Applied Information Technology And Computer Science Vol. 4 No. 2 (2023) 1539-1557

AITCS

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

Rosalinda Food Ordering Mobile Application

Muhammad Zairi Yasar Diri[1], Nurezayana Zainal[1*],

[1]Faculty of Computer Science & Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2023.04.02.087
Received 23 June 2023; Accepted 10 November 2023; Available online 30 November 2023

Abstract: Rosalinda Food Ordering Mobile Application is an innovative solution developed to address the frustrations caused by traditional methods of ordering and serving customers at Restoran Rosalinda. Issues such as unreadable handwritten orders, miscommunication between customers and staff, and unnotified cook times are resolved through this online food ordering system. With just a few clicks, customers can place orders and make quick payments, resulting in faster service compared to long wait times during peak hours. Developed using the Prototyping Model methodology and built with Flutter technologies and Dart language, the user-friendly application improves the overall customer experience while reducing employee workload. By streamlining the ordering process and increasing efficiency, the application helps the restaurant save costs and deliver better service to its customers.

Keywords: Food Ordering System, Mobile Application, Flutter and Dart technology.

1. Introduction

Mobile food ordering application can be defined as mobile application that smartphone users download and use as an innovative and convenient channel to access restaurants, view food menus, place food orders, and make payments without any physical interaction with restaurant staff [1]. Customers can add items from an online menu to their carts, proceed to checkout, select a payment option, and place their orders. Once an order was placed, the system immediately registers it on the back end and notifies the restaurant staff to start preparing the order. This process is usually much faster than a traditional dine-in point of sale, where customers may have to wait for their meals for a longer period of time, especially during peak hours.

Restoran Rosalinda is a well-known restaurant for its delicious tomyam dishes in Parit Raja. The restaurant is located at Taman Universiti, Parit Raja and is easy to find, greeted by a beautiful setting with colorful lamps. It is a great hangout spot for friends, providing good food options at affordable prices. It becomes crowded during dinner time, between 8:00 pm - 10:00 pm. Due to its location, the restaurant is often visited by students from Universiti Tun Hussein Onn Malaysia (UTHM). However, the restaurant still uses a traditional method of taking orders which is manual, by writing it down on a piece of paper and passing it to the cook. This may cause some inconvenience to customers as they have

Corresponding author: nurezayana@uthm.edu.my

to wait for around 50 minutes during peak hours to get their food. To overcome this problem, a mobile application which is called Rosalinda Food Ordering Mobile Application, was developed to make the ordering process more efficient and reduce the waiting time.

The main objectives for this project are to study, analyze and design an online food ordering system using an object-oriented approach, to develop an online food ordering system by using prototyping development model and to evaluate the developed system by using alpha and beta testing. The scope of the project is to create a fully operational online food ordering system that will be used by Restoran Rosalinda and will be easily accessible to both customers and employees.

2. Related Work

2.1 Food Ordering System

The Food and Beverage (F&B) industry is a vital sector in which to employ modern technologies and automation. Food ordering systems, specifically, play a critical role in the transformation of a product's physical structure from disarray to structural order. Food services refer to the activities of food service to customers, whether they are served while seated or serve themselves from a display of items, whether they eat the prepared meals on the premises, take them out or have them delivered [2]. Food ordering can be classified into two methods: manual food ordering systems or online food ordering systems.

2.2 Mobile Application Development

Nowadays, many systems are shifting towards mobile application-based platforms, owing to the ease of use and convenience that they provide. This has made mobile applications a prime focus for system developers. Same goes for Rosalinda Food Ordering Mobile Application, as it appeared to its name, it’s a mobile application-based system allowing customer to order via their mobile devices. Mobile applications consist of software/set of programs that runs on a mobile device and perform certain tasks for the user [3]. Mobile applications may be novel in some countries, but it has rapidly grown as a major sector in the global Information and Communication Technology industry. Their ease of use, user-friendly interface, affordability, cross-platform availability and compatibility even with low-cost and entry-level mobile phones have made mobile applications an attractive focus for developers worldwide. Because

2.3 Flutter Technologies

Rosalinda Food Ordering Mobile Application was developed using the Flutter technologies with Dart as the programming language. Flutter is an open-source mobile application development framework created by Google in 2017 [4]. It is widely used by developers to create applications for both Android and iOS environments. The Flutter framework utilizes the Dart language for its syntax, making it an ideal choice for application development as it allows developers to create applications for both Android and iOS platforms using a single codebase. In addition, Flutter is a popular choice among developers because it allows them to easily construct interfaces or front-end elements without having to code the entire application.

2.4 Study of Existing Related System

Currently, there are numerous food ordering systems available in the market. Conducting a study would provide a more comprehensive understanding of the different types of systems and serve as a valuable guide for the development process. For this review, three food ordering systems were chosen: McDonald's, KFC Malaysia, and The Chicken Rice Shop.

Table 1: System comparison

	McDonald's	KFC Malaysia	The Chicken Rice Shop	Rosalinda Food Ordering Mobile Application
Type of System	Application	Application	Web-based	Application
Sign Up Module	√	√	√	√
Login Module	√	√	√	√
Manage User Profile Module	√	√	X	√
Menu Module	√	√	√	√
Cart Module	√	√	√	√
Payment Module	√	√	√	√
Order Notification Module	√	√	√	√
Product Module	X	X	X	√
Order report Module	X	X	X	√
Sales report Module	X	X	X	√

As demonstrated in Table 1, all of the systems can be compared based on various features such as the Sign Up module, Login Module, Manage User Profile module, Menu module, Cart module, Payment module, Order Notification module, Product module, Order report module, and Sales report module. The results of these comparisons reveal that all four systems have similarities in having Sign Up module, Login Module, Menu module, Cart module, Payment module and Order Notification module. However, it's notable that only The Chicken Rice Shop system does not have the Manage User Profile module, while the other three systems do have the Manage User Profile module. Furthermore, the proposed system, Rosalinda Food Ordering System, is the only one with Product module, Order report module and Admin module.

3. Methodology

Methodology refers to the overarching strategy and rationale for studying research problems [5]. Methodology refers to the systematic and structured approach that includes the specific methods and procedures used to study a problem or a phenomenon, as well as the reasoning behind why these methods were chosen. This approach is essential to ensure that the research is conducted in a rigorous, unbiased and reliable manner. In software development, methodology also refers to the approach or framework that is utilized to guide the creation, design and development of a software application, and to ensure that it is aligned with the goals and objectives of the software project. Figure 1 shows the phases in the prototyping model methodology that will be used to develop the system.

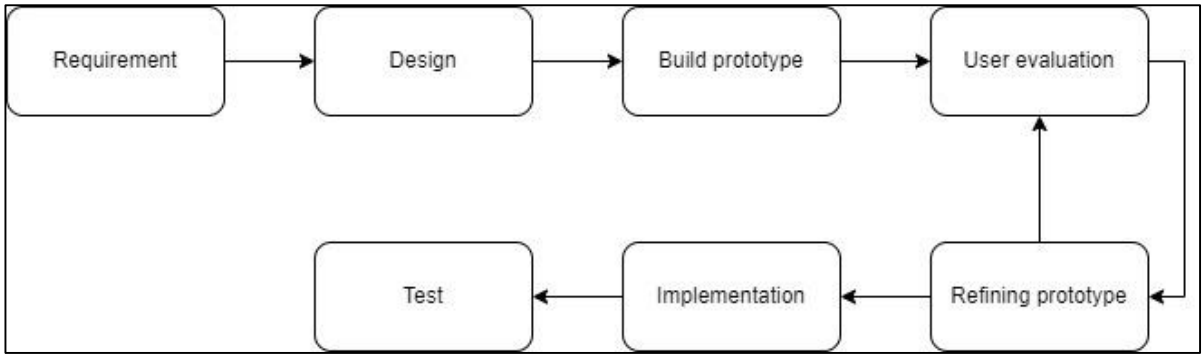

Figure 1: Phases in Prototyping Model

3.1 Requirement phase

The development process starts with the requirement phase, which is crucial in ensuring that the final system meets the needs and expectations of all stakeholders. In this phase, all necessary information and data is gathered to develop the system, including a detailed analysis of all requirements determined through interviews with stakeholders. During the interview, the person in charge at that time, Mr. Zack told all the details about their business and the requirement corresponded to the project. The information gathered during this phase is used to create a clear and comprehensive understanding of the project's goals and objectives, as well as the specific needs and constraints of the stakeholders. This phase is critical to the success of the project, as it sets the foundation for the rest of the development process and helps to ensure that the final system meets the needs of all stakeholders.

3.1.1 Functional Requirements

Functional requirements involve describing the services that the system should provide and specifying how the system will respond to its inputs [6]. They guide design and development and are used for testing and validation to ensure the system meets intended goals, it can be expressed in natural language or represented using flowcharts, block diagrams, or other modeling languages. Table 2 shows the functional requirements for Rosalinda Food Ordering Mobile Application.

Table 2: Functional Requirements

Modules	Functionalities
1. Sign up	System allows new customer to register account to use the system before login into the system.
2. Login	System allows user to login using their own registered account. The system will be redirected to homepage after login successful.
3. Manage User Profile	System allows customer to see and edit their personal information in the system.
4. Menu	System allows customers to browse product through menu page. Each of the product will have price displayed.
5. Cart	System allow customer to edit, add and delete orders before entering the payment page.
6. Payment	System allow customer to pay for the order that has been placed along with total price displayed.

Modules	Functionalities
7. Order Notification	System allow the system to notify customer how much longer the food will be ready to be served to their table. User also can view details of their order summary.
8. Product	System allow the waiter to edit, add and delete the food and drinks available in the menu.
9. Order report	System allow the waiter to view all order list made by customer and filtering it by date.
10. Sales report	System allow manager to view overall report of total sale made on certain date.

3.1.2 Non-Functional Requirements

Non-functional requirements (NFRs) specify the quality attributes of a system such as performance, security, usability and maintainability, they describe how well the system should perform, rather than what it should do (which is defined by functional requirements). Table 3 shows the non-functional requirements for Rosalinda Food Ordering Mobile Application.

Table 3: Non-Functional Requirements

Modules	Functionalities
Usability	System can be used by its intended users to achieve their goals including the ease of learning, efficiency of use, and ease of remembering how to use the system.
Performance	System will be running and how long the time it takes for the system to respond to user requests, such as page load times or query processing, will be measured.
Security	Users must be verified by the system using email and password before can access into the system and all the data must be protected within the security layer of the system.
Scalability	System able to handle increasing numbers of users or workloads without a significant decrease in performance.

3.1.3 User Requirements

User requirements refer to the specific needs, goals, and expectations that users have for a particular product or service [7]. These requirements play a crucial role in guiding the design and development process, ensuring that the final product is tailored to meet the needs and preferences of its intended users. By understanding and incorporating user requirements, organizations can enhance user satisfaction, usability, and overall user experience. User requirements can encompass a wide range of factors, including functionality, usability, performance, security, compatibility, accessibility, and aesthetic appeal. Table 4 shows the user requirements for Rosalinda Food Ordering Mobile Application.

Table 4: User Requirement

No.	User Requirements
1	Customer shall be able to register new account.
2	Customer, Waiter, Cooker and Manager shall be able to login.
3	Customer shall be able to edit personal details in profile.
4	Customer shall be able to view menu.
5	Customer shall be able to add product into the cart.
6	Customer shall be able to checkout to proceed to payment.
7	Customer shall be able to view order notification.
8	Waiter and Manager shall be able to edit product.
9	Waiter and Manager shall be able to view order report.
10	Manager shall be able to view sales report.

3.2 Design phase

Next, the design phase. This phase involves creating a simple yet effective system design, providing an overview of its appearance and functionality based on the gathered requirements. The process begins with understanding user needs and creating a Unified Modelling Language (UML) diagram, including a use case diagram, sequence diagram, activity diagram and class diagram for Rosalinda Food Ordering Mobile Application. UML is a fundamental graphical modelling language that aims to describe, visualize, build, and document virtual systems devices that are based on objects [8]. Then, a wireframe layout is generated by using Figma to visualize the interface structure. This wireframe is transformed into a detailed interface prototype, capturing the system's functionality, layout, and user flow. The phase concludes with the creation of a detailed interactive prototype for user testing and feedback, guiding the implementation phase to ensure alignment with the initial design.

3.2.1 Use Case Diagram

A use case diagram is created during design phase to illustrate the general function and components of the system. Plus, it helps to define, clarify, and organize the requirements for the Rosalinda Food Ordering Mobile Application system. Figure 2 shows the use case diagram for Rosalinda Food Ordering Mobile Application. There are in total 9 use case, which is Register user, Login account, Manage user profile, Add to cart, Make payment, View order notification, Edit product, View order report and View sales report, designed to show the activities and actions that can be performed by 4 different types of actors which will be the user for Rosalinda Food Ordering Mobile Application.

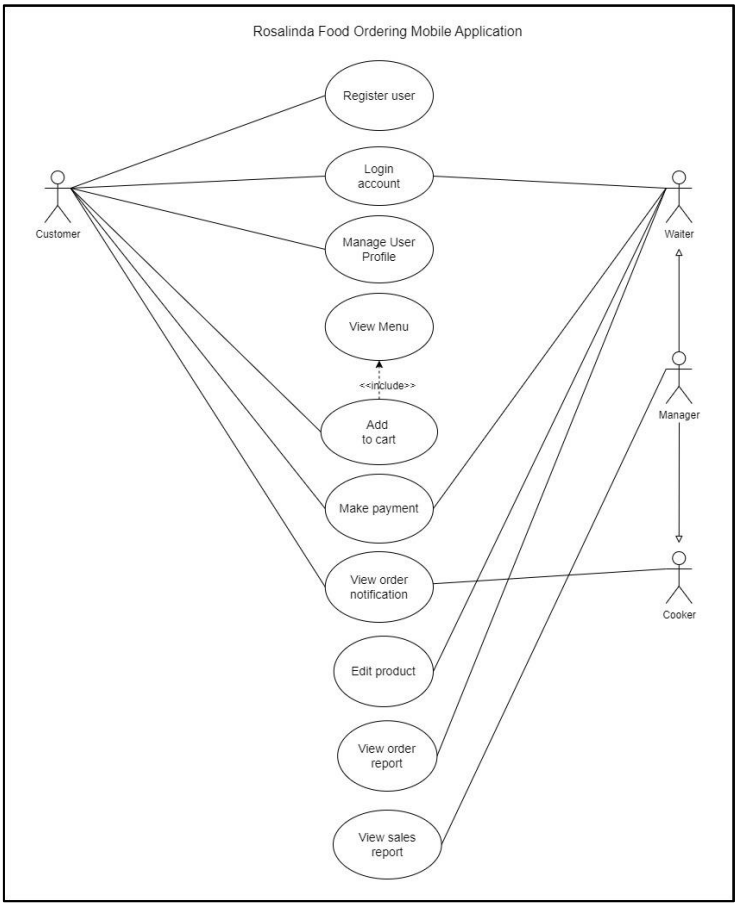

Figure 2: Use Case Diagram

3.2.2 Class Diagram

A class diagram in UML illustrates the classes, attributes, and relationships within a system to showcase its structure, it depicts the static structure of a system, can be used for designing and documenting an already existing one. Figure 3 in Appendix A shows the class diagram for Rosalinda Food Ordering Mobile Application.

3.2.3 Database Schema

A database schema is an illustration of data storage, it defines organization and relationships between tables in a database, it is designed in advance by developers to specify components and their connections, examples can be extracted from the class diagram.

I. Customer (email, password, userid, name, contactNumber)
II. Waiter (waiterID)
III. Cooker (cookerID)
IV. Manager (managerId)
V. Order (orderID, custId, custName, orderDate, orderStatus, orderSummary)
VI. Product (productId, productName, productPrice)
VII. Cart (cartId, productId, productQuantity, cartTotalPrice)
VIII. Sales report (salesReport, date, paymentDetails, orderId)
IX. Payment (orderId, paymentDetails)

3.3 Build Prototype phase

The build prototype phase marks the start of the development of the actual prototype. Based on the data and information collected in the previous phases, a real working prototype is developed with more features and functions added in. The detailed design and interface prototype are used as a guide to ensure that the prototype aligns with the stakeholders' requirements and goals. This phase is critical as it allows the team to test the various functionalities and usability of the system and make any necessary adjustments before moving on to the final implementation.

3.3.1 Interface Design

The art of interface design involves crafting visually stunning and intuitive user interfaces for software and automated devices. When it comes to design, the user interface stands as one element within the realm of future computer applications, serving as a design object. Yet, in practical use, the user interface plays a vital role in determining the visual presentation of the computer application as experienced by its users [9]. Through careful consideration of aesthetics and usability, interfaces were created, that are not only functional but also a pleasure to use. The interface design of this application was designed using Figma software. Figure 4-8 in Appendix B shows the interface for respective modules as stated in the labels.

3.4 Evaluation phase

The next phase is the user evaluation phase, where an evaluation of the system is done by having real users, which the stakeholder tests it and provide feedback. This process helps to identify the strengths and weaknesses of the system and aid in further development and improvements. For this evaluation, only the stakeholder which is the manager and two employees were involved. They were asked to complete a series of tasks and provide their feedback and suggestions regarding the usability and functionality of the system. The collected data is carefully analysed and any issues or areas of improvement are addressed. For starter, Mr Zack ask to remove the booking feature in the mobile application, as it was unnecessary. This phase is crucial in ensuring that the final product is user-friendly and meets the needs of the stakeholders.

3.5 Refining phase

The Refining Prototype phase involves continuous improvement of the prototype based on user feedback and collaboration with stakeholders to align it with their needs. A meet up closely collaborates with stakeholders to ensure that the prototype aligns with their needs and addresses any issues identified during the testing phase. It may involve multiple rounds of feedback and improvements until user satisfaction is achieved, then serves as foundation for final product development.

3.6 Implementation phase

Next phase of the project is the implementation phase, where the implementation involves developing the backend components of the application, such as server-side logic, database management, and integration with external system. This includes setting up user authentication, add to cart data, data fetches from Firestore, etc.

3.6.1 Code segments

This section shows the key elements that are important within the implementation of Rosalinda Food Ordering Mobile Application.

```
Future<void> loginAuth() async {
  try {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: emailController.text,
      password: passwordController.text,
    );
    Navigator.of(context).pushReplacement(
        MaterialPageRoute(builder: (context) => BottomNavigation()));
  } on FirebaseAuthException catch (e) {
    if (e.code == 'user-not-found') {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('No user found for that email.'),
        ), // SnackBar
      );
    } else if (e.code == 'wrong-password') {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('Wrong password provided for that user.'),
        ), // SnackBar
      );
      setState(() {
        loading = false;
      });
    }
    setState(() {
      loading = false;
    });
  }
}
```

Figure 9: User authentication code segment

Figure 4 shows the code segment of authentication elements in the Rosalinda Food Ordering Mobile Application. This code segment is an asynchronous function written in Dart for handling user authentication in a Flutter application using Firebase Authentication. The function attempts to sign in the user with the provided email and password using the signInWithEmailAndPassword() method. If the login is successful, the user is redirected to the Homepage screen by calling the BottomNavigation class. In case of exceptions, such as a user not found or a wrong password, appropriate snack bars are displayed to notify the user.

```
List<CartModel> cartList = [];
List<CartModel> newCartList = [];
late CartModel cartModel;
void addToCart({
  required String image,
  required String name,
  required int price,
  required int quantity,
  required String productId,
}) {
  cartModel = CartModel(
    image: image,
    name: name,
    price: price,
    quantity: quantity,
    productId: productId,
  );
  newCartList.add(cartModel);
  cartList = newCartList;
}

get throwCartList {
  return cartList;
}

int totalPrice() {
  int total = 0;
  cartList.forEach((element) {
    total += element.price * element.quantity;
  });
  return total;
}

void deleteProductFunction(String productId) {
  FirebaseFirestore.instance
      .collection("cart")
      .doc(FirebaseAuth.instance.currentUser!.uid)
      .collection("userCart")
      .doc(productId)
      .delete();

  cartList.removeWhere((item) => item.productId == productId);
  notifyListeners();
}
```

Figure 10: Cart code segment

Figure 5 shows the code segment of shopping cart elements in the Rosalinda Food Ordering Mobile Application. This code segment implements a shopping cart functionality. It includes the addToCart method for adding items to the cart by creating a CartModel object and appending it to the cartList. The throwCartList getter returns the current cart items. The totalPrice function calculates the total price of items in the cart based on their prices and quantities. The deleteProductFunction deletes a product from the cart by removing it from the Firestore database and the cartList.

```
List<FoodModel> foodModelList = [];
late FoodModel foodModel;

Future<void> getFoodList() async {
  List<FoodModel> newfoodModelList = [];
  QuerySnapshot querySnapshot =
      await FirebaseFirestore.instance.collection('foods').get();
  querySnapshot.docs.forEach((element) {
    Map<String, dynamic>? data = element.data() as Map<String, dynamic>?;
    if (data != null) {
      foodModel = FoodModel(
        image: data['image'],
        name: data['name'],
        price: data['price'],
        description: data['description'],
        productId: data['productId'],
      );
      print(foodModel.name);
      newfoodModelList.add(foodModel);
    }
  });
  foodModelList = newfoodModelList;
}

get throwFoodModelList {
  return foodModelList;
}
```

Figure 11: Fetch data code segment

Figure 5 shows the code segment for fetches a list of food items from Firestore in the Rosalinda Food Ordering Mobile Application. The getFoodList function retrieves documents from the "foods" collection, extracts the data from each document, and creates FoodModel objects based on the extracted values. These objects are then added to the newfoodModelList. The foodModelList is subsequently updated with the contents of newfoodModelList. The throwFoodModelList getter returns the current list of food items stored in foodModelList.

3.7 Test phase

A thorough testing is conducted to ensure the system meets requirements and standards. This testing will include functional testing, performance testing, security testing, and usability testing. Once stable, it is launched and made available to users. Regular maintenance including error detection and fix, performance monitoring, and security updates, is important to reduce downtime, ensure smooth operation and longevity, and minimize future issues.

3.7.1 Alpha Testing Result

Alpha testing is an important phase in software development that involves evaluating the system's functionality and performance before its release to a wider audience. The test was conducted by the developer by running the code to check any error and misfunction in system. In this discussion, we will focus on the alpha testing results for the functional and non-functional requirements of the system. Results were shown in table 5 and Table 6.

Table 5: Testing results for functional requirement

Modules	Testing Result
1. Sign up	Testing revealed that the registration process was smooth and allowed users to create their accounts without any major issues.
2. Login	Testing demonstrated that users were able to log in successfully, and upon successful login, the system appropriately redirected them to the homepage.
3. Manage User Profile	Tests confirmed that users could access and modify their profile details without encountering any significant problems.
4. Menu	Tests indicated that customers could navigate through the menu effectively, and the prices were accurately displayed.
5. Cart	Tests revealed that customers could manipulate their cart items as intended, making changes to their orders seamlessly.
6. Payment	Testing demonstrated that the payment process was functioning correctly, and customers could view the total price and complete the payment smoothly.
7. Order Notification	Testing confirmed that the notification feature worked effectively, providing customers with timely updates and access to order details.
8. Product	Tests revealed the system allowed employees to edit, add and delete without encountering any major issues.
9. Order report	Testing demonstrated that employees could access the order reports and filtering it by date.
10. Sales report	Testing confirmed that the manager could access the sales reports.

Table 5 summarizes the testing results for the functional requirements of the Rosalinda Food Ordering Mobile Application. The sign-up and login processes worked smoothly, allowing users to create accounts and access the system without issues. User profile management, menu navigation, cart manipulation, and payment processes all functioned properly. The order notification feature provided timely updates, while employees could edit products and access order and sales reports without major issues.

Table 6: Testing results for non-functional requirement

Modules	Testing Result
Usability	Testing results indicated that the user interface well-organized and easy navigate through different section of the system.
Performance	Test shows response times for typical operations, such as loading pages and retrieving data, were within acceptable limits.
Security	Test results revealed no major security vulnerabilities and user data was adequately protected through encryption and access control.
Scalability	Test shows as user loads increased, the system demonstrated the capability to handle the additional workload without significant performance degradation.

Table 6 presents the testing results for the non-functional requirements of the system. The user interface showed good usability, improving the overall user experience. Performance testing indicated acceptable response times for loading pages and retrieving data. The system exhibited adequate security measures, protecting user data through encryption and access control. Scalability testing demonstrated its capability to handle increased user loads without significant performance issues.

4 Result and Discussion

This chapter provides a concise overview of the User Acceptance Test results for each module, indicating the level of feedback value received. It can serve as a visual representation of the overall acceptance of the system's modules by the users.

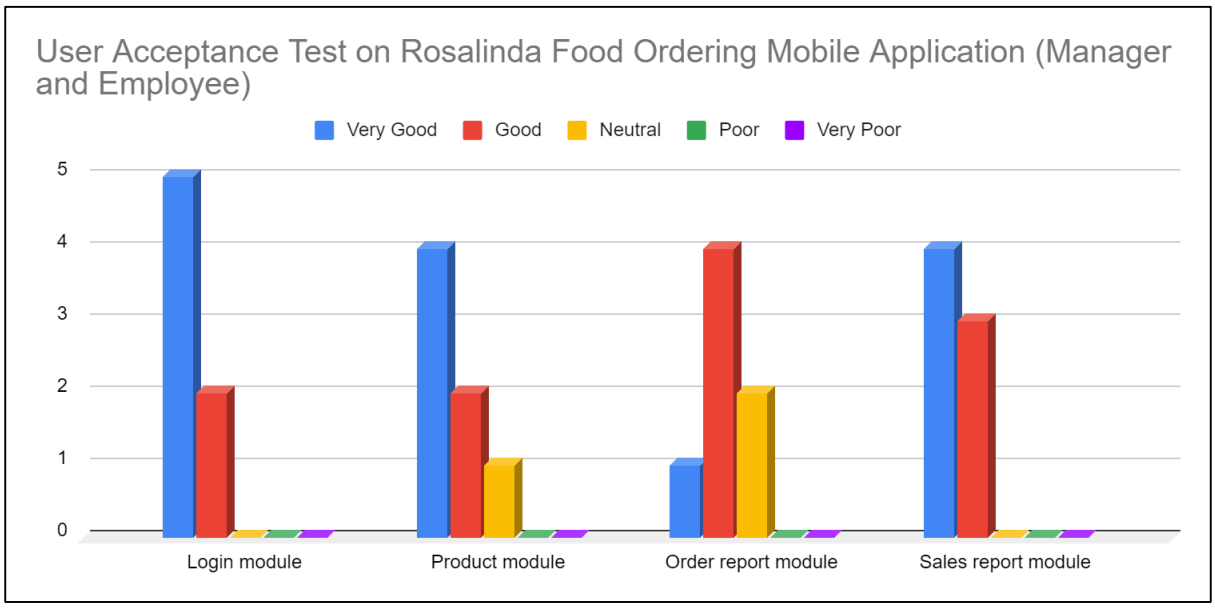

Figure 12: User acceptance test on system functionalities for manager and employee

Figure 4 shows the bar chart of user acceptance test on system functionalities for manager and employee of Restoran Rosalinda. There is a total of 7 respondent were involved in the testing process, the manager and 6 employees. Based on the bar chart, 5 out of 7 respondent responds “Very good” for the login module. For the product module, majority with 4 out of 7 responds “Very good”. Next for

order report module, majority with 4 out of 7 responds "Good". Lastly, for sales report module, majority with 4 out of 7 responds "Very good".

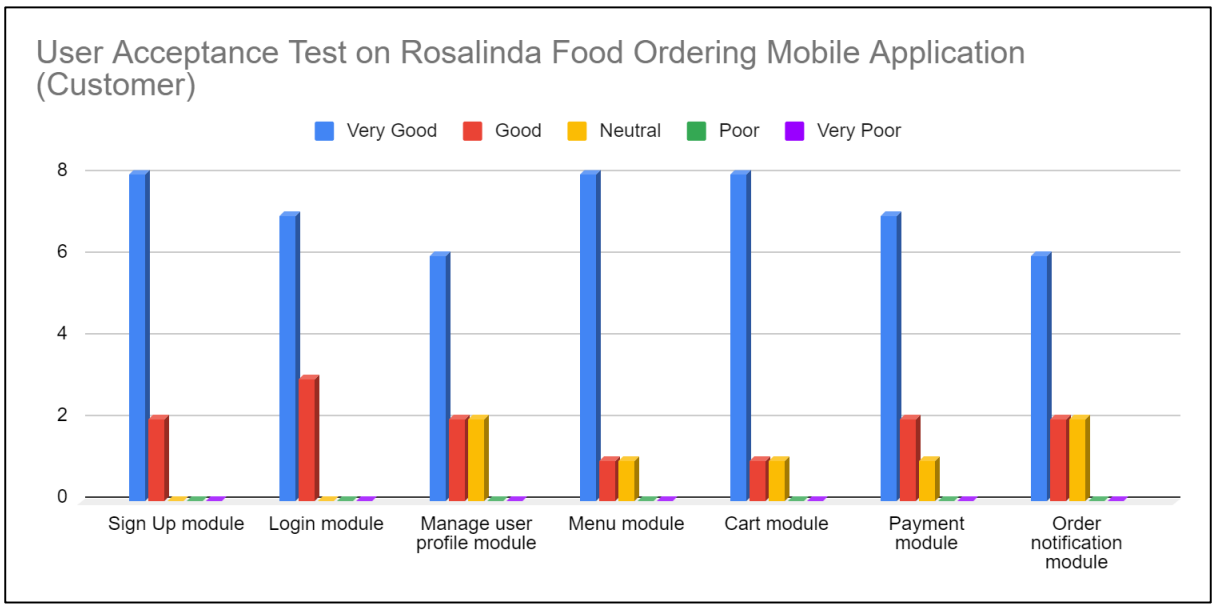

Figure 13: User acceptance test on system functionalities for customer

Figure 5 shows the bar chart of user acceptance test on system functionalities for customer of Restoran Rosalinda. There is a total of 10 respondent were involved in the testing process and they were selected randomly. Based on the bar chart, 8 out of 10 respondent responds "Very good" for the sign up module. For the login module, majority with 7 out of 10 responds "Very good". Next for manage user profile module, majority with 6 out of 10 responds "Very good". For the menu module and cart module, both 8 out of 10 responds "very good". In addition, the payment module, 7 out of 10 responds "Very good". Lastly, for order notification module, majority with 4 out of 7 responds "Very good".

Based on the results of the User Acceptance Test, it has been proven that the Rosalinda Food Ordering Mobile Application has successfully passed the functional testing phase. Majority of users agreed that the app fulfils the functional requirements as specified. Additionally, the app has also met all the security testing requirements, ensuring that the necessary security measures are in place.

5 Conclusion

To conclude, the project successfully achieved its overall objective of developing a functional online food ordering mobile application. The project involved the analysis of existing systems and utilized an object-oriented approach during the design phase, resulting in a modular and flexible system structure. The achievement of this objective has contributed to the advancement of knowledge in the domain of online food ordering systems, showcasing the successful implementation of methodologies used which is the prototyping development model. The implemented system has proven to be user-centric, aligning closely with user requirements and delivering benefits such as streamlined ordering processes, improved customer experience, and enhanced operational efficiency for businesses.

For future work, several recommendations can be considered to further improve the Rosalinda Food Ordering Mobile Application. Firstly, implementing functionalities to personalize user profiles and preferences, such as saved order history and customized food/drink options, would enhance the user experience by providing a more tailored and convenient ordering process. Secondly, optimizing the search functionality, categorization, and filtering options would improve the browsing experience, making it easier for users to find their desired menu items. Next, enhancing the order tracking feature to provide more accurate real-time updates on order status would increase transparency and keep customers informed throughout the delivery process. Lastly, integrating an automated inventory

management system to track ingredient availability and facilitate automatic updates when items run low or go out of stock would optimize operational efficiency and help ensure a seamless ordering experience for customers.

Acknowledgement

The authors would also like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM) for its support and encouragement throughout this project.

References

[1] Alalwan, A. A. (2020). Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse. International Journal of Information Management, 50, 28-44.

[2] ANNUAL ECONOMIC SURVEY 2022. (2021). Retrieved from https://www.dosm.gov.my/v1/uploads/files/AES_2022/Panduan-FnB-AES2022-BI.pdf.

[3] Islam, R., Islam, R., & Mazumder, T. (2010). Mobile application and its global impact. International Journal of Engineering & Technology, 10(6), 72-78.

[4] Wasilewski, K., & Zabierowski, W. (2021). A comparison of java, flutter and kotlin/native technologies for sensor data-driven applications. Sensors, 21(10), 3324.

[5] Tashakkori, A., & Teddlie, C. (Eds.). (2018). Handbook of mixed methods in social and behavioral research (3rd ed.). Thousand Oaks, CA: Sage Publications.

[6] Laplante, P. A., & Kassab, M. H. (2022). Requirements engineering for software and systems. CRC press.

[7] Zhou, J. (2018). "A Study on Gathering User Requirements for Software Development", Journal of Computer Science and Technology, vol. 33, no. 1, pp. 61-69

[8] Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021, March). UML diagrams in software engineering research: a systematic literature review. In Proceedings (Vol. 74, No. 1, p. 13). MDPI.

[9] Bodker, S. (2021). Through the interface: A human activity approach to user interface design. CRC Press.

Appendix A

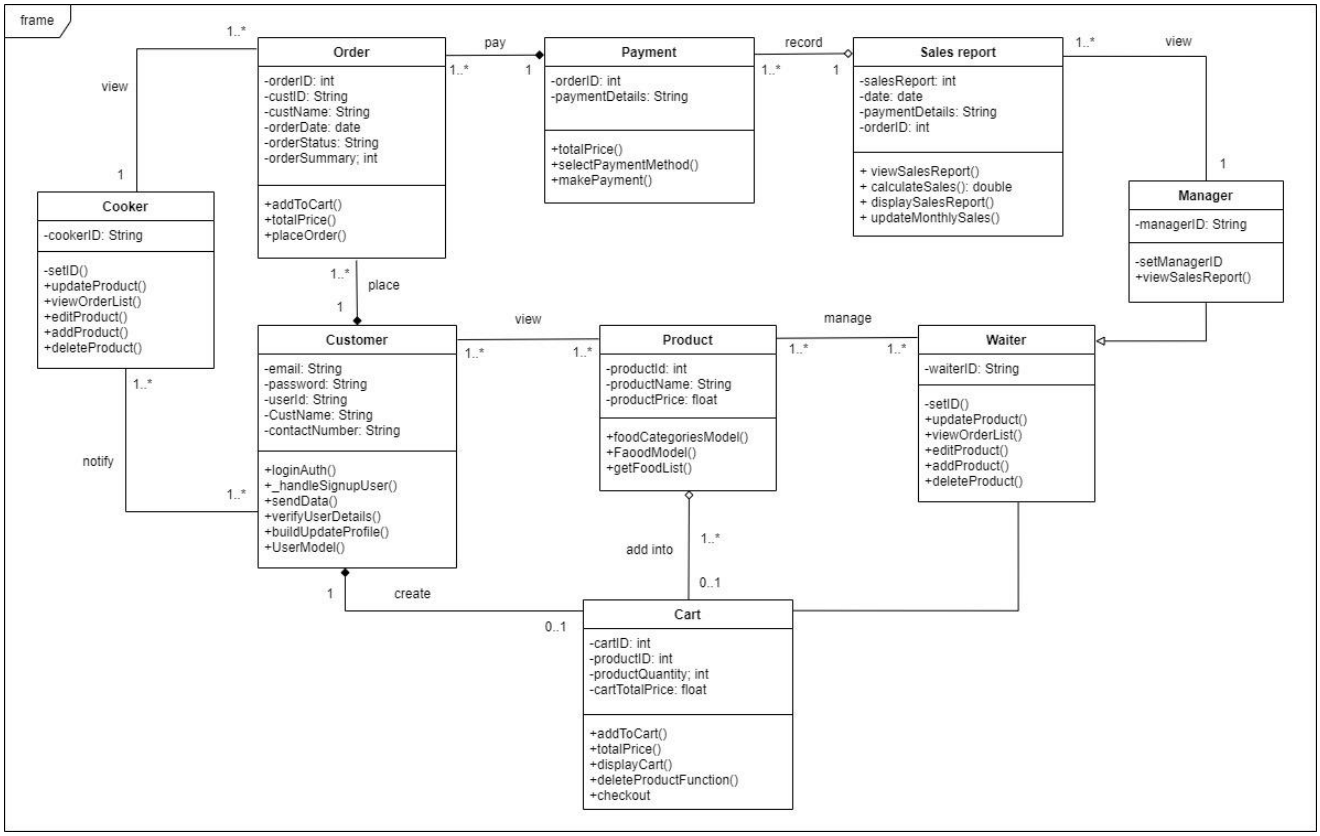

Figure 3: Class Diagram

Appendix B

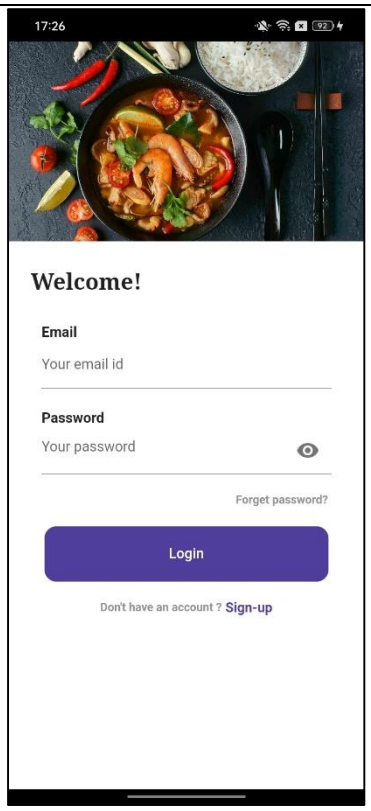

Figure 4(a): Login page

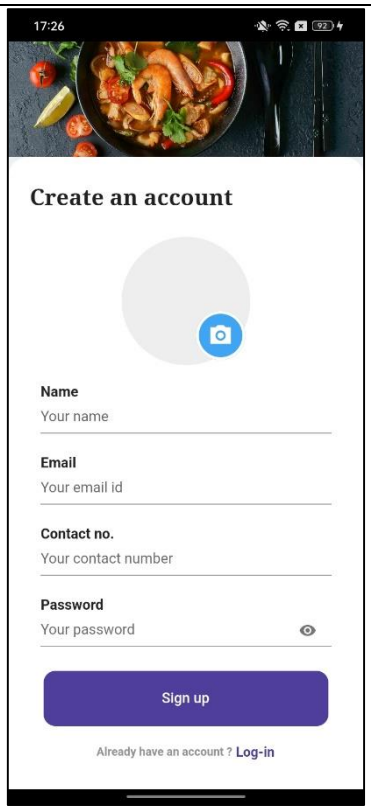

Figure 4(b): Sign Up page

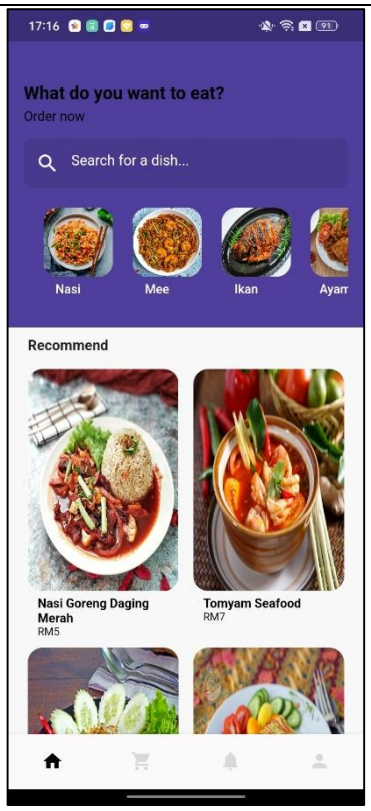

Figure 4(c): Home page

Figure 4: User Interface for Customer

Figure 5(a): Mee Categories page

Figure 5(b): Ayam Categories page

Figure 5(c): Air Categories page

Figure 5: User Interface for Customer (cont)

Figure 6(a): Ikan Categories page

Figure 6(b): Nasi Categories page

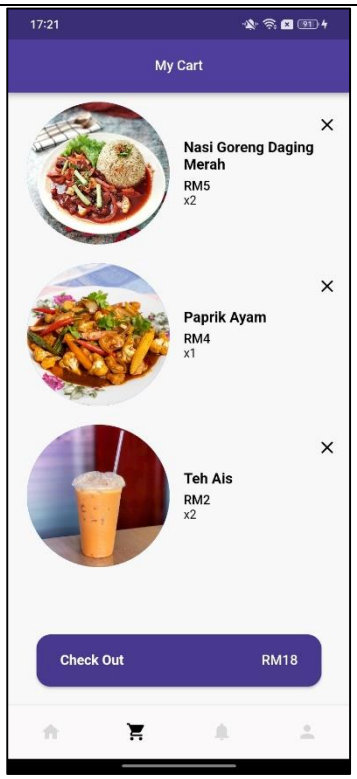

Figure 6(c): Cart Categories page

Figure 6: User Interface for Customer (cont)

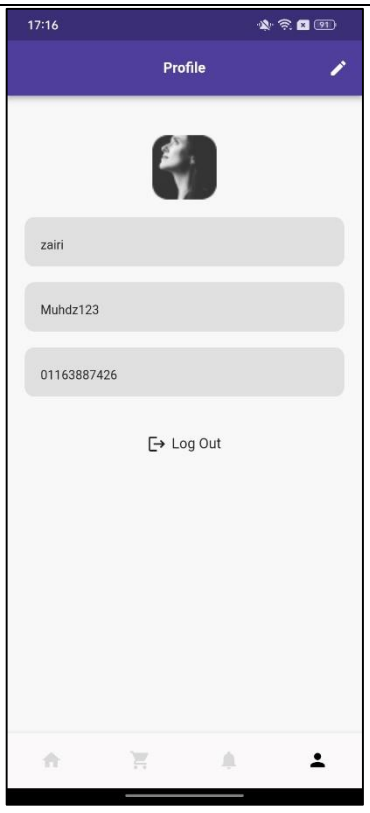

Figure 7(a): Profile page

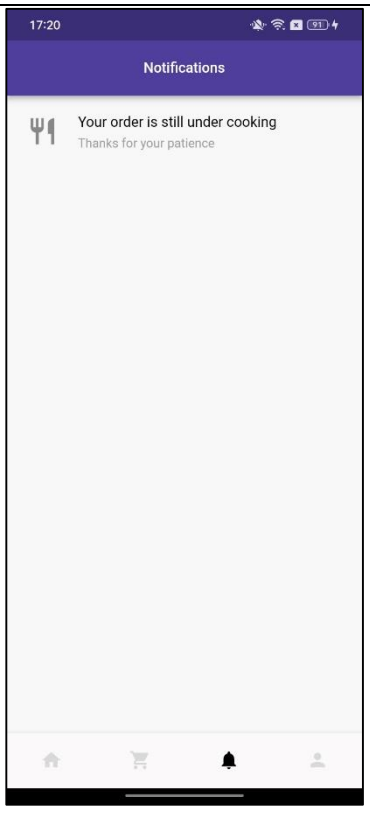

Figure 7(b): Order Notification page

Figure 7: User Interface for Customer (cont)

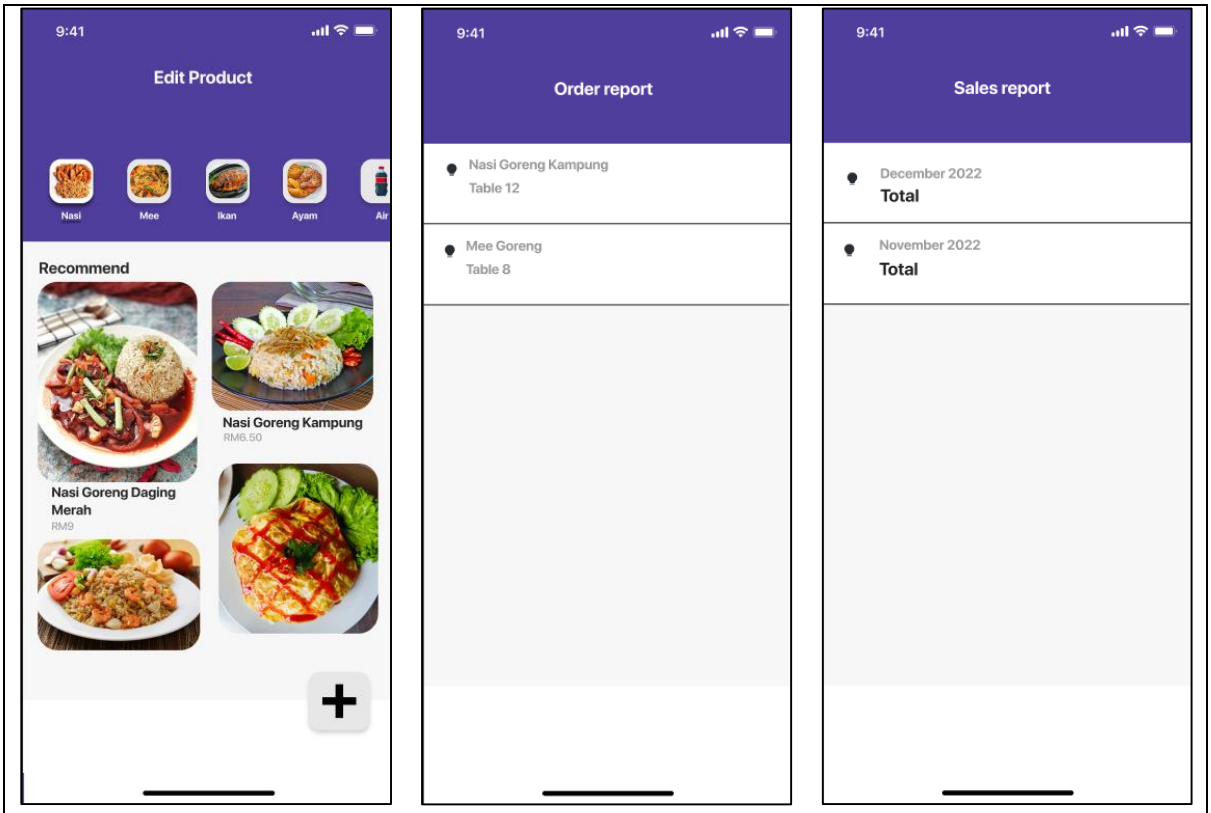

Figure 8(a): Edit Product page **Figure 8(b): Order Report page** **Figure 8(c): Sales Report page**

Figure 8: User Interface for Manager, Waiter and Cooker