Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report

on

"WINKEL: MultiVendor Shopping Platform"

[COMP 313]

(For partial fulfillment of 3rd Year/ 2nd Semester in Computer Science)

Submitted by:

Puja Kadayat (Roll No. 18)
Nikki Kayastha (Roll No. 22)
Siddhant Khadka (Roll No.24)
Sophiya Khadka (Roll No.25)

Submitted to:

Mr. Suman Shrestha
Project Coordinator
Department of Computer Science and Engineering

Submission Date: 13th January 2025

Bona fide Certificate

This project work on

“.....WINKEL.....”

is the bona fide work of

“.....Puja Kadayat, Nikki Kayastha, Siddhant Khadka, Sophiya Khadka.....”

who carried out the project work under my supervision.

Project Supervisor

_____

Dr. Sushil Shrestha
Associate Professor, Lead Researcher Digital Learning Research Lab
Department of Computer Science and Engineering

Date: 13th January 2025

Abstract

A multi-vendor app is a robust software solution designed to facilitate the seamless management of multiple vendors and their products within a single platform. It offers a range of features that simplify the complex process of handling vendor registrations, product listings, order processing, payments, and communication between vendors and customers. By automating these administrative tasks, a multi-vendor app significantly reduces operational overhead for platform administrators while enabling vendors to focus on delivering quality products and services to customers. One of the key advantages is centralized data management, ensuring that all vendor and customer data, product details, transactions, and reviews are stored and accessed efficiently, improving accuracy and consistency across the platform.

Our multi-vendor app is a mobile-based application built using the Flutter framework. It aims to streamline the process of managing multiple vendors, product listings, and customer orders, while offering a unified and intuitive user experience. This project seeks to create a comprehensive platform that supports e-commerce businesses in managing vendors, orders, and transactions seamlessly. Our goal is to offer a scalable solution that empowers vendors to reach wider audiences, helps administrators manage operations more effectively, and enhances the overall shopping experience for customers.

Keywords: *multi-vendor app, mobile e-commerce systems, multi-vendor platforms, cross-platform development (flutter), Real-time Databases, Firebase cloud storage, vendor management in e-commerce*

Acknowledgement

We extend our heartfelt gratitude to all individuals who supported and guided us in completing this project. We are privileged to express our sincerest appreciation to our project supervisor, Dr. Sushil Shrestha, for his invaluable guidance, encouragement, feedback, and constructive criticism throughout the project's duration. We also wish to acknowledge the Department of Computer Science and Engineering (DoCSE) and the entire university for providing us with the opportunity to work on this project.

Sincerely,

Team Winkel

Table of Contents

List of Figures

List of Tables

Abbreviations

The list of all abbreviations used in the documentation is included here

GUI	Graphical User Interface
IDE	Integrated Development Environment
NoSQL	Non-Relational Structured Language
SDLC	Software Development Life Cycle
RAM	Random Access Memory
UX	User Experience
UI	User Interface

Chapter 1 Introduction

1.1 Background

A multi-vendor app is an online platform that allows multiple independent vendors or sellers to offer their products or services to customers in a single application. According to Jain and Jain (2021), these apps create a marketplace where different vendors can set up virtual stores, list their offerings, and manage transactions under a unified platform [1]. Popular examples include Amazon, eBay, and Etsy in the e-commerce space, as well as Uber Eats and Airbnb in service-based markets. Akwukwuma et al. (2024) note that the app's structure facilitates seamless interactions between customers and vendors, providing a diverse shopping experience within one app rather than requiring customers to visit multiple separate websites or stores [2].

Built using Flutter, multi-vendor apps serve as centralized platforms for overseeing vendor registrations, product listings, order processing, payments, and communication between vendors and customers (Edge & Miola, 2022) [3]. The platform automates tasks such as inventory management, order fulfillment, and customer feedback, while providing administrators with comprehensive vendor data for better decision-making. As online marketplaces grow in complexity, these apps ensure efficiency, transparency, and scalability for e-commerce platforms, making them essential tools for businesses seeking to optimize their operations and enhance the overall shopping experience (Akwukwuma et al., 2024) [2].

1.1.1 Problem statement

The lack of intuitive user guidance in existing multi vendor apps creates confusion for users, particularly in understanding how to log in as a vendor or customer and manage essential tasks like adding products, resulting in a suboptimal user experience and reduced platform engagement.

1.2 Objectives

- To build an app that can provide an effective online shopping experience for both sellers and customers.
- To implement communication tools within the app to allow smooth interactions between vendors and customers.
- To provide an intuitive interface for vendors to register, set up their stores, and manage their product listings with ease.
- To streamline the handling of orders and payments, ensuring secure transactions and efficient order fulfillment.

1.3 Motivation and Significance

Implementing a multi-vendor app addresses the inefficiencies of managing diverse marketplaces. Traditional systems often rely on fragmented processes, leading to operational challenges, inconsistent customer experiences, and increased administrative overhead. By centralizing vendor management, order processing, and customer interactions, a multi-vendor app streamlines operations, enhances efficiency, and supports scalability. Jain and Jain (2021) noted that fragmented e-commerce systems result in duplicated efforts and inconsistent data management, highlighting the need for centralized solutions [1]. Akwukwuma et al. (2024) emphasize that usability and intuitive design in e-commerce applications improve user engagement [2], while Edge and Miola (2022) underline the benefits of unified technological platforms like Flutter for building scalable, user-friendly apps [3].

A multi-vendor app creates a dynamic shopping environment with diverse product offerings, comprehensive listings, and personalized recommendations, simplifying decision-making and fostering trust among users. Integrated tools for inventory management, sales tracking, and performance analysis empower vendors to optimize operations and grow their businesses. The success of platforms like Daraz and

Hamrobazaar showcases how centralized systems enhance vendor performance and customer satisfaction. By leveraging advanced tools and technologies, multi-vendor apps address operational inefficiencies while creating a sustainable and engaging ecommerce ecosystem.

Chapter 2 Related Works

2.1 HamroBazar

Hamro Bazar is a popular online classifieds platform in Nepal, allowing users to buy and sell a wide range of new and second-hand items, including electronics, vehicles, real estate, and fashion. It's free to use and widely utilized by individuals and businesses for local trading.

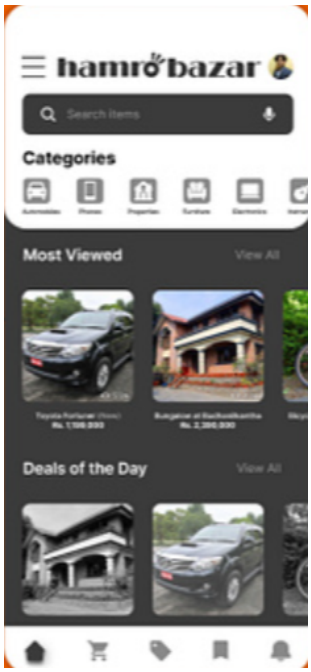


Figure 2.1 UI of HamroBazar Mobile App

It's particularly known for facilitating peer-to-peer transactions, allowing users to post ads for products they want to sell and connect with buyers directly [10]. The platform's simplicity and local focus make it widely used across the country for both new and second-hand goods.

- **Company Background:**

HamroBazar, founded in 2016, is one of the leading online classifieds platforms in Nepal, offering a convenient way for users to buy and sell a variety of new and second-hand goods. It is widely used across the country, enabling individuals and businesses to trade locally. The platform is particularly known for its user-friendly interface and focus on peer-to-peer transactions.

- **Business Model:**

Hamro Bazar operates a free-to-use platform where users can post ads for products they want to sell and directly connect with potential buyers. The platform focuses on local trading, covering categories such as electronics, vehicles, fashion, and real estate. Its revenue comes from premium services like featured ads and business accounts that offer more visibility for sellers.

- **Technology and Innovation:**

Hamro Bazar continually invests in technology to enhance the user experience, from improving the mobile app for better accessibility to optimizing the platform's search and categorization features. The platform also incorporates features like instant messaging to facilitate direct communication between buyers and sellers.

- **Customer-Centric Approach:**

Hamro Bazar prioritizes ease of use, allowing users to easily post ads, browse listings, and contact sellers. The platform’s focus on simplicity, combined with local community engagement, ensures that users find a reliable way to buy and sell both new and second-hand products. Customer feedback is valued and plays an essential role in refining the platform's features.

- **Challenges and Opportunities:**

The online classifieds sector faces challenges such as increasing competition, ensuring trust between buyers and sellers, and managing fraud. However, Hamro

Bazar is positioned to expand in emerging markets and adapt to the growing demand for digital platforms in Nepal. Its user-focused approach allows it to capitalize on evolving consumer behaviors and the shift towards online trading.

2.2 Daraz

Daraz is a leading e-commerce platform in Nepal, offering a wide range of products including electronics, fashion, home goods, and more [8]. Through its app, users can browse, shop, and enjoy services like secure payments, fast delivery, and regular discounts. It also supports sellers with its dedicated seller center.

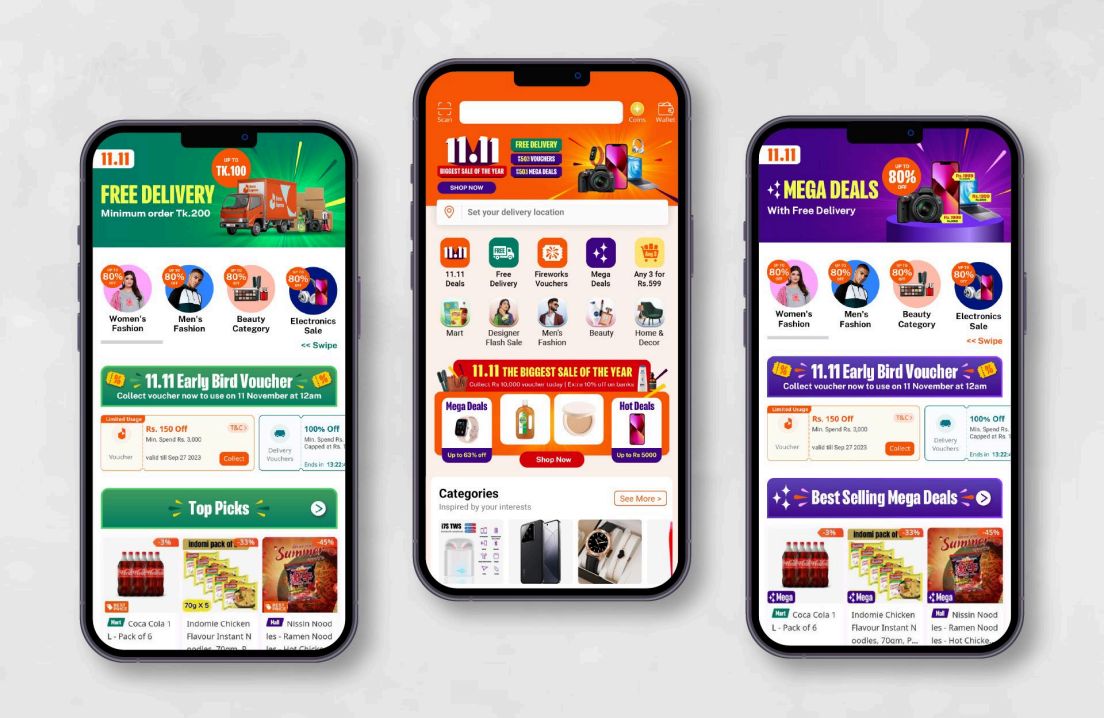


Figure 2.2 DarazUI

- **Company Background:**

Founded in 2012, Daraz is Nepal's leading e-commerce platform, offering a wide variety of products, including electronics, fashion, home goods, and more.

Headquartered in Kathmandu, Daraz is part of the Alibaba Group and serves millions of customers across Nepal and other South Asian markets.

- **Business Model:**

Daraz operates an online marketplace where consumers can purchase products directly from local and international sellers. The platform supports secure payment methods, quick deliveries, and frequent sales and discounts. It generates revenue through product sales, as well as from fees charged to sellers for utilizing the platform's services.

- **Technology and Innovation:**

Daraz utilizes advanced technologies such as data analytics and machine learning to provide personalized recommendations, optimize product listings, and enhance the shopping experience. The company has also integrated AI to improve customer service and streamline logistics for faster delivery times.

- **Customer-Centric Approach:**

Daraz's commitment to customer satisfaction is reflected in its seamless shopping experience, secure payment options, and comprehensive customer support. Regular promotions, discounts, and a broad selection of products ensure that customers can find everything they need at competitive prices.

- **Challenges and Opportunities:**

The e-commerce sector in Nepal faces challenges such as logistics inefficiencies, regulatory hurdles, and competition from both local and international platforms. However, Daraz continues to grow by capitalizing on Nepal's expanding internet penetration, leveraging technology to streamline operations, and meeting the evolving demands of consumers.

2.3 Jeeve

Jeevee is a healthcare app in Nepal that connects users with medical professionals, enabling easy access to online doctor consultations, pharmacy services, and diagnostic lab bookings [9]. It also offers features for maintaining health records and receiving medical advice, making healthcare more accessible and convenient.

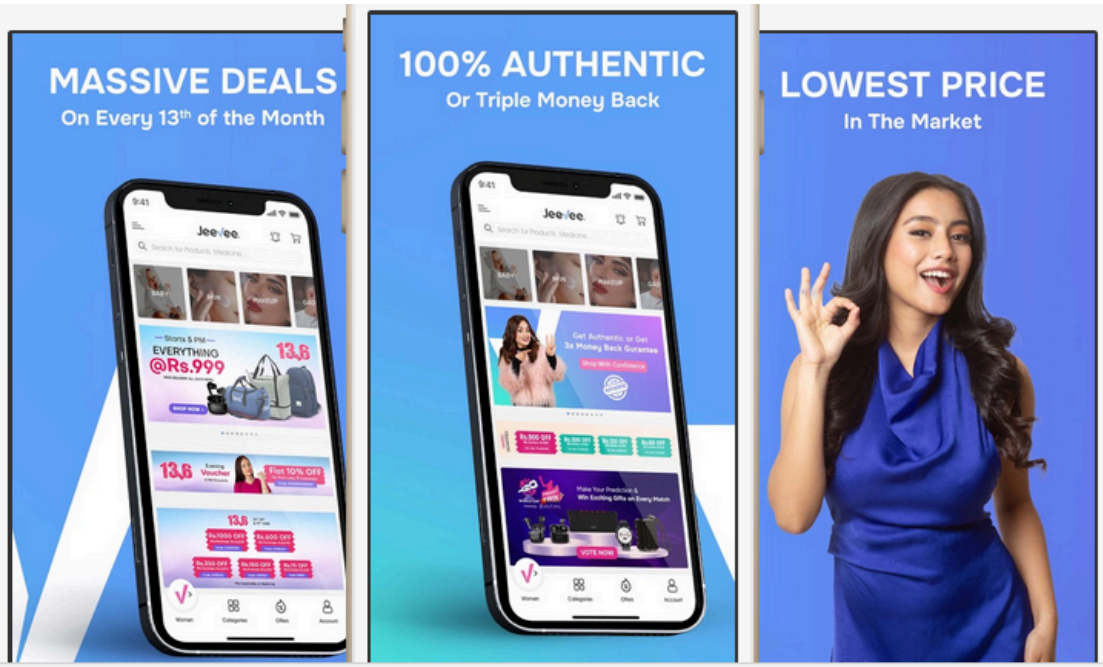


Figure 2.3 Jeeve UI

- **Company Background:**

Launched in 2018, Jeevee is a healthcare app designed to connect users with medical professionals in Nepal. The platform provides access to online doctor consultations, pharmacy services, diagnostic lab bookings, and health record management, making healthcare more accessible to a broader audience.

- **Business Model:**

Jeevee operates as an online healthcare service provider, offering users the ability to book medical consultations, order medicines, and manage health records through its app. Revenue is generated through service fees for consultations, pharmacy orders, and diagnostic bookings.

- **Technology and Innovation:**

Jeevee leverages cutting-edge telemedicine technology, integrating video consultations, appointment scheduling, and secure health records storage. The platform uses data analytics to provide personalized health recommendations and treatment options based on users' health history.

- **Customer-Centric Approach:**

Jeevee is focused on improving healthcare access for users across Nepal, particularly in remote areas where medical services may be limited. The app's intuitive design, wide network of doctors, and integration of health records ensure that users receive timely and effective care. Customer feedback is essential to improving the platform's usability and service offerings.

- **Challenges and Opportunities:**

Jeevee faces challenges in a market where healthcare infrastructure is still developing, and telemedicine regulations are evolving. However, it has the opportunity to expand by increasing awareness of its services, building partnerships with healthcare providers, and tapping into the growing demand for digital health solutions in Nepal.

2.4 Limitations of related works

An analysis of major platforms in Nepal reveals significant limitations. Hamro Bazar lacks secure payment gateways, user verification, and advanced features, restricting functionality and revenue opportunities. Daraz faces challenges with high shipping costs, lengthy delivery times, and a complex vendor management system, which deter

local vendors and affect customer experience. Jeevee, while innovative in healthcare, is limited by its specialized focus, scalability issues, and strict regulatory requirements. These shortcomings highlight the need for a multi-vendor platform that ensures security, streamlined operations, comprehensive vendor support, and a user-friendly experience across diverse business categories.

Chapter 3 Design and Implementation

3.1 Design and System Architecture

In managing the development lifecycle of the multi-vendor app, we adopted the Agile Software Development Lifecycle with a focus on the Scrum framework. Agile was chosen for its adaptability to changing requirements and its iterative approach, ensuring continuous delivery and improvement (Jain & Jain, 2021) [1]. Scrum provided structure through defined roles, regular meetings, and clear deliverables, fostering collaboration and maintaining focus on the project's goals. Furthermore, tools like Flutter were utilized to build cross-platform user interfaces efficiently, leveraging its ability to deliver a single code base for multiple platforms (Edge & Miola, 2022; Flutter Documentation, n.d.; Wikipedia, n.d.) [3] [7] [5].

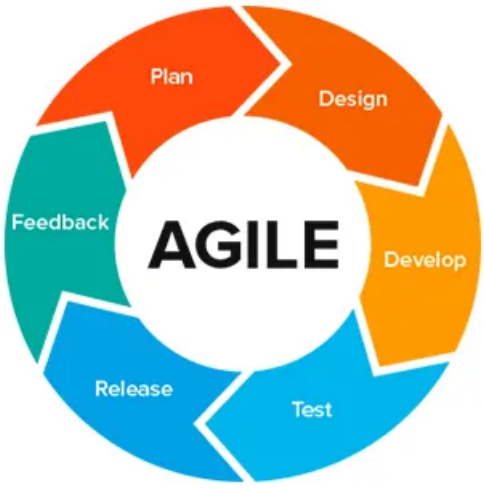


Figure 3.1 Agile SDLC

The Agile development lifecycle, as shown in the diagram, guided development process:

- The first two weeks focused on identifying user needs, creating user stories, and prioritizing a product backlog for the MVP, emphasizing vendor registration, product listing, and order management.
- Foundational modules like authentication and user roles were implemented, followed by vendor features, customer-facing functionalities, and admin tools for vendor approvals and analytics.

The Agile approach ensured flexibility, while Scrum provided structure through sprints, planning, and retrospectives.

3.1.1 System Design

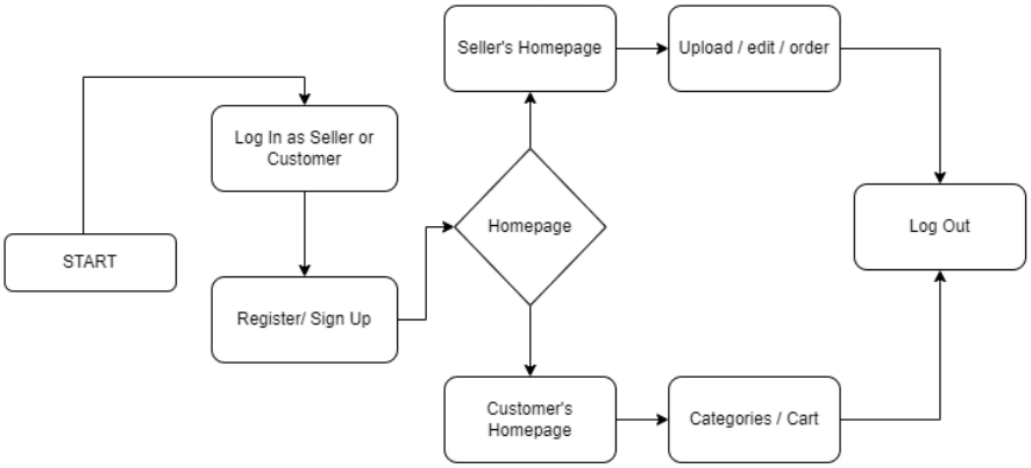


Figure 3.1.1A System Flowchart

The flowchart represents the navigation flow of Winkel. It starts with a login or sign-up page where users can register or log in as either a seller or a customer. After authentication, users are redirected to the homepage, which acts as a central decision point. Depending on their role, sellers are taken to the seller's homepage, where they can upload, edit, or manage their orders. Customers are directed to the customer's homepage, where they can explore categories, add products to their cart, or proceed to purchase. The application includes a logout option to terminate the session, ensuring secure user management.

Figure 3.1.1B System Interaction

This interaction diagram shows the relationship between customers, vendors, and admins, where customers purchase products, vendors list products for sale, and admins approve or manage the platform's content.

3.1.2 Use Case Diagram:

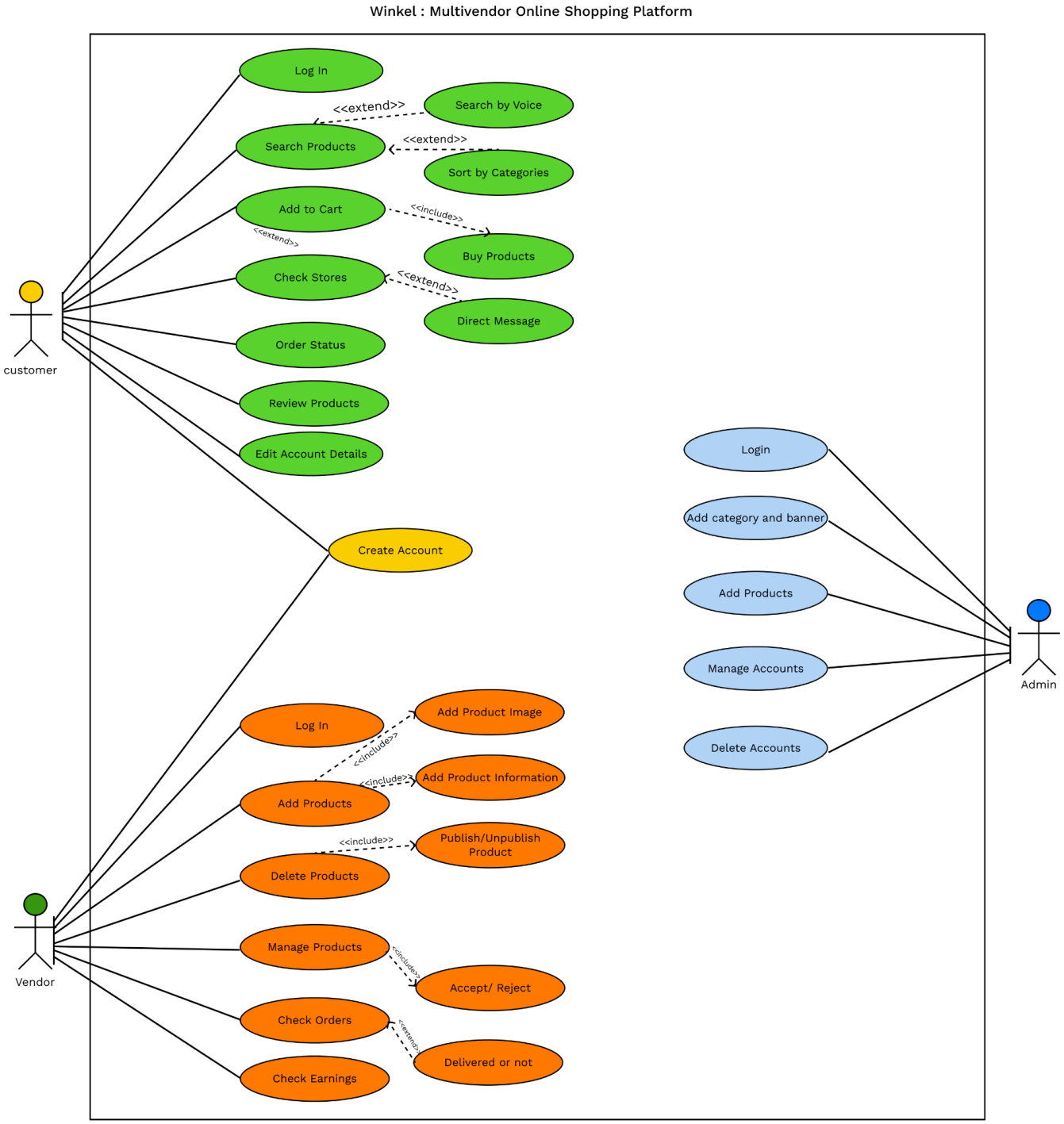


Figure 3.1.2 Use Case Diagram

This figure highlights the functionalities available to users, including vendors, customers, and admins. Vendors can manage products, view orders, and add shipping details, while customers can search, buy, and manage their accounts. Admins oversee account management and overall platform activities..

3.1.3 State Diagram:

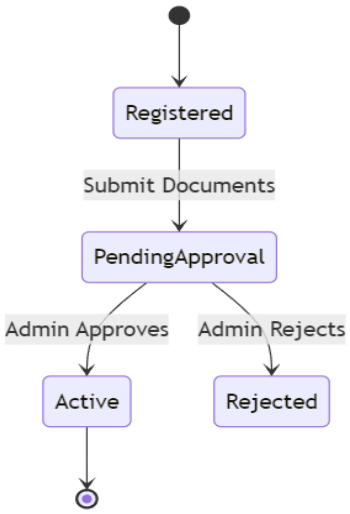


Figure 3.1.3A Vendor Registration

This figure explains the process of registering a vendor account. Vendors need to register their business with respective photos and documents, which is accepted or rejected by the Admin.

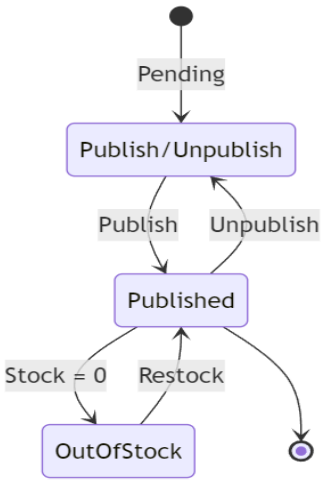


Figure 3.1.3B Product Life Cycle

The image illustrated above shows the product life cycle. Once a vendor uploads the product, they can either publish or unpublish them.

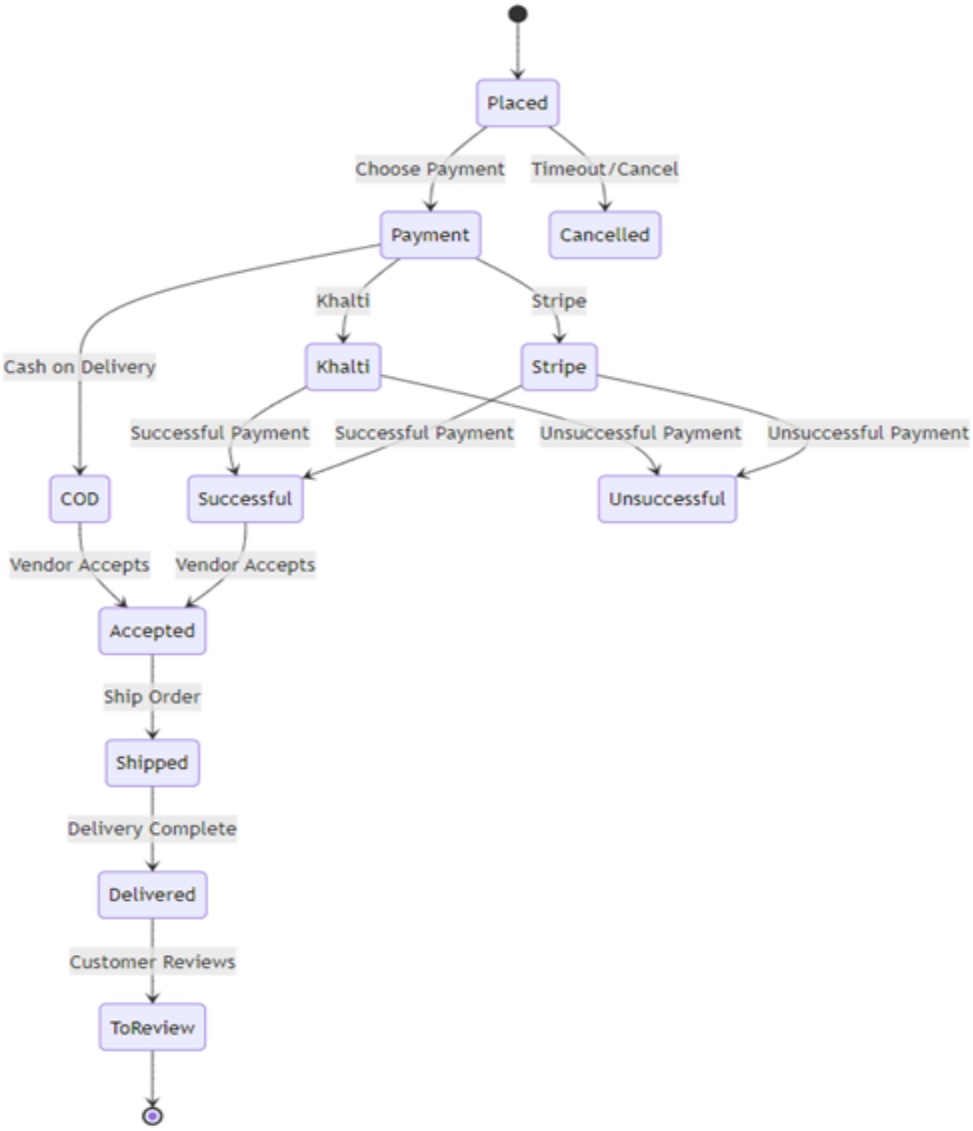


Figure 3.1.3COrder Placing Process

Placing an order is a bit of a long process as shown in the figure above. Once the customer places an order, they need to select a payment method (COD, Stripe, or Khalti). After successful payment, the order is placed and the vendor accepts the order and the order is shipped. After delivery, the items are up for review by the customer.

3.1.4 Sequence Diagram:

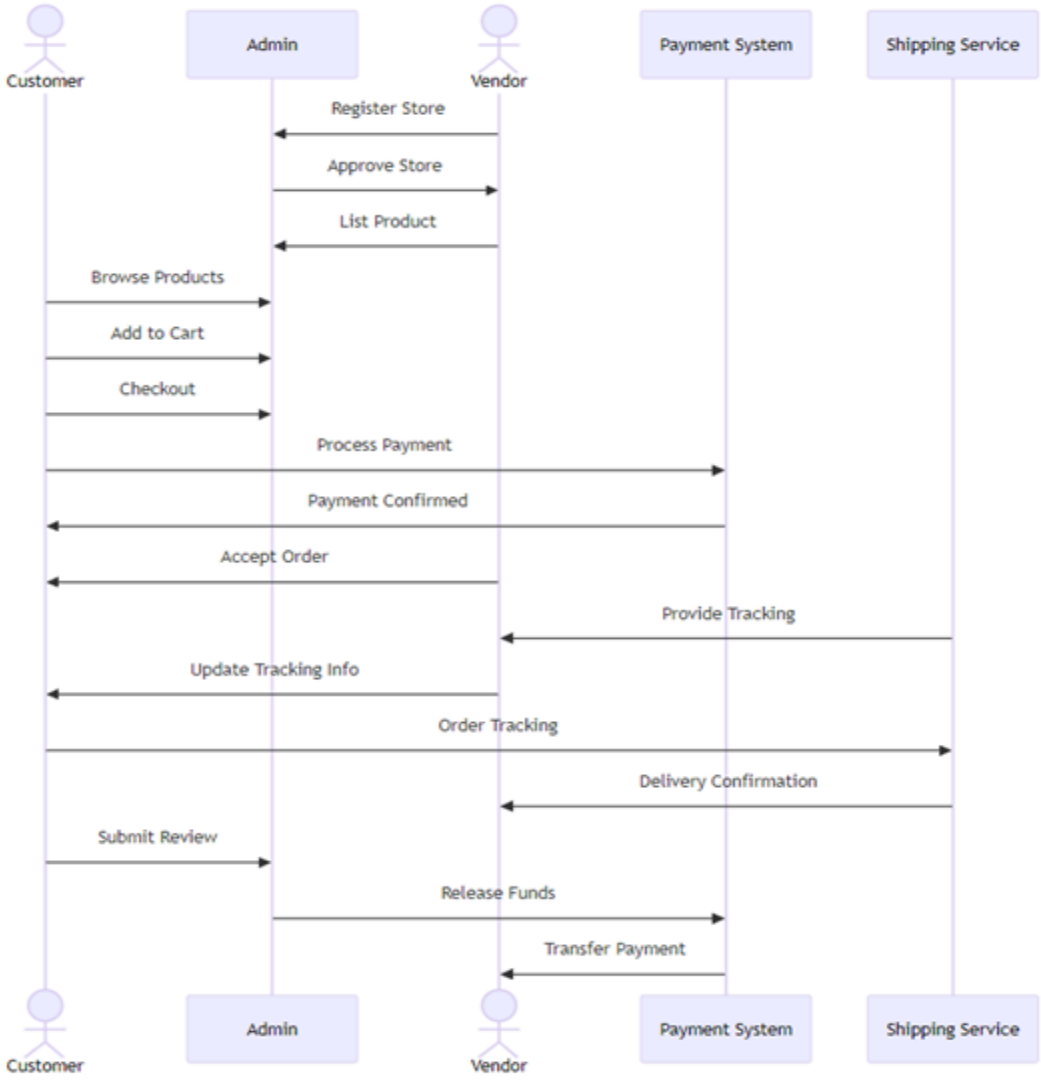


Figure 3.1.4 Sequence diagram of Winkel

The sequence diagram outlines the interactions among five entities: Customer, Admin, Vendor, Payment System, and Shipping Service. The process starts with the vendor registering a store, which the admin approves, allowing the vendor to list products. Customers browse products, add items to their cart, and proceed to

checkout. Payment is processed through the payment system, and upon confirmation, the vendor accepts the order. The vendor updates tracking information after providing the shipping service with tracking details, enabling order tracking by the customer. Once the shipping service confirms delivery, the customer can submit a review. The admin then releases funds, and the payment system transfers the payment to the vendor, completing the transaction. The diagram captures the sequential flow of actions and communication between all participants in the e-commerce process.

3.2 Development Process

3.2.1 Survey:

To gather insights for improving the user experience of our multi-vendor platform, we conducted a survey targeting users of popular e-commerce platforms like Daraz, HamroBazaar, and Jeeve. The questions were designed based on research studies and academic papers. Jain and Jain (2021) emphasized that usability challenges in e-commerce platforms, such as complex navigation and poor search functionality, reduce user engagement and satisfaction [1]. Similarly, Akwukwuma et al. (2024) highlighted the importance of intuitive designs and streamlined processes for improving user retention [2].

Figure 3.2.1 Survey Word Cloud

Key Findings:

- **Complex Navigation**: Many users found the interfaces cluttered and difficult to navigate.
- **Search Inefficiencies**: Users reported inaccurate or irrelevant search results, aligning with common challenges noted in usability studies.
- **Loading Speed**: Mobile performance issues were noted; optimization is a priority.
- **Checkout Frustrations**: A significant number of participants cited the need for fewer steps and faster processing during checkout.
- **Limited Customer Support**: Mixed feedback on support features; live chat will be added for better accessibility.

3.2.2 FrontEnd:

The entire frontend was designed using **Dart** and **Flutter.** Dart and Flutter are ideal for building multi-vendor apps due to their cross-platform capabilities, enabling a single codebase for both Android and iOS with near-native performance. Flutter's customizable widgets and efficient state management allow for a rich UI/UX, while Dart ensures fast development and scalability. This combination reduces costs, speeds up time-to-market, and provides future-proof flexibility.

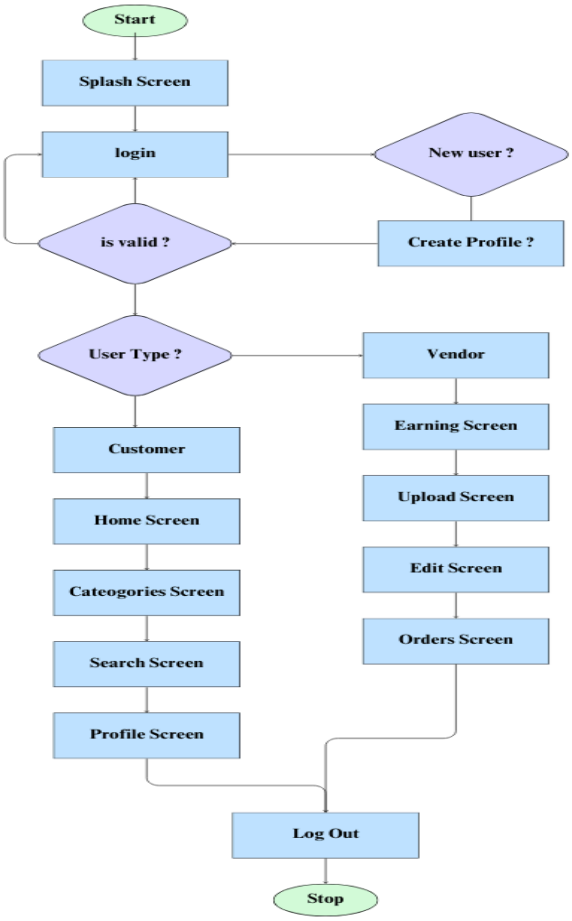


Figure 3.2.2 System Overview

Figure describes that the app starts with a splash screen and then is redirected to a login screen. New Users can create a profile, while existing users are validated. After login, users are split into two types as Customers and Vendors. Customers can access their home, category, search, profile and store screens whereas the vendors can access their earnings, upload, edit, orders and profile screen. As the figure indicates, both the users can log out. All these pages are connected through the use of the bottom navigation bar.

3.2.3 BackEnd:

For this project, we decided to use Firebase as it provides real-time database synchronization, making it ideal for features like live product updates, order tracking, and notifications. Its serverless architecture simplifies backend management, while services like Firebase Authentication, Cloud Firestore, and Cloud Functions handle user management and provide scalable data storage. So, in this project, Firebase functions both as a database and backend.

3.2.3.1 ER Diagram:

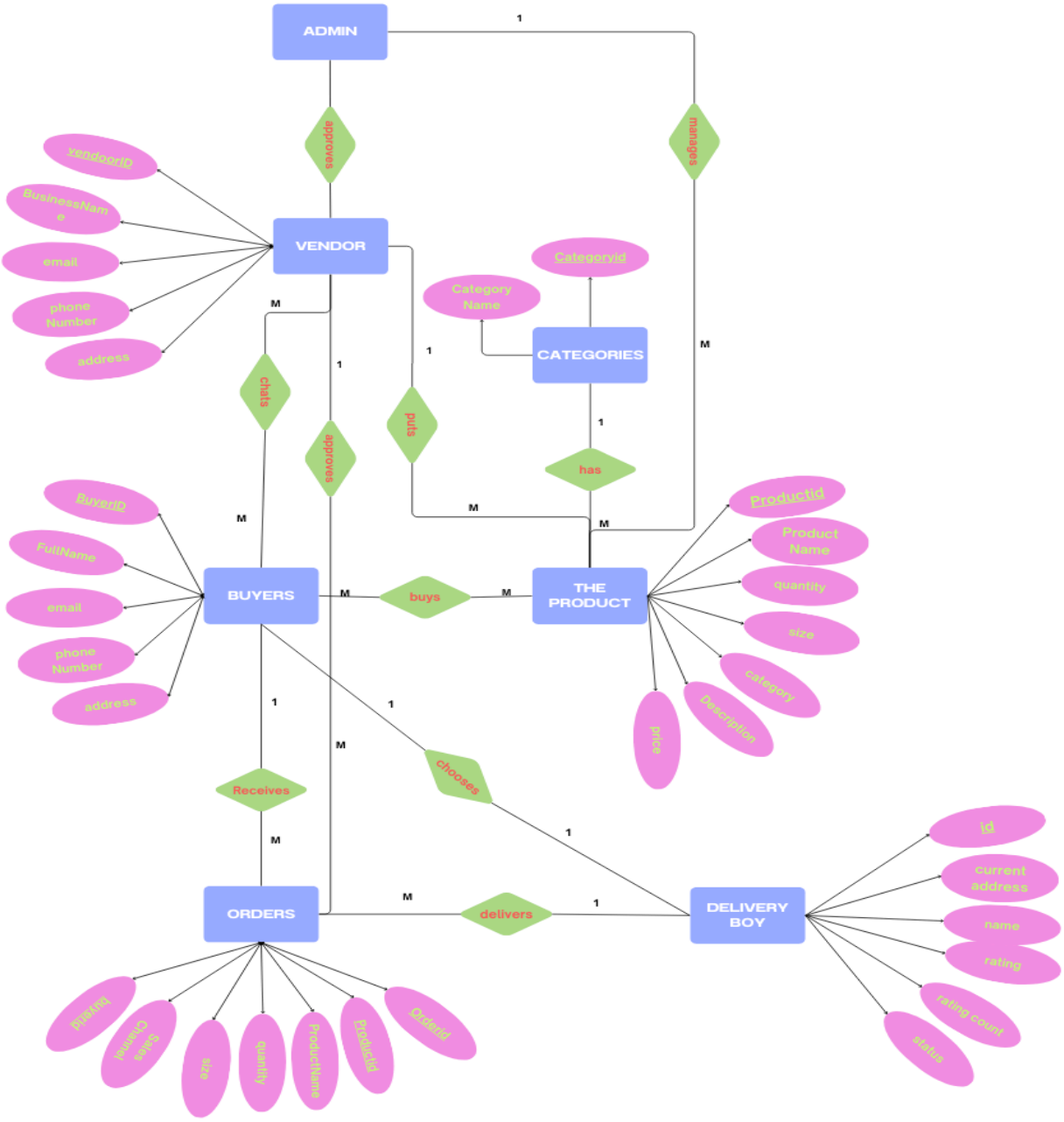


Figure 3.2.3.1 ER Diagram

This Entity-Relationship (ER) diagram highlights the relationships between entities such as Admin, Vendors, Categories, Products, Buyers, Orders, and Delivery Boys.

Key relationships include vendors creating categories and products, buyers purchasing products and placing orders, and delivery personnel delivering those orders. Attributes for each entity are detailed, such as product details, user information, and order specifics, to define their roles within the system.

3.2.3.2 Schema Diagram

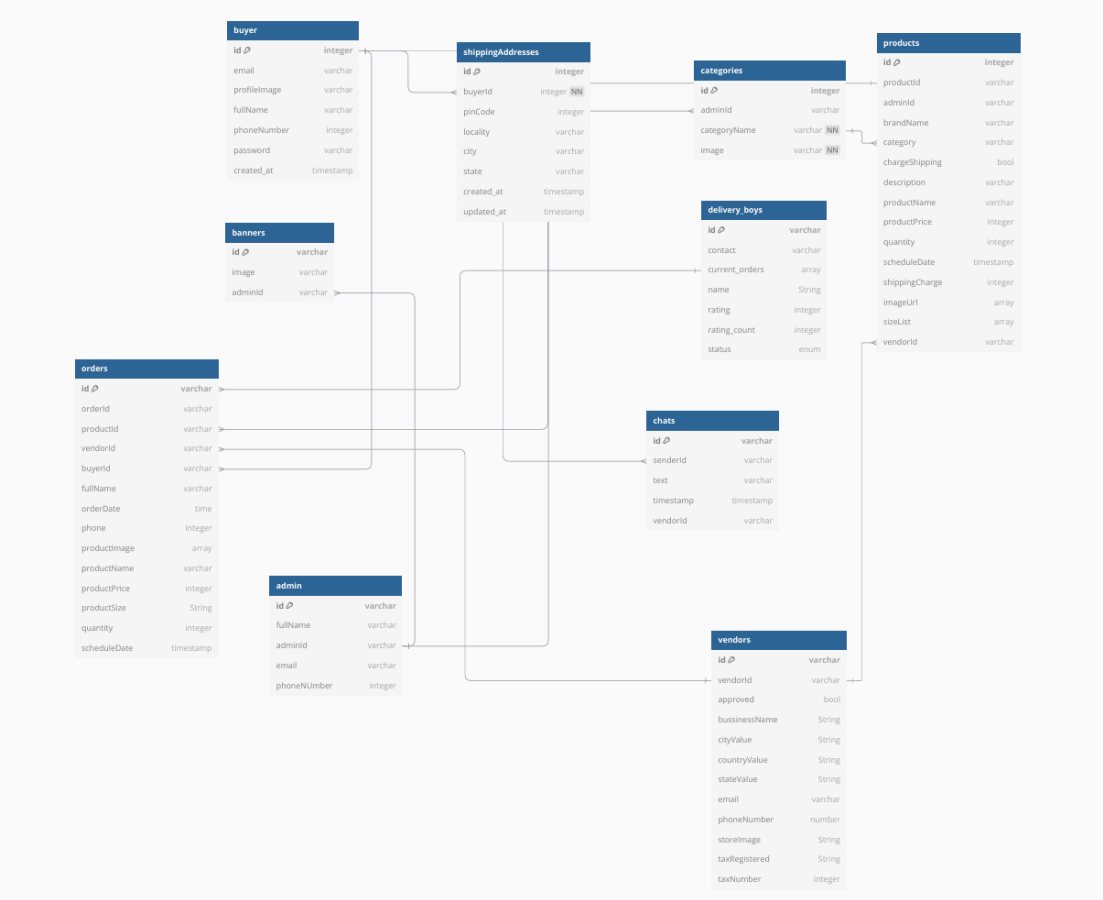


Figure 3.2.3.2 Schema Diagram

This diagram represents a database schema with key entities including buyers, vendors, products, orders, and admins, each containing specific attributes like personal details, product information, and order data. Relationships link these entities, such as buyers placing orders, vendors managing products, and admins overseeing the

platform. The design ensures efficient handling of user interactions, product inventory, and order tracking for seamless e-commerce operations.

3.3 System Requirement Specifications

Software specifications and hardware specifications:

3.3.1 Software Specifications

- **Operating System:** The system should be compatible with popular operating systems like Windows, macOS, and Linux.
- **Front End Tools:** Flutter is an open-source UI software development toolkit created by Google (Wikipedia, n.d.) [5]. It is used to build natively compiled applications for mobile, web, and desktop from a single codebase. Overall, Flutter is a powerful and flexible framework that simplifies the process of building cross-platform applications, offering a balance between performance and productivity for developers.
- **Backend and Database:** Firebase is a comprehensive platform for developing and managing mobile and web applications (Torunoğlu, n.d.) [4]. Acquired by Google in 2014, Firebase provides a suite of tools and services that facilitate various aspects of app development, including backend services, real-time databases, authentication, hosting, cloud functions, and more.
- **IDE:** VS Code is a free, open-source code editor and is useful and popular due its cross-platform nature, intuitive interface, language support, extensions, and integrated terminal.
- **Android Emulator:** An Android emulator simulates an Android device for app testing and debugging.
- **Git:** Version Control System for Collaborative Development.
- **GitHub:** Online Platform for hosting and sharing code repositories.

3.3.2 Hardware Specifications

The hardware requirements for the Winkel system are as follows:

- **Processor:** An Intel Core i3 or equivalent processor is required to ensure smooth functioning of the application.
- **RAM:** A minimum of 2GB of RAM is necessary to handle the application's operations and ensure decent performance.
- **Storage:** Adequate storage space is needed to accommodate the application code, databases, and associated files.
- **Internet Connection:** A stable internet connection is essential for accessing the system and maintaining seamless communication.

Chapter 4 Discussion on the achievements

Throughout our collaboration on this project, we had an outstanding experience working as a team, facing numerous challenges that provided valuable opportunities for learning, exploration, and growth. With effective coordination, thorough research, and dedication, we successfully tackled most of the obstacles encountered. This journey also allowed us to deepen our knowledge and skills in mobile application development, and by focusing on our objectives, we were able to implement several key features in the system.

4.1 Features and Functionality:

A. **Simplified Local Marketplace Approach:** The platform is designed to cater to the needs of local businesses, emphasizing ease of use and accessibility. By simplifying processes like product listing, order fulfillment, and payment handling, the platform lowers barriers for smaller vendors to join and thrive. This local-first approach fosters community-driven commerce and encourages local economic growth.

B. **Clear Separation of Admin, Vendor, and Buyer Roles:** The platform distinctly categorizes user roles into admin, vendor, and buyer, with tailored dashboards and functionalities for each. Admins have complete oversight of the system, vendors can manage their listings and performance, and buyers enjoy a user-friendly interface for browsing and purchasing products. This clear delineation ensures streamlined operations and minimizes confusion or overlaps.

C. **Structured Vendor Verification and Management System:** Our platform implements a robust vendor verification process to ensure that only authentic and reliable sellers are onboarded. This system includes identity checks, only authenticated users are registered and accepted as vendors.

4.2 Comparative analysis with related works

1. Multi-Vendor and Customer Model:

- Winkel: Stands out by allowing users to log in or sign up as both vendors and customers in the same app, streamlining access for users who want to sell and buy simultaneously.
- Daraz: Focuses on a clear distinction between vendors and customers, requiring separate portals.
- Jeevee: Primarily customer-focused, with limited vendor flexibility.
- HamroBazar: Simplistic, allowing anyone to list items, but lacks structured vendor features.

2. Communication and Delivery Integration

- Winkel: Features in-app chat and messaging between buyers and vendors, along with delivery personnel management for seamless logistics.
- Daraz: Offers strong logistics and order tracking but lacks direct buyer-vendor communication.
- Jeevee: Prioritizes delivery for time-sensitive health products without buyer-vendor messaging.
- HamroBazar: No delivery or communication tools, users need to arrange these externally which can be a hassle.

3. Flexibility and Target Audience

- Winkel: Combines a general-purpose e-commerce platform with vendor independence, appealing to users looking for versatility.
- Daraz: Targets retail-focused buyers with deals and promotions.
- Jeevee: Tailored for healthcare and wellness users.
- HamroBazar: Caters to casual buyers and sellers in a peer-to-peer marketplace without structured e-commerce features.

Chapter 5 Conclusion and Recommendation

Working on Winkel has been a challenging yet rewarding journey, allowing us to gain valuable experience in designing and developing a functional mobile application. We successfully achieved our primary goal of creating a user-friendly app that enables users to act as both customers and vendors. The app provides a seamless shopping experience by offering features such as product browsing, order management, and vendor profile handling. While we faced challenges in implementing a secure payment system due to the complexity of real-time transactions and limited resources for third-party integrations, the core functionalities were effectively developed, demonstrating the app's potential as a multi-vendor platform.

To elevate Winkel into a more comprehensive e-commerce solution, we recommend prioritizing the integration of a secure and efficient payment gateway in future iterations. Collaborating with experienced payment service providers and expanding access to advanced APIs will help overcome technical limitations. Additionally, enhancing support for third-party integrations and incorporating more robust features, such as analytics for vendors and advanced search options for users, will improve the overall user experience. By addressing these areas, Winkel can evolve into a reliable and widely-used platform for multi-vendor e-commerce.

5.1 Limitations

Despite the tremendous effort we've invested in bringing this project to fruition within a tight timeframe, we recognize that due to time constraints, limited knowledge, and resource constraints, there is still ample opportunity for improvement in our work, ranging from minor to significant enhancements.

1. **Dependence on Internet Connectivity:** Since the app relies on real-time data synchronization with Firebase, users need a stable internet connection for the

app to function optimally. Poor connectivity can result in slow loading times or transaction failures.

2. **App Size and Performance:** As the app grows with more features, products, and vendors, the app size may increase, potentially leading to longer download times and reduced performance, particularly on devices with lower storage or processing power.
3. **Limited Payment Options:** Depending on the region, the app may only support a limited number of payment methods, restricting users' choices and potentially causing inconvenience for those who prefer alternative payment methods.
4. **Limited Customer Support Features:** Without a dedicated customer support system , users may find it difficult to resolve issues quickly, leading to frustration and negative experiences.
5. **Vendor Challenges:** Managing large inventories can be difficult without advanced tools for bulk uploads or tracking, and limited storefront customization options may hinder vendors from showcasing their products effectively and aligning with their brand identity.
6. **Delayed Updates:** With multiple vendors, there may be delays in product listings or stock updates, leading to potential issues like users trying to buy out-of-stock items or outdated product details.
7. **Device Compatibility:** Although Flutter supports cross-platform development, certain device-specific bugs or performance issues may arise, particularly on less common devices or operating system versions.

5.2 Future Enhancement

- **Social Media Integration:** Integrating social media features into the app can significantly enhance user engagement and marketing reach. By allowing users to share their favorite products or purchases directly on platforms like Instagram, Facebook, or Twitter, the app can tap into social proof and word-of-mouth promotion. Additionally, implementing an affiliate marketing system where users can share referral links and earn commissions can incentivize sharing, driving more traffic and increasing sales for vendors.
- **Advanced Search and Filtering:** Improving the app's search functionality with refined filters such as size, color, brand, and price range will help users find products more efficiently. Adding advanced options like voice search and image-based search can further enhance accessibility, allowing users to find desired items quickly and intuitively. These features can provide a more user-friendly experience, especially for shoppers looking for specific products or categories.
- **Sustainability Features:** Incorporating sustainability into the app can appeal to eco-conscious users and promote ethical shopping practices. Options for filtering products based on eco-friendly certifications, sustainable materials, or other ethical criteria can help customers make informed decisions. Encouraging vendors to highlight their sustainability efforts and showcase eco-conscious products can also create a positive brand image and attract a broader audience.
- **Personalized Recommendations:** Leveraging AI and machine learning to offer personalized product recommendations can greatly enhance the user experience. By analyzing browsing and purchase history, the app can suggest items tailored to individual preferences, making the shopping journey more engaging and efficient. This personalization can boost customer satisfaction and encourage repeat purchases, ultimately driving higher sales for the platform.

References

Jain, S., & Jain, A. (2021, March). An Intriguing Analysis of E-Seller Challenges for Multi-Platform eCommerce. In 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 189-193). IEEE.

Akwukwuma, V. V. N., Chete, F. O., Okpako, A. E., & Nkwor, U. M. (2024). Usability Analysis of E-Commerce Mobile Applications. NIPES-Journal of Science and Technology Research, 6(3).

Edge, R., & Miola, A. (2022). Cross-Platform UIs with Flutter: Unlock the ability to create native multi-platform UIs using a single code base with Flutter 3. Packt Publishing Ltd

Torunoğlu, Z. (n.d.). Firebase Fundamentals. Medium.

Wikipedia. (n.d.). Flutter (Software).

GeeksForGeeks. (n.d.). Flutter Tutorial. Retrieved from GeeksForGeeks.

Flutter Documentation. (n.d.). Flutter Documentation. Retrieved from Flutter.

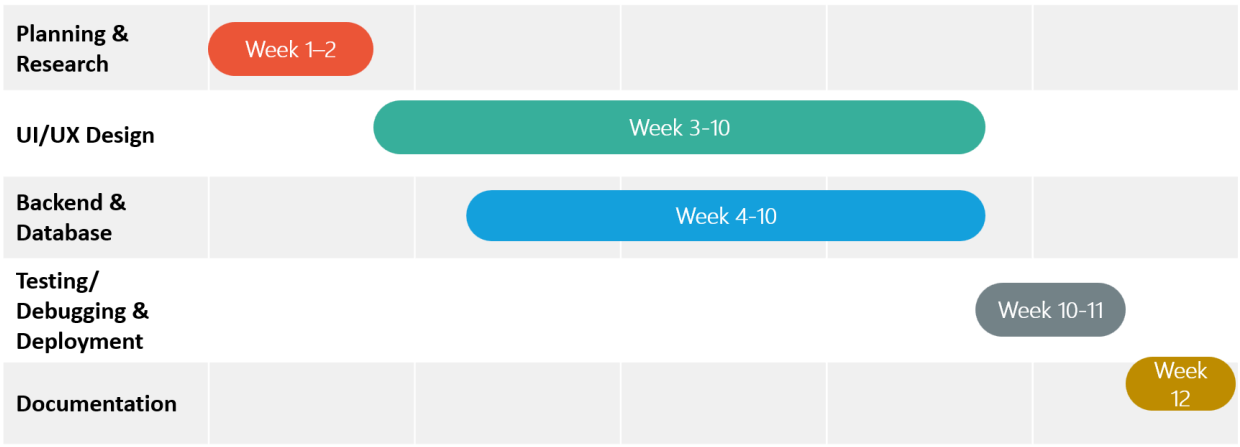
Daraz. (n.d.). Daraz Nepal. Retrieved from Daraz.

Jeeve. (n.d.). Jeeve. Retrieved from Jeeve.

Hamrobazaar. (n.d.). Hamrobazaar. Retrieved from Hamrobazaar.

APPENDIX

Project Timeline



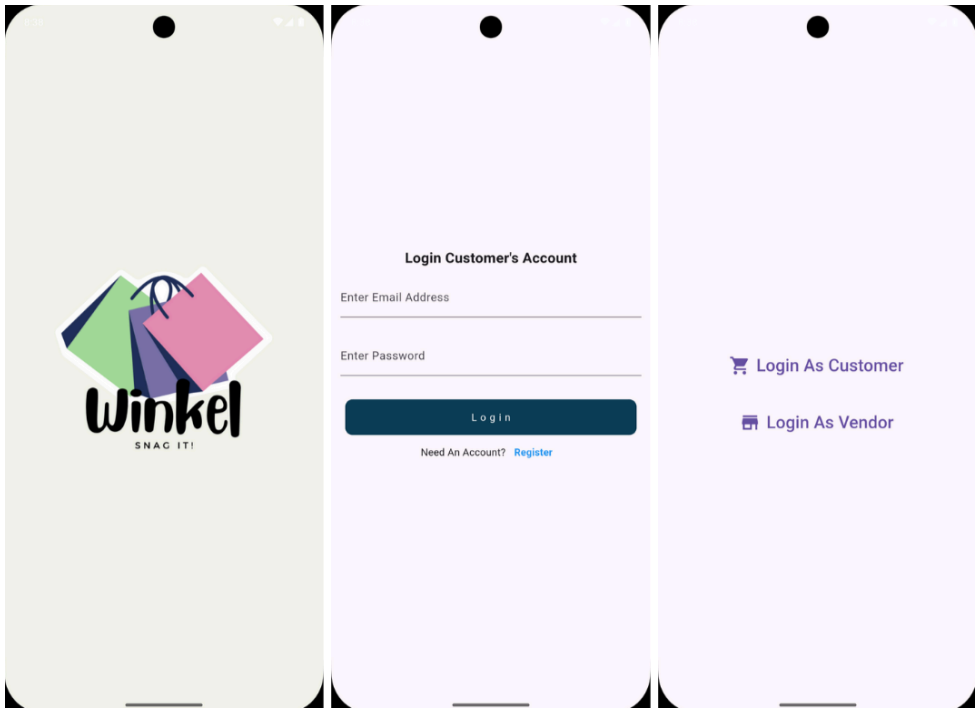
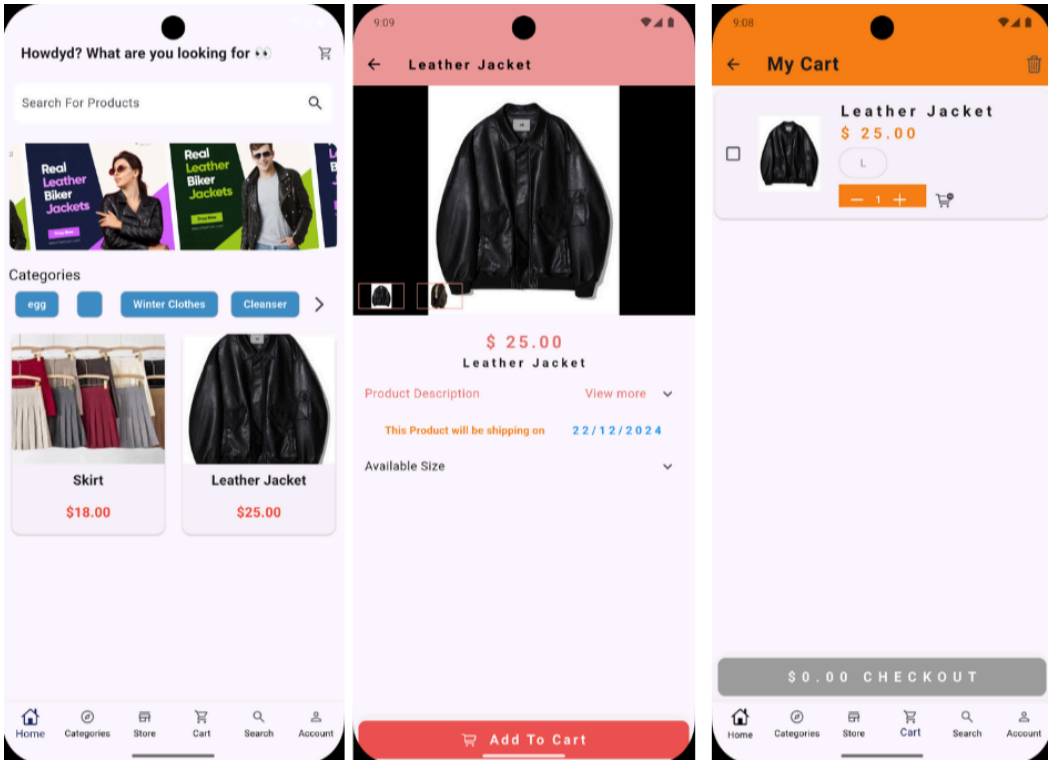


Figure 6.1 Main Screens



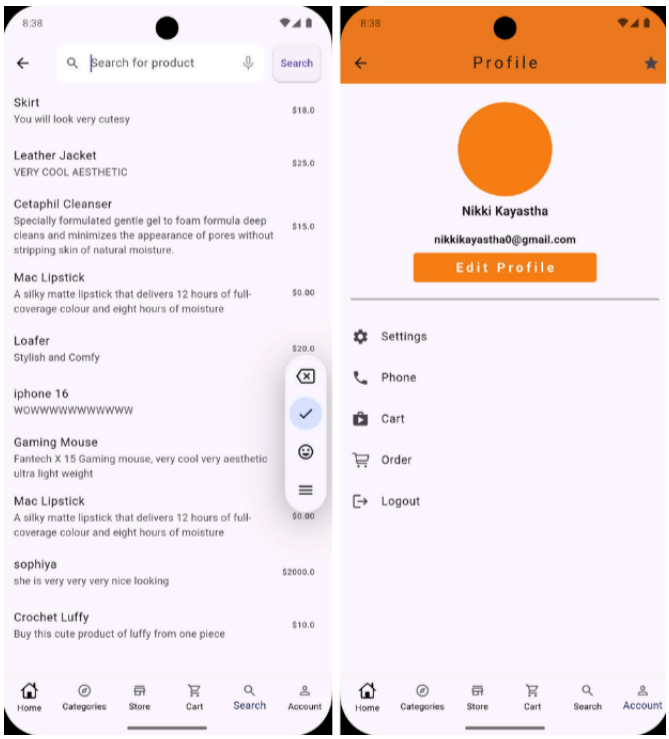


Figure 6.2 Customer Search and Profile Screen

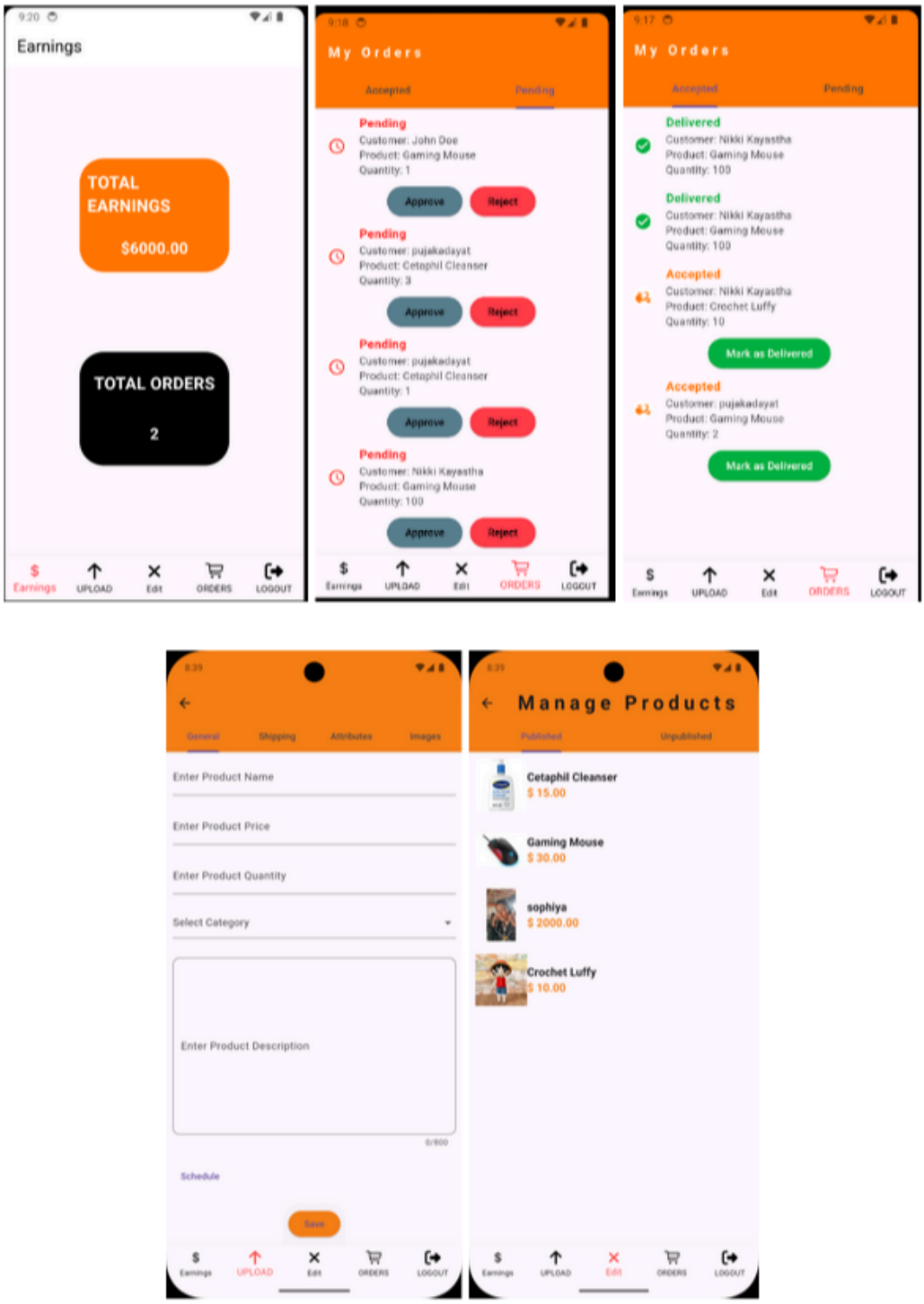


Figure 6.3 Vendor Screens

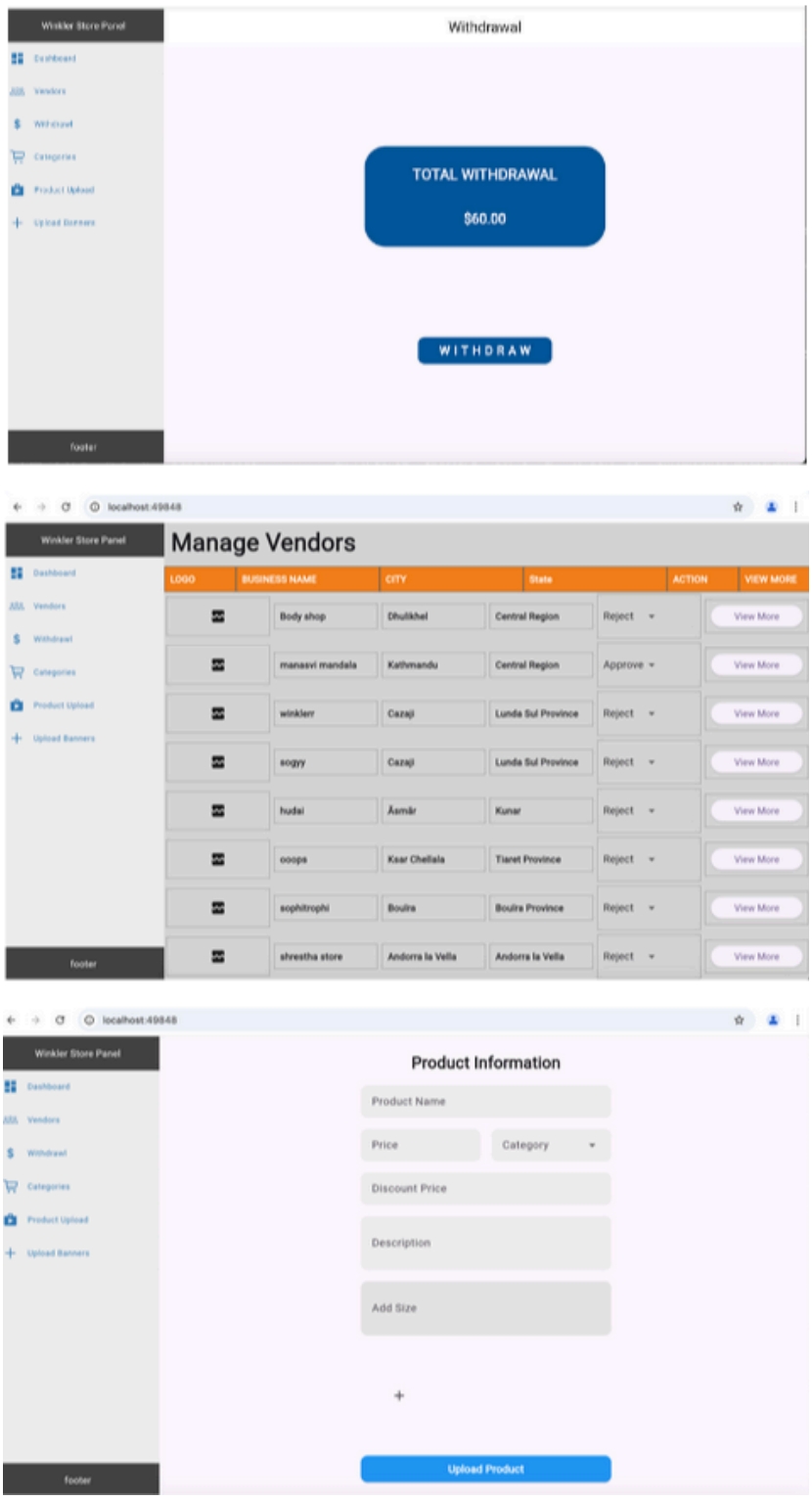


Figure 6.4 Admin Screens