VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

FACULTY OF COMPUTER SCIENCE

GRADUATION PROJECT

TOPIC NAME: BUILD AN E-COMMERCE APPLICATION THAT INTEGRATES AI

Student name: Phan Van Lai 20IT1028
Industry: Information Technology
Major: Software Engineering

Student name: Nguyen Quoc Thanh 20IT965
Industry: Information Technology
Major: Software Engineering

Instructor: Msc. Do Cong Duc

Da Nang, July 2025

VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY

FACULTY OF COMPUTER SCIENCE

GRADUATION PROJECT

TOPIC NAME: BUILD AN E-COMMERCE APPLICATION THAT INTEGRATES AI

Student name: Phan Van Lai 20IT1028
Industry: Information Technology
Major: Software Engineering

Student name: Nguyen Quoc Thanh 20IT965
Industry: Information Technology
Major: Software Engineering

Instructor: Msc. Do Cong Duc

Da Nang, July 2025

INSTRUCTOR COMMENTS

(By the instructor)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Danang, July 2025
Instructor

EXAMINER COMMENTS

(By the examiner)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Danang, July 2025
Examiner

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to all those who supported and accompanied us during our studies and the implementation of this graduation project.

First of all, we sincerely thank **MSc. Do Cong Duc**, our project supervisor, for his dedicated guidance, valuable feedback, and continuous support throughout the development of our project titled "Build an e-commerce application that integrates AI" His mentorship played a crucial role in helping us complete this project successfully.

We also extend our thanks to all lecturers at the Vietnam – Korea University of Information and Communication Technology, The University of Danang (VKU) for providing us with the foundational knowledge and professional skills that enabled us to carry out this project confidently.

We are grateful to our classmates and team members for their cooperation and support throughout the working process.

Lastly, we would like to thank our families for always being a strong source of motivation and encouragement during our studies and project work.

Although we have made great efforts, due to limited experience and time, this report may still contain shortcomings. We sincerely welcome any feedback from our instructors to help us improve in the future.

We are truly grateful.

STATEMENT OF AUTHORSHIP

We hereby declare that this graduation project report, titled “Build an e-commerce application that integrates AI”, is the result of our own work. It has not been copied or plagiarized from other sources. All references and sources of information used in the project have been properly cited and acknowledged.

We confirm that this report has not been submitted, in whole or in part, for any other degree or qualification at this or any other institution.

We take full responsibility for the content of this report.

Danang, July 5th, 2025

NGUYEN QUOC THANH

PHAN VAN LAI

TABLE OF CONTENTS

ABBREVIATIONS

ABBREVIATIONS	MEANING
AI	Artificial Intelligence
API	Application Programming Interface
AOT	Ahead-Of-Time (Compilation)
B2B	Business to Business
CDN	Content Delivery Network
CRD	Create, Read, Delete
CRU	Create, Read, Update
CRUD	Create, Read, Update, Delete
GPS	Global Positioning System
JIT	Just-In-Time (Compilation)
N/A	Not Applicable
RU	Read, Update
SDK	Software Development Kit
SMEs	Small and Medium-sized Enterprises
SSL	Secure Sockets Layer
UI	User Interface
UX	User Experience

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1. Problem statement

In recent years, e-commerce has witnessed remarkable growth both globally and in Vietnam. However, many small and medium-sized enterprises (SMEs) still face challenges in adopting cost-effective and user-friendly digital commerce solutions. There is a growing demand for flexible, scalable, and accessible platforms that can help businesses expand their online presence efficiently.

In this context, building a cross-platform e-commerce application becomes highly relevant. Such an application can support businesses in managing their online stores, handling customer interactions, and streamlining product and order management in a convenient and scalable manner.

2. Aims and Objectives

The overall aim of this project is to develop a cross-platform e-commerce application to improve user experience and support business operations.

The specific objectives of this thesis are to:

- Develop a multi-platform mobile application using Flutter.
- Integrate Firebase as a robust backend for data management and user authentication.
- Build essential e-commerce functionalities, including product listing, shopping cart, user authentication, order tracking, and payment processing.
- Design a user-friendly interface to ensure smooth user interaction and usability.

3. Structure of the thesis

After the *Introduction*, the thesis is structured in three chapters:

Chapter 1: Overview of E-commerce. This chapter presents the fundamental concepts of e-commerce, analyzes popular e-commerce platforms in the Vietnamese market such as Shopee, Lazada, and Tiki, and identifies opportunities and limitations for new solutions.

Chapter 2: System Analysis and Design. This chapter includes requirement analysis, system architecture, database modeling using Firebase, and the design of user interfaces based on Material Design principles.

Chapter 3: AI Integration and Smart Features. This chapter proposes the implementation of AI-driven modules including product image generation using Stable Diffusion, smart product recommendation systems, chatbot integration, and optimization strategies.

Finally, there are *Conclusions*, *Suggestions, References* and *Appendices* related to the topic.

CHAPTER 1. OVERVIEW OF E-COMMERCE

1.1 Overview Of E-Commerce

1.1.1 Definition and Characteristics of E-commerce

E-commerce, or electronic commerce, refers to the buying and selling of goods and services through electronic networks, primarily the Internet. Its key characteristics include digital transactions, automation of processes, global accessibility, and 24/7 availability.

There are several common types of e-commerce models:

- Business to Consumer (B2C): Businesses sell directly to individual customers (e.g., Shopee, Tiki).
- Business to Business (B2B): Transactions between companies, often involving wholesale or bulk orders.
- Consumer to Consumer (C2C): Individuals sell to each other through platforms like Facebook Marketplace or Cho Tot.

E-commerce has experienced significant global growth in recent years. In Vietnam, the market is expanding rapidly due to increasing internet penetration, mobile device usage, and improved logistics. The government's digital economy strategy further supports the development of online commerce.

1.1.2 Popular E-commerce Platforms

In recent years, various e-commerce platforms have rapidly evolved to dominate the digital marketplace in Southeast Asia, particularly in Vietnam. Among the most notable are Shopee, Lazada, and Tiki—each playing a crucial role in shaping the way consumers shop online and how sellers conduct business in the digital age.

Analysis of Leading Platforms

- Shopee

  Shopee is one of the leading e-commerce platforms in Southeast Asia, known for its mobile-first approach, aggressive marketing strategies, and wide range of product categories. It features a user-friendly interface, frequent promotional campaigns (like Flash Sales and Free Shipping), and strong logistics support via Shopee Xpress. It also emphasizes social shopping by incorporating features like live streaming and in-app games.

- Lazada

  Lazada, owned by Alibaba Group, leverages advanced logistics and backend technologies. Lazada stands out for its seller center tools, payment gateways, and integrations with global supply chains. The platform is designed to scale and accommodate both SMEs and large brands.

- Tiki

  Tiki started as a book-selling platform and has expanded to become one of Vietnam's trusted e-commerce sites. It differentiates itself by focusing on genuine products, fast delivery (TikiNOW), and a more curated catalog. Tiki has a strong domestic user base and is known for customer service and reliable product quality.

Advantages and Limitations of Existing Platforms

Table 1.1 Strengths and Limitations of Shopee, Lazada, and Tiki

Platform	**Strengths**	**Limitations**
Shopee	Wide user base, attractive promotions, social features	Fake products, intense competition, seller quality inconsistency
Lazada	Advanced logistics, strong seller tools, Alibaba backing	Complicated seller onboarding, less user engagement
Tiki	High product quality, local trust, fast delivery	Smaller product range, limited international expansion

1.2 Programming language

1.2.1 Dart

Dart is an object-oriented programming language developed by Google, optimized for building efficient and smooth user interfaces across multiple platforms. It is especially prominent when used with Flutter – a popular cross-platform UI toolkit. With its clear and concise syntax, Dart enables developers to quickly build and deploy applications on mobile, web, desktop, and even backend systems.

Advantages of Dart:

- High performance: Dart compiles directly to native machine code, allowing apps to run quickly and smoothly on both Android and iOS.
- Hot reload: Enables real-time updates to the UI and application logic without restarting the app, significantly speeding up development and testing.
- Easy to learn and write: Dart's syntax is friendly and similar to Java or JavaScript, making it accessible for both beginners and experienced developers.
- Rich library ecosystem: Dart offers a wide range of built-in and third-party libraries, making it easy to implement common features like HTTP requests, JSON parsing, animations, and more.
- UI-focused design: Dart is designed with user interface development in mind, making it highly suitable for Flutter's reactive programming model and flexible UI architecture.

Figure 1.1 Dart

1.2.2 Python

Python is a high-level programming language known for its readable and easy-to-learn syntax, widely used in many fields such as web development, artificial intelligence (AI), machine learning, and data science. Python was developed in the late 1980s by Guido van Rossum, and first released in 1991. The goal of Python is to be an easy-to-understand, accessible language for programmers. Over the years, Python has become one of the most popular languages, with strong community support and integrated libraries for a variety of applications.

Figure 1.2 Python

Advantages and benefits:

- Simple, easy-to-read syntax: One of the reasons Python is loved is that its syntax is close to natural language, making it easy for programmers to write and understand source code. This reduces development time and makes code easier to maintain later.

- Versatile and flexible: Python can be used in a wide variety of domains. From web development with frameworks like Django and Flask, to data science, automation, machine learning, and artificial intelligence (AI) with libraries like TensorFlow, scikit-learn, pandas.

- Large community and documentation: Python has a huge community of programmers. This means there are a lot of learning materials, libraries, and libraries, and online tutorials, help programmers easily solve problems encountered.

- Libraries and Frameworks: Python has a huge library ecosystem, especially in areas like data science, AI, and web development. Libraries like NumPy, pandas, and Matplotlib help with fast data processing and analysis, while frameworks like Django and Flask make web development a breeze.

- Suitable for small and large projects: Python can be used for everything from small, one-off projects to large, complex applications. Thanks to its clear syntax and good extensibility, Python is often chosen for both research, academic projects and large commercial products.

- Cross -platform: Python runs on almost every operating system, including Windows, macOS, and Linux, making it easy for developers to deploy applications across multiple platforms.

1.3 Framework

1.3.1 Flutter

Flutter is an open-source UI toolkit developed by Google, first introduced in 2017. It enables developers to build high-performance, cross-platform applications for mobile (Android and iOS), web, and desktop using a single codebase. With Flutter, developers write code in Dart—a modern, object-oriented programming language optimized for building user interfaces efficiently and expressively. One of Flutter's standout features is its "hot reload" capability, which allows developers to see changes in real time without restarting the app, significantly accelerating the development and testing process. Unlike other frameworks that rely on native UI components, Flutter uses its own high-performance rendering engine (Skia) to draw widgets directly to the screen. This gives developers full control over the UI and ensures consistent design and performance across all platforms. Flutter also offers a rich set of customizable widgets and tools, making it easy to create visually attractive, responsive, and smooth user experiences. Thanks to its flexibility, performance, and strong community support, Flutter has quickly become one of the most popular choices for cross-platform app development.

Figure 1.3 Flutter

Advantages and benefits:

- Cross-platform development: Flutter allows developers to write a single codebase and deploy it across multiple platforms, including Android, iOS, web, Windows, macOS, and Linux. This significantly reduces development time and maintenance costs, especially for startups or teams with limited resources.
- High performance: Unlike many other cross-platform frameworks that rely on a bridge to communicate with native components, Flutter uses its own rendering engine (Skia) to draw everything directly to the screen. This results in smooth animations, fast load times, and a native-like performance on all platforms.
- Hot reload: This feature enables developers to instantly see the effects of code changes without restarting the entire application. It boosts productivity, allows for quick experimentation, easier debugging, and faster UI design iterations during development.
- Rich widget library: Flutter provides a comprehensive set of pre-designed and customizable widgets that follow both Material Design (Google) and Cupertino (Apple) guidelines. This makes it easy to build beautiful, responsive, and platform-adaptive user interfaces.
- Consistent UI and behavior: Since Flutter renders the UI itself and does not depend on native platform components, the look and feel of the app remain consistent across different operating systems. Developers don’t have to worry about UI differences caused by OS version variations.
- Strong community and Google support: Flutter has robust backing from Google and a rapidly growing global community. This means frequent updates, strong documentation, and a large ecosystem of plugins, packages, and learning resources that help accelerate development.

Comparison with React Native and Native Development

Table 1.2 Comparison of Flutter, React Native, and Native Development

Criteria	Flutter	React Native	Native Development
Language	Dart	JavaScript	Java/Kotlin (Android), Swift/Obj-C (iOS)
Rendering	Skia (custom UI engine)	Native components via JS bridge	Native system APIs
Performance	Near-native (compiled to ARM code)	Moderate (bridge causes some overhead)	Best (direct access to hardware)
UI Consistency	High (custom-rendered)	Medium (depends on native components)	High (native by default)
Development Speed	Fast (hot reload, single codebase)	Fast (but some bridging limitations)	Slower (separate codebases for each platform)
Learning Curve	Medium (Dart-specific)	Low (JavaScript is widely known)	High (platform-specific languages/tools)
Ecosystem Maturity	Growing rapidly	Mature, widely adopted	Mature, platform-bound
Tooling	Official support via Android Studio, VS Code	Strong community tools	Robust, platform-specific IDEs (Xcode, Android Studio)

1.3.2 FastAPI

FastAPI is a modern and fast web framework developed in Python, focused on building RESTful APIs. It was first released in 2018 by Sebastián Ramírez. FastAPI is notable for its ease of use, high performance, and strong support for Python type hints, which help reduce errors and improve development efficiency.

Some advantages and benefits of FastAPI framework:

- High Performance: FastAPI is built on Starlette and Pydantic, which makes it comparable in performance to frameworks like Node.js and Go.
- Integrated Type Hints: Use Python's type hints to reduce errors, automatically generate API documentation, and powerfully validate data.
- Automatic API documentation: FastAPI automatically generates OpenAPI and Swagger compatible API documentation, making API testing and development easier
- Easy to learn and easy to use: Clean syntax, easy to learn even for beginners, yet powerful enough for complex projects.
- Asynchronous support: FastAPI fully supports asynchronous features (async/await), helping to optimize performance for applications that require processing multiple tasks at the same time.

Figure 1.4 FastAPI

1.4 Platform

1.4.1 Firebase

Firebase is a comprehensive application development platform that provides developers and teams with a rich set of tools and services to build, improve, and scale

web and mobile applications efficiently. Originally launched in 2011 by James Tamplin and Andrew Lee, Firebase began as a real-time backend solution designed to synchronize data between users and applications in real-time. Since its acquisition by Google in 2014, Firebase has grown into a robust development suite integrated with the Google Cloud Platform

Figure 1.5 Firebase

Firebase Services: Authentication, Firestore, Storage, FunctionsFirebase offers a collection of core services that support backend development, enhance user engagement, and facilitate performance monitoring. The most widely used services include:

- Firebase Authentication: Simplifies user identity management by offering multiple login methods such as email/password, phone numbers, and federated identity providers like Google, Facebook, and Apple. It ensures secure authentication while providing ease of integration.
- Cloud Firestore: A flexible, scalable NoSQL cloud database that stores data in documents and collections. It supports powerful querying and real-time data synchronization across devices. Firestore is designed for offline support, enabling users to access content even without an internet connection.
- Firebase Cloud Storage: Ideal for storing user-generated content such as images, audio, and video. It is built on Google Cloud Storage and provides robust upload/download capabilities, secure access controls, and smooth integration with other Firebase services.

- Firebase Cloud Functions: A serverless backend solution that lets developers run backend code in response to events triggered by Firebase features or HTTPS requests. Cloud Functions automatically scale and eliminate the need for server management.

These services are modular and highly integrated, allowing developers to combine them to create scalable, full-featured applications.

Advantages of Firebase in Application Development:

Firebase provides several key benefits for building and maintaining modern applications:

- Real-time Updates: Firestore and Realtime Database allow instant data syncing, critical for apps like chat, collaborative editing, or e-commerce inventory.
- Cross-platform SDKs: Firebase supports Android, iOS, Flutter, and web, making it an excellent choice for cross-platform development.
- Security and Authentication: With Firebase Authentication and customizable security rules, developers can manage access and data protection easily.
- Scalability and Performance: Powered by Google Cloud, Firebase automatically handles traffic spikes and scales globally with minimal developer effort.
- Faster Development Cycles: Hot reload in Flutter, built-in services, and integration reduce development and testing time.
- Serverless Architecture: Cloud Functions allow code execution in response to database changes, authentication events, and API calls without provisioning servers.
- Built-in Analytics and Monitoring: Firebase integrates with Google Analytics, Crashlytics, and Performance Monitoring to provide deep insights into user behavior and app health.

1.4.2 Postman

Postman is a powerful and widely-used API development platform that supports developers in designing, testing, documenting, and managing APIs efficiently. With its intuitive graphical interface, Postman allows users to easily send HTTP requests (such

as GET, POST, PUT, DELETE), observe server responses, and validate API functionality without writing manual code. It has become an essential tool in modern software development workflows, especially in projects that use microservices architecture or RESTful APIs.

Figure 1.6 Postman

Postman offers a wide range of key advantages, including:

- User-friendly and intuitive interface, making it accessible even to those with limited technical knowledge.
- Support for various HTTP methods and data formats, allowing accurate simulation of real-world API interactions.
- Collection management, enabling users to group requests logically by module or function, and easily share them with others.
- Environment variables, which allow seamless switching between different environments like development, staging, and production.
- Built-in automated testing, with JavaScript-based scripts that validate API responses and ensure stability over time.
- CI/CD integration via Newman, enabling automated API testing as part of the software deployment pipeline.
- Team collaboration, with Postman Workspaces allowing multiple team members to work together in real time.

- API monitoring, which periodically checks API health and helps detect issues proactively.

- Automatic API documentation, saving time and simplifying communication with frontend developers, testers, and third-party partners.

Thanks to these advantages, Postman is more than just an API testing tool—it is a vital part of a professional and scalable API lifecycle management system.

CHAPTER 2. SYSTEM DESIGN ANALYSIS

2.1 System survey

2.1.1 Field survey

For the e-commerce application development project, the current state survey reveals that Vietnam's market is experiencing rapid growth with major platforms like Shopee, Tiki, and Lazada dominating the market. However, significant gaps remain in niche segments, B2B e-commerce, or specialized industry sectors. Vietnam's technological infrastructure is well-prepared with a developed electronic payment system, increasingly sophisticated logistics services, and high smartphone penetration rates. Traditional business models of many enterprises have not yet fully undergoing digital transformation, creating substantial opportunities for new e-commerce solutions.

2.1.2 User survey

Through surveying 500 potential customers, we identified that Vietnamese users prioritize fast and convenient shopping experiences with simple and user-friendly interfaces. They desire flexible payment methods, particularly cash-on-delivery and digital wallets. Personal information security and product reviews from other users are considered top priorities. For suppliers, they need simple management tools, bulk product upload support, and detailed revenue reporting. Administrators require comprehensive dashboards, real-time transaction monitoring capabilities, and powerful data analysis tools.

2.1.3 Business diagram

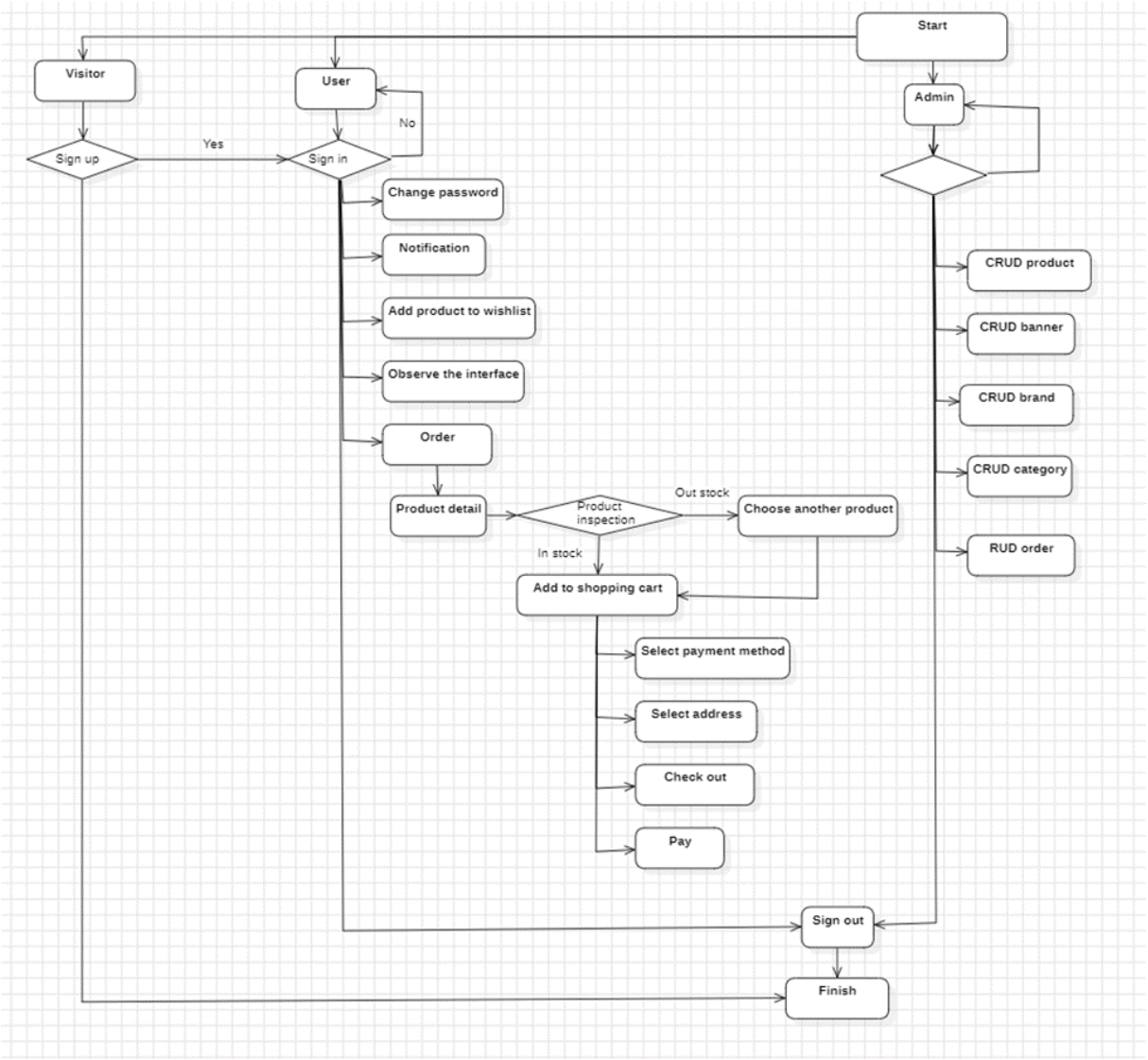

Figure 2.1 Business diagram

2.2 Software requirements specification

2.2.1 Identify the factors

The topic consists of three main factors

- User
- Admin

2.2.2 Functional requirements

***.User**

- Log in, log out, change password.
- Manage profile

- Manage address
- Manage product
- Manage wishlist product
- Managing product in the shopping cart
- Search
- View all product and brand
- View review and rating of product
- View order history and details
- Create, edit photos with AI

***.Admin**

- User management
- Manage media : CRD media
- Manage products : CRUD products
- Manage categories : CRUD categories
- Manage brands :`CRUD brands
- Manage banners : CRUD banners
- Manage order : RUD order
- Manage setting : CRU setting
- Manage profile : RU profile
- Data statistics
- Log in, log out, change password.

2.2.3 Non - functional requirements

- Interface, beautiful, simple, easy to use.
- Ensure security and safety.
- Fast processing speed.

2.3 Use Case Diagram

2.3.1 Overall

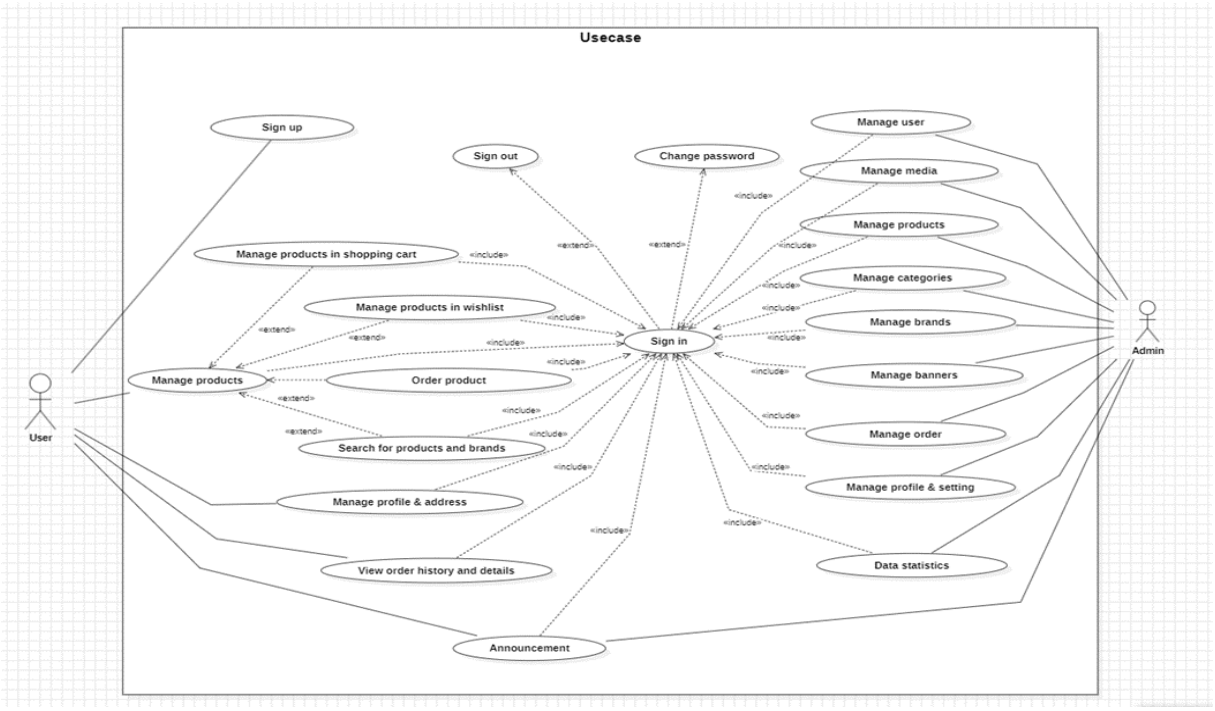

Figure 2.2 Overall use case

2.3.2 Managing product in the shopping cart

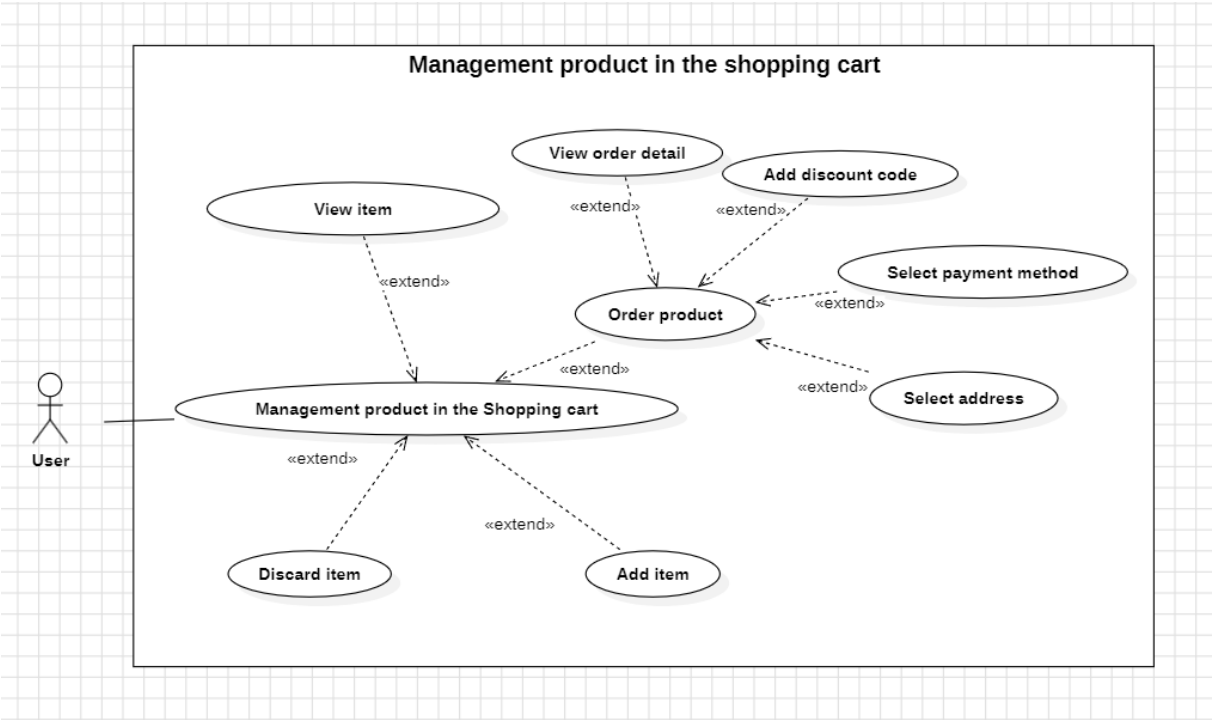

Figure 2.3 Managing product in the shopping cart

Table 2.1 Scenario Use-case "Managing product in the shopping cart"

Use case name	**Managing product in the shopping cart**
Description	Used to manage item-related activities such as collecting, viewing, using, crafting, upgrading, and disposing of items in the World.
Actors	User
Input	User has successfully logged in and enter shopping cart
Output	Successfully perform item management operations (View, add or delete items).
Basic flow	1. The Actor clicks "Shopping cart" icon -> The Use-case begins. 2. The Actor can choose one of the following actions: - View Item: Actor selects the "Inventory" icon. -> Select an item to view details (stats, description). - Add items: Actor can add items => Items are added to inventory. - Discard item: Actor selects item from inventory and discards => Item is removed from inventory. 3. The Actor enter "Checkout" to order product: - Select payment method: PayPal, Master card, Visa, Google Pay, … - Select Address. - Check order detail and enter Checkout to pay. 4. The system saves the inventory updates -> End of use case
Alternative flow	
Exception flow	If the Actor attempts to pay but does not have enough funds: The system will warn that this order cannot be executed.

2.3.3 Search

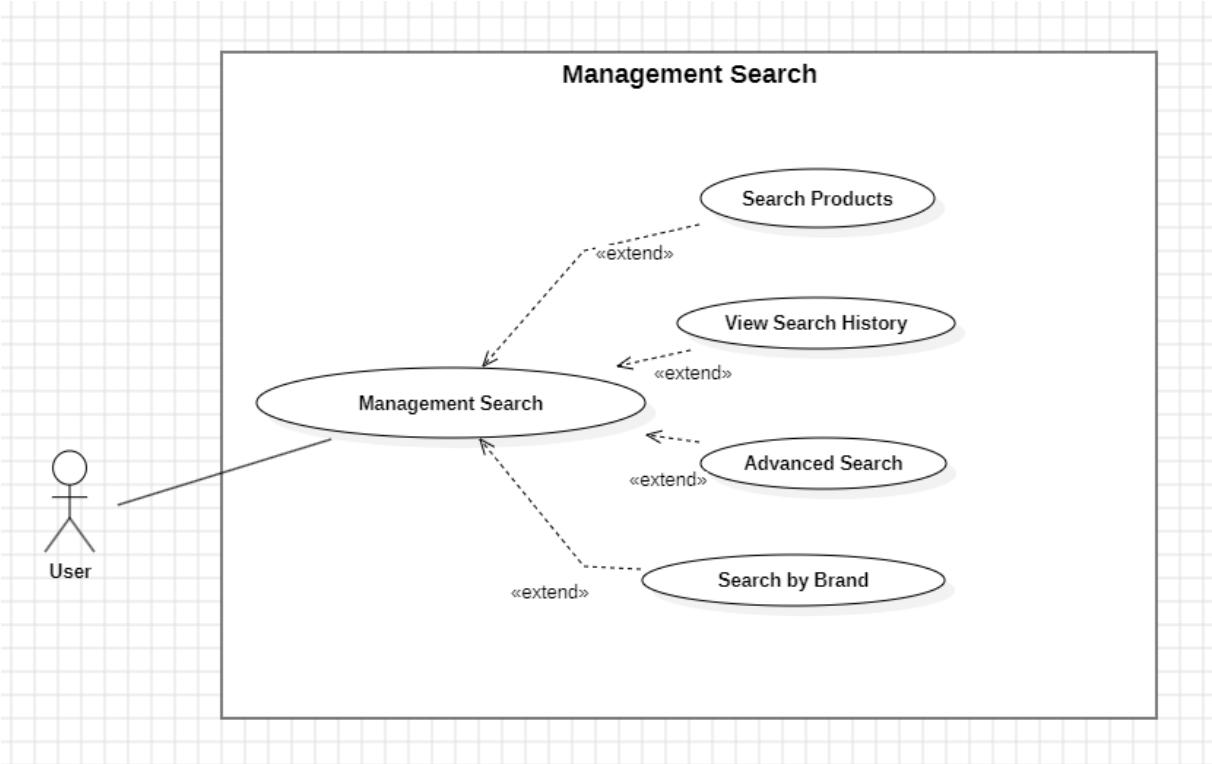

Figure 2.4 Use case search

Table 2.2 Scenario Use-case "Search"

Use case name	**Search**
Description	Used to manage search.
Actors	User
Input	Actor successfully logged in
Output	Show list of product, brand
Basic flow	1. Actor clicks on the "Search box" □Start Use case 2. Actor performs the necessary operations (view search history, search by product or brand) for the user. 3. The system records and updates user edits if any. □End of use case
Alternative flow	
Exception flow	

2.3.4 User

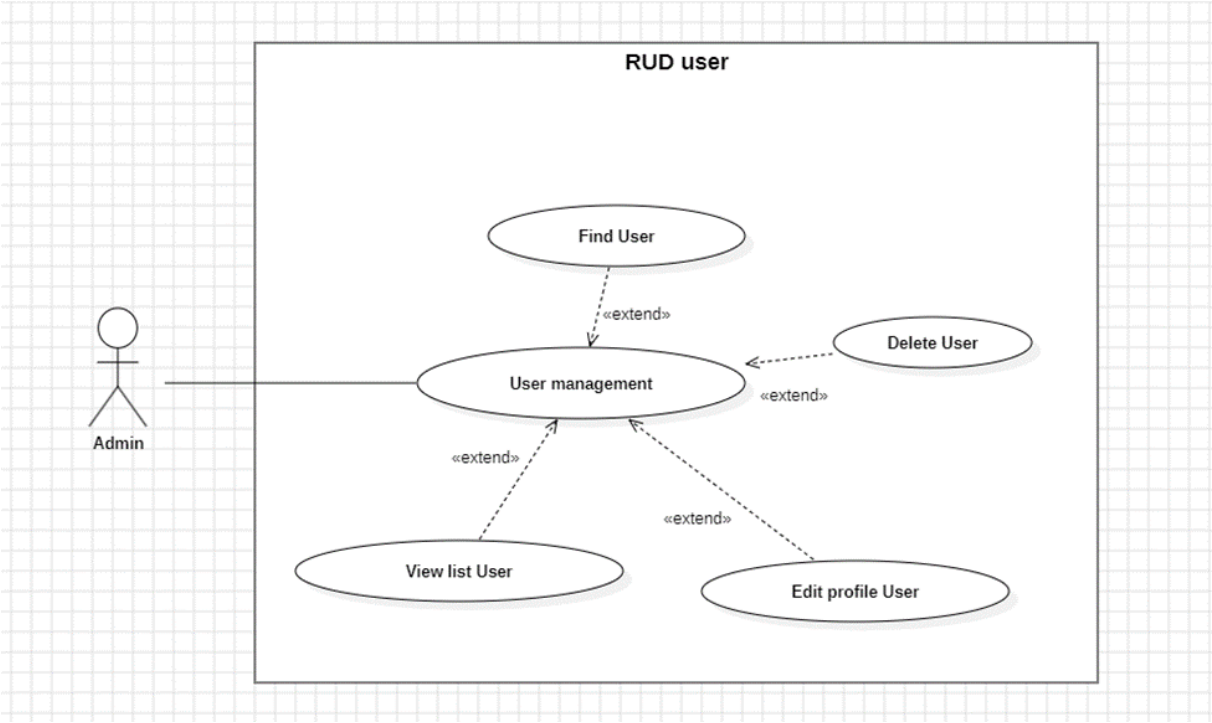

Figure 2.5 Use case User management

Table 2.3 Scenario Use-case "User"

Use case name	**User Management**
Description	Used to manage users.
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of users
Basic flow	4. Actor clicks on the "Customer" menu in the side-bar -> Start Use case 5. Actor performs the necessary operations (view details, edit, delete, search) for the user. 6. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	

2.3.5 Address

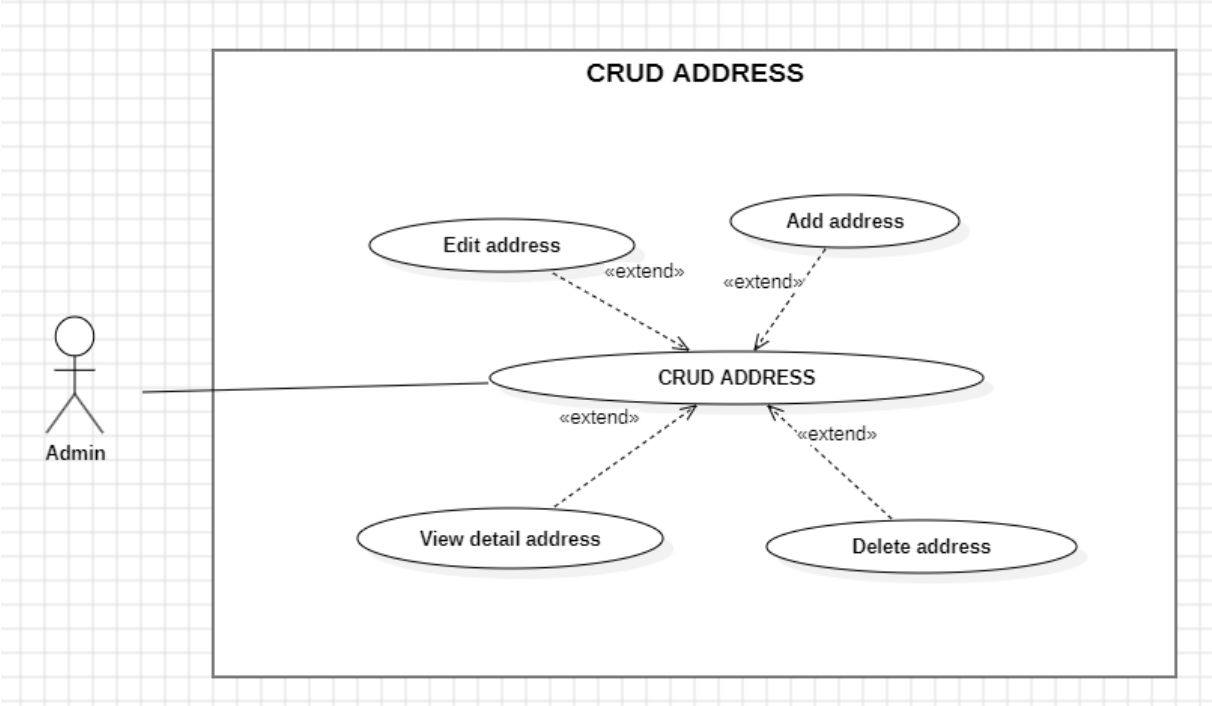

Figure 2.6 Use case address

Table 2.4 Scenario Use-case "Address"

Use case name	Address Management
Description	Used to manage address.
Actors	User
Input	Actor successfully logged in
Output	Show list of address
Basic flow	1. Actor clicks on the "My address" menu in the profile -> Start Use case 2. Actor performs the necessary operations (view details, edit, delete) for the user. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	

2.3.6 Product

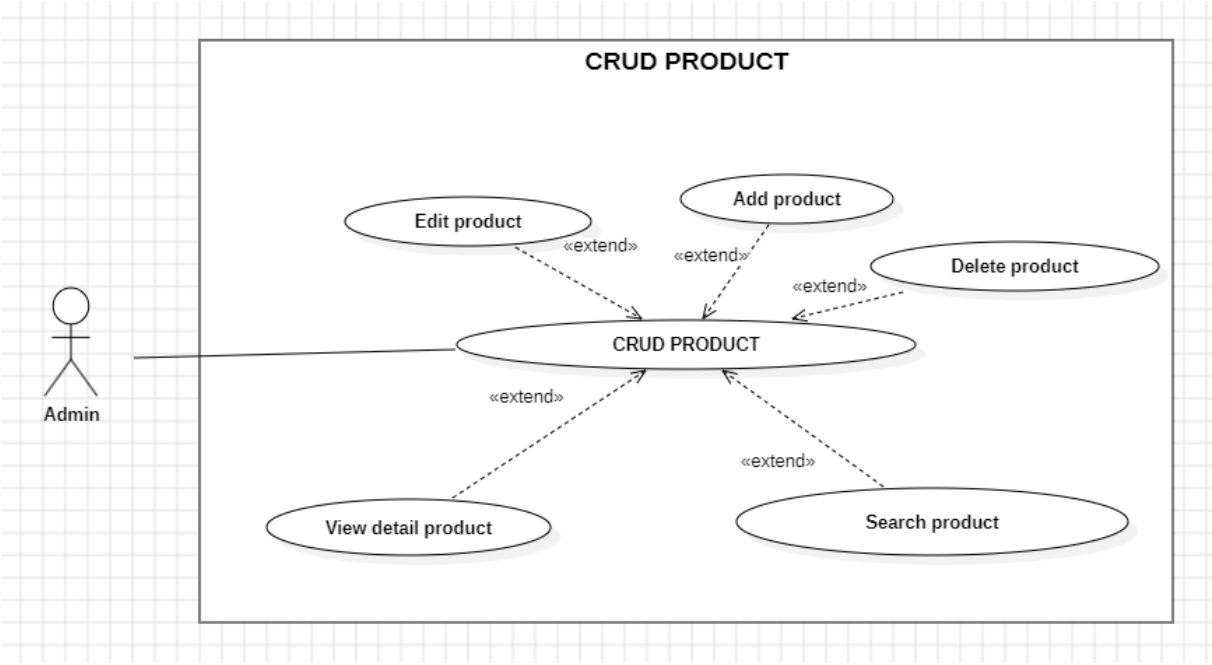

Figure 2.7 Use case product management

Table 2.5 Scenario Use-case "Product"

Use case name	**Product**
Description	Used to manage product
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of product
Basic flow	1. Actor clicks on the "Product" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, create, edit, delete, search) for the product. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.3.7 Banner

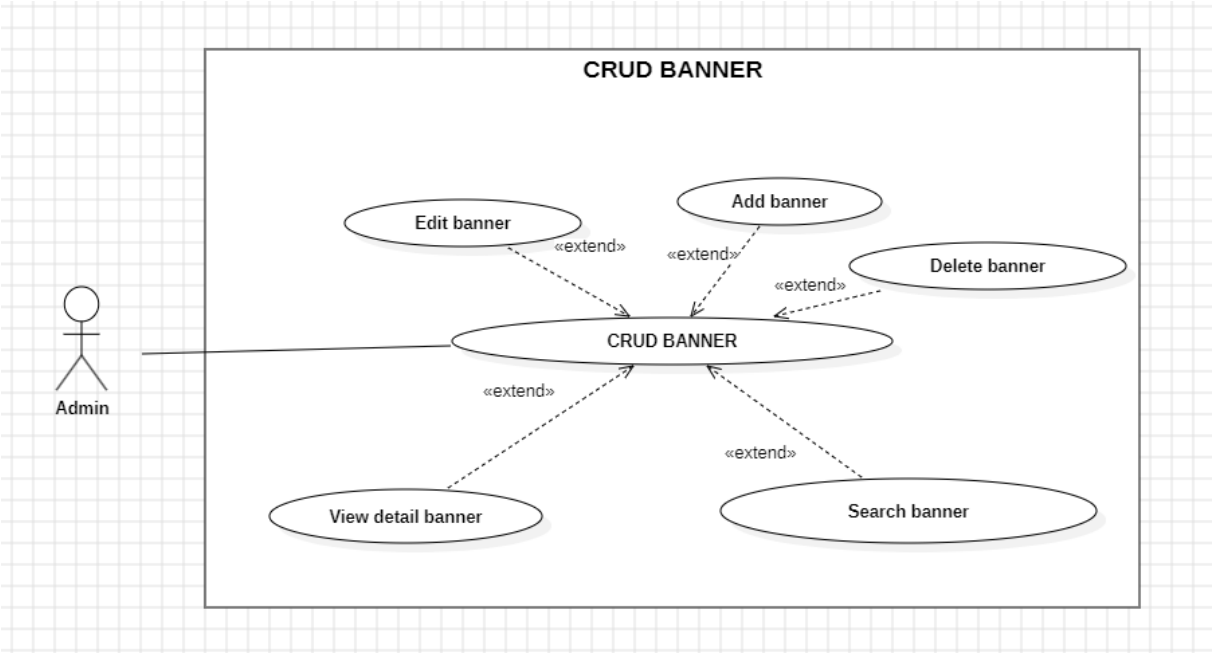

Figure 2.8 Use case banner management

Table 2.6 Scenario Use-case "Banner"

Use case name	**Banner**
Description	Used to manage banner
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of banner
Basic flow	1. Actor clicks on the "Banner" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, create, edit, delete, search) for the banner. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.3.8 Brand

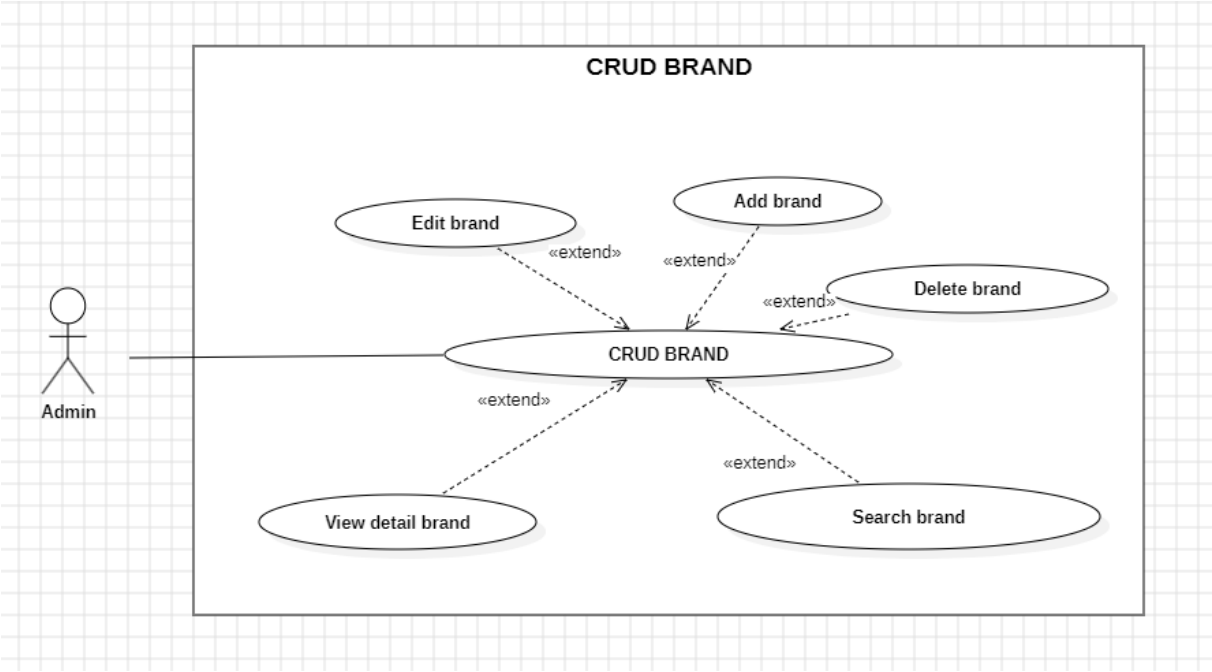

Figure 2.9 Use case brand management

Table 2.7 Scenario Use-case "Brand"

Use case name	**Brand**
Description	Used to manage brand
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of brand
Basic flow	1. Actor clicks on the "Brand" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, create, edit, search) for the brand. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.3.9 Category

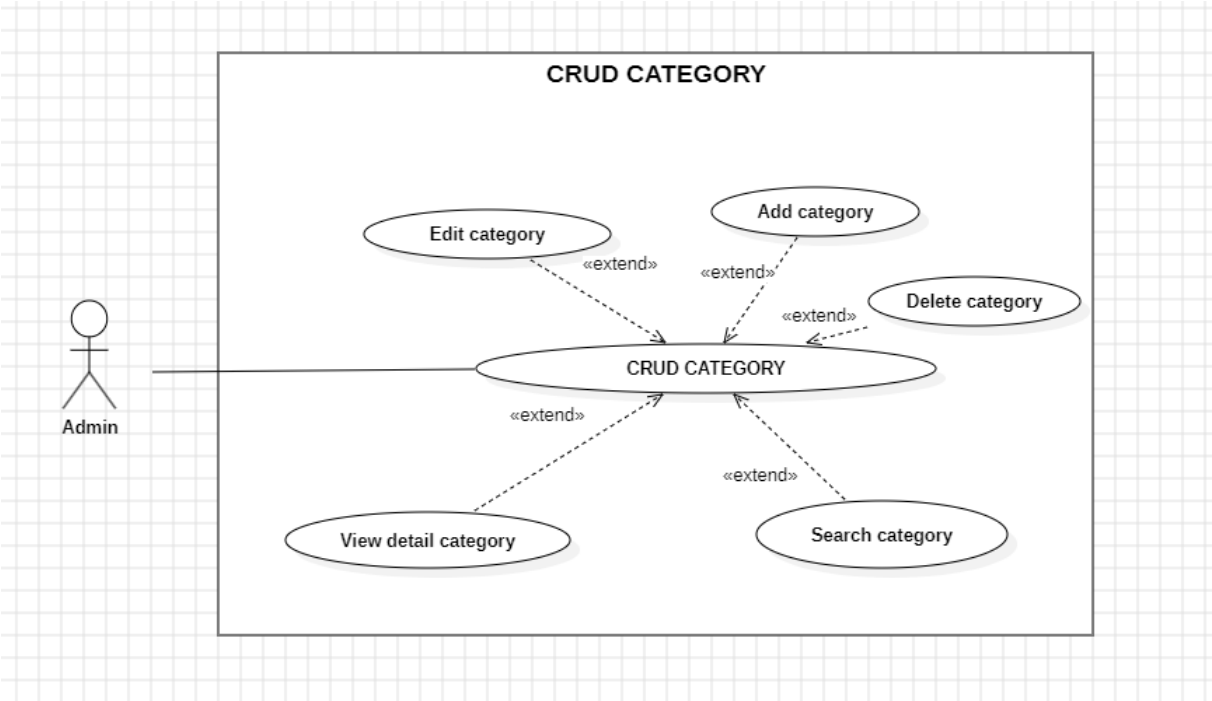

Figure 2.10 Use case category management

Table 2.8 Scenario Use-case "Category"

Use case name	**Category**
Description	Used to manage category
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of category
Basic flow	1. Actor clicks on the "Category" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, create, edit, delete, search) for the category. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.3.10 Media

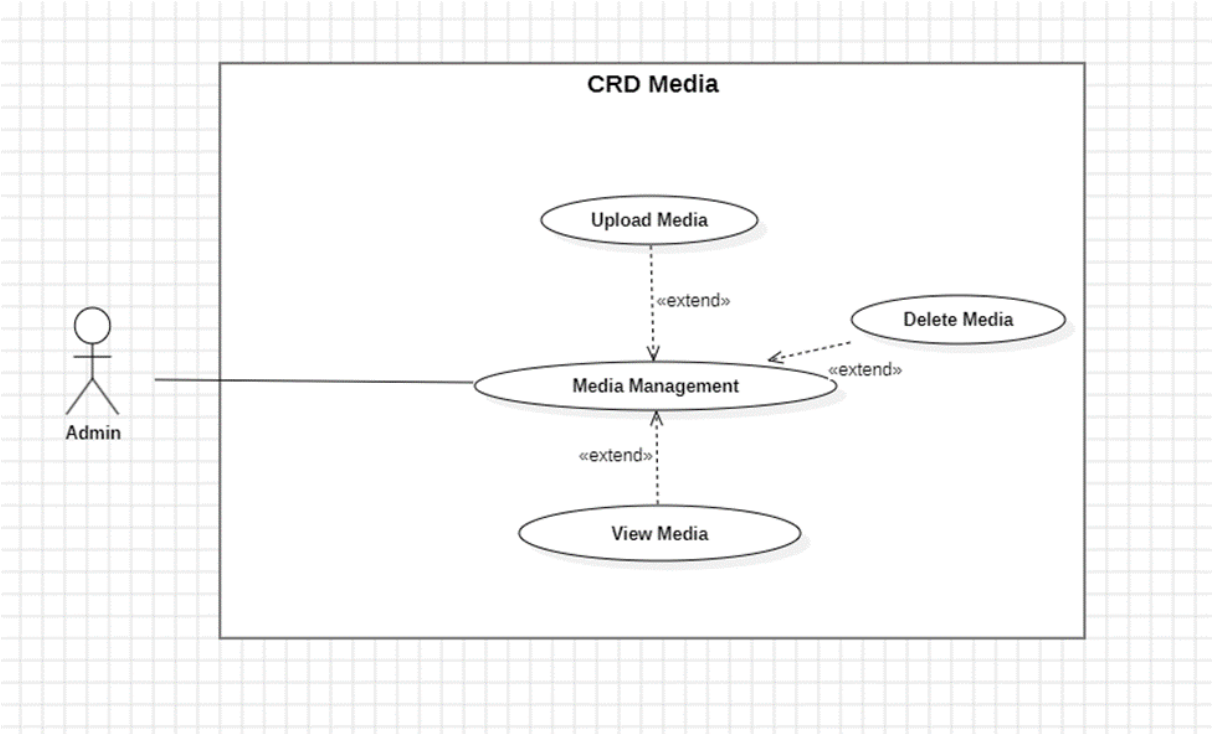

Figure 2.11 Use case media management

Table 2.9 Scenario Use-case "Media"

Use case name	**Media**
Description	Used to manage media
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of media
Basic flow	1. Actor clicks on the "Media" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, create, delete, search) for the media. 3. The system records user uploads if any. -> End of use case
Alternative flow	
Exception flow	If the newly created information is invalid (incorrect format, empty task name, etc.), the system will display an error message and ask for correction.

2.3.11 Order

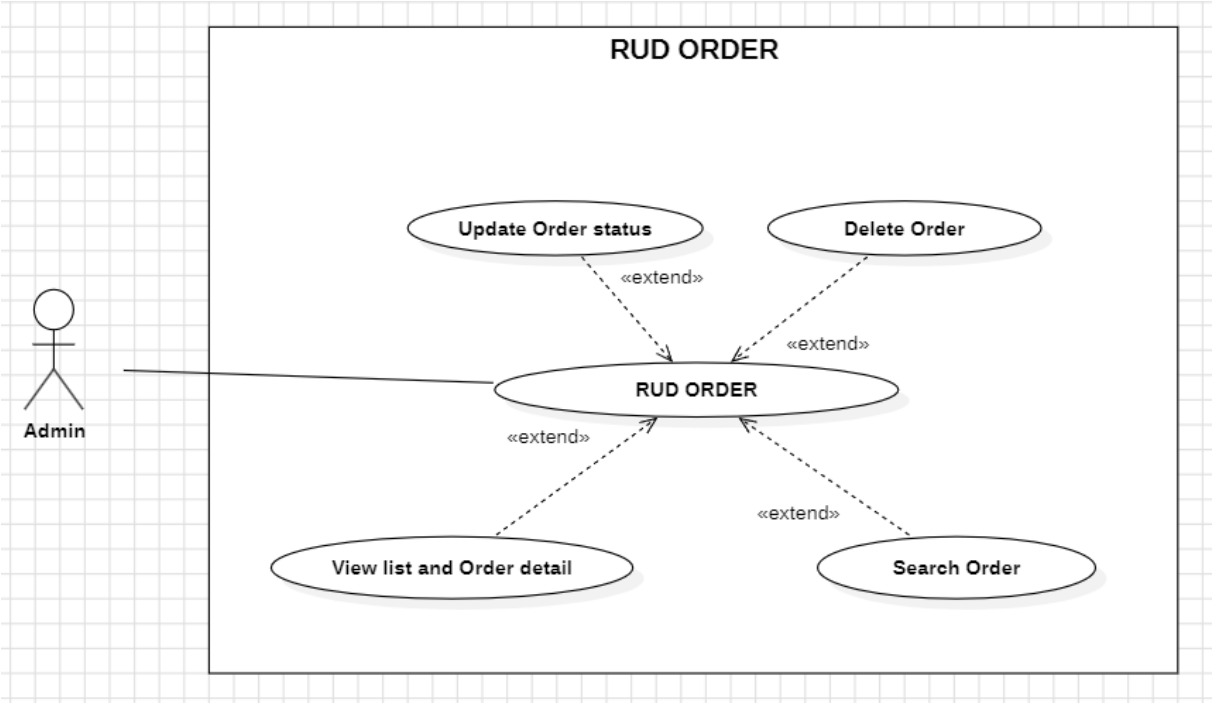

Figure 2.12 Use case order management

Table 2.10 Scenario Use-case "Order"

Use case name	**Order**
Description	Used to manage order
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show list of order
Basic flow	1. Actor clicks on the "Order" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, edit, delete, search) for the order. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.3.12 Profile

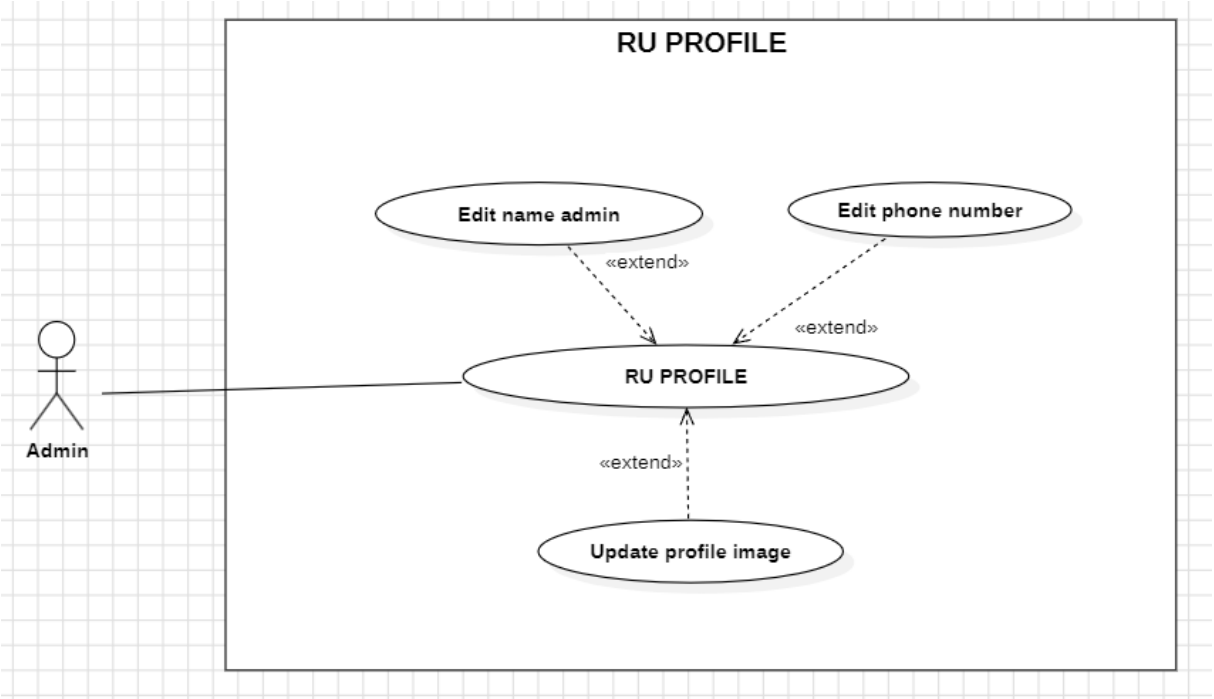

Figure 2.13 Use case profile management

Table 2.11 Scenario Use-case "Profile"

Use case name	**Profile**
Description	Used to manage profile
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show profile detail
Basic flow	1. Actor clicks on the "Profile" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, edit) for the profile. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.3.13 Setting

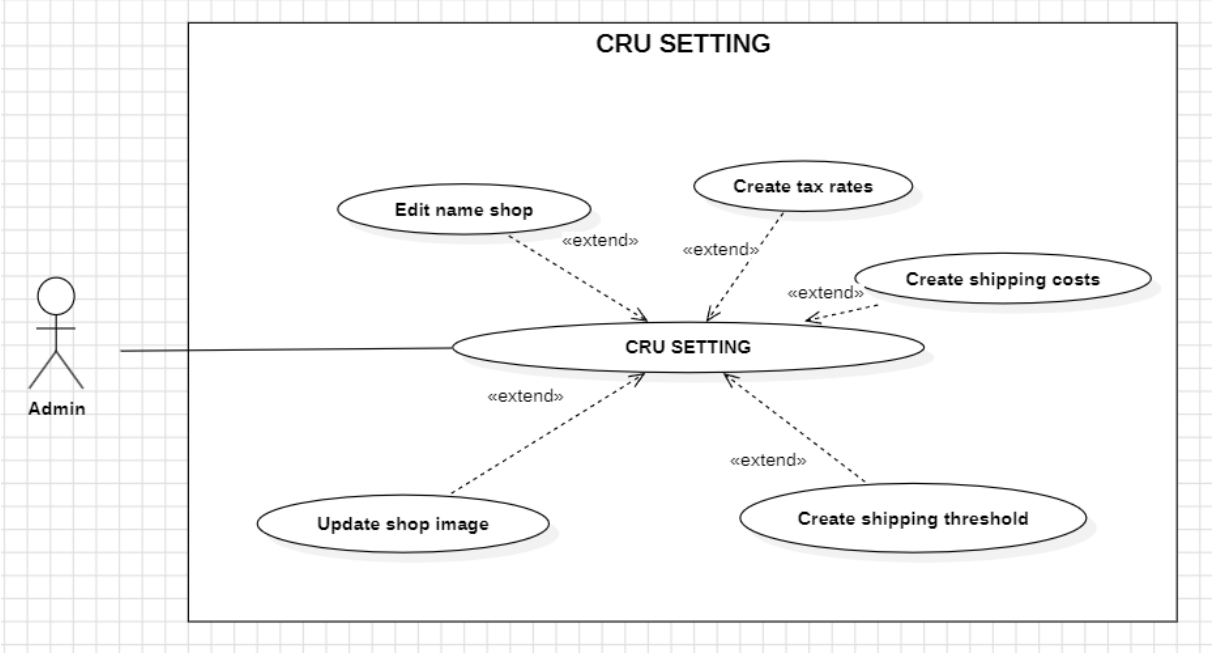

Figure 2.14 Use case setting management

Table 2.12 Scenario Use-case "Setting"

Use case name	**Setting**
Description	Used to manage setting
Actors	Admin
Input	Actor successfully logged in with admin rights
Output	Show setting detail
Basic flow	1. Actor clicks on the "Setting" menu in the side-bar -> Start Use case 2. Actor performs the necessary operations (view details, create, edit) for the setting. 3. The system records and updates user edits if any. -> End of use case
Alternative flow	
Exception flow	If the edited information is invalid (incorrect format, empty task name, etc.), the system will display an error message and request re-editing.

2.4 Workflow Diagram

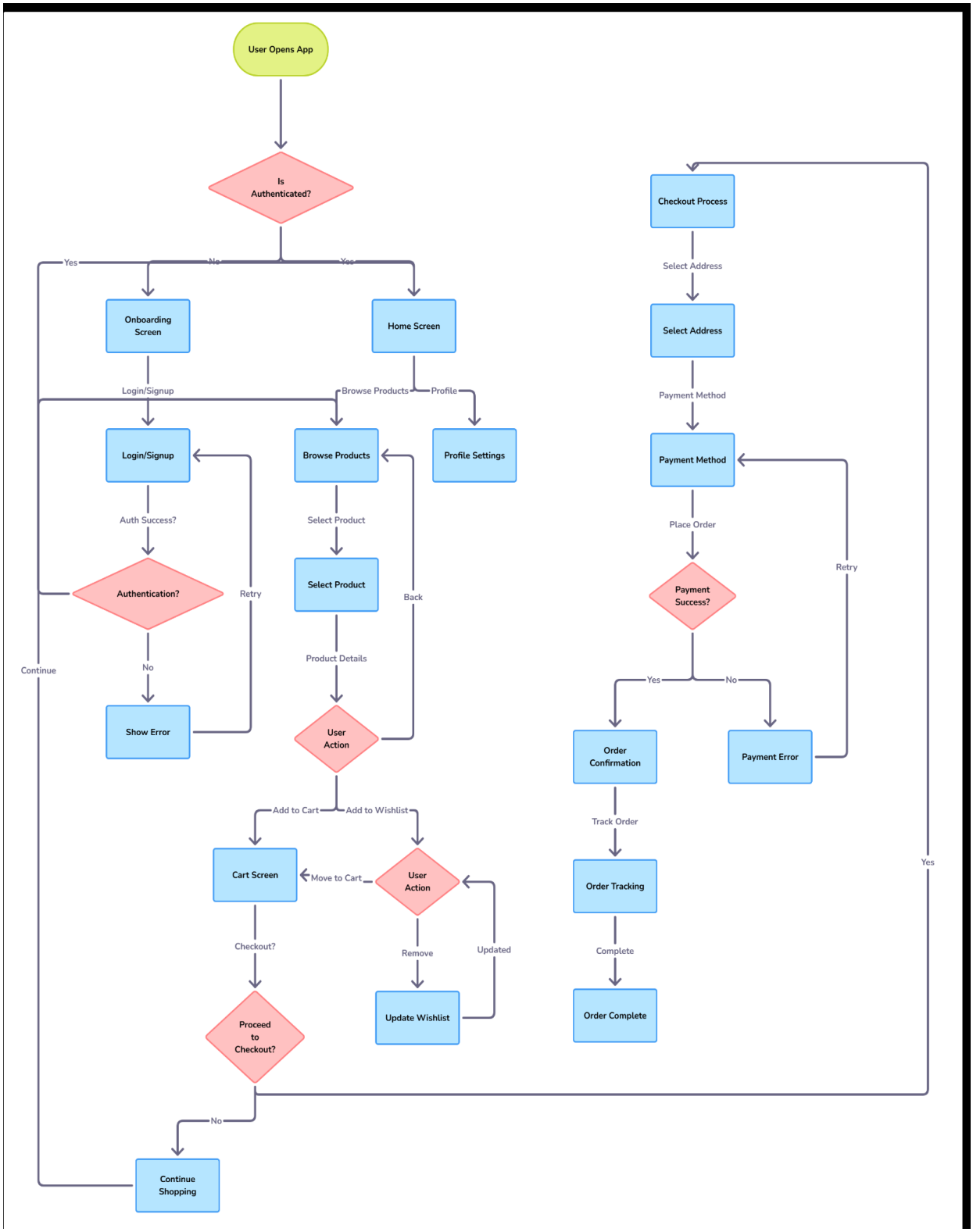

Figure 2.15 Workflow diagram

2.5 Operational diagram

2.5.1 Register and login

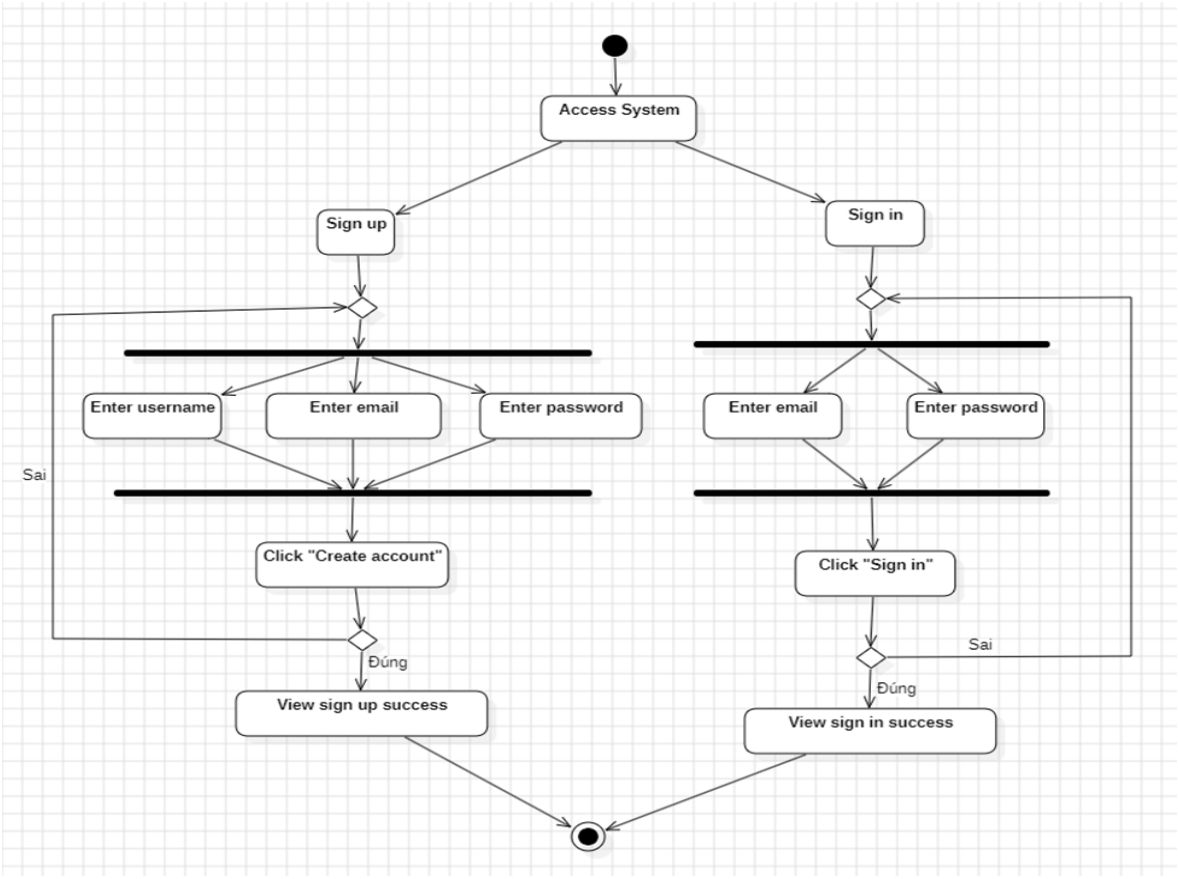

Figure 2.16 Activity "Register and Login"

2.5.2 User

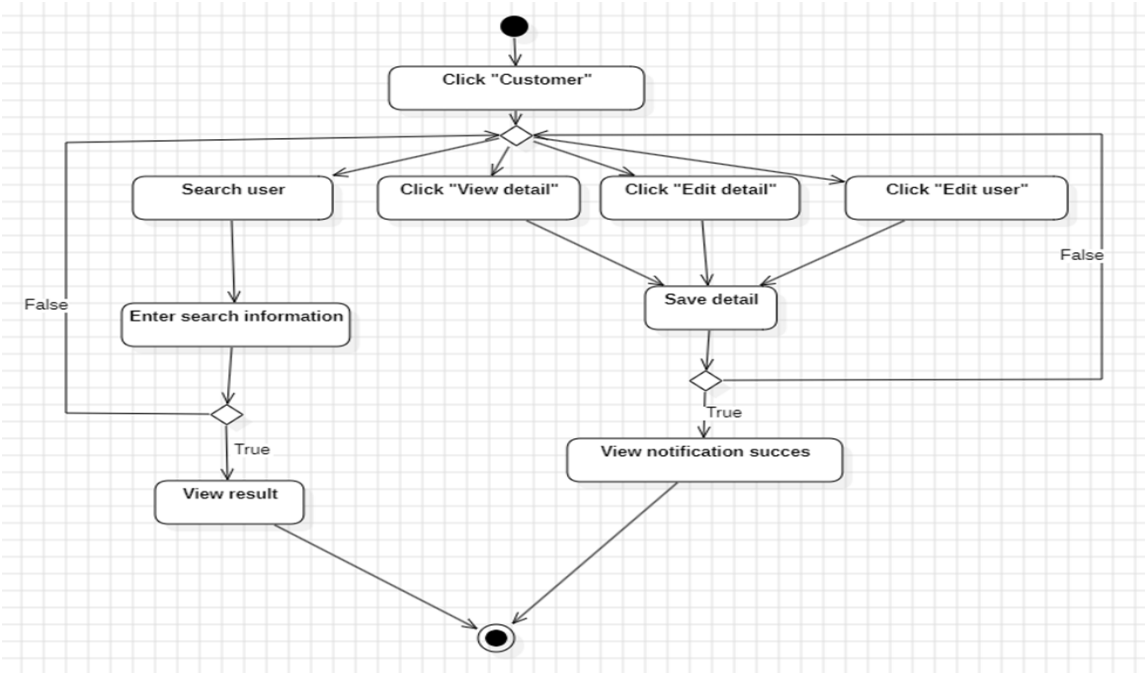

Figure 2.17 Activity "User"

2.5.3 Banner

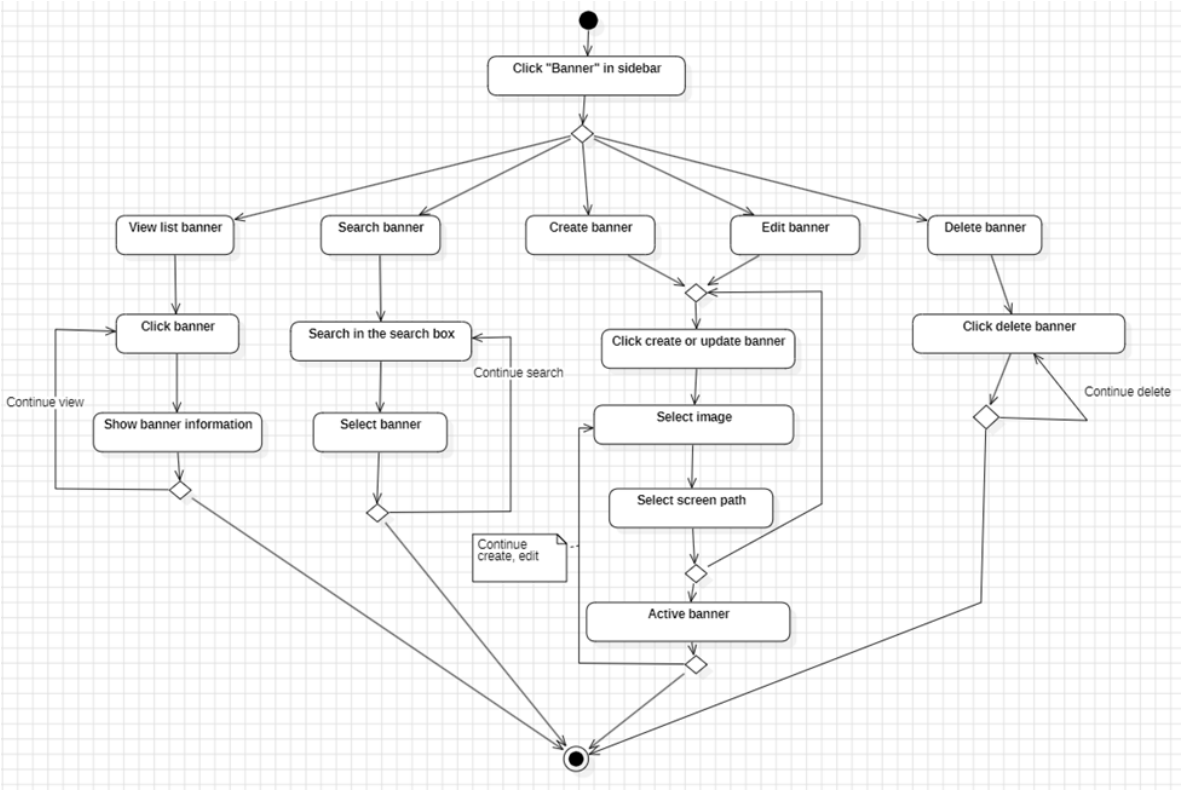

Figure 2. 18 Activity "Banner"

2.5.4 Brand

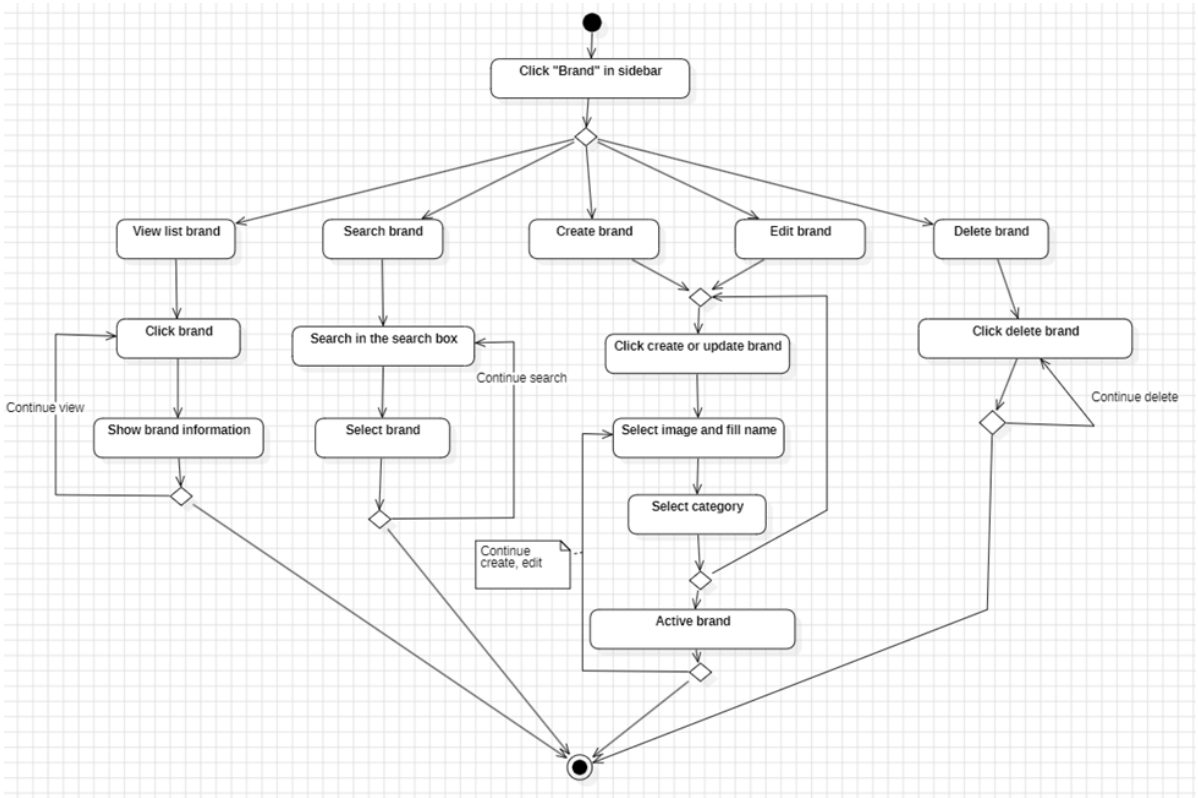

Figure 2. 19 Activity "Brand"

2.5.5 Category

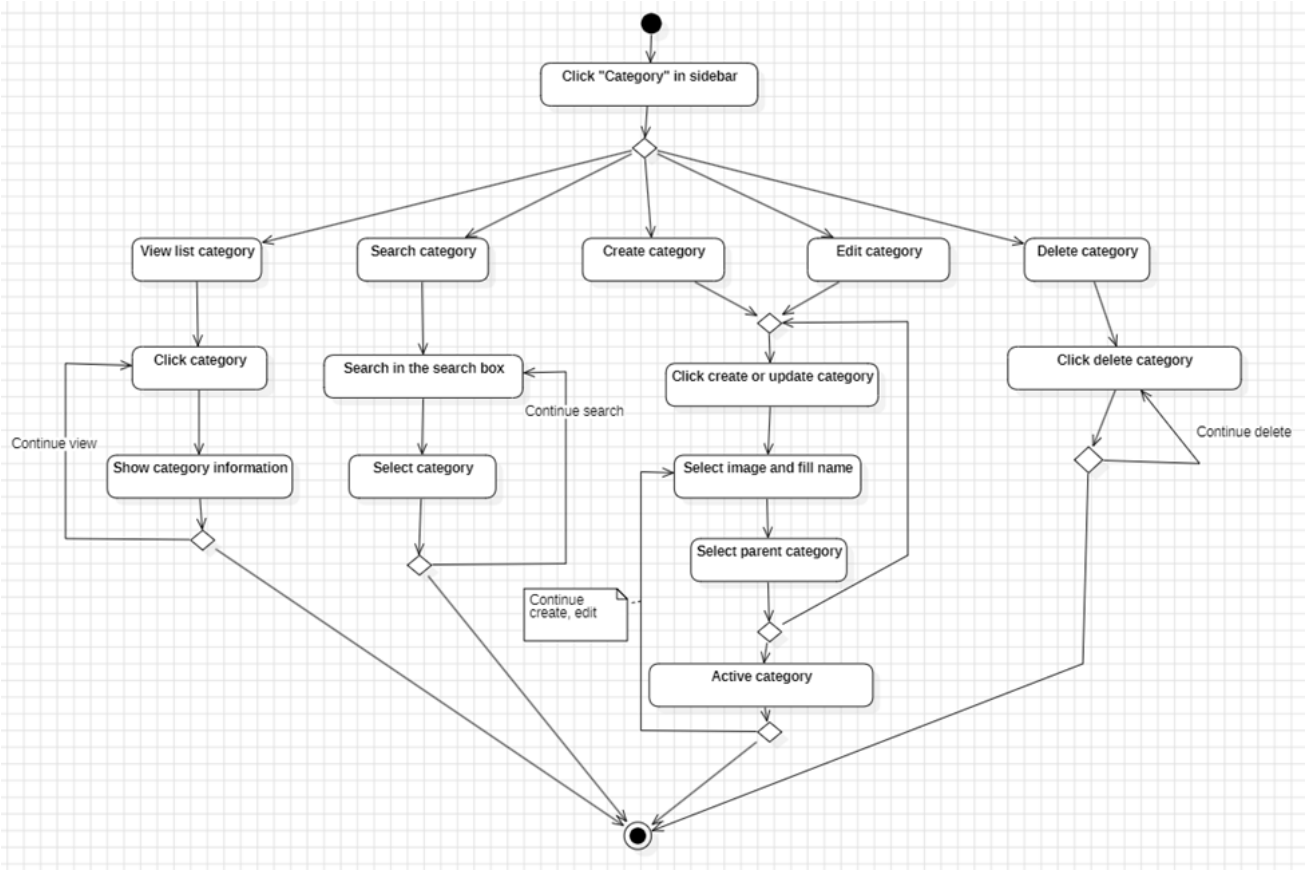

Figure 2. 20 Activity "Category"

2.5.6 Product

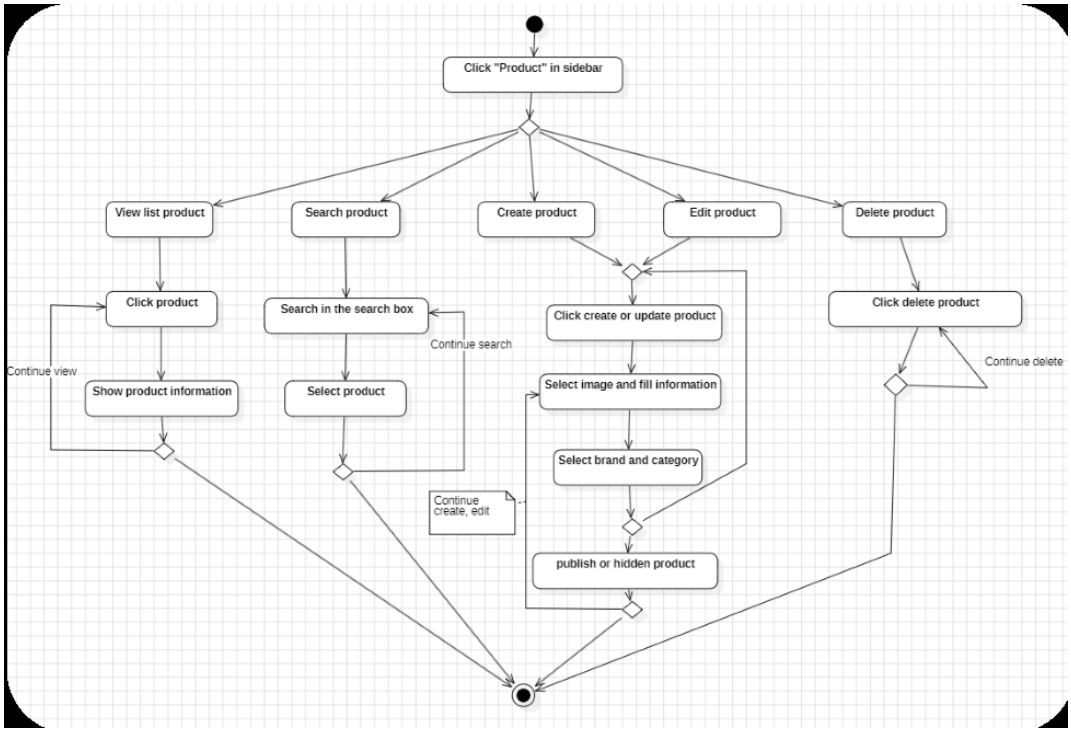

Figure 2. 21 Activity "Product"

2.5.7 Order

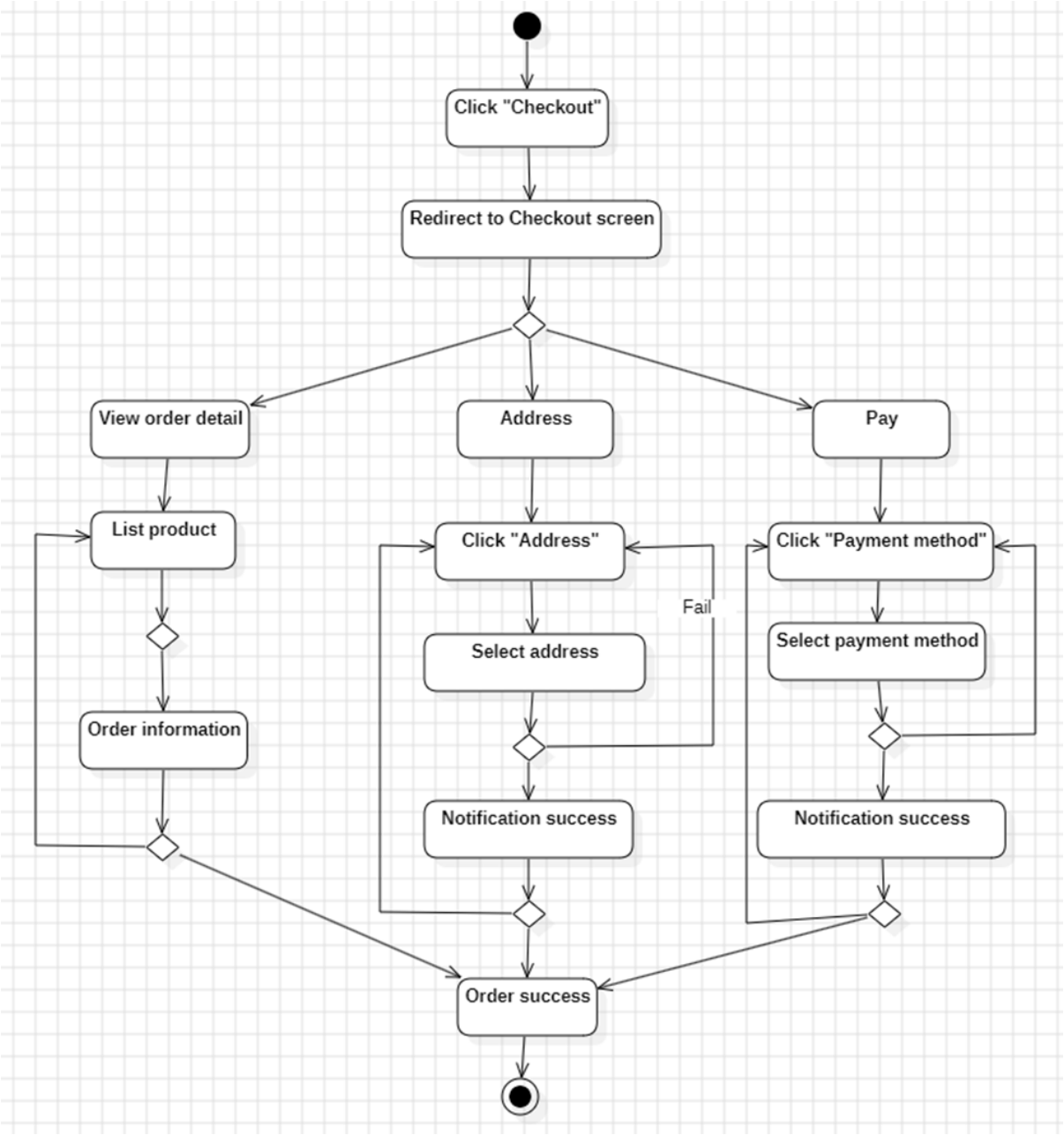

Figure 2.22 Activity "Order"

2.6 Sequential sketch

2.6.1 Register

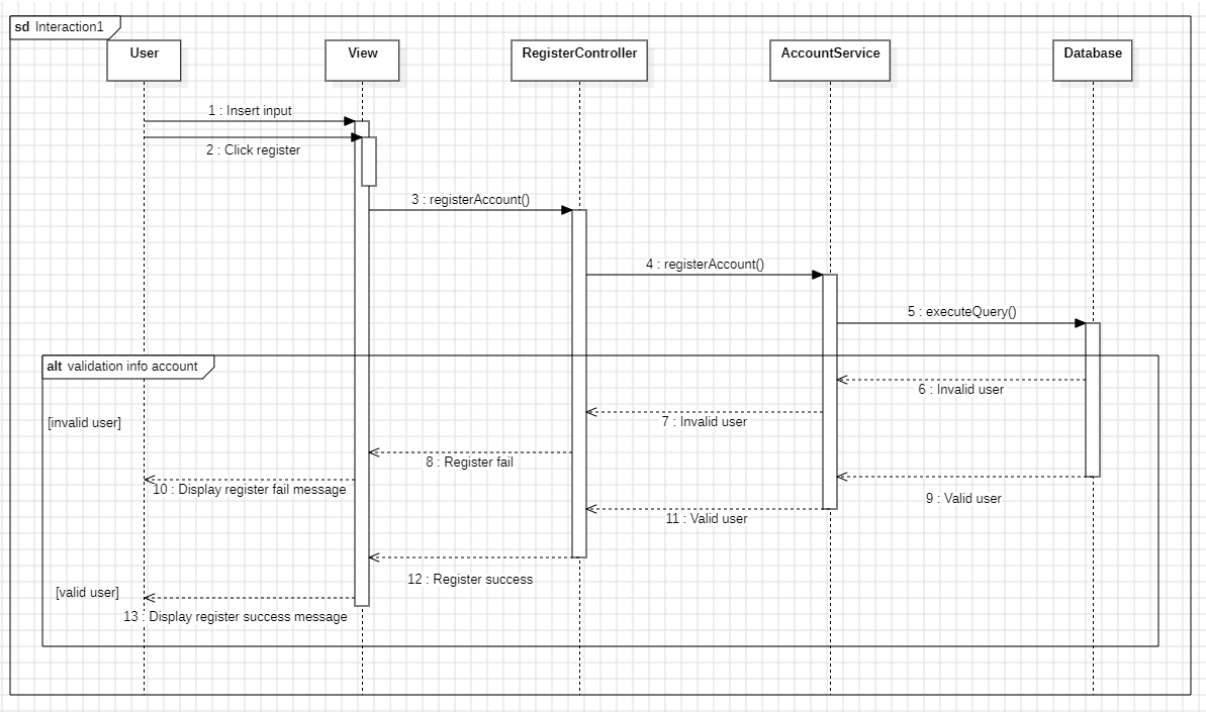

Figure 2.16 Sequential Sketch Register

2.6.2 Login

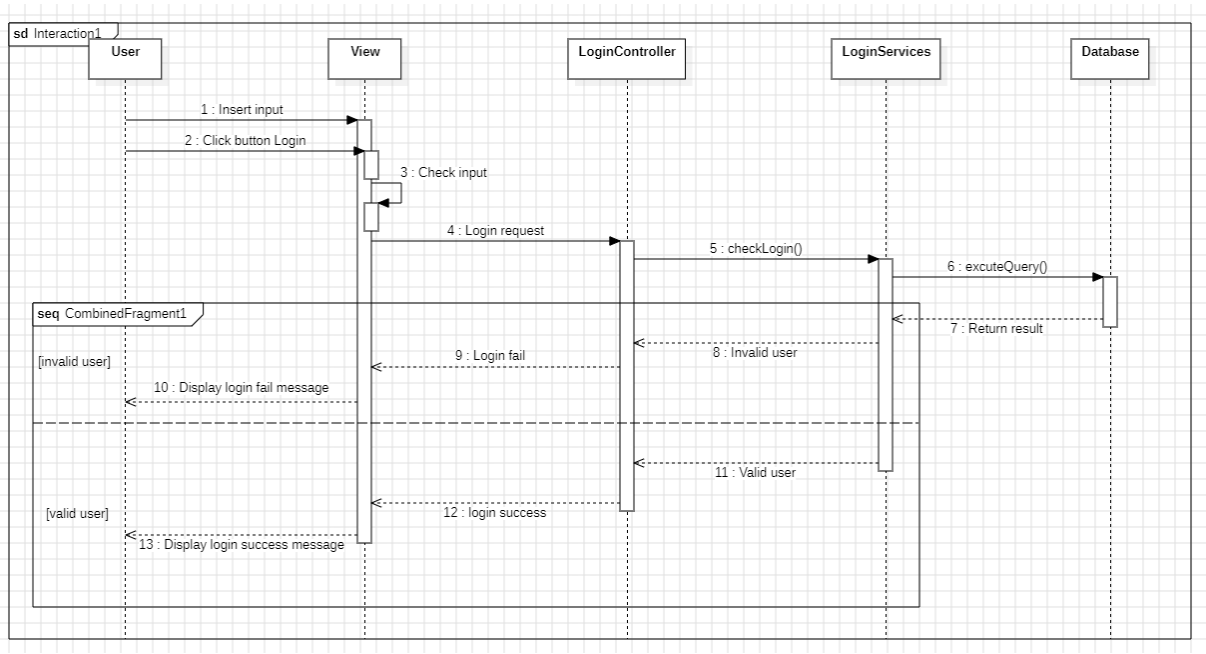

Figure 2.17 Sequential Sketch Login

2.6.3 Product

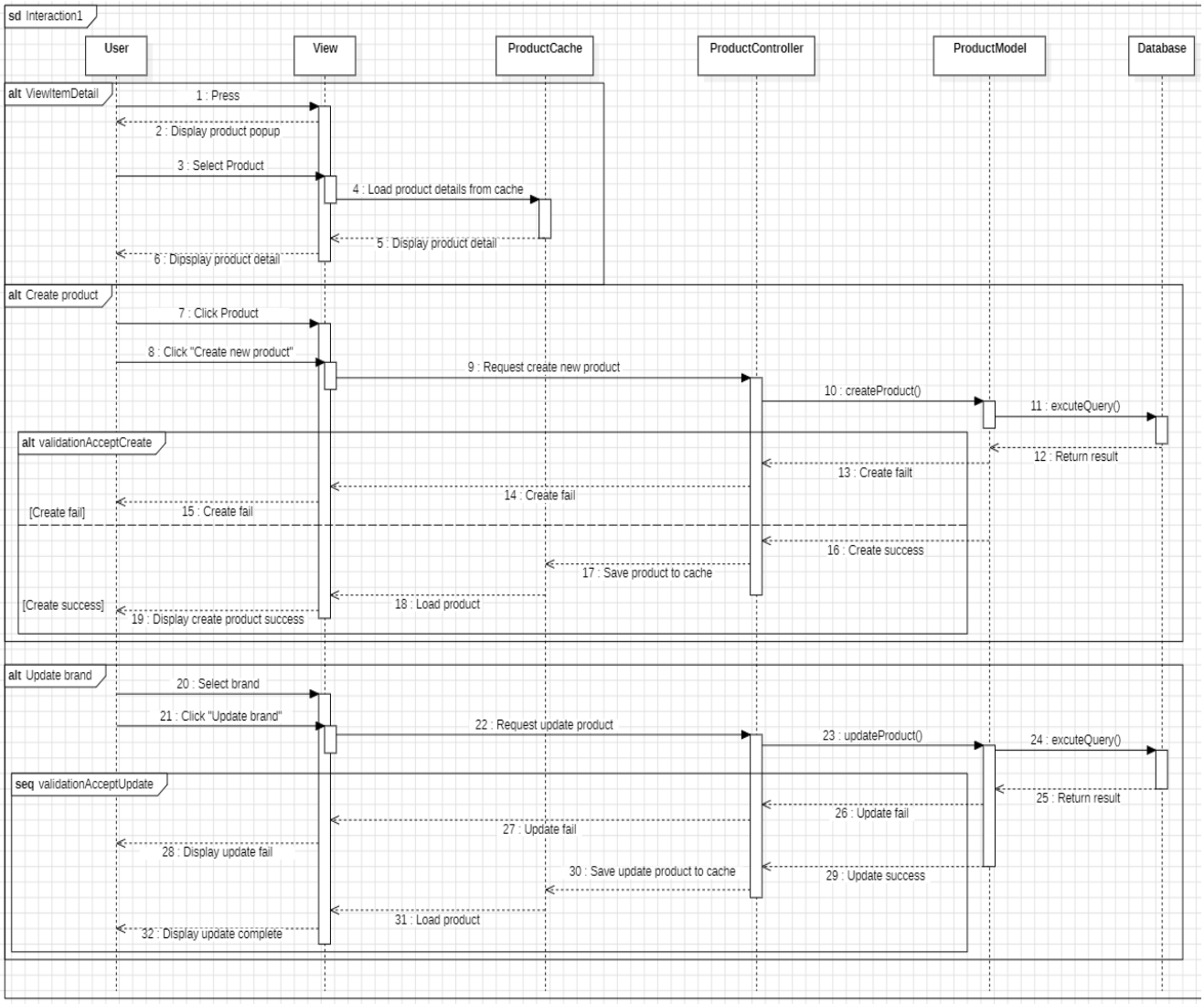

Figure 2.18 Sequential sketch Product

2.6.4 Banner

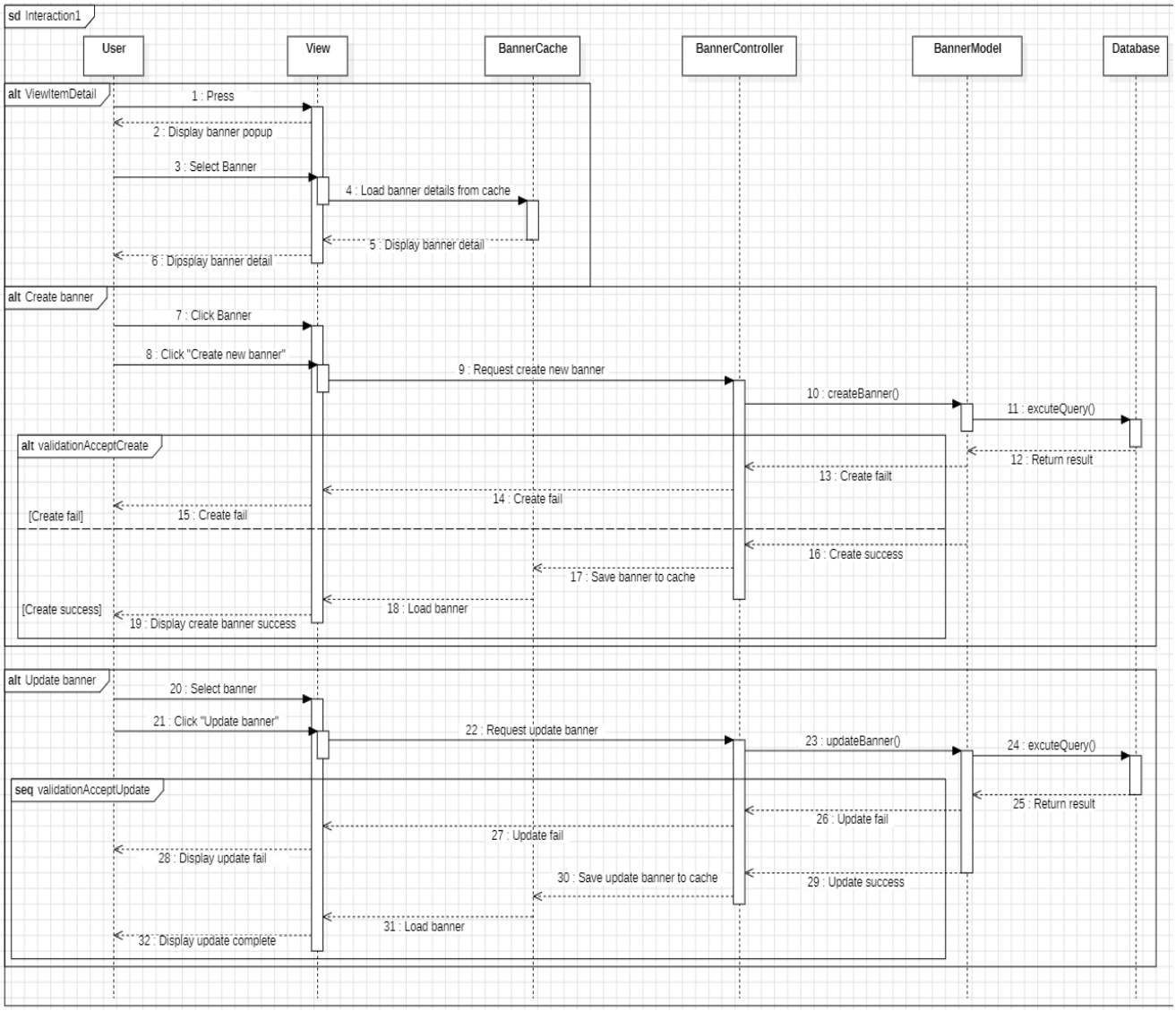

Figure 2.19 Sequential sketch Banner

2.6.5 Brand

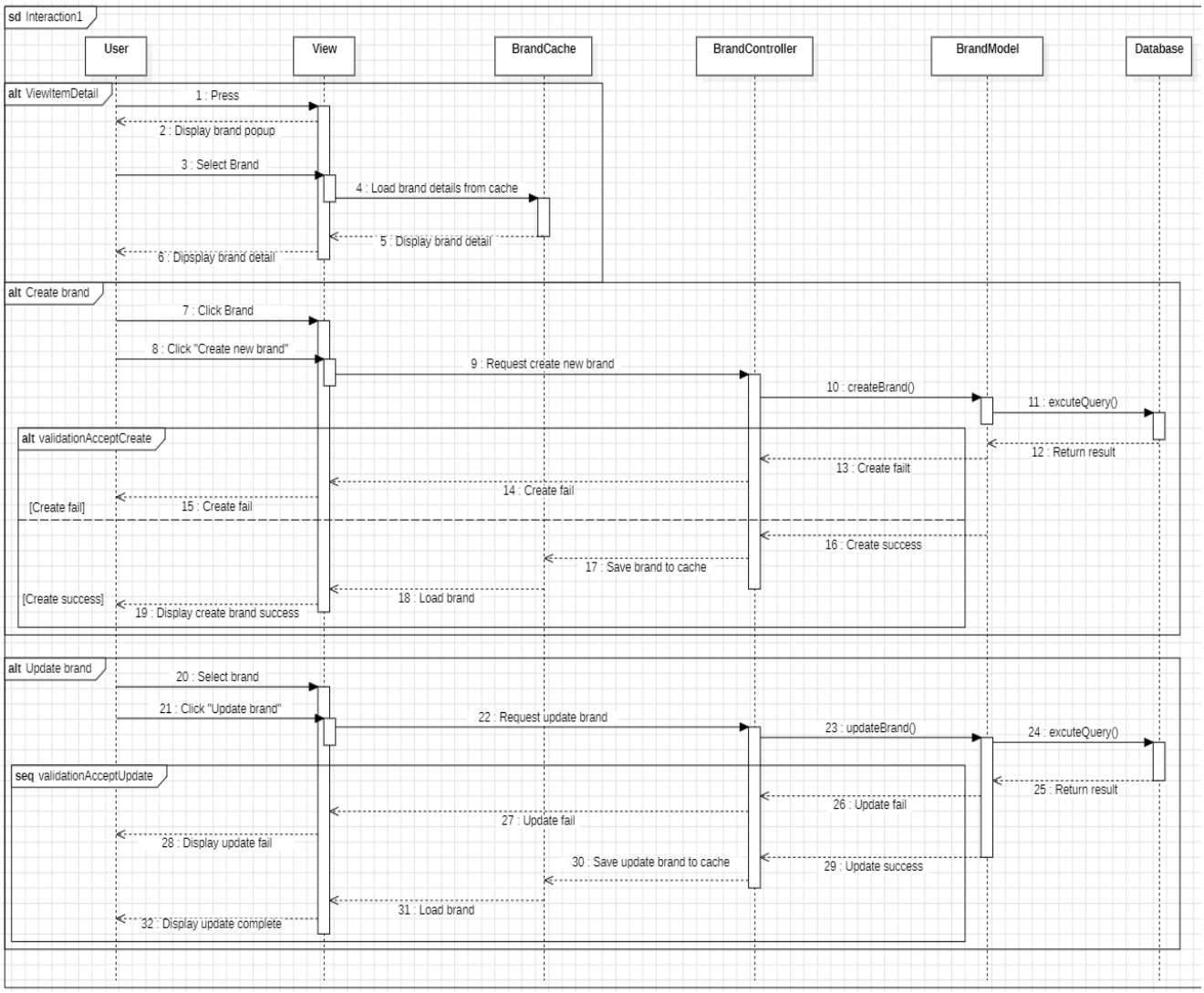

Figure 2.20 Sequential sketch Brand

2.6.6 Category

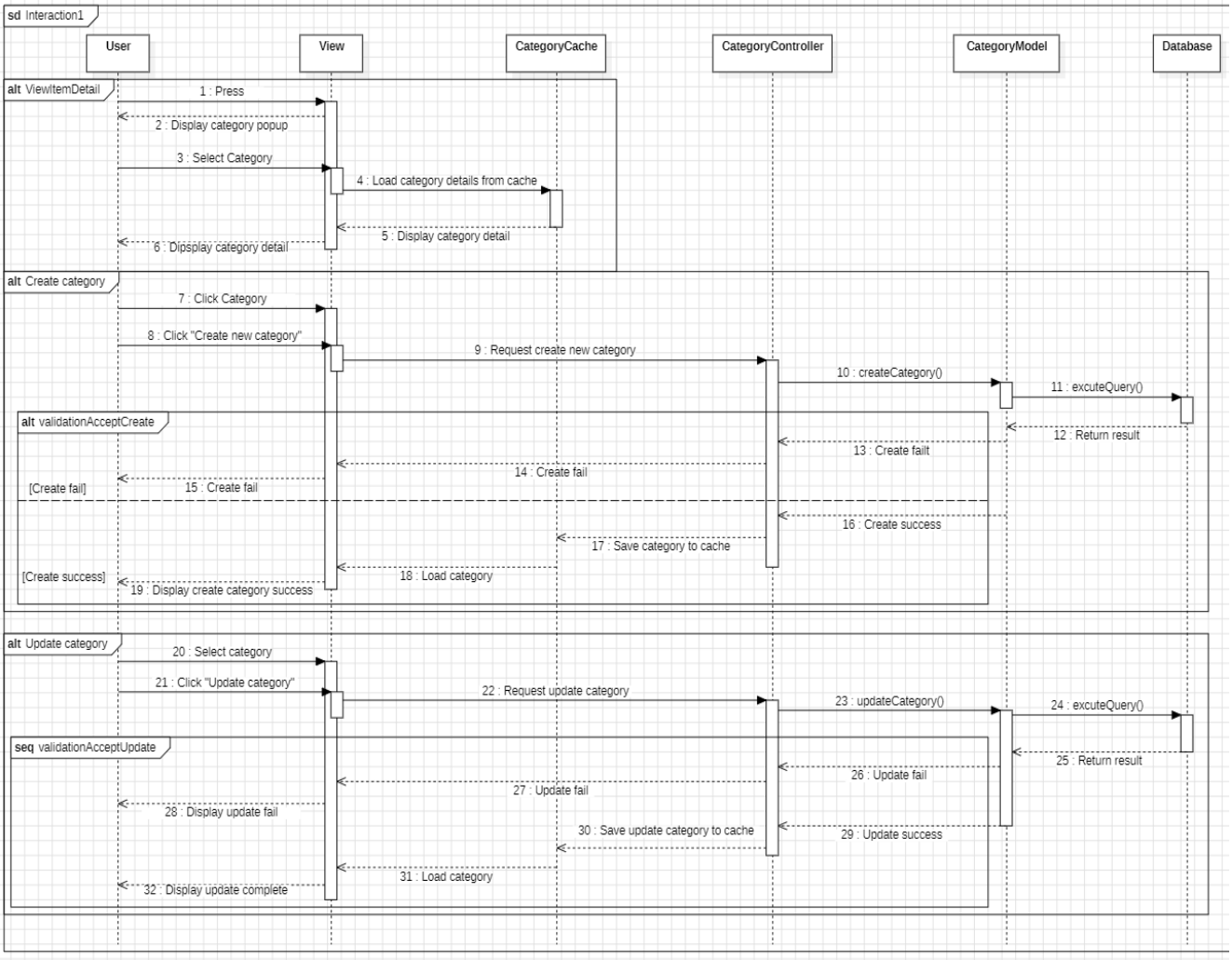

Figure 2.21 Sequential Sketch Category

2.6.7 Order

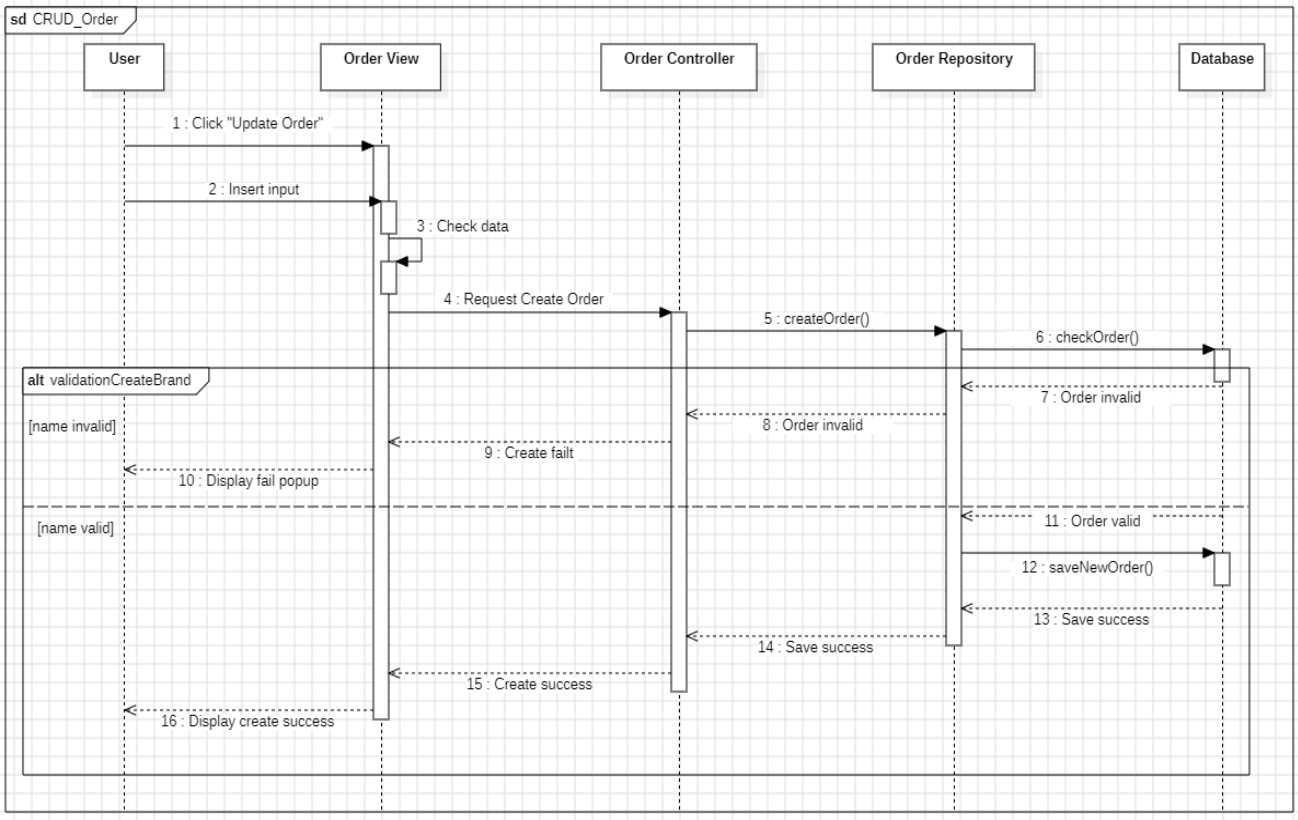

Figure 2.22 Sequential sketch Order

2.6.8 Media

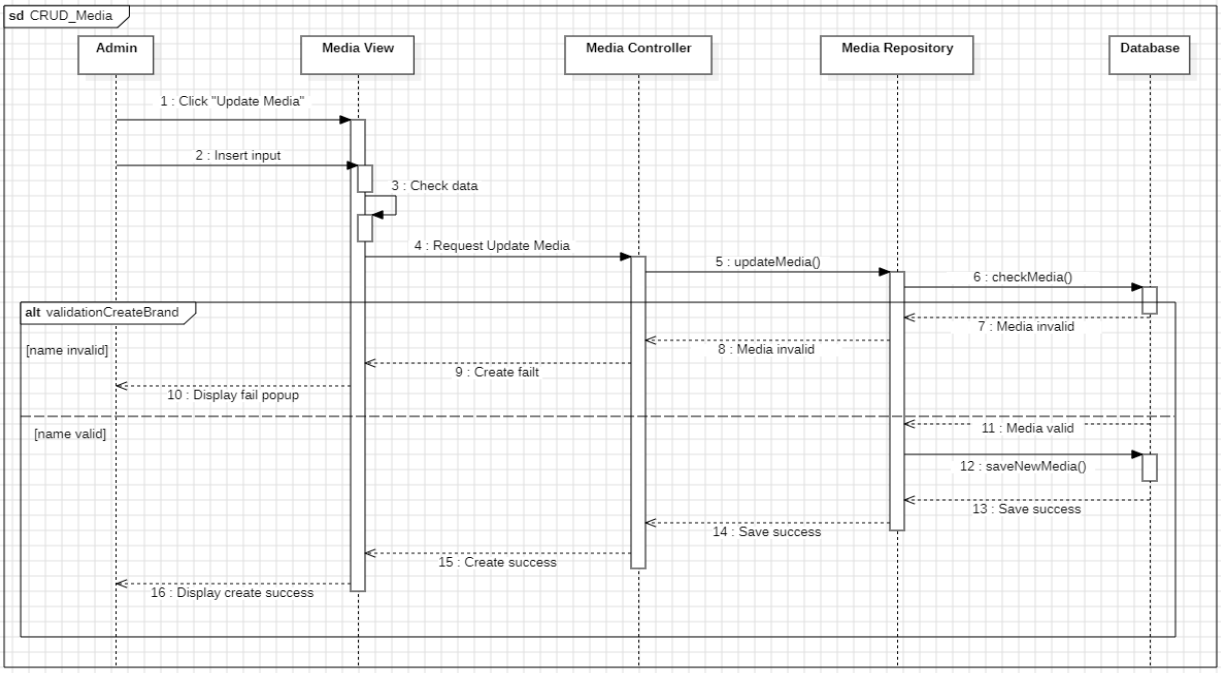

Figure 2.23 Sequential sketch Media

2.6.9 Shopping cart Management

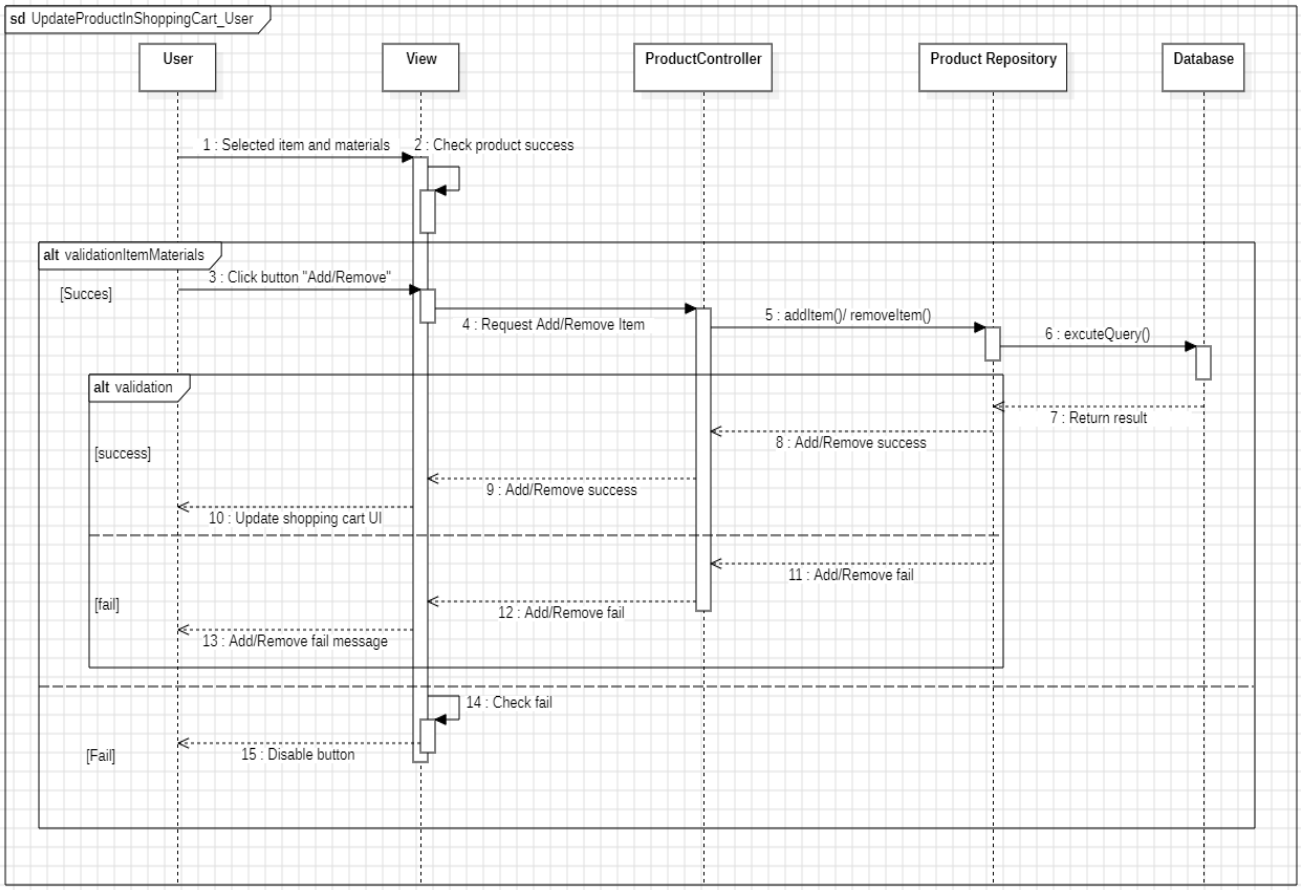

Figure 2.24 Sequential Sketch Shopping Cart

2.6.10 User Management

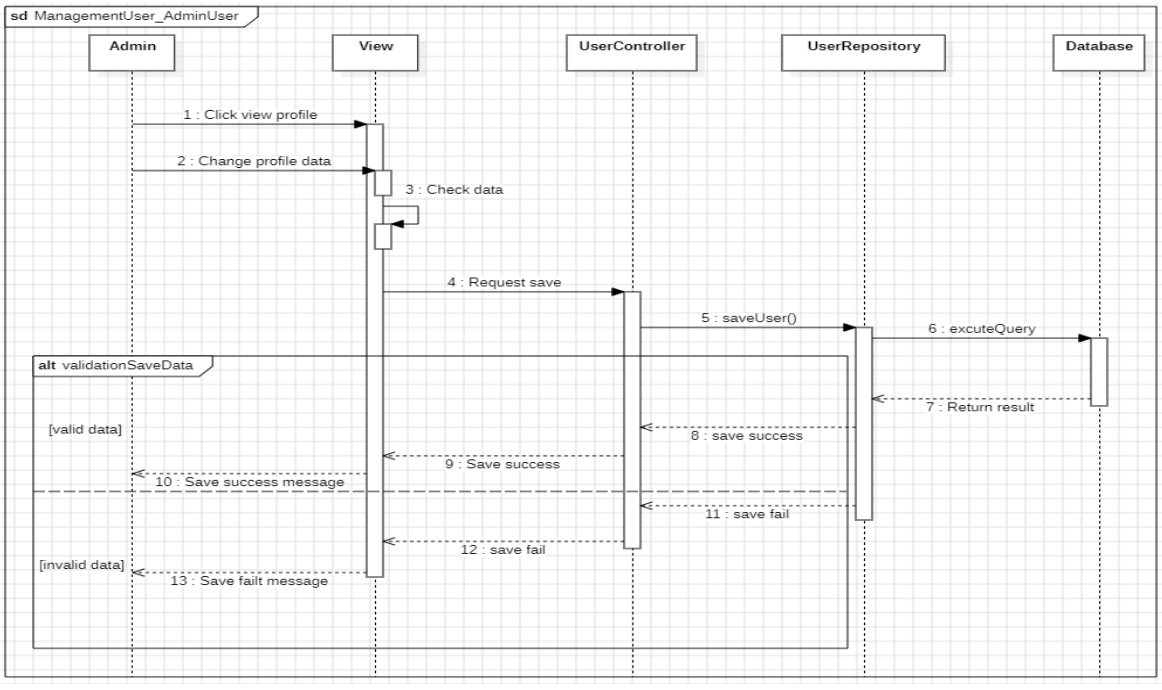

Figure 2.25 Sequential Sketch User

2.7 Class diagram

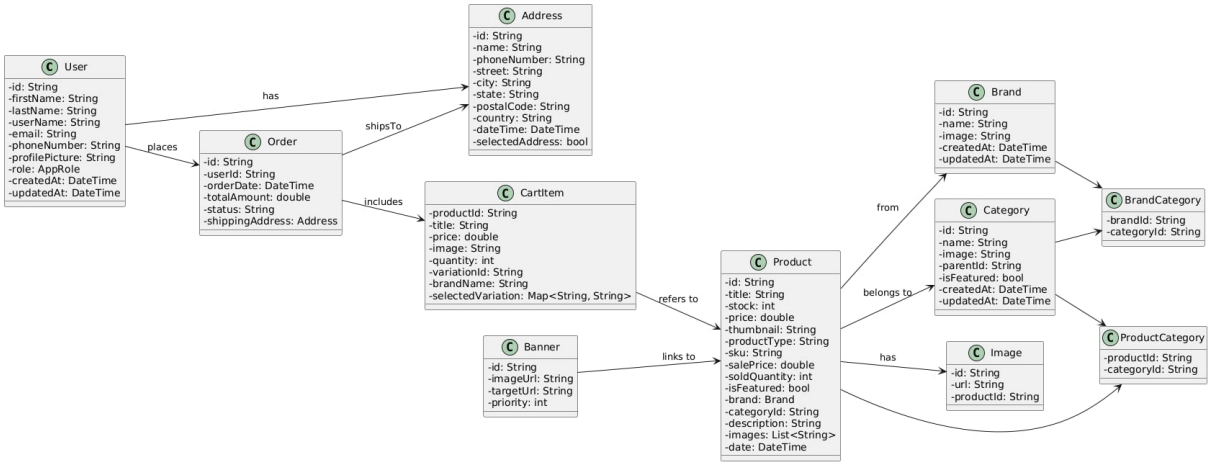

Figure 2.26 Class Diagram

2.8 Database Table Design

2.8.1 User

Table 2.13 "User" database

Attribute	Describe	Type data	Note
id	User unique identifier	String?	Optional
firstName	User first name	String	
lastName	User last name	String	
userName	User username	String	
email	User email address	String	Required
phoneNumber	User phone number	String	
profilePicture	User profile picture URL	String	
role	User role in system	AppRole	Default: user
createdAt	Account creation date	DateTime?	Optional
updatedAt	Last update date	DateTime?	Optional
orders	User order history	List<OrderModel>?	Optional
addresses	User saved addresses	List<AddressModel>?	Optional

2.8.2 Address

Table 2.14 "Address" database

Attribute	Describe	Type data	Note
id	Address unique identifier	String	Required
name	Recipient name	String	Required
phoneNumber	Recipient phone number	String	Required
street	Street address	String	Required
city	City name	String	Required
state	State/Province	String	Required
postalCode	Postal/ZIP code	String	Required
country	Country name	String	Required
dateTime	Address creation date	DateTime?	Optional
selectedAddress	Is selected address	bool	Default: true

2.8.3 CartItem

Table 2.15 "CartItem" database

Attribute	Describe	Type data	Note
productId	Product identifier	String	Required
title	Product title	String	Default: ''
price	Product price	double	Default: 0.0
image	Product image URL	String?	Optional
quantity	Item quantity	int	Required
variationId	Product variation ID	String	Default: ''
brandName	Product brand name	String?	Optional
selectedVariation	Selected variation options	Map<String, String>?	Optional

2.8.4 Order

Table 2.16 "Order" database

Attribute	Describe	Type data	Note
id	Order unique identifier	String	Required
docId	Document ID in Firebase	String	Default: ''
userId	User identifier	String	Default: ''
status	Order status	OrderStatus	Required
totalAmount	Total order amount	double	Required
shippingCost	Shipping cost	double	Required
taxCost	Tax amount	double	Required
orderDate	Order creation date	DateTime	Required
paymentMethod	Payment method used	String	Default: 'Cash on Delivery'
shippingAddress	Shipping address	AddressModel?	Optional
billingAddress	Billing address	AddressModel?	Optional
deliveryDate	Expected delivery date	DateTime?	Optional
items	Order items list	List<CartItemModel>	Required
billingAddressSameAsShipping	Same billing and shipping address	bool	Default: true

2.8.5 Banner

Table 2.17 "Banner" database

Attribute	Describe	Type data	Note
id	Banner unique identifier	String?	Optional
imageUrl	Banner image URL	String	Required
active	Is banner active	bool	Required
targetScreen	Target screen navigation	String	Required

2.8.6

2.8.7 Product

Table 2.18 "Product" database

Attribute	Describe	Type data	Note
id	Product unique identifier	String	Required
title	Product title	String	Required
stock	Available stock quantity	int	Required
price	Product price	double	Required
thumbnail	Product thumbnail image	String	Required
productType	Product type classification	String	Required
sku	Stock Keeping Unit	String?	Optional
salePrice	Sale/discount price	double	Default: 0.0
soldQuantity	Total sold quantity	int	Default: 0
isFeatured	Is featured product	bool?	Optional
brand	Product brand	BrandModel?	Optional
categoryId	Category identifier	String?	Optional
description	Product description	String?	Optional
images	Product images list	List<String>?	Optional
date	Product creation date	DateTime?	Optional
productAttributes	Product attributes list	List<ProductAttributeModel>?	Optional
productVariations	Product variations list	List<ProductVariationModel>?	Optional

2.8.8 Category

Table 2.19 "Category" database

Attribute	Describe	Type data	Note
id	Category unique identifier	String	Required
name	Category name	String	Required

image	Category image URL	String	Required
parentId	Parent category ID	String	Default: ''
isFeatured	Is featured category	bool	Default: false
createdAt	Category creation date	DateTime?	Optional
updatedAt	Last update date	DateTime?	Optional

2.8.9 Brand

Table 2.20 "Brand" database

Attribute	Describe	Type data	Note
id	Brand unique identifier	String	Required
name	Brand name	String	Required
image	Brand logo/image URL	String	Required
isFeatured	Is featured brand	bool	Default: false
productsCount	Number of products	int?	Optional
createdAt	Brand creation date	DateTime?	Optional
updatedAt	Last update date	DateTime?	Optional
brandCategories	Brand categories list	List<CategoryModel>?	Not mapped

2.8.10Image

Table 2.21 "Image" database

Attribute	Describe	Type data	Note
id	Image unique identifier	String	Default: ''
url	Image download URL	String	Required
folder	Storage folder path	String	Required
filename	Image filename	String	Required

sizeBytes	File size in bytes	int?	Optional
mediaCategory	Media category type	String	Default: ''
fullPath	Full storage path	String?	Optional
createdAt	Image creation date	DateTime?	Optional
updatedAt	Last update date	DateTime?	Optional
contentType	File MIME type	String?	Optional
file	HTML File object	File?	Not mapped
isSelected	Selection state	RxBool	Not mapped
localImageToDisplay	Local image bytes	Uint8List?	Not mapped

2.8.11 ProductCategory

Table 2.22 "ProductCategory" database

Attribute	Describe	Type data	Note
id	Relation unique identifier	String	Default: ''
productId	Product identifier	String	Required
categoryId	Category identifier	String	Required

2.8.12 BrandCategory

Table 2.23 "BrandCategory" database

Attribute	Describe	Type data	Note
id	Relation unique identifier	String?	Optional
brandId	Brand identifier	String	Required
categoryId	Category identifier	String	Required

2.9 Interface design

2.9.1 User

2.9.1.1 Onboarding

When the user first installs and opens the application, the onboarding screen will appear. Onboarding consists of 3 screens, after clicking next on the 3rdonboarding screen, they will be redirected to the sign in screen.

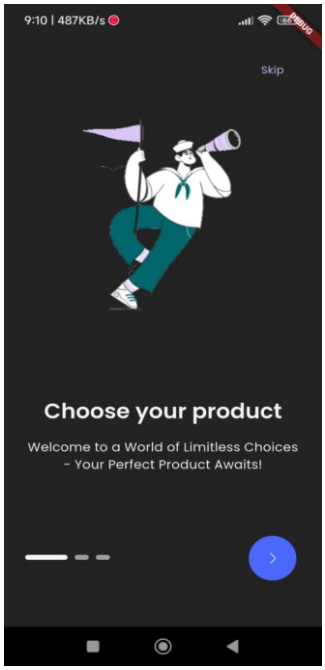

Figure 2.27 "Onboarding 1" interface

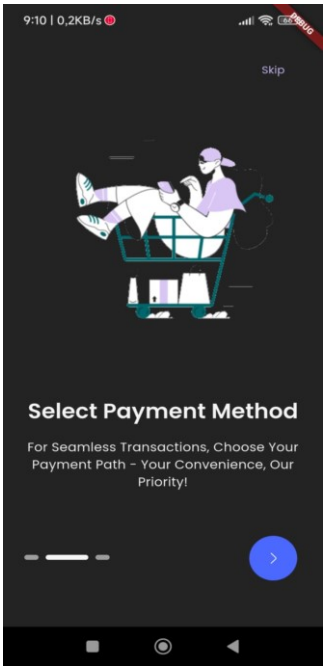

Figure 2.28 "Onboarding 2" interface

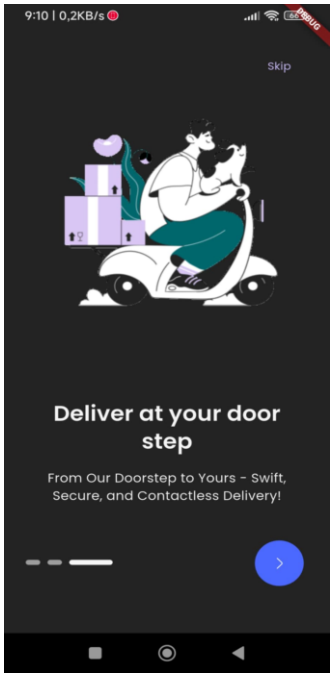

Figure 2.29 "Onboarding 3" interface

2.9.1.2 Login

In the login screen, if the user already has an account, they can enter their email and password to log in, or they can log in with their Google or Facebook account. If the user forgets their password, click "Forget password" to get a new password. Click "Remember me" if you want to save your account for future logins. If you don't have an account, click "Create account" to register for an account.

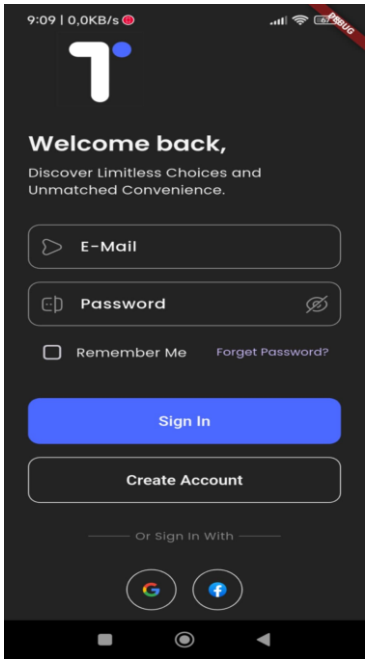

Figure 2.30 "Sign in" interface

Table 2.24 "Sign in" interface

Interface	Sign in		
Describe	The interface allows users to log in to use other functions.		
Access	User clicks on sign in button		
Interface content			
Item	**Type**	**Data**	**Describe**
Email	Textbox-String	N/A	Where to enter login email
Password	Textbox-String	N/A	Where to enter password
Remember Me	Button	N/A	Where to remember passwords
Sign in	Linked Button	N/A	Log in to the system
Forget Password	Linked Button	N/A	Leads to forgot password page

Create account	Linked Button	N/A	Leads to registration page
Work			
Name	**Describe**	**Success**	**Failure**
No email entered	Do not enter email then press sign in button		"Invalid email address" message
No password entered	Do not enter password then press sign in button		"Password is required" message
Sign in with account and password	Process login to the system with the entered user name and password	Redirect to home screen	"Something went wrong. Please try again" message

2.9.1.3 Register

In the register screen, the user must enter all information in the correct format (firstname, lastname, username, email, phone number, password). Then must accept the terms of use. Next, click "Create account" to register an account. Or the user can log in directly with a google or facebook account without having to register.

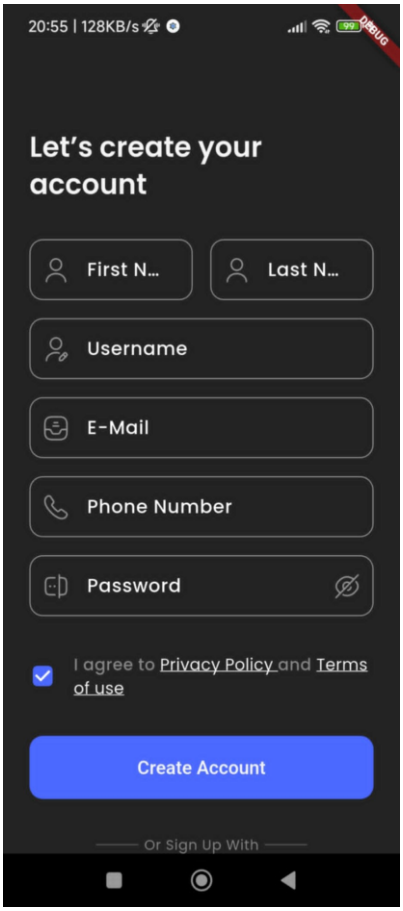

Figure 2.31 "Register 1" interface

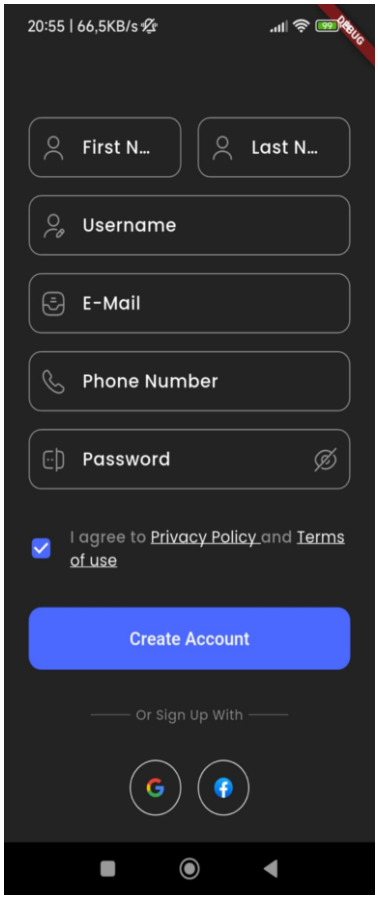

Figure 2.32 "Register 2" interface

Table 2.25 "Sign up" interface

Interface	Sign up		
Describe	Interface allows users to sign up for an account		
Access	User clicks on create account button		
Interface content			
Item	**Type**	**Data**	**Describe**
First name	Textbox-String	N/A	Where to enter first name
Last name	Textbox-String	N/A	Where to enter last name
Username	Textbox-String	N/A	Where to enter login name
Email	Textbox-String	N/A	Where to enter email
Password	Textbox-String	N/A	Where to enter password
Phone Number	Textbox-String	N/A	Where to enter phone number
Check box	Button	N/A	Accept the policy
Create Account	Linked Button	N/A	Leads to registration page
Google sign in	Linked Button	N/A	Log in with account Google
Work			
Name	**Describe**	**Success**	**Failure**
No first name entered	Do not enter first name then press create account		"First name is required" message
No last name entered	Do not enter last name then press create account		"Last name is required" message
No username entered	Do not enter username then press create account		"Username is required" message
No email entered	Do not enter email then press create account		"Invalid email address" message
No phone number entered	Do not enter phone number then press create account		"Phone number is required" message
No password entered	Do not enter password then press create account button		"Password is required" message
No agree policy	Do not click agree checkbox		
Sign up with Google account	Sign in app with Google account	Redirect to login screen	"Something went wrong. Please try again" message

2.9.1.4 Forget Password

If you forget your password, user can click "Forget password" in the login screen. Then enter your email, check gmail and reset new password.

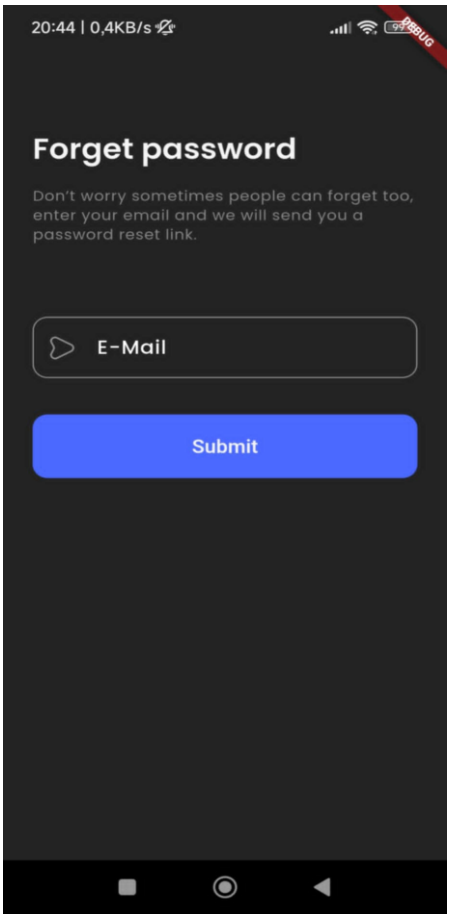

Figure 2.33 "Forget Password" interface

Table 2.26 "Forget" interface

Interface	Forget password		
Describe	Allow users to update their passwords		
Access	User clicks on "Forget password" button in sign in screen		
Interface content			
Item	**Type**	**Data**	**Describe**
Email	Textbox-String	N/A	Fill in registered email
Submit	Linked Button	N/A	Confirm email
Work			
Name	**Describe**	**Success**	**Failure**

Click Email	Fill in registered email	Select the email of the account whose password you want to reset.	
Click Favourite icon	Confirm email	Password reset notification sent to email	

2.9.1.5 Home

In the home interface, user can see products in categories, popular products, advertising banners. User can also click on the shopping cart to see product details

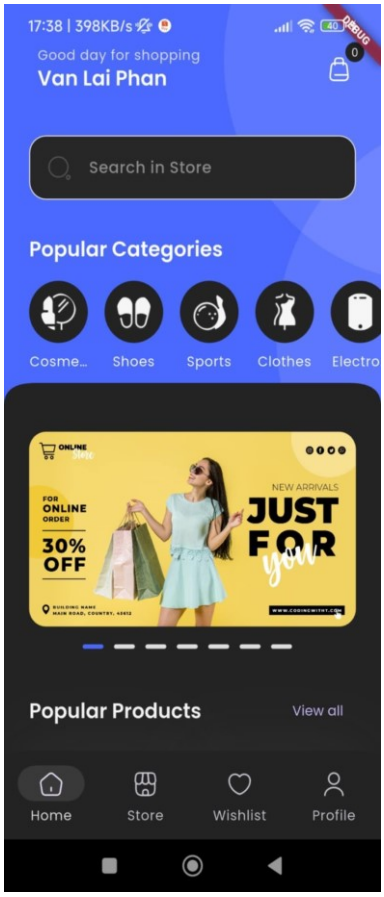

Figure 2.34 "Home 1" interface

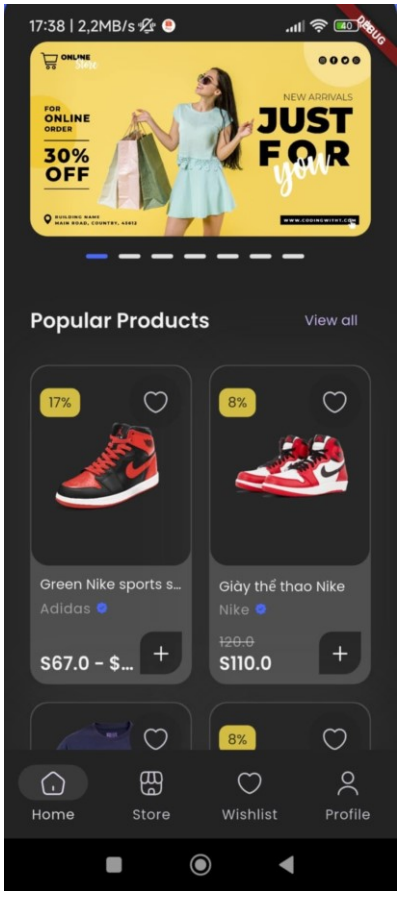

Figure 2.35 "Home 2" interface

Table 2.27 "Home" interface

Interface	Home		
Describe	The interface allows users to access the app to use other functions.		
Access	User clicks on "Sign in" button		
Interface content			
Item	**Type**	**Data**	**Describe**

Text heading	Text	N/A	Greeting and username
Shopping cart	Linked Button	N/A	Show cart details
Search	Textbox-String	N/A	Search for products and brands
Popular Category	Linked Button	N/A	Featured categories
Banner	Linked Button	N/A	Navigate to the screens
Popular Product	Linked Button	N/A	Featured products

Work

Name	**Describe**	**Success**	**Failure**
Click shopping cart	View products in shopping cart	Redirect to shopping cart screen	
Click search	Search for products and brands	Redirect to search screen	
Banner roll	Scroll the banner left or right	View other banners	
Click “View all” button in popular products	View all product	Redirect to all product screen	

2.9.1.6 Shopping cart

When the user clicks on the shopping cart icon on the home screen, the shopping cart details can be viewed. The user can add, reduce or delete products in the shopping cart. If the user wants to pay for the order, click "Checkout"

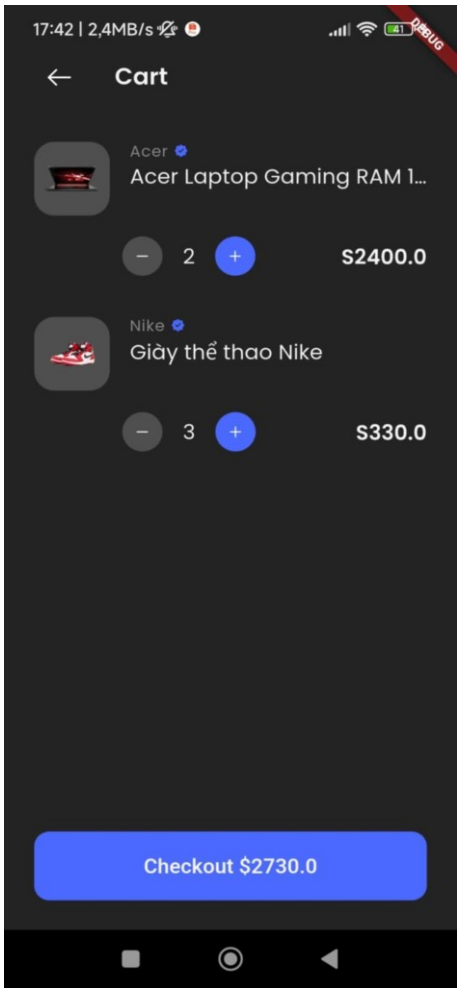

Figure 2.36 "Register" interface

Table 2.28 "Shopping cart" interface

Interface	Shopping cart		
Describe	Allows users to manage items in their shopping cart		
Access	User clicks on "Shopping cart" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Cart
Product	List	N/A	List of items in shopping cart

Check out	Linked Button	N/A	Redirect to order screen
Work			
Name	**Describe**	**Success**	**Failure**
Click - button	Remove product quantity	Each time you press the products quantity decreases by 1	
Click + button	Add product quantity	Each time you press the product quantity increases by 1	
Click Check out button	Pay for the product	Redirect to checkout screen to check information	

2.9.1.7 Checkout

In the checkout screen, users can view the details of the order they want to pay for (fees, prices, product quantities). They can choose the payment method (Paypal, Google pay, Apple pay, Master card, Visa) and address. Users can check the information before purchasing.

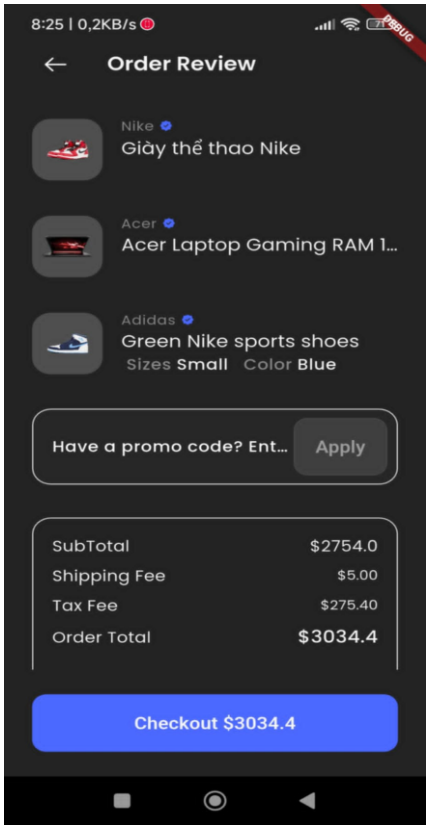

Figure 2.37 "Checkout 1" interface

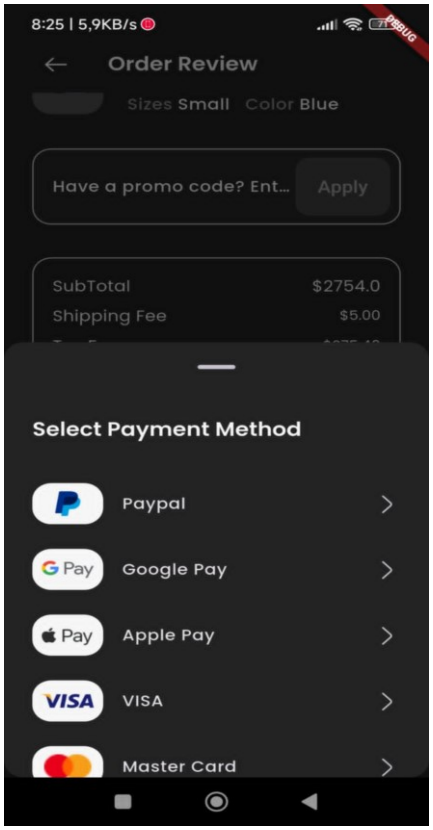

Figure 2.38 "Checkout 2" interface

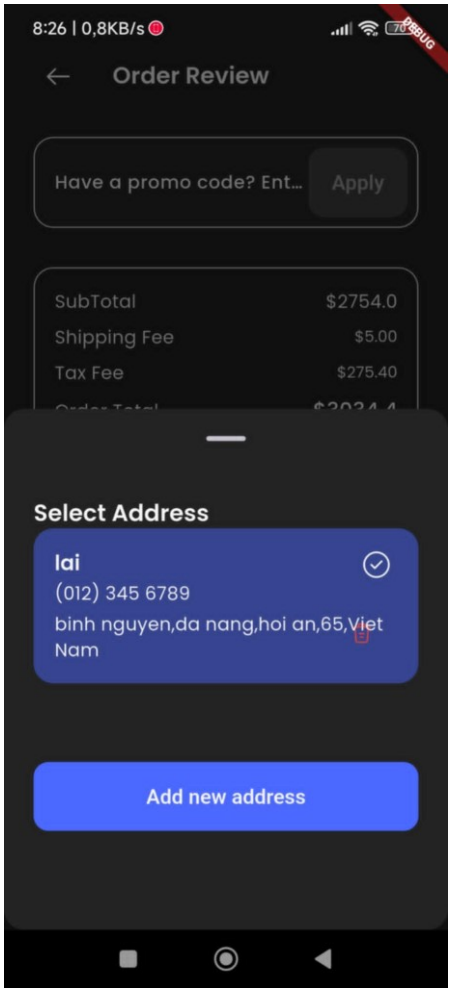

Figure 2.39 "Checkout 3" interface

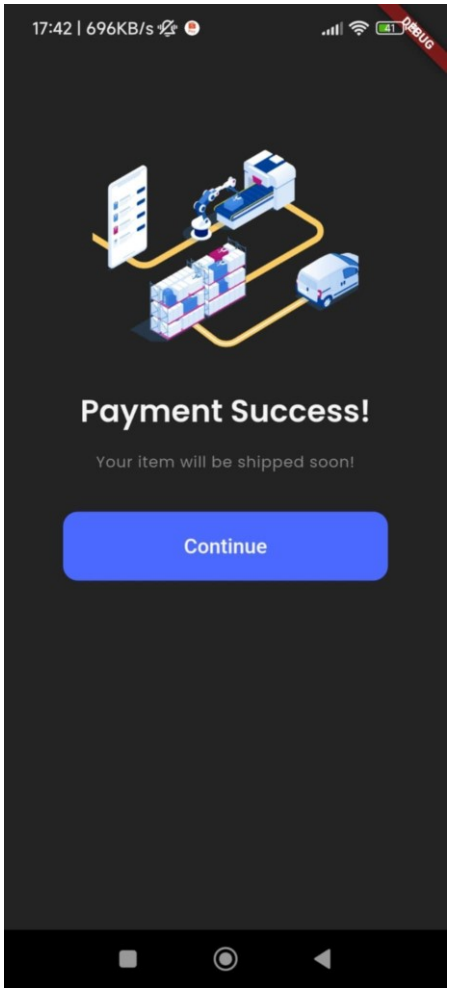

Figure 2.40 "Checkout 4" interface

Table 2.29 "Checkout" interface

Interface	Checkout		
Describe	Allows users to checkout product		
Access	User clicks on "Checkout" button in shopping cart		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Order Review
Product	List	N/A	List of product in shopping cart
Coupon	Textbox-String	N/A	Redirect to order screen
Bill	Text	N/A	Invoice information
Payment method	List	N/A	Choose payment method

Address	List	N/A	Choose address or create address
Checkout	Linked Button	N/A	Redirect to the order success screen
Work			
Name	**Describe**	**Success**	**Failure**
Click change payment method button	Choose payment method	Successfully change the payment method	
Click change address	Choose address or create address	Successfully change the payment method or create new address	
Click Check out button	Pay for the product	Redirect to the order success screen	

2.9.1.8 Search

When the user clicks on the search box in the home screen or store, they can search for products or brands.

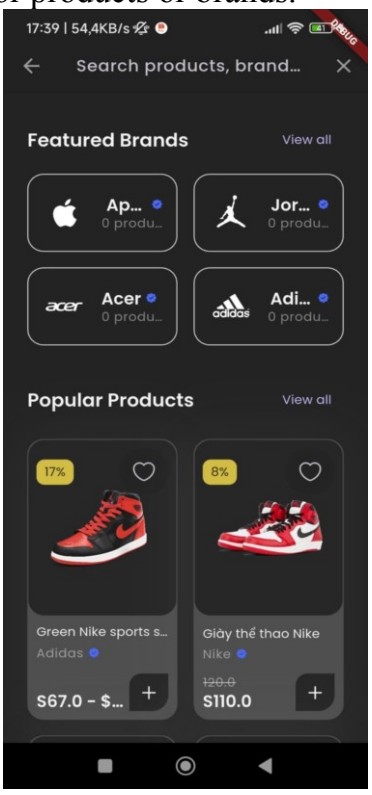

Figure 2.41 "Search 1" interface

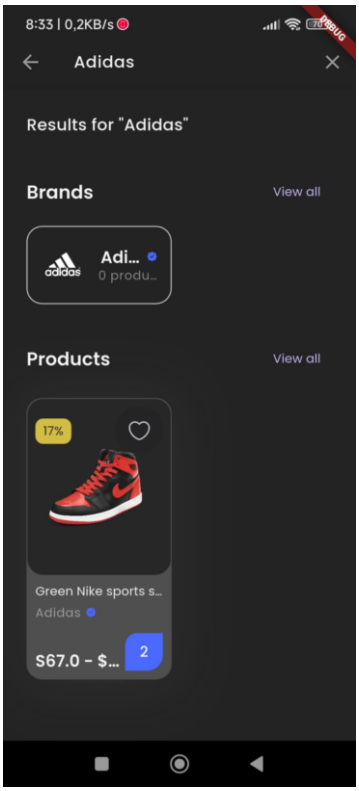

Figure 2.42 "Search 2" interface

Table 2.30 "Search" interface

Interface	Search		
Describe	Allows users to search product or brand		
Access	User clicks on "Search" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Search box	Textbox-String	N/A	Enter the product or brand name you want to search for
Brand	List	N/A	List of featured brands
Product	List	N/A	List of Popular products
Work			
Name	**Describe**	**Success**	**Failure**
Click search box	Enter the product or brand name you want to search for	Display suggestions by product name or brand you want to search for	
Click brand icon button	Choose brand	Redirect to brand detail screen	
Click "View all" button in featured brands	View all brand	Redirect to all brand screen	
Click icon product button	Choose product	Redirect to product detail screen	
Click "View all" button in popular products	View all product	Redirect to all product screen	

2.9.1.9 Category

In the category screen, the user can see all the products in that category

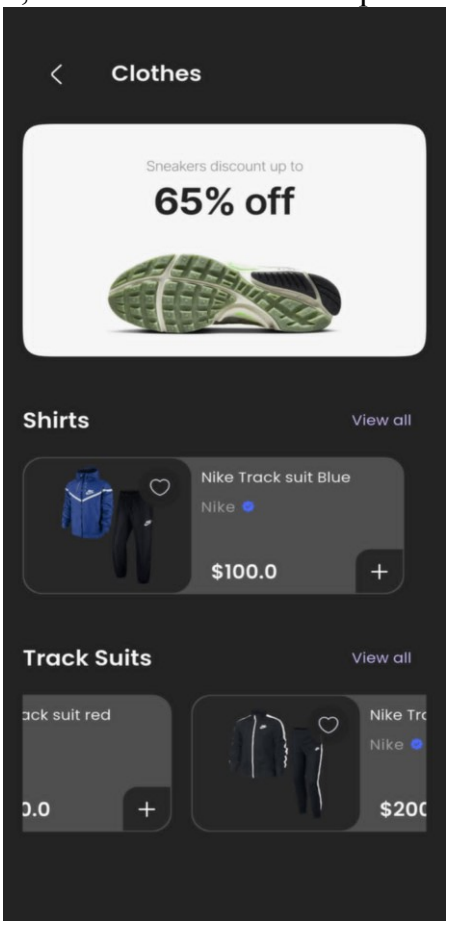

Figure 2.43 "Category" interface

Table 2.31 "Category" interface

Interface	Category		
Describe	Allows users to view detail category		
Access	User clicks on "Category icon" button in home		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of category
Banner	Image	N/A	Banner for category
Child category	List	N/A	List of child category
Work			
Name	**Describe**	**Success**	**Failure**
Click icon product button	Choose product	Redirect to product detail screen	
Click "View all" button in child category	View all product	Redirect to all product screen	

2.9.1.10 Product

In the product detail screen, users can see product images, price, status, brand, variations such as color, material... Users can add products to cart.

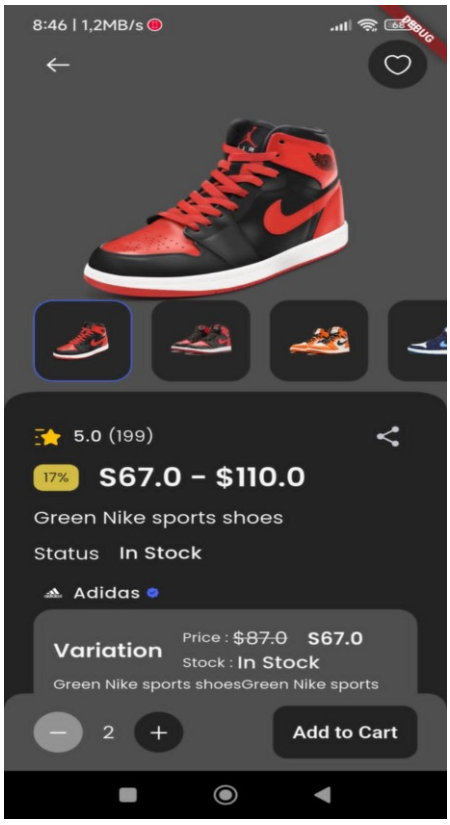

Figure 2.44 "Product 1" interface

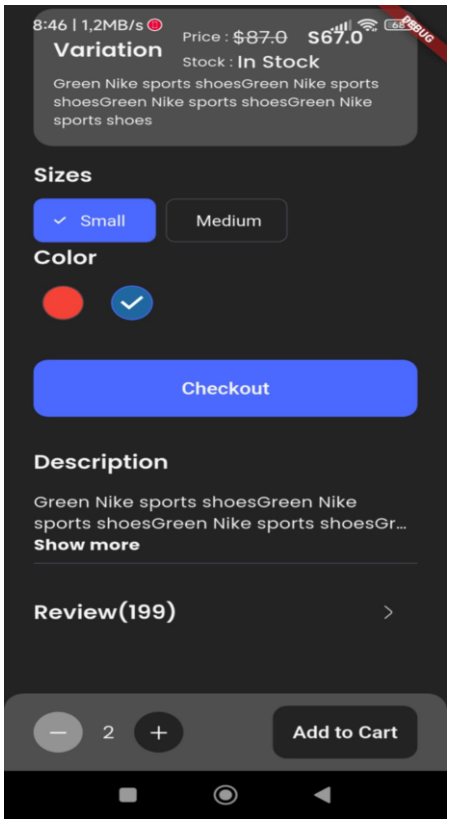

Figure 2.45 "Product 2" interface

Table 2.32 "Product" interface

Interface	Product		
Describe	Allows users to view detail product		
Access	User clicks on "product icon" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Image product	Image	N/A	Banner for category
Thumbnail image	Image	N/A	List of image product
Favourite icon	Button	N/A	Add product to wishlist
Price	Number	N/A	Price of the product
Name product	Text	N/A	Name of product
Status	Text	N/A	Status of product: In stock or out stock
Brand	Text	N/A	Brand of product

Variation	Text	N/A	Variation of product : Size, Color, Material …
Checkout button	Linked Button	N/A	Redirect to checkout screen
Description	Text	N/A	Description for product
Review	Linked Button	N/A	Redirect to review screen
Add, Remove button	Button	N/A	Increase or decrease the quantity of products
Add to cart button	Linked Button	N/A	Add products to shopping cart

Work

Name	**Describe**	**Success**	**Failure**
Click thumbnail image	View product thumbnails	Display thumbnail image	
Click favourite icon	Add product to wishlist	The product were added to wishlist	
Click variable	Choose variation of product	Display price, status, description of variation	

2.9.1.11 All Product

In the product detail screen, users can see product images, price, status, brand, variations such as color, material... Users can add products to cart.

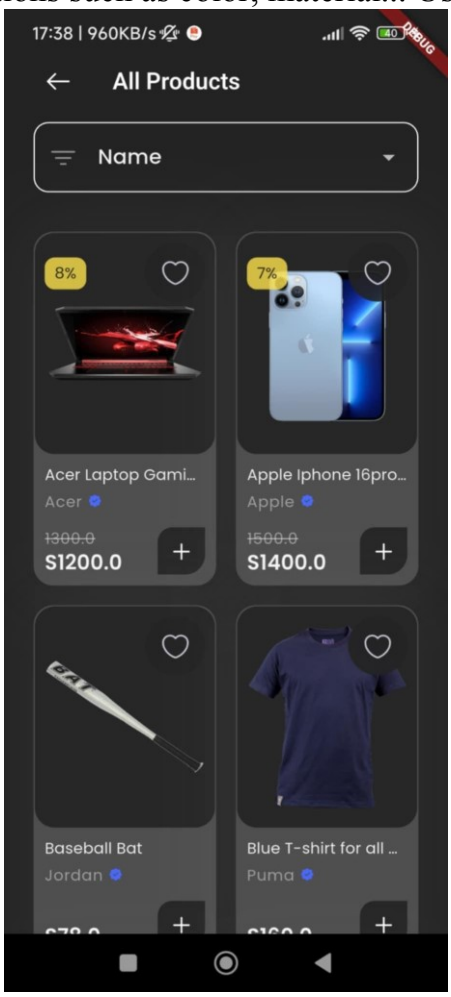

Figure 2.46 "All Product 1" interface

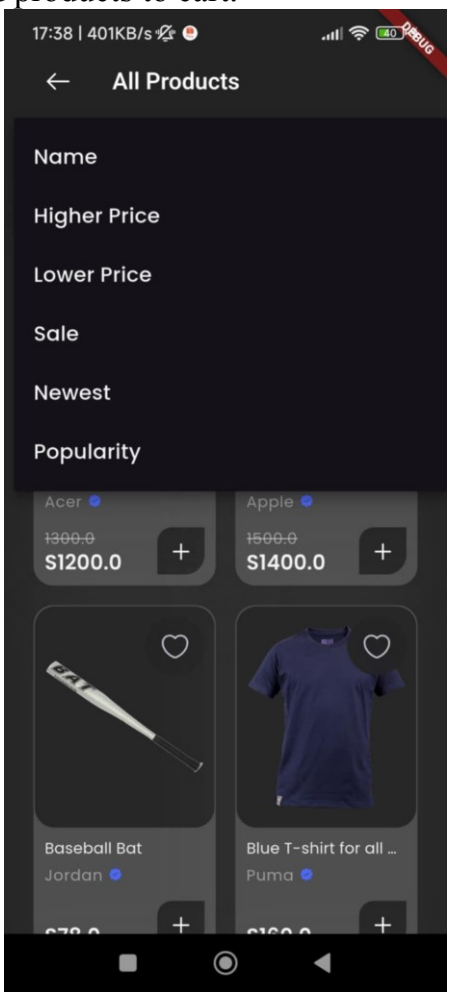

Figure 2.47 "All Product 2" interface

Table 2.33 "All Product" interface

Interface	All Product		
Describe	Allows users to view all product		
Access	User clicks on "view all" button in popular products		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is All Products
Sortable product	List	N/A	Choose ways to arrange products
Product	List	N/A	List of all product
Work			

Name	Describe	Success	Failure
Click sortable product	Choose ways to arrange products	The products are sorted by selection	
Click product	View detail product	Redirect to product detail screen	

2.9.1.12 Brand

In the brand screen, the user can see all products belonging to the brand.

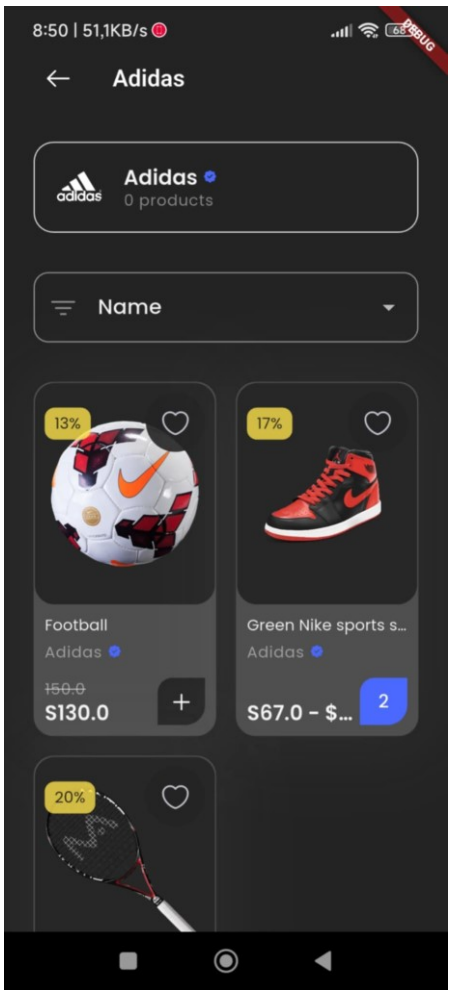

Figure 2.48 "Brandt 1" interface

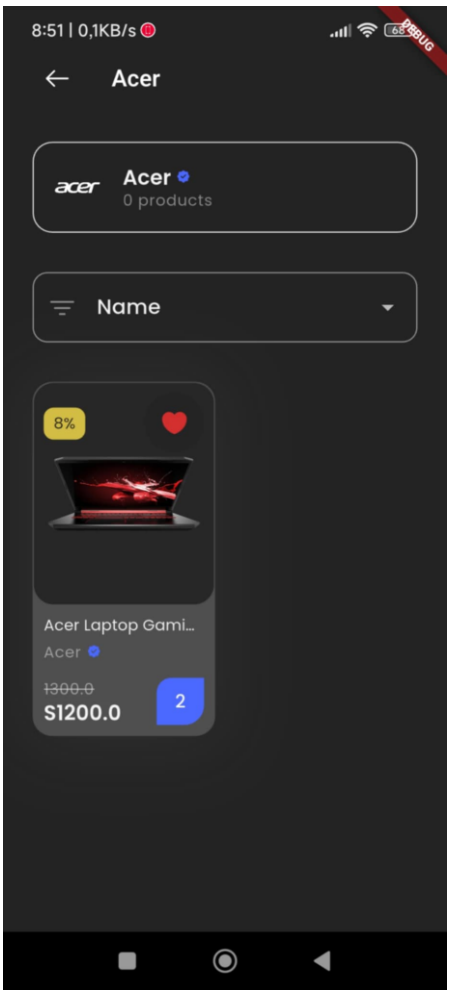

Figure 2.49 "Brand 2" interface

Table 2.34 "Brand" interface

Interface	Brand		
Describe	Allows users to view brand detail		
Access	User clicks on "icon brand" button in store screen		
Interface content			
Item	**Type**	**Data**	**Describe**

Name brand	Text	N/A	Name of brand
Sortable product	List	N/A	Choose ways to arrange products
Product	List	N/A	List of products included in the brand
Work			
Name	**Describe**	**Success**	**Failure**
Click sortable product	Choose ways to arrange products	The products are sorted by selection	
Click product	View detail product	Redirect to product detail screen	

2.9.1.13 All Brand

Users can view all brand in the store

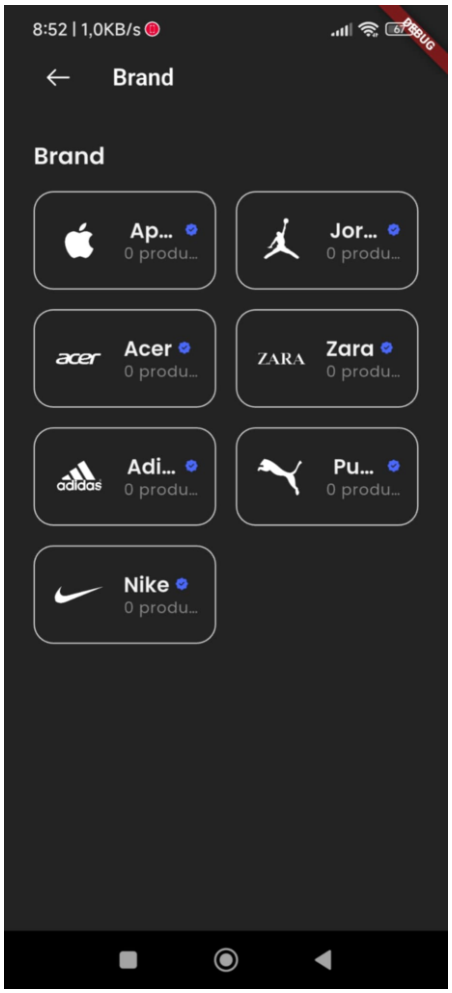

Figure 2.50 "All Brand" interface

Table 2.35 "All Product" interface

Interface	All Product		
Describe	Allows users to view all product		
Access	User clicks on "view all" button in popular products		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is All Brand
Brand	List	N/A	List of all brand
Work			
Name	**Describe**	**Success**	**Failure**
Click brand	View detail brand	Redirect to brand detail screen	

2.9.1.14 Store

In the store screen, users can search for products or brands. The store displays featured brands and products in each category.

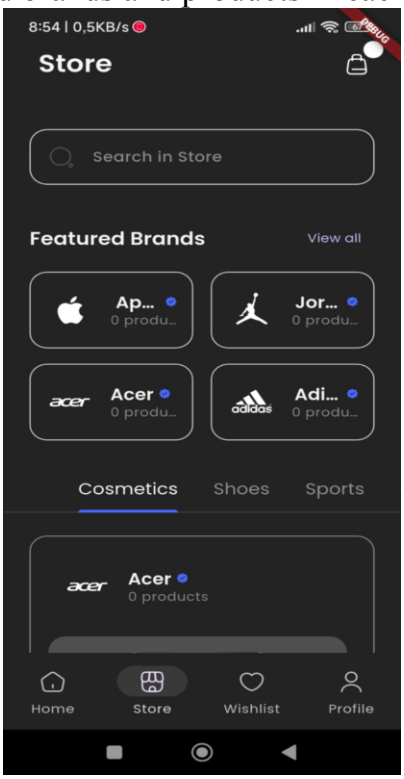

Figure 2.51 "Store 1" interface

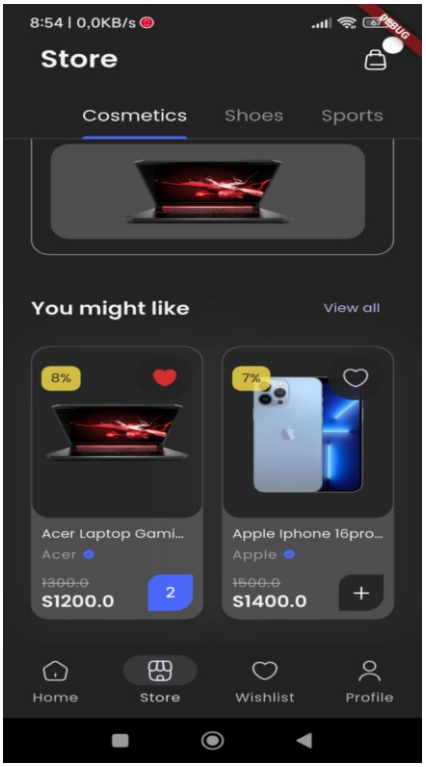

Figure 2.52 "Store 2" interface

Table 2.36 "Store" interface

Interface	Store		
Describe	The interface allows users to access the app to use other functions.		
Access	User clicks on "Store" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Store
Search box	Text	N/A	Enter the product or brand name you want to search for
Brand	List	N/A	List of featured brands
Category	List	N/A	List of featured categories
Product	List	N/A	List of product in category
Work			
Name	**Describe**	**Success**	**Failure**
Click search box	Enter the product or brand name you want to search for	Display suggestions by product name or brand you want to search for	
Click brand icon button	Choose brand	Redirect to brand detail screen	
Click "View all" button in featured brands	View all brand	Redirect to all brand screen	
Scroll and click category	Scroll to view the list of category	Brands and products featured of category	
Click icon product button	Choose product	Redirect to product detail screen	

2.9.1.15 Wishlist

In the wishlist screen, users can see the products they have liked. Click the favourite icon in the product to add or remove from the wishlist.

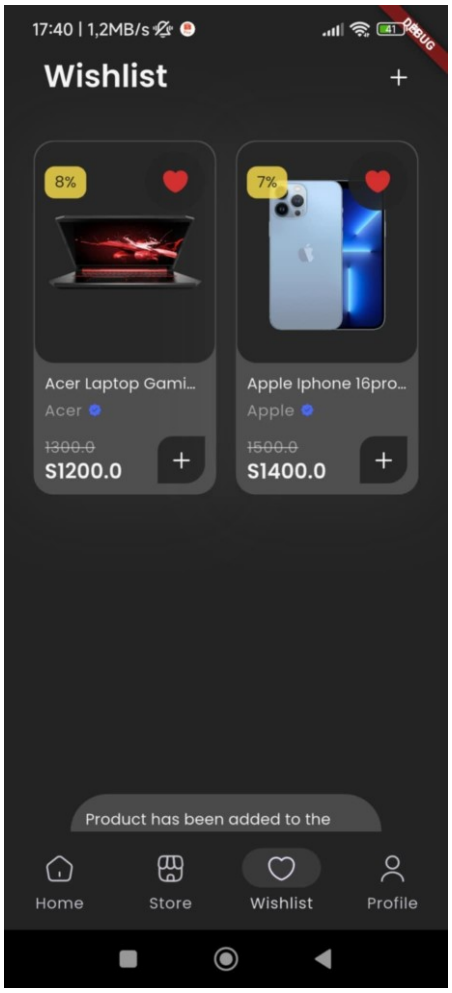

Figure 2.53 "Wishlist" interface

Table 2.37 "Wishlist" interface

Interface	Wishlist		
Describe	Allows users to view favourite products list		
Access	User clicks on "Wishlist" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Wishlist
Add icon	Linked Button	N/A	Choose more products
Product	List	N/A	List of all favourite product

Work			
Name	**Describe**	**Success**	**Failure**
Click Add button	Add products to the wishlist	Redirect to home screen to choose products	
Click Favourite icon	Remove product from wishlist	Remove product from wishlist	

2.9.1.16 Review & Ratings

In the store screen, users can search for products or brands. The store displays featured brands and products in each category.

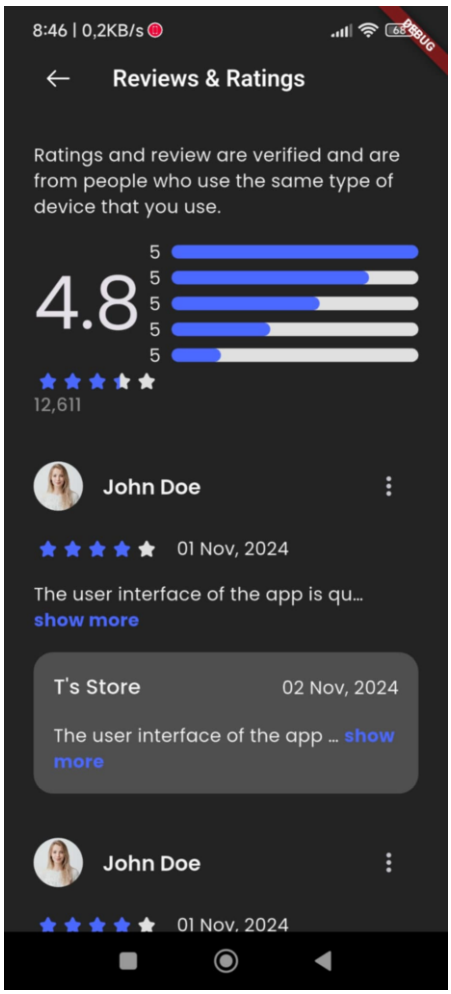

Figure 2.54 "Review & Ratings 1" interface

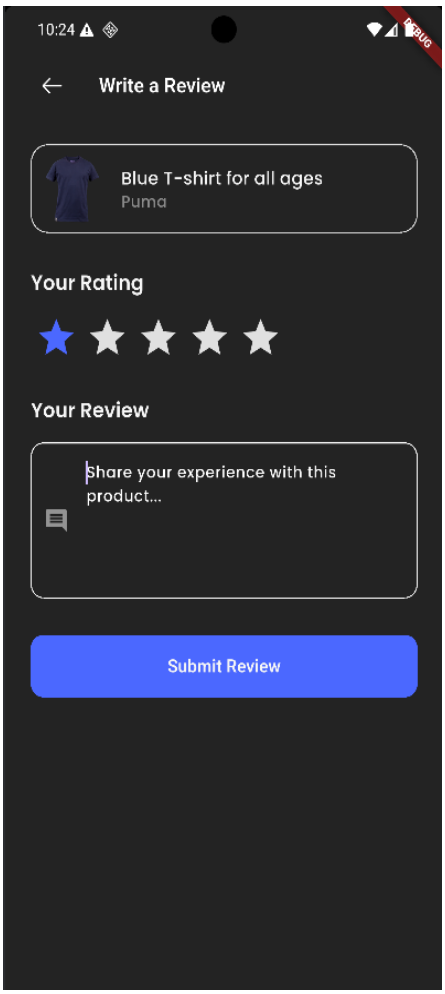

Figure 2.55 "Review & Ratings 2" interface

Table 2.38 "Reviews & Rating" interface

Interface	Reviews & Rating
Describe	Allows users to view all review and rating
Access	User clicks arrow right "Reviews" button in product detail

Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Reviews & Rating
Review	Textbox-String	N/A	Write a review about the product
Rating	Button	N/A	Rating of product
Work			
Name	**Describe**	**Success**	**Failure**
Click box review	Write a review about the product	Successful product reviews	
Click rating icon	Select rating: 1 - 5 star	Successful product rating	

2.9.1.17 Setting

In the setting screen, users can change their information or can experience extensions.

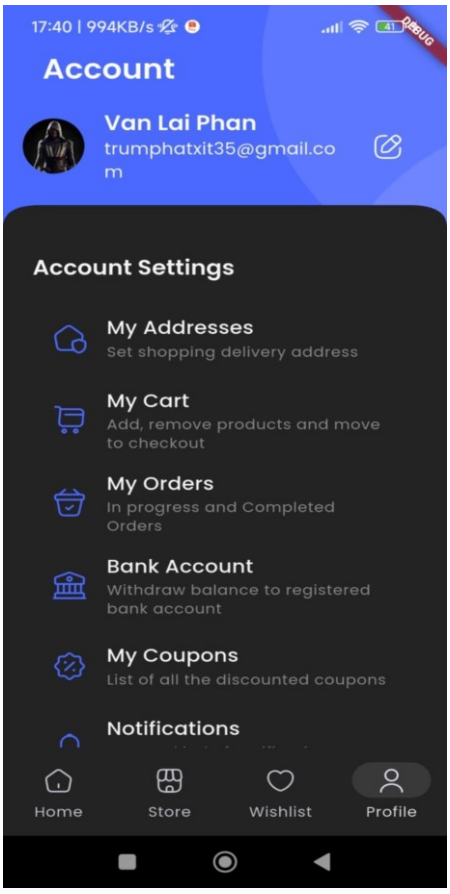

Figure 2.56 "Setting 1" interface

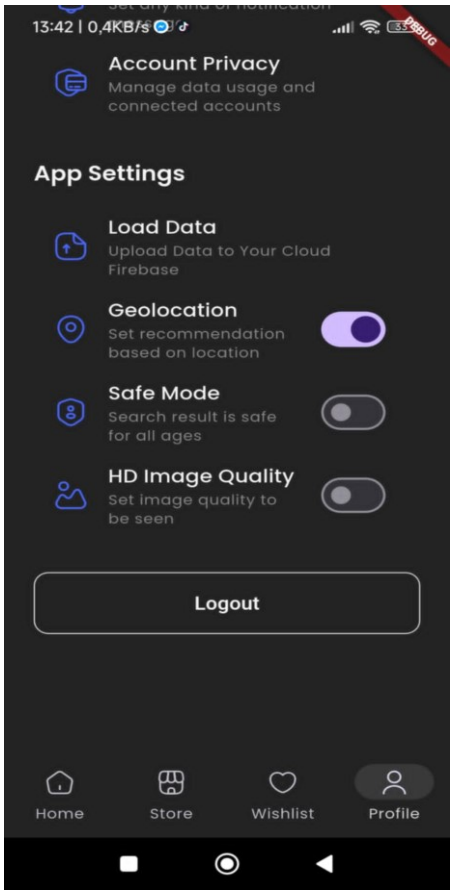

Figure 2.57 "Setting 2" interface

Table 2.39 "Setting" interface

Interface	Setting		
Describe	Allows users to view detail setting screen		
Access	User clicks on "profile icon" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Account
Image User	Image	N/A	User's photo
Username and email	Text	N/A	Username and account registration email
Edit profile icon	Linked Button	N/A	Edit user's profile
Address setting	Linked Button	N/A	Edit user's address
Order setting	Linked Button	N/A	View all orders
Logout	Linked Button	N/A	Sign out of your account
Work			
Name	**Describe**	**Success**	**Failure**
Click edit profile	Click edit profile icon button image	Redirect to profile screen	
Click My Addresses	Click My Address button	Redirect to address screen	
Click My Orders	Click My Orders button	Redirect to Order screen	
Click Logout	Click Logout button	Sign out of your account	

2.9.1.18 Profile

In the profile screen, users can view their details and can also change their photo or name

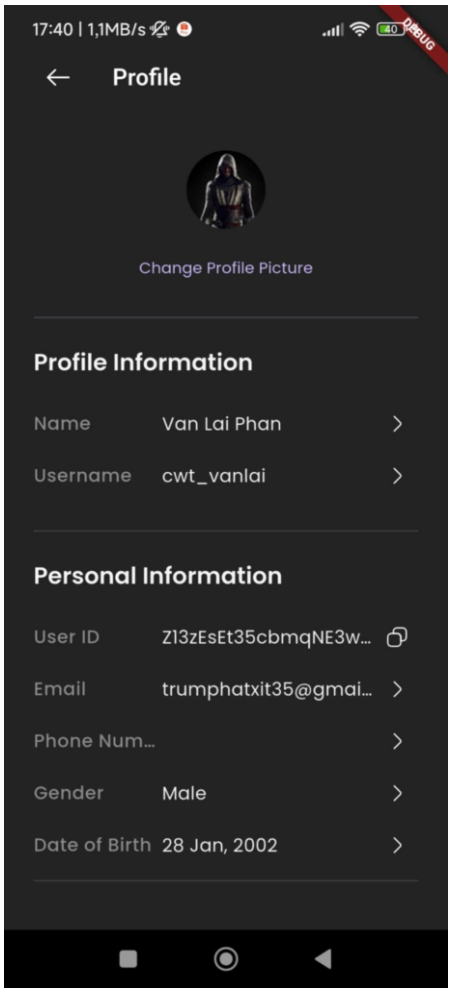

Figure 2.58 "Profile 1" interface

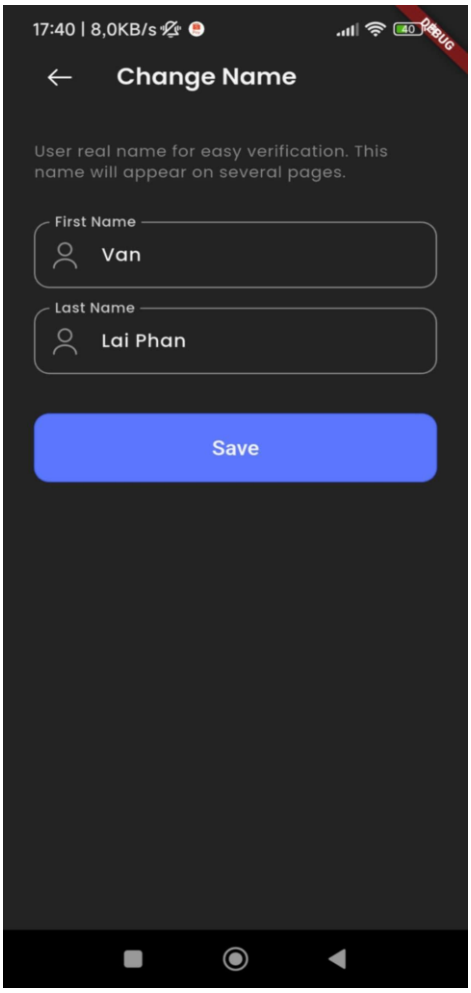

Figure 2.59 "Profile 2" interface

Table 2.40 "Profile" interface

Interface	Profile		
Describe	Allows users to view profile detail		
Access	User clicks on "Click edit profile image" button		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Profile
Image profile	Image	N/A	User's photo
Change Profile Picture	Linked Button	N/A	Change User's avatar

Profile information	Text	N/A	Name and Username of customer
Personal information	Text	N/A	Personal information: User Id, Email, Phone Number, Gender, Date of birth
Work			
Name	**Describe**	**Success**	**Failure**
Click Change Profile Picture	Select photo to change	Change User's avatar	
Click arrow right in Name text	Change User's name	Redirect to Change name screen	

2.9.1.19 Address

In the address screen, users can view or select addresses. They can also create new addresses when they have filled in all the information.

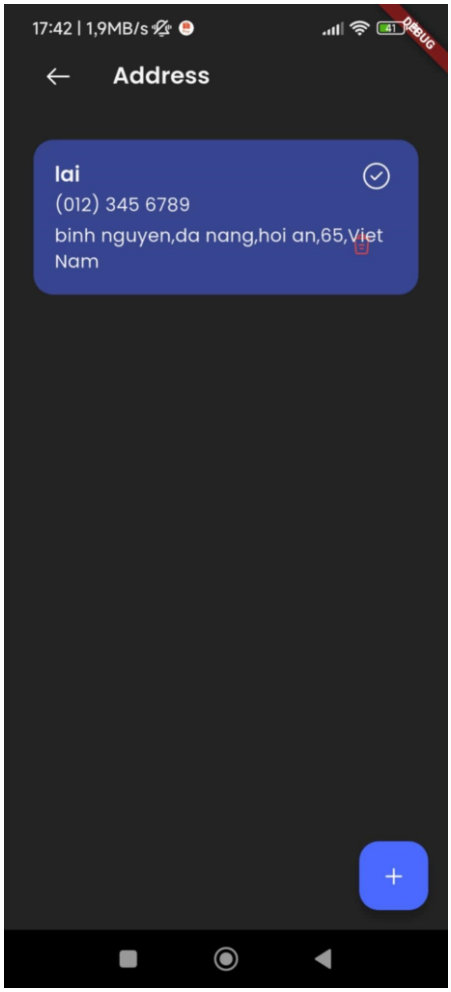

Figure 2.60 "Address 1" interface

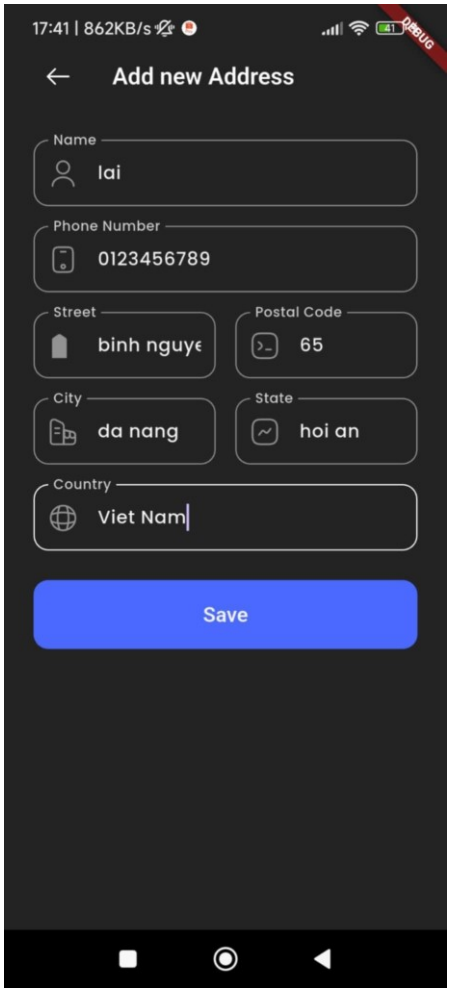

Figure 2.61 "Address 2" interface

Table 2.41 "Address and Add new address" interface

Interface	Address and Add new address		
Describe	Interface allows users select and create new address		
Access	User clicks My address button in setting screen		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Address and Add new address
List address	List	N/A	Select address
Last name	Textbox-String	N/A	Where to enter last name
Add new address button	Linked Button	N/A	Create new address
Name	Textbox-String	N/A	Where to enter username
Phone Number	Textbox-String	N/A	Where to enter phone number
Street	Textbox-String	N/A	Street of the user is live
Postal Code	Textbox-String	N/A	User's postal code
City	Textbox-String	N/A	City of the user is live
State	Textbox-String	N/A	the State of user
Country	Textbox-String	N/A	User's country
Save button	Linked Button	N/A	Redirect to address screen
Work			
Name	**Describe**	**Success**	**Failure**
No address name entered	Do not enter name then press save		"Something went wrong. Please try again" message
No phone number entered	Do not enter phone number then press save button		"Something went wrong. Please try again" message
No street entered	Do not enter street then press save		"Something went wrong. Please try again" message

No postal code entered	Do not enter postal code then press save button		"Something went wrong. Please try again" message
No city entered	Do not enter city then press save		"Something went wrong. Please try again" message
No state entered	Do not enter state then press save button		"Something went wrong. Please try again" message
No country entered	Do not enter country then press save		"Something went wrong. Please try again" message
Click save button	Save changes	Redirect to address screen	

2.9.1.20 Order detail

In the order detail screen, users can view order details including status, information, shipping address, payment method, pricing details.

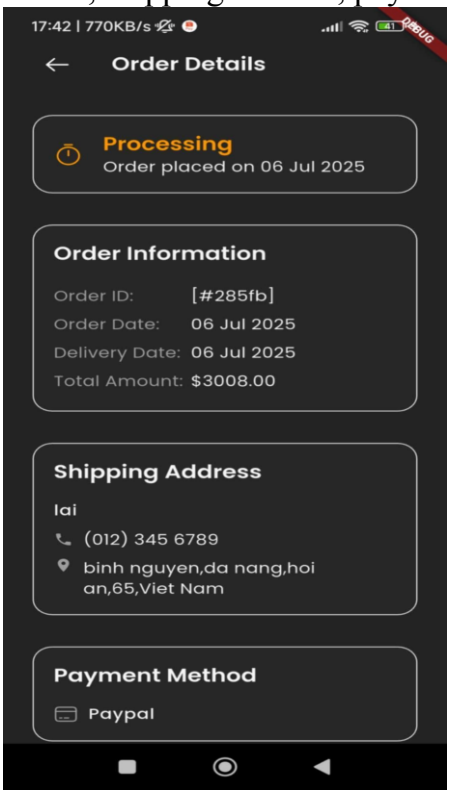

Figure 2.62 "Order detail 1" interface

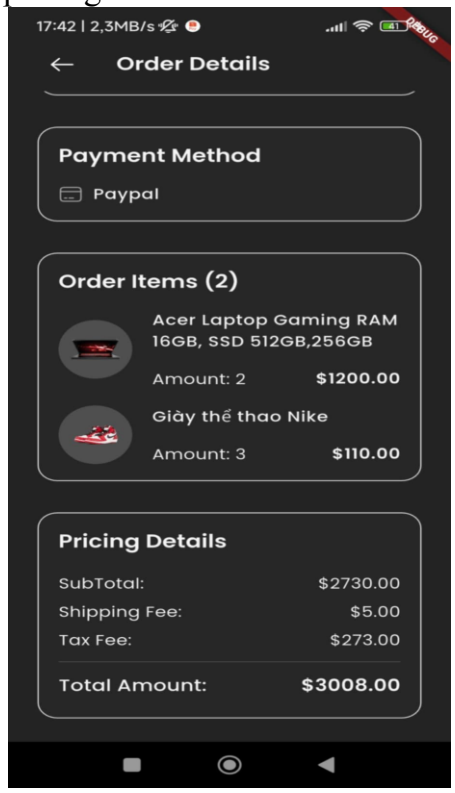

Figure 2.63 "Order detail 2" interface

Table 2.42 "Order Details screen" interface

Interface	Order Details screen		
Describe	Allows users to view detail order		
Access	User clicks on "My order" button in setting screen		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Order Details
Status order	Text	N/A	Status order: pending, processing, shipped, delivered, cancelled.Date order
Order information	Text	N/A	Order information: Order Id, Order Date, Delivery Date, Total amount
Shipping address	Text	N/A	Shipping address: username, phone number, address
Payment Method	Text	N/A	Payment Method: Paypal, Google Pay, Apple Pay, Visa, Master Card, Credit Card
Order items	Image- Text	N/A	All products are to order
Pricing Details	Text	N/A	Pricing Details:SubTotal, Shipping Fee, Tax Fee, Total Amount
Work			
Name	**Describe**	**Success**	**Failure**
Scroll	Scroll to view details	Display Order details	

2.9.2 Admin

2.9.2.1 Login

In the login screen, the admin must enter the correct email and password to be able to log in to the management page.

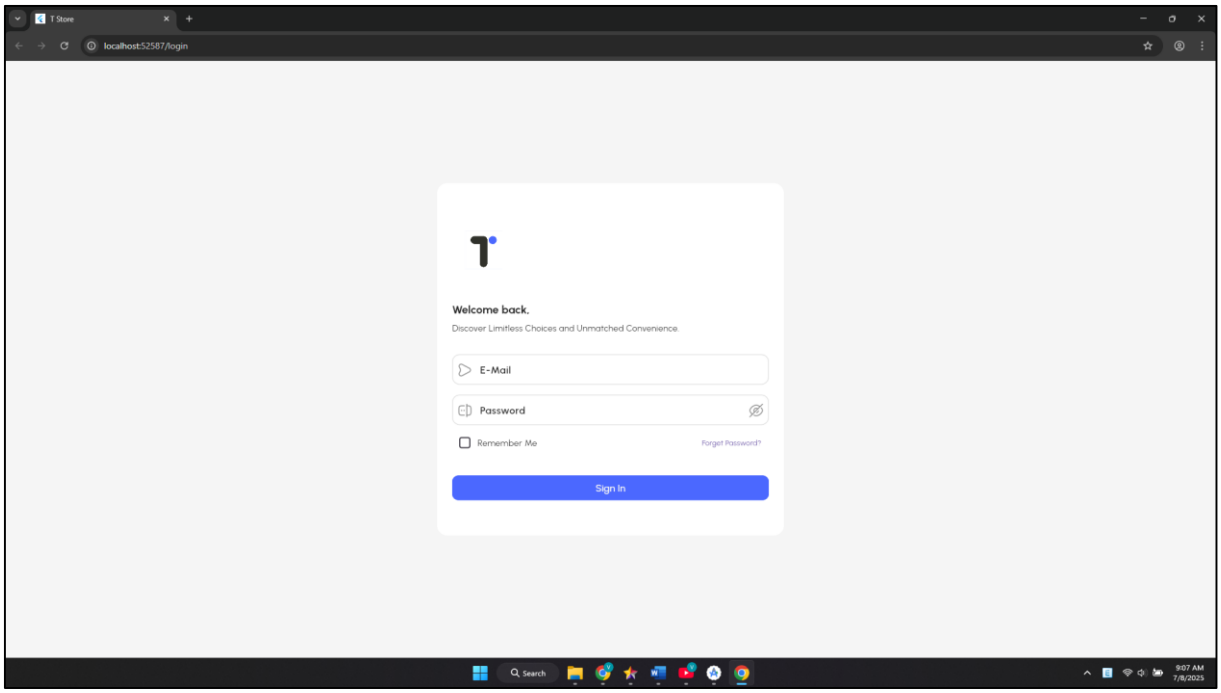

Figure 2.64 "Login" interface

Table 2.43 "Admin Login" interface

Interface	Admin Login		
Describe	The interface allows the admin to login to use other functions.		
Access	Admin access the website		
Interface content			
Item	**Type**	**Data**	**Describe**
Email	Textbox-String	N/A	Where to enter email
Password	Textbox-String	N/A	Where to enter password
Forget Password	Linked Button	N/A	Leads to forgot password page
Remember Me	Button	N/A	Where to remember passwords
Sign in	Linked Button	N/A	Log in to the system

Name	Describe	Success	Failure
No email entered	Do not enter email then press sign in button		"Invalid email address" message
No password entered	Do not enter password then press sign in button		"Password is required" message
Sign in with email and password	Process login to the system with the entered user name and password	Redirect to dashboard screen	"Something went wrong. Please try again" message

2.9.2.2 Dashboard

In the dashboard screen, admin can view statistics using bar and pie charts. In addition, admin can also view total sale, average order value, total order, list order.

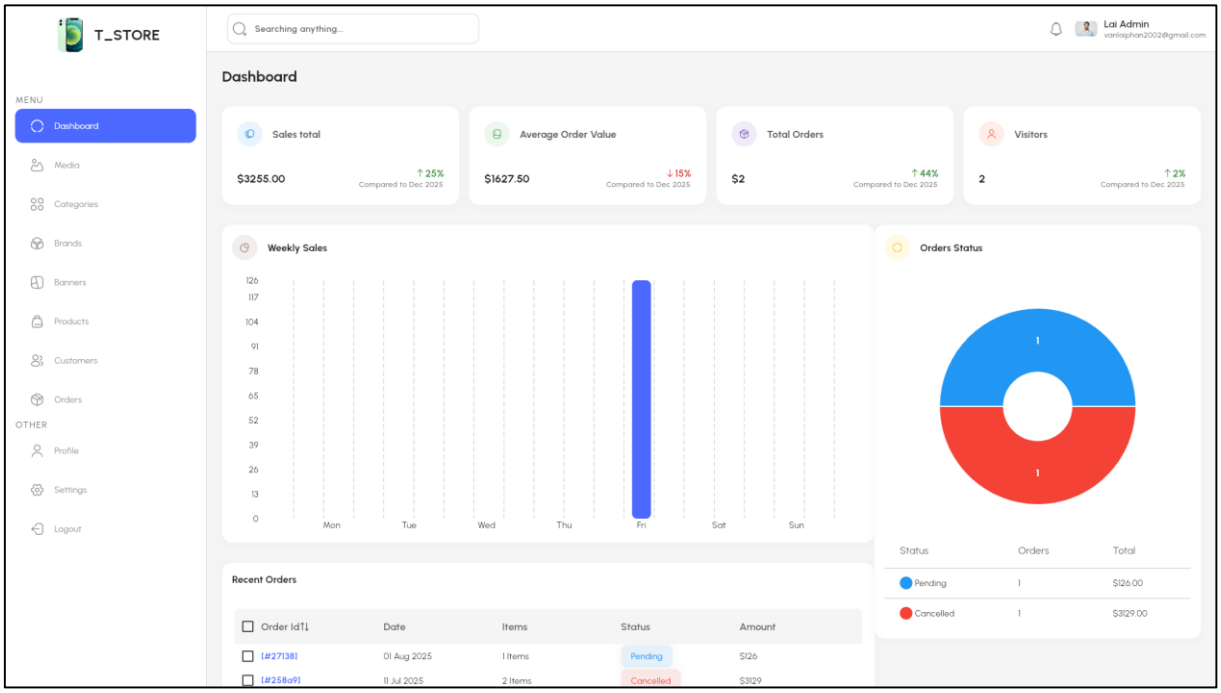

Figure 2.65 "Dashboard 1" interface

Table 2.44 "Dashboard" interface

Interface	Dashboard		
Describe	Allows admin to view sales statistics		
Access	Admin clicks Dashboard in sidebar		
Interface content			
Item	**Type**	**Data**	**Describe**
Dashboard Card	Text	N/A	Dashboard Card: Sales total, Average order value, Total Orders, Visitors
Barchart	Chart	N/A	Order statistics using bar chart
Piechart	Chart	N/A	Order statistics using pie chart
Table recent orders	Table	N/A	List orders
Work			
Name	**Describe**	**Success**	**Failure**
Click dashboard in sidebar	View order statistics	Displays all order statistics	

2.9.2.3 Media

In the media screen, the admin can upload photos and select a storage folder for the photos.

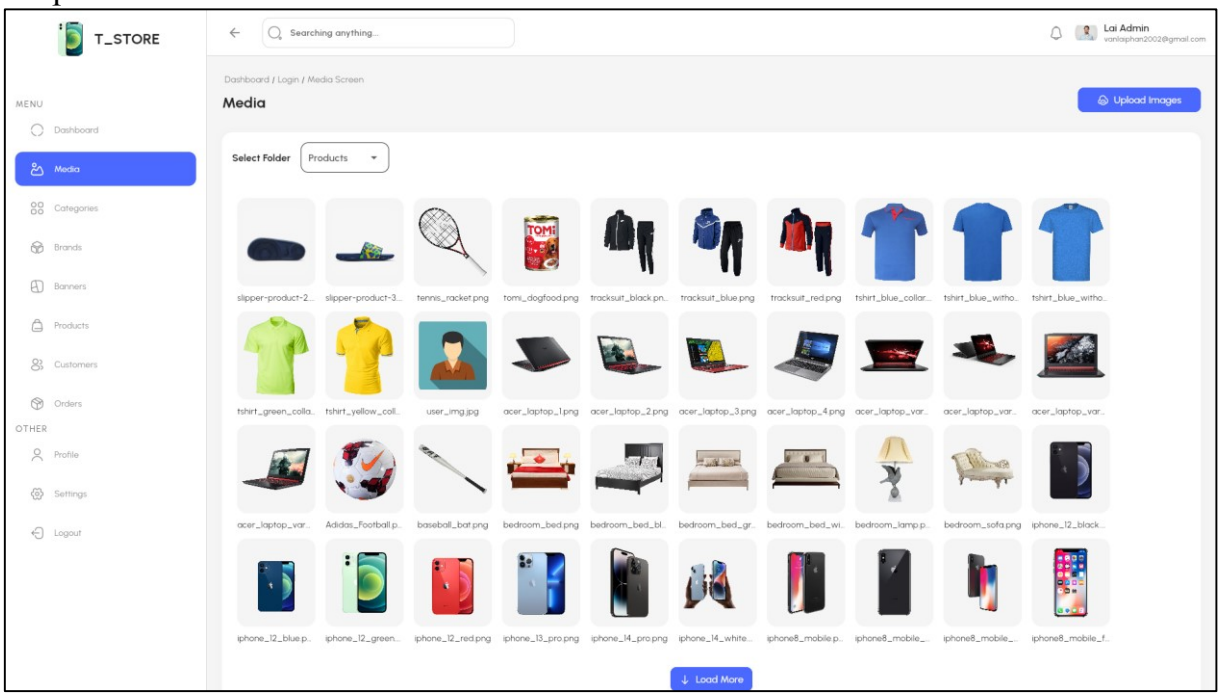

Figure 2.66 "Media" interface

Table 2.45 "Media" interface

Interface	Media		
Describe	Allows admin to view all image upload		
Access	Admin clicks Media in sidebar		
Interface content			
Item	**Type**	**Data**	**Describe**
Name screen	Text	N/A	Name of screen is Media
Select Folder	Button	N/A	Admin can choose upload folder
Upload Image	Button	N/A	Admin can upload images
Work			
Name	**Describe**	**Success**	**Failure**
Click select folder	Admin can choose upload folder	Image has been successfully added to the folder	
Click upload image	Admin can upload images	Image has been successfully added to the folder	

2.9.2.4 Category

In the category screen, admin can view the category list, create, update or delete a category.

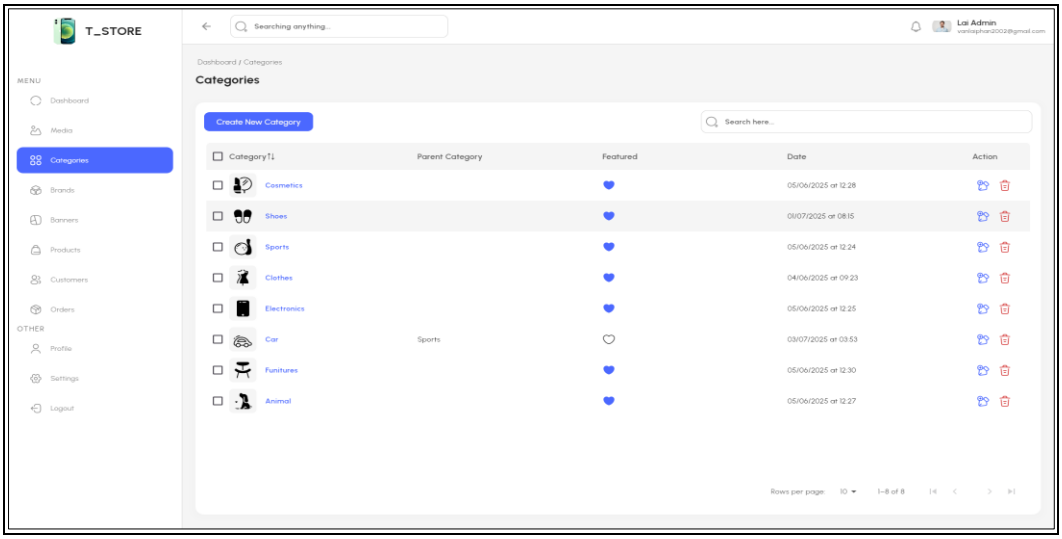

Figure 2.67 "Category" interface

In the create category screen, admin fills in the name, selects the parent category, selects the image, featured category or not to display in the store then clicks "Create" to save.

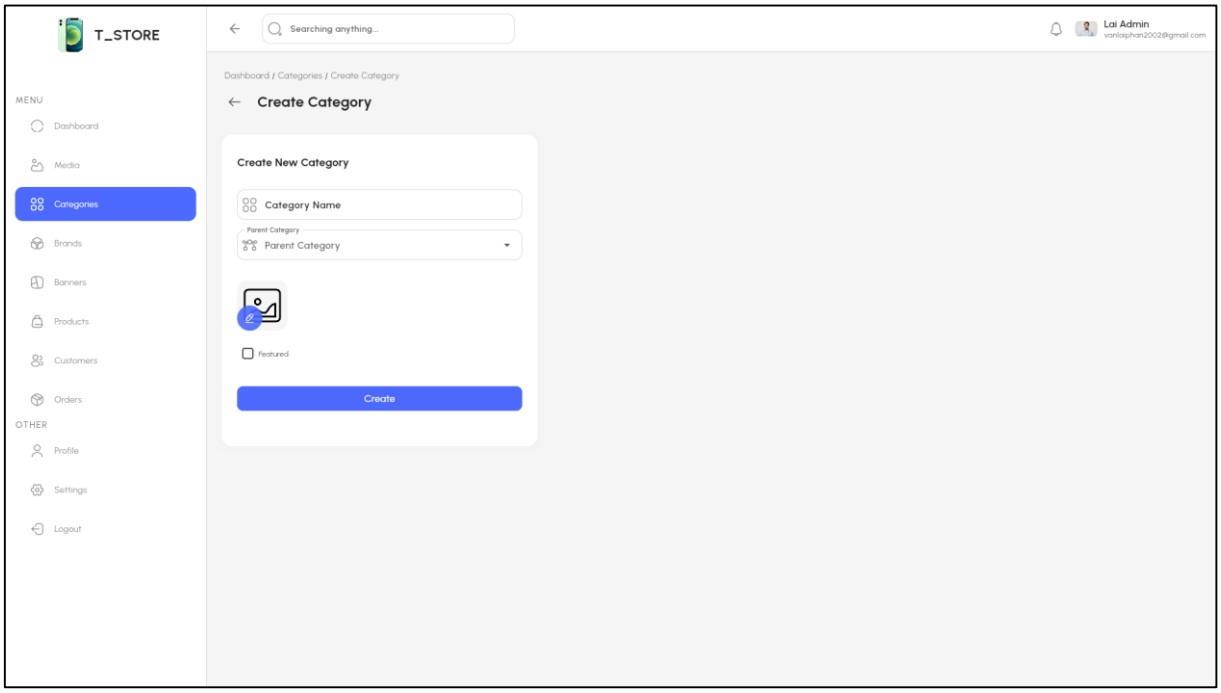

Figure 2.68 "Create category" interface

In the update category screen, admin can change the name, parent category, image, featured category or not to display in the store then clicks "Update" to save.

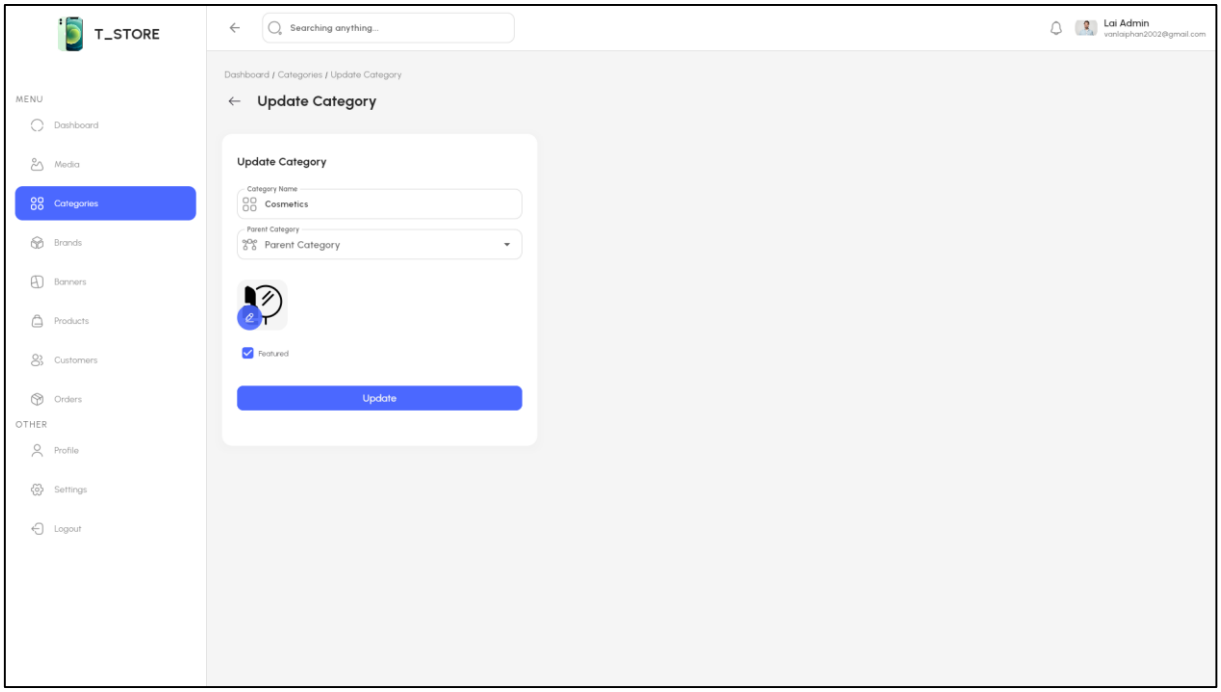

Figure 2.69 "Update category" interface

Table 2.46 "Category" interface

Interface	Category		
Describe	Show list category and admin can create, update, delete category		
Access	Admin clicks Category button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Create new category	Linked Button	N/A	Create new category for app
Table category	Table	N/A	Show all category
Search	Textbox-String	N/A	Search name category
Featured button	Button	N/A	When pressed, the category will be displayed in the app
Update Action Button	Linked Button	N/A	Update category
Delete Action Button	Button	N/A	Delete category
Act			
Name	**Describe**	**Success**	**Failure**
Click Create new category Button	Admin click Create new category	Redirect to Create category screen	
Fill in the search box	Enter the word you want to search for in the search box	Suggest categories by characters typed	
Click Featured button	Admin click Featured category button	Featured categories appear in the category list in the app	
Click Update action button	Admin click Update category button	Redirect to Update category screen	
Click Delete action button	Admin click Delete category button	Remove categories from the table	

2.9.2.5 Brand

In the brand screen, admin can view the brand list, create, update or delete brand.

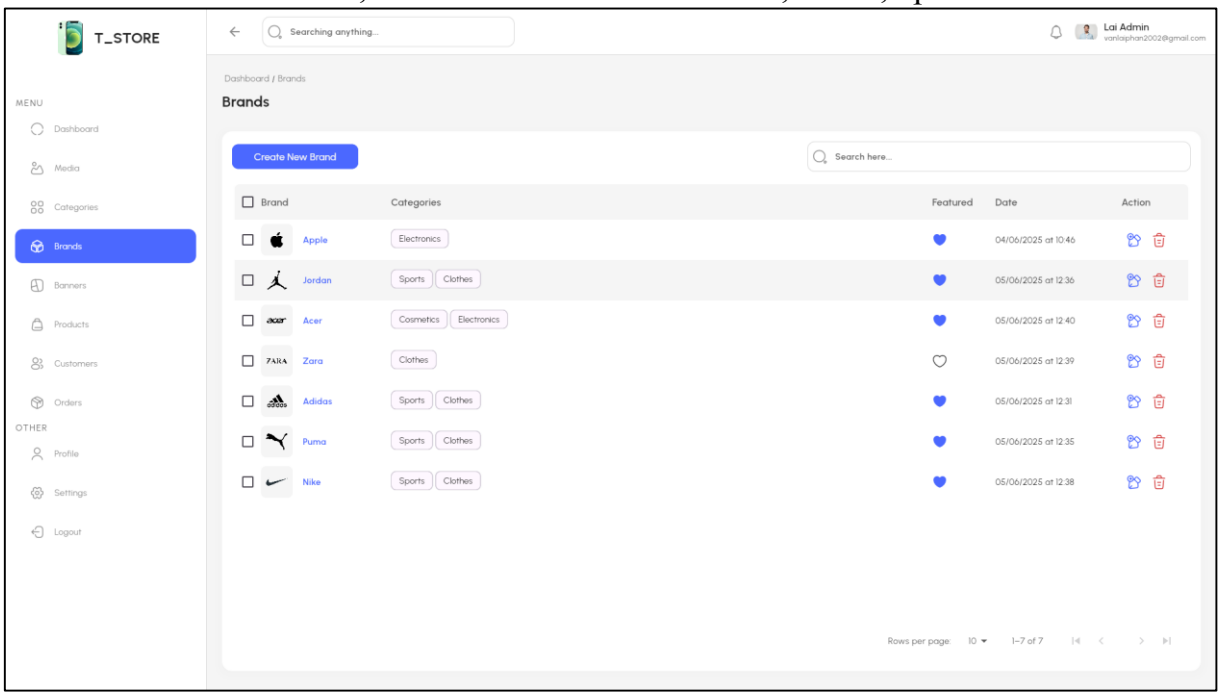

Figure 2.70 "Brand" interface

In the create brand screen, admin fills in the name, selects the category, selects the image, features the brand or not to display in the store then clicks "Create" to save.

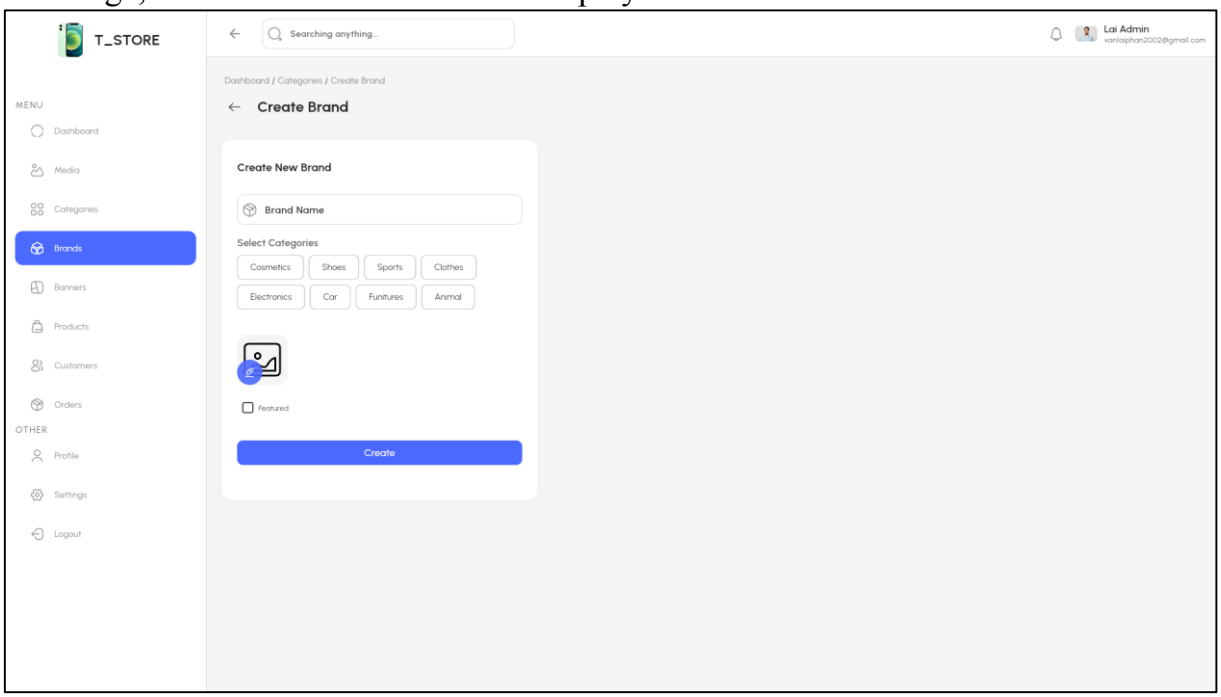

Figure 2.71 "Create brand" interface

In the update brand screen, admin can change the name, category, image, features or not to display in the store then clicks "Update" to save.

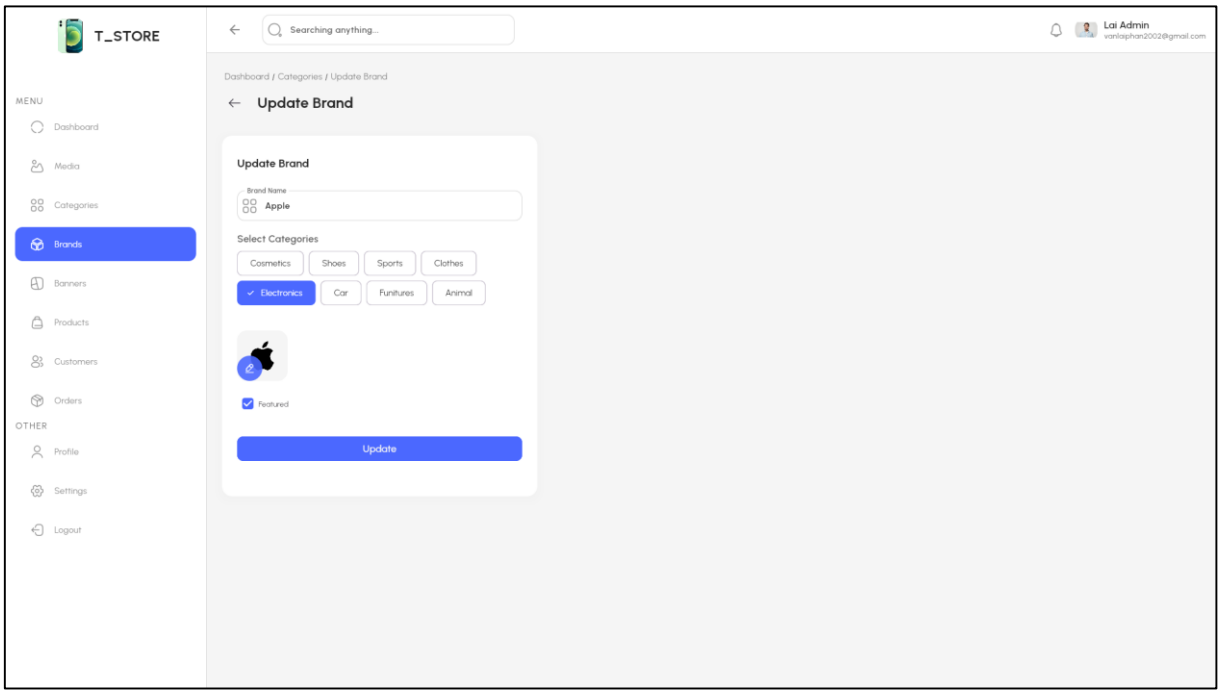

Figure 2.72 "Update brand interface" interface

Table 2.47 "Brands" interface

Interface	Brands		
Describe	Show list Brand and admin can create, update, delete brand		
Access	Admin clicks Brand button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Create new brand	Linked Button	N/A	Create new brand for app
Table brand	Table	N/A	Show all brand
Search	Textbox-String	N/A	Search name category
Featured button	Button	N/A	When pressed, the brand will be displayed in the app
Update Action Button	Linked Button	N/A	Update brand
Delete Action Button	Button	N/A	Delete brand

Act			
Name	**Describe**	**Success**	**Failure**
Click Create new brand Button	Admin click Create new brand	Redirect to Create brand screen	
Fill in the search box	Enter the word you want to search for in the search box	Suggest brand by characters typed	
Click Featured button	Admin click Featured brand button	Featured brand appear in the category list in the app	
Click Update action button	Admin click Update brand button	Redirect to Update brands screen	
Click Delete action button	Admin click Delete brand button	Remove brands from the table	

2.9.2.6 Banner

In the banner screen, admin can view the banner list, create, update or delete banners.

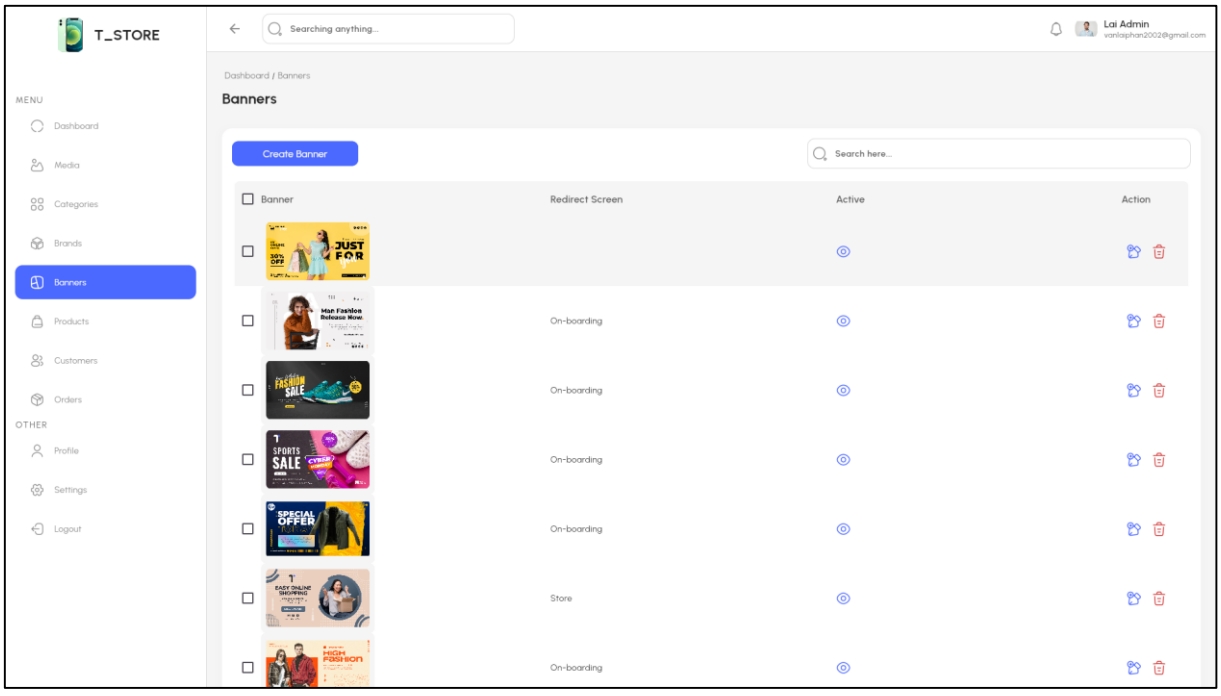

Figure 2.73 "Banner" interface

In the create banner screen, admin selects the image, selects the navigation screen, actives the banner or not to display in the store then clicks" Create" to save

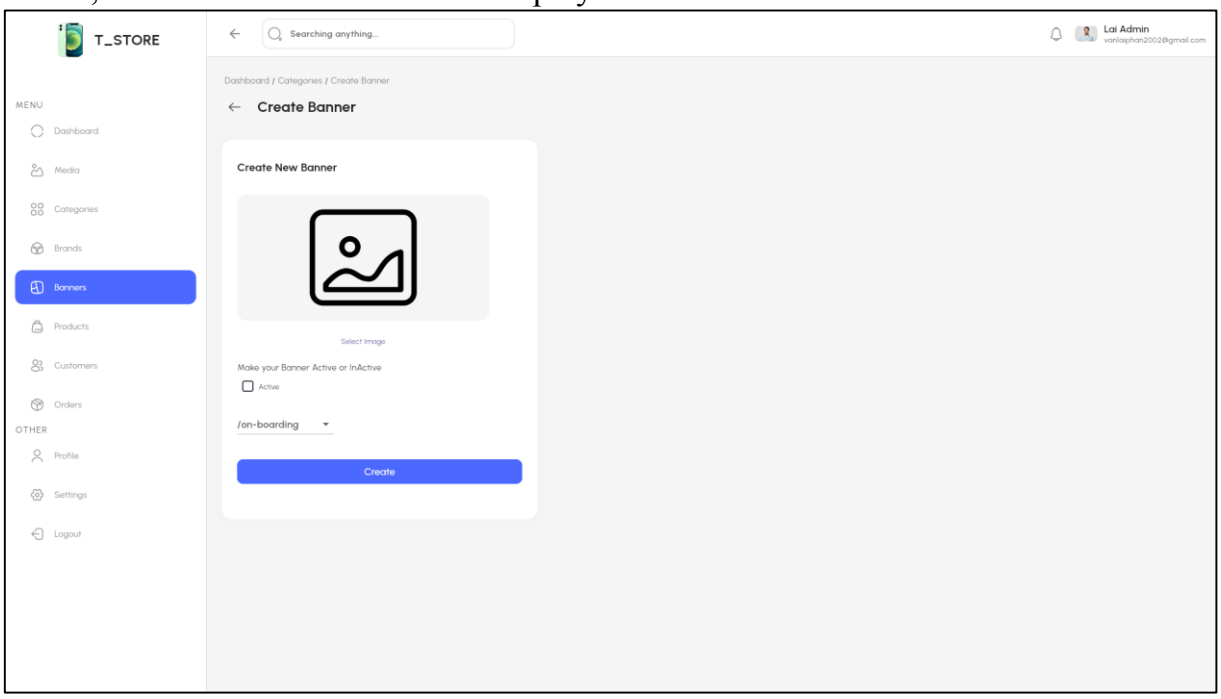

Figure 2.74 "Create banner" interface

In the update banner screen, admin can change the image, navigation screen, actives or not to display in the store then clicks "Update" to save.

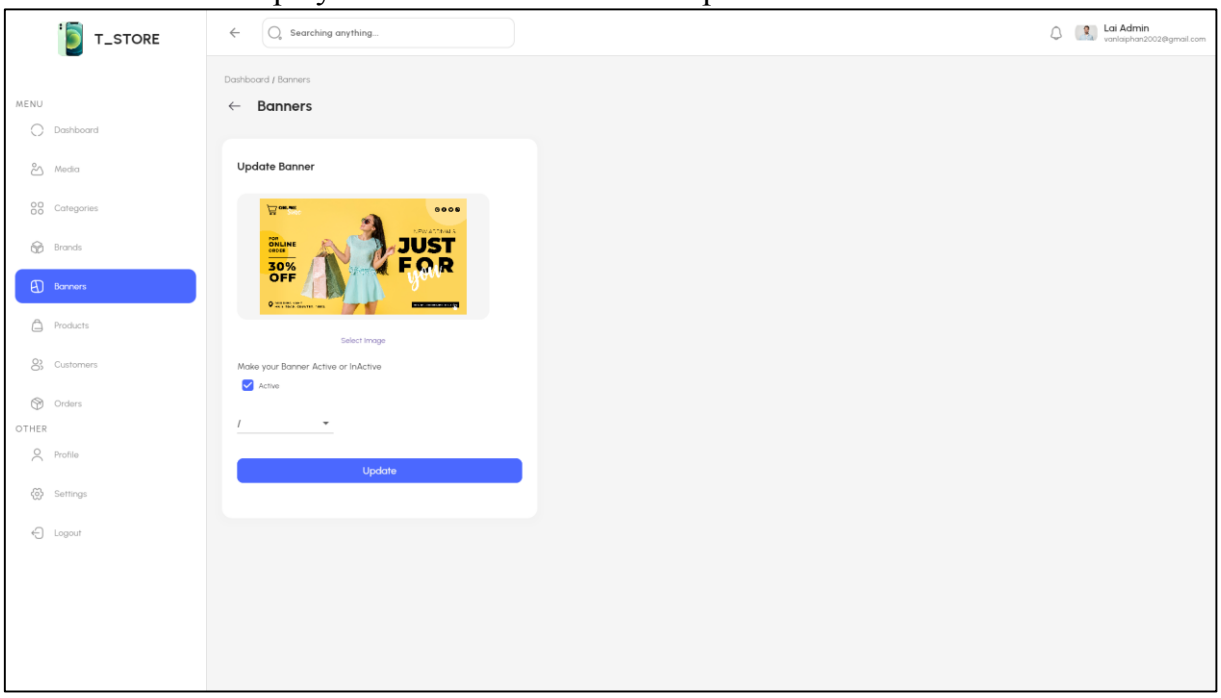

Figure 2.75 "Update banner" interface

Table 2.48 "Banner" interface

Interface	Banner		
Describe	Show list banner and admin can create, update, delete banner		
Access	Admin clicks Banners button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Create new banner	Linked Button	N/A	Create new banner for app
Table banner	Table	N/A	Show all banner
Search	Textbox-String	N/A	Search name banner
Featured button	Button	N/A	When pressed, the banner will be displayed in the app
Update Action Button	Linked Button	N/A	Update banner
Delete Action Button	Button	N/A	Delete banner
Act			
Name	**Describe**	**Success**	**Failure**
Click Create new banner Button	Admin click Create new banner	Redirect to Create banner screen	
Fill in the search box	Enter the word you want to search for in the search box	Suggest banner by characters typed	
Click Featured button	Admin click Featured banner button	Featured banner appear in the banner list in the app	
Click Update action button	Admin click Update banner button	Redirect to Update banner screen	
Click Delete action button	Admin click Delete banner button	Remove banner from the table	

2.9.2.7 Product

In the product screen, admin can view the product list, create, update or delete products

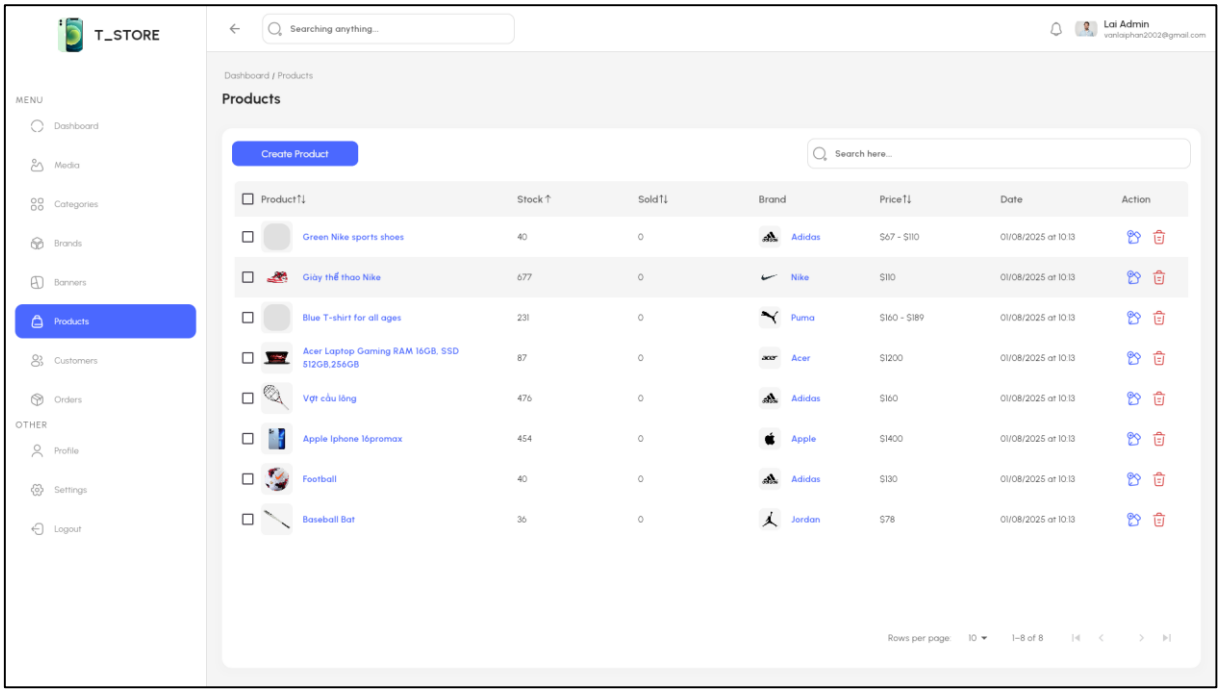

Figure 2.76 "Product" interface

In the create product screen, admin fills in the product information, selects the image, selects the brand, selects the category, publishes or hides to display in the store or not, then clicks "Save changes" to save.

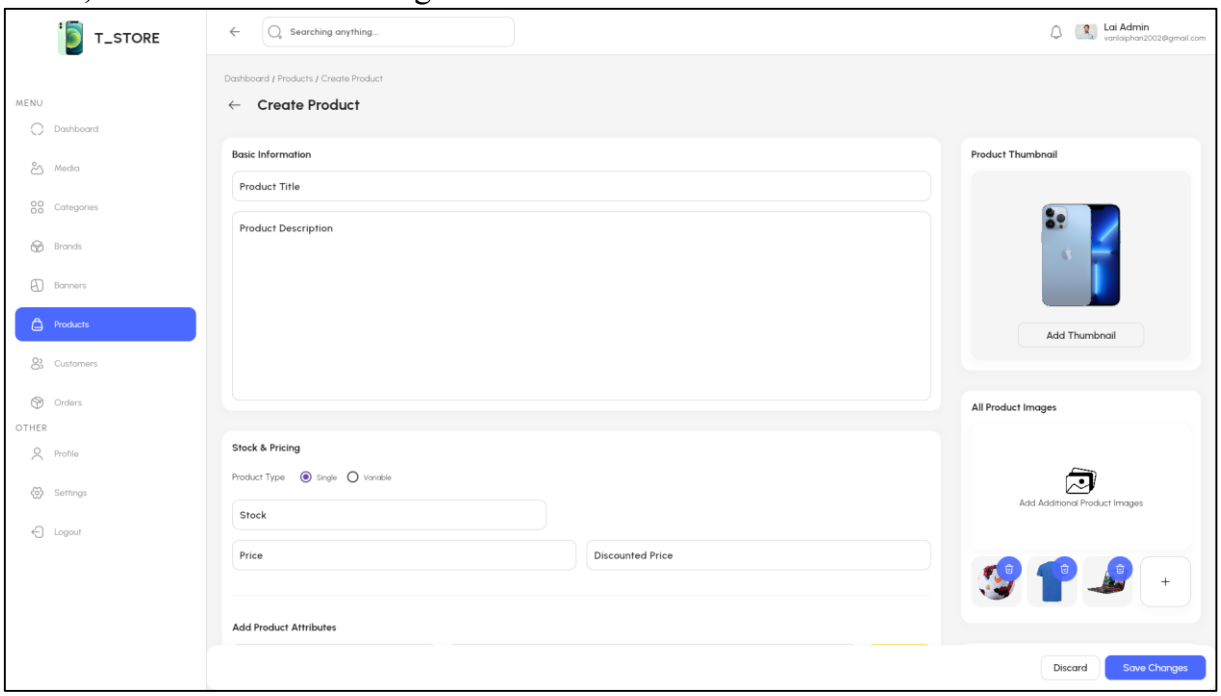

Figure 2.77 "Create product 1" interface

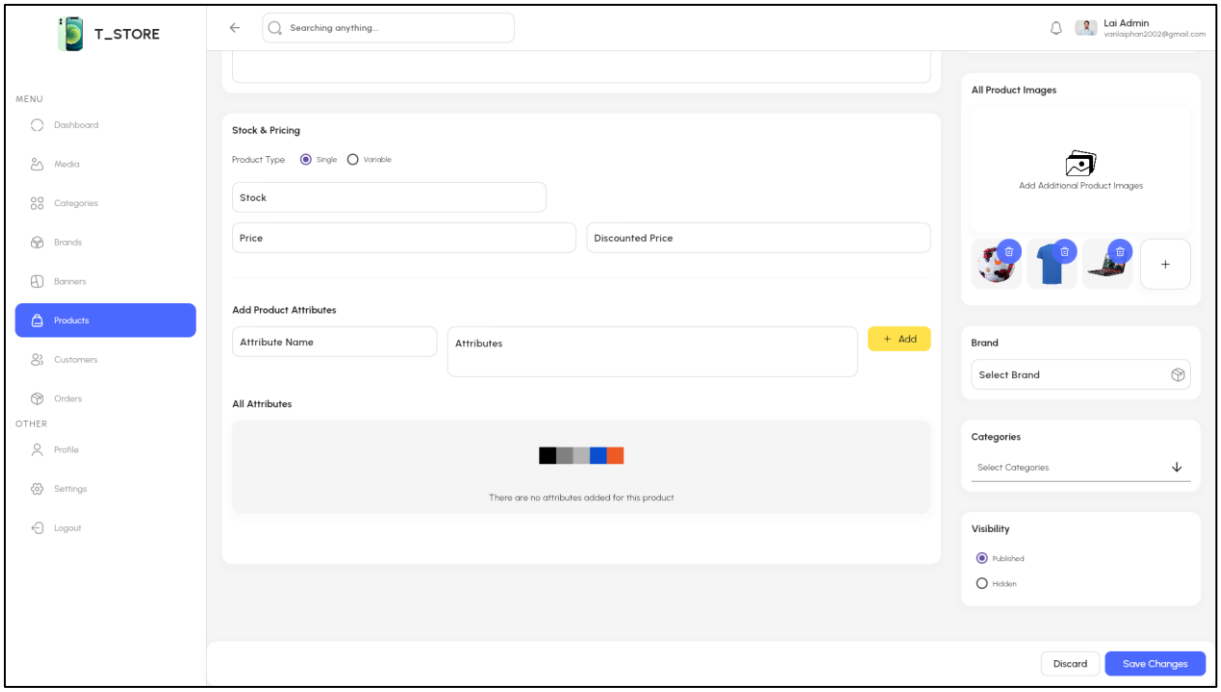

Figure 2.78 "Create product 2" interface

In the update product screen, admin can change the product information, image, brand, category, publish or hides to display in the store or not, then clicks "Save changes" to save.

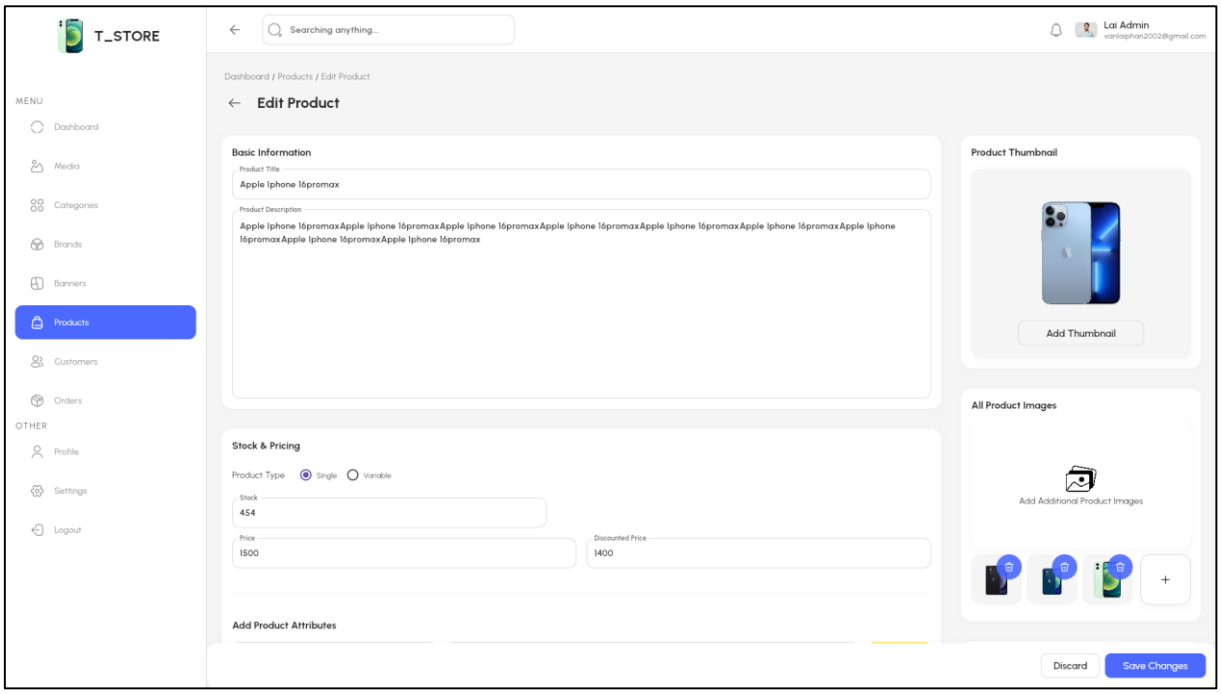

Figure 2.79"Update product 1" interface

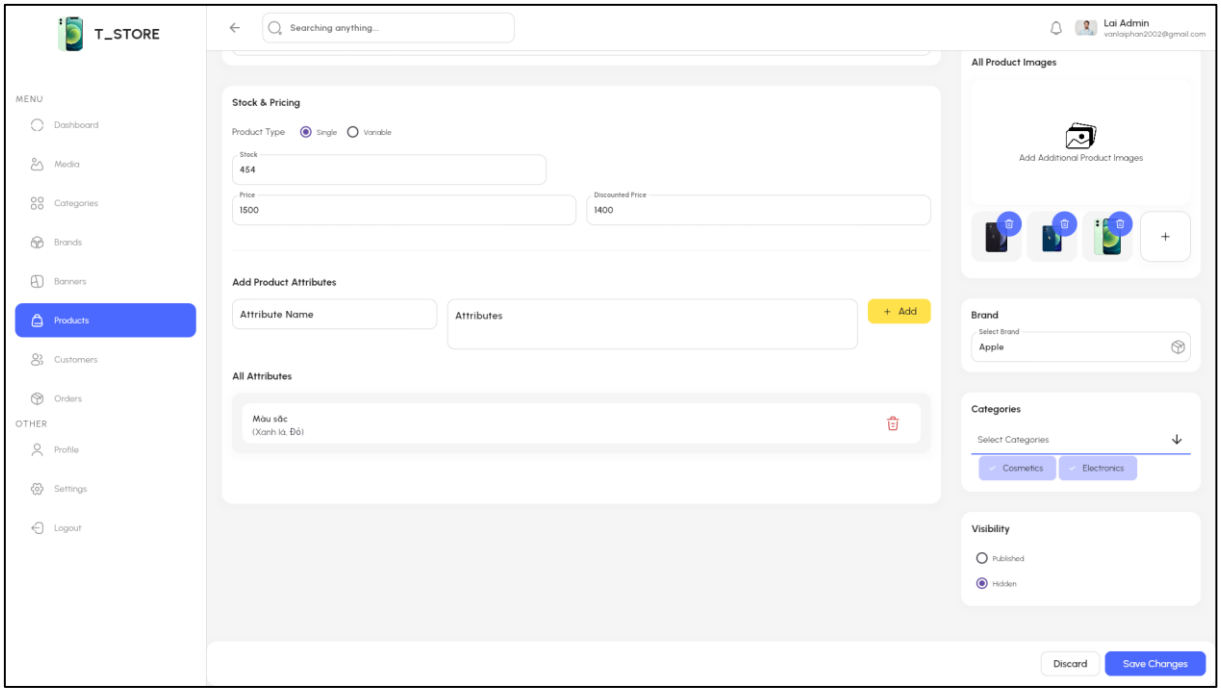

Figure 2.80 "Update product 2" interface

Table 2.49 "Products" interface

Interface	Products		
Describe	Show list product and admin can create, update, delete product		
Access	Admin clicks Product button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Create new product	Linked Button	N/A	Create new product for app
Table product	Table	N/A	Show all product
Search	Textbox-String	N/A	Search name product
Featured button	Button	N/A	When pressed, the product will be displayed in the app
Update Action Button	Linked Button	N/A	Update product
Delete Action Button	Button	N/A	Delete product
Act			
Name	**Describe**	**Success**	**Failure**

Click Create new product Button	Admin click Create new product	Redirect to Create product screen	
Fill in the search box	Enter the word you want to search for in the search box	Suggest products by characters typed	
Click Featured button	Admin click Featured products button	Featured product appear in the product list in the app	
Click Update action button	Admin click Update product button	Redirect to Update product screen	
Click Delete action button	Admin click Delete product button	Remove product from the table	

2.9.2.8 Customer

In customer screen, admin can view customer list, view details or delete customer.

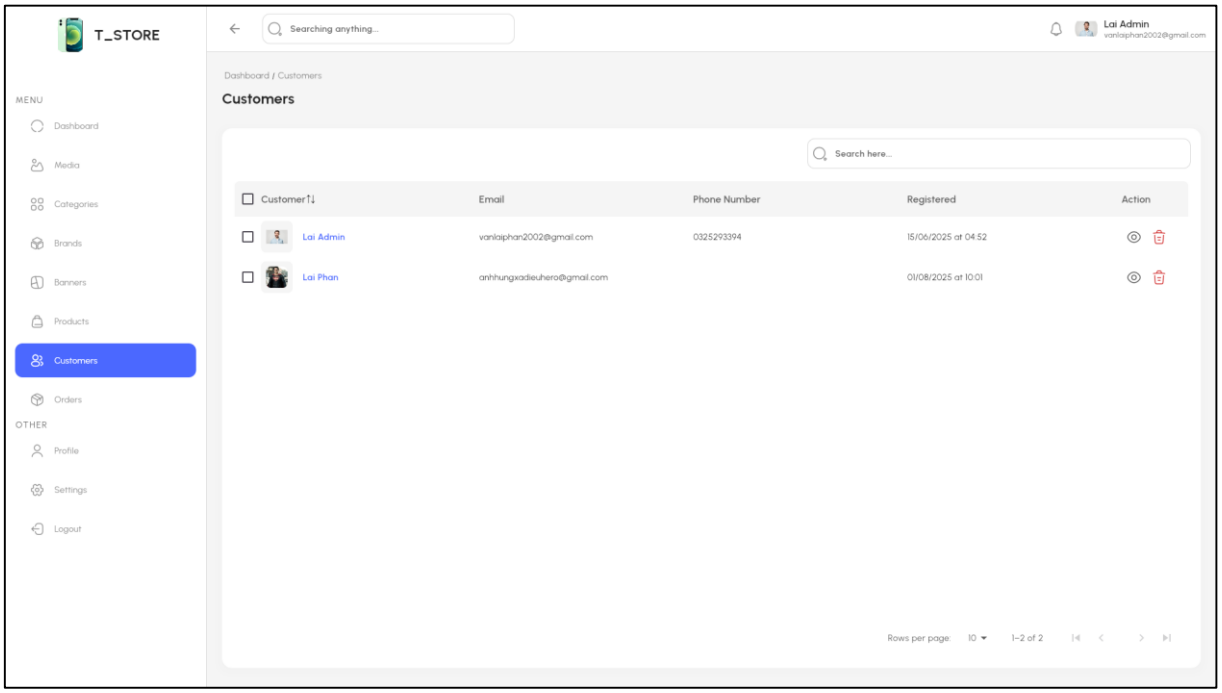

Figure 2.81 "Customer" interface

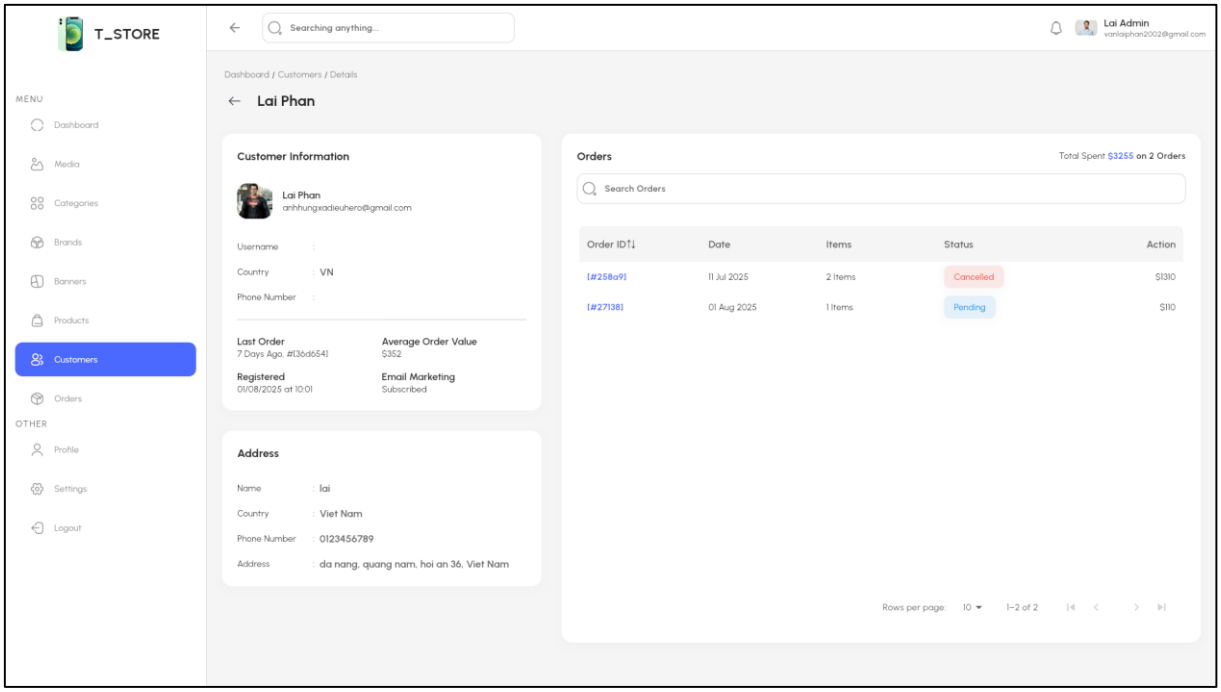

Figure 2.82 "Customer detail" interface

Table 2.50 "Customers" interface

Interface	Customers		
Describe	Show list customer and admin can view customer detail		
Access	Admin clicks Customers button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Table customer	Table	N/A	Show all customer
Search	Textbox-String	N/A	Search name customer
View Action Button	Linked Button	N/A	View customer detail
Delete Action Button	Button	N/A	Delete customer
Act			

Name	**Describe**	**Success**	**Failure**
Fill in the search box	Enter the word you want to search for in the search box	Suggest customer by characters typed	
Click View action button	Admin click View customer button	Redirect to Customer detail screen	

Click Delete action button	Admin click Delete customer button	Remove customer from the table	

2.9.2.9 Order

In the order screen, admin can view the list of orders, view details, update status and delete orders.

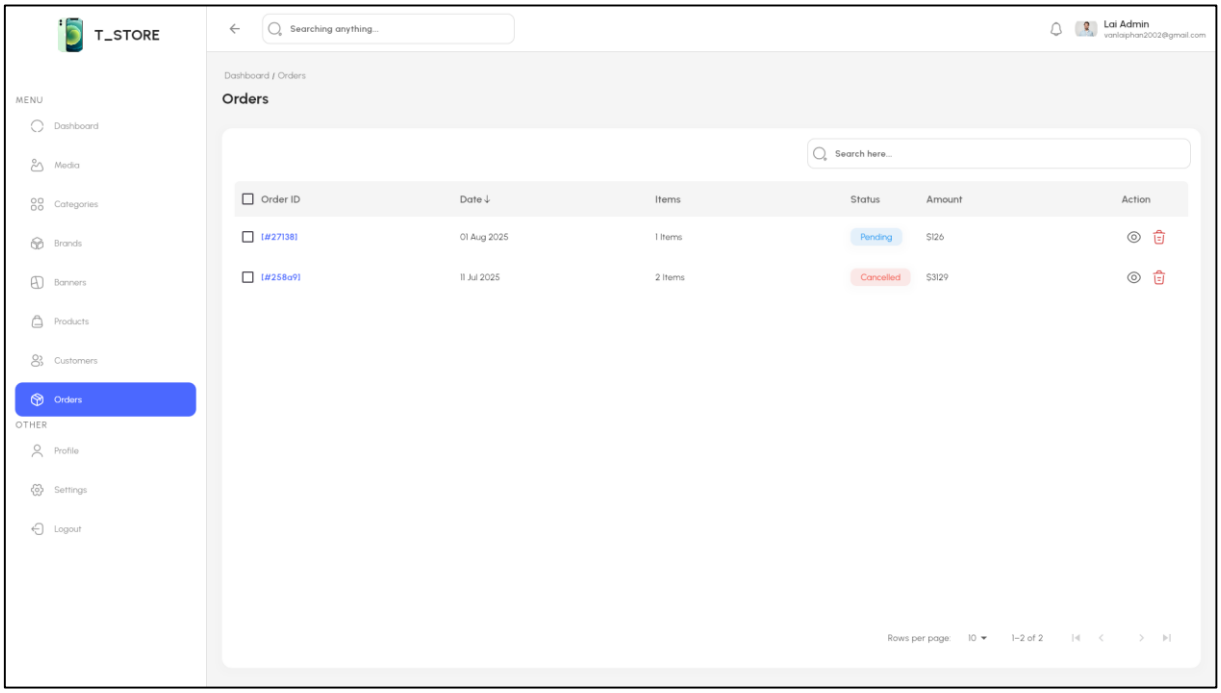

Figure 2.83 "Order" interface

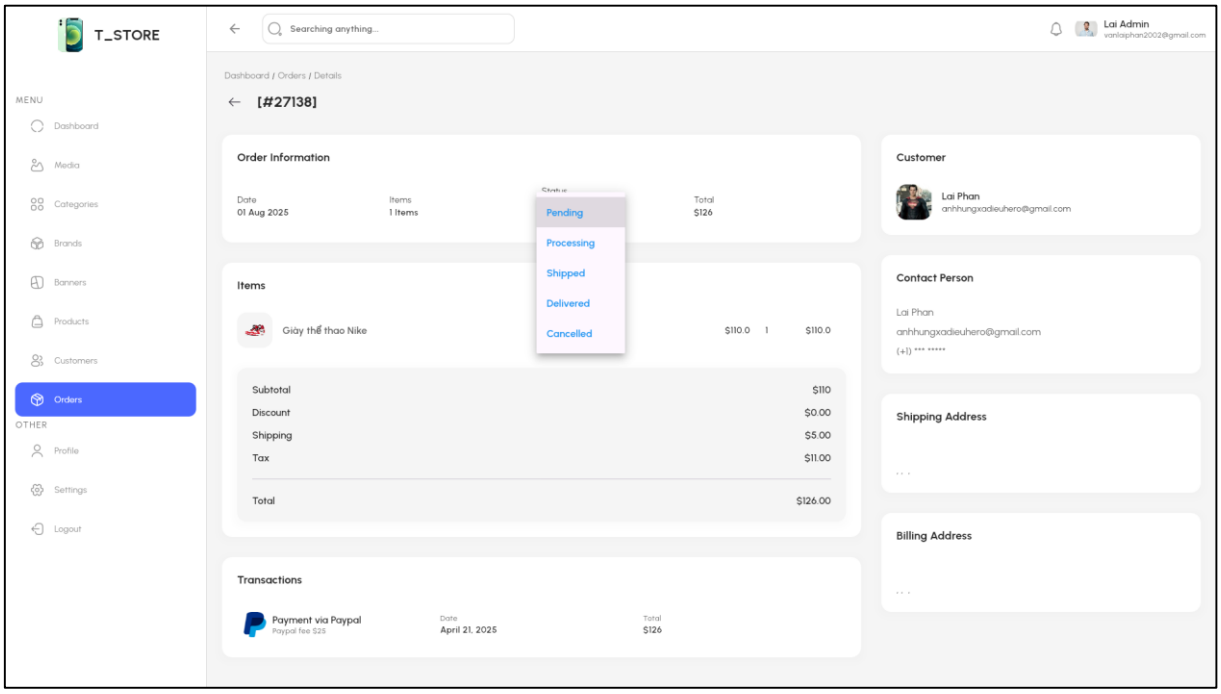

Figure 2.84 "Orders" interface

Table 2.51 "Orders" interface

Interface	Orders		
Describe	Show list order and admin can update,view order detail		
Access	Admin clicks Orders button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Table order	Table	N/A	Show all order
Search	Textbox-String	N/A	Search order
Update Action Button	Linked Button	N/A	Update status order
Delete Action Button	Button	N/A	Delete order
Act			
Name	**Describe**	**Success**	**Failure**
Fill in the search box	Enter the word you want to search for in the search box	Suggest order by characters typed	
Click Update action button	Admin click Update order button	Redirect to Update order screen	
Click Delete action button	Admin click Delete order button	Remove order from the table	

2.9.2.10 Profile

In In the profile screen, admin can view profile detail ,select image admin and change information

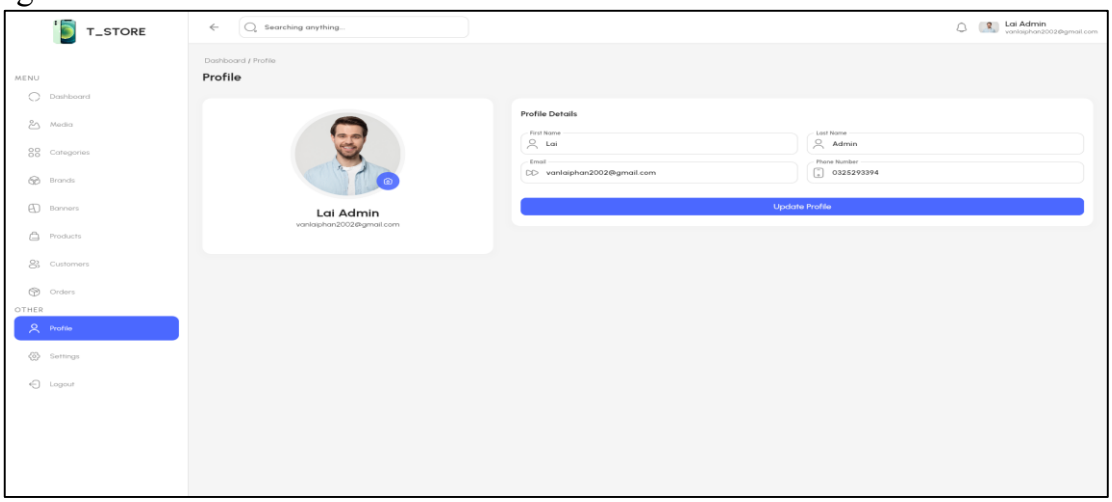

Figure 2.85 "Profile" interface

Table 2.52 "Profile" interface

Interface	Profile		
Describe	Show profile detail		
Access	Admin clicks Profile button in sidebar		
Content			
Item	**Type**	**Data**	**Describe**
Update image Button	Linked Button	N/A	Update profile image
First name	Textbox-String	N/A	Update first name
Last name	Textbox-String	N/A	Update last name
Email	Textbox-String	N/A	Admin's email
Phone number	Textbox-String	N/A	Update phone number
Update profile	Button	N/A	Update all changes
Act			
Name	**Describe**	**Success**	**Failure**
Click Update image button	Admin click Update image button	Redirect to Media screen to select or upload image	
Click First name	Admin click First name to update last name	Success update first name	
Click Last name	Admin click Last name to update last name	Success update last name	
Click Phone number	Admin click phone number to update phone number	Success update phone number	
Click Update profile	Admin click Update profile	All changes have been updated successfully.	

2.9.2.11 Setting

In the setting screen, admin can choose store avatar, rename, create tax rates, shipping cost, free shipping threshold

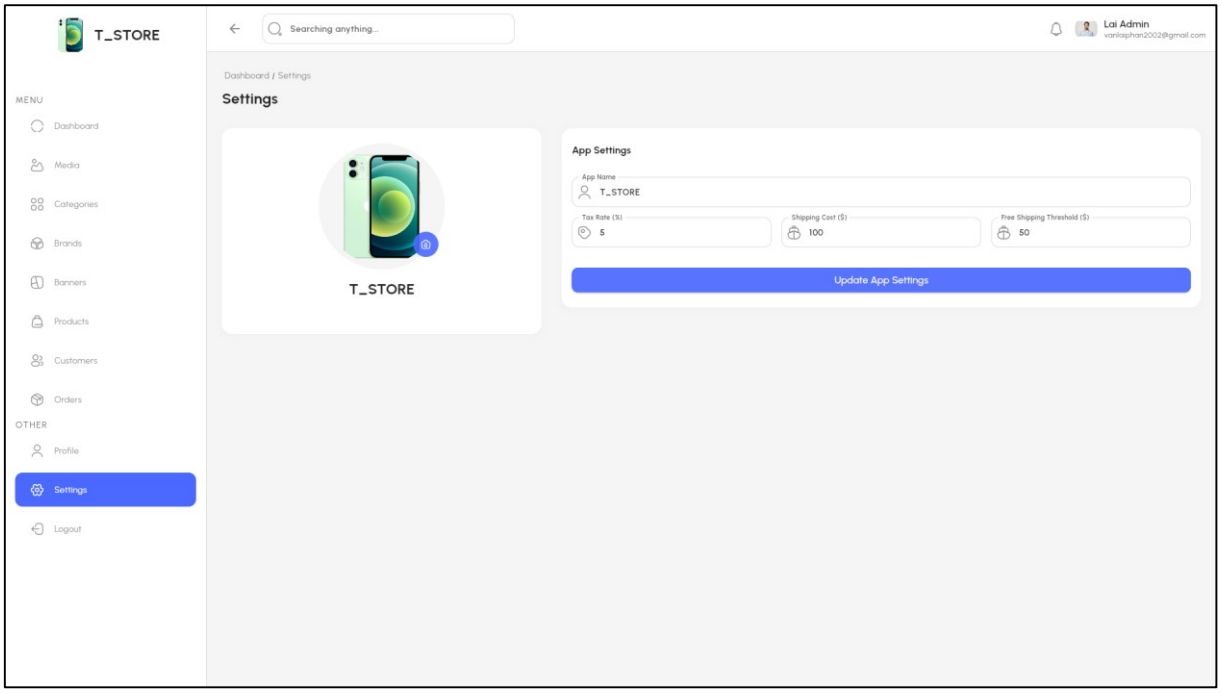

Figure 2.86 "Setting" interface

Table 2.53 "Setting" interface

Item	Type	Data	Describe
Update image Button	Linked Button	N/A	Update image store
App name	Textbox-String	N/A	Update app name
Tax rate	Textbox-String	N/A	Update tax rate
Shipping cost	Textbox-String	N/A	Update shipping cost
Free shipping threshold	Textbox-String	N/A	Update free shipping threshold
Update app settings	Button	N/A	Update all changes
Act			

Name	Describe	Success	Failure
Click App name	Admin click App name button	Success update app name	

Click Tax rate	Admin click Tax rate to update tax rate	Success update tax rate	
Click Shipping cost	Admin click Shipping cost to update shipping cost	Success update shipping cost	
Click Free shipping threshold	Admin click Free shipping threshold to update phone number	Success update free shipping threshold	
Click Update app settings	Admin click Update app settings	All changes have been updated successfully.	

CHAPTER 3. AI-POWERED PRODUCT IMAGE GENERATION AND SMART FEATURES

3.1 Product Image Generation System Using AI

In the context of modern e-commerce, visually appealing product images play a crucial role in influencing consumer behavior. However, not all small and medium-sized businesses have the resources to hire professional photographers or designers. To address this challenge, this project integrates an AI-based product image generation system using Stable Diffusion and ComfyUI into the e-commerce application.

3.1.1 ComfyUI

ComfyUI is a powerful, flexible, and highly modular node-based graphical user interface designed for working with Stable Diffusion models. Unlike traditional interfaces, ComfyUI allows users to build custom image generation pipelines by connecting functional nodes — such as prompt inputs, model loaders, samplers, image processors, and upscalers — in a clear and visually structured manner. This node-based workflow offers unparalleled control and transparency, making it easy to experiment with and fine-tune every aspect of the image generation process.

One of ComfyUI's most notable advantages is its performance efficiency. It leverages GPU acceleration effectively and supports advanced features such as batch processing, latent image manipulation, image-to-image generation, and more. ComfyUI also enables users to load custom models, workflows, and extensions, making it a favorite among artists, researchers, and AI enthusiasts who require both creative freedom and technical depth.

Whether you're a beginner looking to explore AI-generated art or a developer aiming to prototype complex workflows with precision, ComfyUI provides a robust and visually intuitive environment to unlock the full potential of Stable Diffusion.

3.1.2 Image Generation Workflow Design

This ComfyUI workflow is designed to perform advanced image processing combining text-to-image generation, image manipulation, and multi-stage sampling, utilizing the Flux Kontext model.

3.1.2.1 Model Loading and Initialization

The workflow begins by loading three core components:

- UNet model (flux1-dev-kontext) for image generation,
- CLIP encoders (clip_l.safetensors, t5xxl_fp8) for text embedding,
- VAE model (ae.safetensors) for latent encoding/decoding. These are foundational to the prompt-driven image synthesis process.

3.1.2.2 Image Input & Preprocessing

Two images are loaded from the local system. One image is processed using background removal (RMBG), while the other is preserved. The images are then combined using ImageStitch nodes to build a customized presentation layout.

3.1.2.3 Prompt Handling & Translation

User-written Vietnamese prompts are entered via Text Multiline nodes, then automatically translated to English using GoogleTranslateTextNode. These translated prompts are encoded with CLIP and used for further conditioning.

3.1.2.4 Primary and Secondary Sampling Pipelines

- The main prompt is encoded and passed through ReferenceLatent and FluxGuidance, then sampled with KSampler to generate a new latent image.
- A secondary prompt follows the same process in a parallel branch.
- The latent outputs are stored (SaveLatent) and decoded into final images (VAEDecode).

3.1.2.5 Output & Visualization

The final images are rendered and saved with SaveImage. Intermediate results are visually stitched together, allowing for a clear presentation format using custom layout configurations.

3.1.2.6 Advanced Features

- An optional LoraLoaderModelOnly node is applied to fine-tune the model (e.g., for clothing removal or stylistic transformation).

- A Note node defines the workflow principle: “Remove clothes, then add clothes”, indicating a two-phase transformation logic—useful for virtual try-on, cosplay customization, or AI fashion previews.

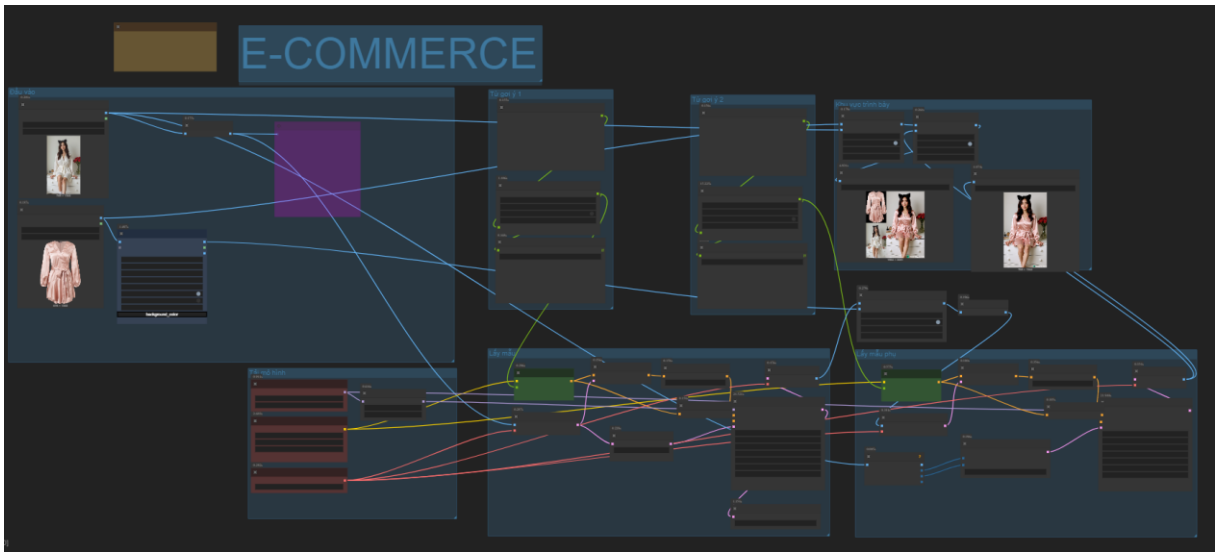

Figure 3.1 Workflow design overall

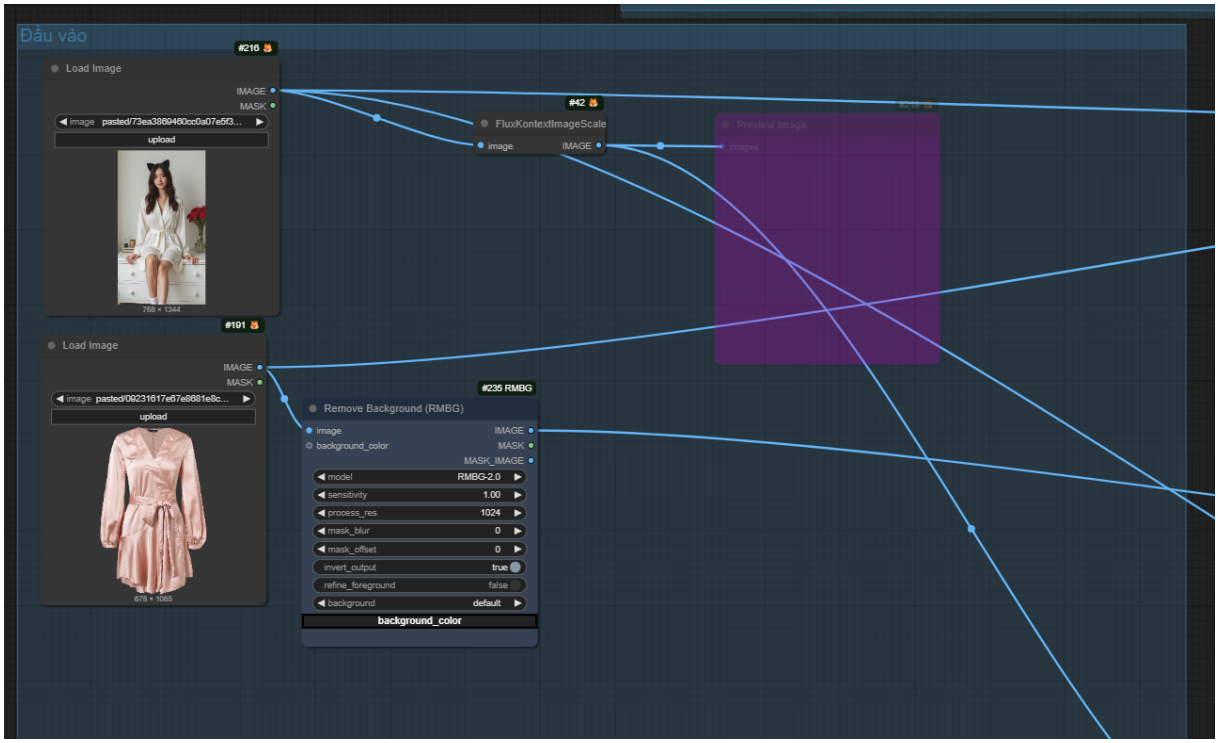

Figure 3.2 Workflow design 1

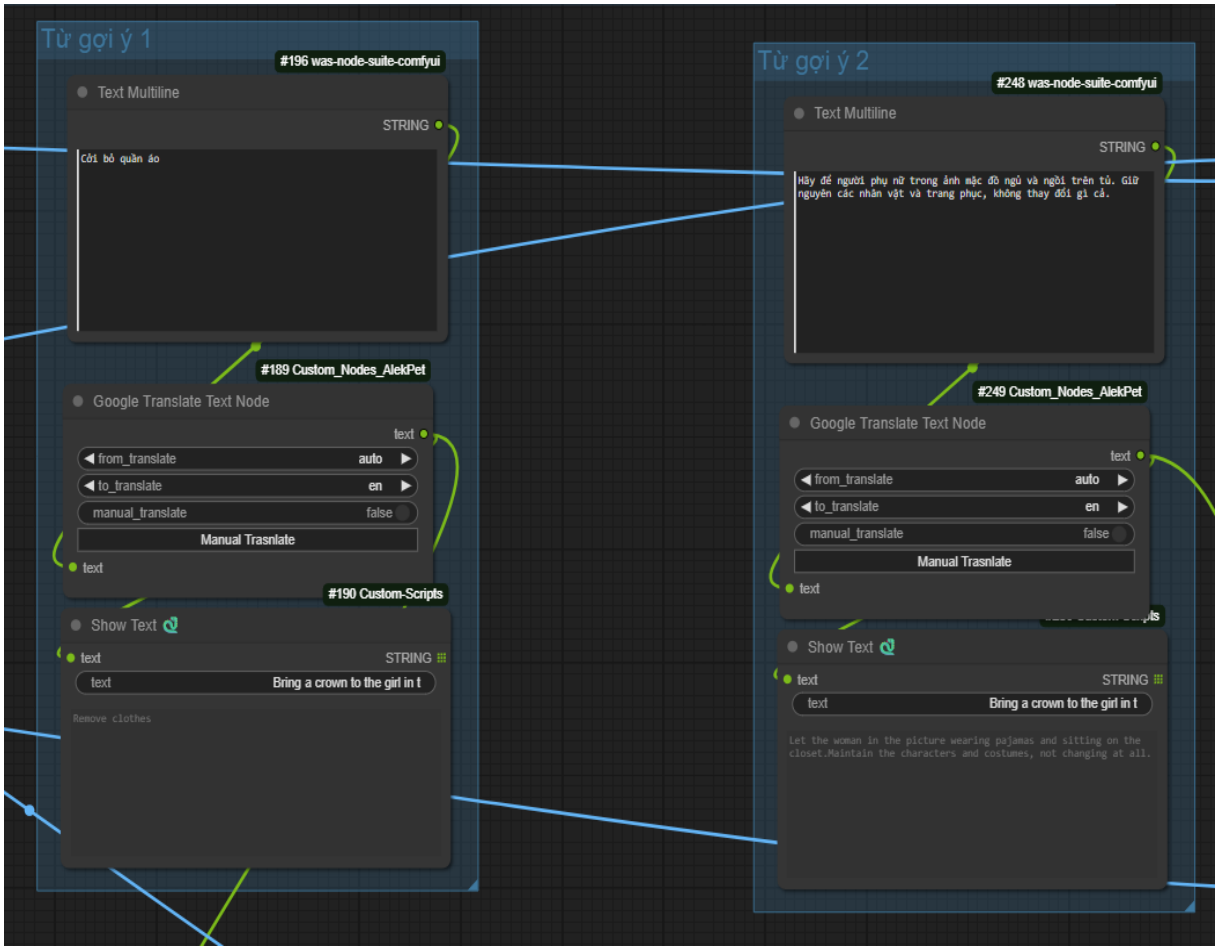

Figure 3.3 Workflow design 2

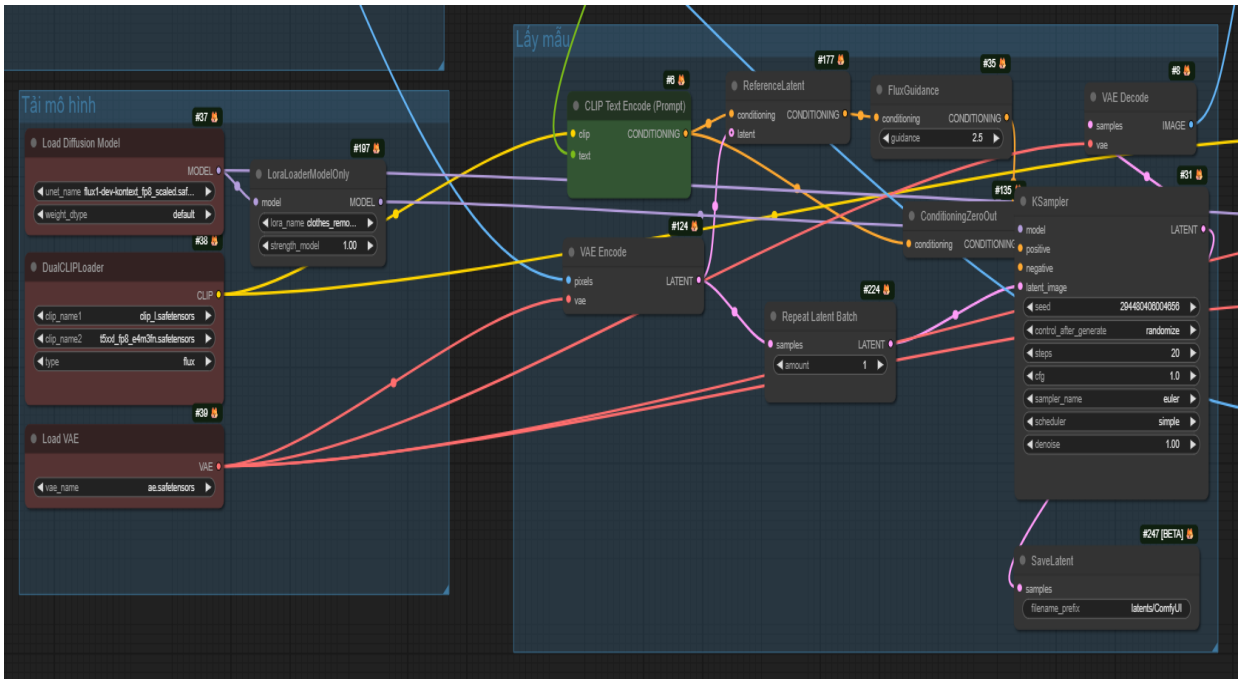

Figure 3.4 Workflow design 3

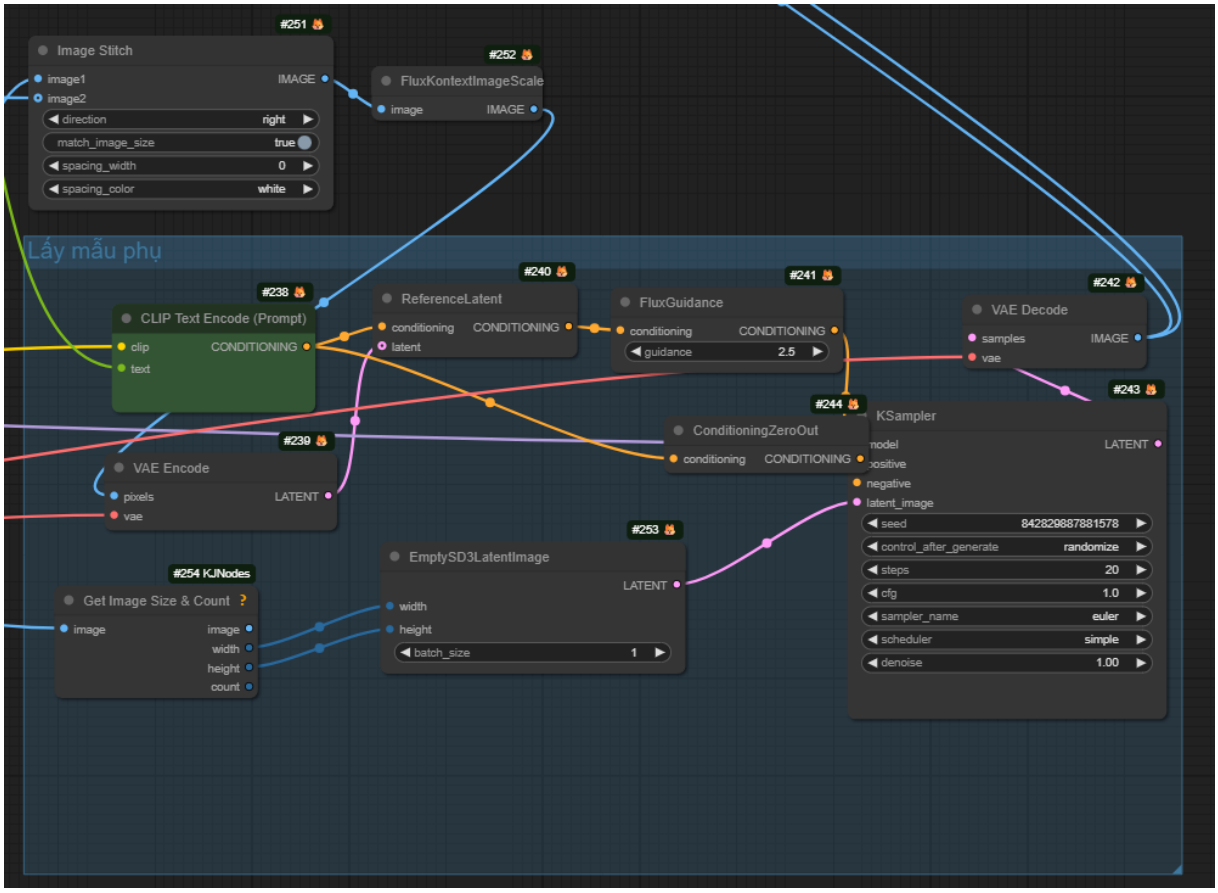

Figure 3.5 Workflow design 4

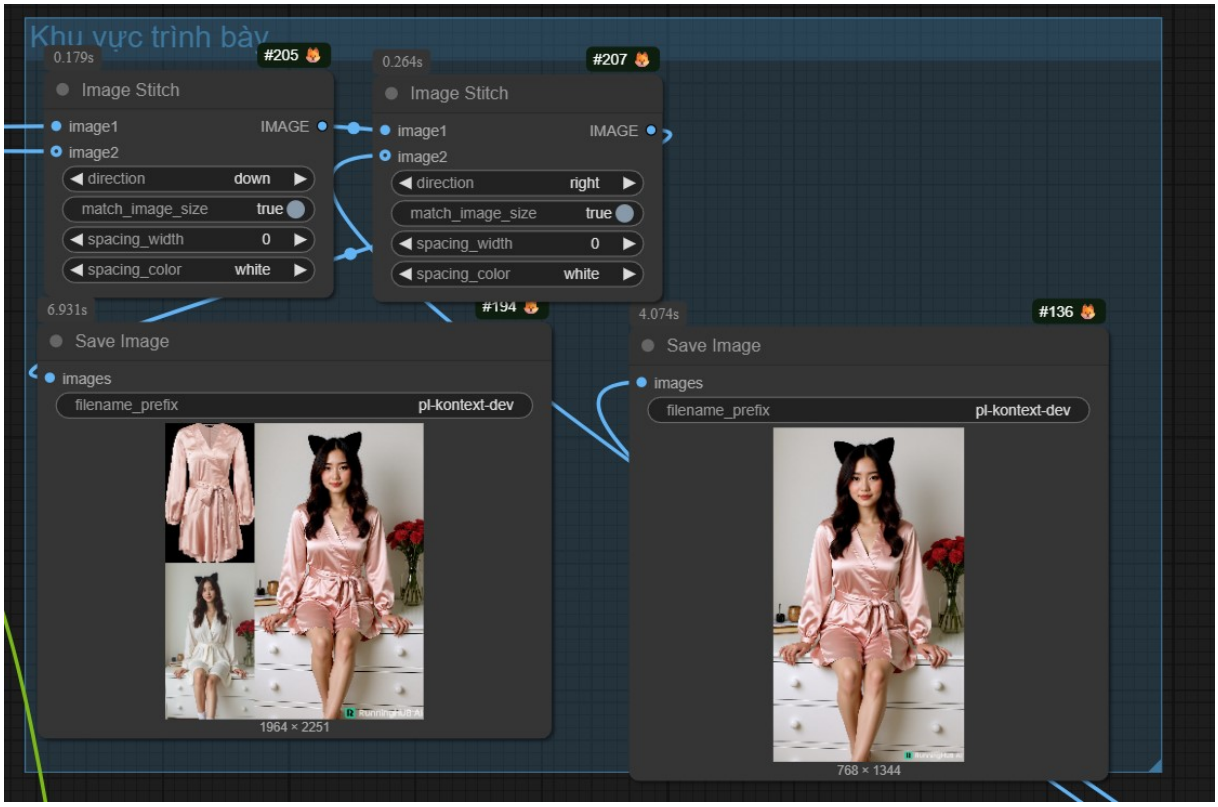

Figure 3.6 Workflow design 5

3.2 AI Screen Design

User selects main and secondary photos, then writes description. Finally click "Start workflow" for AI to follow the description

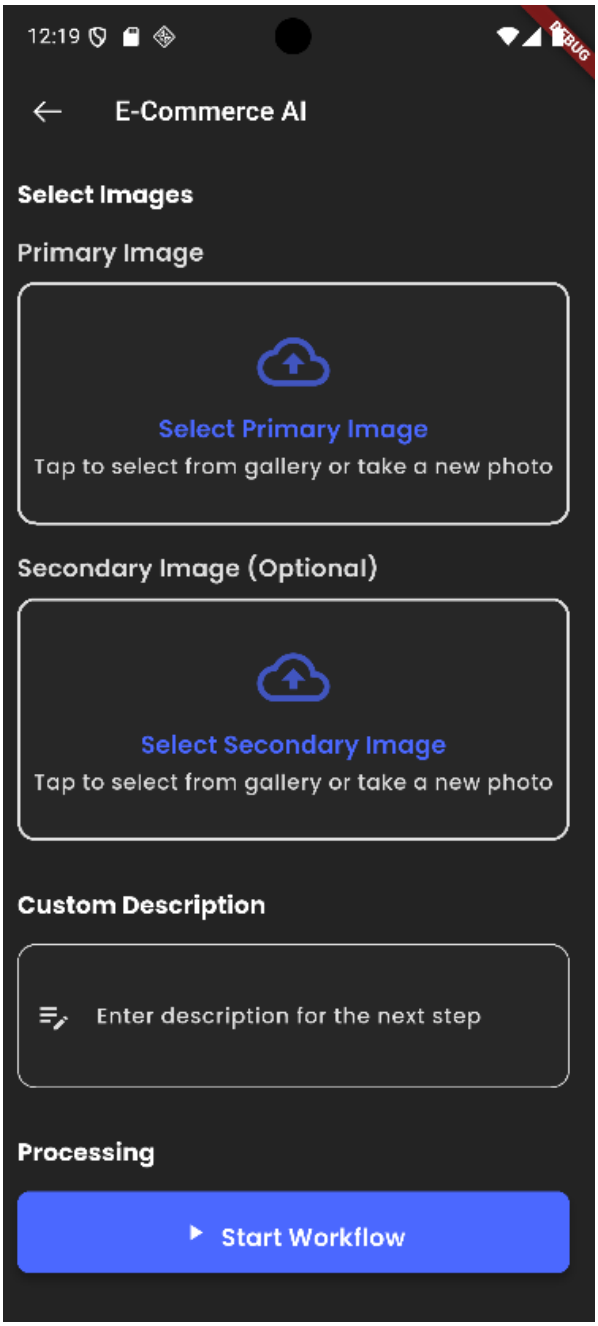

Figure 3.7 AI Image Processing

CONCLUSIONS AND SUGGESTIONS

1. Conclusions

After the process of research and development, the thesis entitled "Building a Cross-Platform E-commerce Application Using Flutter and Firebase" has achieved the following results:

- Successfully developed an e-commerce application that runs smoothly on both Android and iOS platforms.
- Integrated core functionalities such as user registration/login, product management, shopping cart, and basic order processing.
- The system is implemented using Flutter combined with Firebase, ensuring real-time synchronization, stability, and scalability.
- The application has been thoroughly tested through unit testing, integration testing, and user testing.
- The integration of Artificial Intelligence (AI) enhances user experience by improving product recommendations and data management.

From a technical and academic perspective, this project has helped the team to:

- Strengthen and improve programming skills with Flutter, Firebase, UI/UX design, and software project management.
- Effectively apply system analysis and design models in practice.
- Become familiar with incorporating modern technologies such as AI into real-world applications.

2. Suggestions

- In the future, we recommend continuing the development of advanced features such as:
  + A rating and feedback system for products.
  + Product recommendations based on user behavior (AI-based).
  + Integrated payment systems and detailed order tracking.
- Additionally, it is important to further optimize performance and enhance system security to enable the application to be deployed on a larger scale.

REFERENCES

[1] A. Dittmar. “More precise descriptions of temporal relations within task models”. In Interactive Systems: Design, Specification, and Verification, 7th International Workshop DSV-IS, Proceedings, pages 151–168, Limerick,Ireland, 5-6 June 2000.

[2] E. G. Sirer and B. N. Bershad, "Using Production Grammars in Software Testing", in Proceedings of the Second Conference on Domain Specific Languages, Austin, Texas, October 3-5,1999.

[3] Fewster M., Graham D., (1999), *Software Test Automation – Effective Use of Test Execution Tools, Reading,* MA: Addison – Wesley.

[4] Fujiwara S, Bochman G, Khendek F, Amalou M, Ghedasmi A. “Test selection based on finite state models”. IEEE Transactions on Software Engineering 1991; 17(6):591–603.