Pabile: A Cross-Platform App for the University of Mindanao Food Stalls using a Hybrid algorithm for Ordering Application

by Barrientos, Batingal, Baran, Espadero

Submission date: 25-Jul-2023 05:33AM (UTC+0800)
Submission ID: 2136272304
File name: Barrientos,_Batingal,_Baran.docx (3.13M)
Word count: 7039
Character count: 39308

Pabile: A Cross-Platform App for the University of Mindanao Food Stalls using a Hybrid algorithm for Ordering Application

Thomas Jon Barrientos
BS Information Technology
University of Mindanao
Davao City
t.barrientos.516698@umindanao.edu.ph

Aecl Batingal
BS Information Technology
University of Mindanao
Davao City
a.batingal.517873@umindanao.edu.ph

Edlian Paul Baran
BS Information Technology
University of Mindanao
Davao City
e.baran.517207@umindanao.edu.ph

Jenny A. Espadero, MSIT
Faculty, College of Computing Education
University of Mindanao
Matina Davao City, 8000
jespadero@umindanao.edu.ph

Categories and Subject Description
Cross Platform → Information Technology → Merchant App

General Terms
The general terms are the following: Algorithm, Design, and Documentation.

Keywords
Pabile. Popularity, Personalization, and Prioritization algorithm for ordering application, UM Food Stalls, ordering application, Food Delivery.

1. INTRODUCTION

1.1 Project Context

Food delivery services is one of the emerging, and fast-growing sectors in the food market. It is a way to eat out without having to go out in the comfort of our homes and offices. Regardless of whether the consumer prepares the meal themselves, dines out, or purchases food to bring to their homes, offices, or other locations, it relieves them of the stress of having to think about and arrange meals. [1].

The University of Mindanao has its own canteen on campus. People can choose from a range of cuisines sold at food vendors. Food vendors serve a wide range of cuisines, from main courses to beverages and desserts. When a student, teacher, or member of the campus staff visits, the dedicated food sellers make sure they have a pleasant eating experience by serving them fresh and excellent meals. Each booth has a variety of high-quality equipment to store their dishes. One of the places where students may relax and enjoy varied food selections is the canteen and food vendors.

University of Mindanao's food stall attendants are people who rent stalls to sell their dishes, beverages, and sweets. Staff within the food stalls are obliged to pay their rent. Stall owners have the responsibility of managing their stalls, which includes setting up displays, keeping sanitation and hygiene requirements, and assuring the quality of their dishes, beverages, and sweets. They also handle customer relations, taking orders, and serving meals, ensuring that the University of Mindanao community has a nice dining experience. But due to the hassle it would take them only to eat inside the canteen and food court, it somehow changed into an unwanted experience inside the canteen. The Food stall owners at the University of Mindanao may encounter numerous problems. For starters, they may experience changes in customer demand, which could lead to uncertain sales and financial instability. Second, rivalry among different food stalls within the campus grounds might make attracting and retaining consumers difficult. Finally, for stall owners, acquiring and maintaining supplies while keeping expenses under control can be a constant challenge.

The University of Mindanao is currently dealing with a significant issue with its congested food court, particularly in the buildings located far away from the food court. Surveys consistently show that 79% of respondents had difficulty finding a place to dine in the food court. 84% of respondents also went to the food court and were dissatisfied since the dish they wanted was not available or the stall itself was closed. Furthermore, 84% of respondents agreed that having an ordering application to deliver the order to the customer from the food court saves them time instead of going back and forth to the food court.

The major limitation of the study is a cross-sectional study on the food delivery apps (FDAs) ongoing usage intentions in the context of a health and safety crisis, confining it to the context of continued usage intentions during a health and safety crisis [4]. Restaurant owners and operators should prioritize Internet sales via mobile food ordering apps (MFOAs). During the pandemic phase, social isolation is a major approach to managing COVID-19. Customers should be informed and ensured that safety procedures are undertaken while delivering the meal [5]. Food delivery applications are used by many people due to these various reasons namely, convenience, customer experience, delivery experience, ease of searching restaurants, and the of use [14].

The development of online food delivery services helps customers to hire delivery workers to pick up the food on their behalf [2]. The landscape of the food sector has changed because of food delivery. With the constant advancements in technology, online food delivery services have progressively increased over time. They changed from telephone ordering to digital ordering that meets customer needs [3]. Teachers, staff, and some students at the University of Mindanao use food ordering applications to eliminate the inconvenience and increase efficiency so they can use their time

more effectively. Grab and Food Panda includes food ordering and delivery services to their customers in nearby restaurants and delivers it to them by implementing a location-based system. It is an on-demand food delivery service designed to satisfy your hunger as conveniently as possible. [12][13].

The researchers proposed study solution includes developing a Cross-Platform Application called "Pabile" for the University of Mindanao Food Stalls that integrates a hybrid algorithm consisting of popularity, personalization, and prioritization for the ordering application. will lessen the hassle and challenges the students, teachers, and staff often experience in the university's canteen and food court. Pabile was proposed with the same premise of providing ease to clients to fulfill their needs and deliver as much efficiency as each of these apps can supply. Concentrating only on the school grounds to create a far better experience for ordering and purchasing food on the school grounds.

This study aims to enhance the food stalls of the University of Mindanao by implementing a Cross-Platform App with a hybrid algorithm for the ordering application. The Students and Faculty members can order food from a variety of campus locations, saving time and effort by removing the need to physically travel to the dining facilities.

Mobile food ordering apps (MFOAs) have grown in popularity in the restaurant business as a means of increasing client reach while also providing high-quality services. However, the influence of MFOA implementation on customer satisfaction and intent to reuse them remains a topic of discussion. The purpose of this study is to discover and empirically examine significant elements impacting MFOA satisfaction and consumer intention to reuse them in Jordan. The study collected data from a convenience sample of Jordanian customers who had used MFOAs using the expanded unified theory of acceptance and use of technology and MFOA features such as online reviews, ratings, tracking, performance expectancy hedonic incentive, and price value in affecting e-satisfaction and intent to use MFOAS. This study contributes to theory and offers practical implications for academics and practitioners in the MFOA domain.[16]. Implementing two-factor authentication (2FA) with OTP and password in our Pabile application adds an extra layer of security by requiring users to give authentication with both something they know (password) and something they have (OTP). This method reduces the chance of unwanted access even if the user's password is hacked, making the app more secure and the user account safer. The Pabile app for the University of Mindanao food stalls is personalized and improves users' experience through content-based filtering. Based on previous orders, the app analyzes food items and delivers personalized recommendations. By ensuring customers discover and enjoy meal alternatives that match their specific interests within the app, this personalized method saves time. Individual apps for every platform, including an Android app, an iOS app, and a web app, must be created using conventional development techniques. Using cross-platform development significantly lowers the cost, time, and effort of developing applications. [6]. Frameworks for cross-platform development continue to be crucial in the creation of mobile applications. There is an abundance of frameworks that help with the development of cross-platform applications, which proves to be much more advantageous to many developers.[7]. The recommended system can efficiently alleviate APP's information overload. The O2O takeaway restaurant recommendation system is thoroughly examined in this research. Using an examination of the current state of the O2O takeaway restaurant recommendation system.[8]. Food delivery apps have grown rapidly during the last decade. Digital ordering accounts for half of all food delivery visits, expanding beyond traditional dinner delivery to include breakfast and lunch delivery in most cases.[9]. This study reports on an exploratory inquiry aiming at identifying consumer categories using data gathered from meal delivery app users. Consumer views of the importance of food delivery app service quality attributes were used to identify and construct food delivery app user categories. The latent class model (LCM) was used to categorize food delivery app users into homogeneous groups based on food delivery app quality parameters such as usefulness, convenience, design, trustworthiness, pricing, and variety of meal options.[10]. Its goal is to improve the university canteen and food court experience of the students, teachers, and staff that are currently working and studying on the University of Mindanao campus.

With the large number of students enrolled at the university, it can sometimes be difficult for students to find a seat or table to eat at or to purchase the food they want because the stall they wanted to buy it from is closed or the food is currently unavailable. This study aims to help students, teachers, and staff make better use of their time by eliminating the need for them to travel from their rooms to the canteen.

Table 1: Review of Related Studies Comparison

Features	Grab	Food Panda	Maxim	McDelivery PH	Pabile
Communication	✓	✓	✓	✓	✓
Tracking	✓	✓	✓	✓	✓
Rating and Feedback Recommendation		✓		✓	✓
Two Factor Authentication		✓			✓
Content-Based Filtering					✓
Online Payment using g-cash	✓	✓		✓	✓

1.2 Purpose and Description

The purpose of this research is to aid students, teachers, and staff the attendees of the University of Mindanao. When it comes to meals at the University of Mindanao, Pabile consists of providing a food delivery app that will help the students, teachers, and staff to effectively use their time rather than going to and from the

school canteen and food court. Providing an online delivery app that will also help students with long-vacant time to become food deliverers of our application, to have the students select the food they love that exists in the canteen by the touch of their hand upon using the application is also a part of our app features.

The customers will be the students, teachers, and staff, and the couriers will be the staff or students who can also do part-time work by delivering food in their spare time. Customers have the option of exploring menus and selecting their preferred food. They can tailor their orders by including unique instructions. Reviews and ratings can be offered to improve the overall dining experience and display the highest rating on the users' home page. Payments can be made conveniently by online payment (G-cash) or cash on delivery. Customers can check the progress of their order and the delivery status in real-time. Stalls have a real-time update on their menus and offerings on the site, keeping them up to date on availability and pricing. They receive and process orders quickly and engage with clients to handle any order-related concerns. Courier Functions ensure order delivery by accepting requests and picking up orders from booths. They guarantee safe and fast delivery to consumers' locations, collect payments, provide receipts, and track customers. Couriers keep consumers informed of any concerns or delays during the delivery process by using in-app messaging options. On the admin side, they have the function to check courier forms, student forms, and staff. The admin can also approve or disapprove users who register as a courier and check whether the registered courier is fit for being a courier.

1.3 Objectives

1.3.1 General Objectives

The primary objective of this study is to develop a Cross-Platform App for the University of Mindanao food Stalls using a Hybrid algorithm for Delivery apps.

1.3.2 Specific Objectives

To achieve the general objective, the researchers will develop a system that specifically:

1.3.2.1 To create an ordering application that utilizes prioritization and popularity algorithms.

1.3.2.2 To receive/deliver the orders of the users by utilizing the priority algorithm.

1.3.2.3 To personalize the customer's interface in their preferred food, past ordered food, and recommended food by utilizing the personalization algorithm.

1.3.2.4 To track delivery of orders in terms of time and location using Google Map API.

1.3.2.5 To implement an online payment gateway for an ordering application, allowing users to make secure and convenient payments for their orders. The goal is to streamline the ordering process and provide clients with a seamless and speedy payment experience by integrating the payment gateway within the application.

1.3.2.6 To implement an additional layer of verification to the online ordering application by using 2FA (Two-Factor Authentication). The installation of 2FA attempts to secure user accounts from illegal access and offer a higher level of data privacy by forcing users to supply two separate authentication factors, such as a password and a unique code texted to their mobile devices.

1.4 Scope and Limitations

The study's goal is to create a Cross-Platform App called "Pabile" for the University of Mindanao's food stalls, based on a hybrid algorithm that focuses on popularity, priority, and personalization. For students, professors, and staff, the app will have user registration, order placement, order tracking, menu availability toggle, and a rating/review system. The personalization for each order is based on characteristics such as prior orders, average ratings, and positive reviews from the user, the popularity algorithm will present the most popular menu items, enhancing customer happiness and encouraging them to explore popular options.

The priority algorithm will improve delivery efficiency by considering parameters such as user location. Personalization techniques will be used to customize the user experience, considering individual user preferences and order history. The app will learn and adapt to each user's unique choices, enhancing overall satisfaction.

The development process will utilize technologies such as Dart programming language, Flutter framework, PHP, Firebase Cloud Database, Firebase Cloud Messaging, Semaphore API, and Firebase Authentication.

2. METHODOLOGY

The Agile Model is a software development technique that uses iterative development and collaboration between self-organizing cross-functional teams to generate requirements and solutions. It is an incremental paradigm in which software is produced in small increments, with the project scope, requirements, the number of iterations, and the duration specified at the beginning of the process. To deliver features for a release, tasks are separated into time intervals.

Agile approaches are popular in the software development business because they increase interaction between stakeholders, developers, and project managers. However, the quality and amount of testing tasks, generally focused on unitary testing, can pose challenges in integration and product acceptability. The continuous integration (CI) technique is used to avoid integration problems by automating the execution of test cases during version control platform upgrades. However, these tests are frequently focused on unitary testing and are built for certain platforms. This paper describes an open-source system for automating acceptance test cases to facilitate continuous integration. The framework employs Behavioral Driven Development (BDD) to develop test cases for web and mobile drivers for diverse platforms. The proposed framework has been tested and found to be promising. [11].

Figure 1. Agile Model

2.1 Data Gathering

The data gathering techniques used by the researchers in data gathering are online survey questionnaires using Google Forms. The data has been gathered by the research after the problems were found to confirm the validity or the existence of the problems. With the use of messenger to have random survey participants the researchers managed to gather 19 responses out of all the students and teachers attending the University of Mindanao. The researchers prepared 10 questions that greatly impacted validating and providing solutions to the problems.

2.2 Analysis

In this section, the researchers will provide adequate information about the necessity of the system based on the issues and requirements identified by the students, teachers, and staff attending the University of Mindanao. This will include the list of the technical tools used in the system, specifications of the user and system requirements, and a discussion of the feasibility of the proposed system.

2.2.1 User requirements

The proposed project is a Cross-Platform application that aims to help the University of Mindanao, the researchers provide ordering and delivery services to provide a much easier way to experience the food within the canteen and food court. This app can be used to find what stall is currently open, and what food is available, and it can lessen the congestion and the time consumption of the users as they will only have to wait in their respective areas. The researcher will provide a list of user requirements:

2.2.1.1 The application's user interface should be user-friendly, understandable, and easy to use.

2.2.1.2 Allows them to create an account with an OTP verification code for the security of the account.

2.2.1.3 Allows them to easily search and select different kinds of foods and drinks.

2.2.1.4 The application should provide a means of communication such as an internet connection or sim mobile services; they can communicate with the courier.

2.2.2 Software requirements

The researcher recommends the following software specification to make the application run without any problems. The software should be visual studio code and will be the main code editor to develop the cross-platform application.

2.2.3 Technical Tools

The following technical tools will be utilized in the development of the application.

2.2.3.1 Dart language

Dart is the main programming language we are going to use for mobile application development. We also use the framework for dart which is the flutter so the developers can focus on the elements that make the project unique.

2.2.3.2 Flutter Framework

Google created the open-source UI toolkit Flutter to allow developers to create natively built desktop, web, and mobile applications from a single codebase. With a comprehensive selection of pre-built widgets and a reactive architecture, it enables developers to create aesthetically pleasing and high-performance apps.

2.2.3.3 PHP (Hypertext Preprocessor)

The PHP programming language will be used for the admin panel, and because we are developing a cross-platform application, the programming language for the admin panel will be different. If the user application goes down, the admin panel will remain operational, and vice versa.

2.2.3.5 Firebase Cloud Database

Firebase Cloud Database is a database that lets you store and sync data between users in real-time. It will be used to keep consumer information and menus. The Firebase database will cost us about $30 for its monthly maintenance.

2.2.3.6 Firebase Cloud Messaging

Firebase Cloud Messaging provides reliable communication between customers and couriers. The Firebase cloud messaging service costs $3.42 monthly.

2.2.3.7 Semaphore API

Semaphore API provides SMS notification for OTP verification to verify the user's mobile number. Semaphore API costs ₱0.50 per text message.

2.2.3.8 Firebase Authentication

Firebase Authentication will be utilized to enhance user sign-in and authentication. The cost for the Firebase authentication will be $20 for monthly usage.

2.2.3.9 Map Box

The researchers will use this API to create geolocation features for the ordering application. This API will allow for the tracking and mapping of numerous places, such as the positions of delivery

drivers, consumer locations, and possibly the real-time status of deliveries. By integrating this API, the app may give an effective and accurate way to track and manage the delivery process, assuring clients receive fast and dependable service. The monthly cost for this API is $32 for about 116,000 monthly requests.

2.2.3.10. Paymongo
Paymongo is in the process of developing a payment system for businesses that is both seamless and secure. Its goal is to streamline payment processing, improve customer experience, and promote efficient transactions, resulting in increased revenue and customer happiness. The cost for each transaction is ₱40.50 per 1,500 transactions.

2.2.4 Functional and Non - Functional Requirements
2.2.4.1 User's Functional Requirements

2.2.4.1.1 User Registration
The app should allow users to create accounts with OTP verification for security reasons.

2.2.4.1.2 Food Selection
Users should be able to simply search for and select various types of foods and beverages from the provided alternatives.

2.2.4.1.3 Communication
The software should allow consumers and delivery couriers to communicate with one another.
2.2.4.1.4 Order Placement
Users should be able to place food orders through the app.

2.2.4.1.5 Order Tracking
Using the Google Maps API, the app should allow users to follow the delivery of their orders in terms of time and location.
2.2.4.1.6 Menu Availability Toggle
The app should provide a toggle button that allows users to check the availability of menu items.

2.2.4.1.7 Rating and Review System
The app should have a function that allows users to score and review the application's services, with the data saved in a database.

2.2.4.2 User's Non - Functional Requirements
2.2.4.2.1 User-Friendly Interface
To guarantee a seamless user experience, the program should have a user-friendly and simple-to-use interface.

2.2.4.2.2 Security
The app's security of user accounts and personal information should be prioritized.

2.2.4.2.3 Cross-Platform Compatibility
To ensure interoperability across different devices and operating systems, the app should be built utilizing cross-platform technologies such as Dart and Flutter.

2.2.4.2.4 Real-Time Communication
The software should allow consumers and delivery couriers to communicate in real-time.

2.2.4.2.5 Performance and Efficiency
The software should be built and tuned to provide fast and efficient performance, allowing users to place orders and track deliveries swiftly.

2.2.4.2.6 Reliability
To provide a good meal delivery experience, the app should always be reliable and accessible.

2.2.4.2.7 Scalability
The software should be designed to accommodate an increasing number of users and orders without significantly degrading performance.

2.2.5 Feasibility Study
A feasibility study was conducted using a hybrid algorithm for the ordering application to see whether establishing a Cross-Platform App for the University of Mindanao's food stalls would be practical and financially viable. The evaluation looked at a variety of factors to determine whether the enterprise was feasible. Several factors must be considered, including technical feasibility, economic feasibility, operational feasibility, and timetable feasibility.

2.2.5.1 Technical feasibility
The availability of the technical resources and tools required to construct the app was assessed to ascertain the project's technical viability. Researchers identified the required programming languages, systems, databases, APIs, and improvement conditions. The availability of various resources and tools, such as Dart, Flutter, PHP, and Firebase, secured the project's technological viability.

2.2.5.2 Economic feasibility
The economic viability of the project was evaluated to determine whether it is financially feasible. The researchers calculated the cost of development, considering the cost of software, hardware, and any other resources. They also considered how the software may generate revenue, such as commissions or transaction fees. Based on the study, the project was determined to be financially viable.

2.2.5.3 Operational feasibility
The economic viability of the project was evaluated to determine whether it is financially feasible. The researchers calculated the development cost, considering the cost of software, hardware, and other resources. They also considered how the software may generate revenue, such as commissions or transaction fees. Based on the study, the project was determined to be financially viable.

2.2.5.4 Schedule feasibility
The project's timetable feasibility was assessed by considering the time and resources available for development. When the researchers looked at the project's timeframe, they considered the

phases of development, testing, and refining. They also considered the availability of the development team as well as any potential obstacles that could affect the project's timeline. According to the analysis, the project could be completed on time.

Table 2: Project Duration

Project Phases	2023							
	No. of Days	June	July	Aug.	Sep.	Oct.	Nov.	Dec.
Plan Phase	7 days							
Design Phase	10 days							
Develop Phase	100 days							
Test Phase	7 days							
Deploy Phase	4 days							
Review Phase	30 days							

Figure 2. Project Duration

2.3 Design

2.3.1 Conceptual Framework

The conceptual framework will serve as a guide to the researchers to achieve the goals of the study and develop the application. The conceptual framework is anchored into three categories which are input, process, and output.

The popularity algorithm, which prioritizes popular goods based on user ratings, is a significant tool in ordering apps. To assess the popularity of each item, it considers data such as order count, customer reviews, ratings, and overall demand. The system helps customers discover trending options and emphasizes well-liked options by emphasizing popular goods. It is necessary to assign popularity values to each item based on user interactions and order data, such as order frequency or rating averages, throughout implementation. These scores can be used to highlight popular things or to make recommendations for popular options.

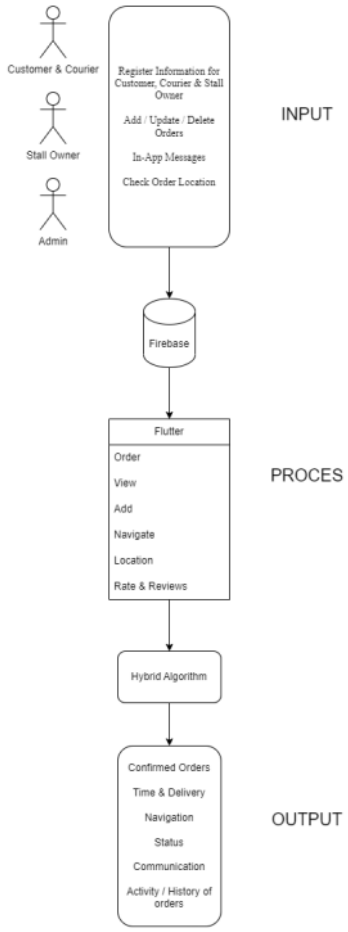


Figure 3. Conceptual Framework

An ordering application's customization algorithm can adjust recommendations to each user's interests and behavior by assessing user data such as previous orders, favorite cuisines, and food categories. This algorithm aids in improving the user experience and ensuring that users get the greatest possible experience.

In an ordering application, the prioritization algorithm determines the order or priority of objects or orders shown to users. It takes into account aspects such as delivery time, closeness to the user's location, and order urgency. Implementation entails considering the projected delivery time, the distance between the user and food stalls at the University of Mindanao, and the urgency of the request. The application organizes and shows orders depending on these factors, with time-sensitive orders being prioritized. It can also consider user preferences, such as preferred delivery times or remembered buildings/classrooms, to improve the order priority process.

INPUT
This system has four types of users: Customer, Courier, Shop Seller, and Admin. For the customer to use the application they will have to register to the application. For the Courier, they will have to register needed information such as name, class schedule, mobile number, and student id to access the application as a courier. Additionally, shop sellers will also register their stall number, names, and address for them to use the application.

PROCESS
The process starts when the customer orders food from the system. Customers can view, add, and update their ordered food. Customers can also rate or review the ratings of each stall to see the top stalls or foods other customers rated. Each ordered food can be located via real-time navigation of the application. The administrator can determine the high rating stalls also the administrator will be checking the application form that the courier provided if it is applicable or not.

Firebase, Flutter, Dart, Semaphore, Mailer, and MapBox are the technologies that this system is using. Dart programming language is the main programming language that this system is using. It enables this system to be compatible with other platforms such as Android, iOS. Semaphore will be the one who sends a SMS notification to the customer about their ordered food. And lastly, Mailer will be the one who sends an email notification to each user about their registered form for verification purposes.

OUTPUT
The output of the system will have the following: Confirmed Orders, Time and Delivery, Navigation Status, Communication, Activity, or History of Food Ordered.

2.3.2 Data Models
This section is where you can see the data models, user interaction, and data flow that the researchers have provided. All will be illustrated on how data flows throughout the application/system and be stored in the database. It also includes the complete system's design, architecture, and functionality.

2.3.2.1 Use-Case Diagram
A use-case diagram is a graphic depiction that shows how users or other actors interact with a system under examination. It gives examples of the numerous use cases or features the system offers. Use cases are represented by ovals, and actors are depicted as stick figures. As you can see in the figure, it shows the relationship between admin, courier, customer, and staff.

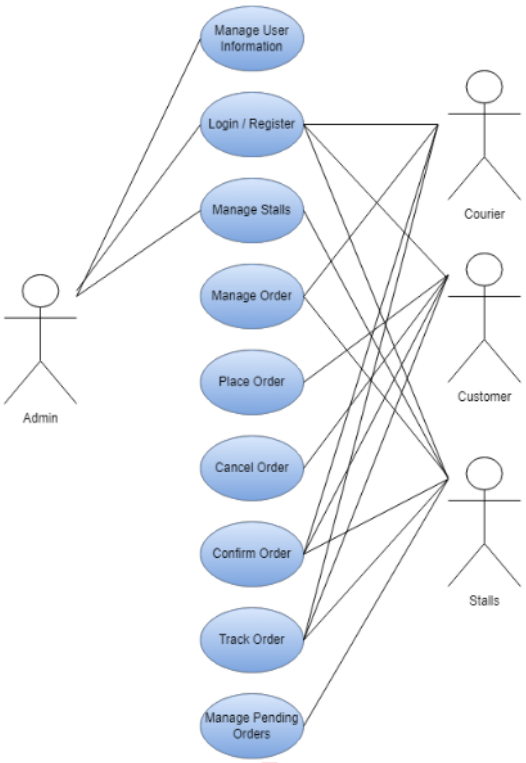


Figure 4. Use-case diagram

This figure also shows the roles of the four users. The users as the customer, admin, delivery courier, and staff or stall operator. The admin manages the user information, and registration of accounts for the users, and helps on managing the stalls inside the application. The users as the customer uses the application to log in to the application and can place, cancel, confirm, and track their orders. The user as delivery courier uses the application to register or log in and can manage, and confirm the order they received, and track the customer to where food shall be delivered. The user as the staff or stall operator uses the application to log in, manage the stall, orders, confirm, and handle upcoming orders, the staff owner can also track the courier and the customer on where the courier is and where the customer is located.

2.3.2.2 Entity Relationship Diagram
The entity relationship diagram (ERD) shows the relationship between the entities in the database. The figure depicts the system's database structure.

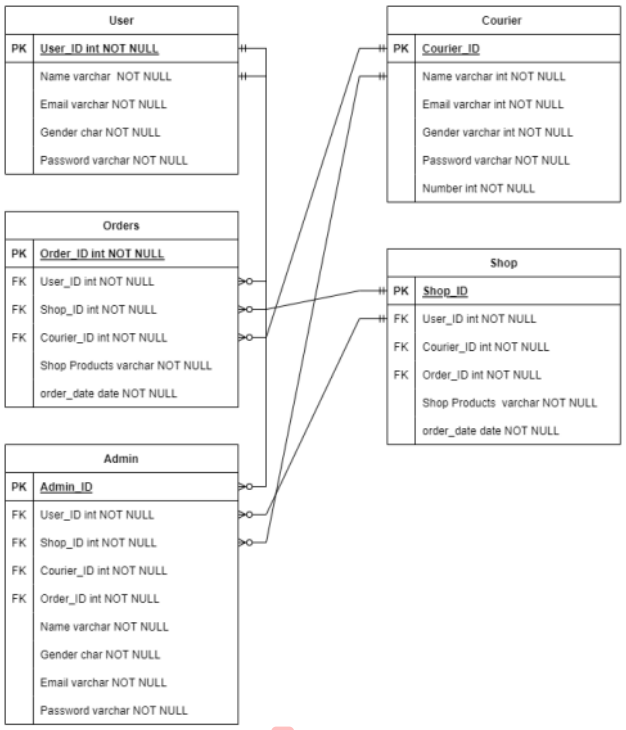


Figure 5. Entity relationship diagram

The entity relationship diagram (ERD) provided depicts the relationships between four main entities: User, Courier, Shop, and Admin. The user entity represents the system's users, with attributes including user_id, name, email gender, and password. The courier entity represents the delivery personnel, identified by Courier_ID, with attributes such as name, email, gender, password, and number. The shop entity relates to specific orders, with attributes such as Shop_ID, user_id, courier_id, shop products, and order_date. Lastly, the Admin entity, associated with users, shops, couriers, and orders, contains attributes like Admin_id, user_id, shop_id, courier_id, name, gender, and password.

2.3.3 Data Dictionary

The following is the data dictionary of the system.

Table 2: Admin

Fields	Key	Type	Null	Description
Admin_ID	PK	int	no	identification of admin
User_ID	FK	int	no	identification of user
Courier_ID	FK	int	no	identification of courier
Shop_ID	FK	int	no	identification of shop
Name		varchar	no	name of admin
Email		varchar	no	email of admin
Gender		char	no	gender of admin
Password		varchar	no	password of admin

The admin table is an important part of the database architecture because it represents administrators who oversee controlling and overseeing the application or system. The Admin_ID field, User_ID field, Courier_ID field, Shop_ID field, Name field, Email field, Gender field, Email field, Gender field, and Password field are all included to record vital information about admins. These fields provide easy reference, relationship establishment, and administration-related data management within the program. The Admin_ID field is the primary key, guaranteeing that each admin has a unique identification, while the User-ID field connects the admin to their user account, the Couried_ID field connects the admin to their courier account and the Shop_ID field connects the admin to the applicable store account. The name field contains the admin's name, the Email field contains unique identifiers, the Gender field contains the admin's gender, and the Password field stores the encrypted form of the admin's password. These fields enable smooth coordination and efficient functioning of the application.

Table 3: User

Fields	Key	Type	Null	Description
User_ID	PK	int	no	identification of user
Order_ID	FK	int	no	identification of order
Name		varchar	no	name of user
Email		varchar	no	email of user
Gender		char	no	gender of user
Password		varchar	no	password of user

The User table is an important aspect of a database schema because it represents users in an application or system. It has fields for storing user-specific data, such as User_ID, Order_ID, Name, Email, Gender, and Password. These parameters enable effective user administration and storage, guaranteeing that each user in the system is uniquely identified.

Table 4: Courier

Fields	Key	Type	Null	Description
Courier_ID	PK	int	no	identification of courier
Order_ID	FK	int	no	identification of order
Class Schedule		varchar	no	class schedule of the courier
Name		varchar	no	name of courier
Email		varchar	no	email of courier
Gender		char	no	gender of courier
Password		varchar	no	password of courier
Number		int	no	number of couriers

The Courier table, which stores information on the couriers associated with the application or system, is an important component of the database structure. It offers areas for collecting pertinent information about the couriers. The primary key is the Courier_ID field, which uniquely identifies each courier. The Order_ID field, which serves as a foreign key, connects the Courier and Order tables, allowing couriers to be associated with specific orders. Class Schedule, Name, Email, Gender, Password, and Number are additional sections that hold information such as the courier's class schedule, name, email address, gender, password, and contact number. This table enables successful coordination and fulfillment of orders within the application or system by facilitating efficient management and tracking of courier-related information.

Table 5: Shop

Fields	Key	Type	Null	Description
User_ID	FK	int	no	identification of user
Courier_ID	FK	int	no	identification of courier
Shop_ID	PK	int	no	identification of shop
Order_ID	FK	int	no	identification of order
Name		varchar	no	name of shop
Email		varchar	no	email of the shop user
Shop products		varchar	no	products that the shop sells
Gender		char	no	gender of shop user
Password		varchar	no	the password of the shop user

The Shop table represents and keeps the information about the shops within the application or system. It offers areas for capturing shop-specific information. The primary key is the Shop_ID column, which uniquely identifies each shop. The foreign keys User_ID, Courier_ID, and Order_ID construct links between the Shop table and the User, Courier, and Order tables, allowing associations with users, couriers, and orders. Name, Email, Shop items, Gender, and Password are additional fields that save information about the shop's name, email address, products sold, gender of the shop user, and password. This table makes it easier to organize and handle shop-related data, allowing for smooth operations and collaboration inside the application or system.

Table 6: Order

Fields	Key	Type	Null	Description
User_ID	FK	int	no	identification of user
Courier_ID	FK	int	no	identification of courier
Shop_ID	FK	int	no	identification of shop
Order_ID	PK	int	no	identification of order
Shop products		varchar	no	products that the shop sells
Order Date		timestamp	no	ordered date

The Order table is an important part of the database schema since it stores information about orders placed within the application or system. It offers areas for capturing essential order details. The primary key is the Order_ID column, which uniquely identifies each order. As foreign keys, the User_ID, Courier_ID, and Shop_ID entries build associations between the Order table and the User, Courier, and Shop tables, respectively, tying orders to the relevant user, courier, and shop. Shop goods and Order Dates are additional fields that record information about the products ordered

as well as the date and time the order was placed. This table enables effective order processing and fulfillment within the application or system by facilitating efficient order tracking and administration.

2.3.4 Architecture Design

The following diagram illustrates the design of the application as the process of preparing for the development of an application system. It deals well with the fundamental aspect of interaction that exists between design and requirement engineering. This is completed by specifying a collection of hardware and software components, along with their respective interfaces.

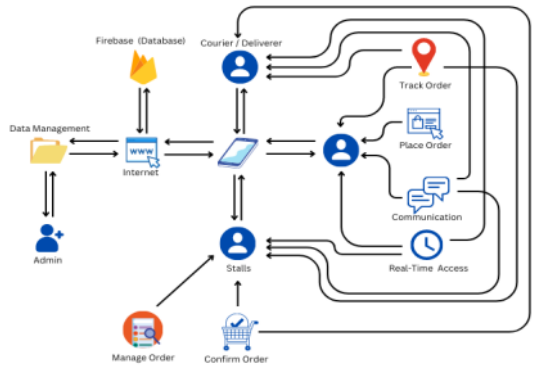


Figure 6. Architecture Design

In the figure above, the admin can only access the data management. Then after the collection of data, it will be passed through the internet then will be processed through the Firebase database. The internet then sends the data through our system then will be passed to the end-user page. Wherein they can see the data for each stall, foods, reviews, and more. Users like couriers and stall owners have functions such as tracking orders, communicating between their customers and couriers, and checking their orders. Stall owners can manage the orders and confirm their orders. The customer can order, track their order, communicate with the stall owner and courier, and have real-time access.

2.3.5 Sample Prototypes

The Screenshots shown in the following figures are sample prototypes of the proposed application.

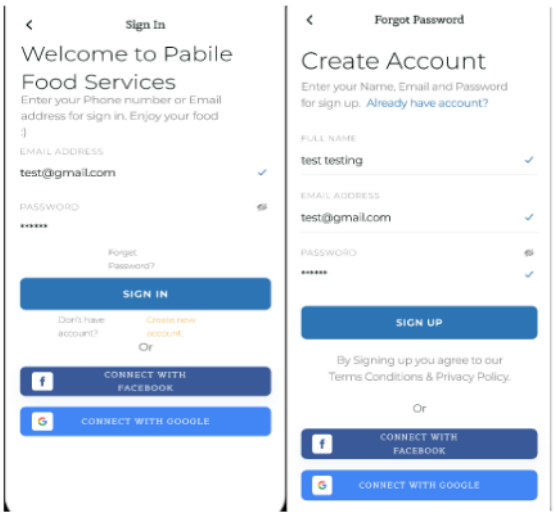


Figure 8. Login Page

The figure shows the Login and Sign-Up page of our application where the user will log in their profile and create their profile.

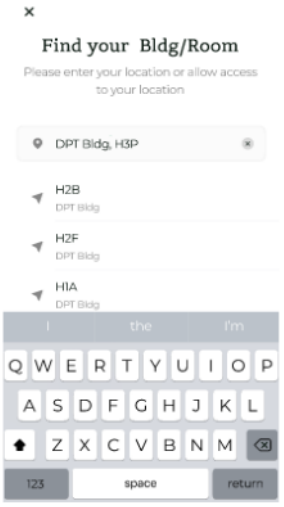


Figure 9. Location Picker Page

On this page, the user or customer will have to select or choose the location where they want the order to be delivered.

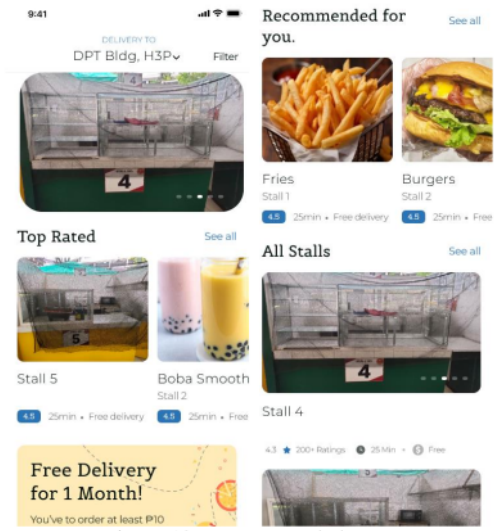


Figure 10. Recommended Page

In this figure, it shows the top-rated, top-recommended, and all stalls available to order food from.

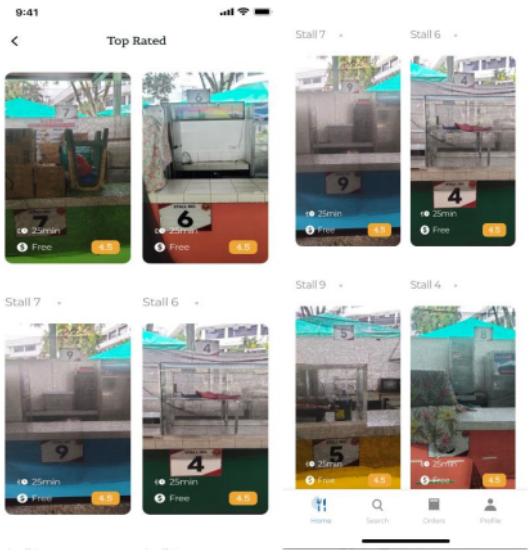


Figure 10.1. Specifics Page

If the customer has selected see all from a specific rating. The app displays all the best rating foods or stalls.

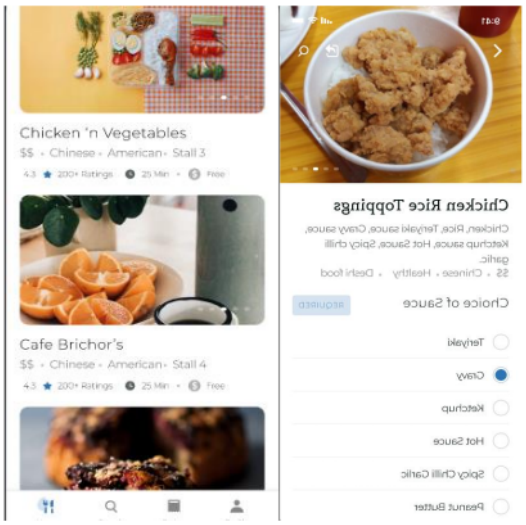
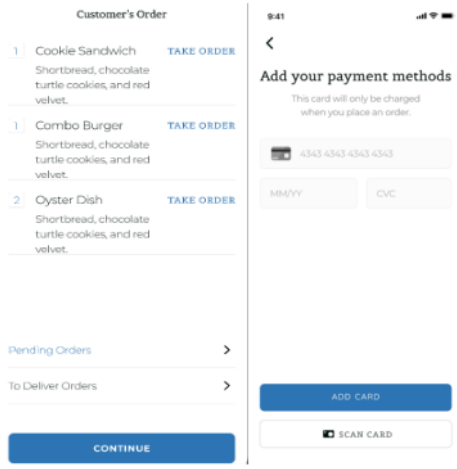


Figure 11. Selecting Order

On this page, the customer can manually select the desired specifications for food.



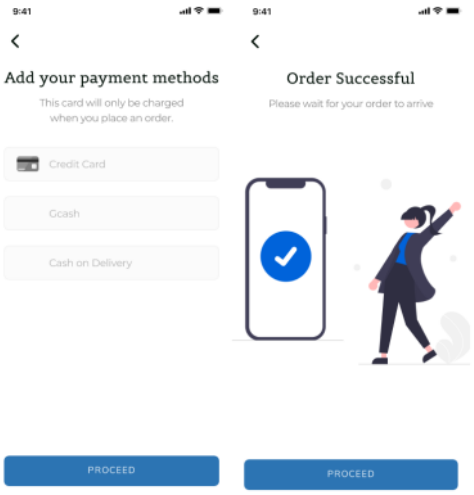
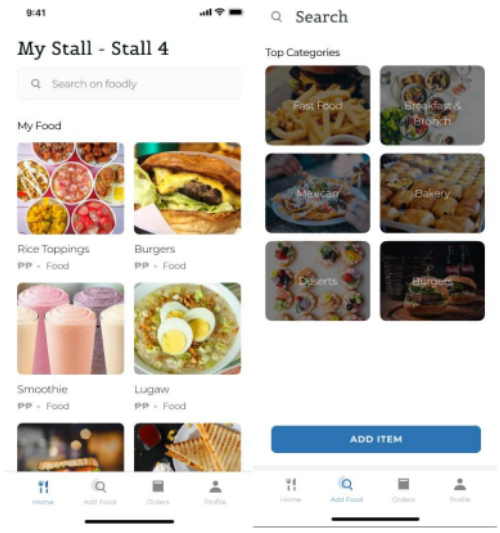


Figure 12. Continue Order

On this page, the application will redirect the user to the customer order page, wherein the customer decides to continue the ordered food, add, delete, or update their ordered food. And after they continue, they will be redirected to the payment method page.



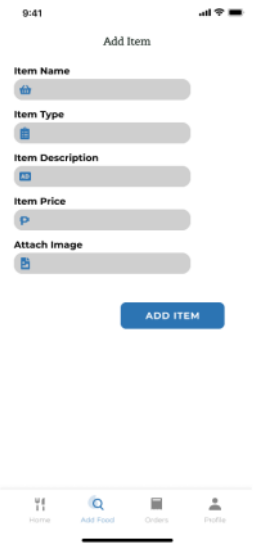


Figure 13. Stall Dashboard

In this dashboard stall owners can see the foods they want to sell and what foods they want to add to their shop.

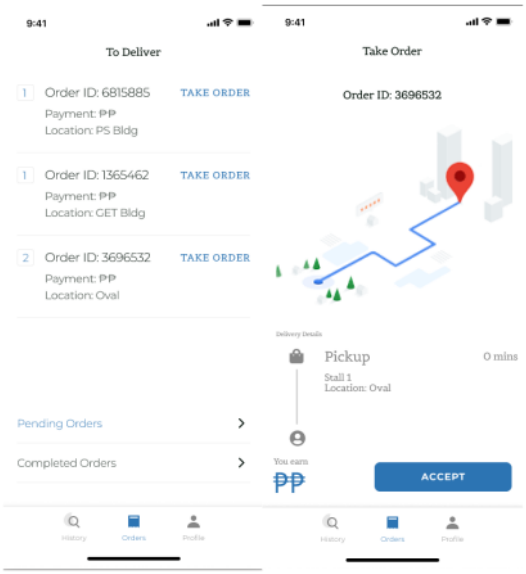


Figure 14. Courier Dashboard

In this figure, the dashboard for couriers displays all the information for each customer's order. And they can choose whether they want to take that order or not.

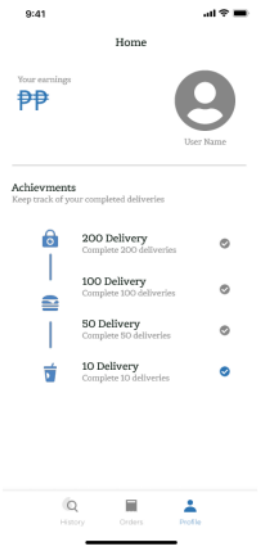


Figure 15. Profile Page

In this figure, the profile page for the courier is displayed. The page shows the courier's history of delivery and its total earnings.

2.4 Development

It is based on the agile scrum methodology. The current state of a project. Projects can be divided into smaller tasks, sprints, or cadences, with the team meeting at the end of each sprint to evaluate project development. And will take the necessary steps once the work backlogs have been defined by the project team. Determines what the team will undertake to finish the project on schedule and to a high standard. The scrum leader directs the actions of the team and the steps that scrum members take to achieve a specific goal. The development team oversees application development.

2.4.1 Software Development Modal Activities

Sprint 1: Application development for all platforms. The development team is assigned to develop an application to test the speed in terms of real-time messages, real-time GPS and real-time ordering food and receiving orders from the customers to stall, the type of device utilized, which influences whether the connection is faster or slower, and the approximate location of the device from the cell site or the router. This task will make use of the cross-platform, which is Flutter, used to create the online ordering app.

Sprint 2: Developing a web. The web interface will then be created by the development team. It uses the web 2.0 platform and the PHP language to collect all data from mobile devices utilizing Java scripts and a database to hold the mobile's condensed data devices. The scrum master makes the development process easier. The product's creator will then check the system's operation and security features.

Sprint 3: Full system functionality. All the functionality is linked by the development team. Sprint 1, and Sprint 2, and ensures the system/s full performance.

3. REFERENCES

[1] L. T. Chai and D. N. C. Yat, "Online food delivery services: Making food delivery the new normal," Journal of Marketing Advances and Practices, vol. 1, no. 1, pp. 62-77, 2019.

[2] M. Chen, M. Hu, and J. Wang, "Food delivery service and restaurant: Friend or foe?" Management Science, vol. 68, no. 9, pp. 6539-6551, 2022.

[3] R. Ramesh, S. V. Prabhu, B. Sasikumar, B. K. Devi, P. Prasath, and S. P. R. Kamala, "An empirical study of online food delivery services from an applications perspective," Materials Today: Proceedings, vol. 80, pp. 1751-1755, 2023.

[4] S. Kumar and A. Shah, "Revisiting food ordering application during COVID-19 pandemic? Investigating the role of emotions," Journal of Retailing and Consumer Services, vol. 62, p. 102595, 2021.

[5] M. Al Amin, M. S. Arefin, N. Sultana, M. R. Islam, I. Jahan, and A. Akhtar, "Evaluating the customers' dining attitudes, e-satisfaction and continuance intention toward mobile food ordering apps (MFOAs): evidence from Bangladesh," European Journal of Management and Business Economics, vol. 30, no. 2, pp. 211-229, 2020.

[6] A. Juneja, R. Kumar, and M. Gupta, "Digitization of Traditional Markets using DART based Cross Platform Development," in Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Dec. 2022, pp. 1-6.

[7] A. El Tom, C. Bogdan, T. A. Majchrzak, and T.-M. Grønli, "Criteria Based Evaluation of Cross-Platform Development Frameworks," 2023. [Online]. Available: https://scholarspace.manoa.hawaii.edu/items/0a559934-1b4f-4c5b-a7a8-3c177e2eb35f. Accessed: Jul. 06, 2023.

[8] L. Wang and B. Yi, "Research on O2O takeaway restaurant recommendation system: Taking ele.me app as an example - cluster computing," Cluster Computing, vol. 22, no. 4, pp. 899-907, Dec. 2019.

[9] Stephens, H. Miller, and L. Militello, "Food Delivery Apps and the Negative Health Impacts for Americans," Frontiers, Feb. 04, 2020.

[10] J. Li, M. A. Bonn, J. Wang, and M. Cho, "Food delivery application user segmentation in the mobile marketing world in China" Journal of the Asia Pacific Economy, vol 28, no 2, pp 484-501, 2023

[11] I. Yanjari, B. Marín, and G. Giachetti, "An open-source framework for cross-platform testing in agile projects," in 2022 41st International Conference of the Chilean Computer Science Society (SCCC), November 2022, pp. 1-8.

[12] M. Lije, "Grab launches GrabFood with four-day dining event in BGC," in Top Gear Philippines, Nov. 7, 2018. [Online]. Available: https://www.topgear.com.ph/news/industry-news/grabfood-launch-event-bgc-a2671-20181107. [13] A. T. Juego, "Foodpanda delivers the goods," in Inquirer Business, Feb. 19, 2017. [Online]. Available: https://business.inquirer.net/224836/foodpanda-delivers-goods.

[13] A. T. Juego, "Foodpanda delivers the goods," in Inquirer Business, Feb. 19, 2017. [Online]. Available: https://business.inquirer.net/224836/foodpanda-delivers-goods.

[14] A. Ray, A. Dhir, P. K. Bala, and P. Kaur, "Why do people use food delivery apps (FDA)? A uses and gratification theory perspective," Journal of Retailing and Consumer Services, vol. 51, pp. 221–230, Nov. 2019.

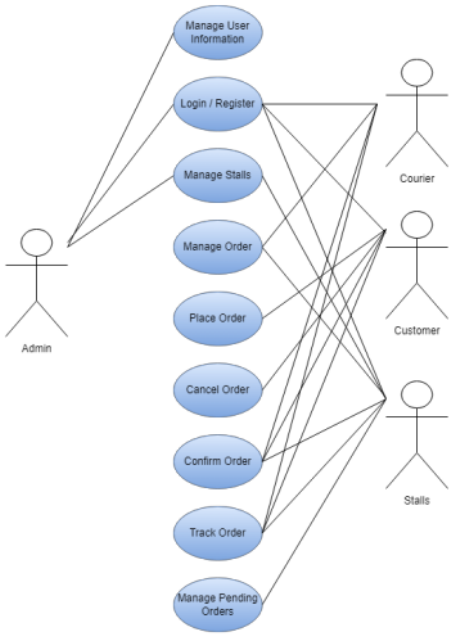
[15] X.W.D Automation et al., "Solving online food delivery problem via an effective hybrid algorithm with intelligent batching strategy," in Proceedings of the International Conference on Intelligent Computing Theories and Application (ICIC), 2021, pp. XX-XX, doi: 10.1007/978-3-030-84529-2_29.

[16] A. A. Alalwan, "Mobile food ordering app: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse," International Journal of Information Management, vol. 50, pp. 28-44, 2020

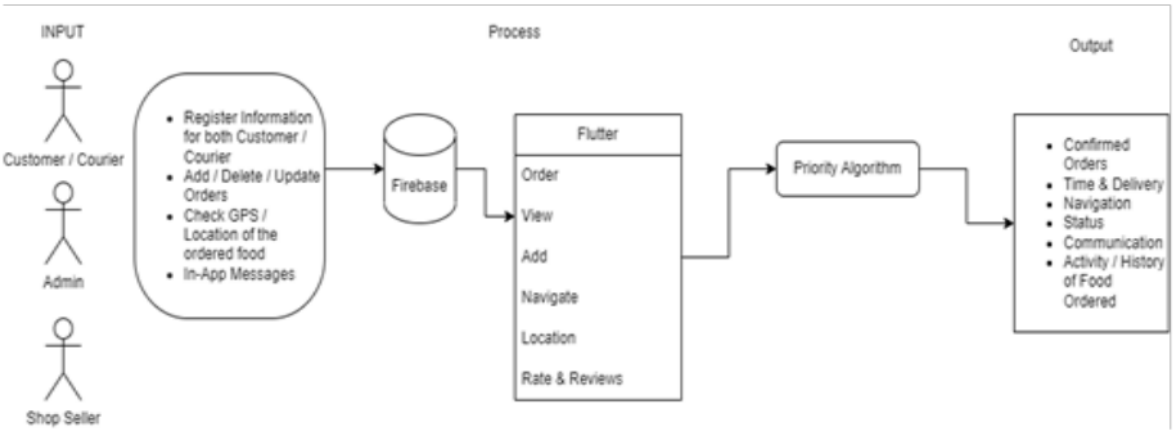
Table 1: RRS

Features	Grab	Food Panda	Maxime	McDelivery PH	Pabilee
Communication	✓	✓	✓	✓	✓
Tracking	✓	✓	✓	✓	✓
Rating and Feedback Recommendation		✓		✓	✓
Two Factor Authentication		✓			✓
Content-Based Filtering				✓	✓
Online payment using g-cash	✓	✓		✓	✓

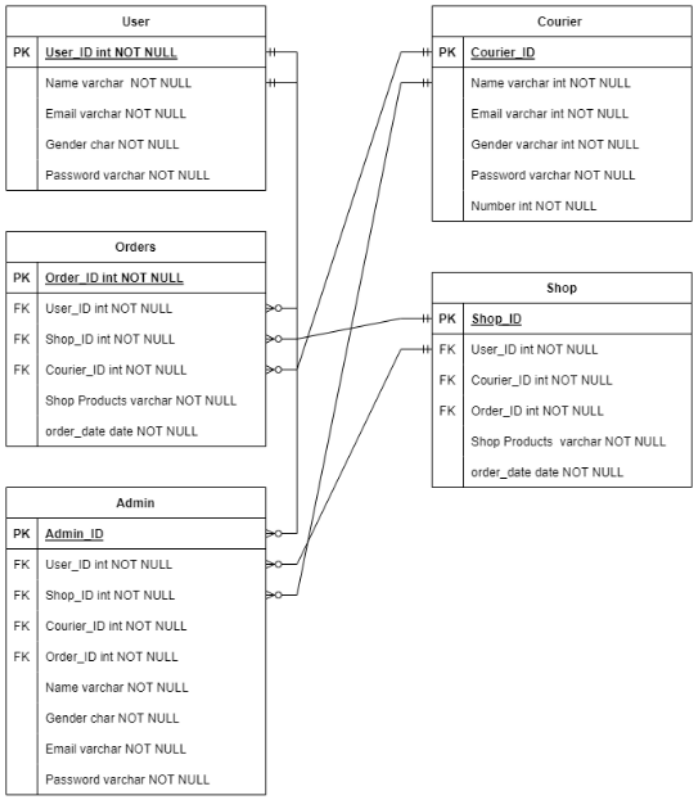
User case diagram



Conceptual Framework



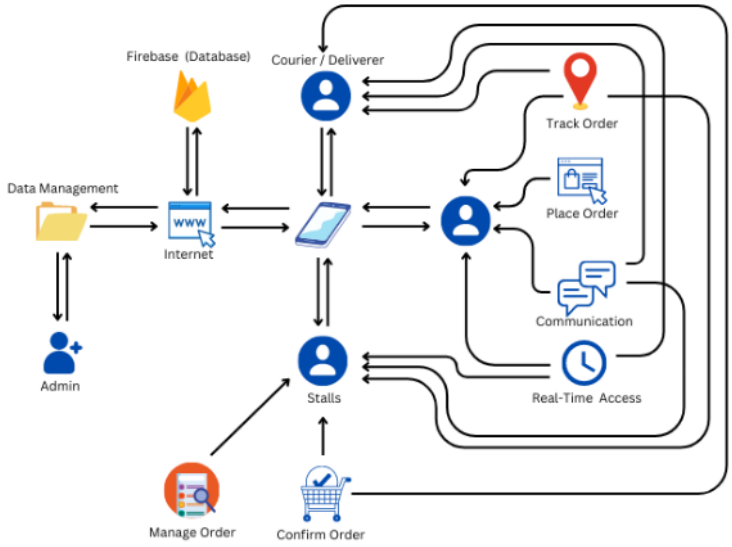
Entity Relationship Diagram



Project Duration

Project Phases	2023							
	No. of Days	June	July	Aug.	Sep.	Oct.	Nov.	Dec.
Plan Phase	7 days							
Design Phase	10 days							
Develop Phase	100 days							
Test Phase	7 days							
Deploy Phase	4 days							
Review Phase	30 days							

Architecture Design



Pabile: A Cross-Platform App for the University of Mindanao Food Stalls using a Hybrid algorithm for Ordering Application

ORIGINALITY REPORT

17%	4%	5%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Mindanao Student Paper	11%
2	www.researchgate.net Internet Source	1%
3	researchers.unab.cl Internet Source	1%
4	Ali Abdallah Alalwan. "Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse", International Journal of Information Management, 2020 Publication	1%
5	Sushant Kumar, Arunima Shah. "Revisiting food delivery apps during COVID-19 pandemic? Investigating the role of emotions", Journal of Retailing and Consumer Services, 2021 Publication	<1%

6	Submitted to Swinburne University of Technology Student Paper	<1%
7	Submitted to University of Southern California Student Paper	<1%
8	smartech.gatech.edu Internet Source	<1%
9	scholars.cityu.edu.hk Internet Source	<1%
10	Linan Wang, Bo Yi. "Research on O2O take-away restaurant recommendation system: taking ele.me APP as an example", Cluster Computing, 2018 Publication	<1%
11	Rex Perez Bringula, Janszen Kiel L. Jose, Arnelle T. Lardizabal, John Raymon D. Lizaso. "Predictors of Usability of a Mobile Intelligent Agent Information Provider for College Students", International Journal of Mobile Human Computer Interaction, 2023 Publication	<1%
12	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1%
13	Ignacio Yanjari, Beatriz Marin, Giovanni Giachetti. "An open-source framework for	<1%

cross-platform testing in agile projects", 2022 41st International Conference of the Chilean Computer Science Society (SCCC), 2022
Publication

14	Submitted to University of Western Sydney Student Paper	<1 %
15	Shiang-Lih Chen McCain, Jeffrey Lolli, Emma Liu, Li-Chun Lin. "An analysis of a third-party food delivery app during the COVID-19 pandemic", British Food Journal, 2021 Publication	<1 %
16	Submitted to University of Portsmouth Student Paper	<1 %
17	dl.lib.mrt.ac.lk Internet Source	<1 %
18	ideas.repec.org Internet Source	<1 %
19	ebin.pub Internet Source	<1 %
20	Jun (Justin) Li, Mark A. Bonn, Juan Wang, Meehee Cho. "Food delivery application user segmentation in the mobile marketing world in China", Journal of the Asia Pacific Economy, 2021 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Pabile: A Cross-Platform App for the University of Mindanao Food Stalls using a Hybrid algorithm for Ordering Application

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17