TINDA-GO: A SARI-SARI STORE MOBILE APPLICATION ORDERING AND INVENTORY MANAGEMENT SYSTEM

A Capstone Project Proposal
Presented to the Faculty of the
Information and Communications Technology Program
STI College Davao

In Partial Fulfilment
of the Requirements for the Degree
Bachelor of Science in Information Technology

Karl Daniel D. Bulasa
Lemuel M. Morquin
Ronan Kenji C. Isuga

May 2025

ENDORSEMENT FORM FOR PROPOSAL DEFENSE

TITLE OF RESEARCH: **Tinda-Go: A Sari-Sari Store Mobile Application Ordering And Inventory Management System**

NAME OF PROPONENTS: Karl Daniel D. Bulasa
Lemuel M. Morquin
Ronan Kenji C. Isuga

In Partial Fulfilment of the Requirements
for the degree Bachelor of Science in Information Technology
has been examined and is recommended for Proposal Defense.

ENDORSED BY:

Dominic R. Bantigue
Capstone Project Adviser

APPROVED FOR PROPOSAL DEFENSE:

Jessiel Chris Hilot
Capstone Project Coordinator

NOTED BY:

Engr. Darwin Ian Guerrero
Program Head

May 2025

APPROVAL SHEET

This capstone project proposal titled **Tinda-Go: A Sari-Sari Store Mobile Application Ordering And Inventory Management System**, prepared and submitted by **Karl Daniel D. Bulasa, Lemuel M. Morquin,** and **Ronan Kenji C. Isuga**, in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology, has been examined and is recommended for acceptance and approval.

Dominic R. Bantigue
Capstone Project Adviser

Accepted and approved by the Capstone Project Review Panel
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Information Technology

Engr. Elvi Lito Ubas, CPE, MEEd
Panel Member

Engr. Darwin Ian Guerrero
Panel Member

Eric P. Ricablanca, MIM
Lead Panelist

Noted:

Jessiel Chris Hilot
Capstone Project Coordinator

Engr. Darwin Ian Guerrero
Program Head

May 2025

TABLE OF CONTENTS

INTRODUCTION

Project Context

In the Philippines, sari-sari stores have long been the foundation of neighborhood commerce, offering accessible and affordable goods to everyday customers. These small, family-run businesses are deeply rooted in Filipino culture and are often the first choice for household needs. Despite their importance, most sari-sari stores still rely on traditional, manual methods—like handwritten inventory, verbal sales recording, and cash-based transactions. This lack of digitalization places them at a disadvantage in a rapidly modernizing and mobile-first society.

As e-commerce and digital platforms rise in popularity, large retail chains and online grocery platforms continue to dominate the market by offering convenience, speed, and real-time accessibility. Unfortunately, smaller community-based stores are often excluded from this technological shift. They lack access to systems that could help streamline operations, record transactions, or even provide data insights that could improve profitability.

TindaGo is a mobile-based solution specifically designed to address these gaps by empowering sari-sari store owners through technology. The application serves as a digital assistant that helps in tracking inventory, logging purchases and sales, managing returns, and recording spoilage or damaged goods. It also allows customers to browse available products and place orders for pickup, providing a structured and user-friendly shopping experience.

By bringing digital tools into the hands of small store owners, TindaGo aims to improve operational efficiency, reduce losses, and create new revenue streams. The system encourages sari-sari store owners to embrace mobile technology not only for easier inventory and transaction management but also to better connect with their customers in a more modern, reliable way.

Purpose and Description

The main purpose of the TindaGo project is to develop a mobile application that digitizes and enhances the daily operations of sari-sari stores. These include inventory management, sales monitoring, order processing, and transaction tracking. With the support of this platform, store owners can operate with better accuracy and insight while customers experience more organized, convenient ordering.

TindaGo is not just an inventory app—it also functions as a sales and reporting system. It records detailed sales information, monitors product availability, and even keeps logs for returned, damaged, or spoiled items. This helps store owners manage their goods efficiently and identify trends that can guide future purchase decisions.

One of the unique features of TindaGo is its built-in transactional revenue model. The app generates income through a small percentage fee based on each product sold via the platform. This ensures sustainability of the app while remaining affordable and fair for sari-sari store users.

Unlike other platforms that focus heavily on delivery services, TindaGo is designed with a pickup-only approach to simplify logistics. Many sari-sari stores operate within walkable distance from their customers, making pickups more practical and cost-efficient. Customers will be able to see dynamic, and itemized listings

With TindaGo, store owners gain real-time visibility into their operations, while customers get a reliable and user-friendly way to order essential goods. The application bridges the gap between traditional micro-retail and modern digital commerce, making everyday community shopping more efficient and future-ready.

Objectives

General Objective

TindaGo's main objective is to develop a mobile ordering and inventory management application tailored for sari-sari stores, providing a streamlined platform that simplifies business processes such as stock monitoring, sales tracking, and customer order handling, while promoting accessibility and digital empowerment for local store owners.

Specific Objectives

- **To develop a sales and inventory module for sari-sari stores with the following submodule:**
    - **To develop a purchase order module**

      This module allows store owners to record and manage items procured for sale. It acts as a digital log for tracking inventory replenishments, supplier details, and purchasing dates, helping store owners efficiently plan stock levels**.**
    - **To develop a sales module**

      This module enables store owners to record both walk-in and in-app transactions. It tracks daily sales activities, provides transaction history, and supports better business decision-making through accurate and organized sales data.
    - **To develop a return goods stock module**

      This feature allows store owners to log items returned by customers due to defects, expiration, or dissatisfaction. It helps maintain customer trust and manages inventory adjustments associated with returns.
    - **To develop a damages and spoilages module**

      This module helps store owners record items that cannot be sold due to damage, spoilage, or expiration. It provides insights into product losses, enabling better inventory and quality control.

- **To develop a module for viewing and ordering available grocery items.**

  Customers can browse, select, and place pickup orders based on registered and nearest sari-sari stores in the provided map. Product listings will include itemized pricing, and descriptions.

- **To develop a store registration module for sari-sari store owners.**

  Registration will require documents such as a valid ID and business permit, subject to admin approval. Once verified, the store owner can manage product listings, monitor stock levels, and track their daily operations.

- **To develop an administrative panel module.**

  Admins will have access to approve or reject store registrations, monitor user and store activities, generate reports, and manage system-level settings for maintaining order and compliance.

- **To develop a customer feedback and rating module.**

  This module enables customers to leave ratings and reviews for sari-sari stores based on their order experience. Feedback will help store owners improve services, while admins can monitor reviews for quality assurance.

Scope and Limitations

TindaGo is a mobile-based sari-sari store management and ordering system designed to support sari-sari store owners and local customers within a specific community. The system aims to modernize traditional store operations by providing digital inventory management, sales tracking, and a convenient pickup-only ordering platform for customers.

Scope

The following key functionalities are included in the scope of this study:

- **Initial Market –** The TindaGo mobile application will initially be available only within Davao City. This geographic limitation allows the development team to focus on optimizing the app specifically for local sari-sari stores and customers in the area.

- **Target Users** – The app is designed for two primary user groups:

    - **Sari-sari Store Owners:** Small neighborhood store operators in Davao City who want to digitize their business operations, manage inventory, monitor sales, and handle returns or damaged goods.
    - **Local Customers:** Community members who want to browse available grocery items and place orders for pickup at their nearby sari-sari stores.

- **Store Registration and Management –** Store owners can register their sari-sari businesses in the app by submitting necessary documents for admin approval. Verified stores can then manage product listings, update stock quantities in real-time, and generate sales reports.
- **Inventory and Sales Tracking –** The app allows store owners to record inventory levels, log sales transactions, and manage product returns, spoilage, or damage. This helps maintain accurate stock data and reduces losses.
- **Pickup-Only Ordering –** Customers can browse product listings, see detailed pricing and place orders for pickup at the sari-sari store location.

- **Revenue Model –** TindaGo uses a transactional revenue model based on percentage fees applied per product and category sold through the platform, supporting the app's sustainability while keeping costs affordable for users.
- **Map-Based Store Browsing and Selection** – Customers can view a map displaying all registered sari-sari stores within their locality. This feature allows users to easily discover nearby stores, view their product listings, and select a preferred store for placing pickup orders.
- **Administrative Control –** Admin users have access to monitor all registered sari-sari stores, approve new registrations, and review sales and inventory reports to maintain system integrity and quality.
- **Mobile Platform –** The system will be developed and deployed exclusively for Android devices to maximize accessibility for sari-sari store owners and customers in the targeted community.

Limitations

While TindaGo provides valuable features for sari-sari store management and customer ordering, it has the following limitations:

- **Device Limitation –** The TindaGo application can only be accessed and used via mobile devices (smartphones and tablets). It will not be available on desktop or web platforms at this stage. For now, iOS devices will not be included yet.
- **Service Type –** The platform supports only a pickup-only ordering system; no delivery or shipping options are offered at this stage, simplifying logistics for stores and customers.
- **Manual Inventory Input –** Inventory and stock data rely on manual entry by store owners. There is no automated inventory scanning or synchronization with external systems.
- **Manual Sales Recording for Walk-ins -** Sales done via walk-in will be recorded through manual input of product names, quantities and customer names. The use of barcode scanners will not be implemented.

- **Internet Dependency –** Users must have an active internet connection for real-time inventory updates, order synchronization, and communication between stores, customers, and admins.
- **Store Participation –** Only sari-sari stores registered and verified by the admin within Davao City can participate in the platform. Unregistered stores cannot use the app.
- **Taxation and Compliance –** The system does not cover taxation processes, government financial reporting, or integration with regulatory compliance systems. Store owners remain responsible for legal compliance.
- **Market Scope –** The app's use is limited to sari-sari stores and customers within Davao City only. Expansion to other cities or regions is not planned in the initial deployment.
- **User Adoption and Digital Literacy –** The effectiveness of TindaGo depends on the willingness and ability of sari-sari store owners to adopt digital tools. Limited digital literacy among some users may impact full utilization of the app's features.

REVIEW OF RELATED LITERATURE/STUDIES/SYSTEMS

Related Studies and/or Systems

This chapter provides a review of relevant studies or systems used in this study.

Grodeliveryo: Mobile Application for Grocery, Purchasing and Delivery (Kaliappen & Aman, 2022)

Grodeliveryo was created to solve traditional grocery shopping problems at Valli Store, including manual inventory and cash-only payments. The app was developed using Unity and Visual Studio Code, with C# as the main language (Kaliappen & Aman, 2022). It digitalized grocery management, customer registration, cart ordering, and feedback collection. Customers could browse products, manage their carts, and receive order updates through real-time notifications. The system helped the store operate faster and improved customer satisfaction during and after the pandemic. Overall, Grodeliveryo made grocery shopping easier and more organized for both staff and customers.

The development process followed the Software Development Life Cycle (SDLC) Waterfall model. Requirements gathering, system design, coding, and testing were done step-by-step without skipping phases. Designs included use case diagrams and database modeling to structure the system clearly. Black-box testing was used to ensure that each feature worked as expected. User acceptance surveys showed that customers found the app easy to use and helpful. The project successfully shifted Valli Store from manual to digital operations with minimal issues.

Grodeliveryo supported two main order methods: delivery to the customer's home and in-store pickup. Customers could pre-order groceries for delivery, making it safer and more convenient during pandemic restrictions. Pickup allowed faster service by preparing orders in advance for collection. Payments were made through a simple cash-on-delivery system to remain accessible for all users. These options gave flexibility based on customer needs and helped reduce in-store congestion. By offering delivery and pickup, Grodeliveryo improved the overall customer experience and store efficiency.

Grab and Go: A Mobile Checkout Application for Groceries (Adwan, E. J., Almehri, A. A., Albuarki, L. F., Adwan, J. E., & Abdin, H. A., 2022)

Grab and Go was designed to eliminate long checkout lines in grocery stores by introducing a mobile "scan and go" system. Customers scan items while shopping, bag them directly, and pay through their mobile devices without needing to queue (Adwan et al., 2022). The developers used a Systematic Literature Review (SLR) and web content analysis (CA) to gather requirements and ensure a user-centered design. Key features included E-wallet payments, click-and-collect ordering, and loyalty point collection to improve customer experience. The app aimed to create a faster, safer, and more flexible shopping process for both customers and retailers. Overall, it focused on enhancing convenience and reducing congestion inside stores.

Grab and Go followed a structured Software Development Life Cycle (SDLC) approach in its creation. User requirements were collected through surveys, and system models like DFDs and ERDs were used for designing workflows. Testing with 273 users revealed a high usability acceptance rate of 98.9%, proving the system's effectiveness. Black-box testing further validated the functionality and reliability of the application. With its easy-to-use interface and efficient features, Grab and Go successfully improved the traditional checkout process. It became a strong solution for stores aiming to modernize customer service and minimize waiting times.

The application offered flexible delivery and pickup methods to meet different customer needs. Customers could use the click-and-collect feature to order online and pick up groceries without entering the store. Home delivery was also available, allowing users to receive their purchases conveniently. Payments were completed digitally through the E-wallet system, eliminating the need for physical cash or cards. These options gave shoppers more control over how they wanted to receive their groceries. By offering both pickup and delivery, Grab and Go provided a seamless and adaptable shopping experience.

Mobile-Based Customers Management System in Ayunadi Supermarket (Ginantra, N. L. W. S. R., Asana, I. M. D. P., Parwita, W. G. S., & Eriana, I. W. E., 2022)

The Ayunadi Supermarket system was built to improve customer management by combining a mobile ordering app with a web-based admin dashboard. Customers could register, browse products, place orders, and receive real-time promotional notifications using Firebase Cloud Messaging (Ginantra et al., 2022). The mobile app was developed with Flutter, while the web backend was created using PHP, connected through RESTful APIs. This system modernized the supermarket's operations by replacing offline promotions with digital communication. Customers were able to interact with the store easily through their smartphones. Overall, the solution aimed to increase convenience and strengthen customer loyalty.

The system development followed the Waterfall model of the Software Development Life Cycle (SDLC). Key stages included requirements gathering, system design using use case diagrams and ERD, coding, and systematic testing. Flutter ensured the mobile app worked across different devices, while PHP supported smooth backend operations. Black-box, compatibility, and portability testing methods were used to check system functionality. The admin dashboard allowed easy management of orders, customer profiles, and promotions. With this dual-platform setup, Ayunadi streamlined both customer engagement and internal management tasks.

Flexible delivery and pickup options were also integrated into the system to meet various customer needs. Customers could choose to collect their orders in-store or request home delivery for greater convenience. Payments and confirmations were handled digitally to speed up the process. This flexibility helped improve customer satisfaction, especially during times when minimizing contact was important. Promotions were pushed directly to customer phones, keeping users updated about special offers. Through these features, Ayunadi successfully enhanced its service quality and competitiveness.

Synthesis

The reviewed systems highlight the growing importance of mobile and digital technologies in improving grocery and retail operations. Grodeliveryo (Kaliappen & Aman, 2022) focused on simplifying grocery management, customer ordering, and delivery services through a mobile application, helping traditional stores shift from manual processes to automated systems. Grab and Go (Adwan et al., 2022) emphasized customer convenience by removing checkout lines and offering flexible payment, pickup, and delivery options using mobile scan-and-go technology. Meanwhile, the Mobile-Based Customers Management System for Ayunadi Supermarket (Ginantra et al., 2022) showed the importance of integrating customer management and ordering systems with real-time communication through mobile applications and web dashboards.

Each system followed a structured Software Development Life Cycle (SDLC), commonly using the Waterfall model, ensuring that requirements gathering, design, implementation, and testing were systematically performed. Testing methods such as black-box testing and user acceptance surveys confirmed that these systems met the functional and usability expectations of their users. Technological tools like Flutter, PHP, C#, Firebase Cloud Messaging, and RESTful APIs were widely utilized to create reliable and flexible platforms capable of supporting modern delivery and pickup services, real-time notifications, and digital payments.

In summary, these studies demonstrated how digital systems could significantly enhance operational efficiency, customer satisfaction, and business competitiveness in the grocery and retail sectors. By offering flexible order fulfillment methods such as delivery and in-store pickup, and by ensuring secure, real-time interactions through mobile technologies, businesses can adapt more quickly to changing consumer behaviors. The proposed study will build upon these insights by developing an integrated system that improves customer interaction, ordering convenience, and business efficiency using the latest mobile and web technologies.

METHODOLOGY

Technical Background

Overview of Current Technologies to be Used in the System

The development of TindaGo: A Mobile Application for Sari-Sari Store Ordering with Inventory Management System is powered by modern technologies selected to meet the specific needs of sari-sari store operations and their local customers. The platform is designed to streamline inventory management, order processing, and sales tracking through a reliable and easy-to-use mobile interface.

To begin the development process, Figma is used for designing the app's user interface. It allows the team to collaboratively create clean, responsive layouts, ensuring the app is user-friendly and visually engaging from the start.

The frontend of the app will be developed using Flutter, which will allow the team to build the app quickly using one codebase. The app will mainly work on Android devices but will be ready for other platforms in the future. For the backend, Firebase Firestore will be used as the main database, which will help with real-time updates like inventory changes and order tracking. Hive will be used to save some data like cart items and order history, even when there's no internet.

To handle logins and user accounts, Firebase Authentication will be used. Users will be able to sign in with their email, and later on, social media logins may also be added. For updates and order notifications, Firebase Cloud Messaging (FCM) will send push notifications directly to the users' phones.

TindaGo will earn revenue through a small percentage per product sold. For now, payments will be done in cash upon pickup, but the team plans to add digital payment options like PayMongo in the future, so users can pay with GCash, Maya, or other e-wallets.

An admin dashboard will be built using Node.js and Express.js to manage store approvals, monitor activities, and generate reports. Jest will be used for backend testing to ensure system reliability. For collaboration and version control, the team will use Git and GitHub. Development will be done using Android Studio for mobile and Visual Studio Code for backend and UI components.

The development of TindaGo: A Mobile Application for Sari-Sari Store Ordering with Inventory Management System will adopt the **Agile model**, a methodology that supports flexibility, user involvement, and continuous improvement—ideal for building a mobile platform tailored to the daily needs of sari-sari store owners and local customers. Agile's iterative approach allows the development team to adapt quickly to user feedback and changing requirements. The model follows six phases: Planning, Design, Implementation, Testing, Review, and Launch *(Refer to page 27, Figure 1).*

The process begins with the Planning (Requirements and Analysis) phase. During this stage, the team identifies and analyzes the system's core requirements. Key functionalities include store registration and verification, product listing and stock monitoring, sales tracking, customer order placement for in-store pickup, return goods and spoilage logging, and admin-level reporting and account management. These requirements are compiled into the product backlog, a flexible list of prioritized tasks that evolves throughout development. Inputs from user research, interviews with store owners, and analysis of local business practices ensure that the system addresses real-world needs specific to sari-sari stores.

Next, in the Design phase, the team focuses on creating the system's visual and structural framework. User interface designs are created using tools like Figma, with attention to simplicity and usability for both store owners and customers. Supporting design tools such as use case diagrams and data flow diagrams define how different modules—such as Inventory Management, Sales Tracking, Customer Orders, and Admin Controls—interact with one another. This stage helps align the technical structure of the system with user workflows and business operations.

In the Implementation phase, features from the backlog are developed during short, time-boxed sprints. The team uses technologies such as Flutter for cross-platform mobile development and Firebase for backend services. Each sprint focuses on specific modules, like inventory logs or customer order forms. Daily stand-ups help the team coordinate, address challenges, and make quick adjustments. This allows development to stay on track while remaining adaptable.

The Testing phase ensures that every feature functions correctly and efficiently. The team conducts unit testing, integration testing, and user experience checks. Features like product management and sales recording are tested thoroughly to avoid system issues. Bugs and usability concerns are addressed before moving forward. This stage protects app reliability and data accuracy.

After testing, the team enters the Review phase, where finished features are presented to stakeholders. Feedback is gathered to improve functionality and realign priorities. The product backlog is then updated based on this feedback. This continuous loop keeps development grounded in real-world use. It also helps the system grow with user expectations.

Finally, approved features move to the Launch phase, where they are deployed for live use. Customers can place pickup orders while store owners manage products, monitor sales, and log spoilage. The team conducts a retrospective at the end of each sprint to evaluate their performance and plan improvements. This continuous improvement ensures the app evolves smoothly.

Through this Agile process, TindaGo is developed to be practical, user-centered, and responsive to the unique needs of local micro-retailers. Its iterative development ensures that each version brings meaningful progress while remaining flexible to community feedback—making it an ideal solution for modernizing the sari-sari store experience without relying on delivery services.

Calendar of Activities

The Gantt chart represents the summary of activities. Listed are the activities and opposite them are their duration or period of execution. The BLUE indicate the activity has started or is not yet finished, while the YELLOW indicate the task is on progress or finished *(Refer to page 27, Figure 2).*

To begin, the developers brainstormed ideas for a feasible and relevant system. This took place in the early weeks of February and was followed by title creation and title defense to finalize the proposed system. Once approved, the data gathering stage began, where information relevant to the system was collected through interviews and document reviews.

The developers then wrote the Project Context, Purpose and Description, and Objectives, which served as the foundation of the proposed system. Scope and limitations were also identified to define the project's boundaries. Revisions for the Purpose and Description, Objectives, and Scope and Limitations were completed in early April to ensure clarity and alignment with the proposal requirements.

To further support the study, the Review of Related Studies/Systems was conducted to analyze existing systems and identify gaps that the project could address. This was followed by the completion of the Technical Background, and the Requirements Gathering phase, where functional and non-functional requirements were identified.

After gathering, the team performed a Requirements Analysis to refine the project’s system requirements. Diagrams such as process flowcharts and use case models were created to provide a visual representation of the system operations.The next steps involved collecting and organizing necessary Resources and completing the Requirements Documentation. An Appendix was compiled to include relevant supporting materials.

Finally, the Proposal Defense was scheduled in May to formally present the finalized proposal to the panel for review and approval.

Resources

- **Hardware**

  This section outlines the hardware resources needed for the project's development, testing, and deployment phases. Two main devices are used, ensuring the system performs well across a variety of setups *(Refer to page 25, Table 1)*

- **Software**

  The software resources include: The project will rely on a combination of stable and widely used software to ensure a smooth development workflow. These tools cover all stages of system design, development, and testing *(Refer to page 26, Table 2)*

Requirements Gathering and Analysis

The development of **TindaGo: A Mobile Application for Sari-Sari Store Ordering with Inventory Management System** began with a structured and community-focused requirements gathering and analysis phase. This phase was critical in defining system features that cater specifically to the daily operations and unique needs of sari-sari store owners in Davao City.

The initial plan was to conduct a series of direct interviews with sari-sari store owners to gather in-depth insights about their inventory practices, business documentation requirements, and challenges faced in day-to-day operations. However, the development team encountered a significant hurdle during this phase. Despite outreach efforts, several store owners were unable to participate due to their busy schedules and the demands of running their stores. Others expressed willingness but could not accommodate the interview sessions at the proposed times. These constraints limited the number of formal interviews the team could conduct.

Despite these limitations, the team was able to engage with a select number of sari-sari store owners over a two-day period to gather essential input that would inform the system's design. On the first day, the focus was on identifying the legal documentation required for operating a sari-sari store. Through these interactions, the team learned that store owners must secure a Barangay Business Clearance from their local barangay office, which certifies that their business is permitted to operate within the community. Additionally, they need a Business Permit issued by the city government to ensure regulatory compliance and legitimacy. For those using a business name different from their own, a DTI Business Name Registration is necessary. Finally, a valid government-issued ID—such as a Philippine passport, driver's license, or UMID card—is required to verify the identity of the business owner.

These findings directly influenced the development of the Store Registration Module in TindaGo. This module is structured to require store owners to upload digital copies of all necessary documents during the registration process. The system assigns the responsibility of reviewing and validating these submissions to the platform's admin users. Only upon successful verification of these documents will the store be approved and granted access to the application.

On the second day, the team explored how sari-sari store owners typically manage their inventory. Many respondents described traditional practices, such as manual inventory logging using notebooks or paper records, where product quantities and sales are tracked manually. Others mentioned relying on visual inspection, physically checking shelves to see which items need restocking. Some store owners use mental tracking, depending on memory to monitor fast-moving products and identify low stock items. A few participants also shared that they only restock once items are reduced to a threshold of about three to five pieces.

In response to these inventory challenges, the TindaGo system was designed to include an Inventory Management Module. This module supports real-time stock monitoring, enabling store owners to update and view current product quantities directly through the app. It features an automated low stock alert that notifies users when inventory levels are critically low. Additionally, the module maintains inventory logs, offering a historical view of stock changes and trends that help store owners make better restocking decisions.

By grounding the system requirements in real-world practices and documented needs, even through a limited sample size, the development team ensured that TindaGo is a relevant and responsive solution. The insights gathered during this phase laid a strong foundation for a system that not only simplifies inventory and order processes but also supports legal compliance and operational efficiency for sari-sari store owners.

Requirements Documentation

This section outlines the initial user interface (UI) design of the TindaGo mobile app, highlighting key screens such as the login process, customer home screen, and store owner dashboard. These mockups offer a visual guide on how users will interact with the system. Each screen reflects the core features of TindaGo, from user verification to product browsing and sales monitoring. The UI emphasizes ease of use, clarity, and mobile responsiveness to ensure smooth navigation for both customers and store partners. The goal is to make daily transactions efficient and accessible.

TindaGo Login Screen

The login and verification process begins by prompting users to enter their mobile number for authentication. After submitting the number, users receive a one-time password (OTP) via SMS for verification. If an account already exists, users must enter their password; otherwise, they can proceed to register. The design uses soft backgrounds, large input fields, and vibrant buttons to create a welcoming and accessible layout. Options like "Resend OTP" and "Change Phone Number" are also included for user convenience. This secure multi-step process helps prevent unauthorized access while keeping the experience simple and fast *(Refer to page 28, Figure 3)*

TindaGo Home Screen (Customer View)

The customer home screen allows users to browse available grocery items from local sari-sari stores. Users can set or view their current location at the top and use the search bar to find products. The screen displays grocery items like "Piattos" in a grid format with clear images and labels, making it easy to scan and choose products. There is no delivery option shown, as TindaGo focuses on in-store pickup or reservation. The navigation bar at the bottom gives quick access to Home, Categories, Cart, and Menu. The layout is clean and user-focused, allowing for fast product selection and seamless navigation *(Refer to page 28, Figure 4)*

TindaGo Store Dashboard

For sari-sari store owners, TindaGo provides a personalized dashboard that combines analytics with daily store operations. At the top of the dashboard is a sales tracking graph, which visualizes performance trends over time—helping store owners make data-driven decisions. Below this, the interface features clickable tiles for managing different modules: Products, Orders, Pick-up Orders, Transactions, Sales, Items, and Revenue. Each tile is clearly labeled and designed for touch interaction, ensuring ease of use on mobile devices. This layout enables store owners to monitor their business, update product availability, and analyze earnings all in one place. The dashboard is optimized for mobility, allowing operations to continue whether at home or in-store *(Refer to page 29, Figure 5)*

Design of Software, System, Product, and/or Processes

The design phase of TindaGo was strategically planned to address the unique operational needs of sari-sari stores in the Philippines, with an emphasis on inventory tracking, sales monitoring, order processing, and customer management. The system was developed with a focus on usability, modularity, and scalability—ensuring the app supports the day-to-day needs of local store owners and customers.

To visualize how information flows through the system, we created Data Flow Diagrams (DFDs) for major processes. These include customer registration, store registration, order management, and admin dashboard activities. Each diagram helped map out how data moves between users, the system, and storage. For instance, store orders flow from customer input to store confirmation and system logging. These diagrams ensured process clarity and guided our backend development *(Refer to pages 29–32, Figures 6–11).*

Use Case Diagrams were also developed to outline system functions based on user roles. Identified users include customers, store owners, and admins. Each diagram illustrated typical actions such as placing orders, updating inventory, or approving registrations. This helped define the necessary features and interactions for each role. The diagrams also ensured that no key function was overlooked during planning *(Refer to pages 33–35, Figures 12–18).*

For User Interface (UI) Design, the team used Figma to create interactive wireframes and high-fidelity prototypes. The UI emphasizes clean navigation and user-friendly workflows, especially for Android devices, which are widely used in the Philippines. Screens were designed for modules such as the store dashboard, inventory tracking, order listings, and sales history—providing sari-sari store owners with a simple but powerful tool for managing operations.

In terms of system architecture, TindaGo adopts the Model-View-Controller (MVC) pattern to ensure modular, organized, and maintainable code. The Model layer handles all business logic and data management tasks, including product listings, user profiles, store records, inventory levels, and sales tracking. The View layer focuses on

displaying a user-friendly interface using design outputs from Figma, tailored specifically for mobile Android devices. The Controller manages user actions, links the model and view, and executes system logic such as confirming orders or updating inventory. This architectural pattern allows for scalable development and easier debugging, as each component operates independently but cooperatively.

The system also applies a procedural design approach to handle sequential business processes such as store registration, product inventory updates, order confirmation, and customer feedback collection. Organized into functions and control flows, this design ensures smooth backend operations including validating store approvals, adjusting stock after each order, processing returns, and generating sales reports. This structure not only simplifies task execution but also supports clear logging and error handling in real-time.

TindaGo's database is built using Firebase Firestore, a cloud-based NoSQL system that organizes data into collections and documents for real-time updates and high scalability. The design centers around key collections such as Products, Sales, Purchase History, Inventory Logs, and Damages and Spoilage, each representing a core function of sari-sari store operations. For example, the Products collection stores details like price, stock, and last restocked date, while Sales tracks transactions with items sold and total amounts. Inventory Logs and Low Stock Alerts help monitor and manage stock changes and shortages in real time. Additional collections like Return Good Stocks and Expenses ensure accurate tracking of returns and operational costs. A basic Users collection supports role-based access, allowing for secure management by cashiers or admins *(Refer to page 36, Figure 19).*

TindaGo's design process reflects a well-structured approach that combines technical architecture, functional modules, and user experience principles. By aligning mobile-first interface design with robust backend logic and real-time data handling, the system is both responsive and scalable—providing sari-sari store owners with a modern yet accessible tool to manage daily operations more efficiently.

REFERENCES

Adwan, E. J., Almehri, A. A., Albuarki, L. F., Adwan, J. E., & Abdin, H. A. (2022, October). A mobile checkout application for groceries. *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, 1–6. https://doi.org/10.1109/ICDABI56818.2022.10041661

Brand, C., Schwanen, T., & Anable, J. (2020). 'Online omnivores' or 'willing but struggling'? Identifying online grocery shopping behavior segments using attitude theory. *Journal of Retailing and Consumer Services, 57*, 1–18. https://doi.org/10.1016/j.jretconser.2020.102221

Ginantra, N. L. W. S. R., Asana, I. M. D. P., Parwita, W. G. S., & Eriana, I. W. E. (2022). Mobile-based customers management system in Ayunadi Supermarket. ADI Journal on Recent Innovation, 4(1), 86–101. https://doi.org/10.34306/ajri.v4i1.767

Kaliappen, K. A., & Aman, H. (2022). Grodeliveryo: Mobile application for grocery, purchasing and delivery. *Applied Information Technology and Computer Science, 3*(2), 757–775. https://doi.org/10.30880/aitcs.2022.03.02.046

Tukkinen, P., & Lindqvist, J. (2015). Understanding motivations for using grocery shopping applications. *IEEE Pervasive Computing, 14*(4), 38–44. https://doi.org/10.1109/MPRV.2015.65

APPENDICES

APPENDIX A

The following pages contains the Resources Agile Model, Gantt Chart, Initial UI Designs, Diagrams, Resource Persons, and Personal Technical Vitae of the researchers.

Table 1. Devices' Specifications

	DEVICE 1	**DEVICE 2**	**DEVICE 3**
Processor (CPU)	Intel® Core™ i5-4440 CPU	AMD Ryzen 5 5600X	Intel(R) Pentium(R) CPU G4560
Operating System	Windows 10 Pro	Windows 10 Pro	Windows 10 Enterprise
Memory	20GB	16GB	8GB
Storage	1TB	512GB	1TB
Monitor/Display	21" Full HD (1920x1080)Resolution	21" Full HD (1920x1080)Resolution	(1920x1080) Resolution
Mobile Device (Test)	Android Emulator via VS Code and AVD Manager	Android Emulator via VS Code and AVD Manager	Android Emulator via VS Code and AVD Manager
Other External Hard Drive and Cloud for backups			

Table 2. Software and Stable Release

SOFTWARE	STABLE RELEASE
Flutter SDK	3.19.3
Dart	3.3.0
Visual Studio Code	1.87.0
Firebase (Authentication, Firestore, Storage)	Managed via Firebase Console
Hive (Local Database)	2.2.3
Android Emulator (via Android Studio)	Giraffe (2023.3.1)
Git	2.44.0
Figma	Web app (Versioned continuously)
Socket.IO	4.7.2 (or latest stable)
Google Maps Flutter Plugin	2.5.0
PayMongo API (E-Wallet + Card Payments)	v1 (via REST API)
Jest	29.7.0
Node.js with Express.js (Optional)	Node.js 20.x LTS / Express 4.x

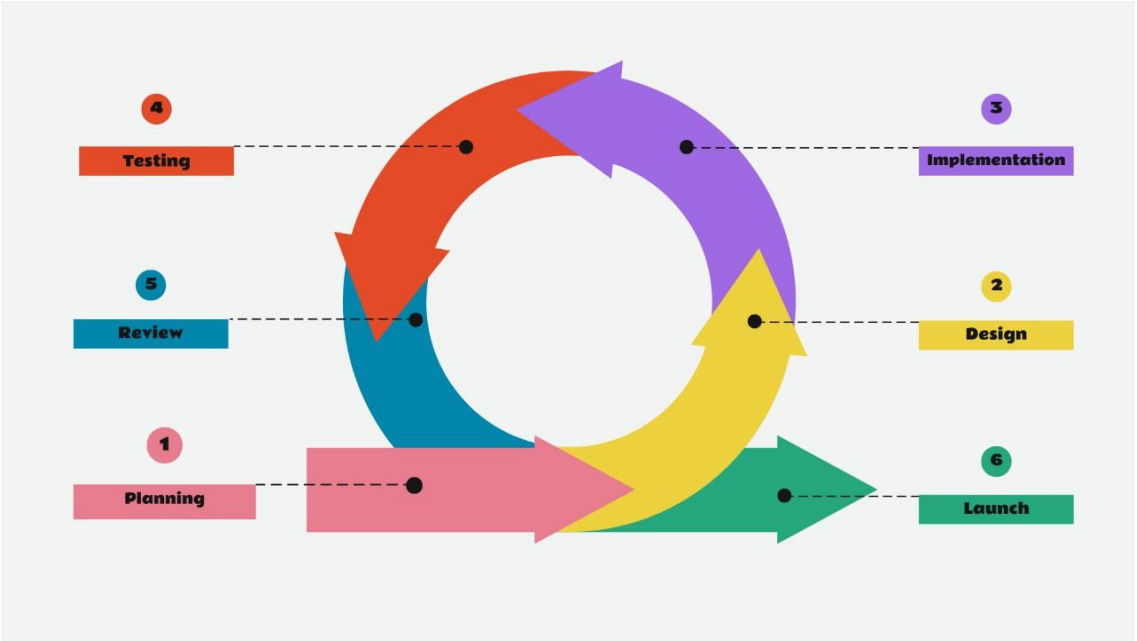

Figure 1: Agile Model

MONTH	FEBRUARY				MARCH				APRIL				MAY			
ACTIVITIES	WEEKS															
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Idea Brainstorming																
Title Creation																
Title Defense																
Data Gathering																
Project Context																
Purpose and Description																
Objectives																
Scope and Limitations																
Purpose and Description (Revisions)																
Objectives (Revisions)																
Scope and Limitations (Revisions)																
Review of Related Studies/Systems																
Technical Background																
Requirements Gathering																
Requirements Analysis																
Diagrams																
Resources																
Requirements Documentation																
Appendix																
Proposal Defense																

Figure 2: Gantt Chart of Activities

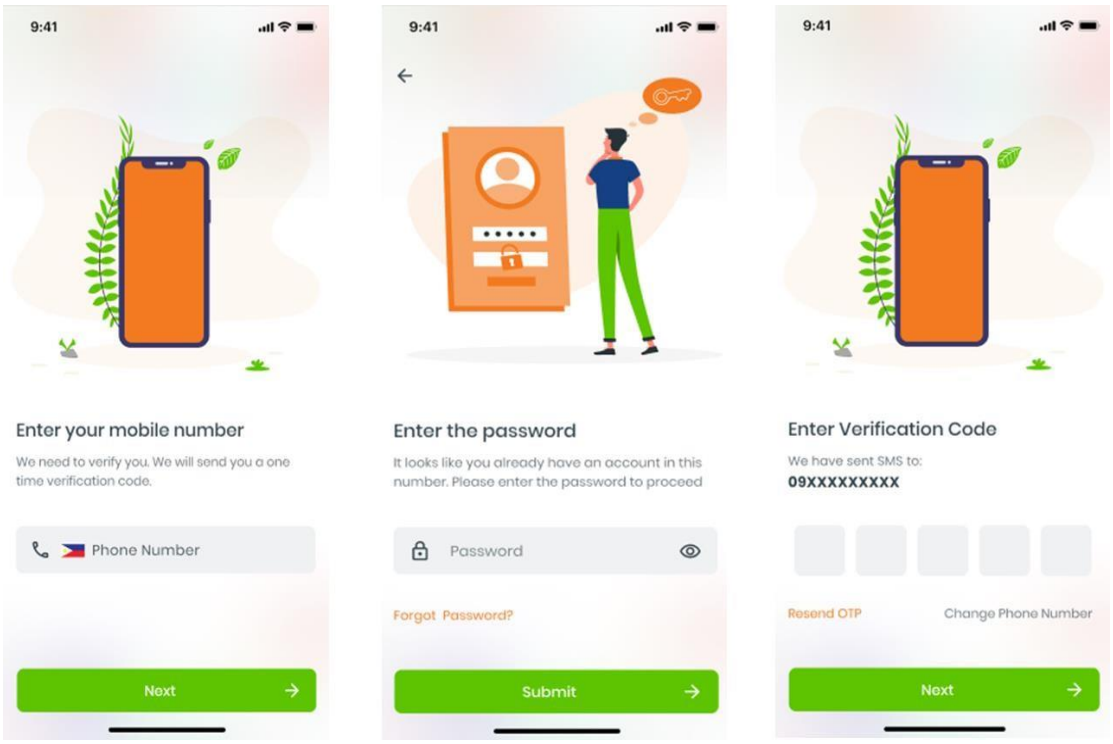

Figure 3: TindaGo Login Screen (UI Design)

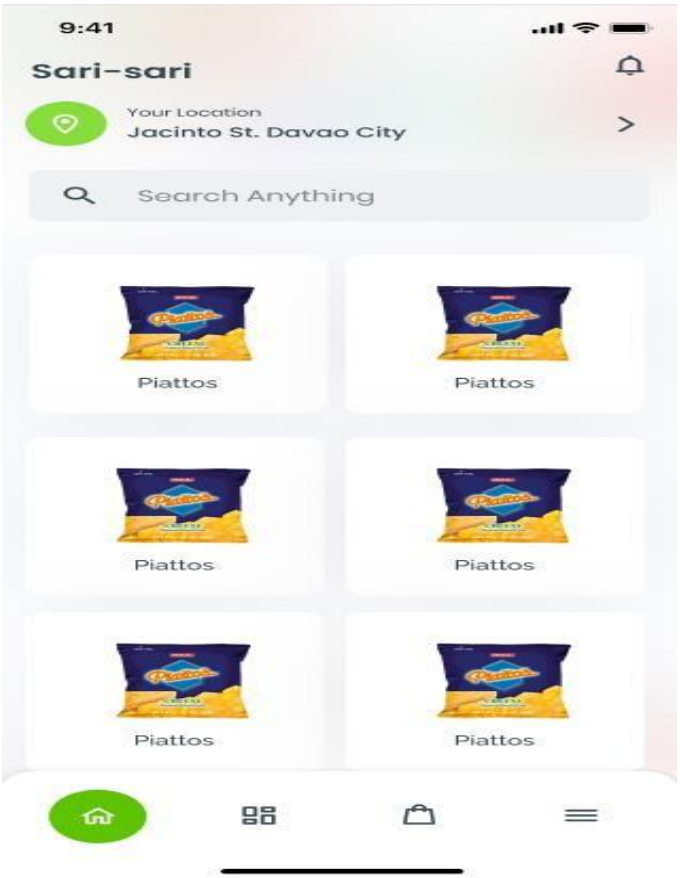

Figure 4: TindaGo Home Screen - Customer View (UI Design)

Figure 5: Store Dashboard (UI Design)

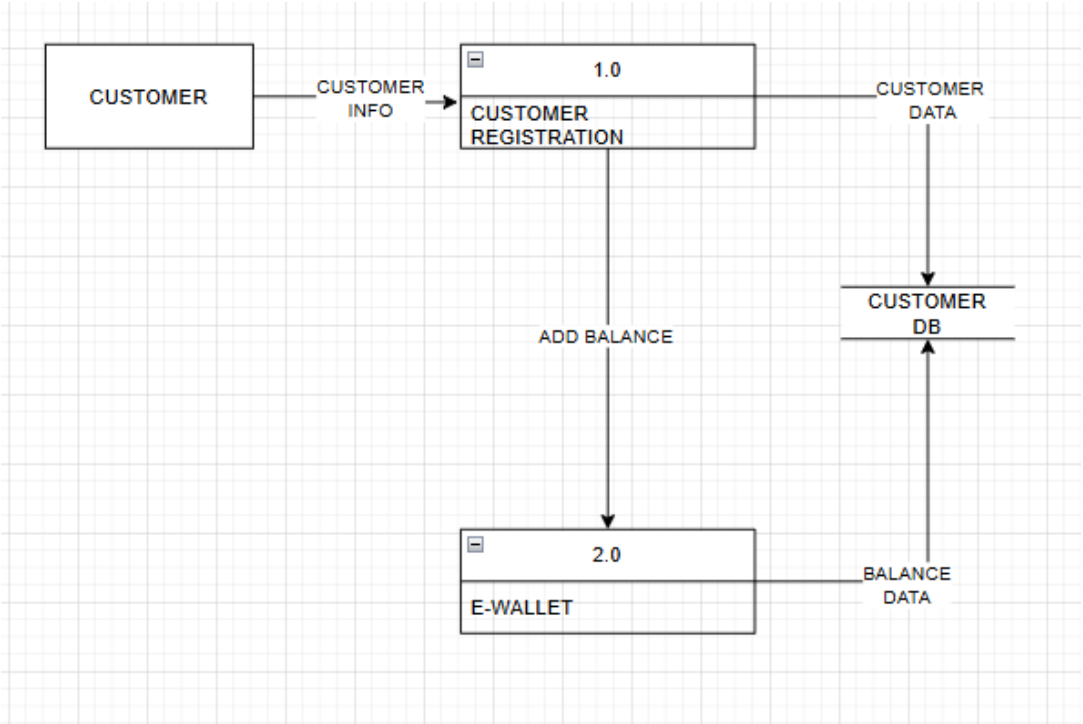

Figure 6: Customer Registration (DFD)

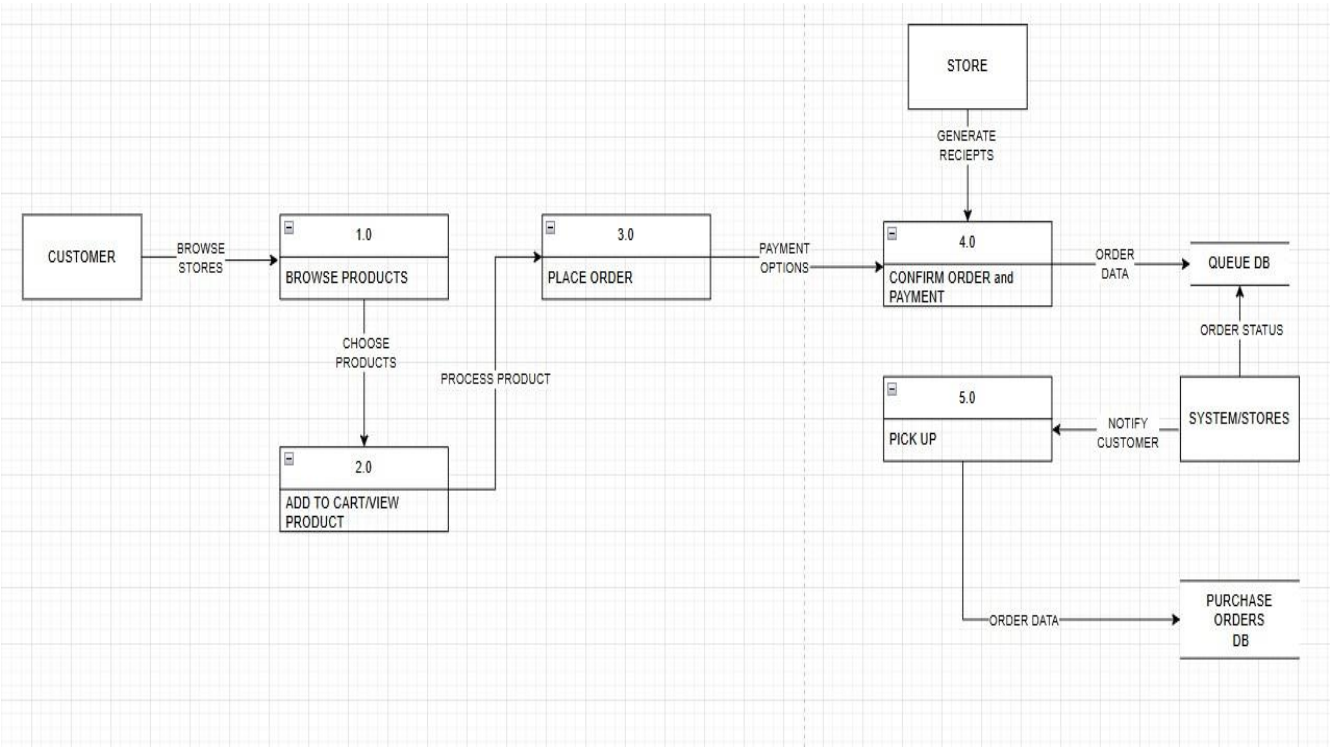

Figure 7: Customer Ordering (DFD)

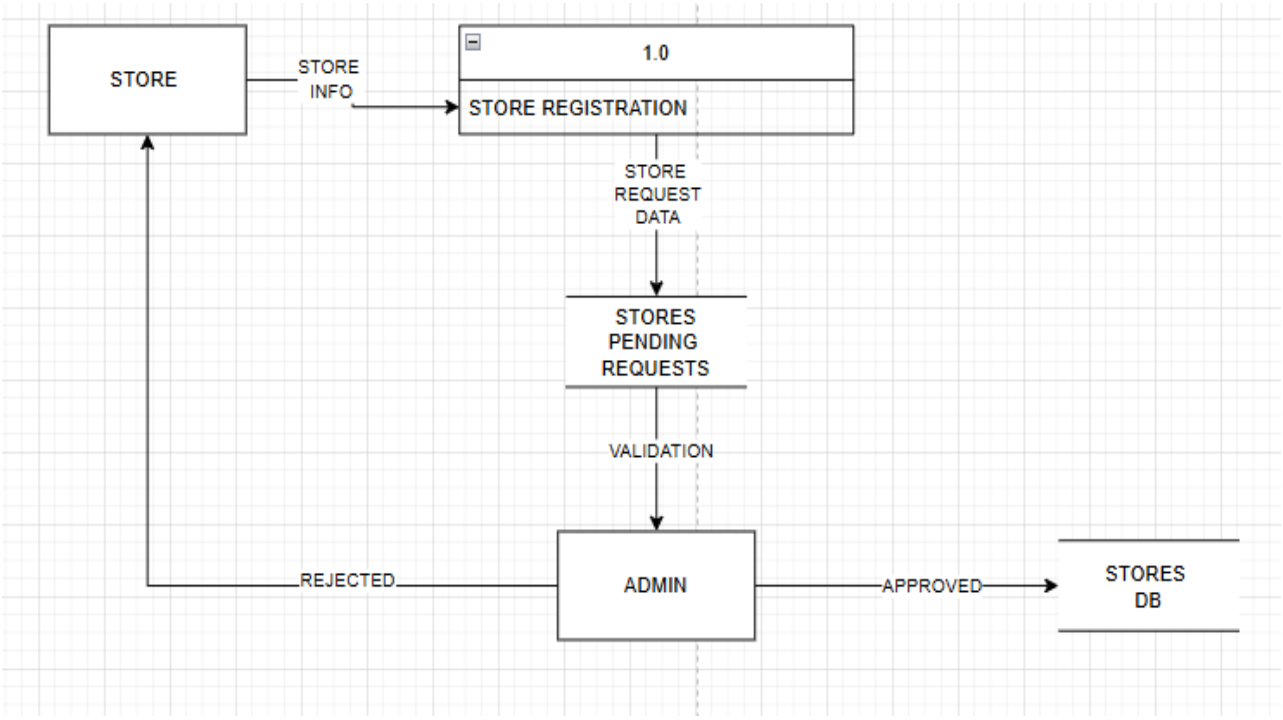

Figure 8: Store Registration (DFD)

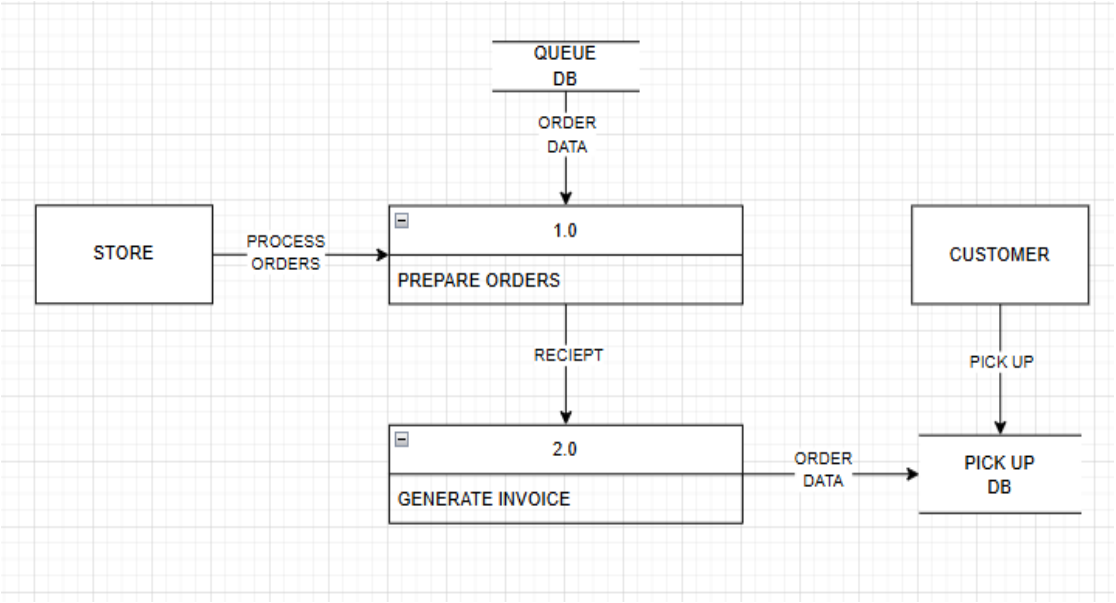

Figure 9: Store Order Management (DFD)

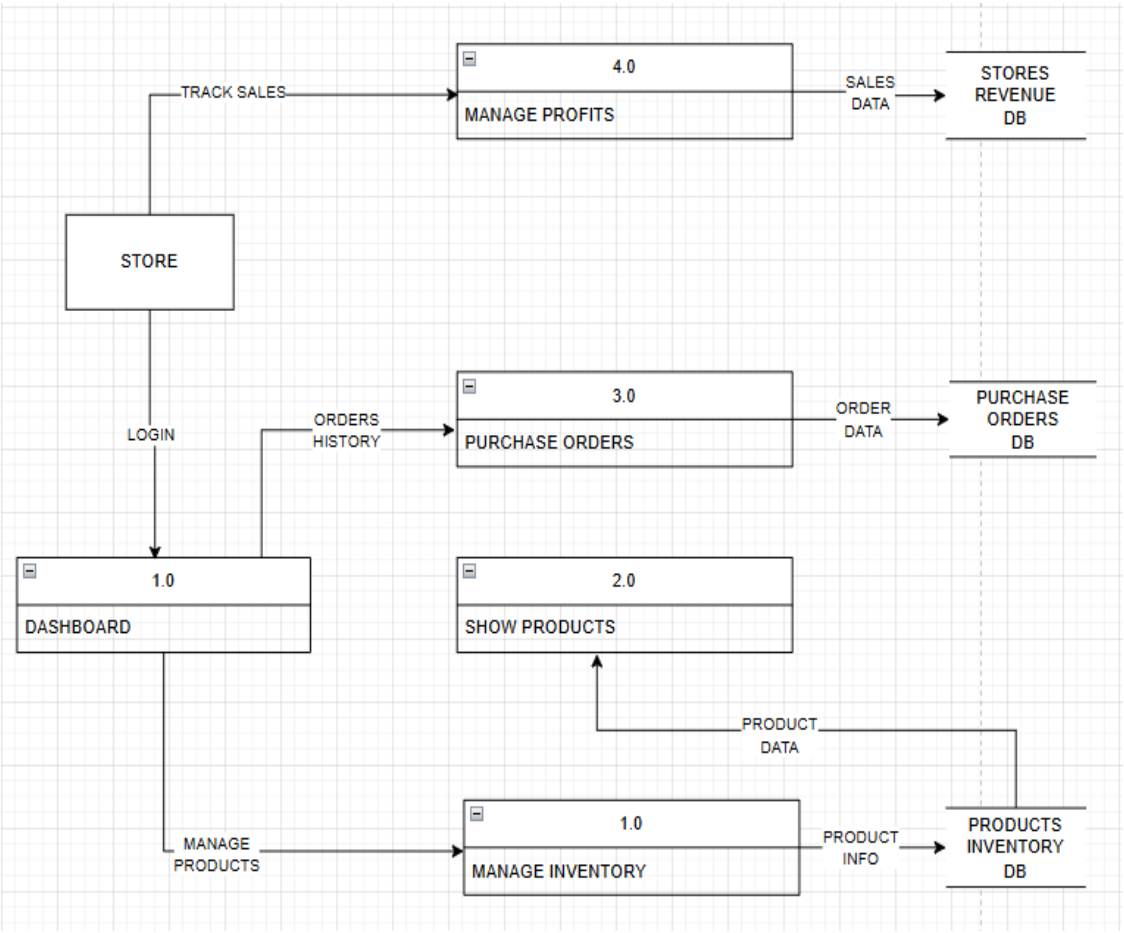

Figure 10: Store Dashboard (DFD)

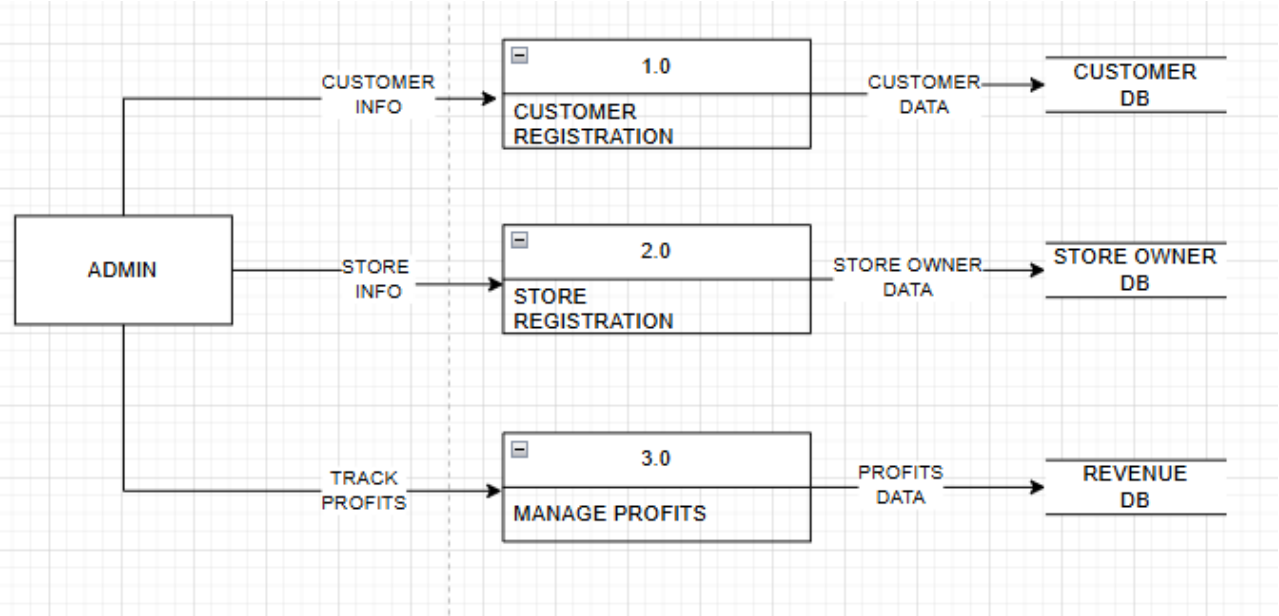

Figure 11: Admin Dashboard (DFD)

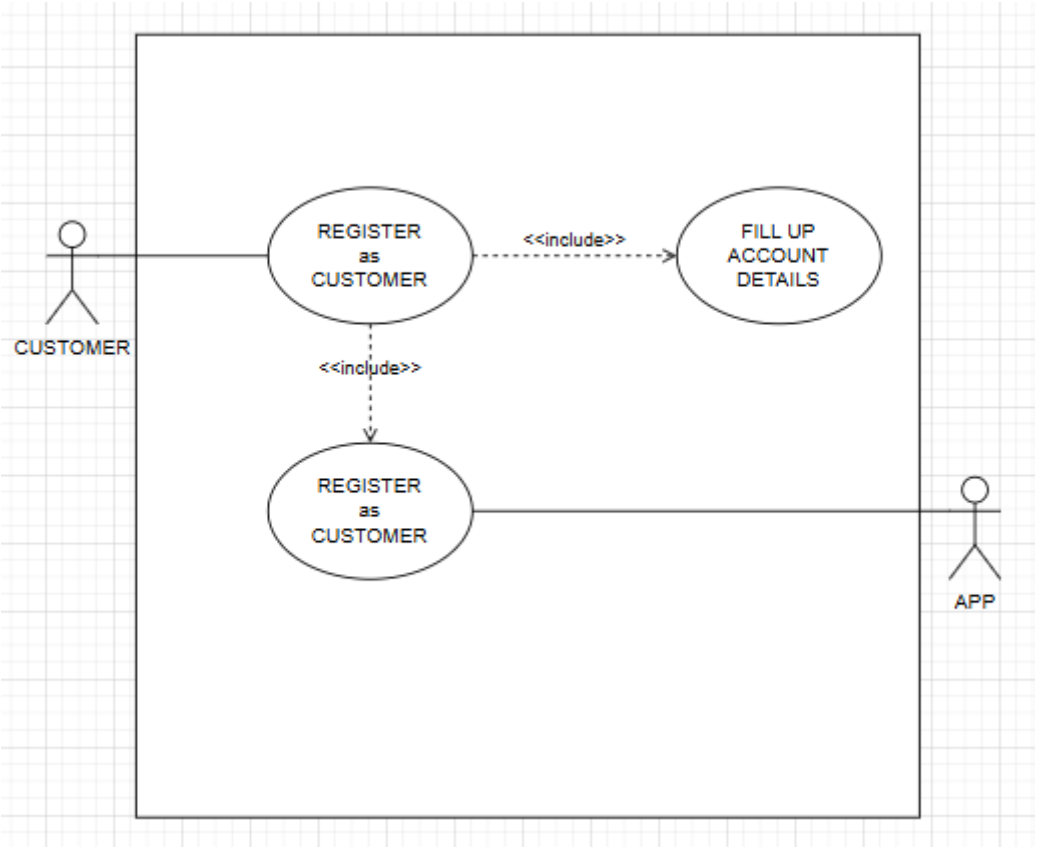

Figure 12: Customer Registration (Use-Case)

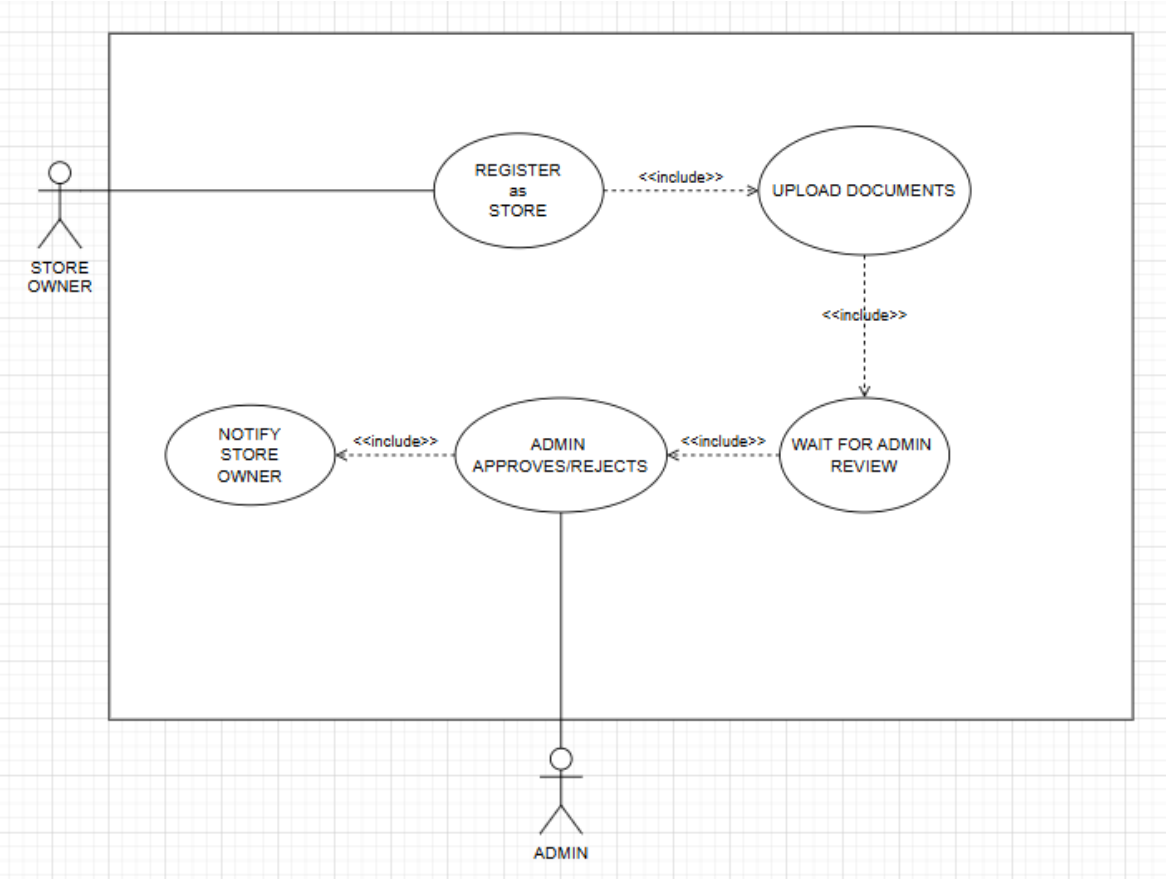

Figure 13: Store Registration (Use-Case)

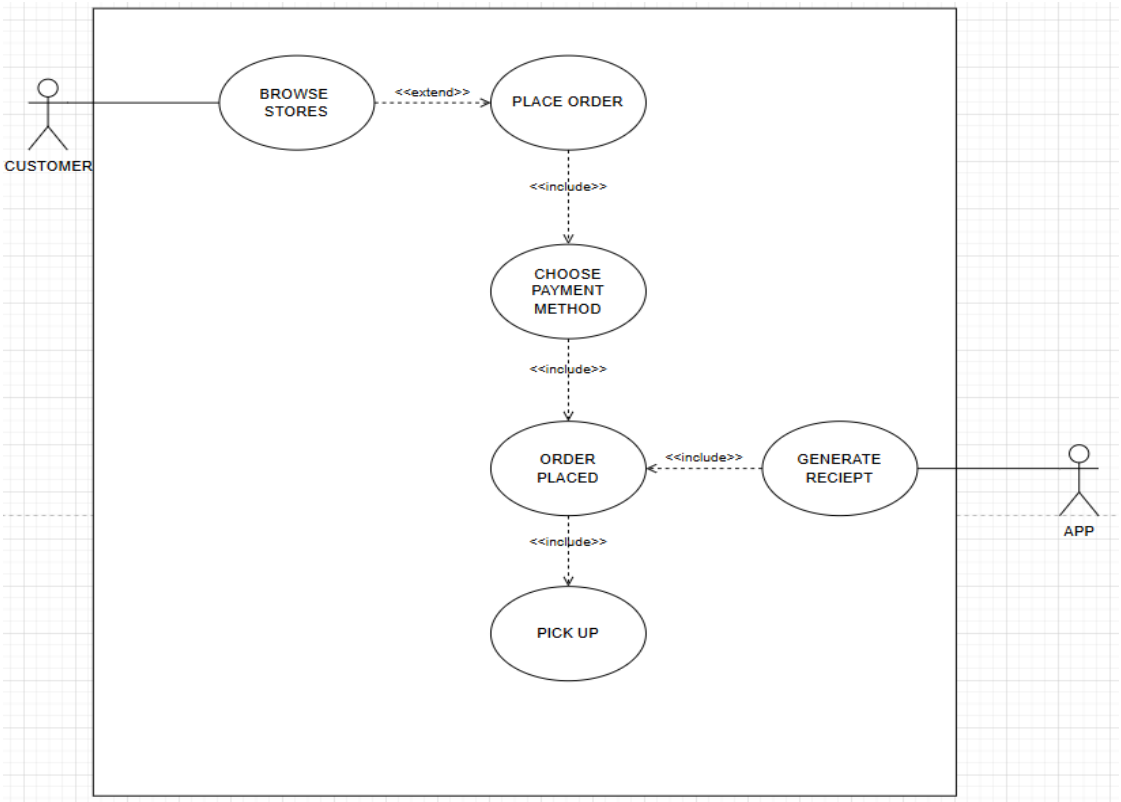

Figure 14: Customer Ordering (Use-Case)

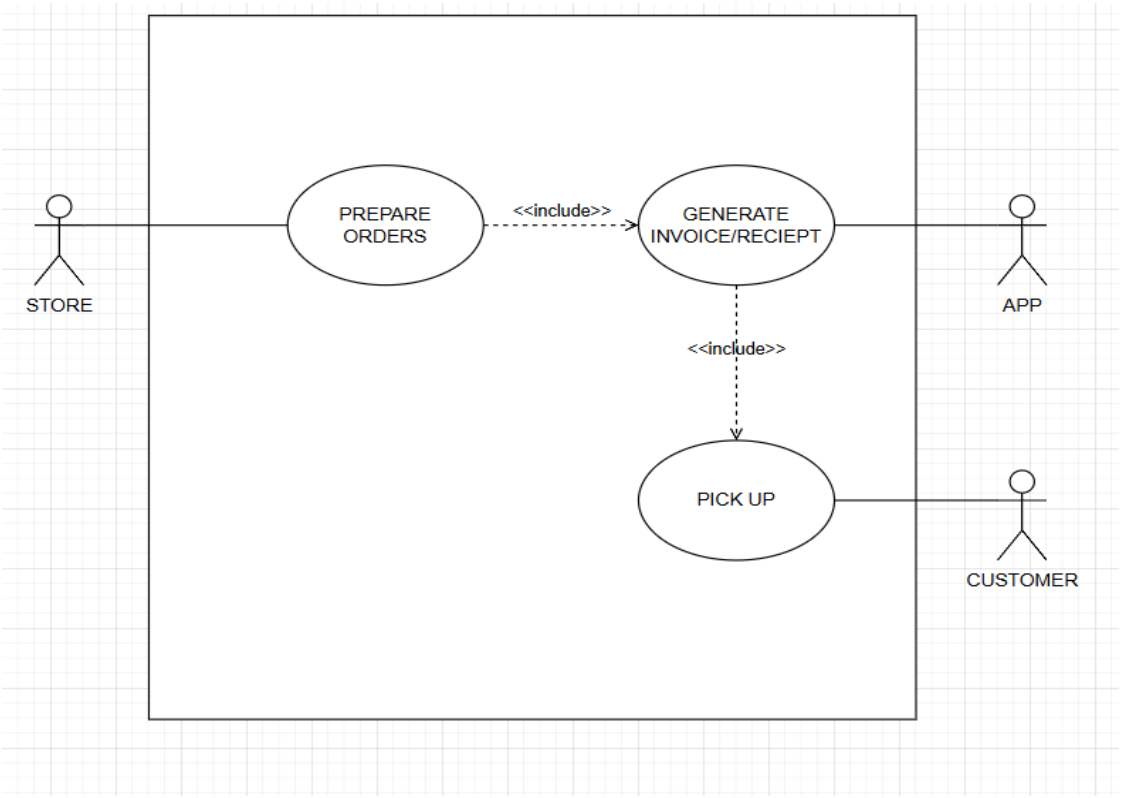

Figure 15: Store Order Management (Use-Case)

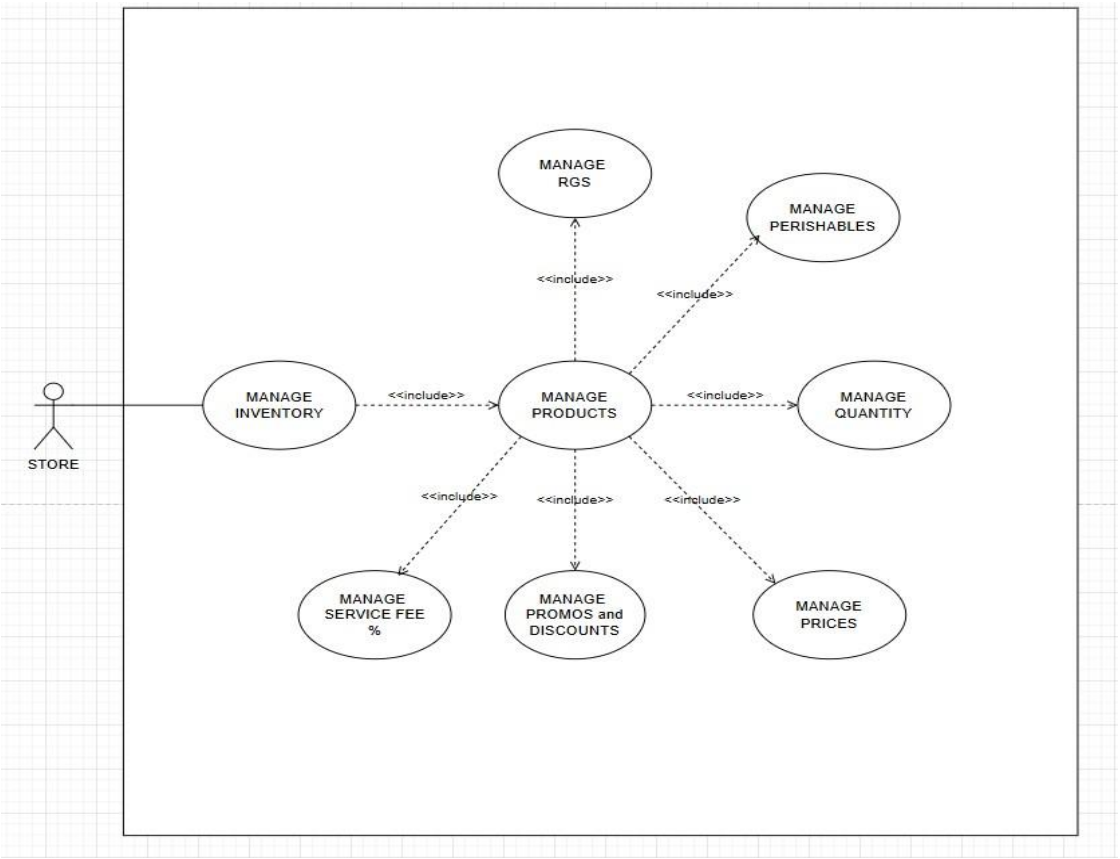

Figure 16: Store Inventory Management (Use-Case)

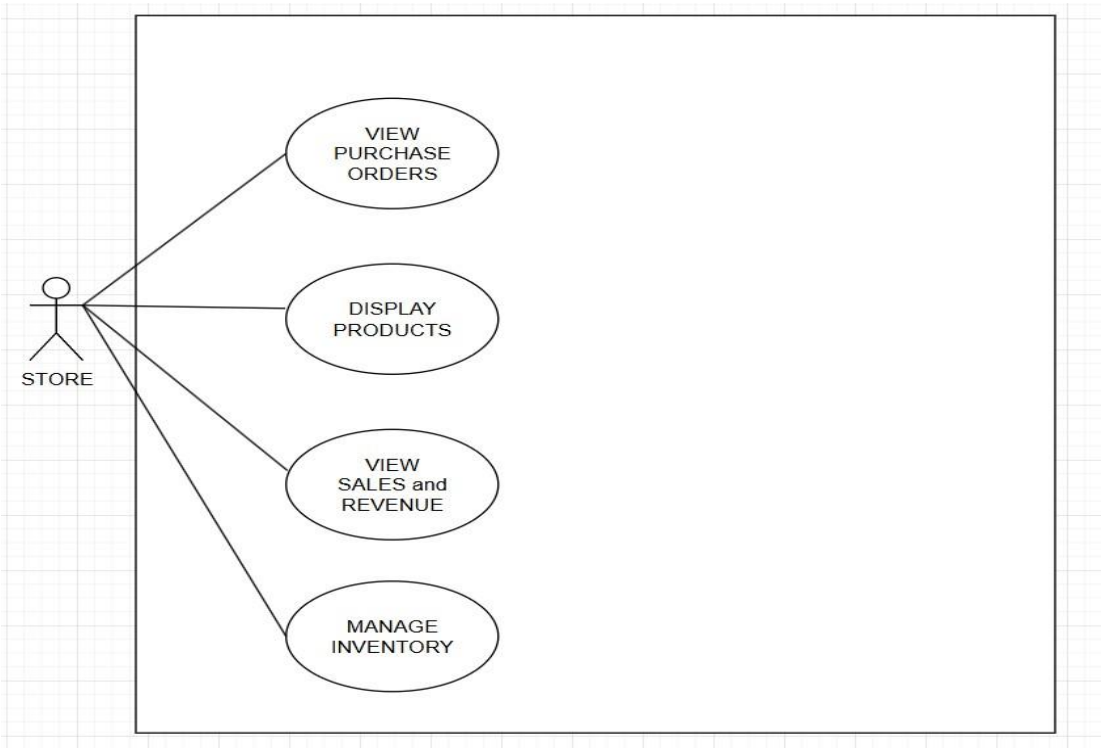

Figure 17: Store Dashboard (Use-Case)

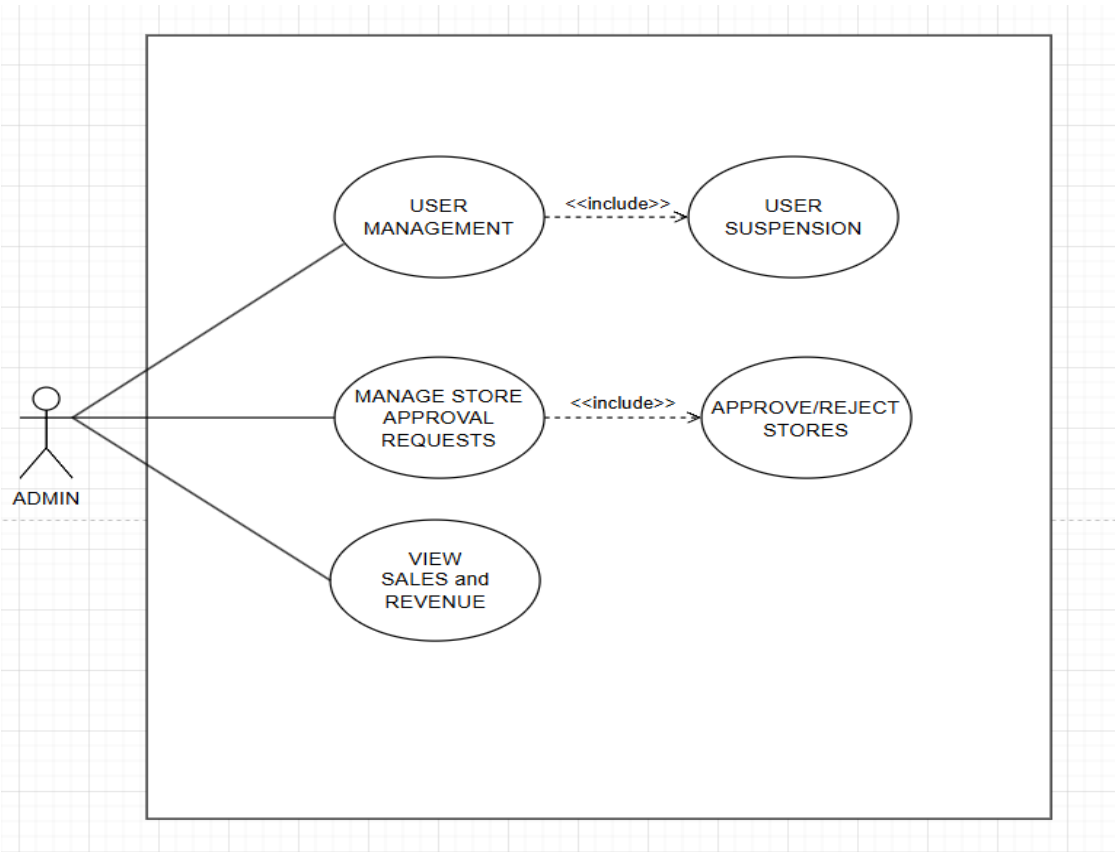

Figure 18: Admin Dashboard (Use-Case)

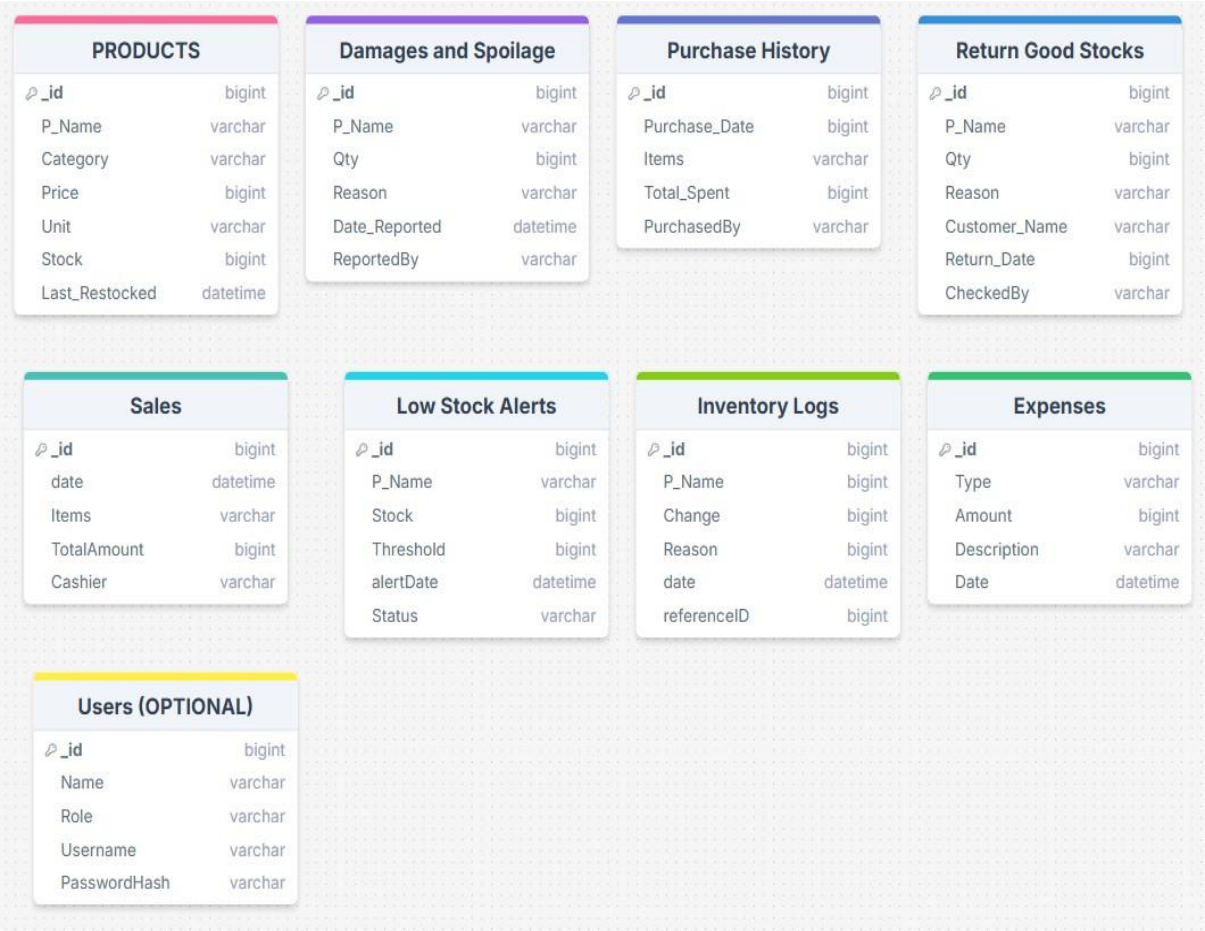

Figure 19: Database (Initial Database Design)

APPENDIX B. RESOURCE PERSONS

Dominic R. Bantigue
Capstone Project Adviser

Jessiel Chris Hilot
Capstone Project Coordinator