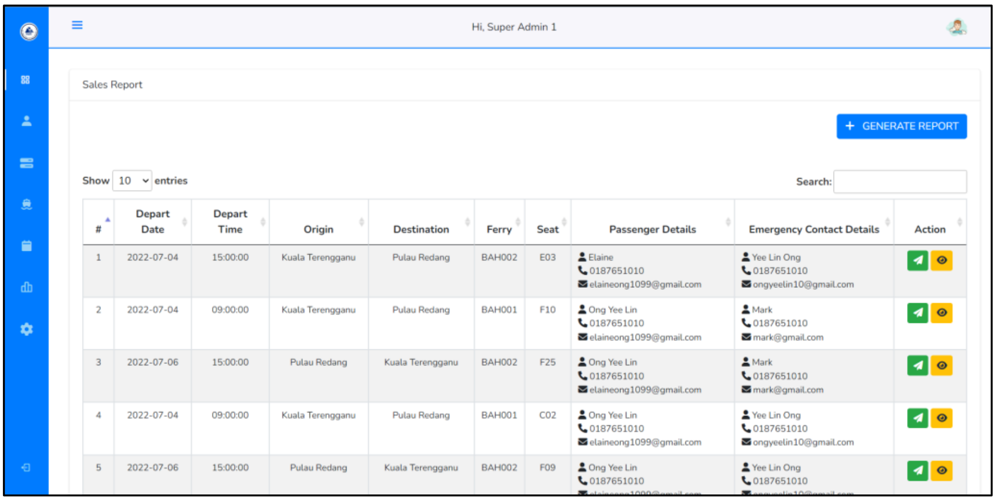
Figure 18: User management module



#	Image	Name	Modal	Capacity	Year of Build	Year of Register	Depth (M)	Breadth (M)	LOA (M)	GRT (TON)	NRT (TON)	Status	Action
1		BAH001	Catamaran	280	2015	2017	1.5	12	15	100	80	Active	
2		BAH002	Passenger Ship	280	2017	2019	1.5	12	28	100	55	Active	

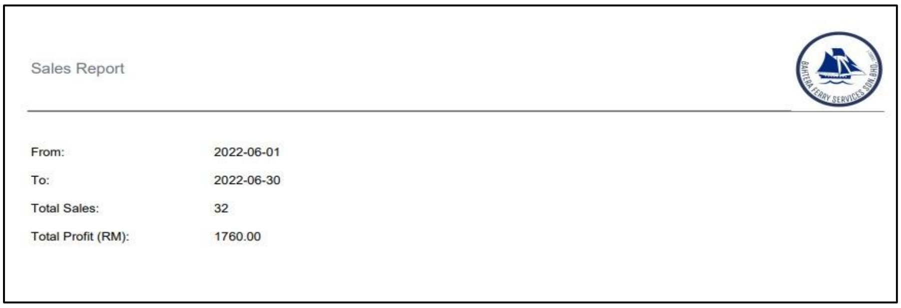
Figure 19: Vessel Module

The administrator can generate a report that shows the total number of passengers on a daily or monthly basis. The report's data will include the total number of sales and the total profit. Meanwhile, the price of the ferry ticket is determined by the seat booked. As a result, a sales report on the total number of sold tickets will be generated with details in PDF. Figure 20 shows the sales record module, Figure 22 shows the generated sales report sample, while Figure 23 shows the code segment for sales record generated pdf session.



#	Depart Date	Depart Time	Origin	Destination	Ferry	Seat	Passenger Details	Emergency Contact Details	Action
1	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	E03	Elaine 0187651010 elaineong1099@gmail.com	Yee Lin Ong 0187651010 ongyeelin10@gmail.com	
2	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	F10	Ong Yee Lin 0187651010 elaineong1099@gmail.com	Mark 0187651010 mark@gmail.com	
3	2022-07-06	15:00:00	Pulau Redang	Kuala Terengganu	BAH002	F25	Ong Yee Lin 0187651010 elaineong1099@gmail.com	Mark 0187651010 mark@gmail.com	
4	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	C02	Ong Yee Lin 0187651010 elaineong1099@gmail.com	Yee Lin Ong 0187651010 ongyeelin10@gmail.com	
5	2022-07-06	15:00:00	Pulau Redang	Kuala Terengganu	BAH002	F09	Ong Yee Lin 0187651010	Yee Lin Ong 0187651010	

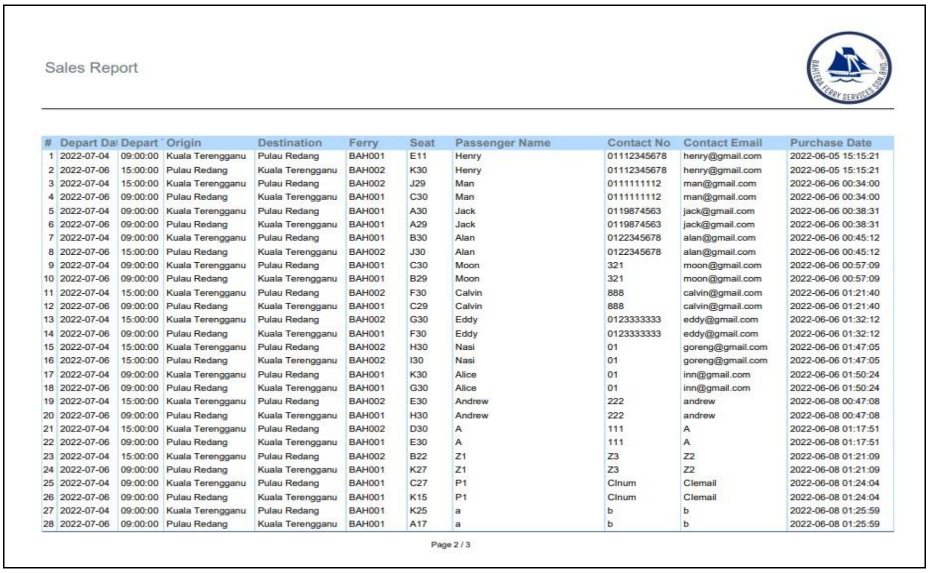
Figure 20: Sales record module



Sales Report

From:	2022-06-01
To:	2022-06-30
Total Sales:	32
Total Profit (RM):	1760.00

Figure 21: Generated sales record pdf sample – page 1

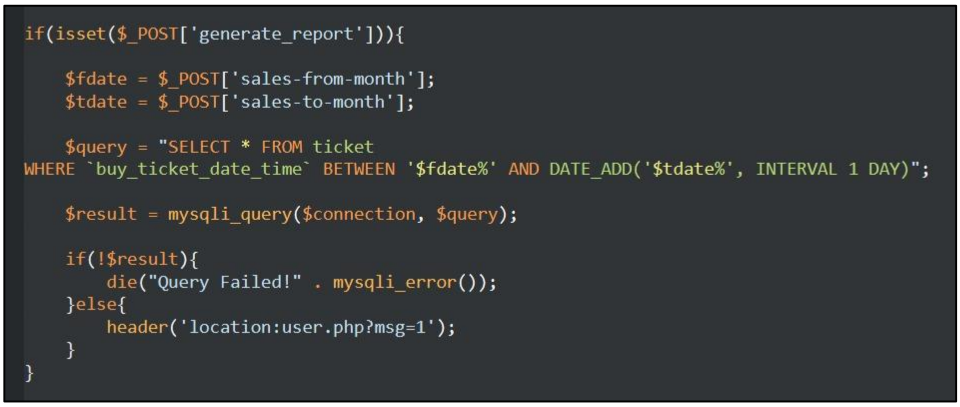


Sales Report

#	Depart Da	Depart	Origin	Destination	Ferry	Seat	Passenger Name	Contact No	Contact Email	Purchase Date
1	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	E11	Henry	01112345678	henry@gmail.com	2022-06-05 15:15:21
2	2022-07-06	15:00:00	Pulau Redang	Kuala Terengganu	BAH002	K30	Henry	01112345678	henry@gmail.com	2022-06-05 15:15:21
3	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	J29	Man	0111111112	man@gmail.com	2022-06-06 00:34:00
4	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	C30	Man	0111111112	man@gmail.com	2022-06-06 00:34:00
5	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	A30	Jack	0119874563	jack@gmail.com	2022-06-06 00:38:31
6	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	A29	Jack	0119874563	jack@gmail.com	2022-06-06 00:38:31
7	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	B30	Alan	0122345678	alan@gmail.com	2022-06-06 00:45:12
8	2022-07-06	15:00:00	Pulau Redang	Kuala Terengganu	BAH002	J30	Alan	0122345678	alan@gmail.com	2022-06-06 00:45:12
9	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	C30	Moon	321	moon@gmail.com	2022-06-06 00:57:09
10	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	B29	Moon	321	moon@gmail.com	2022-06-06 00:57:09
11	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	F30	Calvin	888	calvin@gmail.com	2022-06-06 01:21:40
12	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	C29	Calvin	888	calvin@gmail.com	2022-06-06 01:21:40
13	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	G30	Eddy	0123333333	eddy@gmail.com	2022-06-06 01:32:12
14	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	F30	Eddy	0123333333	eddy@gmail.com	2022-06-06 01:32:12
15	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	H30	Nasi	01	goreng@gmail.com	2022-06-06 01:47:05
16	2022-07-06	15:00:00	Pulau Redang	Kuala Terengganu	BAH002	I30	Nasi	01	goreng@gmail.com	2022-06-06 01:47:05
17	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	K30	Alice	01	inn@gmail.com	2022-06-06 01:50:24
18	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	G30	Alice	01	inn@gmail.com	2022-06-06 01:50:24
19	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	E30	Andrew	222	andrew	2022-06-08 00:47:08
20	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	H30	Andrew	222	andrew	2022-06-08 00:47:08
21	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	D30	A	111	A	2022-06-08 01:17:51
22	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	E30	A	111	A	2022-06-08 01:17:51
23	2022-07-04	15:00:00	Kuala Terengganu	Pulau Redang	BAH002	B22	Z1	Z3	Z2	2022-06-08 01:21:09
24	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	K27	Z1	Z3	Z2	2022-06-08 01:21:09
25	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	C27	P1	Clnum	Clemail	2022-06-08 01:24:04
26	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	K15	P1	Clnum	Clemail	2022-06-08 01:24:04
27	2022-07-04	09:00:00	Kuala Terengganu	Pulau Redang	BAH001	K25	a	b	b	2022-06-08 01:25:59
28	2022-07-06	09:00:00	Pulau Redang	Kuala Terengganu	BAH001	A17	a	b	b	2022-06-08 01:25:59

Page 2 / 3

Figure 22: Generated sales record pdf sample – page 2 (continue)



```
if(isset($_POST['generate_report'])){

    $fdate = $_POST['sales-from-month'];
    $tdate = $_POST['sales-to-month'];

    $query = "SELECT * FROM ticket
WHERE `buy_ticket_date_time` BETWEEN '$fdate%' AND DATE_ADD('$tdate%', INTERVAL 1 DAY)";

    $result = mysqli_query($connection, $query);

    if(!$result){
        die("Query Failed!" . mysqli_error());
    }else{
        header('location:user.php?msg=1');
    }
}
```

Figure 23: Code segment for sales record generate pdf session

In the payment module, the user can pay with PayPal or Cash by clicking the button as illustrated in Figure 24. After the payment procedure for ferry tickets, an enclosed pdf consisting of passenger's details, ticket information, seat number, and boarding time as well as origin/destination will be sent to the payer's email. The code section to integrate the PayPal payment gateway is depicted in Figure 25.

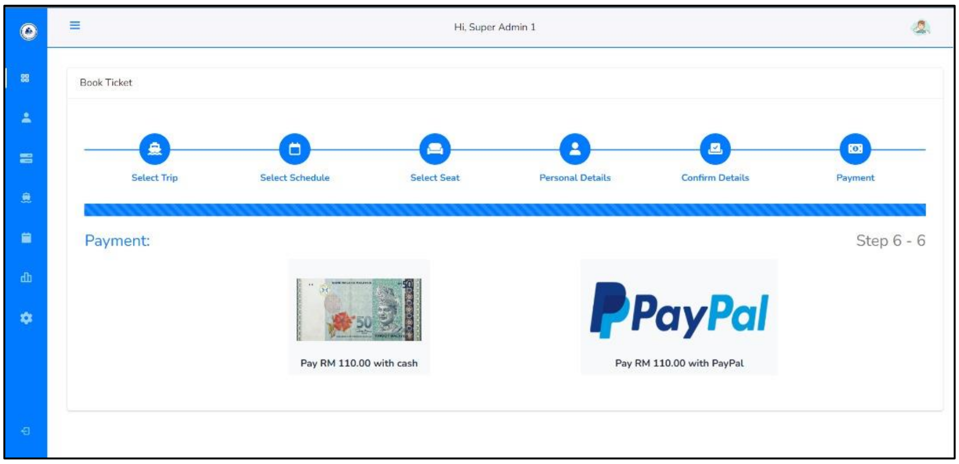
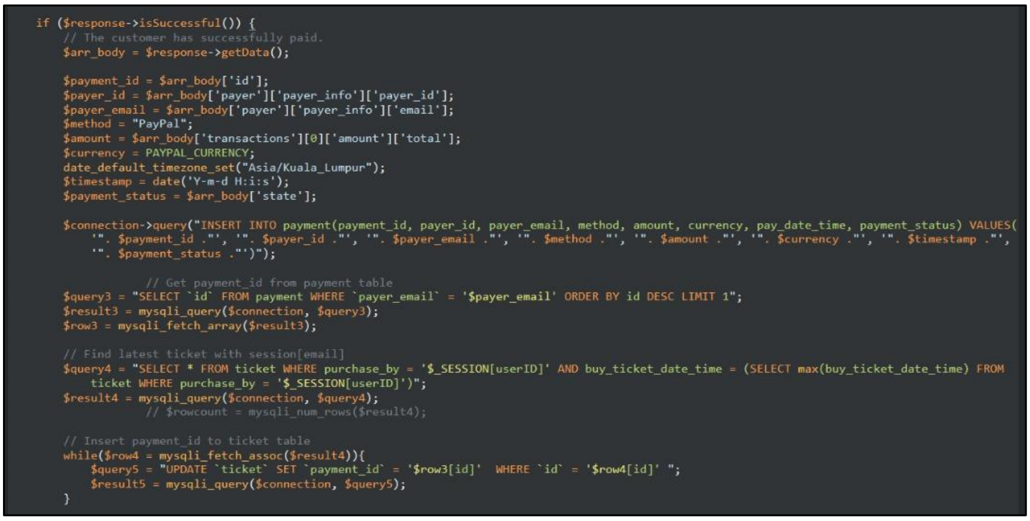


Figure 24: PayPal or Cash Payment



```
if ($response->isSuccessful()) {
    // The customer has successfully paid.
    $arr_body = $response->getData();

    $payment_id = $arr_body['id'];
    $payer_id = $arr_body['payer']['payer_info']['payer_id'];
    $payer_email = $arr_body['payer']['payer_info']['email'];
    $method = "PayPal";
    $amount = $arr_body['transactions'][0]['amount']['total'];
    $currency = PAYPAL_CURRENCY;
    date_default_timezone_set("Asia/Kuala_Lumpur");
    $timestamp = date('Y-m-d H:i:s');
    $payment_status = $arr_body['state'];

    $connection->query("INSERT INTO payment(payment_id, payer_id, payer_email, method, amount, currency, pay_date_time, payment_status) VALUES(
        '". $payment_id ."', '". $payer_id ."', '". $payer_email ."', '". $method ."', '". $amount ."', '". $currency ."', '". $timestamp ."',
        '". $payment_status ."')");

                // Get payment_id from payment table
    $query3 = "SELECT `id` FROM payment WHERE `payer_email` = '$payer_email' ORDER BY id DESC LIMIT 1";
    $result3 = mysqli_query($connection, $query3);
    $row3 = mysqli_fetch_array($result3);

    // Find latest ticket with session[email]
    $query4 = "SELECT * FROM ticket WHERE purchase_by = '$_SESSION[userID]' AND buy_ticket_date_time = (SELECT max(buy_ticket_date_time) FROM
        ticket WHERE purchase_by = '$_SESSION[userID]')";
    $result4 = mysqli_query($connection, $query4);
                // $rowcount = mysqli_num_rows($result4);

    // Insert payment_id to ticket table
    while($row4 = mysqli_fetch_assoc($result4)){
        $query5 = "UPDATE `ticket` SET `payment_id` = '$row3[id]'  WHERE `id` = '$row4[id]' ";
        $result5 = mysqli_query($connection, $query5);
    }
```

Figure 25: Code section to integrate the PayPal payment gateway

A boarding scanner is made out of an Arduino UNO component that has two modes: check-in and check-out. In check-in mode, it can read, validate data from a QR code, and update the check-in time (the moment when the user boards) at the back-end system. The LED will emit green light if the QR code is legitimate. Otherwise, the LED will emit red light. In check-out mode, the machine can read and validate data from QR codes while also updating the check-out time (the time when the user arrives) at the back-end system. If the QR code is not legitimate, the LED will emit red light, else green. Figure 26 illustrates Arduino LED date and QR code boarding scanner while Figure 27 shows Arduino LED gate controlled code.

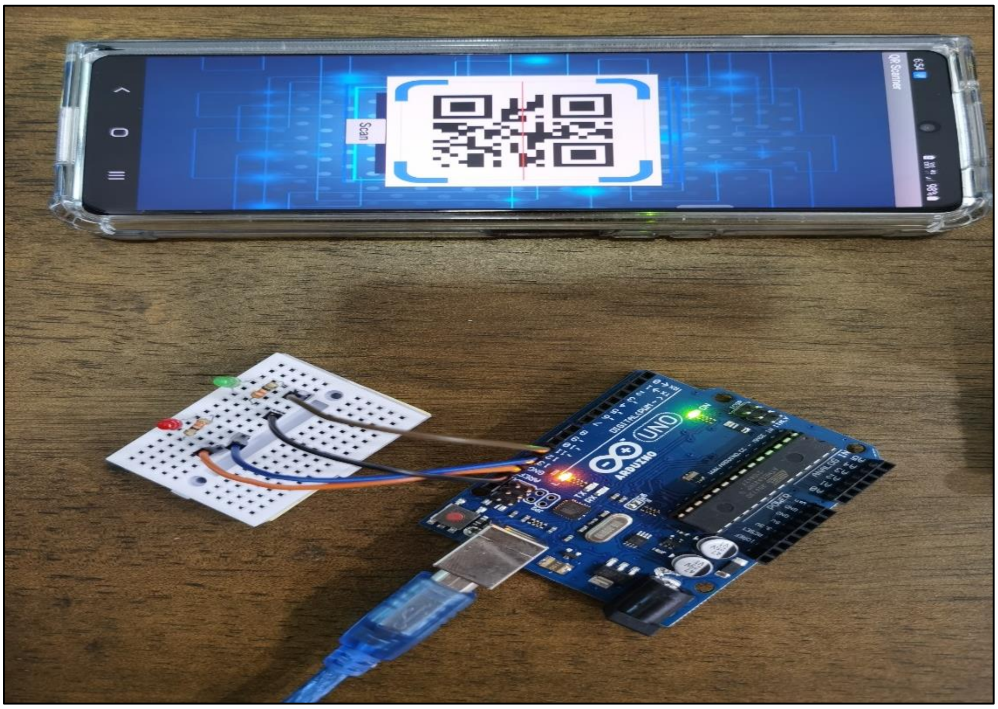
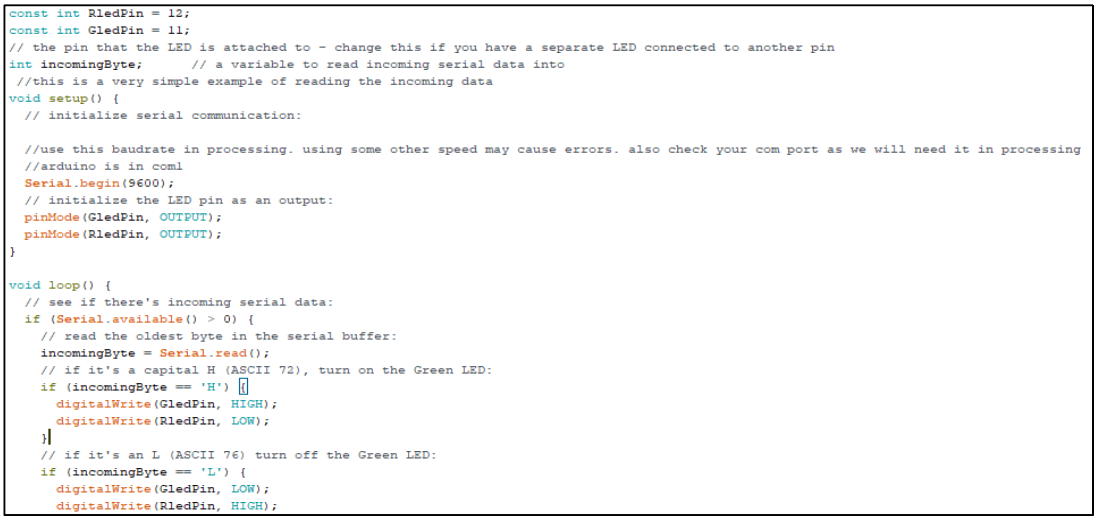


Figure 26: Arduino LED Gate and QR Code Boarding Scanner



```
const int RledPin = 12;
const int GledPin = 11;
// the pin that the LED is attached to - change this if you have a separate LED connected to another pin
int incomingByte;      // a variable to read incoming serial data into
 //this is a very simple example of reading the incoming data
void setup() {
  // initialize serial communication:

  //use this baudrate in processing. using some other speed may cause errors. also check your com port as we will need it in processing
  //arduino is in com1
  Serial.begin(9600);
  // initialize the LED pin as an output:
  pinMode(GledPin, OUTPUT);
  pinMode(RledPin, OUTPUT);
}

void loop() {
  // see if there's incoming serial data:
  if (Serial.available() > 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();
    // if it's a capital H (ASCII 72), turn on the Green LED:
    if (incomingByte == 'H') {
      digitalWrite(GledPin, HIGH);
      digitalWrite(RledPin, LOW);
    }
    // if it's an L (ASCII 76) turn off the Green LED:
    if (incomingByte == 'L') {
      digitalWrite(GledPin, LOW);
      digitalWrite(RledPin, HIGH);
```

Figure 27: Arduino LED Gate Controlled Code

4.5 User Acceptance Test (UAT)

UAT is a type of testing in which the end user or client verifies and accepts the software system before it is moved to the production environment. UAT assists developers in determining the satisfaction of the end user to confirm the quality of the application under test. Table 7 shows the test category of the proposed system, Table 8 shows the result of UAT of the proposed system from the end user, and Table 11Table 11 shows the security checklist of the proposed system.

Table 7: Test category of the proposed system

Test Category	Description
1	Test the functionality of the proposed system by role of administration
2	Test the functionality of the proposed system by role of customer

Table 8: Result of UAT of the proposed system from end user

Module	Test Category	Description	Expected Result	Actual Result
User management module	1,2	Login to the system: I. Insert email II. Insert password III. Click login	System log the user in if the user's input is valid, else it should not log the user in.	Pass
	1,2	Logout from the system: I. Click logout	System log the user out after the user clicks on the logout button.	Pass
	1	Create new user account: I. Create new admin	I. New user form is shown. II. Data stores into the database after clicking the add button. III. A confirmation message appears to let the user know the account was successfully created	Pass
	1	Create, view, update, and delete the user's (administrator or customer) account	I. The users account list is displayed. II. Confirmation message appears after clicking on the button, update or delete button. III. Data is updated and recorded to the database after clicking the confirm button in the confirmation message.	Pass
	2	Register as new user: I. Fill in register form	I. User register form is shown. II. Error message appears if the input is invalid. III. Data stores into the database after clicking the register button. IV. A confirmation message appears to let the user know the account was successfully created.	Pass
	2	View and update user's profile: I. Show the user's profile II. Update profile	I. User's profile is shown with the user's name, email, IC number, contact number, and gender. II. Confirmation message appears after clicking on the update button. III. Data is updated after clicking the confirm button in the confirmation message.	Pass

Table 9: Result of UAT of the proposed system from end user (continue)

Module	Test Category	Description	Expected Result	Actual Result
Ticket reservation module	1	Manage seat's status: I. Select the date and voyage II. Select the seat III. Select the seat's status	I. The voyage's seat list is shown with the seat status. II. The seat status is updated and data is stored into the database.	Pass
	2	Book ticket: I. Select trip II. Select vessel schedule III. Select seat IV. Check passenger information V. Confirm passenger information VI. Make payment	I. Trips include origin, destination, departure date, return date, and number of passengers (pax) appear to select. II. Available vessel schedules appear to select. III. Available seats appear to select. IV. Passenger form appears to fill in the passenger's personal details. V. An editable passenger confirmation form appears. VI. Payment amount is calculated and redirected to the "Make payment" page after clicking proceed to payment button. VII. Page redirect to the payment gateway. VIII. E-ticket with QR code and receipt is sent to the user via email.	Pass
Vessel module	1	Manage vessel: I. Add/view/update the vessel's information	I. Vessel list is shown with details (vessel's image, name, model, capacity, year of build, year of register, depth, breadth, LOA, GRT, NRT and status) II. Confirmation message appears after clicking on the add/update button. III. Data is updated and recorded to the database after clicking the confirm button in the confirmation message.	Pass
	1	Manage vessel schedule: I. Add/view/update/delete the vessel schedule	I. Vessel schedule is shown with the schedule date, time, and terminal. II. Confirmation message appears after clicking on the add/update button. III. Data is updated and recorded to the database after clicking the confirm button in the confirmation message.	Pass
Sales record module	1	Manage the sales record: I. Generate sales report II. View sold tickets in details	I. The sales report is displayed with the correct date range selected, total number of sold tickets, total profit, and sold tickets' details in PDF. II. PDF is downloadable.	Pass
Payment module	2	View the quantity, and price of the ferry ticket.	I. System shows the purchase details correctly. II. System redirects the user to the payment gateway to continue the procedure.	Pass

Table 10: Result of UAT of the proposed system from end user (continue)

Module	Test Category	Description	Expected Result	Actual Result
Boarding scanner module	2	Scan the QR code on the QR scanner	I. System read and verify the data from the QR code. II. LED light emits green light if the QR code is valid, else it emits red light. III. Check-in or check-out times are updated and stored in the database.	Pass

Table 11: Security checklist for the proposed system

No	Checklist	Actual Result
1	Ensure the error message does not directly indicate which part of the authentication and data is incorrect. For example, an error message should not show “incorrect password” or “incorrect email”.	Pass
2	Enforce the complexity of the password. For example, requiring use of alphabetic and numeric as well as special characters to create password.	Pass
3	Enforce password length inside the password policy. For example, minimum 8 characters and maximum 64.	Pass
4	Password should be obscured in the textbox.	Pass

6. Conclusion

Upon completing this project, an IoT-based ferry e-ticketing system with QR code will be launched. This IoT-based e-ticketing system with QR code is recommended for use by Bahtera Ferry Services Sdn. Bhd. It simplifies things for customers while also cutting the cost of making the call and printing all of the tickets on a single sheet of paper. The system includes QR code technology, which makes it easier for the system to record passenger verification data by scanning the QR code with a QR code scanner and an Arduino circuit. The green and red LED lights help to indicate the validity of the QR code. In the future, this system will be maintained and upgraded to fix bugs and errors and make improvements.

Acknowledgment

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

References

[1] Biddulph, R., & Scheyvens, R. (2018). Introducing inclusive tourism. *International Journal of Tourism Space, Place and Environment, 20*(4), 583-588. doi:10.1080/14616688.2018.1486880

[2] Dennis, A., Wixom, B. H., & Roth, R. M. (2012). *System Analysis And Design* (5th ed.). Hoboken, New Jersey: John Wiley & Sons. Retrieved from http://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam/Systemanalysisanddesign.pdf

[3] Freeman, J. (2021). *Complete Guide to Architecture Diagrams*. Retrieved from https://www.edrawsoft.com/architecture-diagram.html

[4] K.Talley, W. (2021). Passenger Ferry Vessels and Cruise Ships: Safety and Security. *International Encyclopedia of Transportation,* 290-295. doi:https://doi.org/10.1016/B978-0-08-102671-7.10138-1.

[5] Law office Malaysia. (2016). *Maritime Law in Malaysia*. Retrieved from Law Malaysia: https://www.lawofficemalaysia.com/maritime-law-in-malaysia

[6] Miniwatts Marketing Group. (2021). *Internet Growth Statistics 1995 to 2019 - the Global Village Online*. Retrieved from Internet World Statistics: https://www.internetworldstats.com/emarketing.htm

[7] Notepub. (2021). *SDLC – Prototyping Model*. Retrieved from Notepub: https://notepub.io/notes/software-engineering/software-development-life-cycle/sdlc-prototyping-model/

[8] NY. (2020). Section III:2 System Requirements Analysis. In J. T. Dillon, & M. McCormack, *NYS Project Management Guidebook* (pp. 31-70). New York State Office for Technology. Retrieved from https://its.ny.gov/sites/default/files/documents/systemreq.pdf

[9] Russell Publishing Limited. (2021). *Examining the cost benefits of contactless e-ticketing in public transport*. Retrieved from Intelligent Transport: https://www.intelligenttransport.com/transport-articles/115016/electronic-tickets-yield-cost-savings-of-up-to-85/

[10] Serap, R., Jabai, D. R., Kamu, A., Hassan, D., & Adam, N. M. (2017). Passengers perception on safety level of ferry transport: A case study in Labuan island, Malaysia. *Journal of Advanced Research in Social and Behavioural Sciences, 6*(1), 7-17 . Retrieved from https://www.akademiabaru.com/doc/ARSBSV6_N1_P7_17.pdf

[11] Techrish Solutions. (2021). *The 7 Phases of Web Development Life Cycle*. Retrieved from Techrish: https://techrish.com/the-7-phases-of-web-development-life-cycle/

[12] Yuzri, E. (2021). *HSC PASSENGER FERRY RESEARCH MALAYSIA MARKET STUDY SUPPLY AND DEMAND*. Retrieved from Academia.edu: https://www.academia.edu/40951773/HSC_PASSENGER_FERRY_RESEARCH_MALAYSIA_MARKET_STUDY_SUPPLY_AND_DEMAND

[13] Zameer, M. A., & Logeswaran, R. (2020). WEB-BASED FERRY BOOKING SYSTEM WITH LIVE LOCATION TRACKING. *Journal of critical reviews, 7*(3), 110-113. doi:http://dx.doi.org/10.31838/jcr.07.03.21