

Mobile-Based Dine-In Self-Ordering Application with QR Code

Tan Gui Mei¹, Nur Afifah Sahadun^{1*}

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

*Corresponding Author: nurafiqah@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2026.07.01.022>

Article Info

Received: 12 June 2025

Accepted: 18 June 2026

Available online: 30 June 2026

Keywords

Self-Ordering, Firebase, QR code,
mobile application

Abstract

Online food ordering systems have become a revolutionary approach to replace traditional methods by creating a contactless environment when taking orders in the food and beverage industry. This project developed a mobile-based dine-in self-ordering application with QR code for Lido Seafood Restaurant. The application was built using the waterfall methodology, utilizing VS Code, with Android Studio used for SDK and emulator setup, as well as Flutter and Dart. Key integrations included Firebase Authentication, Firebase Hosting, Firestore Database, Cloudinary, and Curlec (powered by Razorpay). Customers now conveniently place orders by scanning table-specific QR codes on their mobile devices while the administrator generates QR codes for each table, manages menus, and oversees reports through a dedicated mobile admin dashboard. The completed system underwent system functionality testing and user acceptance testing which confirmed that it meets user requirements. Overall, this digital solution enhances customer satisfaction and boosts dine-in efficiency.

1. Introduction

Nowadays, technology is becoming increasingly crucial to improve operational efficiency and enriching customer experiences, especially in the fast-paced and dynamic restaurant industry [1]. The implementation of the self-ordering concept in the ordering system can enhance restaurant operations when customers are able to browse a digital menu, choose what dishes they want, personalize their orders, and make transactions from their smartphones. This concept can be combined with Quick Response (QR) codes, which have a distinctive image and are frequently used to transfer URLs. They are becoming a growing trend due to the increasing prevalence of mobile devices in our daily lives by utilizing smartphone cameras [2].

Currently, paper-based menus and genius point-of-sale (POS) system are used by Lido Seafood Restaurant to handle order taking. This can result in several common issues such as inefficiencies brought on by the laborious process of manually entering orders, lack of effective mechanisms for the interaction between administrators and customers with the latest information regarding new meals are introduced and order status, lose out on important information that could enhance operation and service without an organized method in place to collect feedback from customers and there is difficulty in placing extra requests to personalize the meals especially the customer who wants to specify their preferences such as spice levels, component replacements, or dietary restrictions in the absence of designated spaces.

As a result, this project focused on helping the owner of Lido Seafood Restaurant, located in Sitiawan, Perak since 1979 and known for its Chinese cuisine, especially Hock Chew dishes, by developing a mobile-based online self-ordering system. The system was built using an object-oriented approach with Android technology and framework to replace the traditional paper-based menu and manual order entry by waitstaff, allowing

customers to place orders by scanning QR codes in a dine-in environment. The expected outcome was that its functionality and performance would fulfill the requirements by enabling customers, as the primary users, to view the menu and place orders online through a user-friendly interface, while the administrator (owner or waiter) could access data in the admin panel and, if needed, perform CRUD (Create, Read, Update, Delete) actions to enhance business profitability and operational efficiency.

2. Related Work

This section discusses the technology and algorithm used to develop mobile-based dine-in self-ordering application with QR code for Lido Seafood Restaurant. Moreover, the domain background of Lido Seafood Restaurant. In addition, there are comparison with three similar existing system that are McDonald's mobile application, KFC mobile application and Tealive mobile application presented in a table.

2.1 Technology Used

Android Studio, the official Integrated Development Environment (IDE) created by Google specifically for Android app development has been used because of its powerful code editor, a visual layout editor, advanced debugging tools and Gradle integration [3]. Besides, Firebase allows for real-time user data synchronization which implies several users can view the changes when data is created or modified [4] and storage as JSON objects also it uses a JavaScript API to communicate with the database, and it may be scaled horizontally [5]. Moreover, QR code has been utilized to retrieve information by scanning devices to scan the code. In addition, Dart is a versatile and open-source programming language that is used to design this application. Furthermore, Google's Flutter mobile cross-platform development framework allows for native app development across from a single codebase, making it a cutting-edge breakthrough [6]. Furthermore, Cloudinary is a cloud-based photo and video management solution with photo transformation features to optimise assets and the ability to serve as a CDN [7]. Lastly, Curlec by Razorpay's direct payment processing ensures secure and transparent transactions, unlike traditional systems that rely on third-party payment services [8].

2.2 Existing Ordering Process in Lido Seafood Restaurant

The restaurant does not currently feature an online self-ordering system, instead depending on waitstaff who utilize a POS system to take customer orders and provide a paper-based menu for viewing, while describing the dishes verbally upon request.

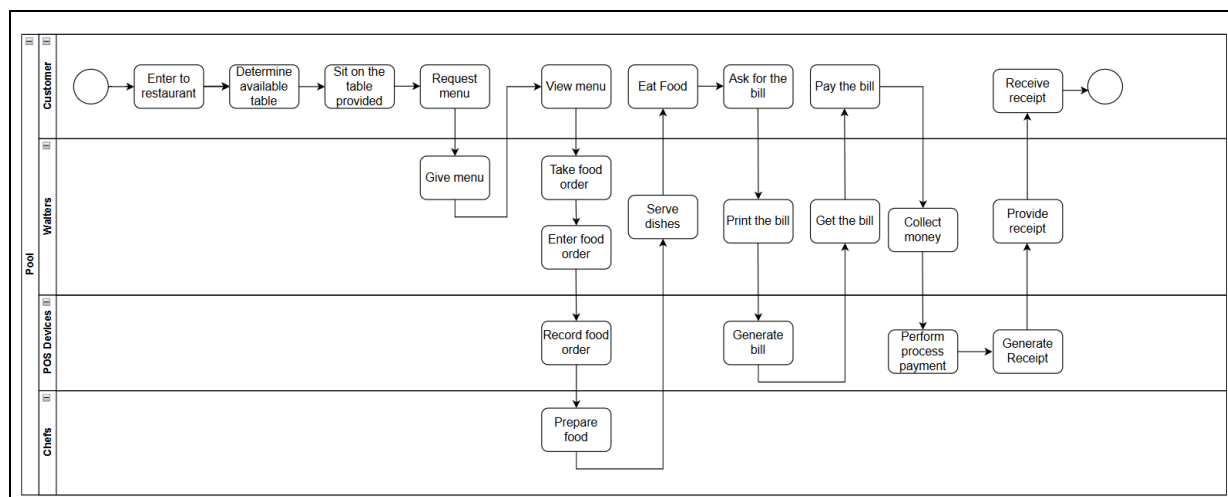


Fig. 1 Business process in Lido Seafood Restaurant

2.3 Study of Existing Related System

First and foremost, McDonald's App allows them to easily choose items and place orders within five miles of a restaurant that accepts Mobile Order & Pay. The restaurant will charge the user's card and begin preparing the user's order as soon as users check-in. Customers will save time because they won't have to wait for their turn [9].

Besides, KFC has introduced a delivery app for iOS and Android users. Customers may pick the delivery address or the nearby restaurant you wish to order from, browse our menu by making a preferred option, and then add the item to the cart. Contactless payment can be made and the order verified after users make selections including any deals and rewards [10].

The Tealive team from Loob Holding has created a lifestyle app that is available on the Android and iOS platforms. One of the main attractions for app users is its own rewards program. Tealive also makes sure that its most devoted clients are consistently involved and rewarded using monthly challenges to create a network that encourages users to constantly interact with the app by giving them access to intriguing, app-only promotions along with the opportunity to earn points [11].

The comparison is based on the respective characteristics or features of the existing applications which are McDonald's Mobile Ordering Application, KFC Mobile Ordering Application, Tealive Ordering Application, and the proposed application is tabulated in a table.

Table 1 Comparison between three existing applications and proposed applications

Features/System	McDonald's Mobile Ordering Application	KFC Mobile Ordering Application	Tealive Mobile Ordering Application	Developed System
Mobile-based	✓	✓	✓	✓
Login and Registration	✓	✓	✓	✓
Utilize QR code to order from table	✗	✗	✗	✓
Supports dine-in self-ordering	✗	✗	✗	✓
Authentication	✓	✓	✓	✓
Search Functionality	✗	✗	✗	✓
Meal Customization	✓	✓	✓	✓
Food Cart	✓	✓	✓	✓
Rating and Feedback	✗	✗	✓	✓
Push notification	✓	✓	✓	✓
Electronic Payment	✓	✓	✓	✓

3. Methodology

The Waterfall model has been chosen as the software development life cycle (SDLC) for this project. The waterfall model is a conventional project management approach that emphasizes a sequential and linear method, structured around distinct phases that must be completed before progressing to the next [12]. The Waterfall methodology consists of seven distinct stages: planning, analysis, design, implementation, testing, deployment, and maintenance. Each stage has its own specific outcomes and a review process to ensure a consistent overall understanding of the system's requirements.

The workflow in each phase by determining the task and output so that the project will be manageable by dividing the development process and clearly defining the task and output that needed to be accomplished and achieved at every step.

Table 2 Task and Output produced in each phase

Phases	Task	Output
Planning	- Determine project title - Preparing a proposal with several elements - Identify and select related stakeholders - Schedule time by assigning specific tasks during the timeframe to complete the project	- A detailed proposal has been generated with the following elements such as introduction problem statement objective and scope - Grant permission from stakeholder for interview purpose - Gantt chart

Table 2: (cont)

Analysis	• Prepare interview question • Schedule time to undergo an interview • Determine and analyze problems from the existing system and user requirements • Select suitable hardware and software	• Summary of the interview • Requirement Specification document • Hardware Specification document • Software Specification document
Design	• Design user interface • Design workflow for the system • Design database schema	• Wireframe • Use Case Diagram • Sequence Diagram • Activity Diagram • Class Diagram
Implementation	• Implement user interface design • Code the functionality of each module with backend and utilize Application Programming Interface (API)	• Installed related software • Aesthetic user interface that allows navigation from one page to another page • Integration of functional module with operations such as Create, Read, Update and Delete in the Firebase
Testing	• Conduct unit testing • Undergo incremental integration testing • Perform user-acceptable testing	• Test plan • User Acceptance Testing Form
Deployment	• Perform hosting for a flutter website by using command firebase deploy in command prompt from the command flutter build web	• Archive an URL link to integrate with QR code

4. System Analysis and Design

4.1 System Requirements

Functional requirements concern the system's behavior and features, which must be implemented for users to complete tasks. On other hand, non-functional requirements, which emphasize how a system should act rather than what it does, specify the limitations or circumstances under which it must perform.

Table 3 Functional requirements of the proposed system

Module	Description
Login	The system must allow customers and the administrator to log in to the application by inputting a valid email address and password as well as reset their password if they have forgotten their password through email.
Logout	The system allows the administrator and customers to log out of the application.
Register	The system should allow a new customer to register his information such as name before logging into application.

Table 3: (cont)

Menu	The system must display the menu with dishes served by Lido Seafood Restaurant, which is divided into different categories.
Food cart	The system must allow customers to add their selected dishes into the food cart then undergo operation such as increasing number of dishes and deleting it.
Search functionality	The system should allow customers to search for their preferred dishes.
Payment	The system should provide cashless payment to the customer compared to only traditional cash payment.
Rating and Feedback	The system should allow the customer to give ratings and feedback regarding the service of the restaurant.
Notification	The system should notify new order to the administrator.
Customer Management	The system should allow the administrator to view, update and delete customer information such as name, phone number and email address.
Menu Management	The system should allow the administrator to view, add, delete and update details about dishes.
Order Management	The system should allow the administrator to filter based on order status, view and update the status of the order.
Payment Management	The system should allow the administrator to view and update the payment status.
Rating and Feedback Management	The system allows the administrator to view the rating and feedback given by the customer.
Category Management	The system allows the administrator to view, add, update and delete the category of dish.
Report Management	The system should allow the administrator to view and generate information regarding orders and sales daily, weekly, and monthly, top-selling dishes, save the report as a PDF, and print the report.
QR code Generator	The system should allow the administrator to generate QR code for each table.

Table 4 Non-Functional requirements of the proposed system

Module	Description
Operational	The system is designed to operate on Android mobile devices.
Performance	The system needs to efficiently manage many orders when dealing with busy periods of operation.
Security	The system involves implementing regulations for encryption when storing passwords in the database and blocking any unauthorized attempts to access sensitive data.
Usability	The system needs to be user-friendly for both customer and restaurant staff or owner, featuring a straightforward and intuitive interface along with easy navigation.

4.2 Use Case Diagram

Use Case diagram illustrates the functions that a system should have, and which users should utilize which features, it ignores any implementation specifics such as data structures, and algorithms. It is used to explain the customer's needs and the purpose of the future system throughout the analysis and design phase.

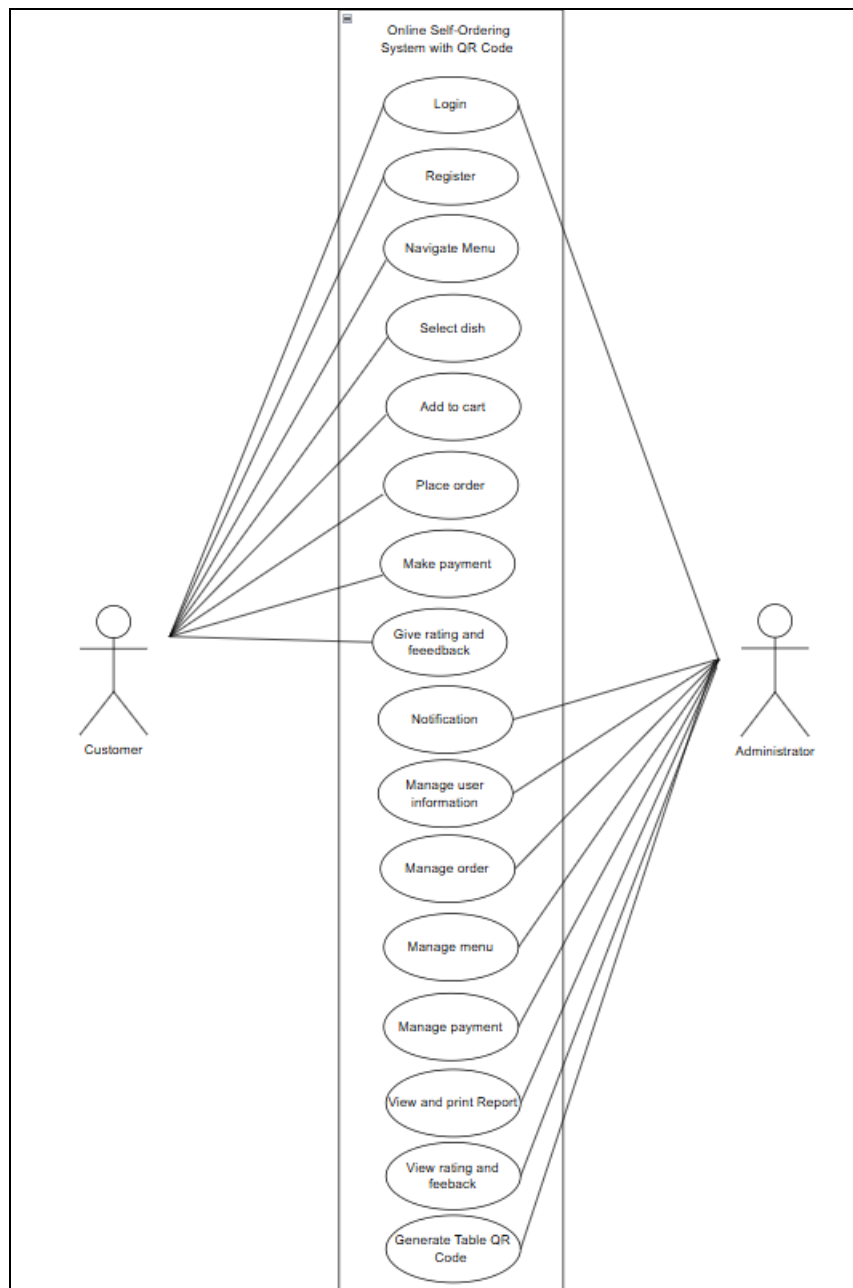


Fig. 2 Use Case of Proposed System

4.3 Sequence Diagram

An essential part of the Unified Modelling Language (UML), a sequence diagram— also known as an event diagram—is used to show how objects interact with one another in a sequential order. It represents any two lifelines' communication as a time-ordered series of events, indicating that these lifelines participated during the runtime. In UML, the message flow is represented by a vertical dotted line that runs across the bottom of the page, while the lifeline is represented by a vertical bar. It includes both branches and iterations.

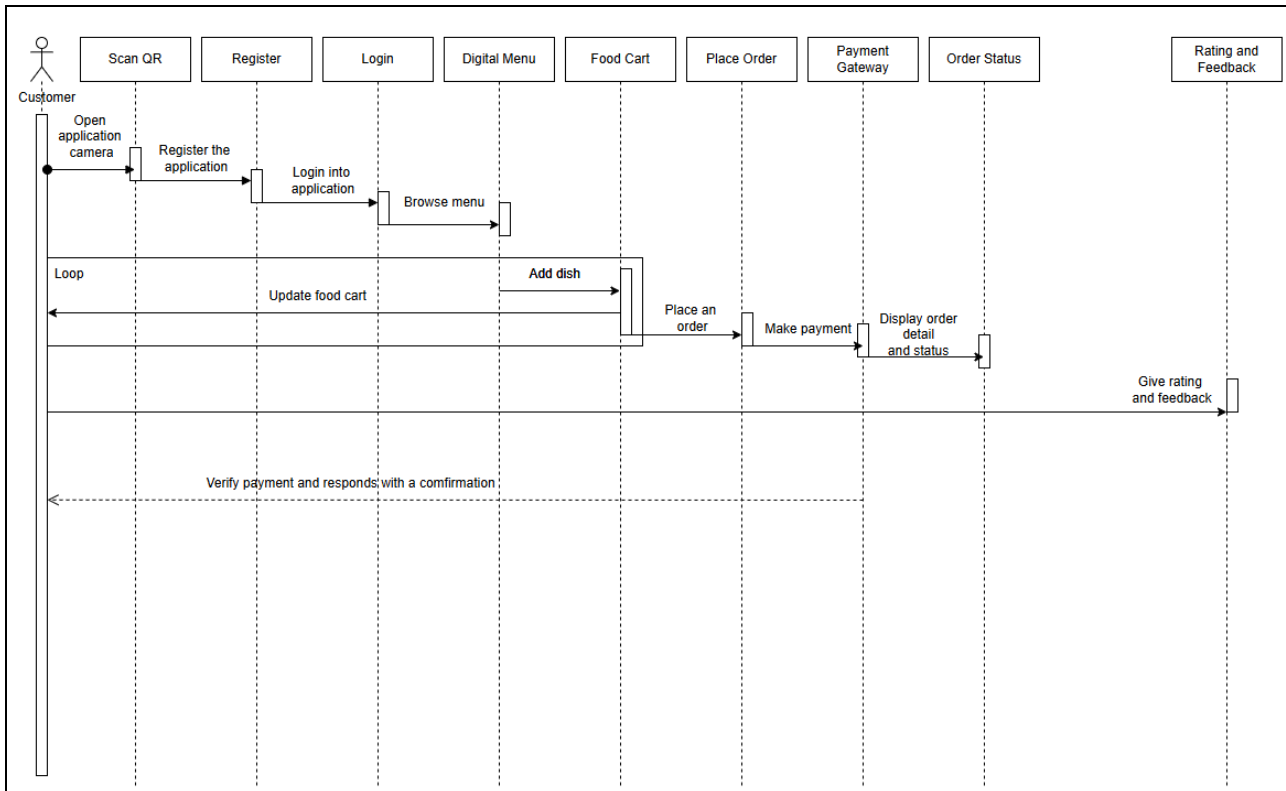


Fig. 3 Sequence Diagram of Customer

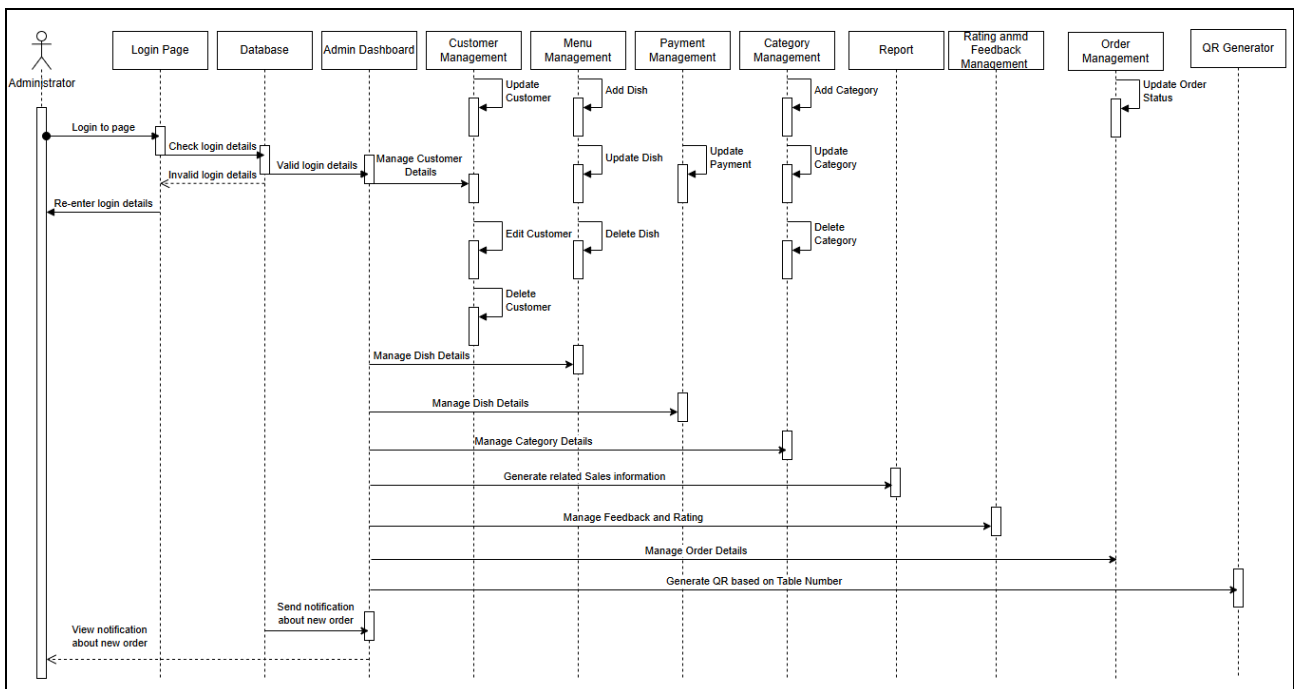


Fig. 4 Sequence Diagram of Administrator

4.4 Activity Diagram

Activity diagrams illustrate the processes involved in how a system operates, aiding in the comprehension of control flow. They depict the sequence of activities and indicate whether they happen consecutively (sequential) or simultaneously (concurrent).

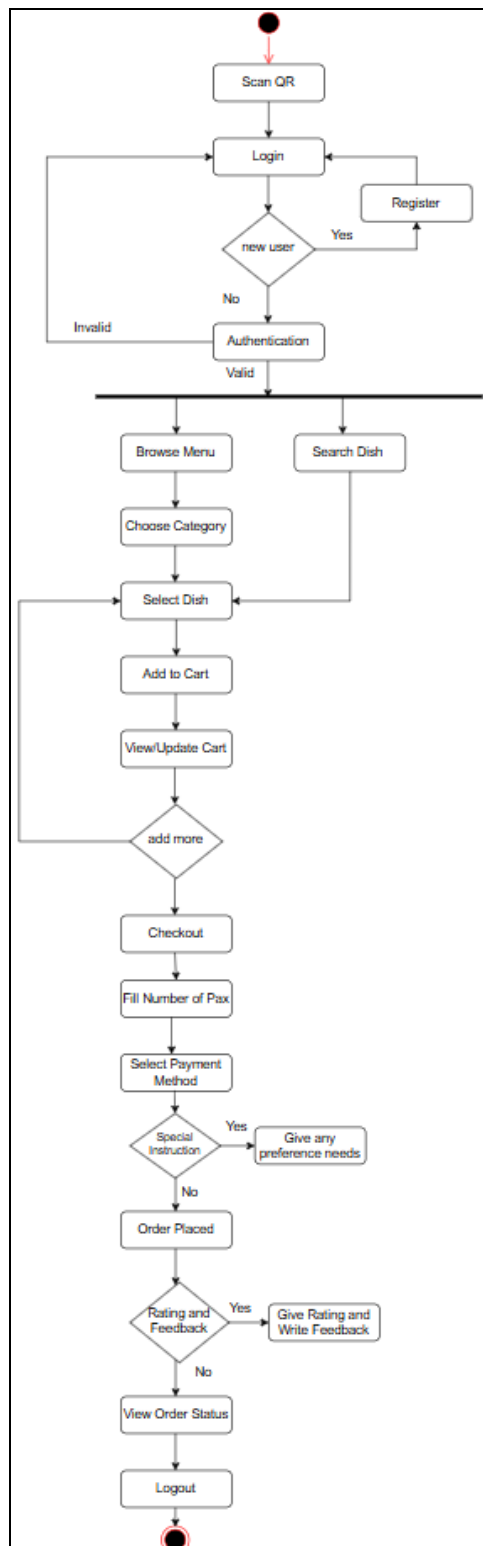


Fig. 5 Activity Diagram for Customer

4.5 Class Diagram

A class diagram highlights the different classes that the system might have, along with their attributes, specific operations, and the relationships formed among objects within that system.

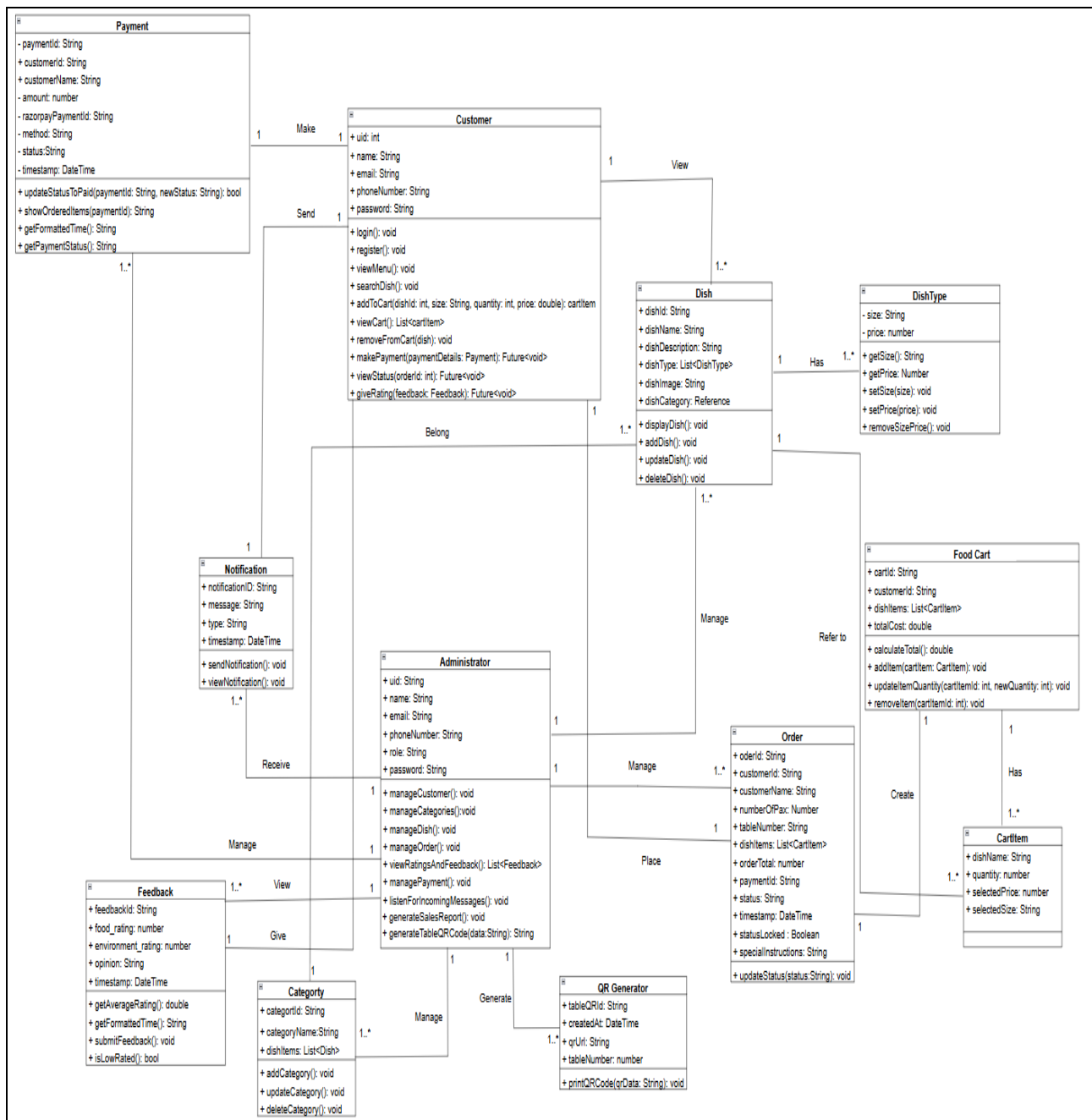


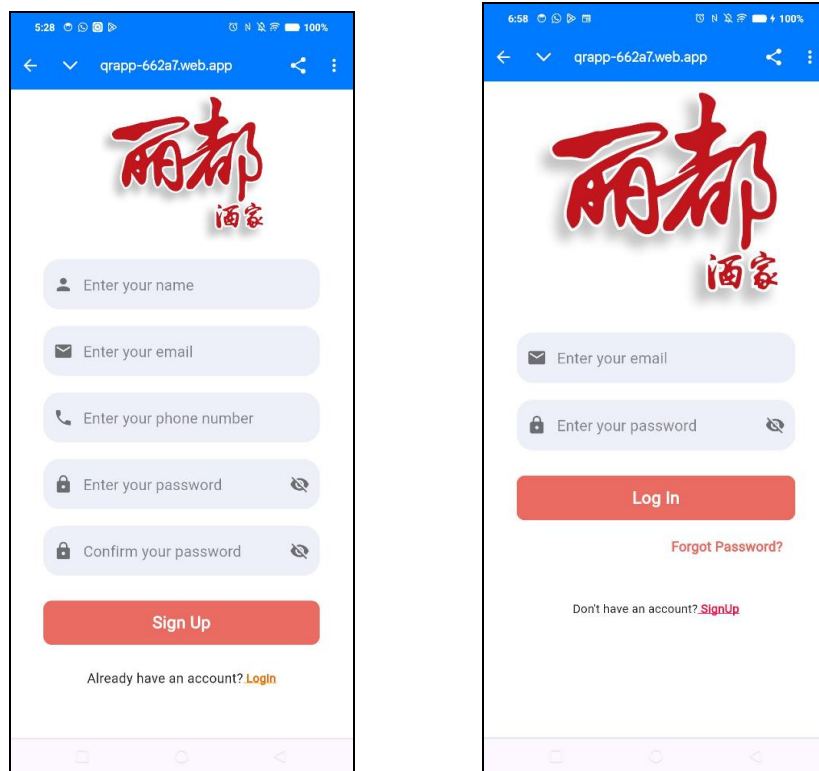
Fig. 6 Class Diagram of Proposed System

5. Implementation and Testing

5.1 Implementation

This section presents and discusses the implementation of the interfaces and functionalities of each module on both the customer and administrator sides of the mobile application, as shown in the outputs below.

5.1.1 Implementation of Customer Sign Up and Login Module



(a)

(b)

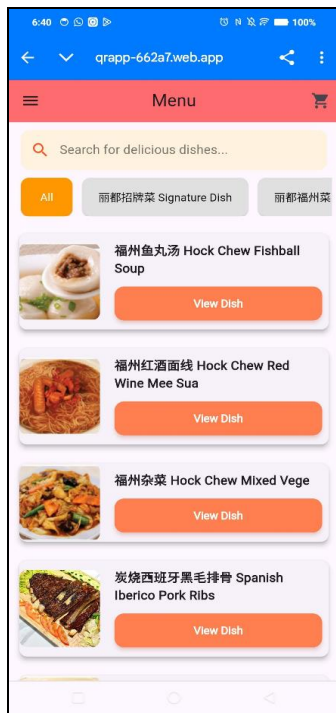


(c)

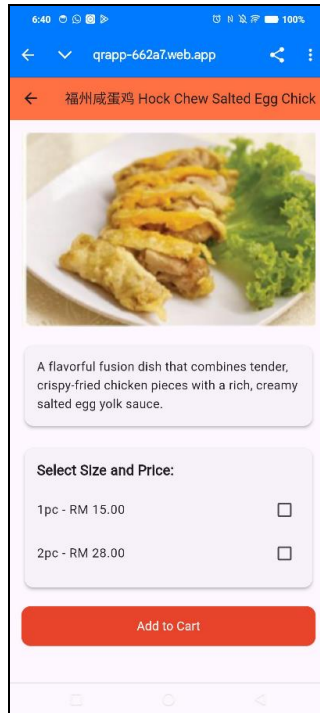
Fig. 7 (a) Customer Sign-Up Interface; (b) Customer Login Interface; (c) Reset Password Interface

Figure 7(a) shows an entry point for customers to sign up after scanning a QR code by providing their personal information such as name, email address, phone number, password and confirm password for security purposes with password visibility toggle for easier entry. An account is successfully created after pressing the Sign-Up button if all the information is given in the correct formats, then a pop-up dialog will notify them. There is also a clickable link “Login” for the customers who have signed up when revisiting the restaurant to navigate login page. Figure 7(b) shows a login page for authorized customers attempting to access the system to provide their email and password that need to be verified credentials after pressing the login button. Meanwhile, customers can reset their password through their email by clicking on the link shown in Figure 7(c) then appear a pop-up dialog to enter a new password. The Sign-up clickable link helps customers who have not created an account navigate back to the Sign-up page.

5.1.2 Implementation of Menu, Dish Detail, Food Cart, Checkout, Payment, Order Status and Feedback Module



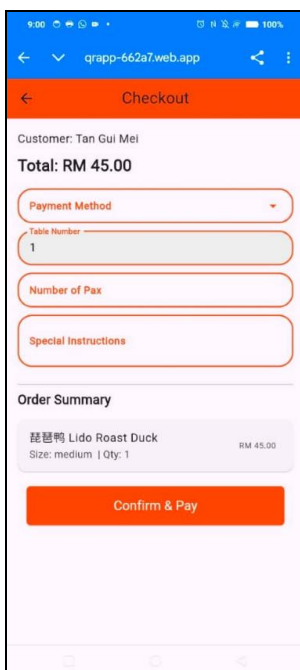
(a)



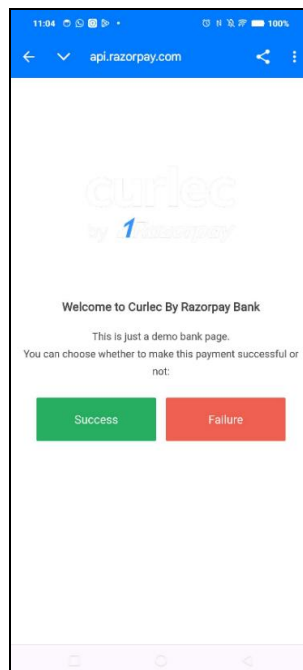
(b)



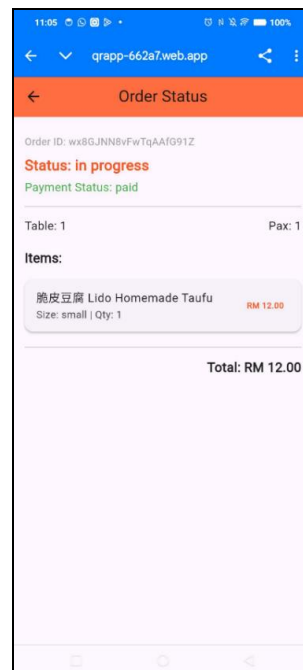
(c)



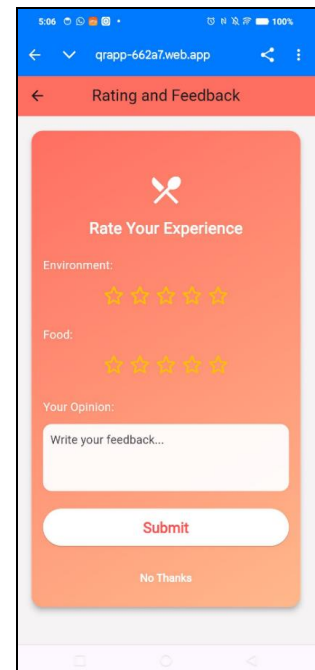
(d)



(e)



(f)



(g)

Fig. 8 (a) Menu Interface; (b) Dish Detail Interface; (c) Food Cart Interface; (d) Checkout Interface; (e) Curlec by Razorpay Interface; (f) Order Status Interface; (g) Rating and Feedback Interface

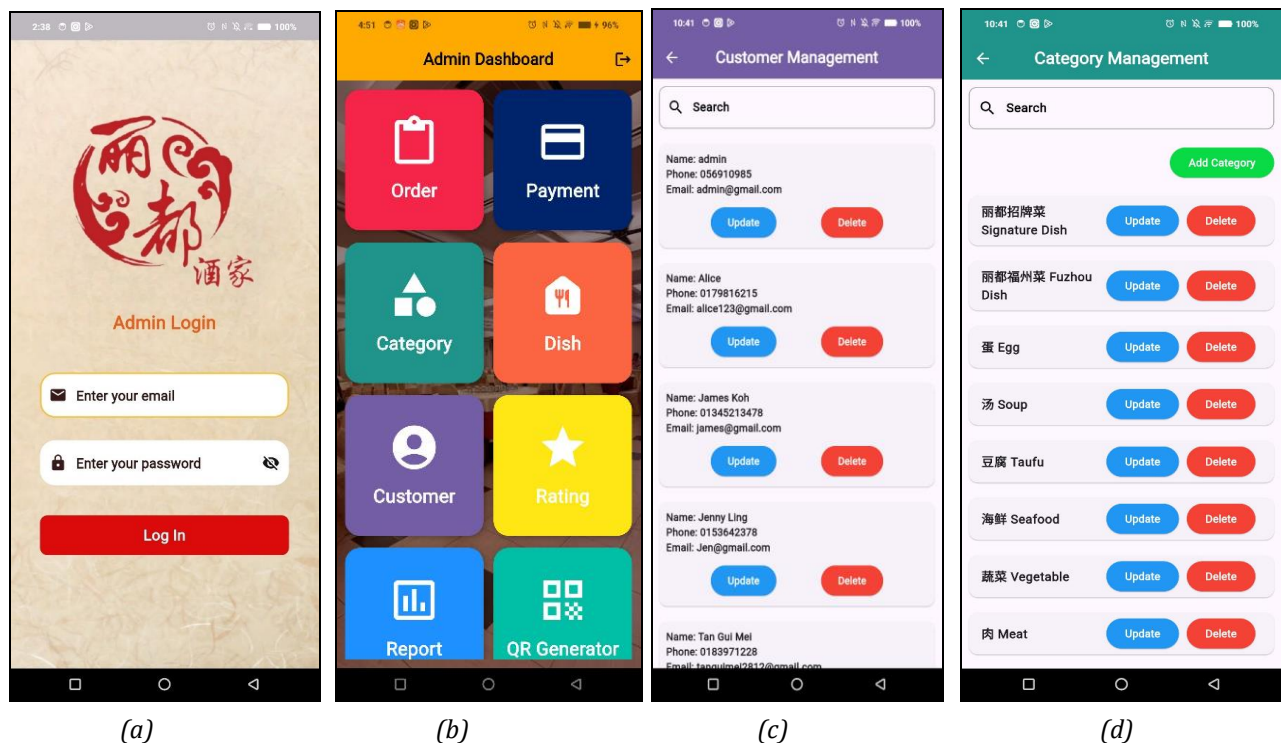
Figure 8(a) shows a digital menu for customers to scroll horizontally to view and choose a category and it will display the dish based on that category. Customers can also search for dishes using Chinese or English. The view button helps navigate each dish's details shown in Figure 8(b) with dish information including horizontally scrollable dish name, dish image, dish description, price, and size the customer to view and tick his preference

size and price then press the Add to Cart button for the food cart page navigation in around 1 second automatically. At the same time, the drawer has several components such as My Order, Rating & Feedback and Logout allow customers to select the activity they want to undergo. It consists of food cart icons that enable customers to temporarily store their selected dish as well as perform add and reduce quantity of the dish concurrently with the subtotal and total price that has shown in Figure 8(c) before moving to checkout and payment page by pressing the button.

Figure 8(d) shows a checkout page to allow the customer to select payment method by using cash or credit card, fill in table number, number of pax and special instructions if required. This page also presents the name of the current user, total price and order summary to the customer. The back arrow allows the customer to navigate back to the food cart page. After confirming all the information and choosing their preferred payment option, customers can press the "Confirm and Pay" button to place the order. A confirmation message is sent to the customer to let them know that their order has been received and is being prepared after it has been successfully submitted. Figure 8(e) shows the verification of payment page for credit cards from the integration of Curlec by Razorpay after the customer provides his card number, card expired date, card verification value, and name of card. Figure 8(f) shows the order status page when the customer has successfully placed an order. It enables the customer to view order id, order status, payment status, table number, number of pax, items have been selected with total price. The customer can go back to the menu by pressing the back arrow icon.

Figure 8(g) shows rating and feedback located at the drawer in the menu page allows the customer to give rating regardless of food and environment of the restaurant also write their comment if he has any valuable suggestion for improvement by pressing submit button or no thanks to navigating back to menu page.

5.1.3 Implementation of Admin Login, Admin Dashboard, Customer Management, Category Management, Dish Management, Order Management, Customer Feedback and Payment Management, Report and Table QR Generator Module



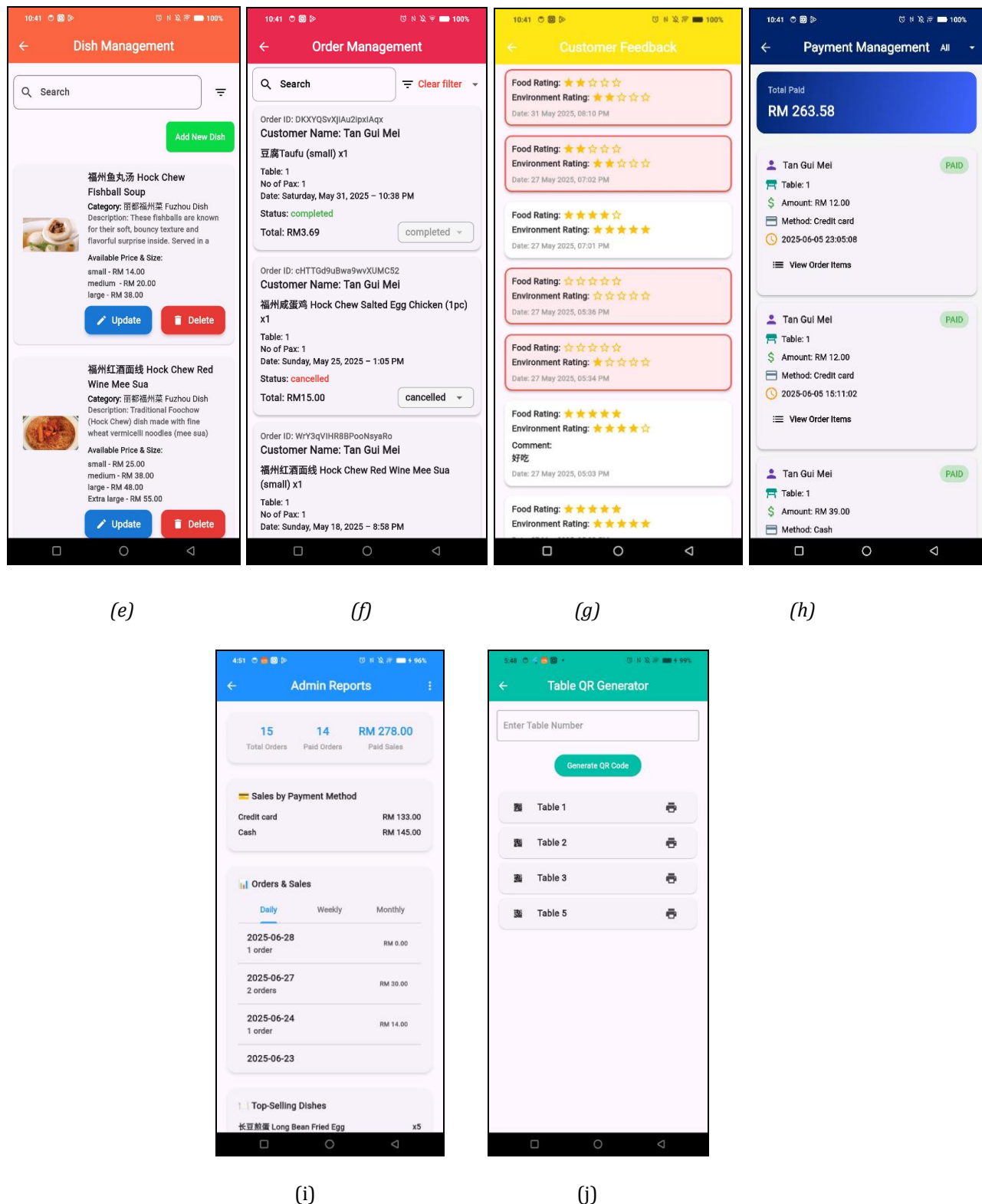


Fig.9 (a) Admin login Interface; (b) Admin Dashboard Interface; (c) Customer Management Interface; (d) Category Management Interface; (e) Dish Management Interface; (f) Order Management Interface; (g) Customer Feedback Interface; (h) Payment Management Interface; (i) Report Interface; (j) QR Generator Interface

Figure 9(a) shows an administrator's login page that the administrator needs to provide his email address and password with correct credentials only can successfully login into admin dashboard in Figure 9(b) which he can perform various tasks allowing operation in restaurant become more efficient and effective.

Figure 9(c) shows customer management that admin can effectively access and search as well as modify or eliminate customer records from the system through the available Update and Delete buttons. Figure 9(d) shows the category management page enables the administrator to handle food categories by adding new ones, modifying existing categories, or removing those that are no longer necessary. The search feature simplifying the process of finding specific categories. Figure 9(e) shows the administrator can change the details of existing dishes, add new ones, or remove dishes from the menu. It is simple to handle menu items because each dish entry shows its category, description, and available prices and sizes. The administrator may also use the search box to find certain meals fast or filter items by category.

Figure 9(f) shows order management allows the administrator to view each order detail. At the same time, the administrator can update order status through dropdown then it will appear a confirmation update dialog before changing the status. To locate the necessary information quickly, the administrator can search for specific orders or filter them by status, such as in progress, completed or cancelled.

Figure 9(g) shows customer feedback so the administrator can examine the ratings and feedback given by customers which it will be highlighted in red if the rank for environment and food is low. Figure 9(h) shows the payment management page in which the administrator can see a summary of all payments that have been completed, which includes the total amount paid. The administrator can also verify the payment status and has the option to click "View Order Items" to check the specific items that were ordered for each transaction. The page also features a filtering option to assist the administrator in sorting or viewing payments according to designated criteria such as paid or unpaid. Figure 9(i) shows Admin Reports report summarizing total orders, paid orders, and sales (RM), plus a breakdown by credit card and cash. It displays daily, weekly, and monthly order counts with sales totals to track trends, and lists top-selling dishes with quantities to show customer preferences. Reports can be exported to PDF or printed for sharing. Figure 9(j) shows the administrator can generate a unique QR code by entering a table number. By clicking the print icon, the admin can preview the QR code with the table number then print it directly with thermal printer.

5.2 Testing

The testing phase is a crucial and interactive process for detecting any bugs and vulnerabilities to maintain the project's quality and ensure it meets established standards, commencing once the developer finalizes the development of a system or application.

5.2.1 Result of Test Plan

This test plan details the testing activities conducted to assess the functionality of each module in the mobile-based dine-in self-ordering application with QR code, ensuring that all features operate as expected.

Table 5 *Test Plan Result*

Test ID	Case	Test Case Description	Expected Result	Actual Result
TC001		Test user sign-up and login functionality. 1. Enter valid email and password during registration 2. Login with the same details 3. Login with incorrect email and password 4. Forgot password with a registered email address	1. Successful registration and login for valid credentials 2. Appropriate error message for incorrect input format 3. Login is denied for unregistered accounts 4. Reset password email can be sent to the email inbox with password reset link	Pass
TC002		Test customers can view the menu and choose dishes. 1. Navigate to the menu page 2. Browse and select a dish from the list by pressing the View button 3. View the dish details then select price and size	1. Dishes are displayed correctly with a user-friendly interface 2. Dish details are visible and correct 3. The checkbox of each price and size can be ticked	Pass

Table 5: (cont)

TC003	Test food cart functionality. 1. Tap the Add to Cart button can navigate to the cart page after selecting a dish 2. The increment and decrement buttons for the quantity of dishes are compatible with the total price	1. The selected dish appears correctly in the cart 2. The buttons for changing the quantity function as intended, and the total price is updated appropriately	Pass
TC004	Test food search functionality. 1. Enter a dish name in the search bar whether in Chinese or English language 2. View the search results matches with the keyword 3. Test with both valid and invalid search terms	1. Relevant dishes are displayed when valid search terms are used 2. No results or a "No items found" message appears for invalid search terms	Pass
TC005	Test Curlec by Razorpay payment integration functionality. 1. Choose Curlec by RazorPay as payment method	1. The Curlec by Razorpay payment gateway has been successfully launched 2. The payment is processed correctly	Pass
TC006	Test customers can receive real-time updates in order status. 1. View order status changes from in progress to completed 2. View payment status changes from unpaid to paid especially for cash payment method	1. Once the all the dish is served to the customer, the order status immediately updates to completed 2. Once the customer pays at the counter using cash, the payment status immediately updates to paid	Pass
TC007	Test notifications are received. 1. The customer placed an order then the administrator received a notification about the new order	1. The administrator will be able to receive notification once the customer has made an order	Pass
TC008	Test customer and administrator's ratings and comments functionality. 1. Provide a rating (1 to 5 stars) for a dish and environment also allows to write comments 2. Administrator can view the submitted rating and comment	1. Rating and feedback should be filled properly then successfully be stored in the database 2. The rating and remark provided for less than 2 stars can be viewed by the administrator	Pass
TC009	Test administrator's ability to manage customer data. 1. View customer details 2. Update customer information (e.g., phone number, email) 3. Delete a customer record	1. Customer data is displayed and updated correctly 2. Customer records are deleted successfully	Pass
TC010	Test the administrator's ability to manage the menu. 1. View the list of dishes 2. Add a new dish 3. Update details of an existing dish 4. Delete a dish	1. Menu updates correctly after add/update/delete operations 2. Changes are reflected in the customer view after login	Pass

Table 5: (cont)

TC011	Test the administrator's ability to manage orders. 1. View details of orders 2. Update the status of an order	1. Orders are displayed with the correct details 2. Status updates are saved and reflected on the customer order status page	Pass
TC012	Test the administrator's ability to manage payment statuses. 1. View payment records 2. Update payment status	1. Payments are displayed with the correct details and amount 2. Payment statuses are displayed and updated correctly	Pass
TC013	Test administrator's ability to manage dish categories. 1. View existing categories 2. Add a new category 3. Updating an existing category 4. Delete an existing category	1. Categories are displayed, added, updated and deleted correctly	Pass
TC014	Test logout functionality for customers and administrators. 1. Click the "Logout" button	1. User is logged out and redirected to the login page	Pass
TC015	Test QR generator functionality. 1. Enter a number for a table	1. The QR code has been generated based on the number entered then display in a list. 2. Same table number will not be generated again	Pass

5.2.2 Result of User Acceptance Test

User Acceptance Testing (UAT), also known as end-user or application testing, is a crucial stage in the software development cycle that verifies the software in real-world scenarios by its target users. In this study, two separate Google Forms containing the same system interface questions, but different system functionality questions (based on the developed modules) were distributed to 10 respondents who were randomly selected while dining in at Lido Seafood Restaurant, including the restaurant owner. User satisfaction was rated using a Likert scale: 1 (Strongly Disagree), 2 (Disagree), 3 (Neutral), 4 (Agree), and 5 (Strongly Agree). The results are presented in tables showing, for each question, the number of respondents (n) who selected each score from 1 to 5, and the corresponding percentage (%) out of the total respondents (n=10). Table 6 presents the user acceptance test results for customers, including the number of respondents, percentage for each score, and the average satisfaction score on a scale from 1 to 5.

Table 6: User Acceptance Test Result for Customers

No	Items	Scale											
		1		2		3		4		5		Average	
		n	%	n	%	n	%	n	%	n	%		
System Interface													
1	The system is responsive across mobile device	0	0	0	0	0	0	8	80	2	20	4.2	
2	The interface is visually appealing and user-friendly.	0	0	0	0	2	20	8	80	0	0	3.8	
3	Navigation through the system is simple and intuitive.	0	0	0	0	1	10	7	70	2	20	4.1	
4	The font size and colors are clear and readable.	0	0	0	0	1	10	9	90	0	0	3.9	

Table 6: (cont)

5	Icons and images are clear and relevant to their purpose	0	0	0	0	3	30	7	70	0	0	3.7
System Functionality												
1	Login and registration process is smooth.	0	0	0	0	0	0	9	90	1	10	4.1
2	Search for and add items to the cart easily.	0	0	0	0	2	20	7	70	1	10	3.9
3	The order placement process is straightforward.	0	0	0	0	1	10	7	70	2	20	4.1
4	The order and payment status can be view clearly	0	0	0	0	2	20	2	20	6	60	4.4

6. Conclusion

The development of mobile-based dine-in self-ordering application with QR code successfully met its main objectives of enhancing the dine-in experience at Lido Seafood Restaurant. The system provided a responsive and user-friendly mobile interface that allowed customers to view menus, place orders, and track order and payment statuses easily. For administrators, it offered effective tools to manage menu items, monitor orders, and oversee customer data, improving operational efficiency. However, the project had certain limitations. Testing was conducted only in a single restaurant setting with a small number of participants, which may not fully represent broader customer needs or operational scenarios. Additionally, advanced features such as AI-driven meal recommendations, real-time inventory tracking, and promotional campaign management were not included in this version. Addressing these areas in future work would help improve scalability, enrich user experience, and add further value to the restaurant's business operations.

Acknowledgement

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

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

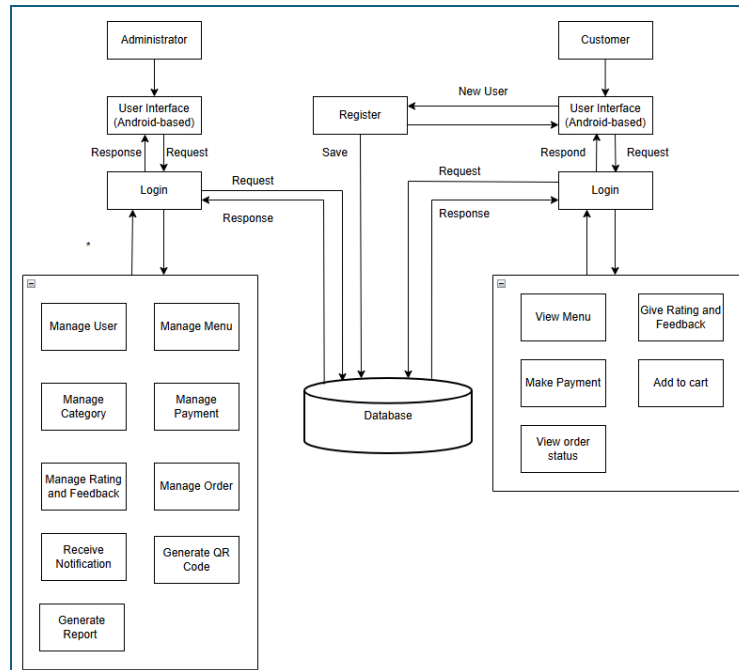
The authors confirm contribution to the paper as follows: **study conception and design:** Tan Gui Mei, Nur Afiqah Binti Sahadun; **data collection:** Tan Gui Mei, Nur Afiqah Binti Sahadun; **analysis and interpretation of results:** Tan Gui Mei, Nur Afiqah Binti Sahadun; **draft manuscript preparation:** Tan Gui Mei, Nur Afiqah Binti Sahadun. All authors reviewed the results and approved the final version of the manuscript.

References

- [1] McMahon, D. (2024, June 21). The benefits of using restaurant apps for Customer Self-Ordering. Altametrics. <https://altametrics.com/topics/the-benefits-of-using-restaurant-apps-for-customer-self-ordering/>
- [2] C.-C. Wong, L.-Y. Chong, S.-C. Chong, and C.-Y. Law, "QR Food Ordering System with Data Analytics," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 249–272, Sep. 2023. <https://doi.org/10.33093/jiwe.2023.2.2.18>
- [3] A. Tahwa, "Design of a Satellite Tracking Mobile App Using Flutter and Android Studio IDE," M.S. thesis, University of South Florida, 2024. <https://www.proquest.com/docview/3089117316>
- [4] Jones, B. L. (2021, December 13). *Introduction to Firebase*. Database Journal. <https://www.databasejournal.com/features/introduction-to-firebase/>

- [5] P. R. Saraf, "A Review on Firebase (Backend as A Service) for Mobile Application Development," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 1, pp. 967–971, Jan. 2022. <https://doi.org/10.22214/ijraset.2022.39958>.
- [6] V. Yakovyna and B. Uhrynovskyi, "Aging of Native and Flutter Applications in Android OS in Various Usage Scenarios," *IEEE Xplore*, Sep. 01, 2021. <https://ieeexplore.ieee.org/abstract/document/9648777>
- [7] F. Ferdianto, W. Witarsyah, W. Theodore, A. Avrialdo, and F. Naufal, "ANIMAL RANGER WEBSITE DESIGN: ANIMAL ADOPTION PLATFORM," *Social Economics and Ecology International Journal (SEEIJ)*, vol. 7, no. 2, Nov. 2023, <https://doi.org/10.21512/seeij.v7i2.10497>.
- [8] A. Kumar, A. K. Shah, S. Suman, and G. Pandey, "GoRide: A Modern Ride-Hailing App Solution," Apr. 2025, <https://doi.org/10.36948/ijfmr.2025.v07i02.40575>.
- [9] McDonald's, "McDonald's Mobile Order & Pay | McDonald's," *Mcdonalds.com*, 2023. <https://www.mcdonalds.com/us/en-us/mobile-order-and-pay.html>
- [10] KFC, "KFC Malaysia | Now available for Delivery and Self Collect!," *Kfc.com.my*, 2025. <https://kfc.com.my/>
- [11] L. Holding, "Tealive for Android," *Softonic*, 2024. <https://tealive.en.softonic.com/android>
- [12] Gurung, G., Shah, R., & Jaiswal, D. P. (2020). Software Development Life Cycle Models-A Comparative Study. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 30–37. <https://doi.org/10.32628/>

Appendix A: General System Architecture



Appendix B: Activity Diagram for Administrator

