

HUNG PHAM

FOOD ORDERING SYSTEM USING ANDROID MOBILE APPLICATION

FOOD ORDERING SYSTEM USING ANDROID MOBILE APPLICATION

Hung Pham
Bachelor's thesis
Autumn 2022
Degree in Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme, Information Technology

Author(s): Hung Pham

Title of the thesis: Food Ordering System Using Android Mobile Application

Thesis examiner(s): Meija Lohiniva

Term and year of thesis completion: Spring 2023

Pages: 60

The purpose of the thesis was to design and develop a mobile food ordering application, specifically the Android application, which uses Android Studio as a platform for developing the application and Google Firebase as a server for storing all information for the application. The idea of the application was to combine the food ordering needs of android users and Android developers to create a proper Android application. This thesis aimed to develop a prototype for food ordering systems and provide readers with basic knowledge about implementing an Android application.

The theoretical part of the thesis discussed how to create a native mobile application by providing three ways of building a native mobile application which is native, hybrid, and web-to-native. The above ways will be compared and thereby giving an overview of the good options for creating a native mobile application as well as for the empirical part.

The empirical part of the thesis described the process of implementing mobile food ordering applications on Android devices. The mobile application was developed with the help of software such as Android Studio and Google Firebase, which was used to store the database.

The result of this thesis project was a working Android mobile application that includes basic functionality for a food ordering application. In addition, evaluating the completeness of the application by considering whether the application has met the initial requirements or not and from there giving ideas for future improvement and development. Besides, the effectiveness of using the chosen method outlined in the theoretical part in creating an Android mobile application is also evaluated. And from there, assessing the importance of choosing a development method as well as predicting the future of mobile applications development.

Keywords: Android, Java, Authentication, Firebase, develop, native, hybrid, web, application, testing

ACKNOWLEDGMENTS

I want to thank my supervisor Ms. Meija Lohiniva for guiding me patiently throughout this thesis and providing valuable comments to improve my thesis report. Besides it, great thanks to an Internet search engine that helped me a lot in collecting the knowledge for this thesis and my application.

CONTENTS

Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

1	INTRODUCTION	10
1.1	Project Inspiration.....	10
1.2	Project Objectives	11
1.3	Project Scope	11
2	SOLUTIONS FOR DEVELOPING MOBILE APPLICATION (THEORETICAL BACKGROUND).....	13
2.1	Native solutions	13
2.1.1	IOS.....	13
2.1.2	Android	15
2.2	Hybrid Solution	18
2.2.1	React Native	19
2.2.2	Flutter.....	19
2.2.3	Xamarin.....	20
2.3	Web to native solution	21
2.3.1	Good Barber	21
2.3.2	Shoutem.....	21
2.4	Comparison between all solutions	21
3	DEVELOPMENT OF THE PROJECT'S APPLICATION	23
3.1	Project Development	23
3.2	Project plans.....	25
3.2.1	Use Case Diagram.....	25
3.2.2	System Flowchart (Customer).....	26
4	DESIGN.....	28
4.1	User Interface Design.....	28
4.2	Database design	40
4.2.1	Firebase Realtime Database.....	40
4.2.2	Firestore Database	40
4.2.3	Authentication and Storage.....	42

5	TESTING	43
5.1	Development Tools	43
5.2	Unit Testing	43
6	SYSTEM EVALUATION	56
6.1	Evaluation of system status.....	56
6.2	Future Enhancement.....	56
7	CONCLUSIONS	58
	REFERENCES	59

LIST OF FIGURES

Figure 2-1-1: iOS architecture
Figure 2-1-2: Android architecture
Figure 2-4: Comparison of Native and Hybrid solution
Figure 3-1: Diagram of an evolutionary model
Figure 3-2-1: Use Case Diagram
Figure 3-2-2: System Flowchart
Figure 4-1-1: Splash Screen
Figure 4-1-2: Login Page
Figure 4-1-3: Register Page
Figure 4-1-4: Forgot Password Page
Figure 4-1-5: Register and verify Email
Figure 4-1-6: Reset Password by Email
Figure 4-1-7: Verify Email
Figure 4-1-8: Reset Password Email
Figure 4-1-9: Navigation bar
Figure 4-1-10: Main Page
Figure 4-1-11: Foods by Category
Figure 4-1-12: Food Details
Figure 4-1-13: My Cart
Figure 4-1-14: Remove Item in Cart
Figure 4-1-15: Search Engine
Figure 4-1-16: Customer's Profile
Figure 4-1-17: Discount Page
Figure 4-1-18: Item in Discount Page
Figure 4-1-19: Order Placement Page
Figure 4-1-20: Order History Page
Figure 4-2-1-1: User Database in Firebase Realtime Database
Figure 4-2-2-1: Data in Firestore Database
Figure 4-2-3-1: Authentication

LIST OF ABBREVIATIONS

IDE – Integrated Development Environment

XML - Extensible Markup Language

JSX - JavaScript XML

App - Application

SDK – Software Development Kit

UI – User Interface

NoSQL – Not only SQL

AVD – Android Virtual Device

Q-commercial – Quick commercial

1 INTRODUCTION

1.1 Project Inspiration

The birth and development of the Internet have led to the formation and continued growth of a new industry called E-commerce. Age of Technology 4.0 focuses on the development of technology, which means that all data, information, and user needs will now be stored and managed under the Internet system. Combining the above two things, we can see how technology is and will significantly influence industry and life (deepaman07ad, 2021) (1).

The vigorous development of E-commerce in recent years is now not only in the shopping field but also leads to the new development of the food and beverage industry called Q-commerce. Most e-commerce transactions in the food and beverage fields are made available through mobile phones and tablets. Seeing the interest in the topic related to Q-commerce, I decided to dig deeper to find out how it works and how to form a food ordering application. However, the more I researched, I found that there are many solutions with which to create a food ordering application from a native to cross-platform (hybrid), and there is also a way to rely on third party websites to help ~~at~~ create the application. I discovered that each of the methods mentioned above has its strengths, and it is challenging to choose one. After thinking for a while about the topic of the thesis, it was decided to create a food ordering application on Android device, an option that belongs to the native solution.

Two reasons were summarized to bring up the topic; the first is a personal reason since I am pretty familiar with working with Android, so this will help in reducing time and also improving the quality of the final product. Besides, the number of Android phone users accounts for a large number, so that the topic will be accessible to many people. The second reason is that the native solution will make it easier to learn and understand especially for beginners because it only focuses on one aspect of application creation. Through this project, I hope to learn more about the development of food ordering applications along with training skills in software development on Android applications. In addition, providing Android programming beginners with the most basic view of different methods for creating a mobile application. The solutions for creating a mobile application will be described in Chapter 2, and a prototype of developing a food ordering application on the Android platform will be in Chapter 3 and Chapter 4.

1.2 Project Objectives

- To develop a database to store user information with email verification to avoid duplicate data and virtual users.
- To develop a database to store and update food product information.
- To develop a link between user and food product information for buying and payment.
- To develop a mobile food ordering system that allows the users to have a real-time view and update of the food information.
- To develop a mobile food ordering system that has a payment system that uses a Stripe (or PayPal) account.

1.3 Project Scope

At the end of the project, there will be a mobile food ordering application prototype. For the server side, the user information for login and registration will be stored in the Firebase real-time database. For registration, users need to make an authentication through email for verification to connect to the food product information of the application. The food product information will be stored in the Firebase Firestore database, and the change in food product information is also done online on the Firestore database. Users have to log in or register first for the client side. After that, they will be able to view the food menu food detail, order the food by adding it to the cart and modify the amount of food they want, and finally, the payment process that uses a Stripe (or PayPal) account. Below are modules that had or will be made shortly.

Login and Registration Module

This module contains the user's information. For login, there are email and password; for registration, there are email, username, address, password, and confirm password. All data above will be stored in the Firebase real-time database.

User Authentication Module

This module is for the new user's registration. After registering to the app, users need to verify by accepting the verification link sent in their email. It provides security to ensure there will be no virtual users and only valid users can use the application.

Menu Information Module

After logging in or registering, users will be redirect to the application's main page, where they can see the food menu and their information.

Cart Module

In this module, users can see information about the food they chose containing the date, time, food name and price, quantity, and the total amount they have to pay. In case users make an order of more than one food, the information about the total price that users have to pay will be updated.

Payment Gateway Module

After adding the food to the cart, users will process the payment and are redirected to the Stripe (or PayPal) page to process their payment.

2 SOLUTIONS FOR DEVELOPING MOBILE APPLICATION

This chapter will cover three solutions for developing an Android application, which is Native, Hybrid, and Web to native. Thus, the readers are provided with an overview of the direction of developing an application and a comparison to help readers choose the solution that best suits their needs.

2.1 Native solutions

The native solution is the first solution for mobile application development. Native solution builds applications using programming languages (such as Swift for iOS and Java for Android) and tools specific to a particular mobile platform. This solution provides the best performance, user experience, and full advantage of the device's capabilities. Up to now, it is always the most preferred solution in mobile app development, focusing on platform development such as IOS, Android, and Windows phones.

2.1.1 IOS

2.1.1.1 History

Apple Inc developed iOS, ~~which~~ was released as iPhone OS in June 2007. It boosted the phone industry informing a counterweight to Android dominance. In July 2008, App Store was opened leading to the strong development of iOS phone. Currently, iOS covers about 26.98% of the market in Mobile Operating System market share (Mobile Operating System Market Share Worldwide,2022) (2).

From 2007 to present, iOS has published many versions to improve user experience. At the time, there were sixteen big iOS versions with many new features. Currently, version sixteen is leading in the market share of iOS Mobile Operating System with 51.9%, (Daniel Jilg, 2022) (3).

2.1.1.2 Architecture

The iOS operating system was built based on four layers. Its communication does not occur directly; instead, there is a layer between the Application Layer and Hardware Layer that helps to facilitate communication. The lower-level layers in iOS provide the basic services, and the higher-level layers provide graphics and interface-related services (Omkar Kumbhar, 2022) (4). The image below shows the layers:



Figure 2-1-1 iOS architecture

- **Core OS:** the lowest layer in iOS is also the system's foundation that takes care of tasks such as system management and networking and directly affects the hardware.
- **Core Services:** provides services based on the Core OS layer. It provides basic access to iOS services.

- **Media:** provides services for audio, video, animations, and graphics.
- **Cocoa Touch:** The Application Layer acts as an interface for the developer to work with the iOS Operating system. It supports touch and motion events and many more features.

2.1.1.3 Development Tools

This section aims to list the most used development tools. Among the existing tools for developing an iOS application are XCode and AppCode.

With XCode, it was released by Apple, and it is an IDE (Integrated Development Environment). The XCode environment allows developers to quickly write, compile, debug and publish the app on AppStore. In addition, XCode has features to support developers in designing and testing graphic interfaces without writing a single line of code. Besides, developers can build, install, and run apps and debug in a Mac-based Simulator with iOS SDK (Xcode, n.d.) (5).

For AppCode, it is a tool built on the IntelliJ IDEA platform and launched by JetBrains. As a single integrated development environment, AppCode supports many programming: Objective-C, Swift, C++, JavaScript, XML, HTML, and CSS. Besides, intelligent code assistance helps developers create clean code, catch errors, and fix it quickly (AppCode Features, n.d.) (6).

2.1.2 Android

2.1.2.1 History

In 2003, Android was initially developed by Android Inc and intended to be used as an operating system for digital cameras. A year later, Android became an operating system for smartphones. Realizing the potential of Android operating systems, Google Inc decided to buy Android Inc in 2005. On November 5, 2007, Google announced the Open Handset Alliance's founding, including Intel Corporation, Motorola, NVIDIA Corporation, Texas Instruments Incorporated, LG Electronics, Samsung Electronics, Sprint Nextel Corporation, and T-Mobile (The Editors of Encyclopaedia Britannica, 2020) (7).

In October 2008, the first cellular telephone that used the Android operating system was released, named T-Mobile G1. Up to now, Android is always on top and ranked number one in the mobile operating systems market share with 72.37% (Mobile Operating System Market Share Worldwide, 2022) (8).

2.1.2.2 Android Architecture

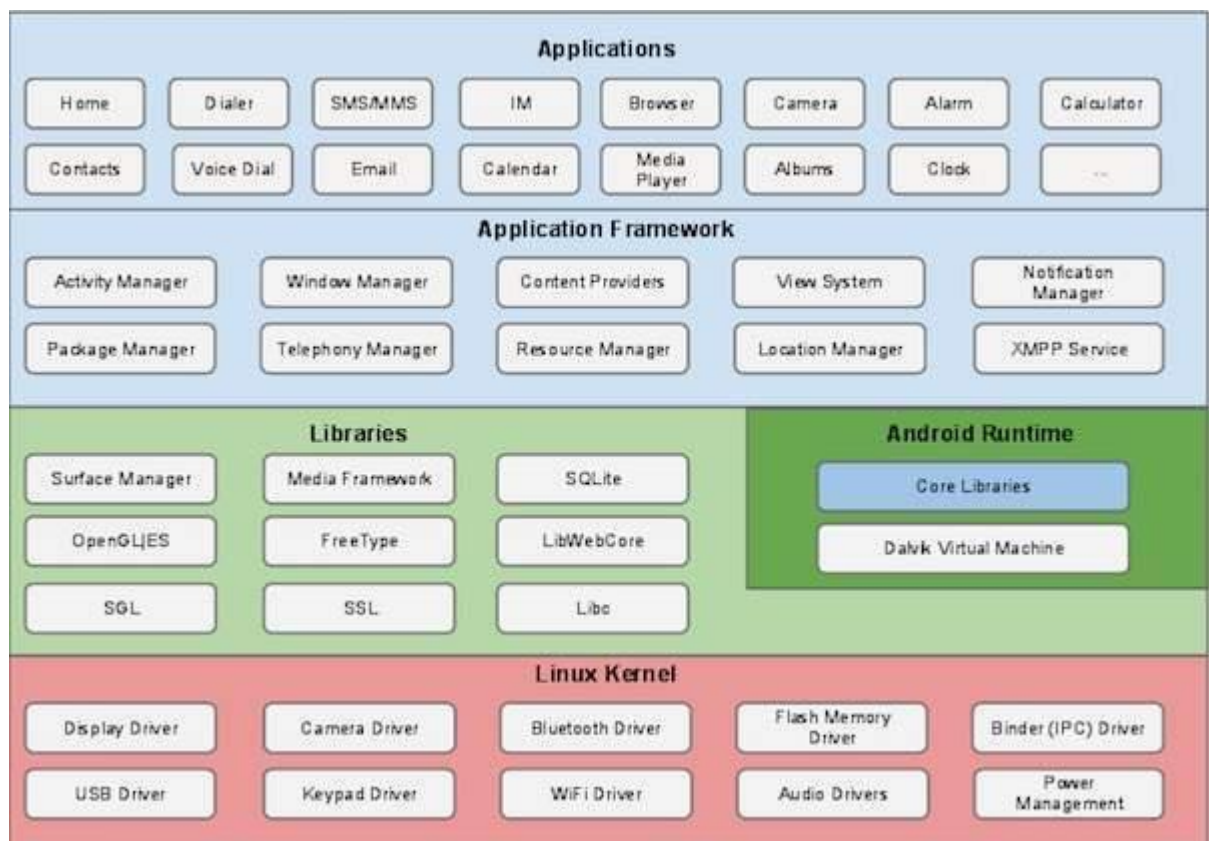


Figure 2-1-2 Android architecture

The android architecture contains five main components (Praveenruhil, 2021) (9):

- **Applications:** The top layer of the Android architecture. This layer contains pre-installed and third-party applications installed from the play store. The Applications run on Android Runtime with the help of classes and services provided by the Application Framework.
- **Application Framework:** This framework provides many higher-level services to applications in the form of Java classes. It includes an activity manager, notification manager, view system, package manager, etc., which are helpful for the development of the application according to the prerequisite.
- **Libraries:** This section contains basic Java libraries specifically for Android development. Libraries were used for building user interfaces, graphics, and database access.
- **Android Runtime:** An application provides a set of libraries that help developers program and write Android applications using Java programming languages. At the same time, this component supports the core set of libraries that developers use to develop Android applications.
- **Linux Kernel:** This is the core of Android architecture. This component contains the entire low-level communication device used to manage the hardware operations on Android devices. Linux is responsible for adding basic features and managing the network, drivers, memory, processing, and security.

2.1.2.3 Development tools

This section aims to enumerate a set of development tools most used in Android development. Two development tools are top-rated in Android development: Android Studio and Eclipse.

Android Studio is the most preferred tool for Android developers. It is an official integrated development environment for Android application development. Android Studio is a free download supported by Google featuring code altering, troubleshooting, and testing devices inside a simple-to-use intuitive interface. Some top features that make Android Studio popular are Visual Layout Editor, Fast Emulator, APK Analyzer, Intelligent Code Editor, Real-time Profiles, and Support Kotlin programming (Android Studio, n.d.) (10).

Eclipse is an integrated development environment (Eclipse IDE) for computer programming, initially developed by IBM and now by the Eclipse organization. It is primarily used for Java but can still be used to program in other languages such as C, C++, Ruby, etc. The developer can use Eclipse to extend the source code by inserting the plugins for the project. In particular, Eclipse is free software with many useful plugins such as Maven, Spotbugs, etc (Eclipse (software), n.d.) (11).

2.2 Hybrid Solution

A hybrid solution or Cross-platform solution is a combination of native mobile apps and web applications where the core of the application is written by using web technologies (Design Rush,2022) (12).

Along with the appearance of the cross-platform development, it gives developers as well as businesses a lot of value and benefits:

- **Wider customer base:** developers or businesses now do not have to choose which audience to target, i.e., iOS or Android users, as cross-platform software runs both.
- **Platform consistency:** The shared codebase in a hybrid solution helps ~~to~~ deal with the navigation and design difference between, i.e., iOS and Android, thereby saving more effort.
- **Reusable content:** Instead of having to write each program corresponding to each operating system, with cross-platform development it just needs to build one codebase that can run for all operating system requirements.
- **Quicker development and cost reduction:** Just building one codebase that can run on multiple platforms helps increase product development, thereby saving time, resources, and money.

Besides the advantage of hybrid solution brings for mobile app development, it also has some drawbacks compared with the native solution:

- **Requires more expertise:** To maintain the application's performance, hybrid solutions require developers to have enough skills and expertise.

- **Complex code design:** To create a codebase that can run on multiple platforms makes design and code more complicated to be responsive to each platform at runtime.
- **Longer update feature time:** It takes a while to update new features because it needs to meet the requirements of each platform when updating.

This section below provides an overview of some popular hybrid mobile application development methods.

2.2.1 React Native

Facebook developed React Native for internal app development. The first iOS mobile app development version was released in March 2015, and the Android version was released in September. It works the same ways as React but allows developers to build applications for both mobile and desktop. In addition, because most of the code can be shared with React Native, it is easy to develop simultaneously for multiple platforms on mobile phones (O'Reilly, 2015) (13).

React Native applications are built based on JavaScript and XML markup, known as JSX. Then, the “bridge” part of React Native invokes the native rendering APIs in Objective-C (for iOS) and Java (for Android). Thus, the application will render the mobile UI components and look like the other mobile applications created natively.

React Native is mainly used to bring the good to excellent user experience, a suitable option for an app requiring high responsiveness and intuitive use.

2.2.2 Flutter

Flutter is a mobile app SDK (software development kit) created by Google with the purpose of building high-performance, high-fidelity web, desktop, and cross-platform apps that run on Android and iOS devices. Flutter uses Dart, a reactive programming language that makes developing faster and easier (Feza Roheel, n.d.) (14).

Some top features that make Flutter popular among many cross-platform solutions are a Modern and reactive framework, easy-to-learn Dart programming language, running the same UI for multiple platforms, high-performance applications, and hot reload.

The significant advantage of Flutter provides fast and dynamic code writing with the support of the hot-reload feature. Developers can see the alterations made in the code straight away in the application in just a few seconds without restarting anything. Additionally, it brings better collaboration between designers and developers, thus creating unanimity in the team and reacting to the alterations accordingly.

2.2.3 Xamarin

In the beginning, Xamarin was called Mono as an open-source project used to develop a version of .NET for Linux in early 2000. Later, Mono's original developers founded a company called Xamarin. The Mono project continued, developed, and was renamed Xamarin, just like the company name in May 2011. In 2016, Xamarin became a part of Microsoft and was used as cross-platform application development for iOS and Android devices. Up to now, Xamarin has been considered a pioneer in the cross-platform development field (Rohin Tak, Jhalak Modi, 2018) (15).

Creating an application using Xamarin can be done in two ways: Traditional Xamarin Development and Xamarin.Forms. The only difference between the above two ways is that with the traditional way, it is necessary to build UI natively for each platform and the Xamarin.Forms, the UI code is shared for all platforms. With the backend, it is the same for all platforms. One more thing about Xamarin is that it requires developers to know about C# programming language.

So far, Xamarin can still be used in cross-platform development but is not entirely welcome because it is too old if compared to React Native or Flutter. Besides, everything in Xamarin is quite complicated to use, even the text editor and some features even charge. Xamarin is "not dead, but it is moribund." (Dave Francis, 2020) (16).

2.3 Web to native solution

With the continuous development of technology and the increasing demand of users and businesses for mobile applications, a new solution for developing mobile applications called Web to native solution has been created. Web to native means that the solution combines with the support of paid websites to create on-demand applications. This section provides information about several websites that support creating mobile applications.

2.3.1 Good Barber

Good Barber is the first App Builder ruled by a Design System. Good Barber brings to customers best practices in UI and UX. It can be said Good Barber is an appropriate place for developers who want to learn more about designing mobile application and business serious about design and UX. With the price of 35 euros per month for the eCommerce app, 200 euros per month to create unlimited apps, and 30 days a free trial, it is a good idea to try it for the first time. (Good barber website, n.d.) (17).

2.3.2 Shoutem

They provide No-Code App Builder for iOS and Android platforms. With four steps, customers can easily make native iOS or Android applications. Besides, Shoutem also includes user management, analytics, etc., that make the monthly and yearly prices from 50 to 200 dollars, and a fourteen-day free trial. (Shouten website, n.d.) (18).

2.4 Comparison between all solutions

This section aims to compare all solutions, thereby giving the most reasonable solution. The following chart below shows the difference between the hybrid and the Native solution.

Hybrid	Native
HTML, CSS and Javascript languages + a Hybrid framework (Ionic, NativeScript, React Native, Xamarin ect).	Platforms dictate the language used (e.g. Swift for iOS, Java or Objective-C for Android, etc.)
One code that runs anywhere	Separate code needed for each platform
Performance isn't as strong as Native apps	Users experience the most responsive and fastest possible app
Less costs	More expensive to code (and code must be written according to each platform the app runs on)
Less time to code	More time needed to code
Faster development cycle	Slower development cycle
Users will always be on the latest version	Users need to download updates, therefore users may be on different versions at any given time
Little to no integration capabilities with phone elements	Apps are able to access elements such as camera, microphone, calendars, GPS, etc. on mobile
Less secure	Highly secure

Figure 2-4 Comparison of Native and Hybrid solution

The figure above shows the strengths and weaknesses of the two solutions; it is hard to say which is better here. Each solution in each case and demand will promote its strengths; for example, if developers work on a mobile application that relies on elements such as a camera, GPS, SMS, microphones, etc. native solution will be the better choice. Still, a hybrid solution will be the right choice if developers are constrained by budget and time. So, to choose the better option, it is necessary to define the goal that best serves the end user first.

About the Web to native solution, it is pretty hard to compare it with the two solutions above because it is for non-developers, and many factors need to be experienced more to have the best comparison. So, the Web to native solution can only be used as a reference for those who do not know about non-code app builders.

3 DEVELOPMENT OF THE PROJECT APPLICATION

3.1 Project Development

To carry out this project, of the three solutions mentioned in chapter 2, the Native solution was used. The reason to use this solution is that this project focuses on building a mobile food ordering application on Android devices, so it is not necessary to use either a Hybrid solution or Web to Native solution. In addition, it will have a faster and better performance.

After choosing a suitable solution for the project, research and develop a good model is a must-have. In this project, the evolutionary model was used. The figure 3-1 below describes the flow of this project. The first stage, stage 0, is the planning phase, where several diagrams were created to provide an overview of the project and, thus, used as documentation to develop the entire project. Next is stage 1, where the analysis, as well as the design phase, is done. In the analysis phase, the minimum requirements and features required in this project were determined. In the design phase, the design of the user interface was completed. The following stages start from 2 to end without any limitation; in this part, the implementation phase was done along with the testing phase. Each feature was created and tested until satisfaction. Finally, there was the system prototype phase, where the project's final product was completed.

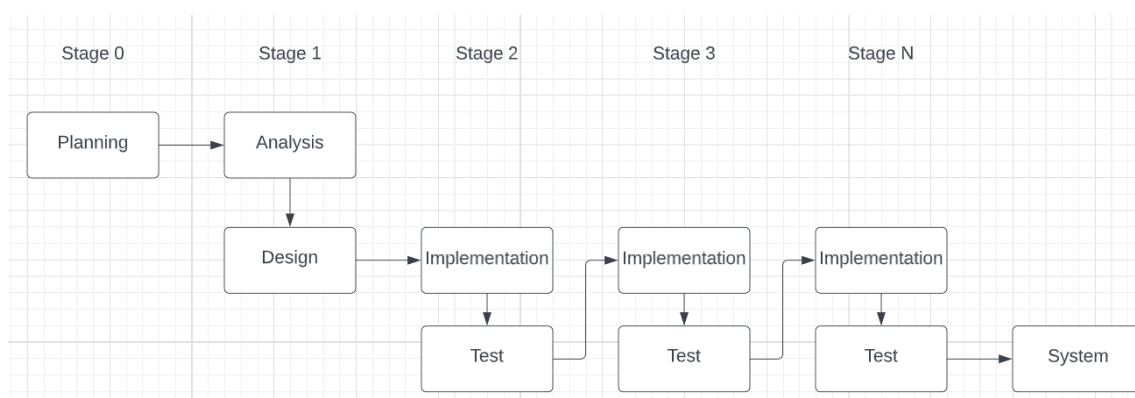


Figure 3-1 Diagram of an evolutionary model

To be more specific about each phase:

Planning phase

During this phase, Lucid Chart and Creately (two tools to create a diagram and chart online) were used to create a Use Case Diagram and System Flowchart to provide an overview of the operation and development of the project.

Analysis phase

In this phase, project objectives and scopes are identified. Besides, some solutions for making a mobile application are reviewed to give a suitable solution to start the project. This phase was already mentioned in chapter 1 and chapter 2 of the thesis.

Design phase

In the design phase, the user interface for the mobile food ordering application must be completed. Thus, the structure of the database was also created.

Implementation and testing phase

In this phase, each module that has been proposed will be developed and tested one by one until all modules are completed.

System prototype

It is the project's last phase, where the product's first version is introduced.

3.2 Project plans

3.2.1 Use Case Diagram

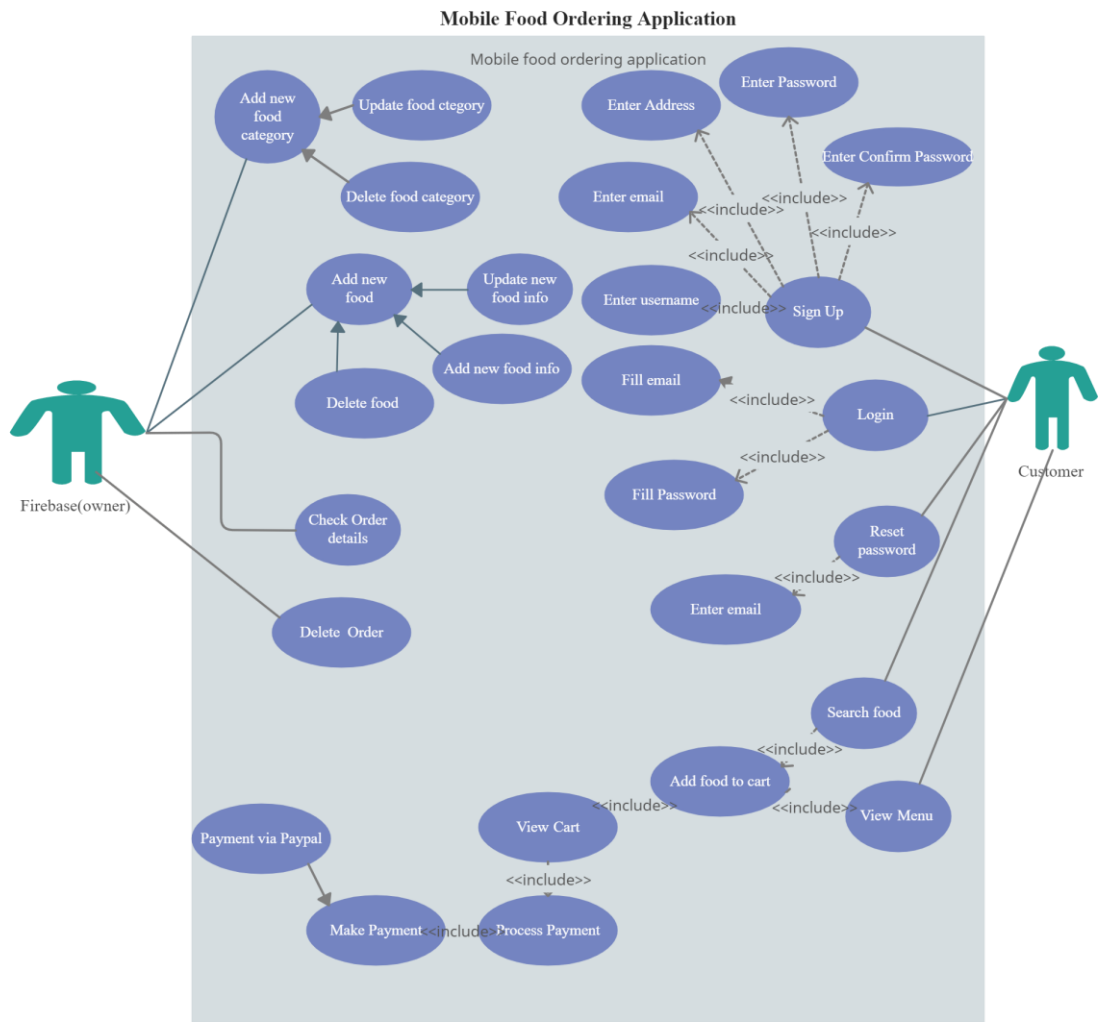


Figure 3-2-1 Use Case Diagram

For figure 3-2-1 above, the firebase database will act as the owner of the mobile food ordering application. The owner will handle the food management, which includes adding new food categories, updating food categories, and deleting food categories. In addition, the owner also needs to add new food, update new food info, add new food info, and delete food. Besides this, the owner is also responsible for checking order details and deleting orders if needed. Customers first sign up if they are new to the application by entering their username, password, address, email, and confirm password. After filling in all blanks in the sign-up form, the customers will receive an email to verify their email, and then they will be redirected to the application's main page. If customers already have an account in the application, they are also redirected to the main page. In addition, if customers forget their password, they can reset the password by entering their email, and they will receive an email that lets them change the password. Customers can see the menu on the main page, select the food, place them into the cart, and checkout. In addition, it also has a search engine that will help customers in sorting their food.

3.2.2 System Flowchart (Customer)

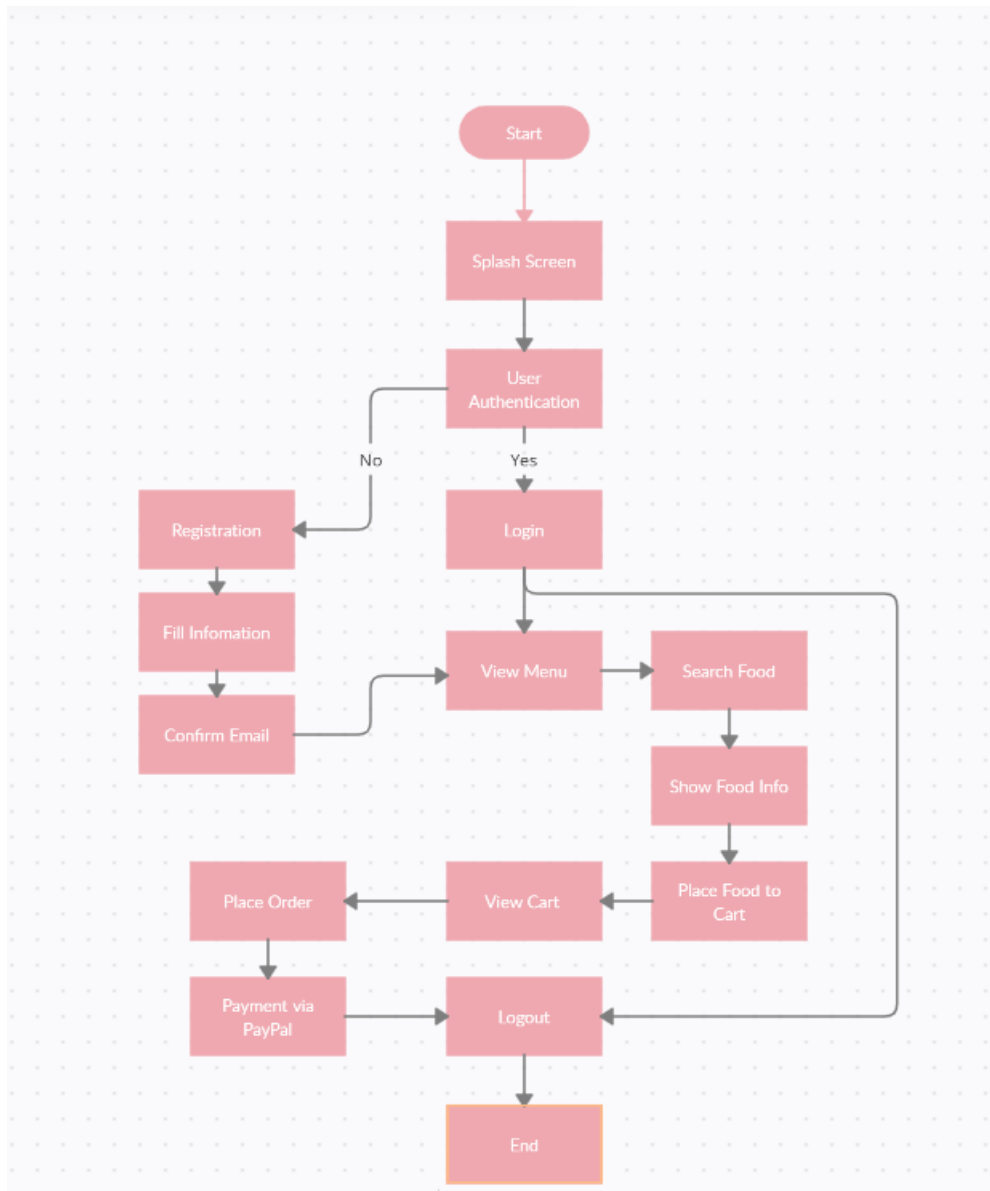


Figure 3-2-2 System Flowchart

For figure 3-2-2 above, the application will start with a splash screen, and customers will be redirected to the user authentication interface. Customers are required to log in or register if they are new to the application to be able to see the menu. Customers can choose between search engines on the main page to search for food and navigate through the menu. By clicking on the food image, customers can see the food information, select their desired food, and add it to the cart. Customers can view and check their food in the cart interface and then be moved to the Stripe (or PayPal) page, where they will pay for the food, they choose.

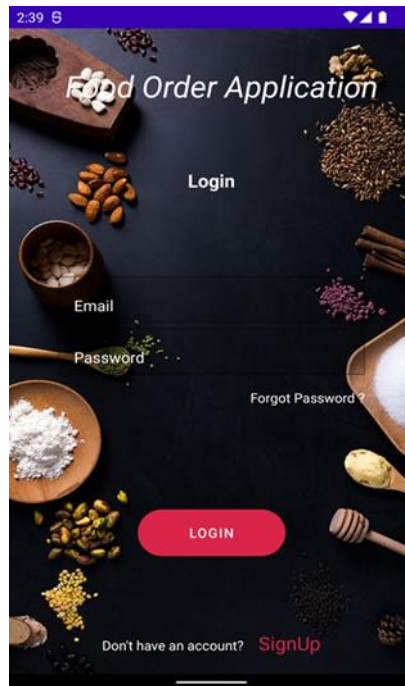


Figure 4-1-2 Login Page

Figure 4-1-2 above is a screenshot of the Login Page; users must enter their email and password to be redirected to the main page. Besides, two other features are “Forgot password,” where users are moved to forget password page to reset the password, and “**Signup**,” where new users can register an account.

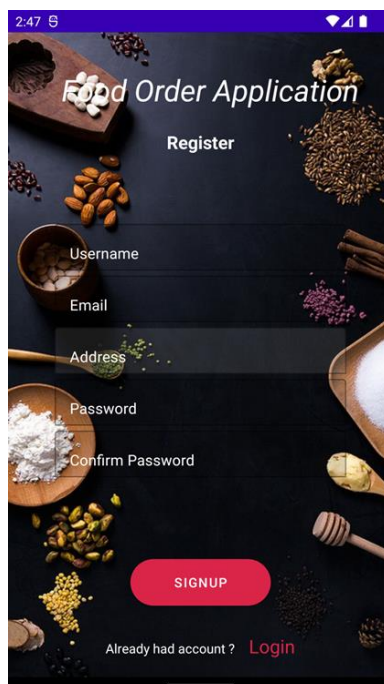


Figure 4-1-3 Register Page

Figure 4-1-3 above is an image of Registration Page; users must fill in all the blanks above to receive an email to verify their account. If users want to go back to Login Page, they can click on the red button “Login.”

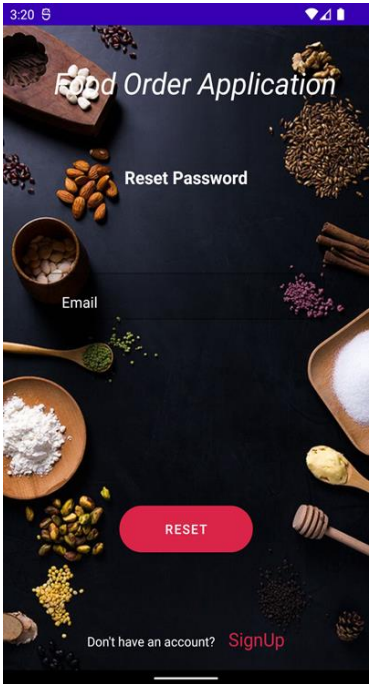


Figure 4-1-4 Forgot Password Page

Figure 4-1-4 is a screenshot of the Forgot Password Page, where users can reset their password using their registered email. After changing the password, users can go back to the Login Page to log in again by clicking the red button “Login.”

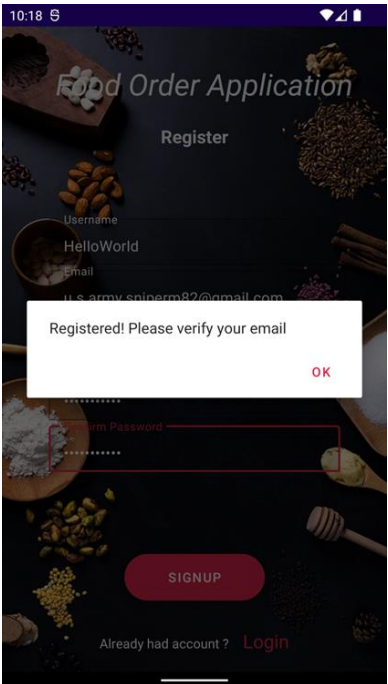


Figure 4-1-5 Register and verify Email

Figure 4-1-5 shows a message to notify that customers have successfully registered, and they need to verify their email before Login. This screen will appear when the user has filled all the blank in the Register Page and click on button “SignUp”.

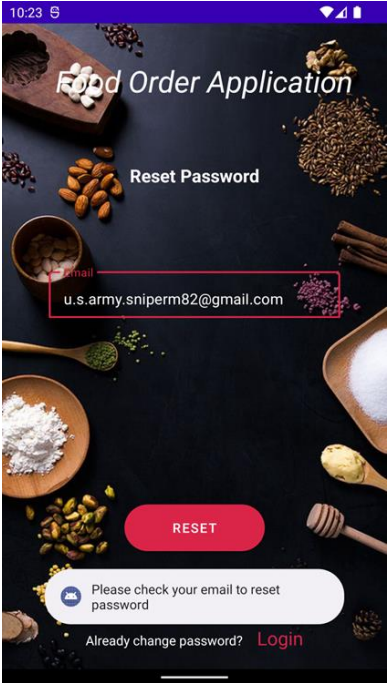


Figure 4-1-6 Reset Password by Email

Figure 4-1-6 shows a message to notify customers to check their email to reset the password. There will be an email sent from Firebase to customers to reset their password.

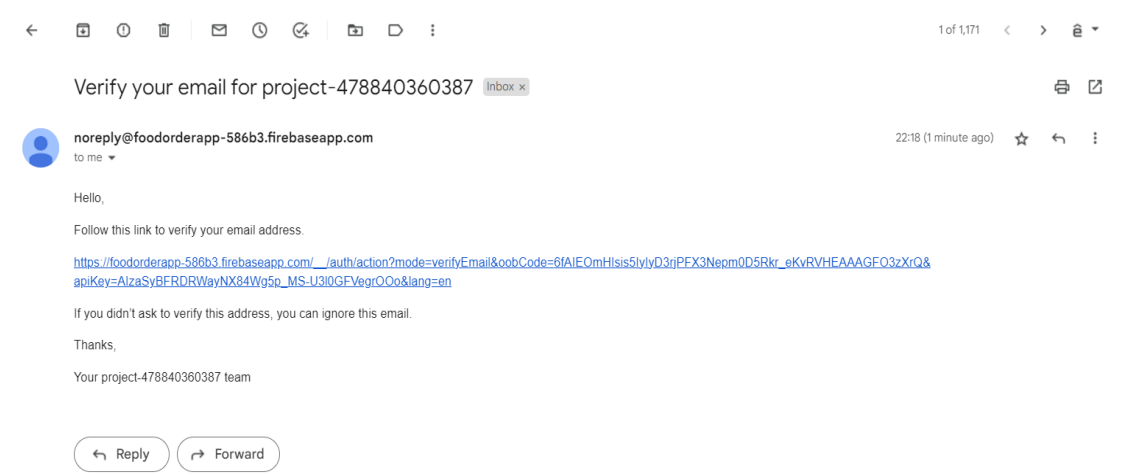


Figure 4-1-7 Verify Email.

Figure 4-1-7 is a screenshot of when customers have received an email to verify their email to log in to the application.

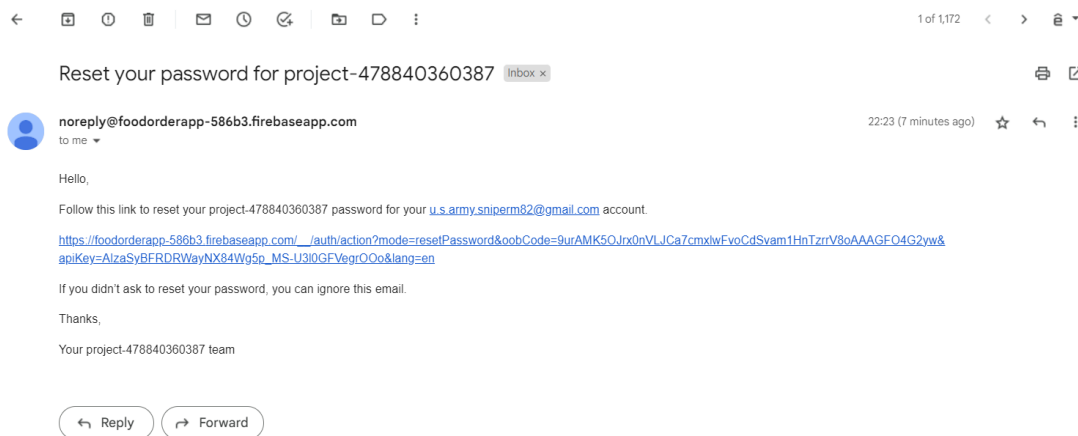


Figure 4-1-8 Reset Password Email.

Figure 4-1-8 is a screenshot of an email sent from Firebase to customers to reset password.

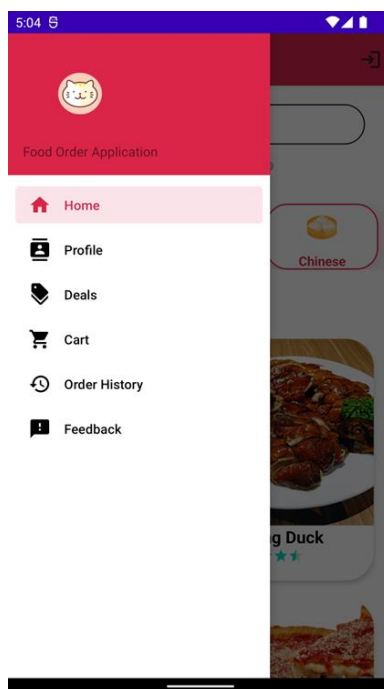


Figure 4-1-9 Navigation bar

Figure 4-1-9 shows a list of contents and services that the application provides. Users can navigate and see the information they need by clicking on each item above.

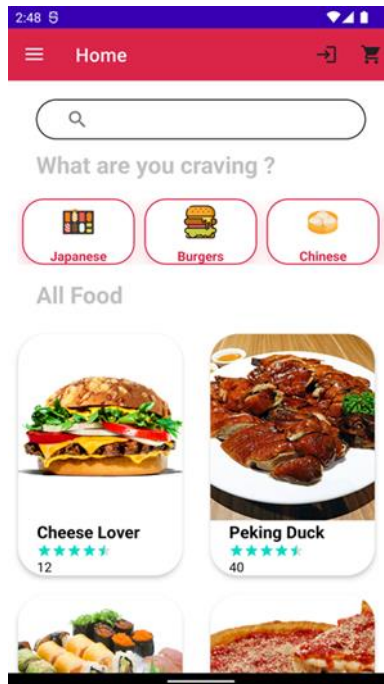


Figure 4-1-10 Main Page

Figure 4-1-10 is a screenshot of the main page showing the food menu. This page contains a search engine, a category bar to choose food by category, and a food list that lists all the food the application has.

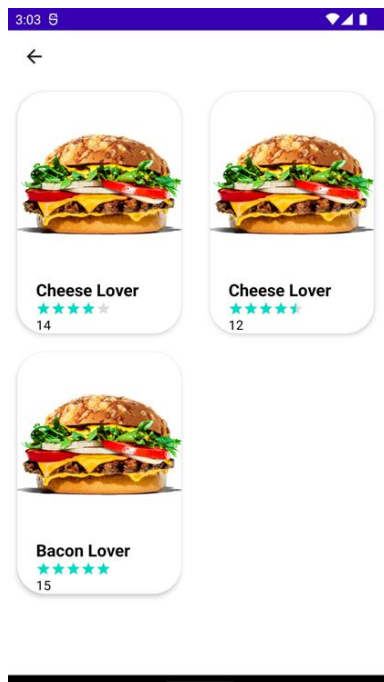


Figure 4-1-11 Foods by Category

Figure 4-1-11 shows the list of foods by category. There are four categories on the Main Page; when the customers choose one it will redirect them to the screen contains all the foods that have the same category as the one customer selected.

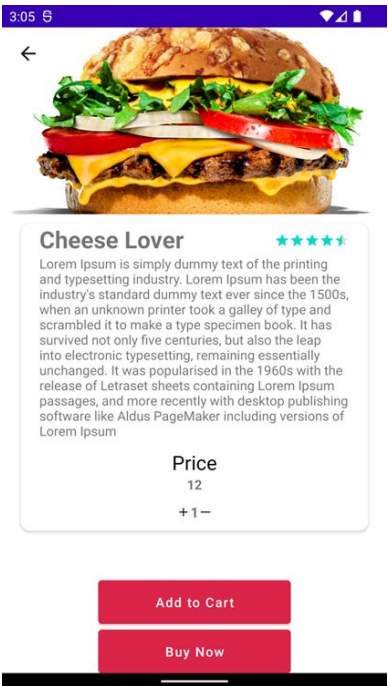


Figure 4-1-12 Food Details

Figure 4-1-12 shows the details of the food. Customers can increase or decrease the amount of food by clicking “+” or “-”. “They can also move to the cart to see their chosen food or choose “Buy Now” to redirect straight to the payment page.

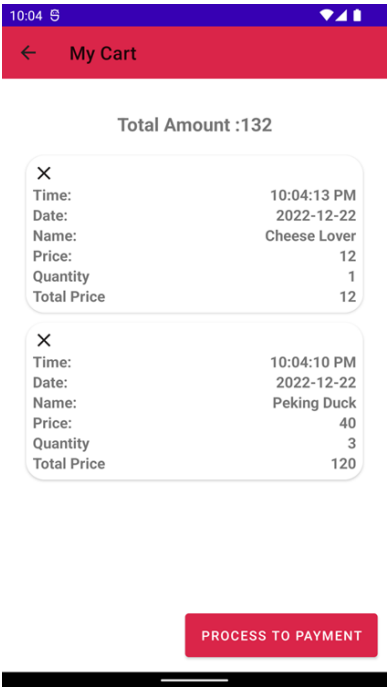


Figure 4-1-13 My Cart

Figure 4-1-13 shows all the information about the items added to the cart which contain time, date, price, quantity and total price based on the quantity. This screen also contains total price if customers choose multiple foods.

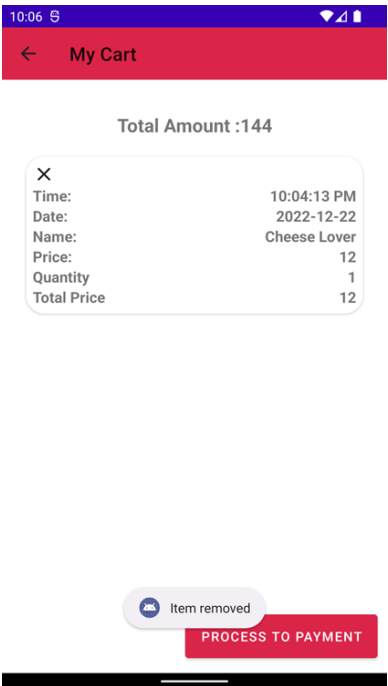


Figure 4-1-14 Remove Item in Cart

Figure 4-1-14 shows that customers can delete their food item after placing it into the cart. There is an icon “X” on each food item in the Cart and if customers want to delete the food item, they click on the icon and the food item will disappear and a message will appear to notify to customers that the food was removed.

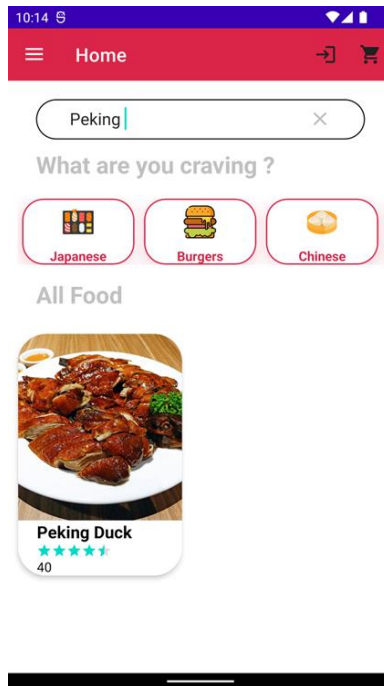


Figure 4-1-15

Figure 4-1-15 shows the screenshot about how the Search Engine in the application works. When customers need to find their desired food, they can type the food name in the Search Engine and the food that contains characters or name will appear under “ALL FOOD”.

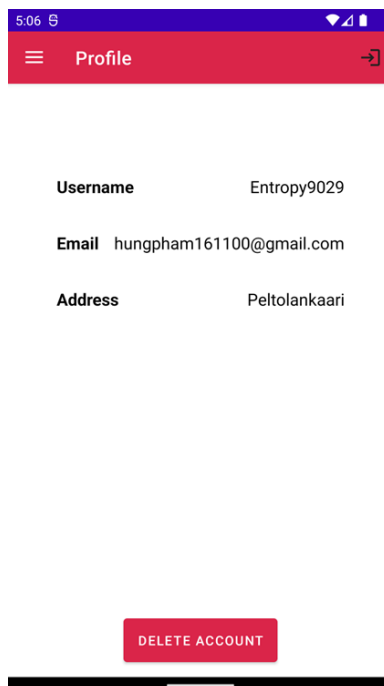


Figure 4-1-16 Customer's Profile

Figure 4-1-16 shows the Screen that contains customer information. This Screen will include information about the customer's username, email and address. This Screen also has a button to delete the customer's account. When clicking ~~to~~ the button, there will be a question if the customer wants to delete their account or not.



Figure 4-1-17 Discount Page

Figure 4-1-17 shows information about discount food based on the type of meals. There will be three types of meals in this Page which are: Breakfast, Lunch and Dinner. Under each type of meal name, the amount of discount is shown.

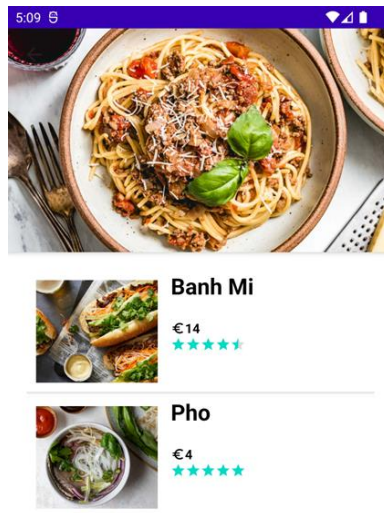


Figure 4-1-18 Item in Discount Page

Figure 4-1-18 shows the food with a discount. This Screen contains all the foods that have a discount and when clicking ~~to~~ the food image, it also shows the food details which are the same with Food Details Page.

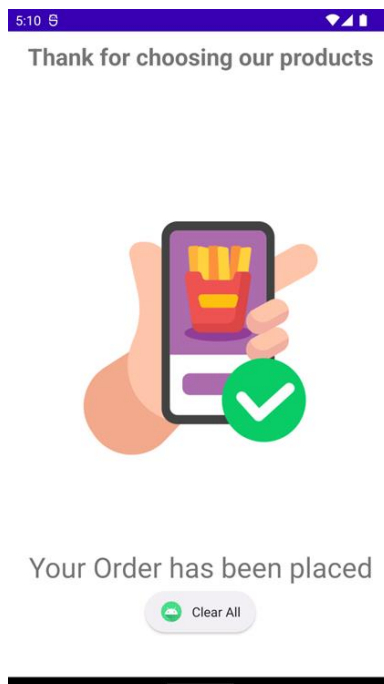


Figure 4-1-19 Order Placement Page

Figure 4-1-19 shows information about the order confirmed in the database. After selecting all the desired foods in the cart, customers can click to the button “**Process To Payment**” which can be seen in the Figure 4-1-13 and Figure 4-1-14 and then they are redirected to the Order Confirmation Page. In this Page, the customer’s order will be added to database and all the food items in cart will be cleared.

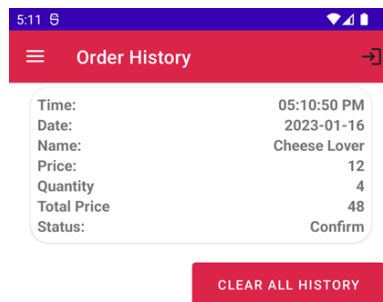


Figure 4-1-20 Order History Page

Figure 4-1-20 shows information about the user’s order history. After processing the payment in the Order Confirmation Page, the order will be added to the database and also to the Order History Page so customers can see the ordered food items. This Screen also contains a button to delete all the history if customers do not want to see their order history anymore.

4.2 Database design

4.2.1 Firebase Realtime Database

Firebase Realtime Database is a cloud-hosted NoSQL database used to store and sync the data between connected users in real-time. The data in Realtime Database is stored in JSON format and can be synced with any platform such as Android, Web, and IOS.

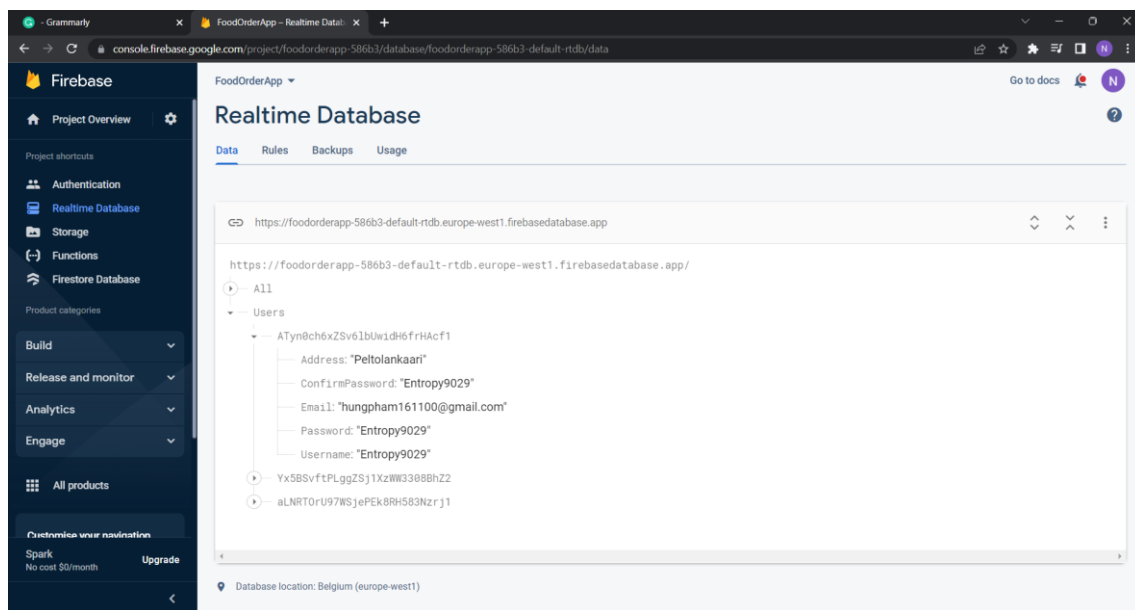


Figure 4-2-1-1 User Database in Firebase Realtime Database

For the server side of the project, Firebase Realtime Database was used for storing user information after registering for the application. The user's database contains the user's ID, which is a unique number, and inside each ID the five values are Username, Email, Address, Password, and Confirm Password.

4.2.2 Firestore Database

Firestore Database was developed based on the success of the Firebase Realtime Database. Firestore is an upgraded copy of the Realtime Database. It is also a NoSQL database, and it can store and process data in real-time. Besides, and it can be used on many platforms. The strength that makes Firestore different from its predecessor is that now the model and the way the data are processed have become more accessible.

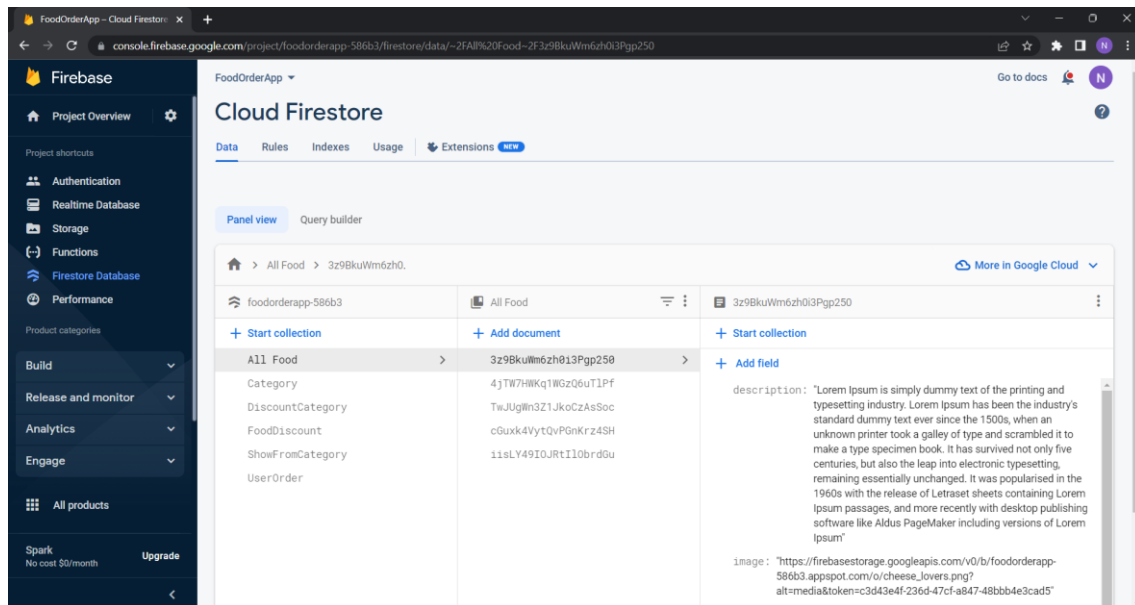


Figure 4-2-2-1 Data in Firestore Database

In this project, Firestore Database was used for storing data related to Food and Cart. The item's database contains information about food and its category. Additionally, the database holds information about the user's Cart.

4.2.3 Authentication and Storage

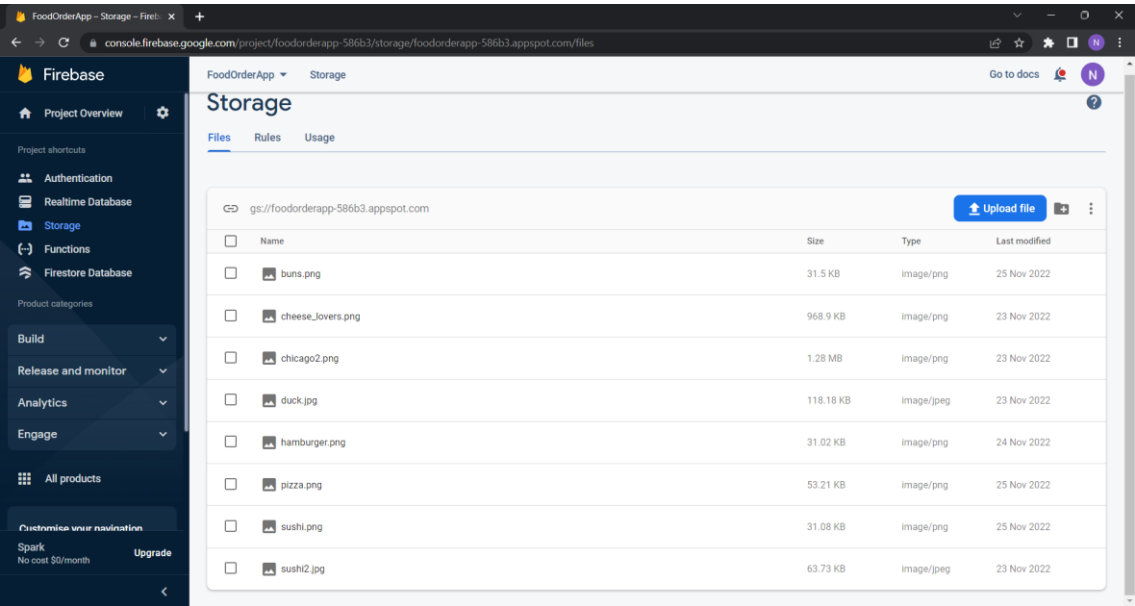


Figure 4-2-3-1 Image Storage

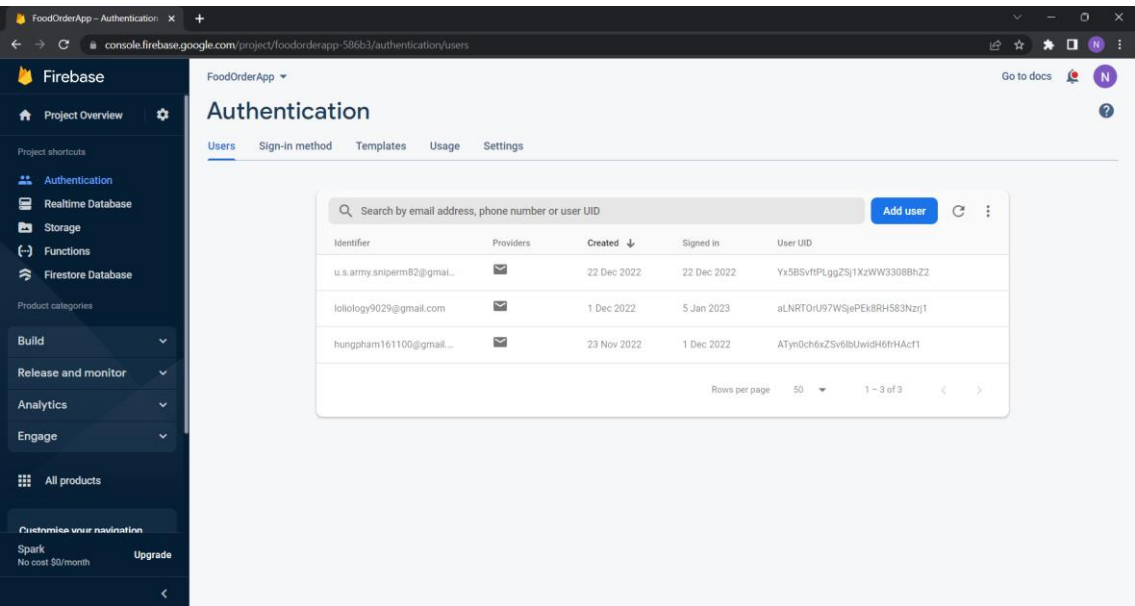


Figure 4-2-3-2 Authentication

Figure 4-2-3-1 is a screenshot of where all the images of the foods are stored for the Firestore Database to use in adding or modifying food images. And Figure 4-2-3-2 is a photo of all verified users connected to the application.

5 TESTING

After finishing the design phase, the implementation phase will begin. In this phase, the database structure is first designed and built. Besides, the client side is also developed to create initial links between the client side and the database (server side). This application's project follows the evolutionary model, so the unit testing of each function will be tested continuously to minimize the bugs and errors in the project.

5.1 Development Tools

To get this project done in a complete way as well as with stable performance and the ability to follow the model proposed at the beginning of the thesis, the implementation phase and testing phase were done together with the support of some development tools. For the implementation phase of the client side, Android Studio was used as a workplace and a development environment for the project. Besides, the Android SDK supported providing necessary API libraries and tools to handle the code. Parallel with that, in the testing phase, Android Virtual Device (AVD) and an actual device (Huawei phone) were used to test each function to find errors and bugs. The programming language primarily used in this project is Java. For server-side implementation, almost all features of Google Firebase were used to deal with the data.

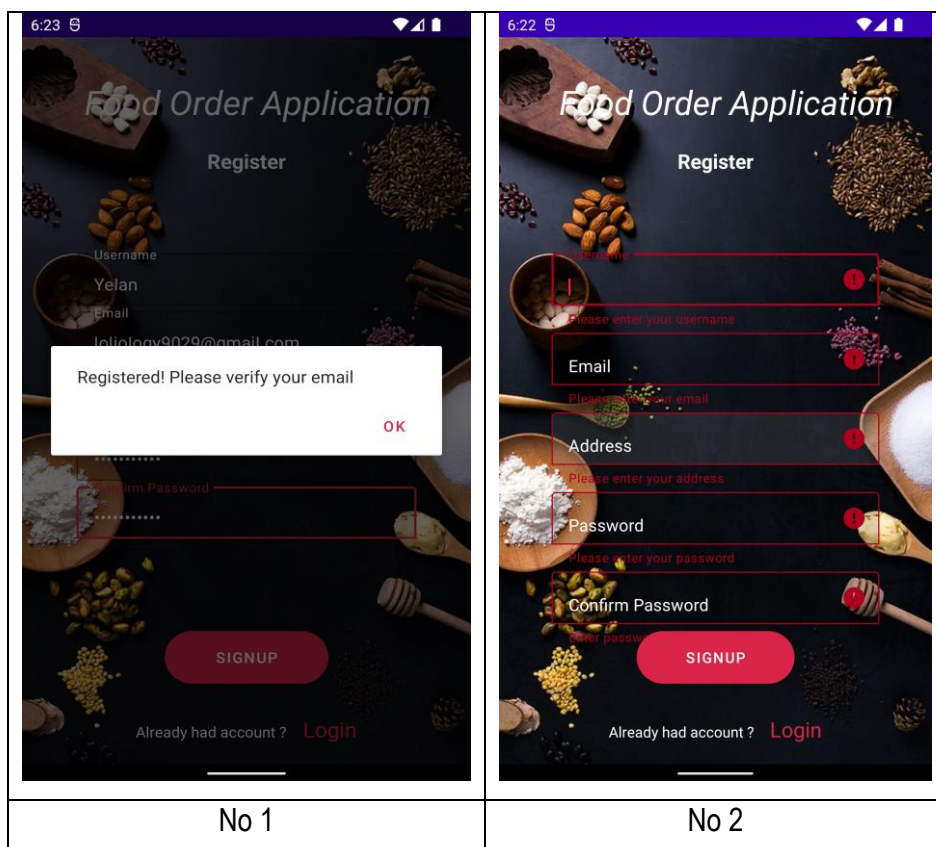
5.2 Unit Testing

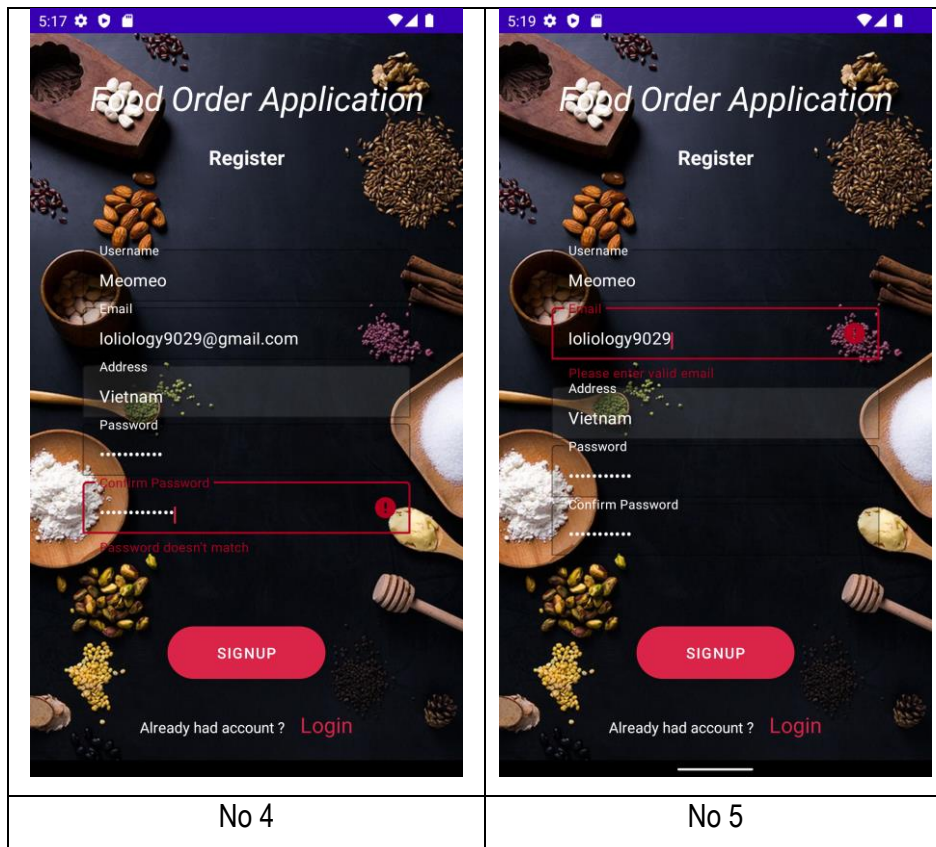
To test this application, unit testing will be used. Using this testing, a developer can find out bugs or errors that can affect the application's performance. Below are some completed parts have been tested:

Unit testing 1: Sign up as a new user.

Testing Objective: Guarantee that a new user is created and verified successfully.

No	Description	Expected Result	Final Result
1	Users fill all the information blank on Register Page to create a new user.	Showing a message that a new user has been created	Successful
2	Users don't fill in any information or miss some blanks on Register Page	Showing a message below the missing blank to notify the user to fill in	Successful
3	Users verified their email for Login Page.	Users receive an email sent from firebase to verify their email.	Successful
4	In the Confirm Password blank, users fill in the password that does not match a password in the Password blank.	Showing a message below the Confirm Password blank to notify the user to fill in the matching password.	Successful
5	In the Email blank, users fill in the wrong email format.	Showing a message to notify users that their email is invalid	Successful

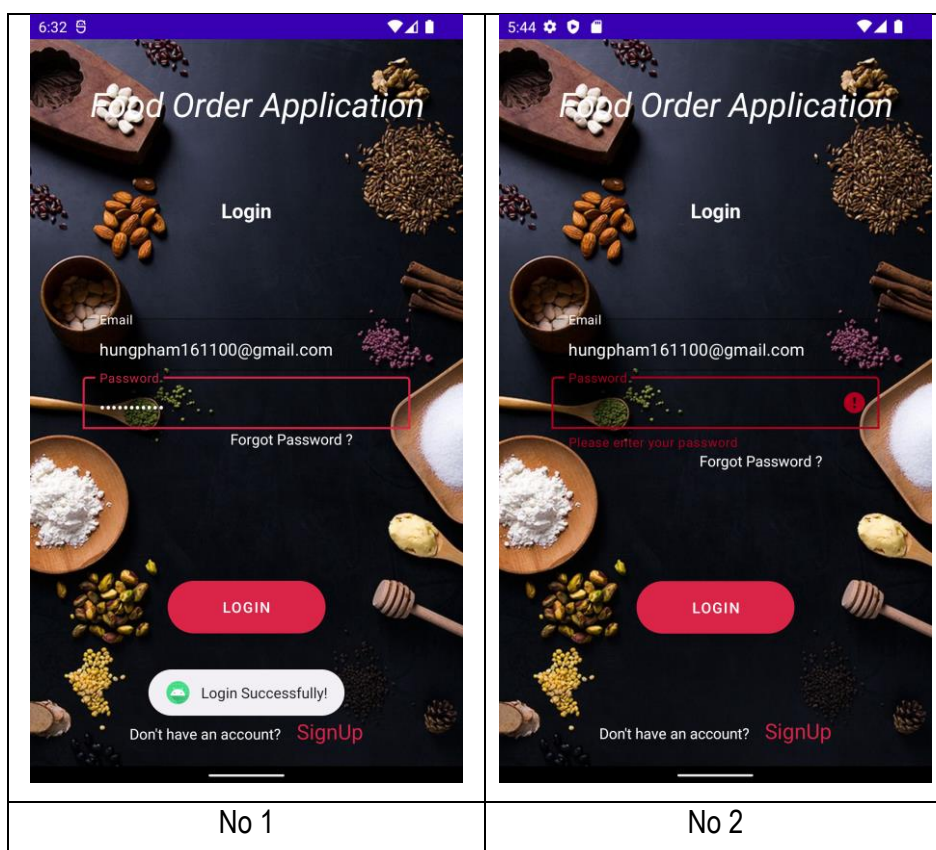


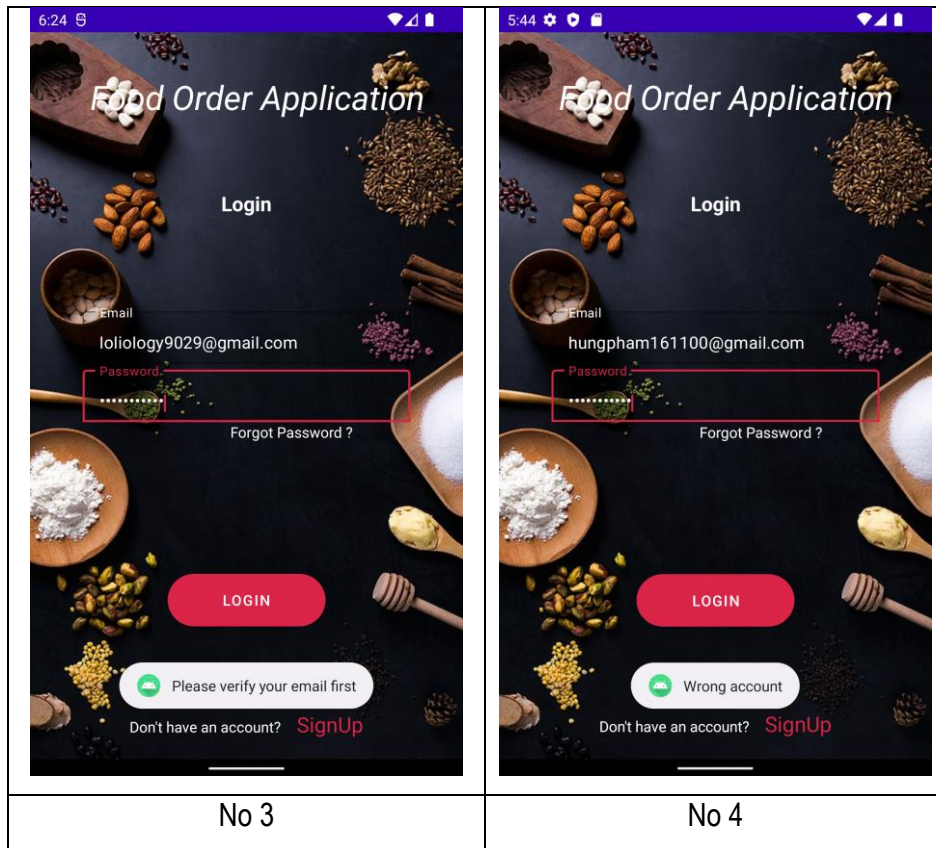


Unit testing 2: Login as a user.

Testing Object: Users can log in using their information and enter the Main Page successfully.

No	Description	Expected Result	Final Result
1	Users fill all the information blank on Login Page to enter the system	Showing a message that the login is successful	Successful
2	Users don't fill in any information or miss some blanks on the Login Page	Showing a message below the missing blank to notify the user to fill in	Successful
3	Users log in but have not yet verified their email.	Showing a message to notify users to verify their email first	Successful
4	Users log in but use the wrong information.	Showing a message to notify users that their information was wrong	Successful


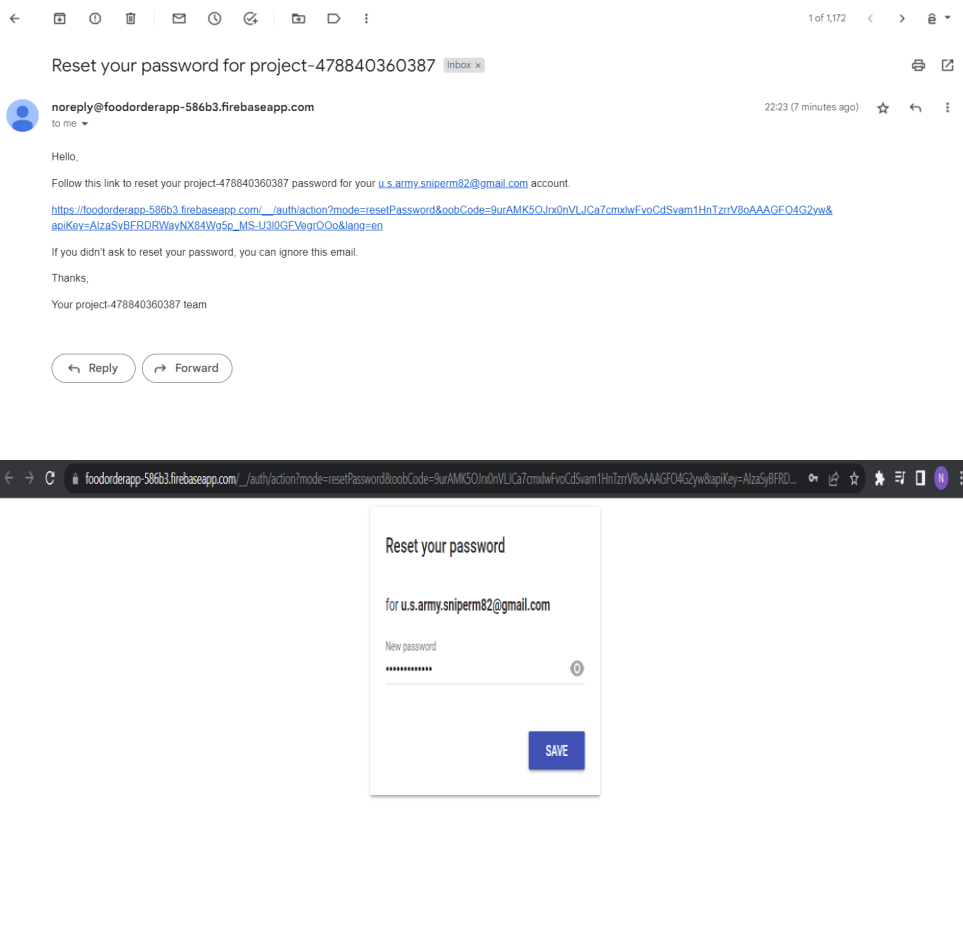




Unit testing 3: Reset Password

Testing Object: User can reset their password in case of forgetting.

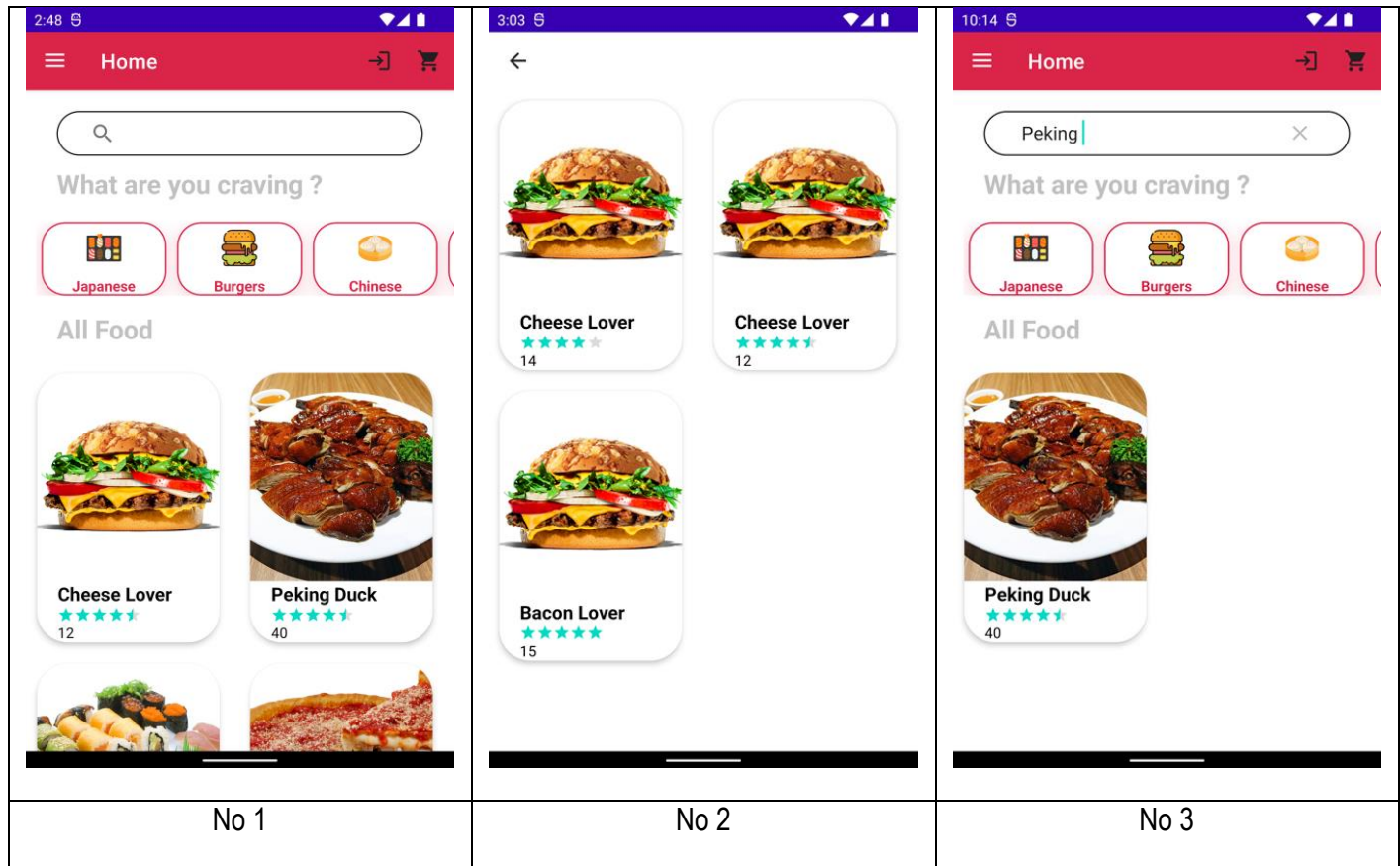
No	Description	Expected Result	Final Result
1	Users do not fill the Email blank in Reset Password Page	Showing a message that users need to fill in their registered email.	Successful
2	Users reset their passwords with emails.	Users received an email with a link to change the password.	Successful

	
No 1	No 2

Unit testing 4: Finding foods.

Testing Object: Users can see all the foods on the Main Page, they can choose by category, or use the search engine.

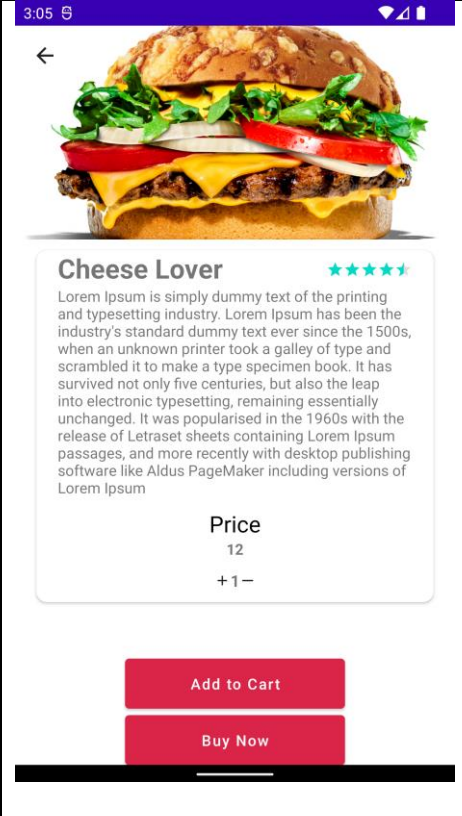
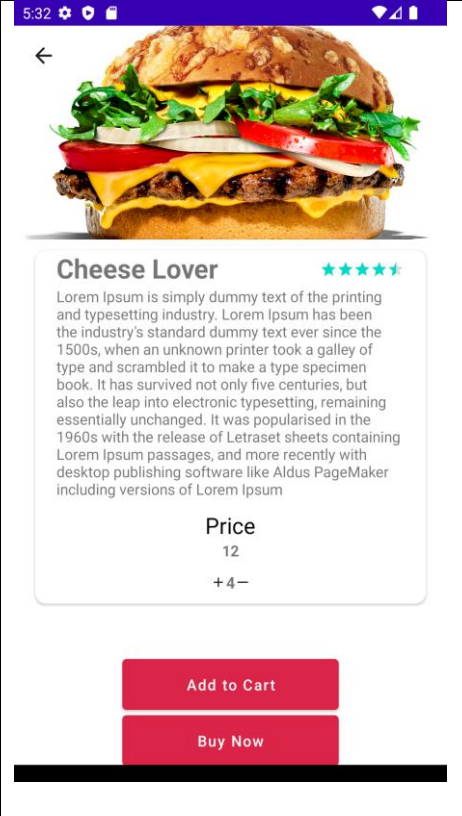
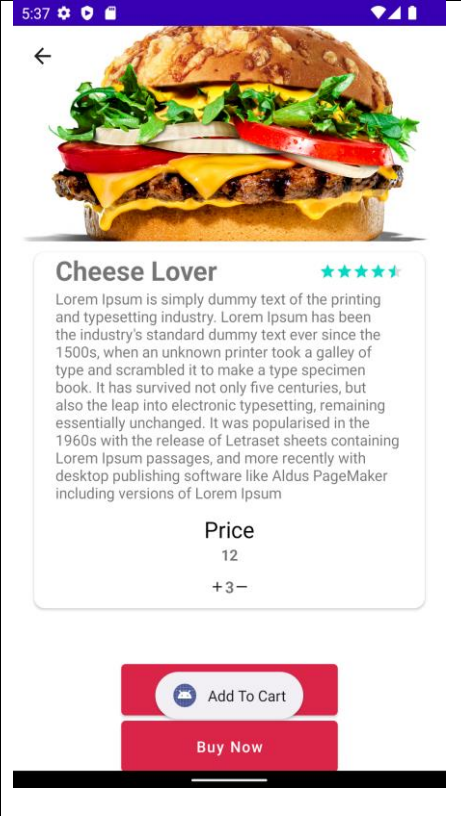
No	Description	Expected Result	Final Result
1	Users see all the food on the Main Page	All food images appeared on the Main Page	Successful
2	Users find their desired foods by category.	Users click on the category, and food lists by category will appear.	Successful
3	Users find their desired foods by using search engine	Showing foods filtered by search engine	Successful



Unit testing 5: Choose foods and add them to the cart.

Testing Object: Users can see the details of the foods, increase or decrease the quantity of food and add it to the cart.

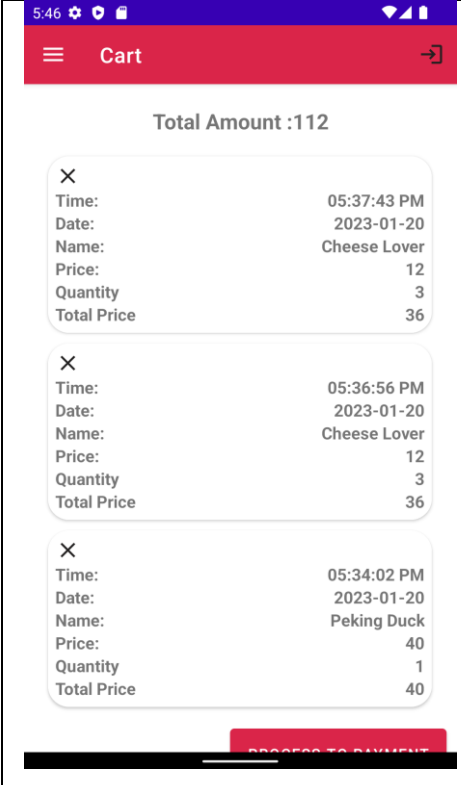
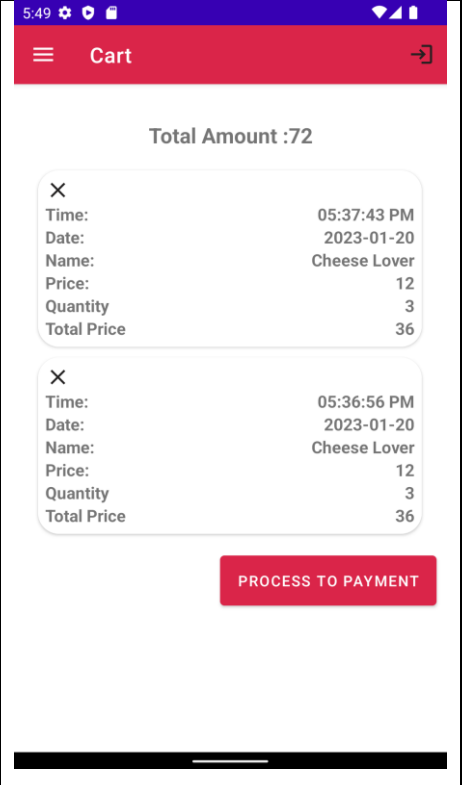
No	Description	Expected Result	Final Result
1	Users see the food details	A page shows the details of the food	Successful
2	Users can modify the quantity of food.	The amount increases from 1 to 2 when clicking on "+" and the opposite when clicking on "-".	Successful
3	Users can add the food to the cart.	Showing a message to announce to users that food has been added to the cart	Successful

		
No 1	No 2	No 3

Unit testing 6: Modify cart.

Testing Object: Users can see and modify all foods they add to the cart.

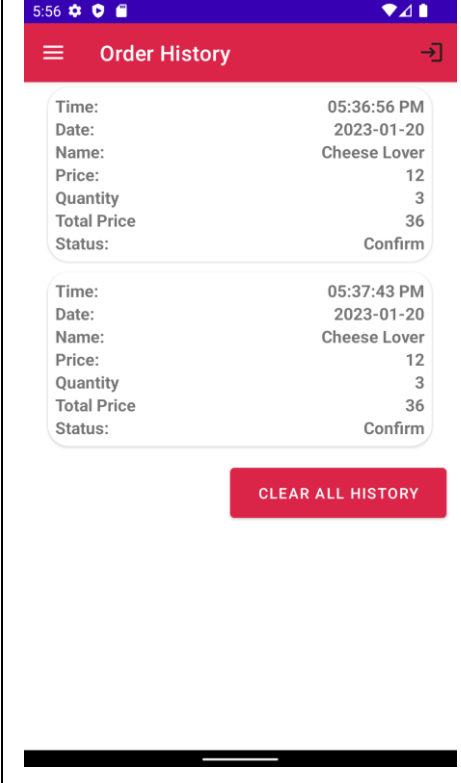
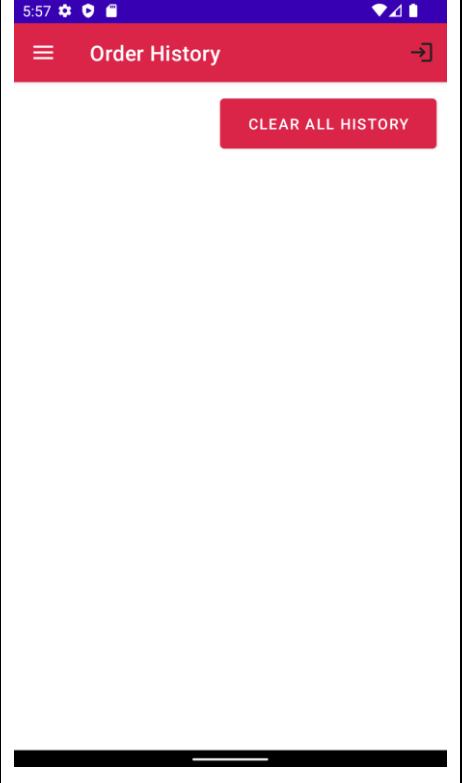
No	Description	Expected Result	Final Result
1	Users see all foods that they added to the cart and the total price of it	Showing the basic information of added food and calculating the total cost on the Cart Page	Successful
2	Users delete foods that they do not want and update the total price	The food selected for deletion will disappear from the page, and a new full price will appear	Successful

	
No 1	No 2

Unit testing 7: Check order history.

Testing Object: Users can see all the orders they have already made.

No	Description	Expected Result	Final Result
1	Users see all the order history.	Showing all the orders that users have already made	Successful
2	Users can clear all the order history	All order history will disappear from the Page	Successful

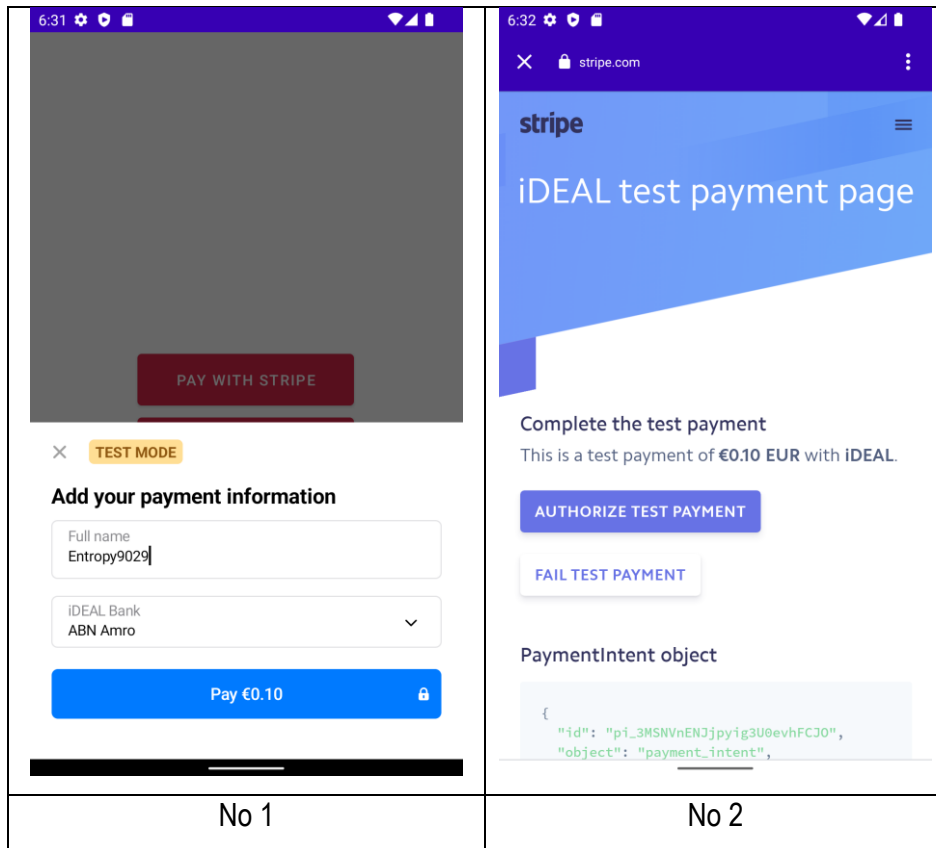
	
No 1	No 2

Unit testing 8: Check Out

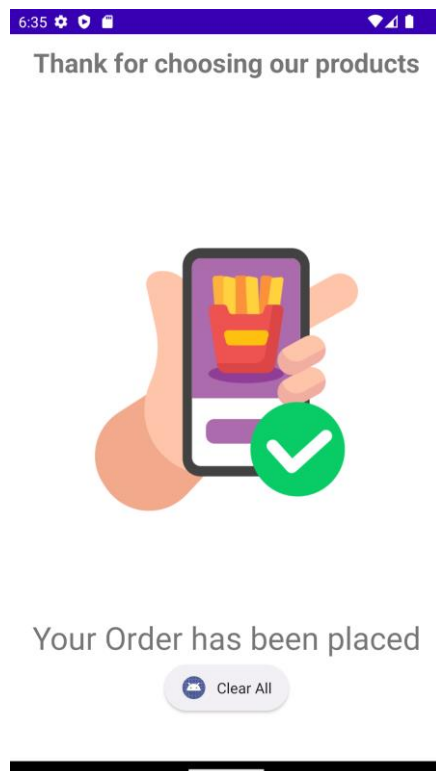
Testing Object: Users can make payments with Stripe payment.

No	Description	Expected Result	Final Result	Current status
1	Users can check out using the Stripe payment gateway.	A virtual invoice will appear in the Stripe management dashboard	Not successful	Manage to make a test payment but cannot yet connect to the User payment

Current Status for Payment Gateway:



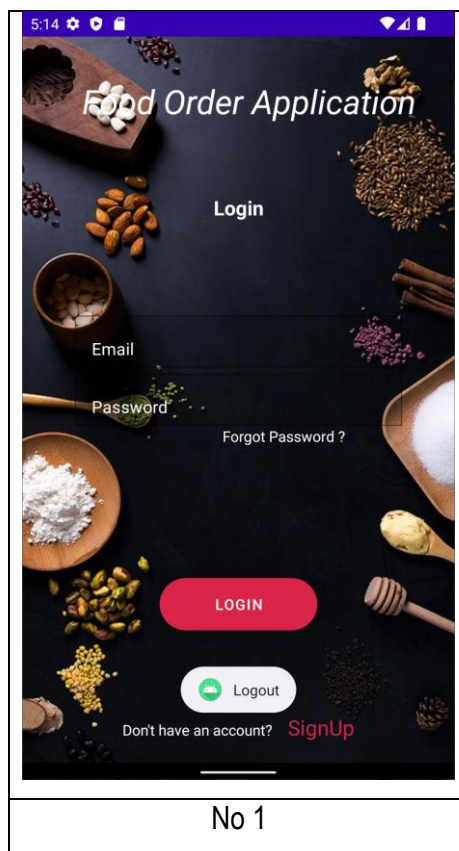
Replacement: A Place Order successfully page was created to temporarily replace the payment gateway, which is now under development.



Unit testing 9: Users Logout

Testing Object: Users can log out of the system.

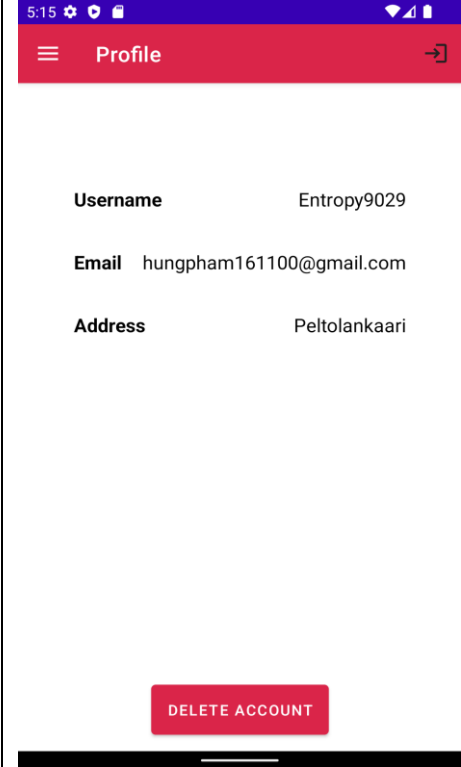
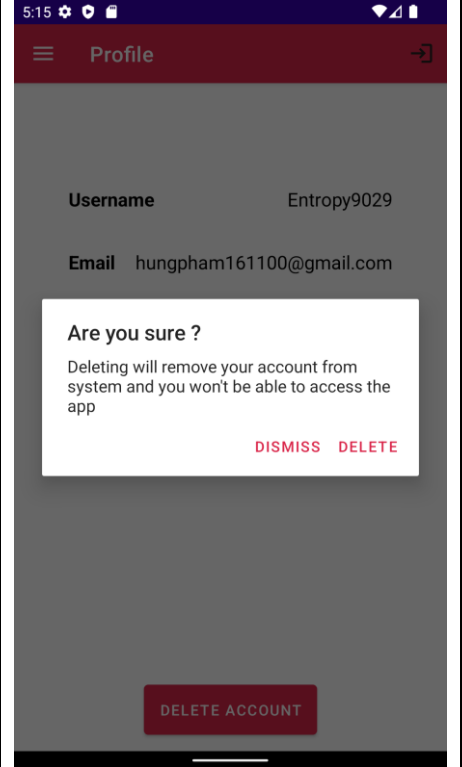
No	Description	Expected Result	Final Result
1	Users can log out of the system whenever they want	Users click on the sign-out icon on the top right of the application, and a message shows that users log out successfully.	Successful



Unit testing 10: Users Profile

Testing Object: Users can see their profile and delete their account if they no longer use the application.

No	Description	Expected Result	Final Result
1	Users can view their profiles.	Information about Username, Email, and Address will appear on Profile Page.	Successful
2	Users can delete their account if they no longer use the application	Showing a dialog for the user to choose to delete their account or not	Successful

	
No 1	No 2

6 SYSTEM EVALUATION

The system provided the most basic features of a food ordering application which contains User Authentication (Login, Registration and Verification), Dashboard Page, Search Engine, Profile Management, Order Management (Add to Cart, Modify Cart, Place Order, Check Order History), Checkout. The features mentioned are consistent with the project scope outlined in the beginning of the thesis. In this chapter, the system will be assessed and from there planning for future developments.

6.1 Evaluation of system status

Till now, features which are User Authentication, Main Screen, Search Engine, Profile Management, Order Management and Checkout have all been built, run and tested successfully except Checkout which was now still under development. Most features are tested on AVD (virtual device) and real device (Huawei phone) to make sure that there are no problems, errors or bugs as well as having a good UI. According to the unit testing in Chapter 5, almost all features can run and give notifications to the user if there is an error or bug.

Furthermore, currently there are also some limitations in the system. The system is built under Android environment so only users with Android phone can use it. Moreover, the system still lacks many features that can make the application more complete such as Notification, Tracking system, Rating and Promotion Codes/Discount.

6.2 Future Enhancement

1. Payment Checkout

The completion of the payment system is at the forefront and will be further developed. More payment methods will be developed and updated to the application for better convenience to the users.

2. Improve existing features and Development other important features.

The current features can work stably but the connection as well as the way it works is not completely optimized to be used in business. Besides, features such as Notification, Tracking System, Rating, Review and Promotion will be further investigated and developed in order to improve the interaction between users and systems.

7 CONCLUSIONS

First, general assessment of the effectiveness of using the native solution in developing a food ordering system on Android devices in this project is discussed. Second, the importance of choosing a solution in developing applications is reviewed. Finally, a prediction or thought about future of Q-commercial in general as well as in mobile development in particular is made.

Regarding the effectiveness of using the native solution in developing a food ordering system on Android phones in this project, one of the most vital points is that the performance and scalability seem pretty good. Besides, being able to run on multiple AVD and real devices simultaneously makes brainstorming and designing UI even better. Moreover, if there are any bugs or errors or a need to update a new feature, the native solution provides quickness and simplicity in handling. For data processing, with the support of Google Firebase, it is easier to store and use data in the database.

In an era where everything moves so fast, making the right choice can bring significant benefits; it saves time and money and helps follow the right path thereby minimizing the possibility of being outdated in technology. Combining the information from the chapter above with the effects shown by using the native solution in this project, it is essential to define a solution to start developing a mobile application. It is necessary not only for the beginner but also for experts.

“Every cloud has a silver lining” – this idiom can be used to describe the rapid growth of Q-commercial. The corona pandemic caused most people to be isolated and quarantined which made shopping harder than ever, and this spurred the growth of Q-commerce. Getting acquainted with Q-commerce during the two years of the pandemic is enough to make it an indispensable part of human life. In the coming years, Q-commerce will continue to grow strongly and open up more categories such as pharma and electronics to meet human needs (Tanya Kakaria, 2022). This phenomenon will have a massive impact on mobile app development and combined with the growth of 5G and AI technologies, there will be a bright future for mobile app development in upcoming years (Perfect Learning, n.d.).

REFERENCES

(1) deepaman07ad, 2021. What is the Impact of E-Commerce on the Society?. Search date 25.01.2023.

<https://www.geeksforgeeks.org/what-is-the-impact-of-e-commerce-on-the-society/>

(2) Mobile Operating System Market Share Worldwide. Search date 05.01.2023.

<https://gs.statcounter.com/os-market-share/mobile/worldwide>

(3) Daniel Jilg, 2022. iOS versions market share. Search date 10.01.2023.

<https://telemetrydeck.com/blog/ios16-market-share/>

(4) Omkar Kumbhar, 2022. iOS Architecture. Search date 04.12.2022.

<https://redfoxsec.com/blog/ios-architecture/>

(5) XCode. Search date 25.01.2023. <https://developer.apple.com/documentation/xcode>

(6) AppCode Features. Search date 25.01.2023 <https://www.jetbrains.com/objc/features/>

(7) The Editors of Encyclopaedia Britannica, 2020. Android operating system. Search date 08.12.2022. <https://www.britannica.com/technology/Android-operating-system>

(8) Mobile Operating System Market Share Worldwide. Search date 25.01.2023.

<https://gs.statcounter.com/os-market-share/mobile/worldwide>

(9) Praveenruhil, 2021. Android Architecture. Search date 08.12.2022.

<https://www.geeksforgeeks.org/android-architecture/>

(10) Android Studio. Search date 08.12.2022. <https://www.javatpoint.com/android-studio>

(11) Eclipse (software). Search date 08.12.2022. [https://en.wikipedia.org/wiki/Eclipse_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software))

(12) Design Rush,2022. The Ultimate Guide TO Hybrid App Development. Search Date 10.12.2022.

<https://www.designrush.com/agency/mobile-app-design-development/trends/hybrid-app-development>

(13) O'Reilly, 2015. Learning React Native. Search date 10.12.2022.

<https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>

(14) Feza Roheel. What are the features of Flutter. Search date 10.12.2022.

<https://www.educative.io/answers/what-are-the-features-of-flutter>

(15) Rohin Tak, Jhalak Modi, 2018. Mobile DevOps. Search date 10.12.2022.

<https://www.oreilly.com/library/view/mobile-devops/9781788296243/72473bdb-ac6f-457f-9f10-63d252f63c9f.xhtml>

(16) Dave Francis,2020. Is Xamarin Dead?. Search date 10.12.2022.

<https://foresightmobile.com/blog/isxamarin-dead#:~:text=Well%2C%20it's%20not%20quite%20dead,MAUI%20%2D%20Multi-form%20App%20User%20Interface.>

(17) Good barber website. Search date 11.10.2022. <https://www.goodbarber.com/>

(18) Shoutem website. Search date 11.10.2022. <https://shoutem.com/>

(19) Tanya C Kakaria, 2022. What's next for Q-commerce: Trends & future ahead. Search date 25.01.2023.

[What's next for Q-commerce: Trends & future ahead \(indiatimes.com\)](https://www.indiatimes.com/What's-next-for-Q-commerce-Trends-&-future-ahead/)

(20) Perfect Learning. The Future of Mobile App Development: Emerging Technologies, Trends, and Advancements. Search date 13.02.2023

[The Future of Mobile App Development: Emerging Technologies, Trends, and Advancements | Perfect eLearning](#)