

Fitri Tomyam Seafood Menu Order Android Mobile Application

Mazuda Mohamad Zaki, Nurezayana Zainal*

Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2021.01.01.009>

Received 10 March 2021; Accepted 22 April 2021; Available online 31 May 2021

Abstract: Fitri Tomyam Seafood Menu Order Android Mobile Application is an android based mobile application that was developed to aid the business process in the aspect of delivery orders for Fitri Tomyam Seafood restaurant. The existing manual process of taking delivery orders where delivery orders are taken via WhatsApp platform is not an efficient method in managing delivery orders and provides poor customer service. Therefore, to build this project, an object oriented software development method was used in this project. Whereas, this application was developed using Android Studio and Firebase database. Overall, Fitri Tomyam Seafood Menu Order Android Mobile Application enables this restaurant to provide better customer service in delivery orders.

Keywords: Delivery, Mobile, Android

1. Introduction

Fitri Tomyam Seafood is a food restaurant that specializes in Thailand cooking. It is located at Lorong Haji Kosnee, Kampung Parit Jelutong, 86400 Parit Raja, Johor. This restaurant serves a variety of foods at affordable prices. Due to its strategic location, this restaurant provides delivery orders for customers living nearby the area. However, the process of delivery orders does not facilitate good customer service. As such, delivery orders are taken via WhatsApp platform and more time is spent to eventually just order. This eventually arises several problems in handling delivery orders. Therefore, this project attempts to design an android mobile application that can help Fitri Tomyam Seafood restaurant to reduce manual workings in order delivering aspects, develop Fitri Tomyam Seafood Menu Order Android Mobile Application and to test Fitri Tomyam Seafood Menu Order Android Mobile using functional testing and user acceptance testing.

1.1 Problem Statements

There are several difficulties and lack in the process of order delivery at the restaurant. Firstly, the orders are done via WhatsApp platform. The customers are not provided with the menu directly and tend to ask the types of dishes provided. This gives them an uneasy imagination on what types of foods that are sold in this restaurant. Staff is likely to be repeating the same answers but to different customers relating this type of question.

Besides that, the customer has to spend a little longer when ordering for delivery. This is because customers have to wait for their message of orders to be replied for confirmation of orders to be placed. Sometimes, customers have to wait for almost 1 hour for their order message to be seen by the staff that is available. This is one of the inefficient processes of delivery orders faced by the restaurant.

Apart from that, due to the unsystematic method in handling delivery order. Customers do not know their delivery order status and orders for delivery are not taken in sequence, as in first come first served. They are often jumbled up. It is caused by the messages that pop up in the recent received sequence. Eventually, this roots to one of the causes towards customers' dissatisfaction in delivery service. Table 1 shows the scope module of the system.

Table 1: Scope module of the system

Module	Function	User
Registration	Customer can register a new account by filling up the registration form correctly.	Customer
Login	Enables user to login into the system using email and password User can only access if the entered email and password is valid	Owner, staff, customer
View menu	Customer shall be able to view the menu and select the menu item to be ordered. It shall then be added into the cart	Customer
Manage cart	Customer shall be able to add, delete and update items inside their cart before proceeding to check out	Customer
Place delivery order	Customer shall be able to place their delivery order directly using this system. Staff will receive the confirmed delivery orders of the customer.	Customer
View delivery order status	Customer shall be able to view their delivery orders' status	Customer
Manage profile	Customer shall be able to edit their profile for the purpose of ordering details.	Customer
Manage delivery order	Staff can view and manage the delivery orders received. Staff is able to update each delivery orders' status	Staff
Manage menu items	Staff can insert new menu items. Staff can also update and delete menu items of the menu.	Staff
View report	The owner can view the overall no of delivery orders received days and by hours.	Owner

1.2 Project Significance

The first significant of this project is that the current manual delivery orders will be improved into an android mobile application. This system will potentially help all the stakeholders involved. In the viewpoint of the owner and the staff, this system can make their business process work smoother. They can minimize their normal manual work. They can view customers' orders faster and reduce the risks of making errors while ordering. Besides that, the customers can place delivery orders according to their budget and preferences in terms of the food comfortably by using this system. Lastly, this project can gain customers' trust and satisfaction when using this system.

2. Literature Review

2.1. Android Application

An android application is the software application that runs on the Android platform. Nowadays, the android system in the electronics market is increasingly popular especially in the smartphone

market. Besides that, it provides a convenient hardware platform for developers enabling them to spend less effort in realizing their ideas [1 - 3].

2.2. Manual Food Ordering System

Manual food ordering system usually happens when the waiter of the shop approaches the customers once the customers have been seated. They will then distribute the menu and would be standing aback until the customers have made their decision on what to be ordered. At the moment when customers give their orders, the waiter would be jotting down the selected menu onto a piece of paper. However, when the situation is at the peak hours, where the restaurant is too crowded with the incoming customers, this manual system eventually would cause a hassle and the quality of the service may drop as such not many waiters are available in taking the orders in one moment and the distributing of orders to the customers would not be in sequences. As a result, this manual system may cause customers' dissatisfaction towards the service of the restaurant.

2.3. Online Food Ordering System

Online food ordering system is a way that allows customers to purchase food online without having to go to the restaurant. Since the growth and the usage of technology are rapidly increasing and changing, therefore customers prefer to have things that simplify their lives. The way of ordering through online is as follows; firstly, the customer would place the food ordered through the restaurant website respectively. Once confirmed, the chef would then receive the ordered menu. The food would then be delivered or can be taken from the restaurant based on the selected method of delivery. Some of the advantages of this ordering method include making the ordering process easier where customers can just view the menu and place their order directly without having to go to the restaurant. The con of using this method is that users must have an internet connection.

2.4. Existing System Review

Table 2 shows the comparison of similar existing systems with the proposed system (Fitri Tomyam Seafood Menu Order Android Mobile Application). The existing systems that are selected to be compared are Mc Donald's mobile application, Domino's Pizza mobile application and Marrybrown online ordering system. The results of the comparison are shown in Table 2.

Table 2: Comparison between existing system and proposed system

Feature	Mc Donald's	Domino's Pizza	Marry brown	Proposed System
Android platform	✓	✓	X	✓
Register	✓	✓	✓	✓
View menu	✓	✓	✓	✓
Place delivery order	✓	✓	✓	✓
Login & logout	✓	✓	✓	✓
View order status	✓	✓	X	✓
Manage cart	✓	✓	✓	✓
Cash on delivery	✓	✓	✓	✓

3. Methodology

Object oriented software development is the combination of systems development life cycle and object oriented programming [4]. This method has been used throughout building this project. The object oriented software development life cycle (SDLC) consists of several macro processes which are object-oriented analysis, object-oriented design and object-oriented implementation and object-oriented

testing phase [5]. Object-oriented Software Development lifecycle model can allow one to produce designs that are traceable across requirements, analysis, design, and implementation and testing.

In general, during the planning phase, problem statements and objectives are being identified. During the analysis phase, the system requirements and the user roles are identified. Moving on into the design phase, where the requirements are translated in depth details. Whereas, in the implementation phase, the detailed designs are refined into the system deployment and ensure the satisfaction of the user needs. Finally, in the testing phase, the developed system is tested to ensure the fulfillment of the requirements of the system. The results and discussion section presents data and analysis of the study. This section can be organized based on the stated objectives, the chronological timeline, different case groupings, different experimental configurations, or any logical order as deemed appropriate. Table 3 summarizes the task and output for each object-oriented phase.

Table 3: Task and output for each phase

Phase	Task	Output
Planning	<ul style="list-style-type: none"> • Find an organization. • Identify the problem statement, objectives and scope for the project • Collect requirements from the interview and report it for further guidelines. 	<ul style="list-style-type: none"> • Discussion with the supervisor regarding the proposed title • Interview with the client of the organization. • List of objectives, problem statement, scope, proposal, Gantt Chart
Analysis & Design	<ul style="list-style-type: none"> • Analyze the user requirements and business requirements gathered previously. • Identify the comparison of existing systems. • Identify the methodology used • Identify the system user and the system requirement • Designing user interfaces • Identify the entities and the attributes for the entities • Design the table structure 	<ul style="list-style-type: none"> • Download software to be used • Collect and review the journals and articles • Hardware and software specification • User interface • UML diagrams: use case, sequence diagram, class diagram, flow chart and data dictionary are drawn.
Implementation	<ul style="list-style-type: none"> • Write the coding to build the system. • Create and develop database 	<ul style="list-style-type: none"> • System interface • Application developed using Android Studio. • Develop database using Firebase
Testing	<ul style="list-style-type: none"> • Functional testing • Finding defects of the entire system 	<ul style="list-style-type: none"> • Test cases

4. Analysis and Design

This section explains about the system design that will visualize the situation and the flow of the system that will be developed. The implementation of analysis and design is important to make sure the system developed achieves the user requirements and the objectives. System Requirement Analysis and Unified Modeling Language (UML) Diagram which are Use Case Diagram, Sequence Diagram, Class Diagram and Activity Diagram has been used as the model in providing clearer understandings on the proposed system flow.

4.1. Use Case Diagram

Figure 1 shows the use case diagram of the system. The boundary of the diagram is the name of the system which is Fitri Tomyam Seafood Menu Order Android Mobile Application. There are three stakeholders involved in this system which are the customer, owner and staff. These stakeholder acts as the actor in the use case diagram. Based on each module of the system, the use cases are produced. As an explanation to Figure 1, the customer is associated with particular use cases which are register, login, view menu, manage cart, place delivery order, view delivery order and manage profile. As for staff, the staff is able to login, manage menu items and manage delivery order. Therefore, the staff is associated with their respective use cases. Lastly, the owner is associated with login and view delivery order report use cases.

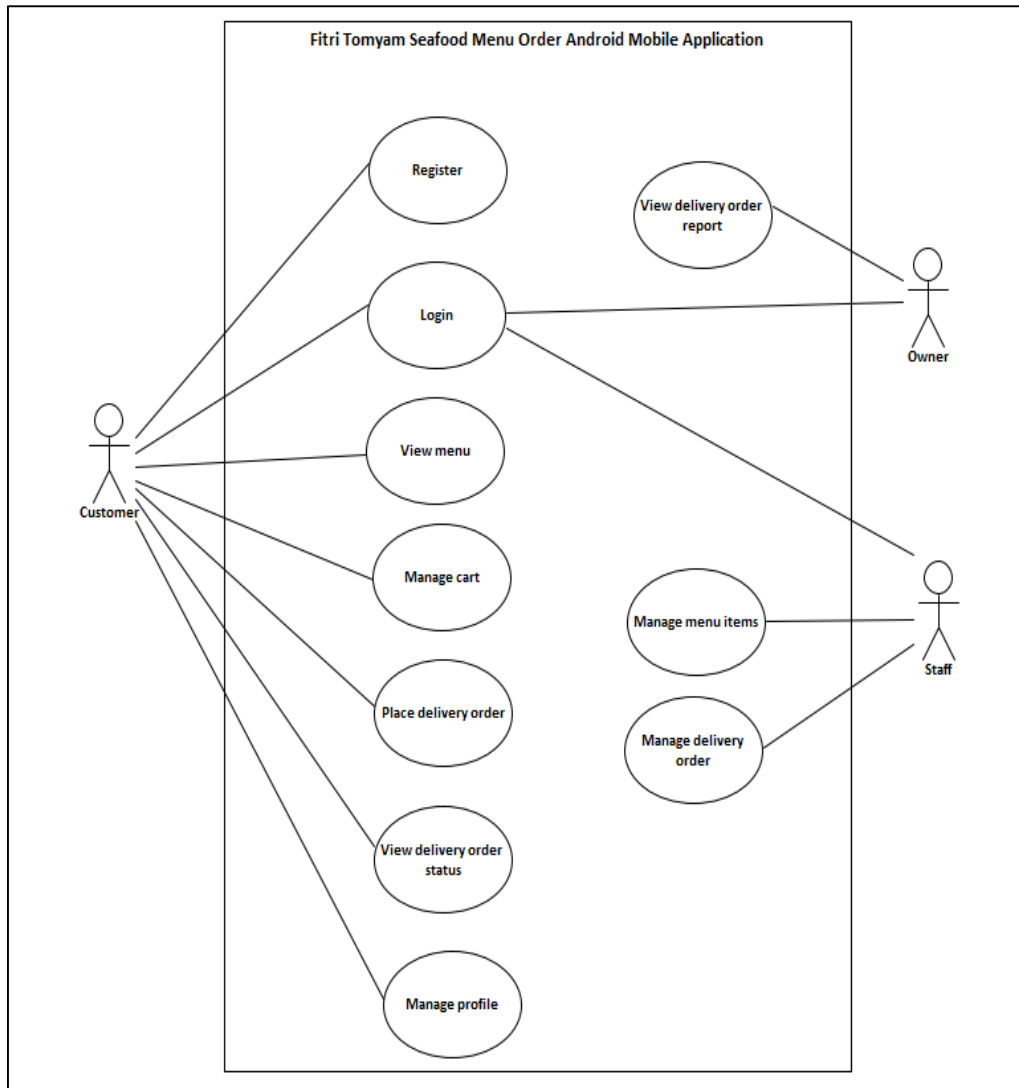


Figure 1: Use case diagram of Fitri Tomyam Seafood Menu Order Android Mobile Application

4.2. Class Diagram

The class diagram illustrates each classes and related attributes and methods of each class existed in the system. Figure 2 shows the class diagram of the system. There are nine classes involved in the class diagram which are staff, customer, owner, menu, cart, order, delivery, payment and report. Each class has its own attributes and methods required to build the system functionalities.

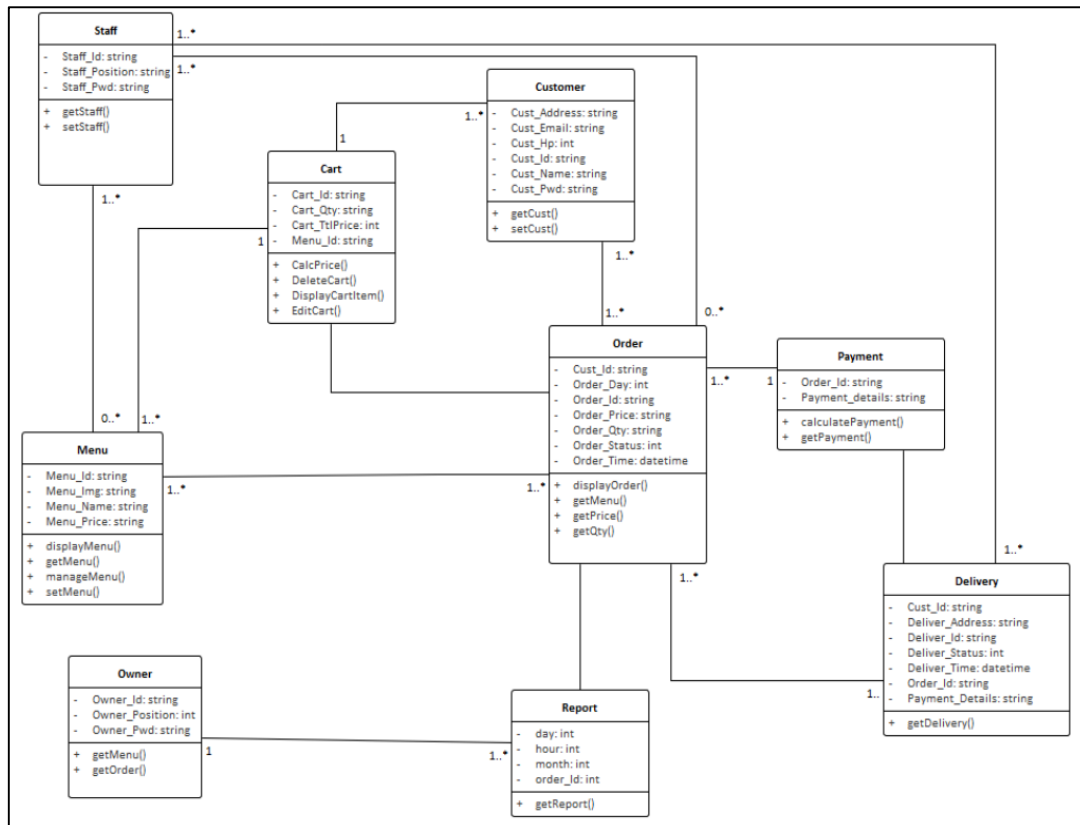


Figure 2: Class diagram of Fitri Tomyam Seafood Menu Order Android Mobile Application

4.3. Activity Diagram

The activity diagram enables user to understand the system easier as it represents the flow and activities of the system. Figure 3 shows the activity diagram of the system for the customer role.

From the start point, the system will display the login page. If the customer has an account, the customer needs to enter the email and password. The system will then verify and validate the input email and password. If the input is valid, the system will display a success login message and direct the customer to the customer dashboard page. Else, if the customer does not have an account, the customer needs to head to the register page. Customer needs to fill in the register form. The system will then authenticate and verify the form. Upon success registration, the system will direct the customer to the login page. However, if the registration is invalid, the system displays the error message.

In the customer dashboard page, the system will display the menu page upon clicking on the view menu tab. If the customer clicks on the add to cart, the system will add the menu item selected by the customer into the customer’s cart and shall display a success message.

At the cart page, the system will display the menu items selected by the customer previously. The customer is able to edit menu items. If any editing’s done, the system will display the latest menu item in cart. If the customer clicks on the checkout button, the system will display the order details dialog and confirm button. Supposing that the customer clicks on the confirm button, the system shall display a successful order message and clear the cart page. The system will direct the customer to the order status page. Here, the system will update the order status accordingly. However, if the customer does not click on the confirm button, the system will display back the cart page.

As the customer heads to the profile page, the system displays the profile details of the customer. Customer clicks on the update button and system shall display the profile detail dialog. If a customer

edits any profile details and confirms it, the system will display the updated profile details at the customer profile page.

Lastly, the customer is able to logout of the system.

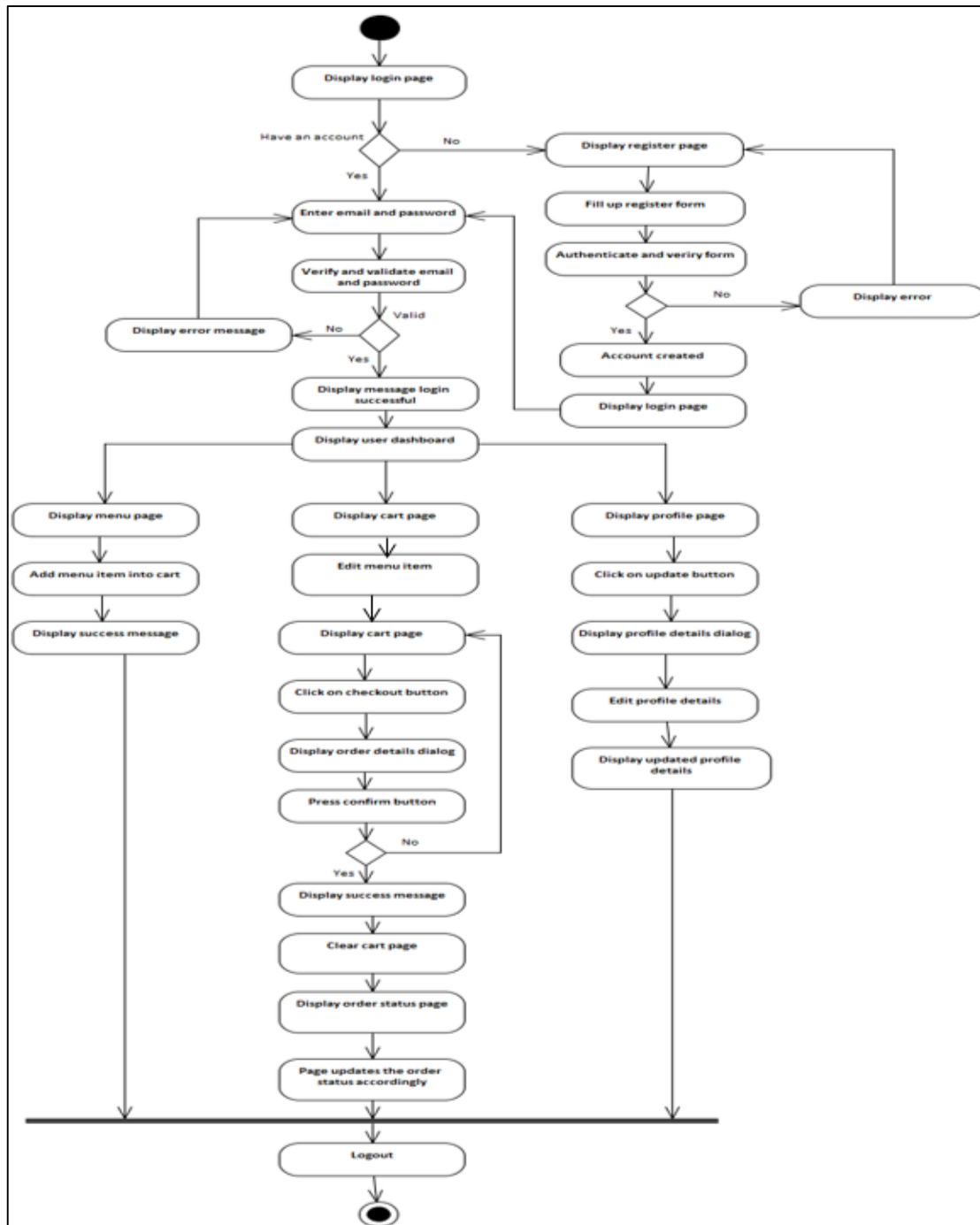


Figure 3: Activity diagram of Fitri Tomyam Seafood Menu Order Android Mobile Application (Customer)

5. Implementation and Testing

5.1. Implementation Phase

Fitri Tomyam Seafood Menu Order Android Mobile Application was developed using Android Studio IDE. The Java programming language and extensive mark-up language (XML) was used.

Firestore database and firebase authentication were also used to store data. Figure 4 shows the Java editor for this project.

```

1 package com.example.dfit;
2
3 import ...
4
5 public class Login extends AppCompatActivity {
6     EditText mEmail, mPassword;
7     Button mLoginBtn;
8     ProgressBar progressBar;
9     TextView mRoleNumber, forgetTextLink ;
10
11     FirebaseFirestore fStore;
12     FirebaseAuth fAuth;
13     String userID;
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.Login);
18         findViewById(R.id.createAccount).setOnClickListener((view) -> {
19             startActivity(new Intent(getApplicationContext(), Register.class));
20         });
21     }
22 }
    
```

Figure 4: Java editor

Registration of users is done by using firebase authentication to ensure non-redundant email addresses being used. Where else, Firestore is used to store the user’s details. Figure 5 shows the code segment for authenticating and registering users whereas Figure 6 shows the code segment for login by differentiating the user’s role. While Figure 7 shows the code segment to save and check whether a menu item has been added to cart. Figure 8 shows the code segment to change delivery order status from active to preparing. Figure 9 shows the interface of Fitri Tomyam Seafood Menu Order Android Mobile Application.

```

// register user in firebase
fAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        Toast.makeText(context: Register.this, text: " Register successful", Toast.LENGTH_SHORT).show();
        userID = fAuth.getCurrentUser().getUid();
        DocumentReference documentReference = fStore.collection("collectionPath: \"User\").document(userID);
        Map<String, Object> user = new HashMap<>();
        user.put("Fullname", fullname);
        user.put("Email", email);
        user.put("Phone", phone);
        user.put("Address", address);
        user.put("Role", role);
        documentReference.set(user).addOnSuccessListener((OnSuccessListener) (aVoid) -> {
            Log.d(TAG, msg: "onSuccess: user profile is created for " + userID);
        }).addOnFailureListener((e) -> {
            Log.d(TAG, msg: "onFailure: " + e.toString());
        });
        startActivity(new Intent(getApplicationContext(), Login.class));
    } else {
        Toast.makeText(context: Register.this, text: "Please insert email in correct format ", Toast.LENGTH_SHORT).show();
    }
});
    
```

Figure 5: Code segment for authenticating and registering user

```

if(temporaryRoleHolder.equals("1")){
    Intent admin = new Intent ( packageContext: Login.this, Admin_Dashboard.class);
    startActivity(admin);
    Toast.makeText(context: Login.this, text: "Welcome Admin ", Toast.LENGTH_SHORT).show();
}

if(temporaryRoleHolder.equals("2")){
    Intent customer = new Intent ( packageContext: Login.this, Cust_Dashboard.class);
    startActivity(customer);
    Toast.makeText(context: Login.this, text: "Welcome ", Toast.LENGTH_SHORT).show();
}

if(temporaryRoleHolder.equals("3")){
    Intent staff = new Intent ( packageContext: Login.this, Staff_Dashboard.class);
    startActivity(staff);
    Toast.makeText(context: Login.this, text: "Welcome", Toast.LENGTH_SHORT).show();
}
    
```

Figure 6: Code segment differentiating user’s role for login


```

if(dataSnapshot.exists()){
    Snackbar snackbar = Snackbar.make(findViewById(android.R.id.content),
        text: "Sorry " + nameTemp + " has already added to cart", Snackbar.LENGTH_LONG);
    snackbar.show();
}
else {
    y.child("total").setValue(j);
    y.child("name").setValue(nameTemp);
    y.child("price").setValue(priceTemp);
    y.child("imageUrl").setValue(image);
    y.child("cumulativeTotal").setValue(priceTemp);
}
    
```

Figure 7: Code segment differentiating user’s role for login

```

final DatabaseReference changeOrderStatus = FirebaseDatabase.getInstance().
    getReference().child("Order (Details)/user ID:" + CustID + "/" + OrderID);
changeOrderStatus.child("status").setValue("Preparing");
    
```

Figure 8: Code segment differentiating user’s role for login

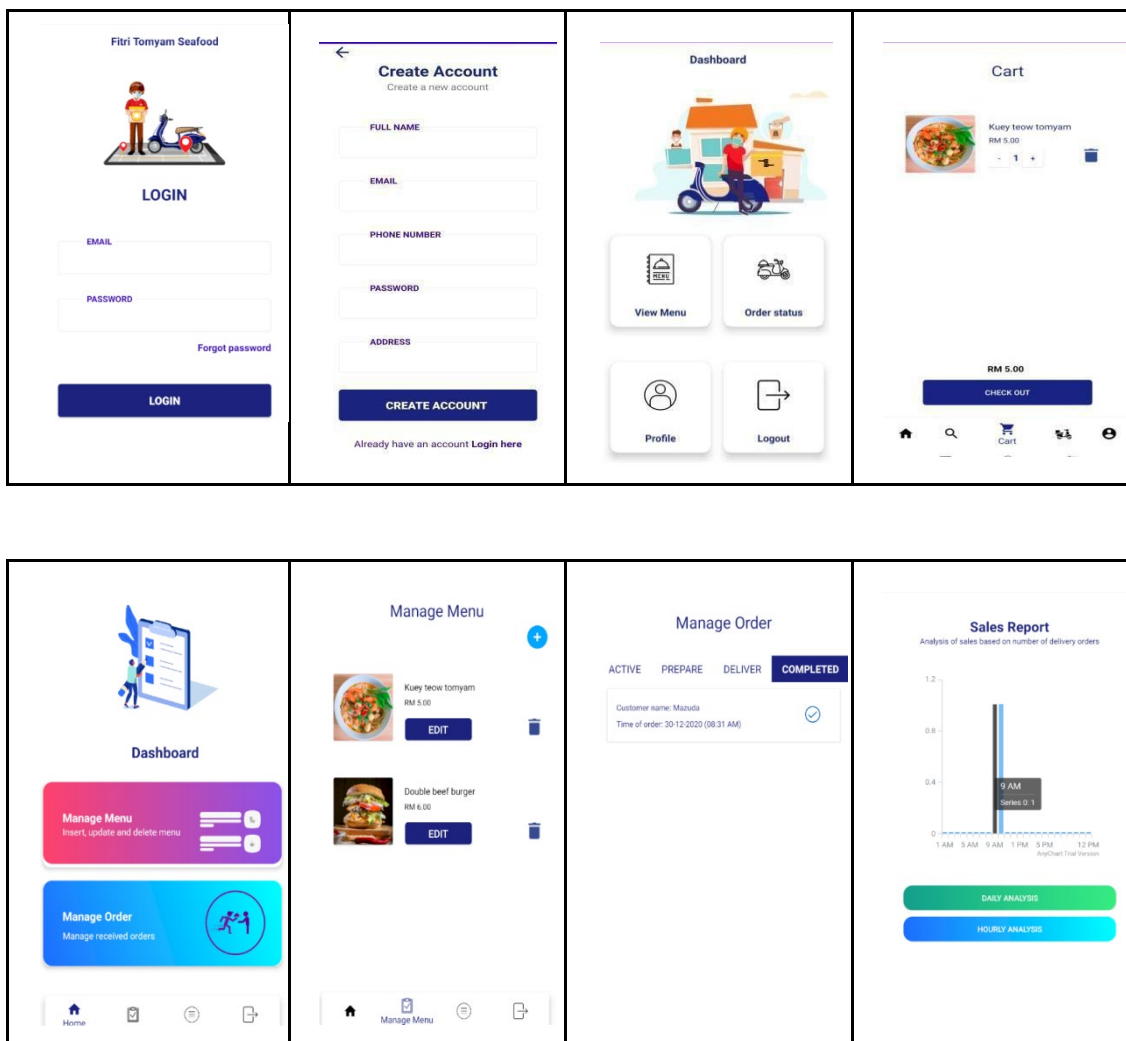


Figure 9: Interfaces of the application

5.2. Test Plan

Table 4 shows the functional testing plan of Fitri Tomyam Seafood Menu Order Android Mobile Application. The purpose of the test plan is to ensure that the modules are functioning based on their requirements and expected results.

Table 4: Test plan of the system module and result

Module	Test case	Expected Result	Actual Result
Register module	<ul style="list-style-type: none"> Customer can register a new account by filling up the registration form correctly 	<ul style="list-style-type: none"> Message “Registered successfully” , system displays login page 	Pass
Login module	<ul style="list-style-type: none"> Users are allowed to login into the system using their valid email and password 	<ul style="list-style-type: none"> System will display “Login successful” and direct users to their respective dashboard based on their roles. 	Pass
View menu	<ul style="list-style-type: none"> Customer is able to view the menu and select their menu item to be added into cart 	<ul style="list-style-type: none"> Menu list is being displayed and menu item added into cart for the selected menu item 	Pass
Manage cart	<ul style="list-style-type: none"> Customers are able to increase and decrease the quantity of menu items in cart. Customer is able to delete menu item in cart 	<ul style="list-style-type: none"> Menu item in cart decrease & increase when action done Menu item is deleted from cart 	Pass
Place delivery order	<ul style="list-style-type: none"> Customer is able to place their delivery order directly 	<ul style="list-style-type: none"> Customer’s order is received by the staff manage order page 	Pass
View delivery order status	<ul style="list-style-type: none"> Customer is able to view their delivery order’s status 	<ul style="list-style-type: none"> System displays updated delivery order status for each customer 	Pass
Manage profile	<ul style="list-style-type: none"> Customer is able edit and update their profile details for the purpose of ordering details 	<ul style="list-style-type: none"> System displays updated customer profile 	Pass
Manage delivery order	<ul style="list-style-type: none"> Staff is able to view and manage the delivery orders received Staff is able to update each delivery orders’ status 	<ul style="list-style-type: none"> System display received delivery orders. System updates customer order status page 	Pass
Manage menu items	<ul style="list-style-type: none"> Staff is able to insert new menu item, update and delete menu items of the menu 	<ul style="list-style-type: none"> System displays new menu item/ updated menu item. When deleted menu item, menu item is deleted 	Pass
View report	<ul style="list-style-type: none"> The owner is able to view the overall counts of orders received by days and by hours 	<ul style="list-style-type: none"> System displays reports based on the counts of delivery orders by days and hours 	Pass
Log out	<ul style="list-style-type: none"> Users are able to log out of the system 	<ul style="list-style-type: none"> System clears user information and redirects to the login page 	Pass

5.3. User Acceptance Testing

Upon completion of implementing the system, a user acceptance testing has been conducted. In this testing, the Fitri Tomyam Seafood Menu Order Android Mobile Application was tested by a group of users. These users consist of customers, staff and the owner of the restaurant. Each user is required to evaluate by providing their responses via answering the user acceptance questionnaires provided in the Google Form. The evaluation was mainly based on the system interface and system features. The results of the responses were then converted into a clearer view by summarizing it into a bar chart.

Figure 10 shows the bar chart regarding the result of the user interface evaluation. It consists of four levels of satisfaction which are poor, fair, good and very good. Under the user interface evaluation, it focuses on a few criteria which are the ease in understanding the interface, the layout content, text style and the interface design. Overall, each criterion had the highest evaluation on “very good” satisfaction.

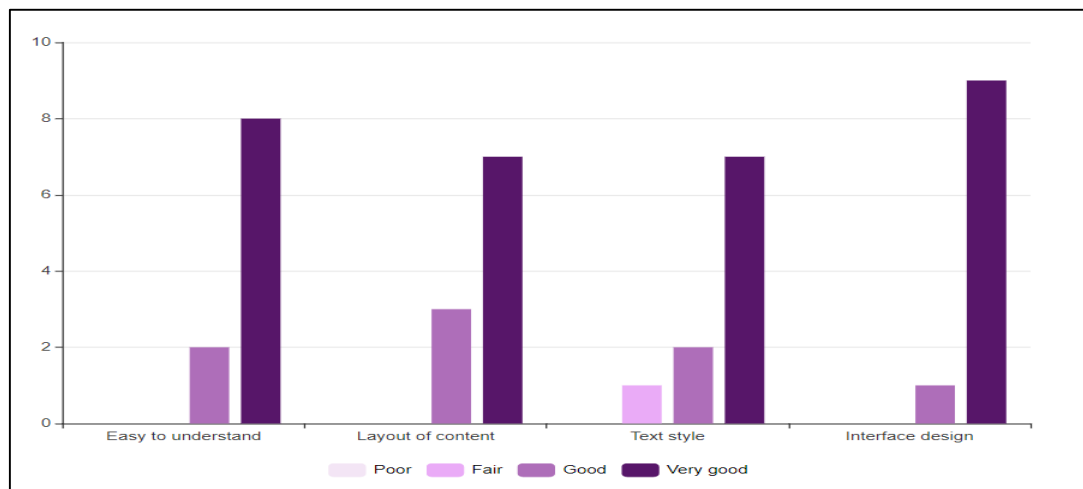


Figure 10: Result of user interface evaluation

Figure 11 and 12 illustrates the bar chart regarding the result of the system features evaluation. The evaluation was to evaluate whether the system functionalities as stated in the requirement meets the developed system. Therefore, the modules of the system which are register, login, view menu, manage cart, place delivery order, view delivery order status, manage profile, manage delivery order, manage menu item and view delivery order report act as the main base and criteria for this evaluation segment. Also, it consists of four levels of satisfaction which are poor, fair, good and very good. Based on the result, overall each criterion had the highest evaluation on “very good” satisfaction. Majority of each criterion, the users evaluated on good satisfaction too. To conclude it, the overall graph for the user acceptance testing represents that the users are considered satisfied with the design and features provided by the developed system.

6. Conclusion

In conclusion, Fitri Tomyam Seafood Menu Order Android Mobile Application has been successfully developed. Few advantages, limitations and future improvements have been identified. The advantages of this application includes customers can place delivery orders directly through this application and get it delivered to their place. Customers are able to view the menu along with its prices more easily and systematically. Next, this system helps in reducing manual tasks in taking customer’s delivery orders. The staff is also able to add, delete and update their menu item in the system with less effort and expenses.

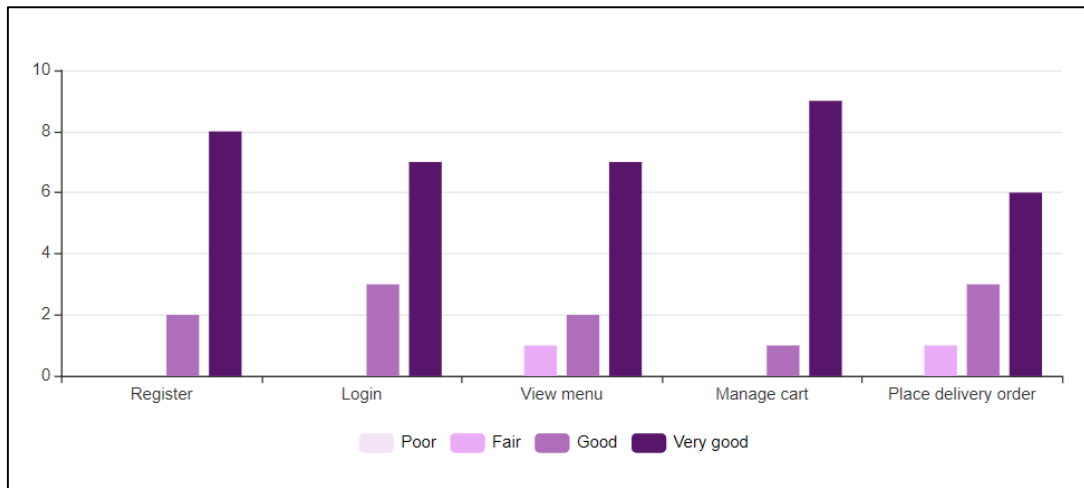


Figure 11: Result of the system features evaluation



Figure 12: Result of the system features evaluation

The limitations of the application include; the inconvenient method to change delivery addresses where customers must go to the update profile page to update their address when there are changes. Besides that, the system only generates sales reports based on the counts of delivery orders received instead of financial reports. The staff shares the same account in handling delivery orders and is not able to edit and cancel delivery orders. Lastly, the delivery areas and time are not restricted automatically.

The future improvements that can be done are by providing a change address option in the confirm page directly to make it more convenient. Next, allowing the view sales report module to generate the reports on sales profits. Next, the system should provide staff registration and their own account. Lastly, the system should also be integrated with the Google Maps APIs in order to restrict areas that are not offered delivery orders by the restaurant.

Acknowledgement

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

References

- [1] L. Ma, L. Gu, and J. Wang, “Research and Development of Mobile Application for Android Platform,” vol. 9, no. 4, pp. 187–198, 2014.
- [2] S. G. Jeff McWherter, Professional Mobile Application Development. Indianapolis: John Wiley & Sons, Inc., 2012.
- [3] K. S. R. S. B. Dhonde, “Android Based Intelligent E-Restaurant Ordering System,” no. May, pp. 19–20, 2016
- [4] R. A. Johnson, An Introduction to Java Programming and Object-Oriented Application Development. Boston: Cengage Learning, 2012.
- [5] O. K. Eason and O. Eason, “Information Systems Development Methodologies Transitions: An Analysis of Waterfall to Agile Methodology Agile Methodology,” 2016