BELLY SHACK ORDER & DELIVERY MANAGEMENT SYSTEM (FOOD DELIVERY ORDER)

Version 1.0

Date: April 27, 2025

Revision history: April 27, 2025

Authors:
Grace J. Diana, Project Manager
(2201101815@student.buksu.edu.ph)

Ethil Jane D. Meliton, Quality Assurance and Documentation Specialist
(2101101252@student.buksu.edu.ph)

Chary Niel Q. Paderog, Lead Developer
(2201102223@student.buksu.edu.ph)

Table of Contents

1. Document Overview

1.1 Scope

The "Belly Shack Order & Delivery Management System" project focuses on making order and delivery easier and faster for businesses. This document explains how the app works, including the tools and technology we used to build it, such as Android Studio, Java, and Firebase. It also describes the design of the app and how we tested its features to make sure everything works properly. The system supports three types of users: the chef, the rider, and the customer, and each has a specific role and flow inside the app. Customers can place orders, riders can track deliveries, and chefs can manage food preparation easily. We also included the system requirements needed, such as a stable internet connection and a device that can run the app smoothly. Some major features explained here are email notifications, in-app messaging, reCAPTCHA for user verification, and generating reports based on user feedback. However, the project does not yet include real GPS tracking devices or online payment systems connected to third-party services. These features are planned for future updates so we can improve the system even more. Overall, this document gives a complete overview of what the current version of the app can do and what improvements are coming next.

1.2 Audience

The *Belly Shack Order & Delivery Management System* involves several key groups who play important roles in its development and success. Each group has a unique purpose in making sure the app works well, looks good, and meets business needs. From writing the code and designing the interface to testing features and planning future updates, teamwork is essential. Understanding the responsibilities of developers, designers, testers, stakeholders, and maintenance teams helps ensure smooth development and long-term support. This section explains how each team contributes to the app's functionality and ongoing improvement.

Developers: Both front-end and back-end developers will refer to this document to understand the app's structure, coding practices, and integration points using Android Studio, Java, and Firebase. It provides the technical context needed to implement or expand the current features and support planned upgrades like GPS tracking and online payment systems.

UI/UX Designers: Designers will use this document to ensure that the interface and user flows are aligned with the roles of chefs, riders, and customers. It helps them build an intuitive, accessible, and responsive experience tailored to each user type's needs.

Quality Assurance (QA) Testers: A tester will rely on this document to verify that all implemented features, such as messaging and report generation, function correctly. It outlines testable components and helps identify areas that need debugging before the app goes live or receives updates.

Project Stakeholders: Business owners, product managers, and other decision-makers will use this document to assess the system's current capabilities, user support features, and future development plans. It offers insights into how the system meets business goals like improving delivery efficiency and customer satisfaction.

Future Maintenance Teams: This group will be responsible for keeping the system updated and functional after the initial release. They will rely on this document for insights into the app's architecture, known issues, and plans for future enhancements.

2. Project Overview

2.1 Executive Summary

The Belly Shack Order & Delivery Management System is a mobile app designed to make the order and delivery process faster and easier for food businesses. It helps both customers and staff have a smooth and convenient experience. The app will make sure that orders are correct and are delivered on time. It also makes it easy for customers to follow their orders from the kitchen to their door. The system is created to reduce mistakes and improve communication between the business, the chef, and the delivery rider. Customers will not have to wait without knowing where their order is. It also helps the business manage multiple orders in a simple and organized way. The main goal is to help businesses work better while keeping customers happy.

Objectives:

This system aims to improve how food orders are taken, prepared, and delivered. One goal is to make sure each order is accurate and reaches the right customer. Another goal is to let customers and riders track the order easily in real time. It also helps businesses know how their staff are doing with performance reports. The app wants to make the whole ordering process faster and more reliable. By using the app, businesses can also schedule deliveries better to avoid delays. The system also plans to be easy to use so even new users won't have a hard time learning it. In the future, the app may include GPS tracking and online payments to make it even better.

High-Level Features:

The Belly Shack Order & Delivery Management System has several main features that streamline the order and delivery process. Customers can easily place orders through a user-friendly interface and track the status of their orders in real-time. Chefs can manage food preparation and view incoming orders in a dedicated section of the app. Riders can efficiently track deliveries and receive route updates for smooth delivery management. The system also includes a feedback report feature, allowing users to provide valuable insights that help improve the service. These main features work together to ensure a seamless experience for customers, chefs, and riders.

2.2 Problem Statement

The Belly Shack Order & Delivery Management System aims to address several challenges faced by businesses in the food delivery industry. One of the key issues the app intends to solve is the lack of real-time tracking for both customers and riders, which often leads to delayed deliveries, miscommunications, and poor customer experiences. Additionally, managing orders manually can result in inefficiencies, errors, and delays in food preparation. The app will streamline the process by allowing chefs to easily manage orders, riders to track deliveries in real time, and customers to monitor the status of their orders. Another major problem the system seeks to address is the difficulty in tracking performance and generating accurate reports, which can hinder businesses from optimizing their operations. The app will provide a simple way to schedule deliveries, track progress, and generate reports for daily or monthly performance analysis.

User Needs:

The Belly Shack Order & Delivery Management System addresses several user needs to improve the order and delivery process. Customers need real-time tracking to monitor the progress of their orders and receive updates to avoid delays and miscommunication. Riders require an efficient platform for tracking deliveries and navigating routes to ensure timely and accurate deliveries. Chefs need an easy-to-use system to manage orders, reducing errors and delays in food preparation. Clear communication between customers, riders, and chefs is essential to ensure smooth interactions and reduce misunderstandings. Additionally, business owners need a simple way to generate performance reports and analyze delivery operations to improve efficiency and customer satisfaction.

Market Analysis Summary:

The food delivery industry has seen significant growth, especially with the rise of online ordering and delivery apps. Customers are increasingly seeking faster, more reliable, and transparent services. However, many businesses still face challenges like delayed deliveries, miscommunication, and inefficient order management. Current solutions often lack real-time tracking and fail to provide adequate tools for managing deliveries, leading to poor customer satisfaction. The *Belly Shack Order & Delivery Management System* addresses these gaps by offering real-time tracking for both customers and riders, improving communication, and providing chefs with efficient order management tools. Additionally, the system's ability to generate performance reports gives businesses valuable insights into their operations, enabling them to optimize processes and improve overall efficiency. As demand for reliable and efficient food delivery services continues to grow, this app presents an opportunity to meet the needs of businesses looking to enhance their delivery operations and customer experience.

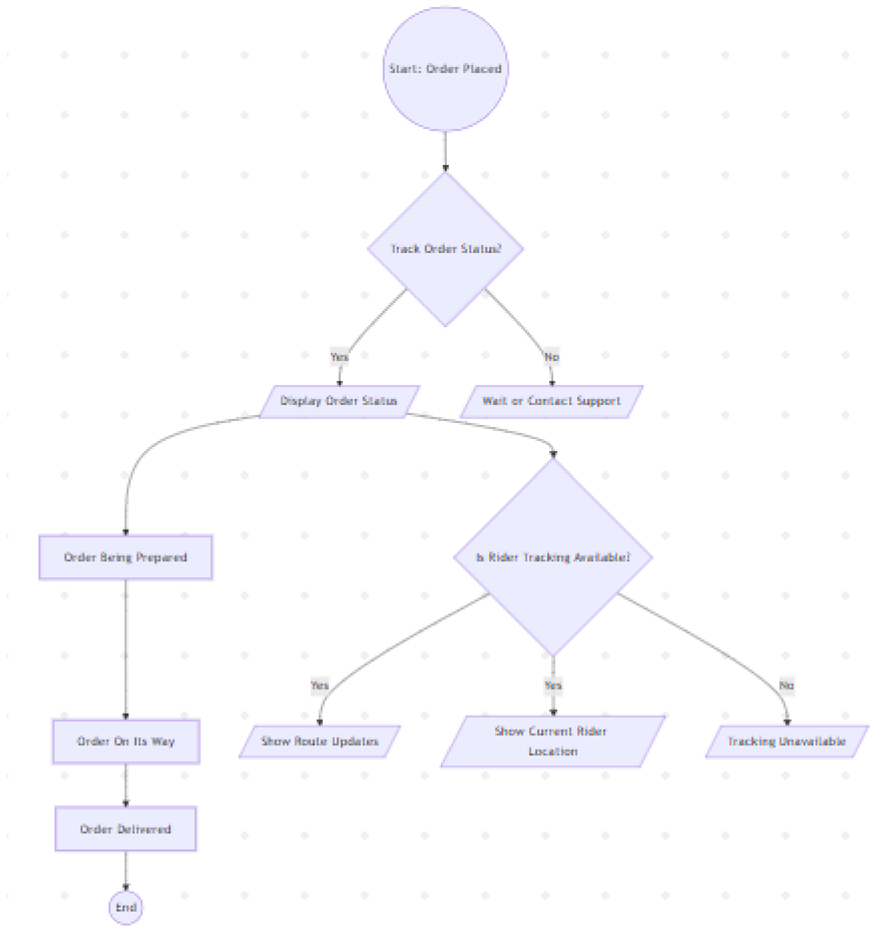
3. **Functional Specifications**

3.1 Feature List

Features 1: Real-Time Order Tracking

The real-time order tracking feature allows customers to track their orders from start to finish. As soon as an order is placed, customers can see the current status, such as whether the order is being prepared, on its way, or has been delivered. This transparency reduces customer anxiety about when their food will arrive. Riders also have access to real-time tracking, which helps them stay on course and avoid delays. The system shows the rider's current location and the most efficient route to take. It also provides updates if there are any issues, like traffic or detours. This keeps everyone informed and improves communication. With real-time tracking, both customers and riders have better control over the delivery process.

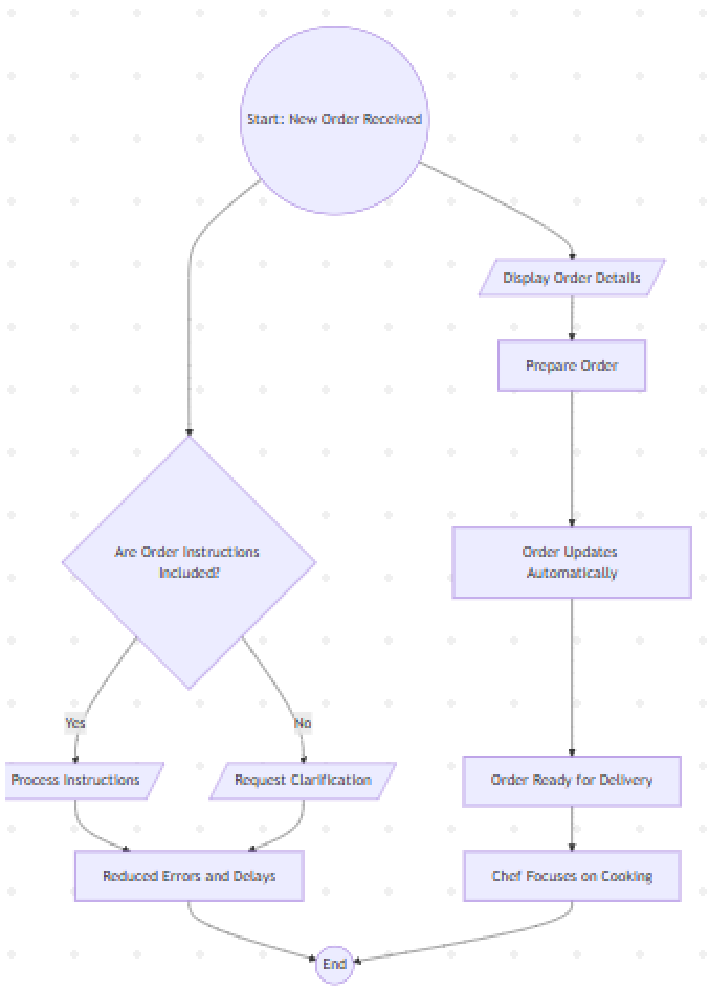
User Interaction Flow



Feature 2: Order Management for Chefs

Chefs need a smooth way to manage orders in the kitchen. With the order management system, incoming orders are displayed clearly, so chefs can prepare food in the correct order. It helps eliminate confusion and ensures that all orders are completed on time. Each order includes details like the dishes to be prepared and any special instructions. The system automatically updates the status of each order, so chefs know when to start and finish preparing. By managing orders digitally, the chance of errors is reduced, and preparation is more organized. The app helps chefs stay focused on cooking, while the system handles the order flow. This leads to quicker and more efficient food preparation.

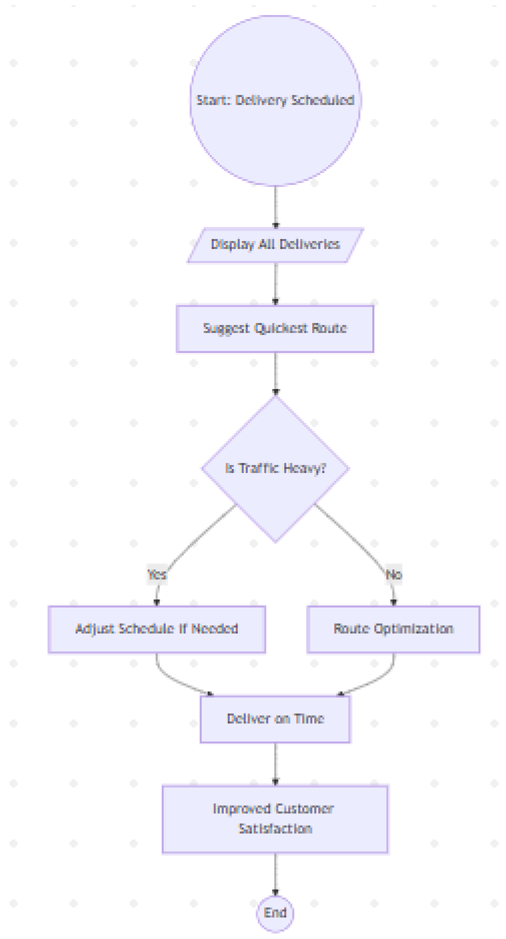
User Interaction Flow



Feature 3: Delivery Scheduling & Route Optimization

The delivery scheduling feature allows riders to manage their deliveries efficiently. Riders can see all their deliveries for the day and plan the best route to take. The system suggests the quickest route based on traffic and distance, helping riders save time. Riders can also adjust the schedule if needed, such as when they are delayed or need to handle a last-minute change. The route optimization feature helps riders avoid heavy traffic and choose the most efficient roads. This leads to faster deliveries, improving customer satisfaction. The system ensures that no delivery is forgotten, and every order is delivered on time. Delivery scheduling and optimization make the entire process smoother for riders and customers.

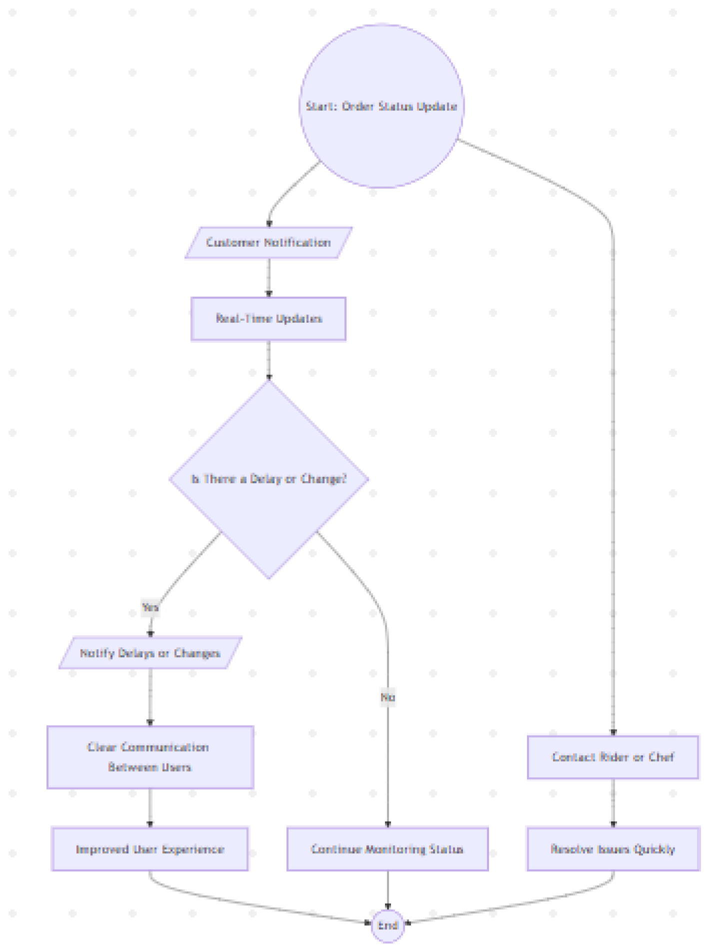
User Interaction Flow



Feature 4: Clear Communication

Clear communication is crucial to the success of the system. Customers, riders, and chefs all need to be informed about the order status and any changes. The system sends notifications to keep everyone updated in real-time. If there are any delays or issues, such as a late order or route change, users are notified immediately. This prevents confusion and ensures that everyone is on the same page. Customers can also contact the rider or chef if they have questions about the order. By making communication easy and efficient, the system reduces misunderstandings and improves the overall experience. It ensures that customers feel informed and riders and chefs can address any issues quickly.

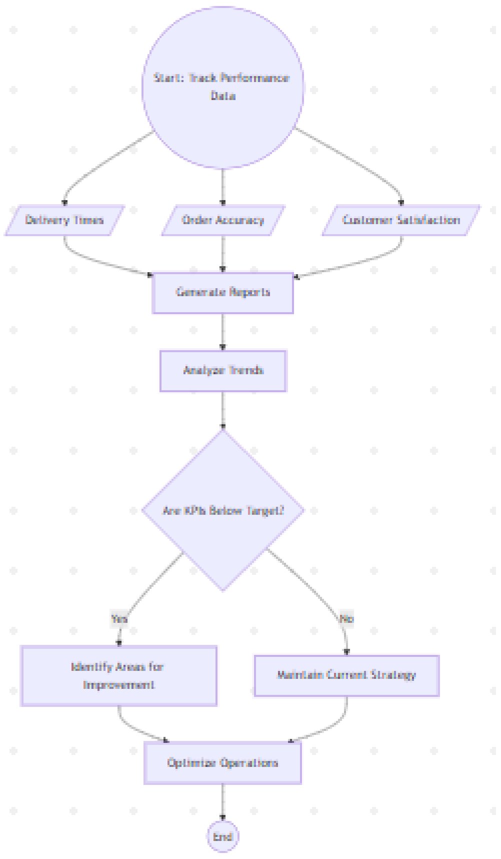
User Interaction Flow



Feature 5: Performance Tracking & Reports

Performance tracking and reports give businesses a way to measure and improve their delivery operations. The system tracks important data, such as delivery times, order accuracy, and customer satisfaction. At the end of each day or month, business owners can generate reports to analyze these metrics. Reports show trends over time, allowing businesses to spot areas where they can improve. For example, they may notice that deliveries in certain areas are slower, or that certain times of day have more delays. By reviewing these reports, businesses can make better decisions and optimize their operations. Performance tracking helps businesses enhance efficiency and increase customer satisfaction. It also provides valuable insights for growth and improvement.

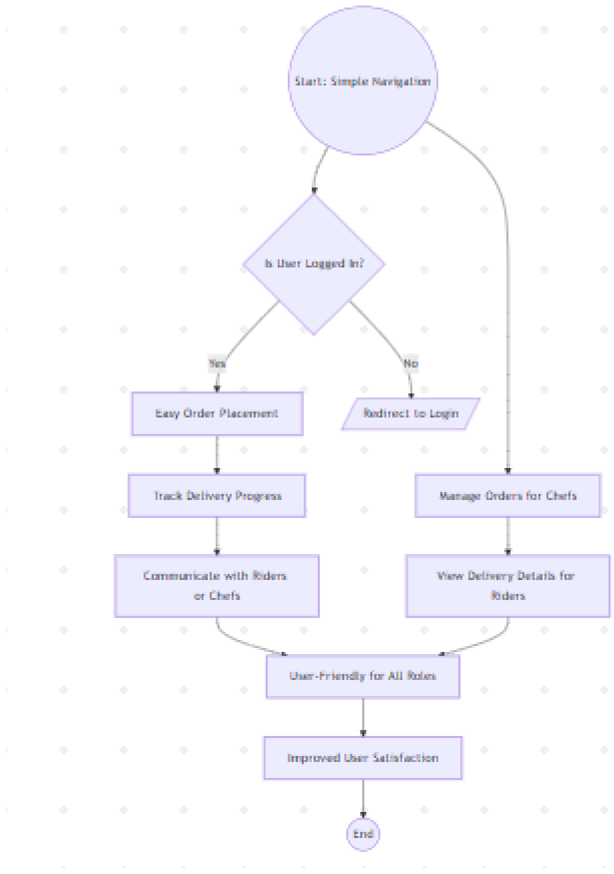
User Interaction Flow



Feature 6: User-Friendly Interface

A user-friendly interface is essential for making the app easy to use for everyone involved—customers, riders, and chefs. The app is designed with simple navigation, so users can quickly find what they need. Customers can easily place orders, track their deliveries, and communicate with riders or chefs. Chefs can manage their orders with minimal effort, seeing all the details clearly. Riders can view their deliveries and optimize their routes without confusion. The interface reduces the learning curve and makes using the app intuitive for new and experienced users alike. An easy-to-use interface ensures that the system is accessible to everyone, improving satisfaction and efficiency. This simplicity makes the app appealing to a wide range of users, from business owners to customers.

User Interaction Flow



3.2 User Stories & Requirements

User Stories

- As a customer, I want to place and track my order easily so I know when it will arrive.
- As a chef, I want to see incoming orders and update their status so I can manage food preparation.
- As a rider, I want to get delivery assignments and update status so customers are informed.
- As an admin, I want to see reports so I can analyze performance.

Acceptance Criteria

- Orders can be placed, accepted, or canceled smoothly.
- Real-time delivery tracking must show rider location accurately.
- Reports must be generated in PDF or Excel formats.
- Users can successfully leave ratings and comments after delivery.

Non-functional Requirements

- **Performance:** App should respond within 2 seconds for most actions.
- **Security:** User data must be safe using Firebase Authentication.
- **Reliability:** System should work with 99% uptime.
- **Scalability:** Apps should support many users without slowing down.

4. **Technical Specifications**

4.1 Architecture

Overview Diagram: A high-level system architecture diagram that outlines components such as the mobile client, backend services, APIs, databases, and third-party integrations.

Design Patterns

The Belly Shack Order & Delivery Management System follows the MVVM (Model-View-ViewModel) design pattern to keep the app organized and easy to manage. In this pattern, the Model handles all the data like orders, deliveries, and user information. The View is what users see on the screen and how they interact with the app. The ViewModel works between the Model and the View, taking care of data processing and sending updates to the screen. This setup helps keep the code neat and makes it easier to change or fix things later. It also makes it simple to test parts of the app without breaking others. Developers can work on one part of the app without worrying too much about the rest. Using MVVM makes the app more stable and easier to maintain in the long run.

4.2 Platform-Specific Considerations

iOS and Android Guidelines

The Belly Shack Order & Delivery Management System is designed mainly for Android and follows Google’s Material Design to make the app simple and easy to use. It uses familiar icons, buttons, and navigation so users can understand and use it without confusion. The layout and design are made to work well on different screen sizes, especially on smartphones. Even though iOS is not yet supported, the design is planned in a way that we can add iOS support in the future. All the screens are clean and clear so customers, chefs, and riders can use the app smoothly. The design also considers users with special needs to make the app more accessible. Common design standards are used so users feel comfortable using the app. These choices help users move from one feature to another without trouble. A stable experience is important since users need to manage orders, track deliveries, and send feedback. Overall, the design supports the main goal of the app: to make food ordering and delivery easier and faster.

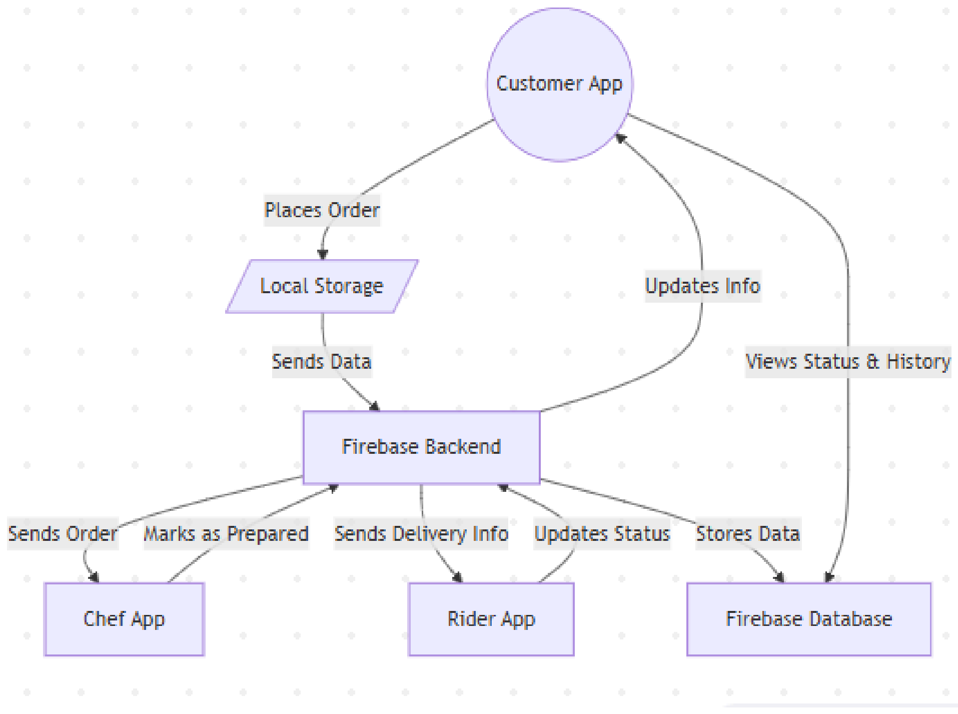
Hardware & OS Compatibility

- Minimum OS: Android 7.0 (Nougat)
- Recommended OS: Android 10+
- Compatible Devices: Most smartphones with 2GB RAM and above
- Network: Requires stable mobile data or Wi-Fi connection

4.3 Data Management

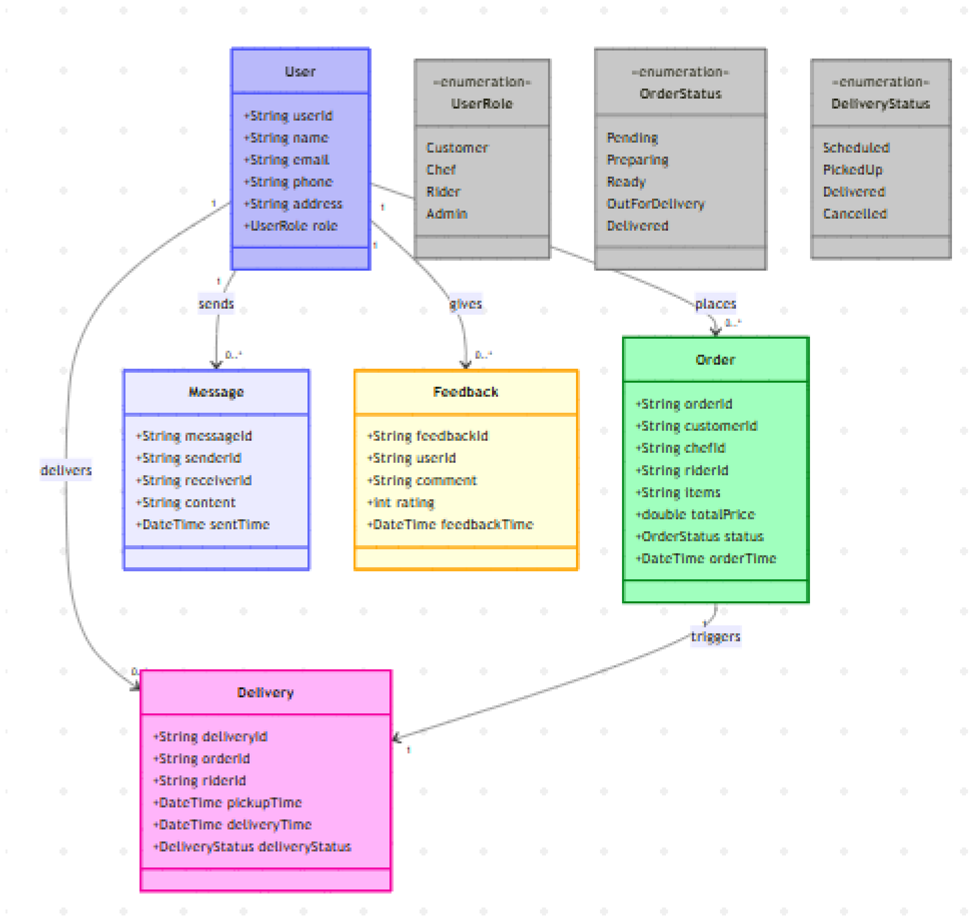
Data Flow Diagrams

The data flow in the Belly Shack Order & Delivery Management System shows how information moves between the mobile app, local storage on the device, and the back-end systems like Firebase. When a customer places an order, the app first stores the data temporarily on the device and then sends it to the server through a secure internet connection. The back-end processes the order and sends it to the chef's app so they can start preparing the food. At the same time, the rider receives the delivery details through their app. As the rider updates the status, like "picked up" or "delivered," the data is sent back to the server and shared with the customer in real time. This cycle continues for each new order. The system also stores order history, feedback, and user details safely in the database. This smooth flow of data helps the app work properly, avoid delays, and keep everyone updated.



Database Schemas

In the Belly Shack Order & Delivery Management System, database schemas help organize how data is stored and managed both on the user's device (local database) and on the server (Firebase). The local database stores temporary data like current orders, user login sessions, and app settings so the app works smoothly even when the internet is slow or not available. On the server side, the Firebase database keeps more permanent data like user profiles, full order history, delivery records, messages, and feedback reports. Each piece of data is arranged into collections, such as users, orders, or deliveries, with each item containing fields like name, address, status, and timestamps. For example, the "orders" collection might include fields like customer ID, food items, total price, and delivery status. This structure makes it easier to search, update, and organize data quickly. All user data is connected using unique IDs to keep things accurate and secure. These database schemas are important for keeping the app organized and making sure everything runs without problems.



API Endpoints (Sample using Firebase REST)

The Belly Shack Order & Delivery Management System uses simple API endpoints to help the app talk to the server and manage data. The **GET Orders** endpoint is used to get a customer's order list by using their user ID. This allows the app to only show orders that belong to that specific customer. It helps customers view their past and current orders easily and keeps their data safe and separate from others.

POST Order: Endpoint is used when a customer places a new order. It sends the order details to the server where it is saved and shared with the chef. This makes sure the kitchen staff knows what to prepare. It helps make the process faster and reduces errors when handling many orders.

PATCH Update Status: Endpoint lets the system update the status of an order. For example, it shows when food is being cooked, ready for pickup, or already on its way. This helps riders know when to pick up orders and customers stay updated with real-time progress. Everyone can follow along without asking for updates, making the service more smooth and clear.

4.4 Security & Privacy

Authentication: Ensures that only legitimate users can access the system. For our app, we use Firebase Authentication, which allows users to sign in using their email, password, or phone number. This is important because it verifies the user's identity before they can interact with any part of the system. By relying on Firebase, we take advantage of a secure, well-established authentication service that protects user data. This method ensures that only real users can access their accounts and prevents unauthorized access.

Authorization: Determines what a user can do in the app based on their role. In our system, each user is assigned a specific role, such as chef, rider, or customer. Role-based access control (RBAC) ensures that each user only sees and interacts with features relevant to their role. For example, a rider can track orders and see delivery locations, but cannot access order management features meant for chefs. This system helps maintain security and ensures that no user can misuse or interfere with functions that don't belong to them.

Data Encryption in Transit: To ensure data security during online communication, we use HTTPS encryption for data transmitted over the internet. HTTPS encrypts the data being sent between the user's device and the server, making it unreadable to anyone who might try to intercept it. This is crucial because it protects sensitive information like passwords and order details from hackers. By using HTTPS, we ensure that user data is safe even during transmission. This measure adds an additional layer of security for users when interacting with the app online.

Data Encryption at Rest: To ensure that stored information is protected from unauthorized access. All data saved in Firebase is encrypted to ensure that it cannot be read or altered by anyone without the correct permissions. This is especially important for user data, like personal information and order history. By using encryption, we provide a high level of protection against potential data breaches. This keeps sensitive information safe even if the storage system is compromised.

Privacy Compliance: We follow privacy guidelines to ensure that user data is handled responsibly and in compliance with regulations. Our app adheres to basic GDPR-lite principles, informing users about the type of data we collect and how it will be used. By providing clear details on data collection practices, users can make informed decisions about their privacy. Additionally, users have control over their information, allowing them to manage what is shared. This commitment to privacy ensures that users' personal data is protected according to established rules.

Privacy Policy: Before using the app, users must agree to our privacy policy, which outlines how their personal information is handled. This policy explains what data we collect, why we collect it, and how it will be used. Users can review and accept this policy to ensure they are comfortable with our data practices. We believe in transparency and giving users control over their information. This step is part of our commitment to maintaining user trust and protecting privacy.

Collected Data: Our app collects basic information such as the user's name, phone number, address, and order history. This data is necessary to provide users with a smooth and personalized experience. For example, we use the name and address to ensure the correct delivery of food orders. The order history helps track customer preferences and improve future service. We ensure that this information is stored securely and only used for improving the app's functionality.

Data Use and Sharing: The data we collect is used solely to improve the app and enhance the user experience. We do not share user information with third parties without explicit consent. This means that any information you provide remains private and is used only within the system. For example, order history helps us provide better recommendations, while contact details are used for communication. Our goal is to make sure users' personal information stays safe and is used only to improve the app's features.

4.5 Third-Party Integration

Firebase SDK: I can help with user login, syncing data in real time, and tracking how people use the app. This tool is important because it keeps user accounts safe and lets the app update information quickly without needing to refresh. It also helps developers understand what users do in the app, which makes it easier to improve the app over time. Firebase makes the app more stable and easier to manage. It plays a big role in keeping everything connected and running smoothly.

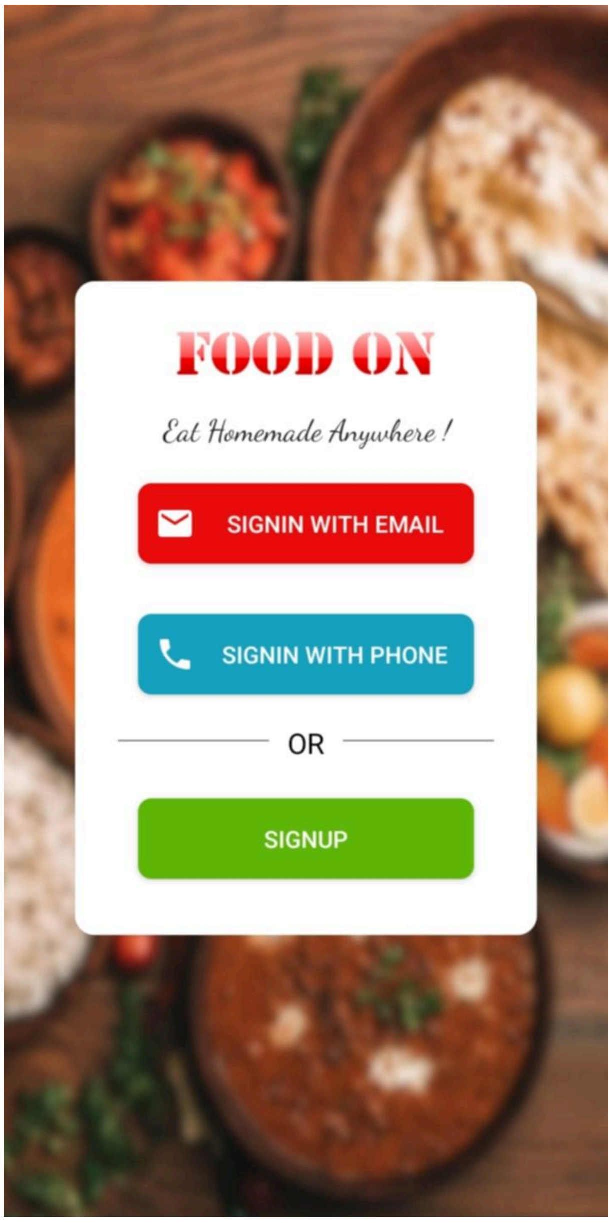
Google Maps SDK: It can help riders find the best routes and allows customers to track their orders live. This tool improves the delivery experience by showing exact locations and estimated arrival times. It saves time for riders and builds trust with customers since they can see where their food is. Google Maps also helps avoid delays and makes the whole delivery process faster and more efficient. It's one of the key tools that supports real-time tracking in the app.

reCAPTCHA SDK: It can add an extra layer of security by checking if the person using the app is a human and not a robot. It is often used during login or when sending feedback. This helps protect the app from fake users or spam. reCAPTCHA is simple for real users but strong enough to stop threats. Using it makes the app safer for everyone.

1. UI/UX Design Specifications

5.1 Wireframes & MOckups

A. Chef Interface



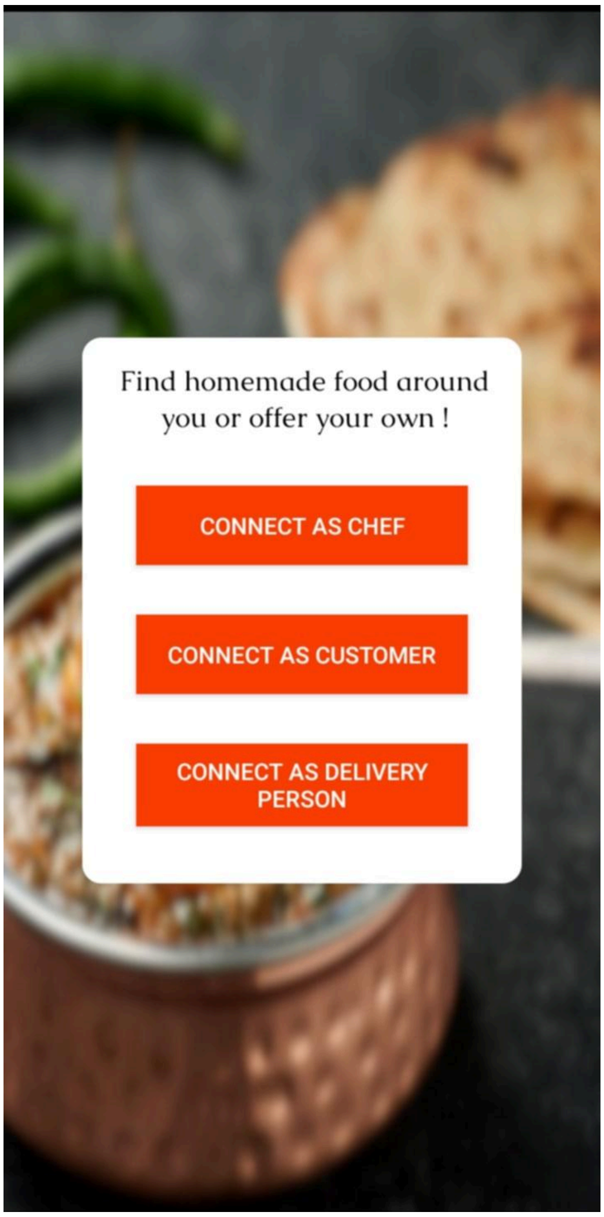

Find homemade food around you or offer your own !
CONNECT AS CHEF
CONNECT AS CUSTOMER
CONNECT AS DELIVERY PERSON

Figure 1. Login screen wireframe

Figure 2. Connect as Chef wireframe

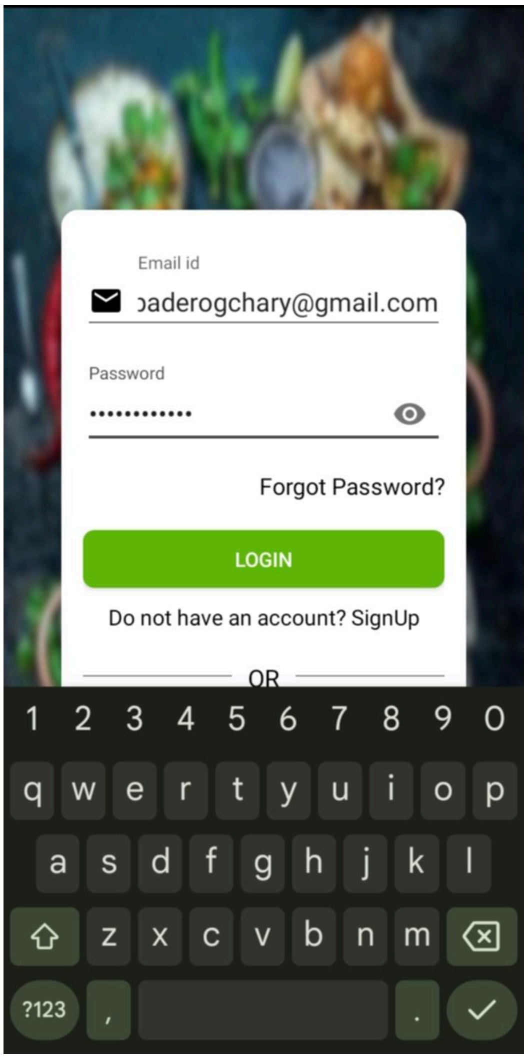


Figure 3. Login as Chef

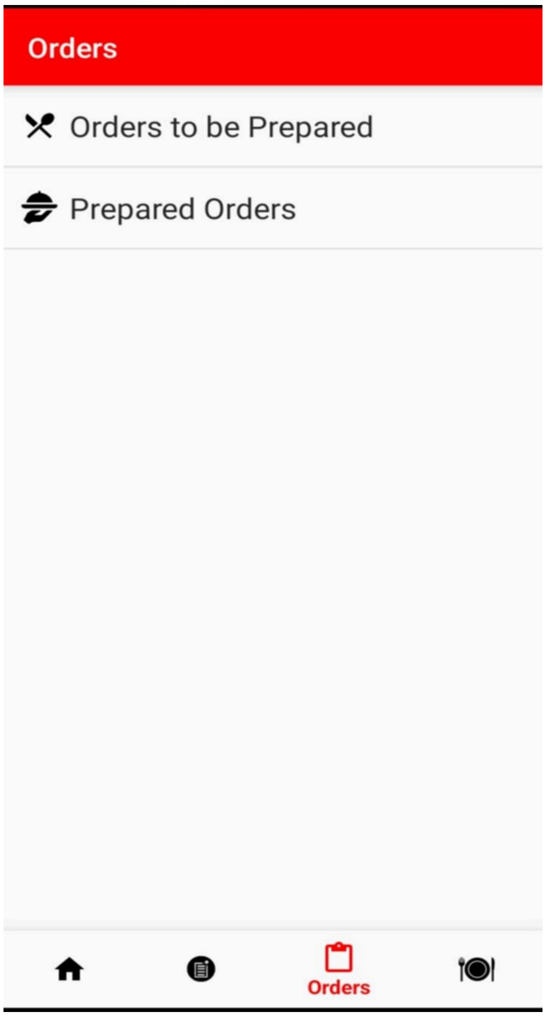
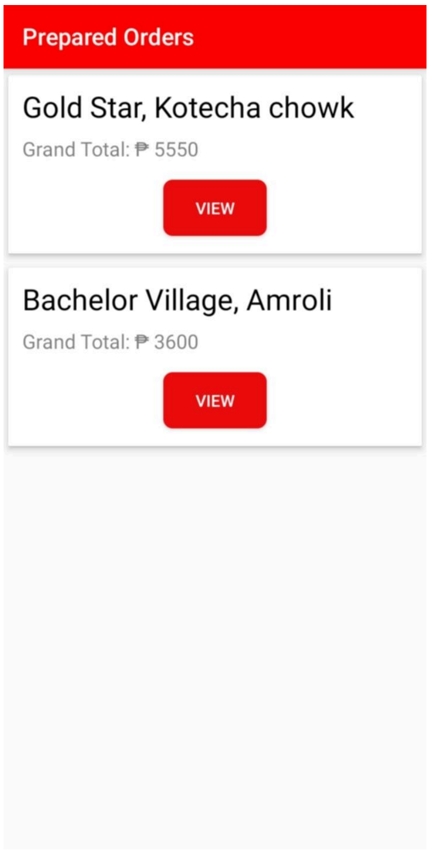


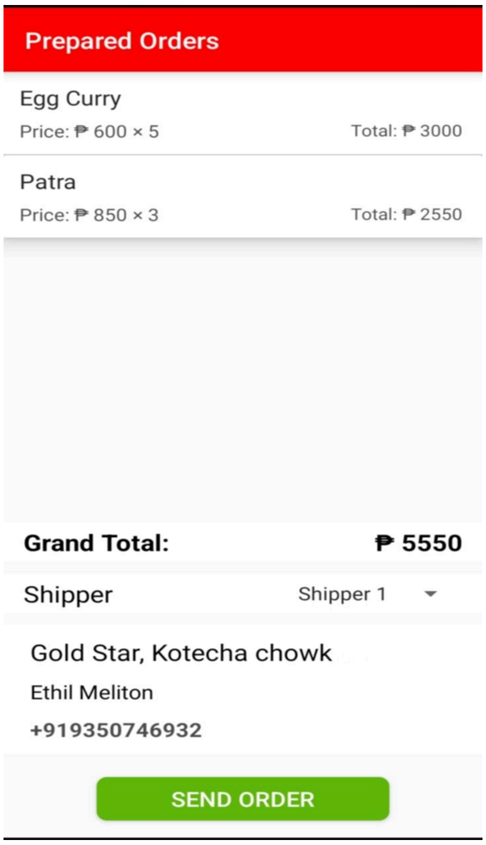
Figure 4. Order Management


Prepared Orders

Gold Star, Kotecha chowk
Grand Total: ₱ 5550

Bachelor Village, Amroli
Grand Total: ₱ 3600

Figure 5. List of order to be prepared


Prepared Orders

Egg Curry
Price: ₱ 600 × 5 Total: ₱ 3000

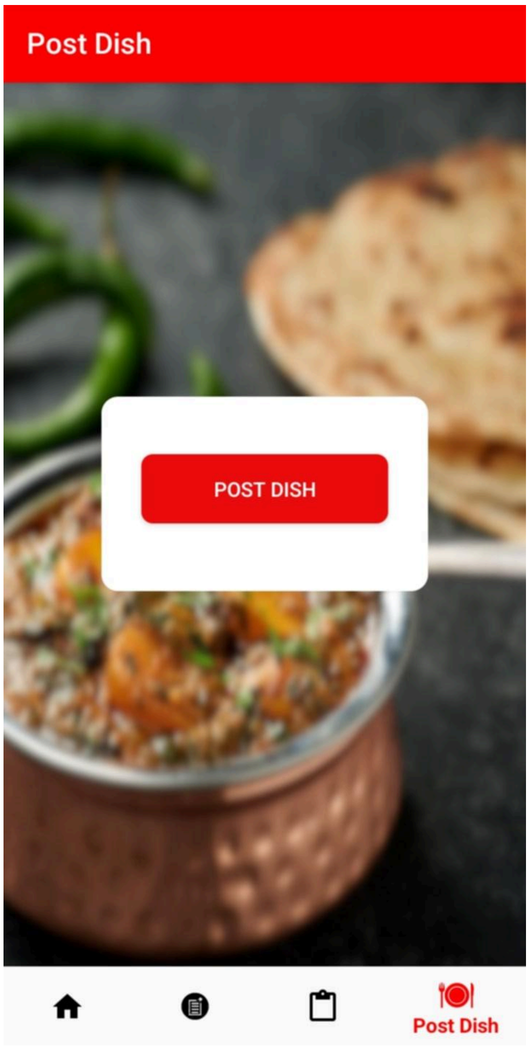
Patra
Price: ₱ 850 × 3 Total: ₱ 2550

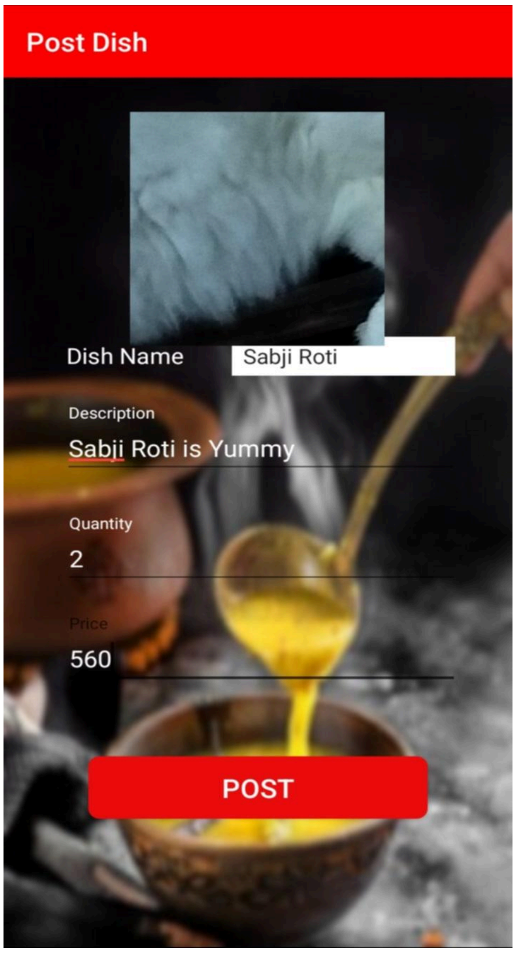
Grand Total: ₱ 5550

Shipper: Shipper 1

Gold Star, Kotecha chowk
Ethil Meliton
+919350746932

Figure 6. An approved order and to be assigned to the shipper





Dish Name Sabji Roti

Description
Sabji Roti is Yummy

Quantity
2

Price
560

Figure 7. Post Dish Management

Figure 8. Post Dish with specific description

B. Costumer

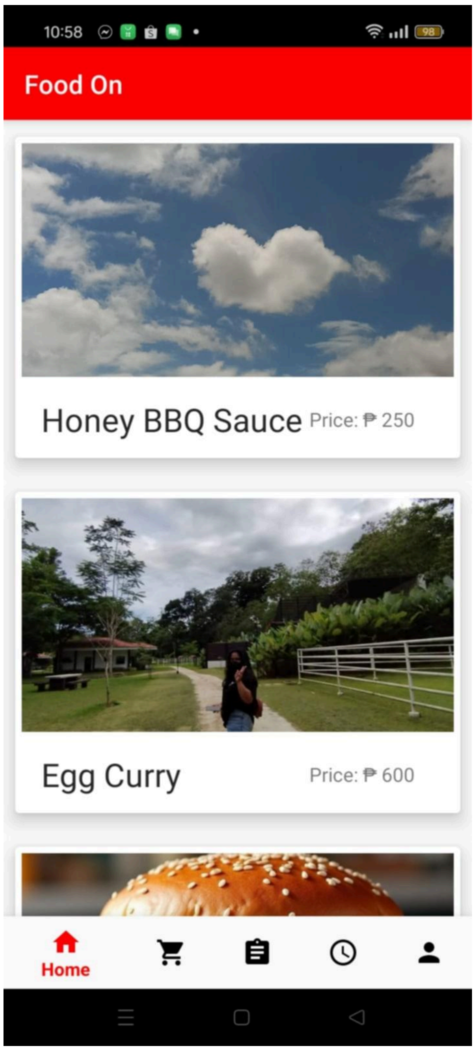


Figure 9. Customer Home Page

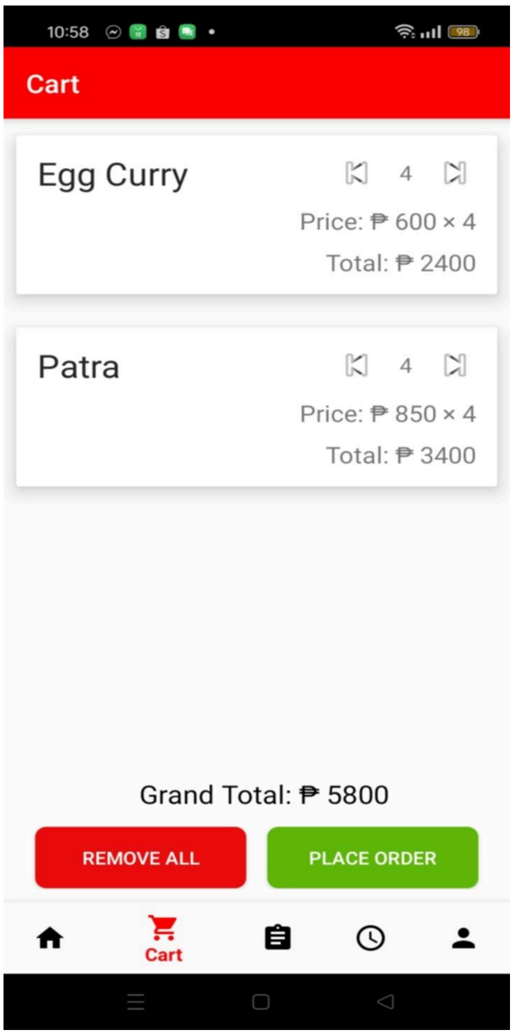
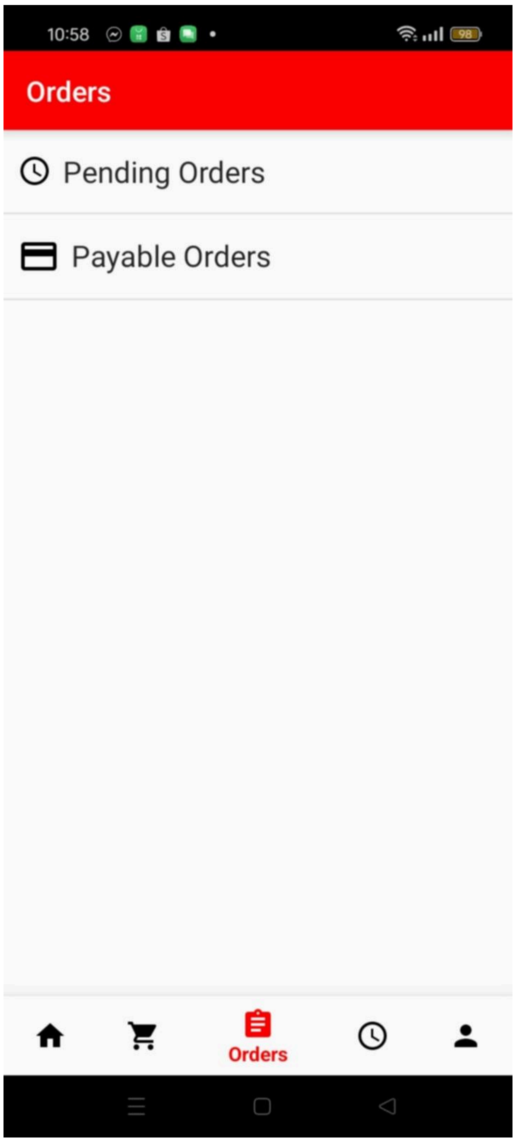


Figure 10. Cart Management



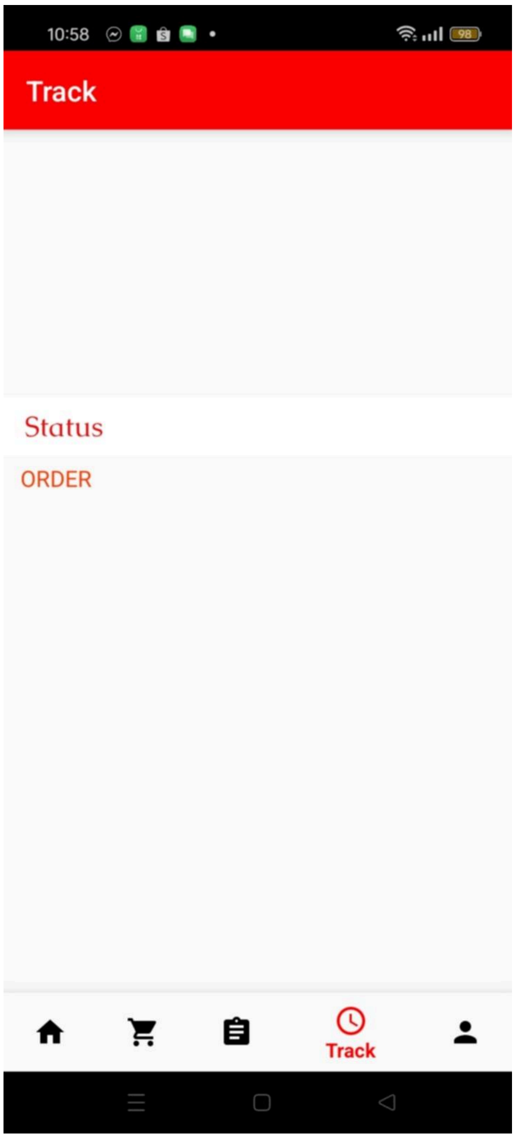
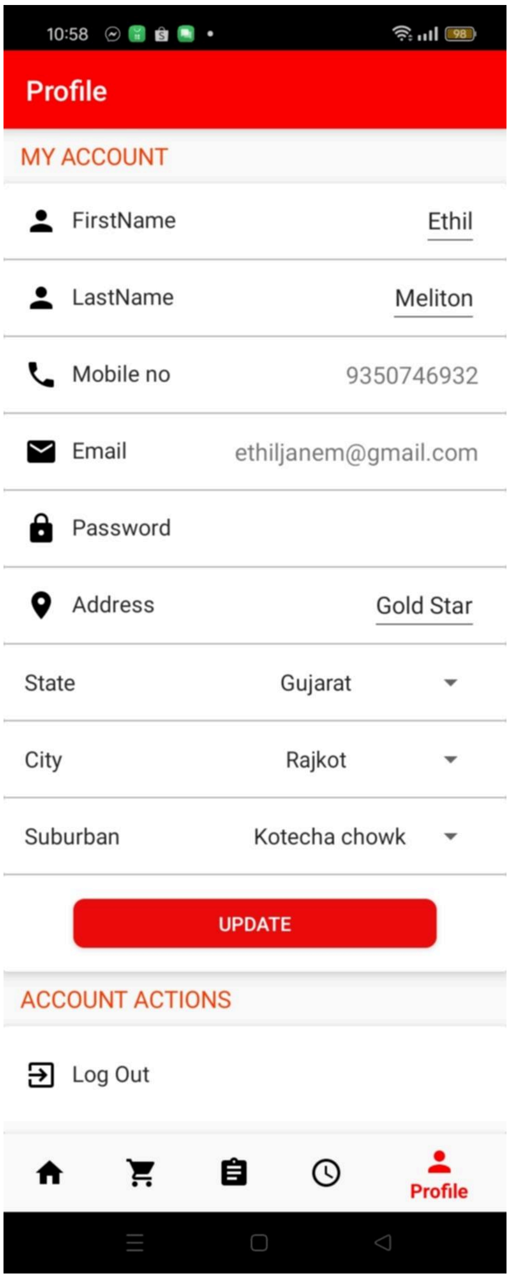


Figure 11. Order Management

Figure 12. Track Management



10:58

Profile

MY ACCOUNT

FirstName	Ethil
LastName	Meliton
Mobile no	9350746932
Email	ethiljanem@gmail.com
Password	
Address	Gold Star
State	Gujarat
City	Rajkot
Suburban	Kotecha chowk

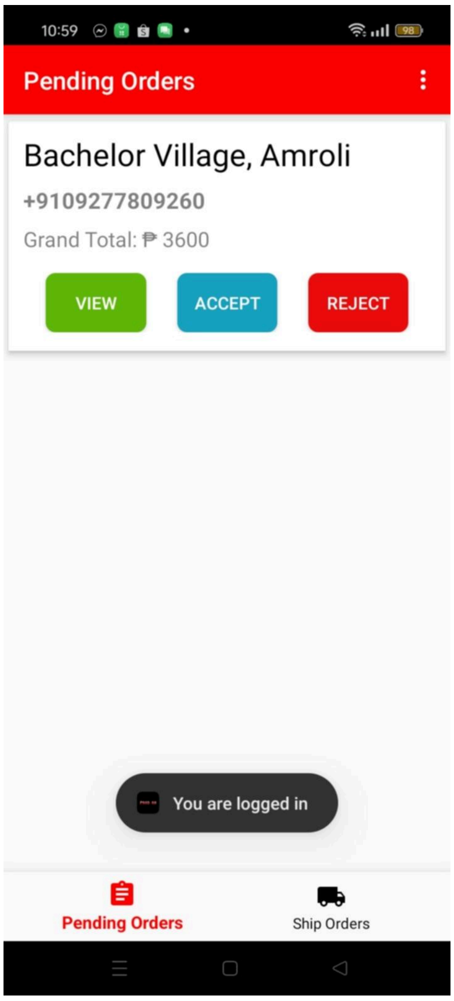
UPDATE

ACCOUNT ACTIONS

Log Out

Figure 13. Customer Profile

C. Delivery Person



Pending Orders

Bachelor Village, Amroli

+9109277809260

Grand Total: ₱ 3600

Figure 14. Order Managemen

2. Deployment & Maintenance

6.1 Deployment Plan

- Final Testing & QA
    - Conduct final regression and functional testing.
    - Run tests on various devices and OS versions to ensure compatibility.
- Store Listing Preparation
    - Prepare all assets: app icon, feature graphics, screenshots (various resolutions).

Appendices

7.1 Glossary

Technical Terms and Acronyms: Define key terms for clarity.

- API (Application Programming Interface)
    - A set of tools and protocols that allow different software components to communicate with each other.
- JSON (JavaScript Object Notation)
    - A lightweight format for storing and exchanging data, commonly used in APIs.
- Android Studio
    - The official Integrated Development Environment (IDE) for Android app development, provided by Google.
- APK (Android Package Kit)
    - The file format used to distribute and install apps on Android devices.
- AAB (Android App Bundle)
    - A publishing format that includes all compiled code and resources, letting Google Play generate APKs for different device configurations.
- SDK (Software Development Kit)
    - A collection of software tools and libraries needed to develop applications for a specific platform (e.g., Android SDK).
- Gradle
    - A build automation tool used in Android Studio to compile code, manage dependencies, and package APKs.
- Activity
    - A single screen in an Android app that users interact with (similar to a webpage in a website).

- Intent
    - A messaging object in Android used to request an action from another app component, such as starting a new activity.
- UI (User Interface)
    - The visual elements of the app that users interact with, such as buttons, text fields, and menus.
- XML (eXtensible Markup Language)
    - A markup language used to define layout files in Android development.
- Firebase
    - A platform by Google that offers services such as real-time database, authentication, cloud messaging, and analytics for mobile apps.
- Emulator
    - A virtual device that mimics an Android phone or tablet on your computer, used for testing apps without a physical device.

7.2 References & Resources

Haxxorsid. (n.d.). *Food order management system* [Source code]. GitHub. https://github.com/haxxorsid/food-ordering-system

FreeCodeCamp.org. (2023, February 16). *Savor the flavor with our food delivery app in Flutter & Dart | Mobile app development guide* [Video]. YouTube. https://youtu.be/z8NHvp4tbrQ?si=H04EVP3eoskHXk1b

CodeWithMazn. (2021, March 3). *FOOD DELIVERY APP || Demo || Java || Android Studio || Firebase* [Video]. YouTube. https://youtu.be/Pzw1wwh8ldM?si=BC7gJb_OSTv6XM7C

Tech Projects Arena. (2022, October 25). *Splash & start - Food ordering app with admin app in Android #1 - Android Studio project* [Video]. YouTube. https://youtu.be/TwLilW0fH9c?si=FwGO__gTp1FP_aR9

Mermaid. (n.d.). *Live editor for generating diagrams and flowcharts*. Mermaid JS. Retrieved May 14, 2025, from https://mermaid.live