Smartshop: AI-Powered Virtual Try-On E-Commerce App

Submitted By

Jubaer Islam Nahin

221-35-1070

Supervised By

Dr. Md. Fazla Elahe

Assistant Professor & Associate Head

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Jubaer Islam Nahin
Date of Birth	:	01-03-2001
Title	:	Smartshop: AI-Powered Virtual Try-On E-Commerce App
Academic Session	:	2022-2025

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my project to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

Nahin
_____
(Student's Signature)

221-35-1070
_____
Student ID
Date: 25-12-25

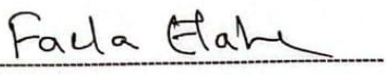
Fazla Elahe
_____
(Supervisor's Signature)

Dr. Md Fazla Elahe
_____
Name of Supervisor
Date: 25-12-25

APPROVAL

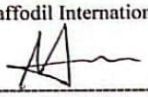
This project titled on "**Smartshop: AI powered E-Commerce App**", submitted by **Jubaer Islam Nahin (ID: 221-35-1070)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



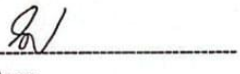
Chairman

Dr. Fazla Ealhe
Assistant Professor & Associate Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 1

Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

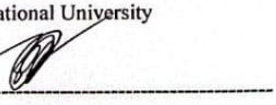


Internal Examiner 2

Dr. Shabnom Mustary
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Md. Rajib Mia
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



External Examiner

Mohammod Abul Kashem, PhD
Professor
Department of Computer Science and Engineering
DUET, Bangladesh

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

(Supervisor's Signature)

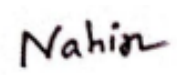
Full Name : Dr. Md. Fazla Elahe

Position : Assistant Professor & Associate Head

Date : 25 December 2025

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.



(Student's Signature)

Full Name : Jubaer Islam Nahin
ID Number : **221-35-1070**
Date : 25 December 2025

Smartshop: AI Powered Virtual Try-On E-Commerce App

Jubaer Islam Nahin

ID: 221-35-1070

Project submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science/Master of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

November 2025

ACKNOWLEDGEMENTS

First and foremost, I express my gratitude to The Almighty Allah for providing me with the strength, knowledge, and opportunity to successfully complete this Smartshop project.

I want to say I would like to express my profound gratitude to my supervisor, [Dr. Md. Fazla Elahe] of. the Department of Software Engineering, to her professional advice, priceless recommendations, and constant support in this venture. Her assistance has been significant. part in the construction of this work. I also would like to say my personal thanks to all the people who participated in the surveys. and testing stages of this project. Their eagerness and positive participation. Feedbacks were needed in the validation and improvement of Smartshop.

I would like to take this moment to acknowledge all the faculty members of the Department of Software Engineering for their guidance, motivation, and support during my academic journey.

Lastly, I would like to express my profound gratitude to my parents and my well wisher for their unconditional love, patience, and support. Without their encouragement and understanding, the completion of this project would not have been achievable.

DEDICATION

I therefore declare that I have done this project under the oversight of "**Dr. Md. Fazla Elahe**", **"Assistant Professor & Associate Head"**, Department of Software Engineering, Daffodil International University. Also declare that neither entire record nor any portion of this record has been submitted somewhere else for my degree.

ABSTRACT

E-commerce is currently emerging as one of the rapidly developing sectors in the globe that are transforming how individuals browse, buy and relate to products. Customers have become increasingly impatient and demanding in their desire to receive more rapid services, intelligence, and personalization in their shopping experience since the emergence of digital platforms. Meanwhile, companies are seeking applications that can assist them to conduct their operations effectively without necessarily having to engage in manual labor in the process. AI gradually becomes a major component of online shopping, which opens the opportunities of such aspects as virtual assistance, individual recommendations, and intelligent visualization of products. These tendencies indicate the obvious tendency towards more intelligent and interactive online shopping, which generates a significant demand in platforms that would be supportive of the convenience and intelligent automation.

Table Of Contents

ACKNOWLEDGEMENTS **II**

ABSTRACT **III**

CHAPTER 1 **1**

INTRODUCTION **1**

1.1 Background 1

1.1.1 Context and Relevance 1

1.1.2 Problem Identification 1

1.1.3 Purpose and Justification 1

1.1.4 Scope 2

1.2 Project Planning and Initiation 2

1.3 Target User Profile and Tentative Elicitation Process 4

1.1.5 1.3.1 Target User 4

1.1.6 1.3.2 User profile 5

1.3.3 Elicitation Process 7

1.4 Project Block Diagram 8

1.5 System Requirements 8

1.5.1 Hardware Requirements 8

1.5.2 Software Requirements 9

1.5.3 Constraints and Dependencies 9

1.6 Project Scheduling 9

1.7 Summary 11

CHAPTER 2 DESIGN AND IMPLEMENTATION **11**

2.1 Introduction 11

2.2 Functional Requirements 11

2.3 Non-Functional Requirements 14

2.3.1 Performance 14

2.3.2 Reliability 14

2.3.3 Portability 14

2.4 Object-oriented System design using UML 17

2.4.1 Use Case Diagram 17

2.4.2 Case Description 18

2.4.3 Activity Diagram 36

2.4.4 Sequence Diagram 44

2.4.5 Class Diagram 54

2.4.6 ER Diagram 55

2.5 Coding: Appendix A 55

CHAPTER 3 **56**

SOFTWARE TESTING **56**

3.1 Introduction 56

3.2 Testing Features 56

3.3 Testting Stretagies 57

3.4 Test Cases 58

3.5 Summary 63

CHAPTER 4 **63**

CHAPTER 5 **65**

USER MANUAL **65**

5.1 Introduction 65

5.2 Project Functionalities 65

5.3 Summary 75

REFERENCES **77**

APPENDICES 78

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Context and Relevance

E-commerce is now among the most rapidly developing industry in the world and has transformed the manner in which individuals shop, browse and socialize with goods. Customers want quicker services, smarter service, and more self-centred shopping experiences with the emergence of digital platforms. Meanwhile, companies are in need of the tools that would allow them to work effectively without constant manual work. The inclusion of artificial intelligence in online retail is gradually emerging as one of the most important elements of this industry, which provides possibilities in the form of virtual assistance, personal recommendations, and intelligent product visualization. These trends have indicated a distinct inclination towards smarter and more interactive online shopping settings generating a high demand on platforms that are both convenient and intelligently automated.

1.1.2 Problem Identification

Even though the online shopping is growing, several of the current online shops cannot offer real seamless customer experiences. Delays in responding to users, product ambiguity, limited customization, and doubts on how a product will be applied in real life are some of the common challenges encountered by users. At the business level, the store owners have to rely on employees who do not always have time and this influences customer service and the overall performance of sales. Other e-commerce solutions have tried to solve these problems, yet majority of them still accept the use of static interface and simplistic features. They have no real-time support, proper visualizing products, and independent systems which operate even when their owners are not online. These weaknesses point out the existence of a smarter and more receptive online shopping system.

1.1.3 Purpose and Justification

Smartshop is aimed at presenting a new generation of e-commerce that can provide a better customer and business experience with the help of AI. The project will streamline the process of online shopping by incorporating functions like a chatbot 24/7, intelligent product recommendation and virtual try-ons, among others, to make the process more comfortable, engaging, and trustworthy. Not only do these features aid in making better decisions by the customers but also lower the rate of returns and enhance satisfaction. To the owners of stores Smartshop offers unlimited automated help so that they can run their business more effectively without the use of human workers only. The platform is valuable because it closes the gap between existing e-commerce constraints and the need to increase the availability of smart service provision.

1.1.4 Scope

Smartshop is a broad area of features to the customers and the administrators. On the customer side, the system consists of browsing and product viewing, management and cart, order placement, and AI-enhanced customer services, including personalized suggestions and virtual try-on. In the administrative department, the system has product management and inventory update and order tracking tools. The project is aimed at providing an easy, intelligent, and automated way to shop at a cost that can be realistically developed. The scope also makes sure that Smartshop will operate as a full e-commerce system that is supplemented with smarts that will enhance the overall shopping experience.

1.2 Project Planning and Initiation

Phase 1 Preliminary Analysis & Project Scope Definition:

Smartshop seeks to solve the issue of the gap between the basic stage of online shopping and the advanced fashion guidance by offering a personalized shopping experience, which is enabled by artificial intelligence. It has a mobile and web application, which uses the flutter technology and provides features such as virtual try-on, size prediction, outfit suggestions, visual search, real-time inventory updates, and a chatbot to assist customers. The aim is to reduce the number of products returned due to improper sizing, provide fashion advice via AI, and reduce the shopping process.

Smartshop allows customers to view, search and purchase fashion products with such opportunities as 360deg product images, personal recommendations, search of images and so forth. The key stakeholders in this project are end users, shop owners, development team and designers.

Phase 2 Market Feasibility Analysis (or Market Research):

Target market: The youthful group of 16-40 years old, technologically advanced, and liking to shop over the Internet, and who take a particular interest in fashion.

Trends: Artificial Intelligence in Shopping and Fashion Technology is on the Increase. Expansion in Mobile Shopping and Visual Commerce. The customers are more apt to customised and fast shopping experiences. The systems offered by online markets like Amazon, Zalando, and Asos offer the recommendation system, yet they are not fully integrated with such features as visual search and chatbots based on artificial intelligence.

Opportunity: The inclusion of various intelligent capabilities into a small application. Serving particular target audiences in terms of size estimation and profile in the shape of an avatar.

Phase 3 Technical Feasibility Analysis:

Technology Stack:

- Flutter for cross-platform UI
- Firebase
- Firestore/SQL database for data handling
- REST APIs for communication

ML/AI stack:

- Tensorflow lite for on-device inference.
- Server-side ml model training and deployment.

Feasibility: Prediction and recommendation tools based on size can be trained on publicly available data or it may be trained on synthetic data in case user data is not available at the outset. The two techniques that are frequently applied in visual search are feature extraction and nearest neighbour matching.

Dangers: Accuracy of machine learning models. Integrating several of the large-scale images methods. Mobile Environment Performance Optimization.

Phase 4 Financial Feasibility Analysis:

Development costs: Flutter and firebase lower first-time expenses for infrastructure and development.The utilization of open-source libraries and tools is pushed as much as possible.

Revenue models:

- Earning money through affiliate marketing or commissions from product links.
- In-app purchases (add-ons or early access).
- AI-Powered Fashion Designer (optional).
- Brand Partnerships & Sponsored Content.
- Commission on Vendor Sales.
- Subscription-Based AI Assistant.

The initial investment that would be needed on the platform is quite low, yet the chances of users actively using the platform and generating revenue in numerous ways of monetization are high.

1.3 Target User Profile and Tentative Elicitation Process

1.1.5 1.3.1 Target User

Smartshop caters to a wide range of fashion-forward individuals who prioritize convenience, personalization, and smart shopping experiences. The main demographic groups that we focused on are:

1) Young Adults and Students (Ages 16–25)

2) Working Professionals (Ages 26–40)

3) Fashion Enthusiasts and Influencers etc.

1.1.6 1.3.2 User profile

Table 1: User Profile for Customers

User Class	**Note on Characteristics**
Type of user	Customer
Age range	18-75
Frequency of use	When it is needed
Mandatory	No
Computer experience	No
Education	Minimal
Goal	Take a good service
Language skills	Bangla, English
Number of users	Many
Training	No
Others system use	No
Way of working	

Table 2: User Profile for Authority

User Class	**Note on Characteristics**
Type of user	Authority
Age range	30-45
Frequency of use	Most of the time per day
Mandatory	Yes
Computer experience	Experienced
Education	Graduated
Goal	Keep track of the system functionalities and smooth progress of the business

Language skills	Bangla, English
Number of users	1
Training	May accept some training but be unwilling to repeat it
Others system use	No
Way of working	Full support from the system

Table 3: User Profile for Product Manager

User Class	**Note on Characteristics**
Type of user	Product Manager
Age range	30-45
Frequency of use	Most of the time per day
Mandatory	Yes
Computer experience	Experienced
Education	Graduated
Goal	Provide the best service
Language skills	Bangla, English
Number of users	1-2
Training	May accept some training, but be unwilling to repeat it
Others system use	No
Way of working	Full support from the system

Table 4: User Profile for Employee

User Class	**Note on Characteristics**
Type of user	Employee
Age range	22-45
Frequency of use	Most of the time per day
Mandatory	Yes
Computer experience	Not necessarily

Education	Minimum SSC passed
Goal	Provide the best service
Language skills	Bangla, English
Number of users	5-10
Training	May accept some training, but be unwilling to repeat it
Others system use	No
Way of working	Packaging, delivering and maintaining the orders

1.3.3 Elicitation Process

To guarantee that the features align with the actual needs of the users, Smartshop employs a user-cantered design approach that incorporates structured elicitation techniques such as:

1) Online Surveys
2) User Interviews
3) Competitor Feature Analysis
4) Prototyping & Feedback
5) Social Listening & Trends

1.4 Project Block Diagram

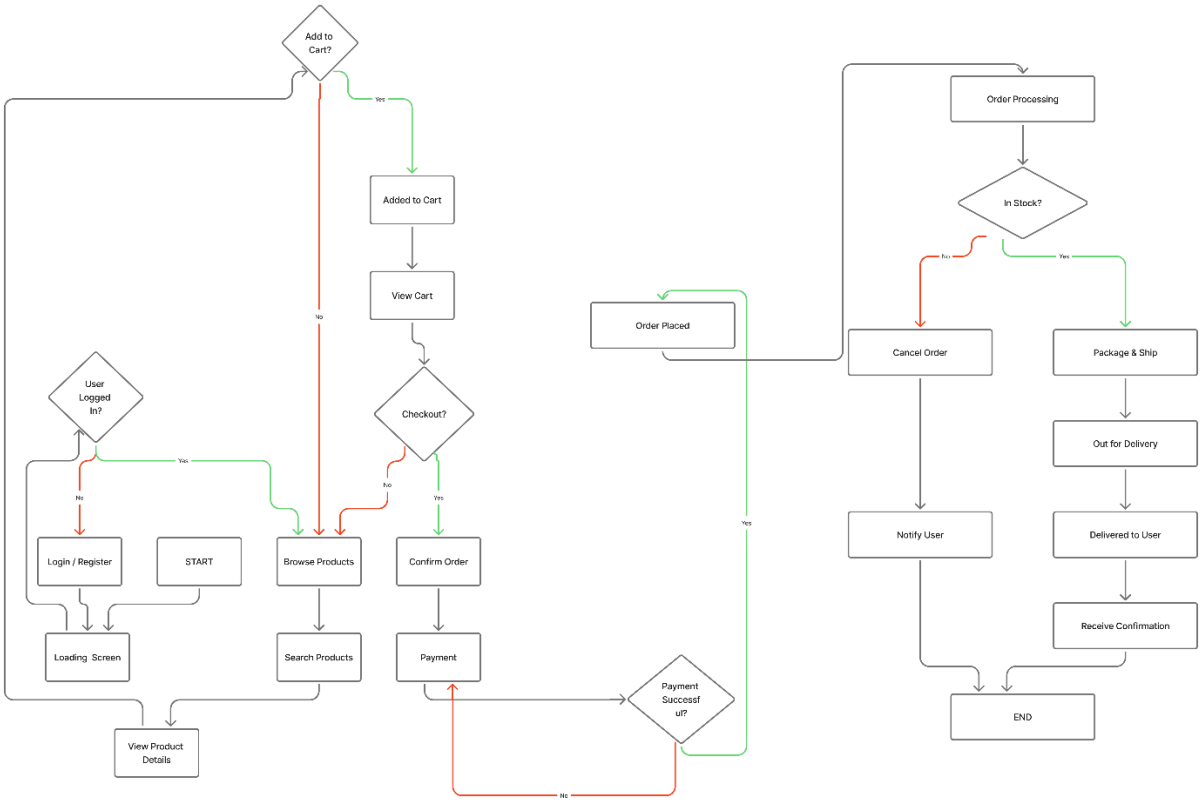


Figure 1: System Block Diagram

1.5 System Requirements

1.5.1 Hardware Requirements

To run Smartshop smoothly, the system does not demand heavy or specialized hardware. A standard device with moderate specifications is sufficient for both development and usage. For users, any modern smartphone, tablet, or computer with stable internet

connectivity is enough to access the platform. For administrators or developers, a computer with at least 8 GB of RAM, a multi-core processor, and 10–20 GB of free storage is recommended to comfortably run the development tools, database services, and testing environments. Since most of the processing is handled online, the overall hardware requirements remain minimal and accessible for general users.

1.5.2 Software Requirements

Smartshop is based on the set of regularly used and trusted software. On the development front, it will need Flutter SDK, Android Studio or VS code, and the back-end solution, like Firebase or MySQL. To be deployed, the system will require a secure hosting environment and a backend service which can handle the real time interactions with the users. The application can be accessed with just a modern web browser or an updated Android/iOS operating system by the end-users. The software components employed in the project are highly supported and therefore the system is easily maintained and it is flexible to the current industry standards.

1.5.3 Constraints and Dependencies

Even though Smartshop has smart functions, it has some limitations and dependencies. The chatbot, the recommendation engine, the virtual try-on and others are features that require stable internet connection and robust backend services otherwise performance can be impacted. The system also relies on APIs externally in order to perform functions such as AI processing and authentication, and therefore any disruption or downtime of such services will have a temporary effect on functionality. Moreover, because the platform is mostly designed to work in a mobile and web-based environment, older gadgets and older operating systems might not function with all functionalities without any difficulty. In spite of these limitations, Smartshop is intended to work effectively on these limitations but offer an improved shopping experience.

1.6 Project Scheduling

Time Frame:

Phase	Duration	Description
Requirements	15 days	Requirement Gathering – Identify features, user needs, and key project goals.
Feasibility & Planning	15 days	Analyze market, technical, and financial feasibility; finalize tech stack.
Design	20 days	Design wireframes, mockups, and app UI flow based on user-centric principles.
Architecture & UML	20 days	Create UML diagrams, system architecture, and database schema.
Development – Part 1	30 days	Implement core features: user auth, product catalog, wishlist, and cart.
Development – Part 2	30 days	Integrate advanced features: product detail view, checkout, and payment system.
AI Features	30 days	Add Size Prediction, Visual Search, Recommendations, and Chatbot functionality.
Testing & QA	25 days	Conduct testing (unit, integration, UI/UX), fix bugs, and refine performance.
Deployment	15 days	Deploy app to Android, iOS, and Web platforms; set up CI/CD and hosting.
Documentation	15 days	Prepare technical documentation, user manual, and final project report.

1.7 Summary

This chapter provided the background of the Smartshop system, which includes setting the context within which the project was worked out, the issues that the project will solve, and the benefit it will bring to the customers and business owners. It also discussed the increasing trends toward smart and convenient e-commerce and outlined the gaps that exist in the current platforms as well as how Smartshop intends to fill these gaps with the functions of AI-assisted, personalized recommendations, and virtual try-ons, among other things. The scope of the project, the feasibility analysis, and the system requirements have also been introduced in the chapter and provided insight into the technical and practical aspects behind the system. In general, the chapter preconditions the detailed specifications, which will be further discussed, and the overall vision of why Smartshop is topical, needy, and possible in the digital contemporary world.

CHAPTER 2 DESIGN AND IMPLEMENTATION

2.1 Introduction

The chapter introduces the structural and behavioral design of the Smartshop system which transforms the requirements that have been discussed above into specific technical models. It presents the functional and non-functional requirements that would determine how the system behaves, and thereafter UML diagrams that include using case diagram and case descriptions which demonstrate how the system works within itself. Also activity diagrams, sequence diagram, class diagram and ER diagram that each are used to explain various aspects of system interaction and workflow. This chapter, together with the implementation approach, gives a clear understanding of how the Smartshop is designed, how the various parts interact with each other, and how the system has attained its desired functionality in a well-organized and dependable way.

2.2 Functional Requirements

FR01	Sign Up
Description	Sign Up offers people to establish accounts, which guarantees them of customized functions and convenience in conducting their shopping in

	the system. It entails safe retrieval of user data in the creation of accounts.
Stakeholder	Customer, Product Manager

FR02	Sign in
Description	Sign In allows the user to access his or her account safely which unlocks the use of customized features and provides them with a convenient shopping experience within the system. It also entails user authentication to the account.
Stakeholder	Customer, Product Manager, Authority

FR03	Forget Password
Description	Forget Password enables a user to change his or her account password in case of failure to log in. The system will authenticate the user by email or phone, provide a reset link or OTP and will allow the user to set a new secure password. This will guarantee that accounts are recovered without jeopardizing security.
Stakeholder	Customer, System Administrator, Product Manager

FR04	View Product
Description	View Product will enable the users to discover the information, photos, and specifications of shoes in the system which will contribute to a better shopping experience. It is the provision of full-scale information to make well-informed buying choices.
Stakeholder	Customer, Product Manager

FR05	Add Product
Description	Add Product allows the authorized employees to add a new product to the system increasing the number of footwear. It entails adding product information, pictures, and stock to view and make purchases.
Stakeholder	Authority, Product Manager

FR06	Update Product

Description	Update product enables the authorized people to change information, including details, images, and inventory levels, which guarantees the right presence in the system. It entails dynamics in real time to maintain the product-related information up to date and pertinent.
Stakeholder	Authority, Product Manager

FR07	Search Product
Description	Search Product enables customers to find footwear products quickly by typing keywords, category filters, sizes, price filters or sorting. It enhances navigation and efficient purchase of certain products by customers.
Stakeholder	Customer, Product Manager

FR08	Add to Cart
Description	Add to Cart allows customers to save the chosen items in a shopping cart so that they can review them and then purchase them. It allows to update the quantities, delete items, and see the overall cost, and it also allows users to control their selection prior to checkout.
Stakeholder	Customer, Product Manager

FR09	Make Payment
Description	Make Payment supports the online commerce that is safe and smooth to users who make purchases of the footwear in the system. It entails financial transactions processing which makes it easy and secure to make a payment.
Stakeholder	Customer, Authority

FR10	Sign out
Description	Sign Out enables users to safely log out keeping their account information safe and leading to improved system security in the system. It means ending the user session in order to guarantee privacy and unauthorized access.
Stakeholder	Product Manager, Customer, Authority

2.3 Non-Functional Requirements

2.3.1 Performance

To have a smooth shopping experience, Smartshop should offer quick service and promptness in the interaction. The app must be able to load the necessary screens fast and respond to the user actions with no apparent lag. Optimized API calls and appropriate caching policies should be used to render product lists, images and recommendations. The virtual try-on as an aspect of machine learning models should deliver the results within a reasonable time to avoid frustration among users. The system must also take care of the memory and have good performance in the mid-range and high-range devices.

2.3.2 Reliability

Reliability is used to ensure that Smartshop is a stable system that can perform and operate steadily both in the normal conditions and high usage conditions. Minimum downtime will be visible in the application, and the most important features, including user login, addition of item to cart, order placement, and production of try on results must be able to work perfectly without failure. The integrity of data is critical; none of the data must be lost or duplicated in any manner. The system should be capable of graceful handling of unexpected network problems through retrying of operation or displaying of suitable error messages. There should be logging and error-monitoring systems to help to detect and solve problems promptly.

2.3.3 Portability

Portability means that Smartshop can be executed well on various platforms and types of devices. The application is developed with Flutter and should therefore become compatible with both Android and iOS devices across different screen size and resolution. There should be responsiveness in the interface to various layouts so that it can be used in smartphones and tablets. The backend services are also supposed to be platform independent in case they have to be expanded in the future into web or desktop version in case it is necessary. In general, Smartshop is supposed to be simple to roll out, upgrade, and scale to a variety of venues.

2.3.4 Usability

Smartshop must be user friendly, simple and easy to navigate. The interface has to make product information readable, fonts need to be readable, icons have to have a meaning and few steps have to be required to carry out activities like searching, adding products to the cart or checking out. Error messages and prompts should be easily understandable to enable the users to easily know what has gone wrong and correct it. A uniform UI pattern should be used in the design to ensure it is natural to both the novice and veteran users.

2.3.5 Security

The security is needed to safeguard the user accounts, personal information, and transactional data. Any communication between the backend and the mobile app should be encrypted by the use of secure knowledge such as HTTPS. Passwords and payment details are some of the sensitive information that needs not to be written in plain text. It should be authenticated by secure mechanisms which operate and make use of tokens and unauthorized access should be eliminated by proper role and permission management. Vulnerabilities should be guarded by regular security checks and updates.

2.3.6 Scalability

Smartshop should be in a position to manage the increase of the users, products and the system operations without affecting the performance. The backend system must be capable of horizontal or vertical scaling to ensure that the application is not sluggish by the increasing traffic particularly during sales or special campaigns. Database query, product search, and recommendations system have to be optimized in such a way that they can be run effectively as the dataset continues to increase with the time.

2.3.7 Maintainability

The system must be simple to upgrade, debug and extend. Smartshop is required to adhere to clean coding, modularity and appropriate separation of concerns in order to allow the developers to fix bugs or add features with the least amount of effort. Appropriate documentation- API references, code comment and architecture diagrams- needs to be upheld. Changes must be tracked with version control (e.g. Git) to facilitate a smooth collaboration between developers.

2.3.8 Availability

Smartshop needs to be available to the user whenever required. The system should strive to achieve high availability, which is an element that will reduce the downtime during updates or maintenance. Important operations such as login, payment processing, product retrieval as well as the virtual try-on APIs should continue functioning even when there are high traffic. There should be back up systems and failover systems to enable the application to be available in case any component fails.

2.3.9 Efficiency

Smartshop needs to utilize system resources (memory, network bandwidth, battery power) in an efficient manner. Background activities should also be kept to a minimum in order to lower battery usage of devices. Data usage should be reduced by image compression, caching and optimized network requests, to reduce loading time. The backend should also be able to streamline the use of CPU and storage in order to provide fast responses.

2.3.10 Compatibility

Smartshop should work smoothly with various OS versions, device models, and third-party services. The app must support commonly used Android and iOS versions and integrate properly with services such as Firebase, payment gateways, and ML model

APIs. The system should remain functional even when external APIs undergo minor updates or changes.

2.4 Object-oriented System design using UML

2.4.1 Use Case Diagram

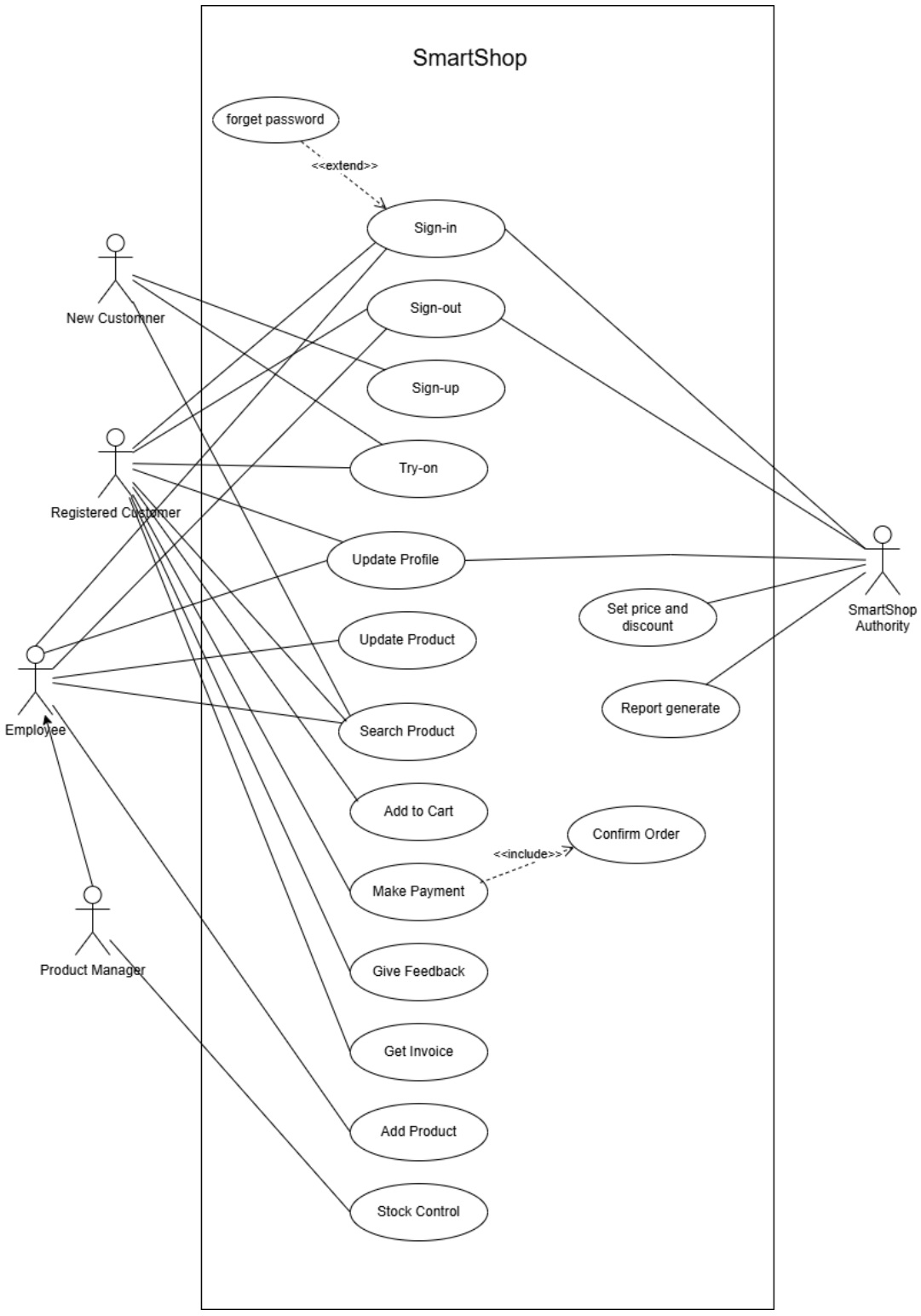


Figure 2: Use case Diagram

2.4.2 Case Description

Case Description-01: Sign Up

<table>
<tr><td>Use Case</td><td>Sign Up</td></tr>
<tr><td>Goal</td><td>Users can Sign Up to sign in to the system.</td></tr>
<tr><td>Precondition</td><td>Users must install the Smartshop app for signing up.</td></tr>
<tr><td>Success End Condition</td><td>Notification: Successfully signed up!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Submission Not Submitted”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>New Customer
Authority</td></tr>
<tr><td>Trigger</td><td>Users will request a registration form to fill up</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press “Sign Up” Button</td></tr>
<tr><td>2.</td><td>Provide Sign Up form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Submit” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them!!!
Successfully Signed Up!!! Notify</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>2.1</td><td>System doesn’t work</td></tr>
<tr><td></td><td>2.1.a. Try again later!</td></tr>
<tr><td>4.1</td><td>The visitor did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. checked by the system & notify the “Please fill up the box”</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Details did not Save”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Case Description-02: Sign In

<table>
<tr><td>Use Case</td><td>Sign In</td></tr>
<tr><td>Goal</td><td>Users can enter by signing in to the system.</td></tr>
<tr><td>Precondition</td><td>Users must be signed up first.</td></tr>
<tr><td>Success End Condition</td><td>Notification: Successfully Signed in!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Sign in failed”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer, Employee, Product Manager</td></tr>
<tr><td>Trigger</td><td>Users will request a sign in to enter the system.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press “Sign in” Button</td></tr>
<tr><td>2.</td><td>Provide Sign in interface</td></tr>
<tr><td>3.</td><td>Enter User id and password</td></tr>
<tr><td>4.</td><td>Users provide the “Sign in” button.</td></tr>
<tr><td>5.</td><td>Verified and Signed in.</td></tr>
<tr><td>6.</td><td>Notification: “Sign in Successful”.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>Server not found</td></tr>
<tr><td></td><td>4.1.a. Try Again!!</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>Information Error!!</td></tr>
<tr><td></td><td>6.1.a. Notification: “Enter the correct credentials”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Case Description-03: Forget Password

<table>
<tr><td>Use Case</td><td>Forget Password</td></tr>
<tr><td>Goal</td><td>Users can restore forgotten passwords.</td></tr>
<tr><td>Precondition</td><td>The user must sign in to the system.</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Password restored successfully!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Password restore failed”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer, Employee, Product Manager</td></tr>
<tr><td>Trigger</td><td>User will request to restore the pin</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Sign in to the system</td></tr>
<tr><td>2.</td><td>Select the forget password” option</td></tr>
<tr><td>3.</td><td>Enter the registered number.</td></tr>
<tr><td>4.</td><td>Press “Send verified code” Button.</td></tr>
<tr><td>5.</td><td>Enter the verified code.</td></tr>
<tr><td>6.</td><td>Set new password.</td></tr>
<tr><td>7.</td><td>Press “Save & Apply.”</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Server down</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>2.1</td><td>Not responding</td></tr>
<tr><td></td><td>2.1.a Reload.</td></tr>
<tr><td>3.1</td><td>Wrong phone number</td></tr>
<tr><td></td><td>3.1.a. Notification: “Choose the valid phone number”</td></tr>
<tr><td>5.1</td><td>Wrong OTP code</td></tr>
<tr><td></td><td>5.1.a. Notification: “Choose the valid OTP code”</td></tr>
<tr><td>6.1</td><td>System error.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Case Description-04: Update Profile

<table>
<tr><td>Use Case</td><td>Update Profile</td></tr>
<tr><td>Goal</td><td>Users can update their profile if they need any changes on the profile.</td></tr>
<tr><td>Precondition</td><td>Must visit profile information.</td></tr>
<tr><td>Success End Condition</td><td>Notification "Profile Information Updated"</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Profile Information not Updated”</td></tr>
<tr><td>Primary Actors</td><td>Customer, Product Manager, Employee</td></tr>
<tr><td>Secondary Actors</td><td>Authority</td></tr>
<tr><td>Trigger</td><td>The user will request to update the profile.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>Press “Registration” Button</td></tr>
<tr><td>2.</td><td>Provide registration form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Submit” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them !!!
Successfully Registered!!! Notify</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Checked By the system & Notify by “Please! Fill Up the Box”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Details did not Save”</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Users will get 30 mins to fill up.</td></tr>
</table>

Case Description-05: Add To Cart

<table>
<tr><td>Use Case</td><td>Add to Cart</td></tr>
<tr><td>Goal</td><td>Allow users to add a selected product to their shopping cart.</td></tr>
<tr><td>Precondition</td><td>Users have to register & Sign in their shopping cart.</td></tr>
<tr><td>Success End Condition</td><td>The selected product is successfully added to their user’s shopping cart</td></tr>
<tr><td>Failed End Condition</td><td>The selected product fails to be added to the cart, and the user receives an error notification.</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Register User
Authority</td></tr>
<tr><td>Trigger</td><td>User will request to add to cart.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1.</td><td>User Visit Dashboard</td></tr><tr><td>2.</td><td>Select “Add To Cart”</td></tr><tr><td>3.</td><td>View cart</td></tr><tr><td>4.</td><td>Edit card</td></tr><tr><td>5.</td><td>Check Out</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>2.1</td><td>System Error</td></tr><tr><td></td><td>2.1.a. Try again</td></tr><tr><td>4.1</td><td>Continue Shopping</td></tr><tr><td></td><td>4.1.a. Explore Different Footwear</td></tr></table></td></tr>
</table>

Quality Requirements	The system should add the product to the cart within 3 seconds to ensure a smooth user experience.

Case Description-06: Give Feedback

Use Case	Give feedback
Goal	To provide feedback on users purchased products
Precondition	The user is logged into their accounts & has purchased at least one product
Success End Condition	User feedback is successfully submitted and stored in the system
Failed End Condition	User fails to be submitted and shows an error message
Primary Actors: Secondary Actors:	Registered user Authority
Trigger	User selects the “Give Feedback” option on the purchased product page
Description / Main Success Scenario	1. User visits Dashboard 2. Select “Request to Feedback” option 3. System displays a feedback form 4. User fills out feedback option 5. Press “Submit Feedback” 6. Feedback successfully submitted

Alternative Flows	

2.1	System Error
	2.1.a. Try Again
5.1	Submission Failed
	5.1.a. Try again

Quality Requirements	Users will get 30 mins to fill up

Case Description-07: Make Payment

Use Case	Make Payment
Goal	To pay bill
Precondition	User must be order something
Success End Condition	Notification “Payment successful.”
Failed End Condition	Notification “Payment Not Successful.”
Primary Actors: Secondary Actors:	Customer Authority
Trigger	Users will request to pay their bill by providing service.
Description / Main Success Scenario	

1.	Visit Dashboard.
2.	Select “Pay Bill”
3.	Enter “Bill ID Number”.
4.	Provide payment option
5.	Enter “Payment Option”
6.	Then pay the bill.

Alternative Flows	2.1 Server Down Again later 3.1 Bill ID not match 3.1.a. Enter the correct Bill ID. 4.1 System Not responds 4.1.a. Please try again later 6.1 System Error 6.1.a. Try again
Quality Requirements	The user Will fill up all the details in 30 minutes.

Case Description-08: Add product

Use Case	Add Product
Goal	To make products available for customers.
Precondition	Users have register & Sign in their account.
Success End Condition	“Product added Successfully” notification.
Failed End Condition	“Product addition failed” notification.
Primary Actors: Secondary Actors:	Authority, Employee
Trigger	Primary actors will add their products tot make them visible for their customers.
Description / Main Success Scenario	1. Request to login. 2. Go to the dashboard. 3. View product list. 4. Select product category. 5. Add expected product. 6. Press “Submit” Button.

Alternative Flows	<table><tr><td>1.1</td><td>Login Failed</td></tr><tr><td></td><td>7.1.a Try again.</td></tr><tr><td>3.1</td><td>Product list empty.</td></tr><tr><td></td><td>3.1.a Refresh again</td></tr><tr><td>4.1</td><td>Category list empty.</td></tr><tr><td></td><td>4.1.a Try again.</td></tr><tr><td>6.1</td><td>Submission failed.</td></tr><tr><td></td><td>6.1.a Try again.</td></tr></table>
Quality Requirements	Must add product information within 1 hour.

Case Description-09: Update Product

Use Case	Update product
Goal	To update new available products in stock.
Precondition	Users have to enter the new available products.
Success End Condition	Notification “Update Successfully”.
Failed End Condition	Notification: “Update Failed”
Primary Actors: Secondary Actors:	Employee, Manager
Trigger	Users will request to update new products.
Description / Main Success Scenario	<table><tr><td>1.</td><td>Request to Sign in.</td></tr><tr><td>2.</td><td>Check info.</td></tr><tr><td>3.</td><td>Go to the Dashboard.</td></tr><tr><td>4.</td><td>View product list.</td></tr><tr><td>5.</td><td>Search product type</td></tr><tr><td>6.</td><td>Update expected product.</td></tr></table>

Alternative Flows	<table><tr><td>1.1</td><td>Sign in failed</td></tr><tr><td></td><td>1.1.a Try Again.</td></tr><tr><td>4.1</td><td>Product list empty.</td></tr><tr><td></td><td>4.1.a Try again</td></tr><tr><td>6.1</td><td>Product not found .</td></tr><tr><td></td><td>6.1.a Search again.</td></tr><tr><td>8.1</td><td>Submission failed.</td></tr><tr><td></td><td>8.1.a Try again.</td></tr></table>
Quality Requirements	User will fill up within 30 minutes.

Case Description-10; Set Price and Discount

Use Case	Set Price and Discount
Goal	To set prices and discounts of the products
Precondition	Employee has to add product in the system.
Success End Condition	Notification: “Price set successfully” or “Discount set failed”.
Failed End Condition	Notification: “Price set Failed” or “Discount set failed”.
Primary Actors: Secondary Actors:	Authority
Trigger	Authority will request a price and discount.
Description / Main Success Scenario	<table><tr><td>1.</td><td>Request to Sign in.</td></tr><tr><td>2.</td><td>Go to Dashboard.</td></tr><tr><td>3.</td><td>Select product list.</td></tr><tr><td>4.</td><td>Search product.</td></tr><tr><td>5.</td><td>Set price.</td></tr><tr><td>6.</td><td>Set discount.</td></tr></table>

<table>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>Sign in failed</td></tr>
<tr><td></td><td>1.1.a Try Again.</td></tr>
<tr><td>4.1</td><td>Product list empty.</td></tr>
<tr><td></td><td>4.1.a Try again</td></tr>
<tr><td>6.1</td><td>Product not found.</td></tr>
<tr><td></td><td>6.1.a Search again.</td></tr>
<tr><td>5.1</td><td>Set price failed</td></tr>
<tr><td></td><td>5.1.a try again</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Users must set price and discount within 30 minutes.</td></tr>
</table>

Case Description-11: Search Product

<table>
<tr><td>Use Case</td><td>Search product</td></tr>
<tr><td>Goal</td><td>To search products available in “Smartshop Ai” System.</td></tr>
<tr><td>Precondition</td><td>Customers and employees must use the system.</td></tr>
<tr><td>Success End Condition</td><td>Change in interface to dashboard</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Product showing failed”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Employee, Customer</td></tr>
<tr><td>Trigger</td><td>Employees and customers will search in the search bar.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Request to Sign in.</td></tr>
<tr><td>2.</td><td>Go to the Dashboard.</td></tr>
<tr><td>3.</td><td>Type search keyword in the search bar.</td></tr>
<tr><td>4.</td><td>Press “Search Button’</td></tr>
<tr><td>5.</td><td>Check product</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Sign in failed</td></tr>
<tr><td></td><td>1.1.a Try Again.</td></tr>
<tr><td>3.1</td><td>Product not found.</td></tr>
<tr><td></td><td>6.1.a Try different keyword.</td></tr>
<tr><td>4.1</td><td>Empty product list.</td></tr>
<tr><td></td><td>4.1.a Try again.</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Users must get the search result within few seconds.</td></tr>
</table>

Case Description-12: Stock Control

<table>
<tr><td>Use Case</td><td>Stock Control</td></tr>
<tr><td>Goal</td><td>Allows manager to check the availability of the products.</td></tr>
<tr><td>Precondition</td><td>Manager must be signed in to the system.</td></tr>
<tr><td>Success End Condition</td><td>Notification “Products in the list showed”</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “No available products”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Manager</td></tr>
<tr><td>Trigger</td><td>Manager checks the products availabity.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Request to sign in.</td></tr>
<tr><td>2.</td><td>Go to the Dashboard.</td></tr>
<tr><td>3.</td><td>Press “Stock Control”.</td></tr>
<tr><td>4.</td><td>View Products</td></tr>
<tr><td>5.</td><td>Check specific products with keyboard search.</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Sign in failed</td></tr>
<tr><td></td><td>1.1.a. Try Again.</td></tr>
<tr><td>2.1</td><td>Server error.</td></tr>
<tr><td></td><td>2.1.a Try again.</td></tr>
<tr><td>4.1</td><td>Empty product list</td></tr>
<tr><td></td><td>4.1.a Try again.</td></tr>
<tr><td>5.1</td><td>Product not found.</td></tr>
<tr><td></td><td>5.1.a Try a different keyboard</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Manager must control stocks within 30 minutes.</td></tr>
</table>

Case Description-13: Report Generate

<table>
<tr><td>Use Case</td><td>Report Generate</td></tr>
<tr><td>Goal</td><td>Allows authority to generate and view reports.</td></tr>
<tr><td>Precondition</td><td>Authority must be signed in to the system .</td></tr>
<tr><td>Success End Condition</td><td>Notification “Report generated successfully”</td></tr>
<tr><td>Failed End Condition</td><td>Notification “Report generation failed”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Authority</td></tr>
<tr><td>Trigger</td><td>Authority generates reports of products sales information or of employees salary history.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Request to Sign in.</td></tr>
<tr><td>2.</td><td>Go to the Dashboard.</td></tr>
<tr><td>3.</td><td>Press “Generate Report”.</td></tr>
<tr><td>4.</td><td>View statistics.</td></tr>
<tr><td>5.</td><td>Search product report.</td></tr>
<tr><td>6.</td><td>Press “Get report”.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Sign in failed</td></tr>
<tr><td></td><td>1.1.a Try again.</td></tr>
<tr><td>2.1</td><td>Server error.</td></tr>
<tr><td></td><td>2.1.a Try again.</td></tr>
<tr><td>3.1</td><td>Report generation failed</td></tr>
<tr><td></td><td>3.1.a Try again</td></tr>
<tr><td>5.1</td><td>Report not found.</td></tr>
<tr><td></td><td>5.1.a Try again.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Authority must get reports within 5 minutes.</td></tr>
</table>

Case Description-14: Sign out

<table>
<tr><td>Use Case</td><td>Sign out</td></tr>
<tr><td>Goal</td><td>Users can Sign out from the system.</td></tr>
<tr><td>Precondition</td><td>The user must sign in to the system.</td></tr>
<tr><td>Success End Condition</td><td>Notification: “Sign out successful”.</td></tr>
<tr><td>Failed End Condition</td><td>Notification “System Error”.</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer, Employee, Manager
Authority</td></tr>
<tr><td>Trigger</td><td>The user request a Sign out</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Sign in to the system.</td></tr>
<tr><td>2.</td><td>Select the “Sign out” Option.</td></tr>
<tr><td>3.</td><td>Press the “Sign out” Button.</td></tr>
<tr><td>4.</td><td>Sign out successful.</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Server down</td></tr>
<tr><td></td><td>1.1.a Try again</td></tr>
<tr><td>2.1</td><td>Not Responding</td></tr>
<tr><td></td><td>2.1.a try again</td></tr>
<tr><td>3.1</td><td>Not Responding</td></tr>
<tr><td></td><td>3.1.a try again</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will immediately Sign out After hitting the Sign out Button.</td></tr>
</table>

Case Description-15: Get Invoice

<table>
<tr><td>Use Case</td><td>Get Invoice</td></tr>
<tr><td>Goal</td><td>Allows user to get invoiced of their purchase.</td></tr>
<tr><td>Precondition</td><td>Users have to order.</td></tr>
<tr><td>Success End Condition</td><td>Invoice generation successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Invoice generation failed.</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customers
Authority</td></tr>
<tr><td>Trigger</td><td>User will request to get invoice of their purchase.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User Visit Dashboard</td></tr>
<tr><td>2.</td><td>Selects products they want to purchase.</td></tr>
<tr><td>3.</td><td>Confirms purchase and proceeds to pay.</td></tr>
<tr><td>4.</td><td>Selects payment method and purchase product.</td></tr>
<tr><td>5.</td><td>Requests for invoice.</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Server Error.</td></tr>
<tr><td></td><td>1.1.a Try again.</td></tr>
<tr><td>2.1</td><td>Product selection failed</td></tr>
<tr><td></td><td>2.1.a bad internet connection</td></tr>
<tr><td>4.1</td><td>Wrong credentials.</td></tr>
<tr><td></td><td>4.1.a Give the correct information</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The system should generate invoice within 3 seconds to ensure a smooth user experience.</td></tr>
</table>

Case Description-16: Confirm Order

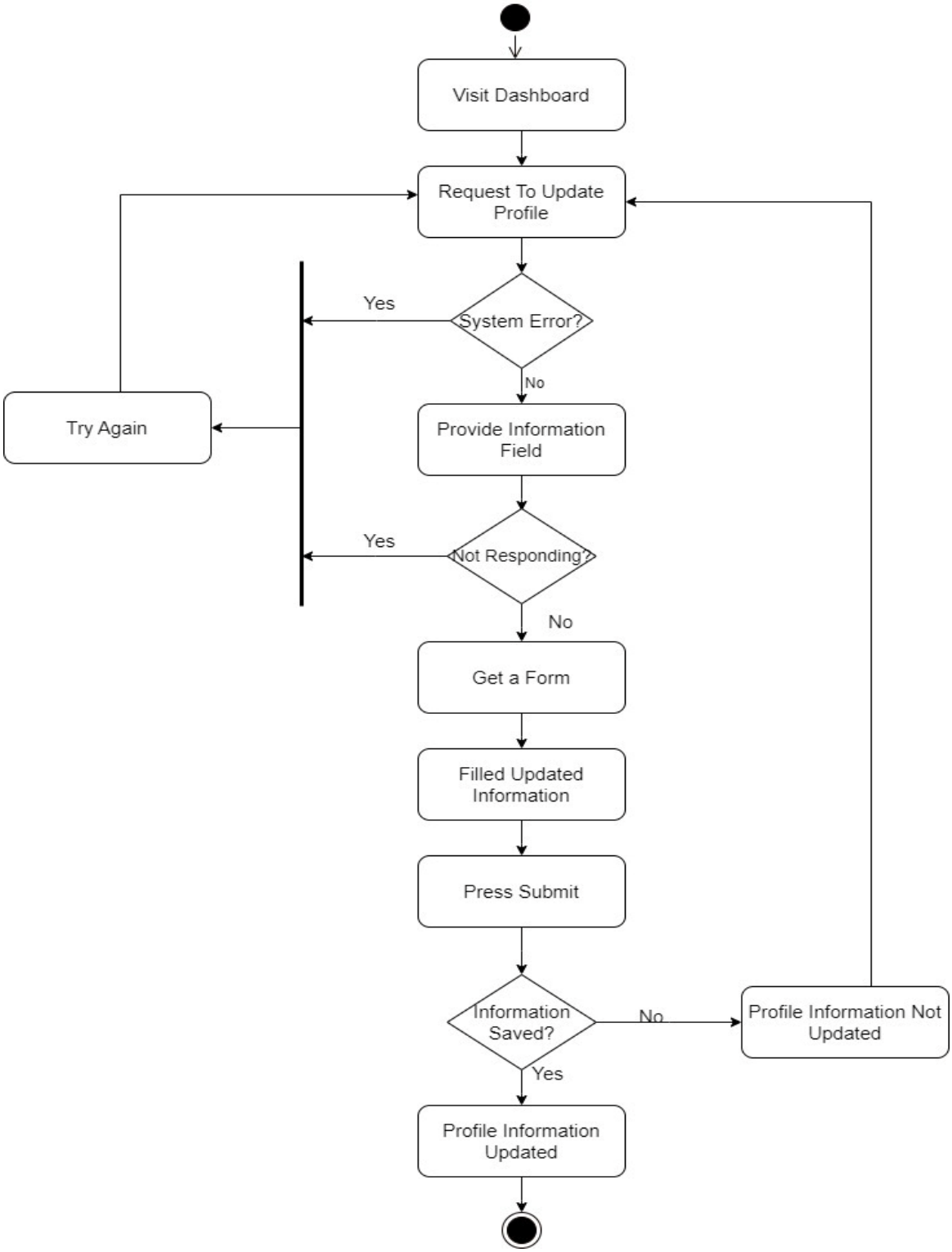
<table>
<tr><td>Use Case</td><td>Confirm Order</td></tr>
<tr><td>Goal</td><td>Allow users to confirm order.</td></tr>
<tr><td>Precondition</td><td>Users must order.</td></tr>
<tr><td>Success End Condition</td><td>Product ordered successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Order failed.</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer
Authority</td></tr>
<tr><td>Trigger</td><td>Users will request to order their desired product.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User Visit Dashboard</td></tr>
<tr><td>2.</td><td>User views the product.</td></tr>
<tr><td>3.</td><td>Select the product for detail information.</td></tr>
<tr><td>4.</td><td>Adds to cart for order.</td></tr>
<tr><td>5.</td><td>Confirms to order to purchase.</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Server Error.</td></tr>
<tr><td></td><td>1.1.a Try again.</td></tr>
<tr><td>3.1</td><td>Product selection failed</td></tr>
<tr><td></td><td>3.1.a bad internet connection</td></tr>
<tr><td>5.1</td><td>Confirmation failed.</td></tr>
<tr><td></td><td>5.1.a Connection pulling issues.</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The system should launch camera within 5 seconds.</td></tr>
</table>

Case Description-17: Try-On

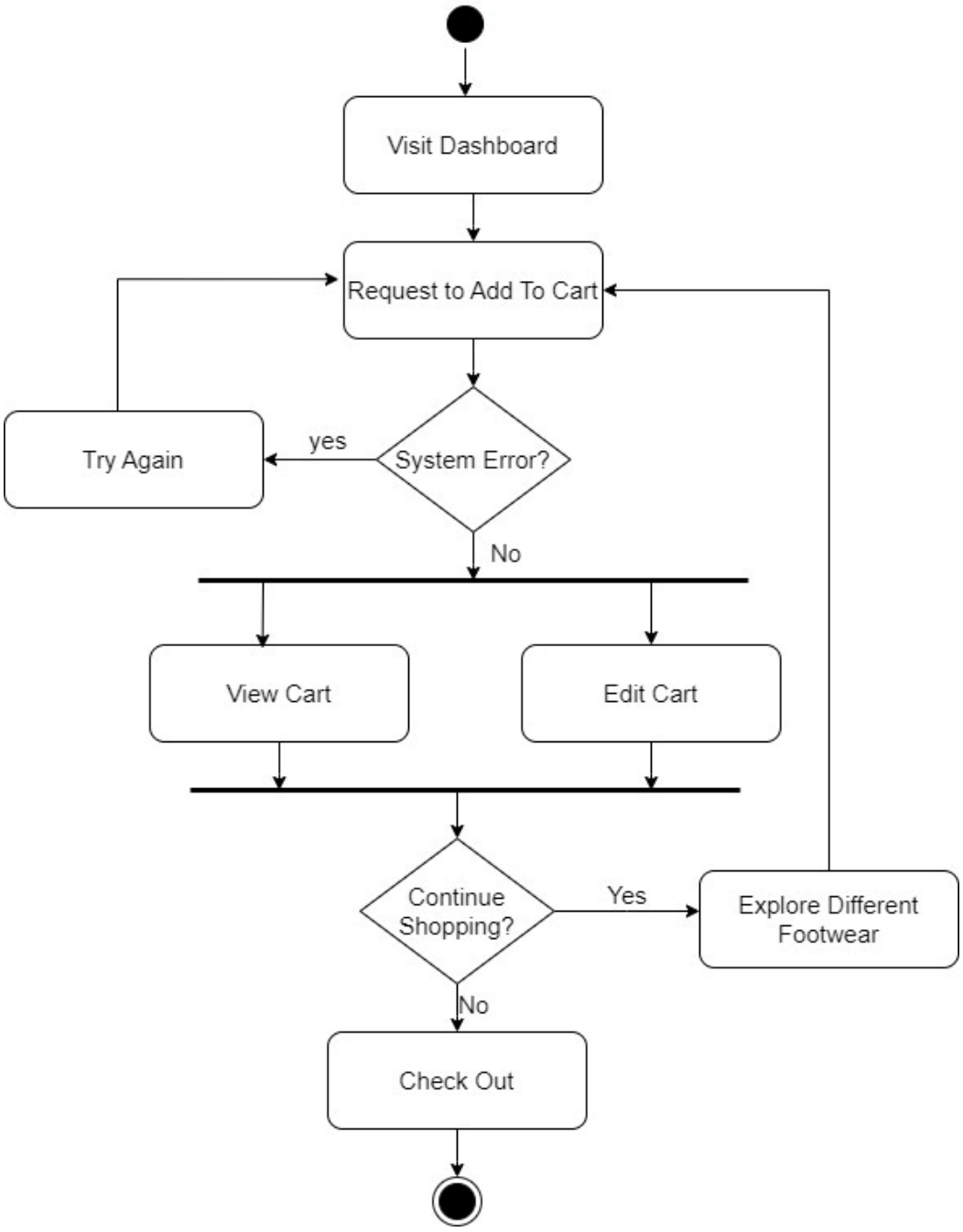
<table>
<tr><td>Use Case</td><td>Try-On</td></tr>
<tr><td>Goal</td><td>Allows users to virtually try-on products by uploading their image.</td></tr>
<tr><td>Precondition</td><td>Users should select a product.</td></tr>
<tr><td>Success End Condition</td><td>Try-on successful.</td></tr>
<tr><td>Failed End Condition</td><td>Image upload failed.</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customers
Authority</td></tr>
<tr><td>Trigger</td><td>Users will request to try on their selected products by uploading their image.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User Visit Dashboard</td></tr>
<tr><td>2.</td><td>Selects products they want to try-on.</td></tr>
<tr><td>3.</td><td>Camera launches for try-on.</td></tr>
<tr><td>4.</td><td>User check how they would look in the product.</td></tr>
<tr><td>5.</td><td>User proceeds to the next step</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Server Error.</td></tr>
<tr><td></td><td>1.1.a Try again.</td></tr>
<tr><td>2.1</td><td>Product selection failed</td></tr>
<tr><td></td><td>2.1.a bad internet connection</td></tr>
<tr><td>3.1</td><td>Camera launch failed.</td></tr>
<tr><td></td><td>3.1.a Plugin malfunctions.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The system should launch camera within 5 seconds.</td></tr>
</table>

1.1.7 2.4.3 Activity Diagram

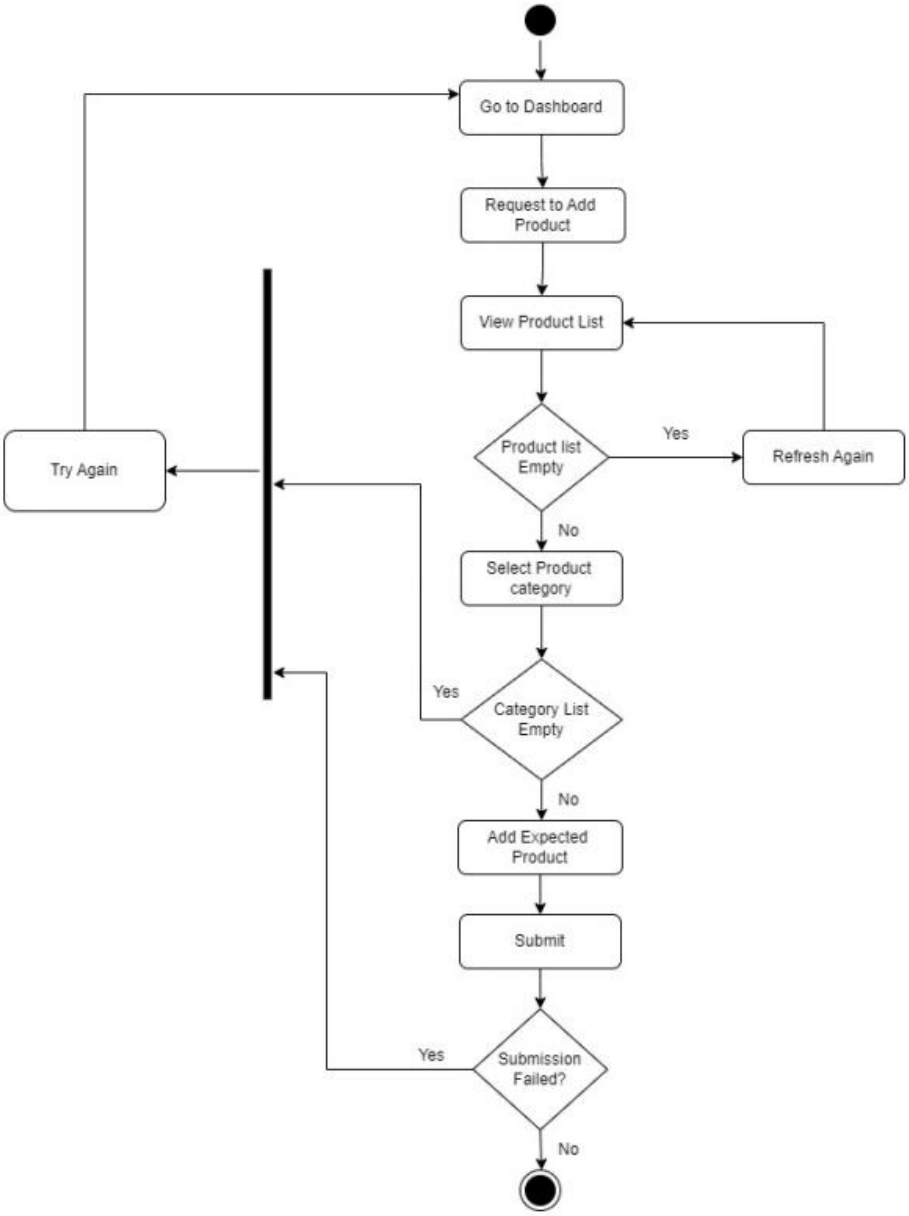
Activity Diagram-1: Update Profile



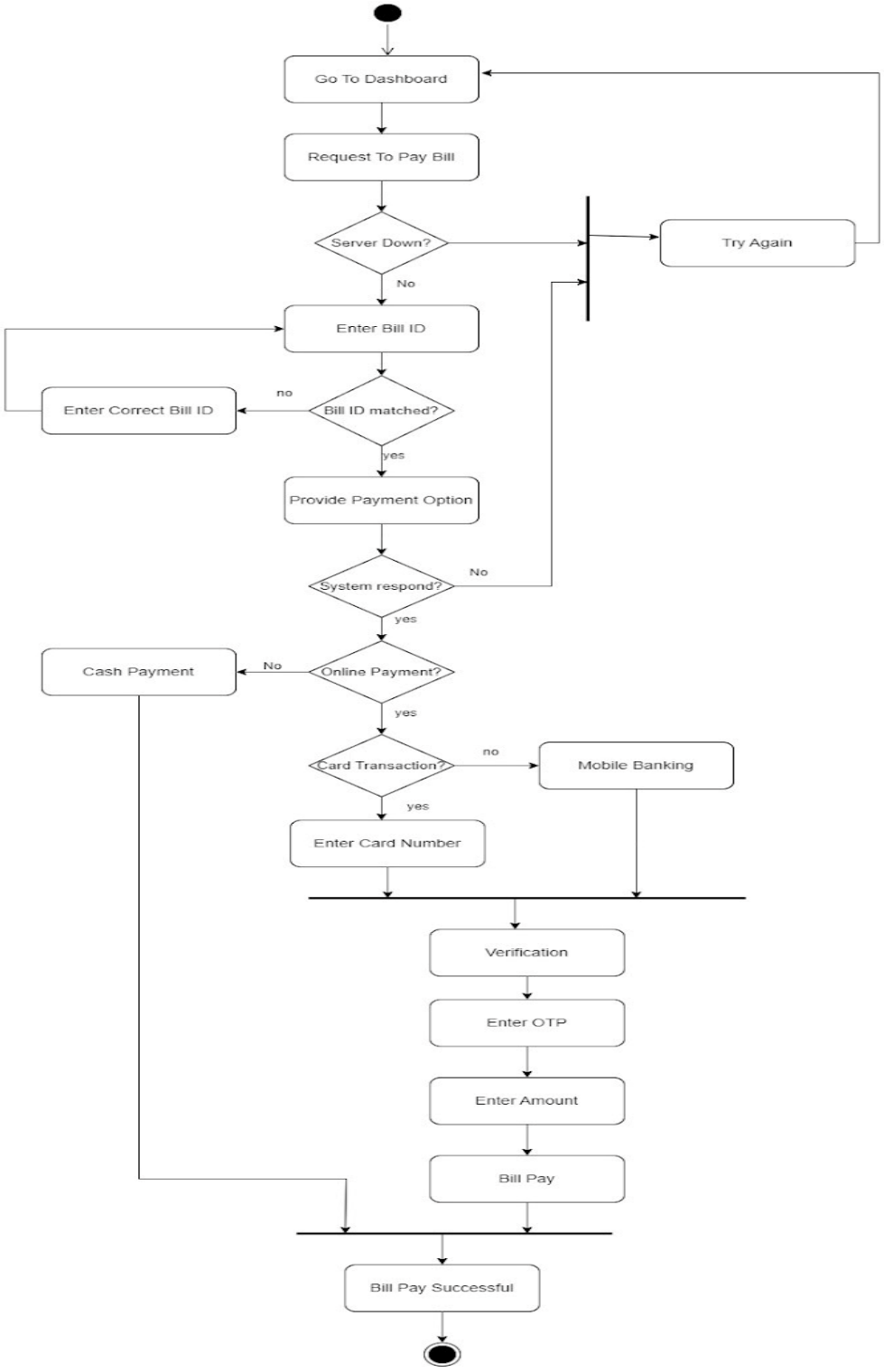
Activity Diagram-2: Add To Cart



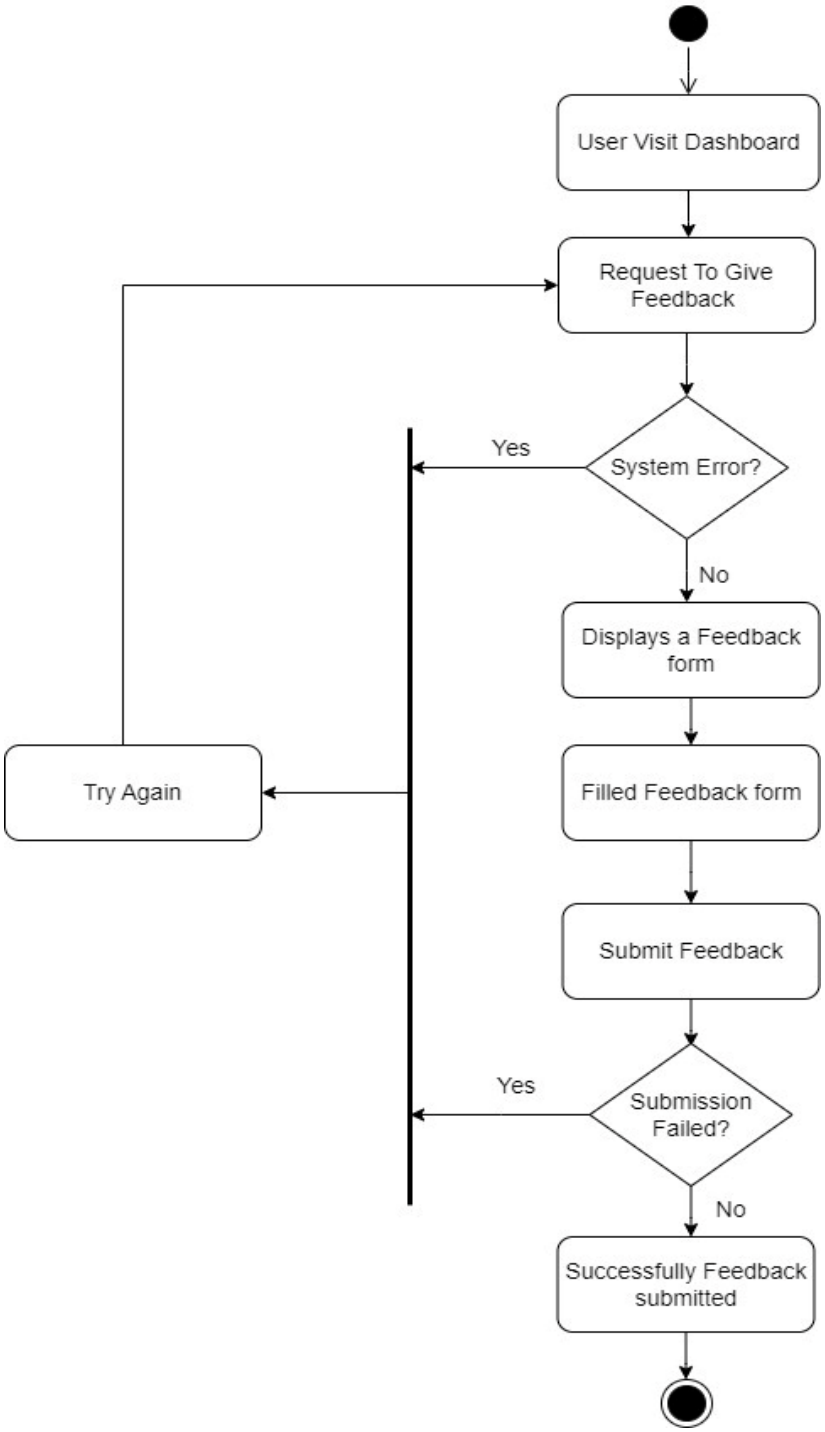
Activity Diagram-3: Add Product



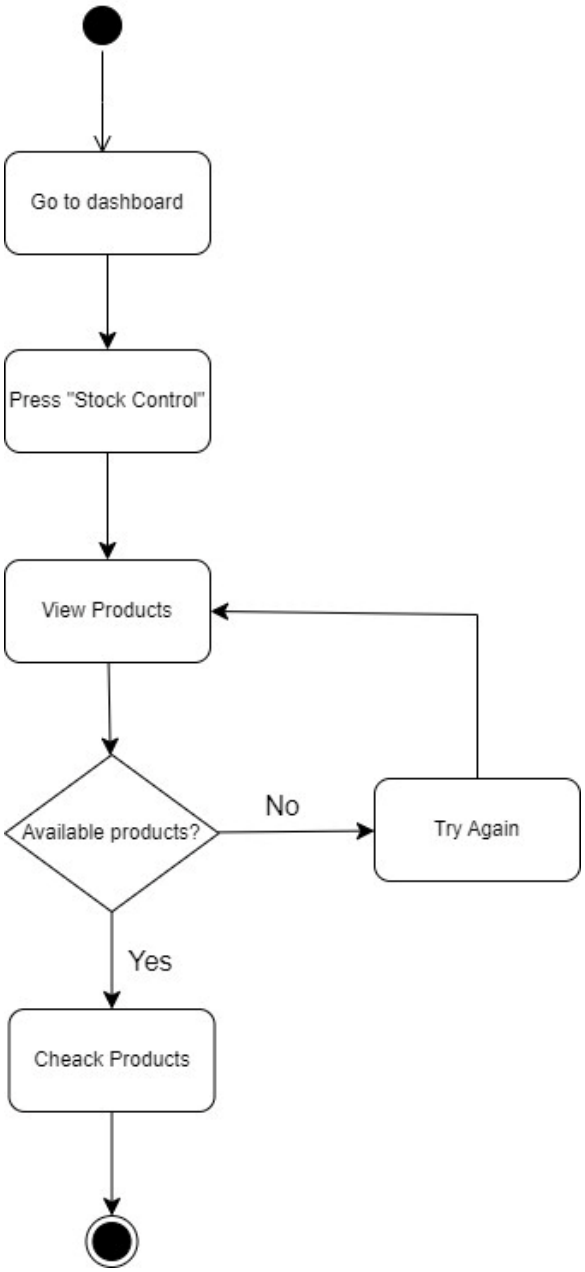
Activity Diagram-4: Make Payment



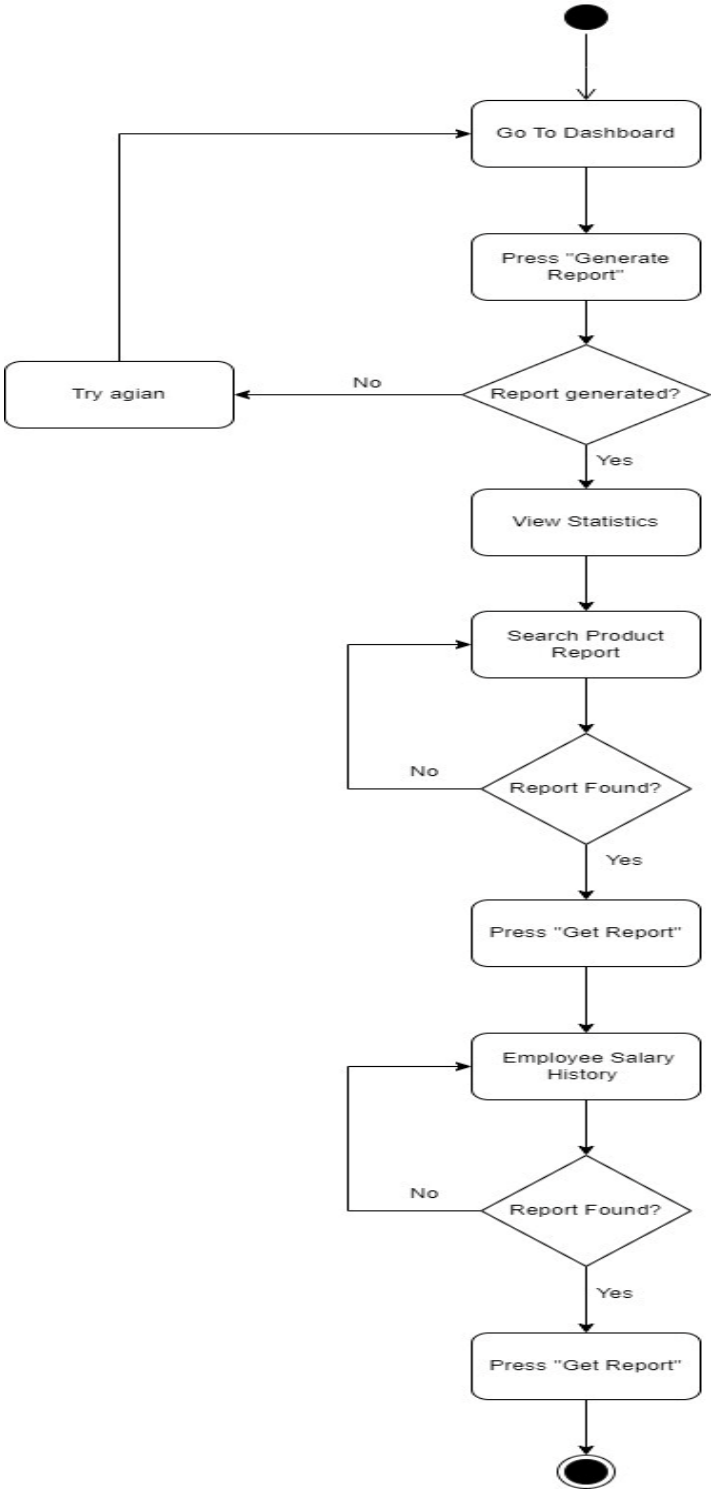
Activity Diagram-5: Give feedback



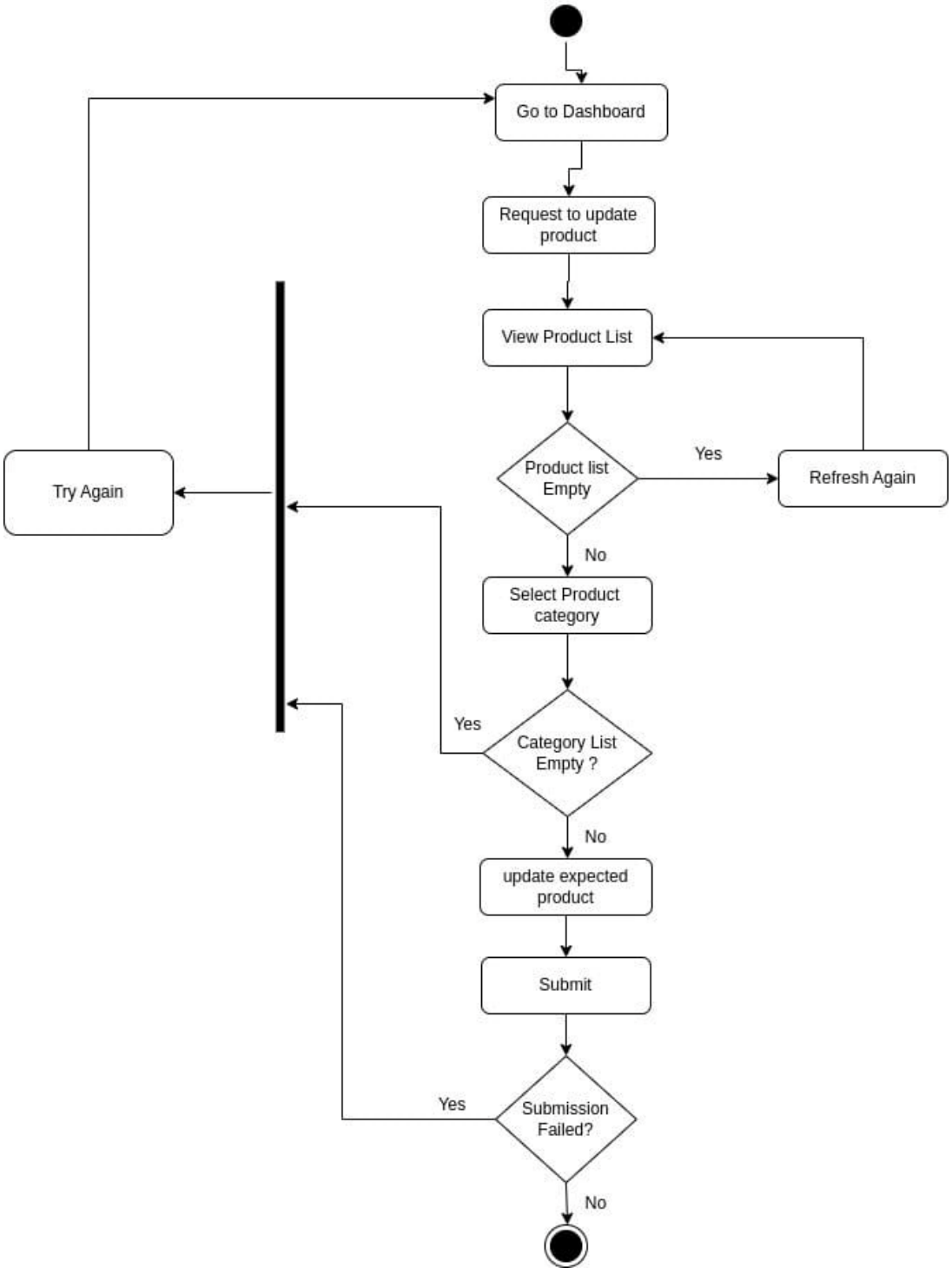
Activity Diagram-6: Stock Control



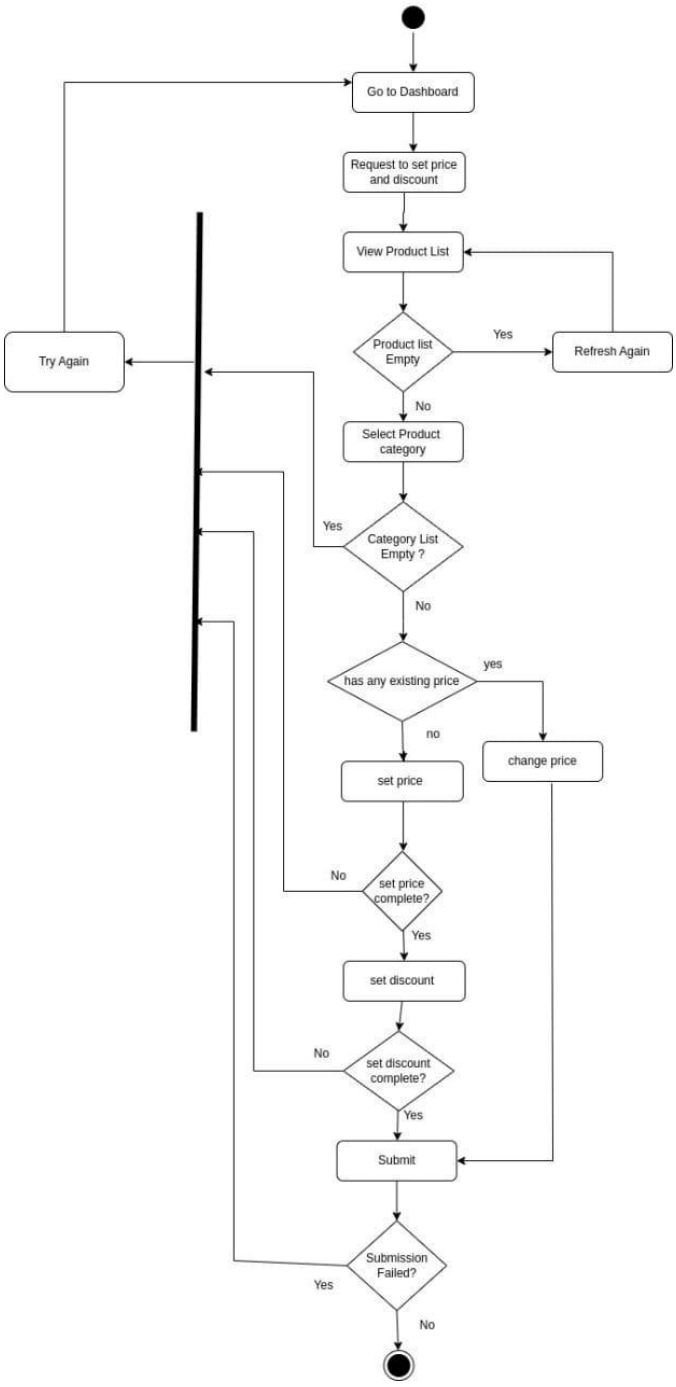
Activity Diagram-7: Report Generate



Activity Diagram-8: Update Product



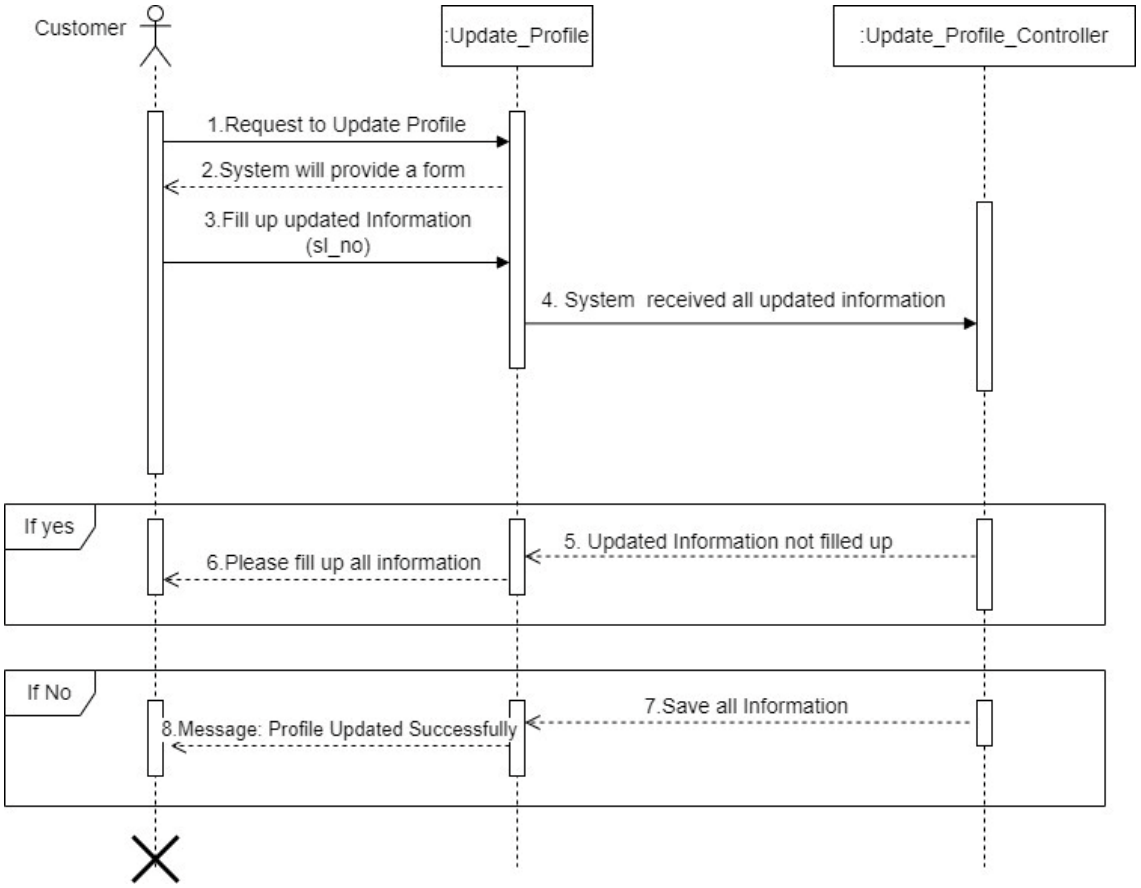
Activity Diagram-9: Set Price and Discount



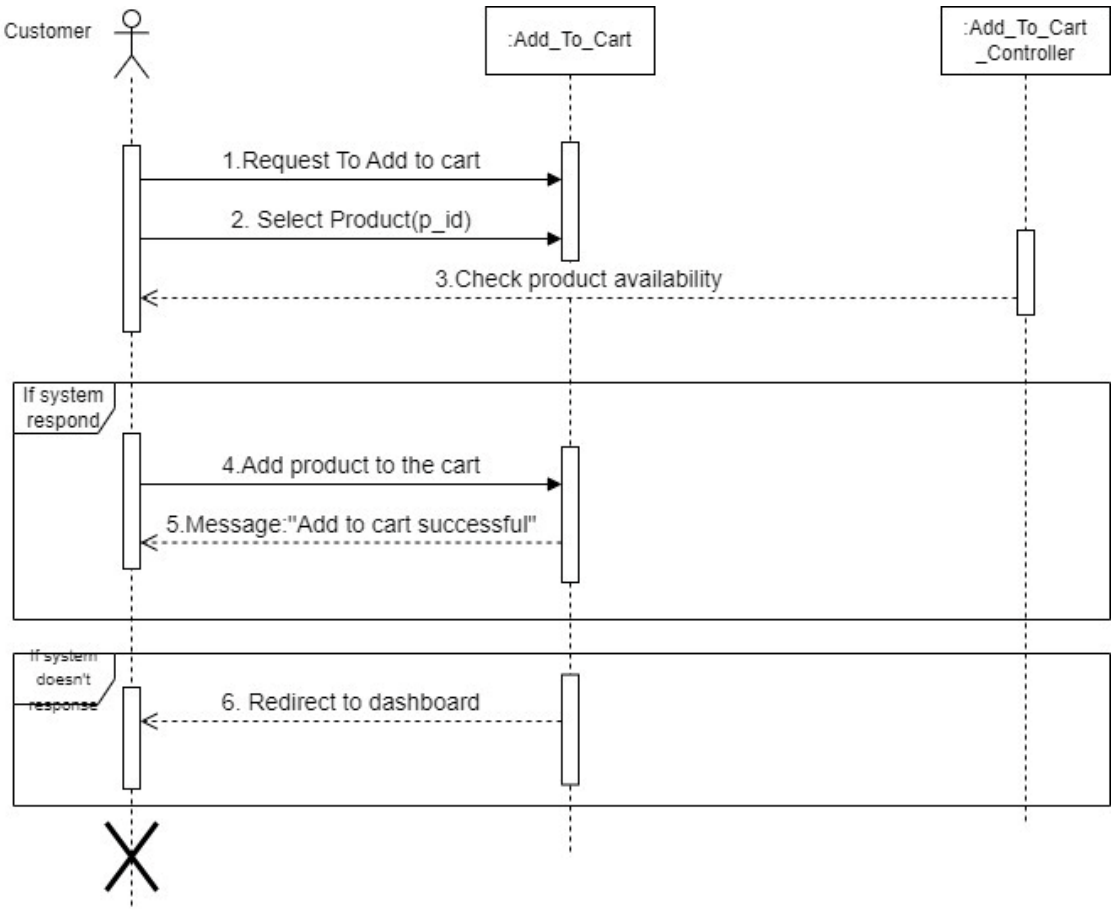
1.1.8

1.1.9 2.4.4 Sequence Diagram

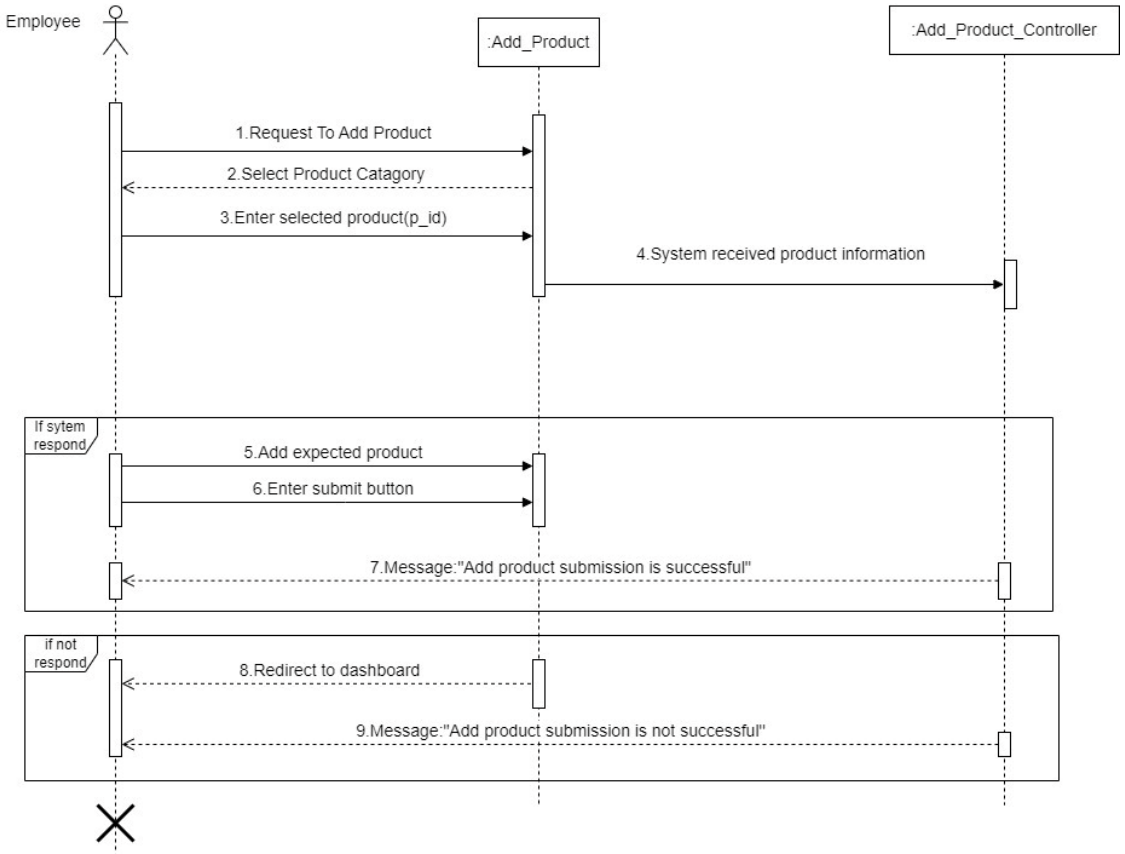
Sequence Diagram-1: Update Profile



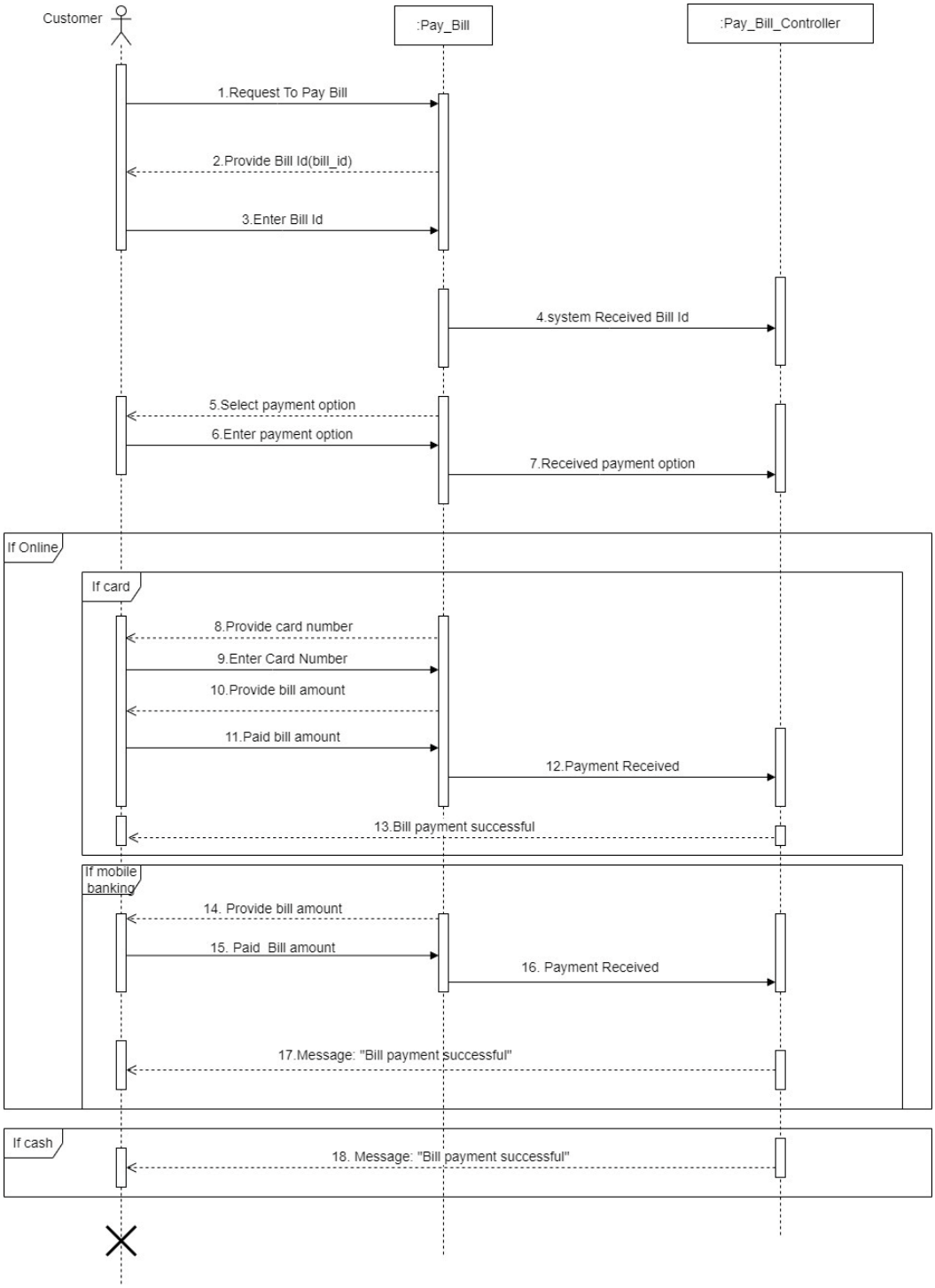
Sequence Diagram-2: Add to cart



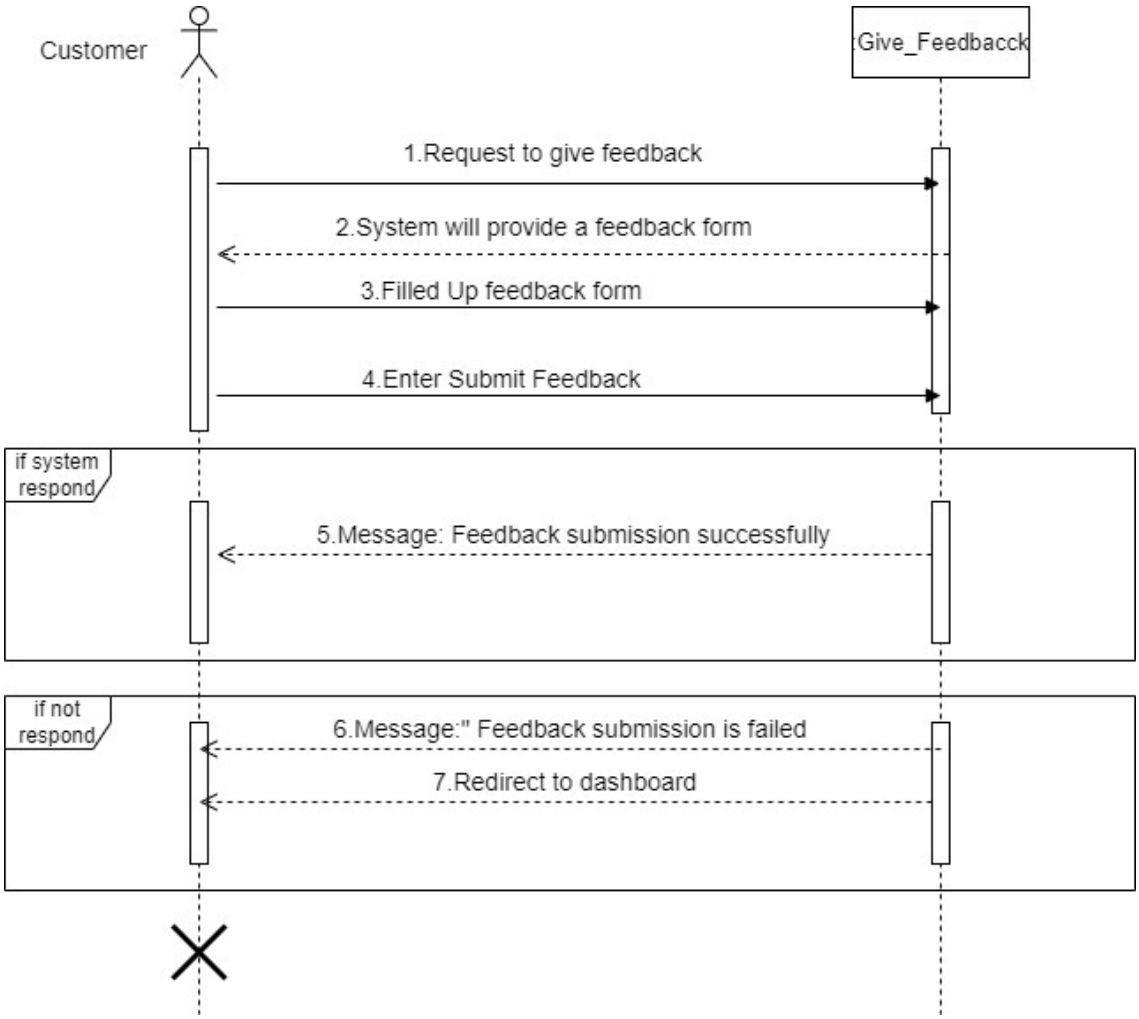
Sequence Diagram-3: Add Product



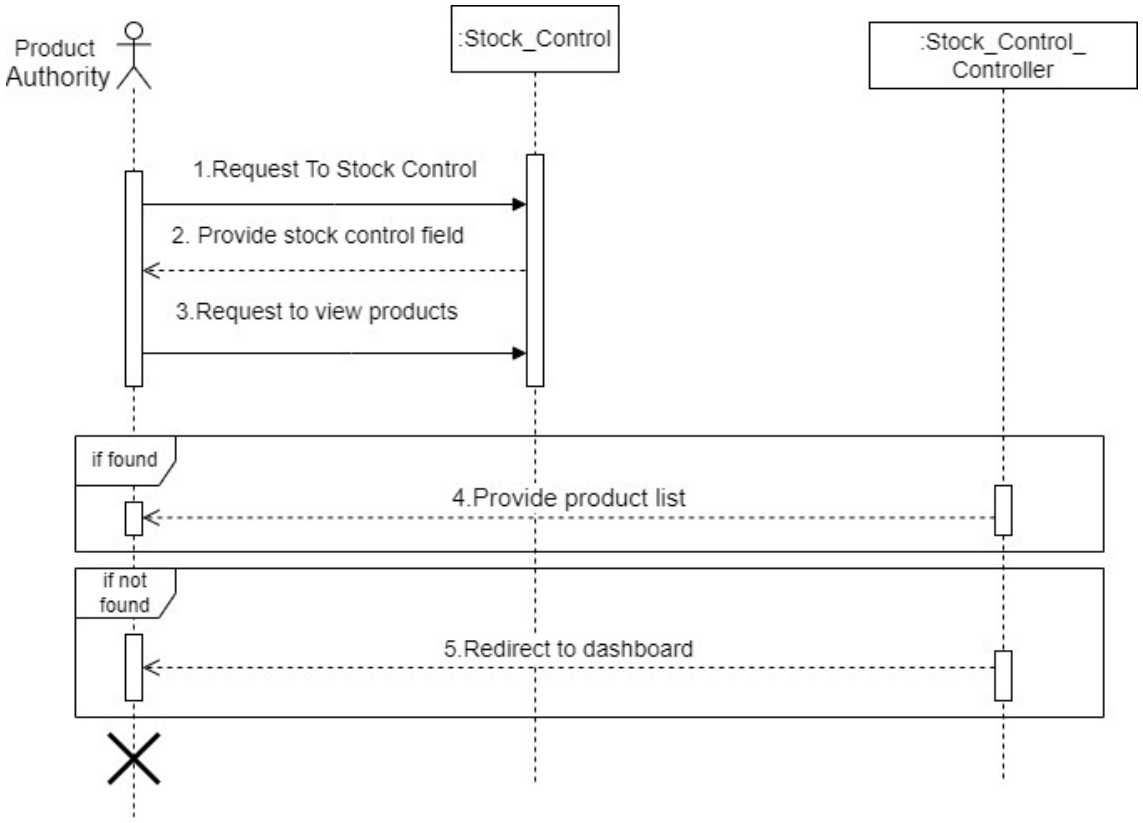
Sequence Diagram-4: Pay Bill



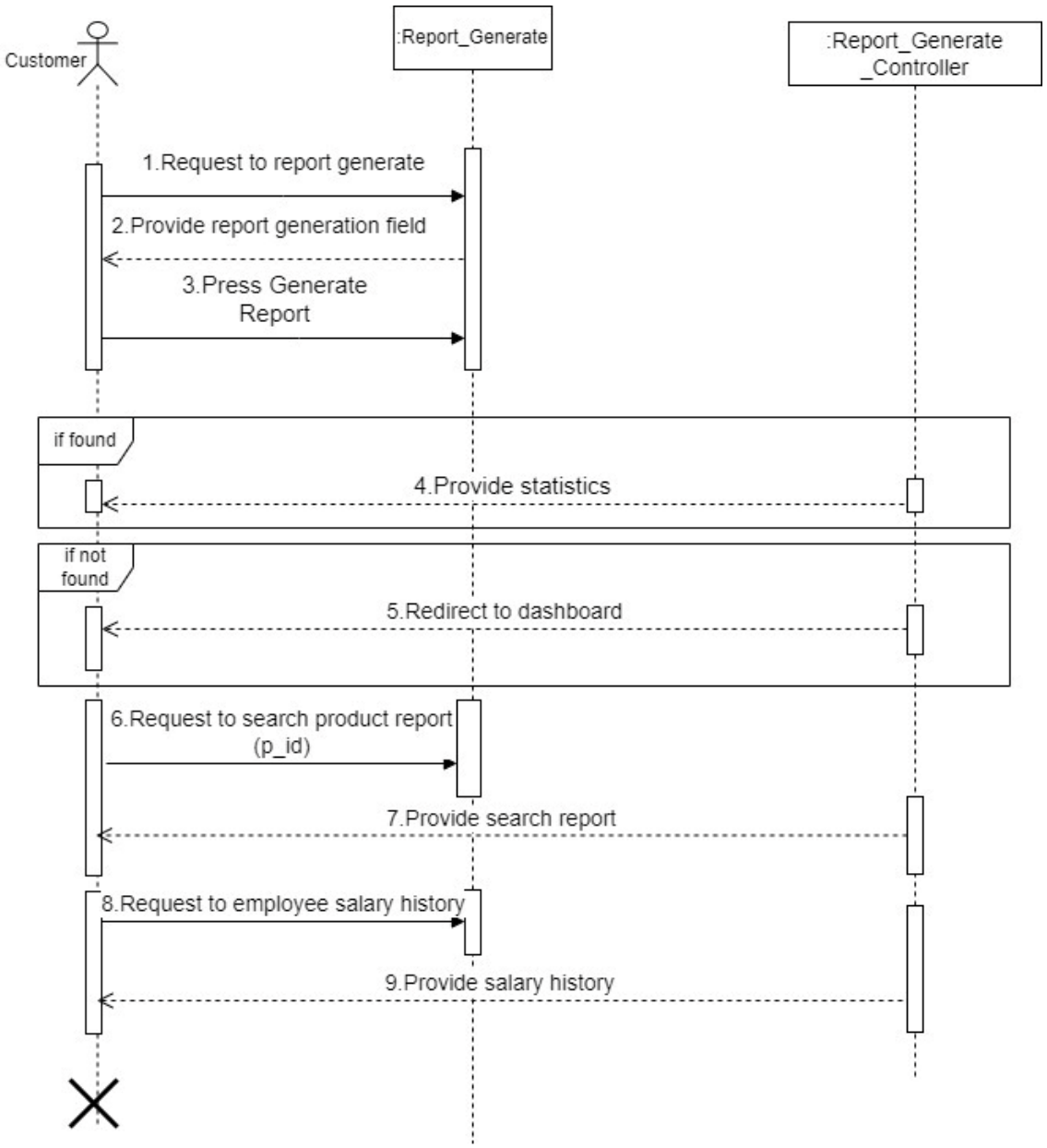
Sequence Diagram-5 : Give Feedback



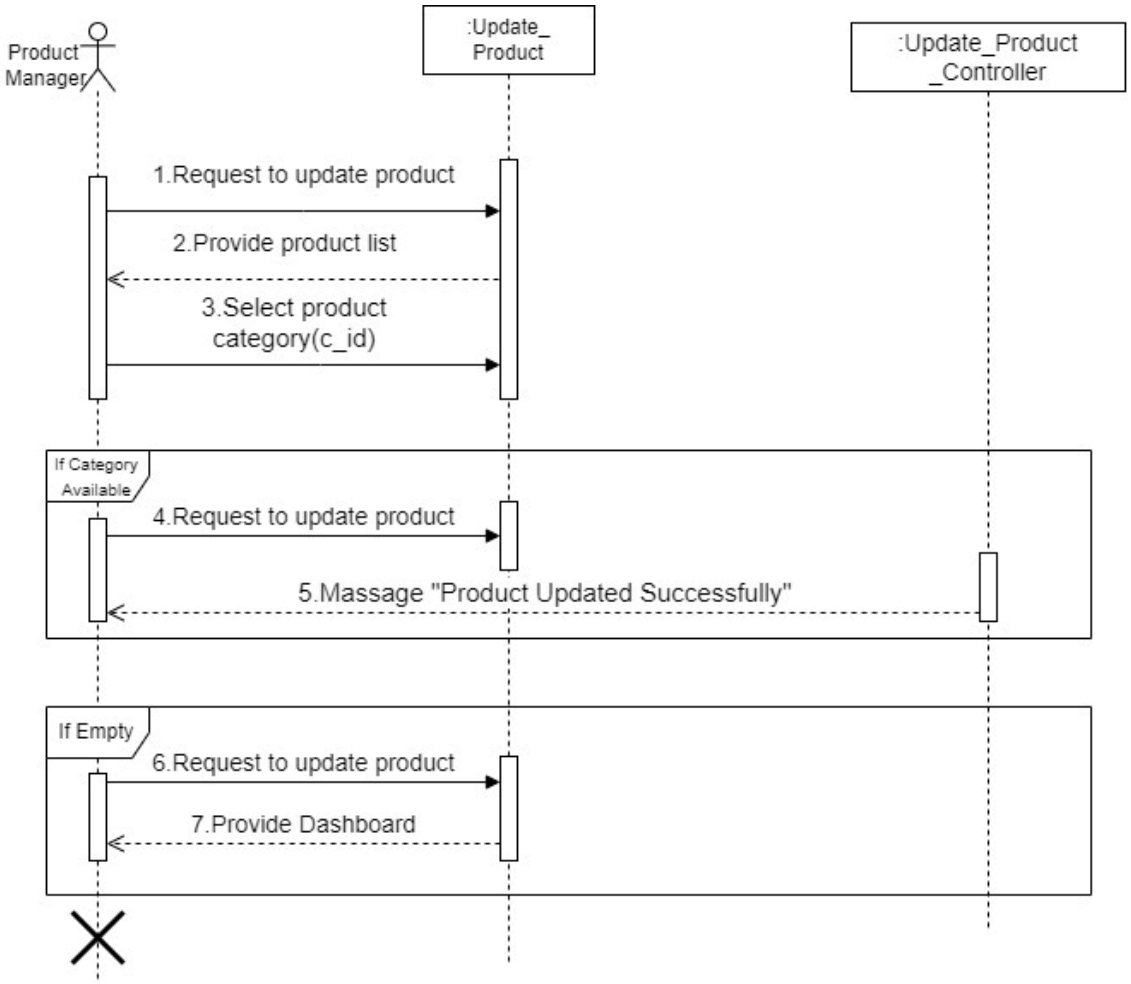
Sequence Diagram-6: Stock Control



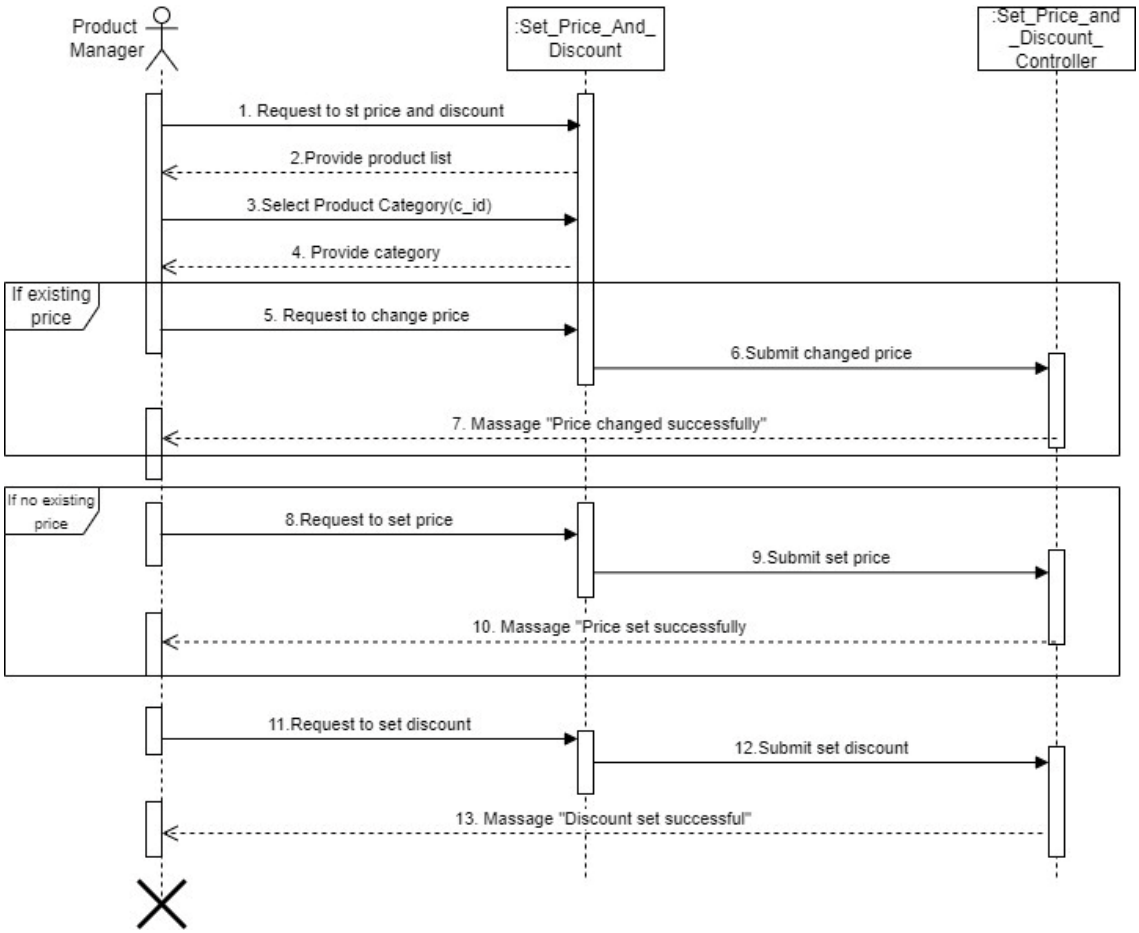
Sequence Diagram-7 : Report generate



Sequence Diagram-8: Update Product



Sequence Diagram-9: Set price and discount



2.4.5 Class Diagram

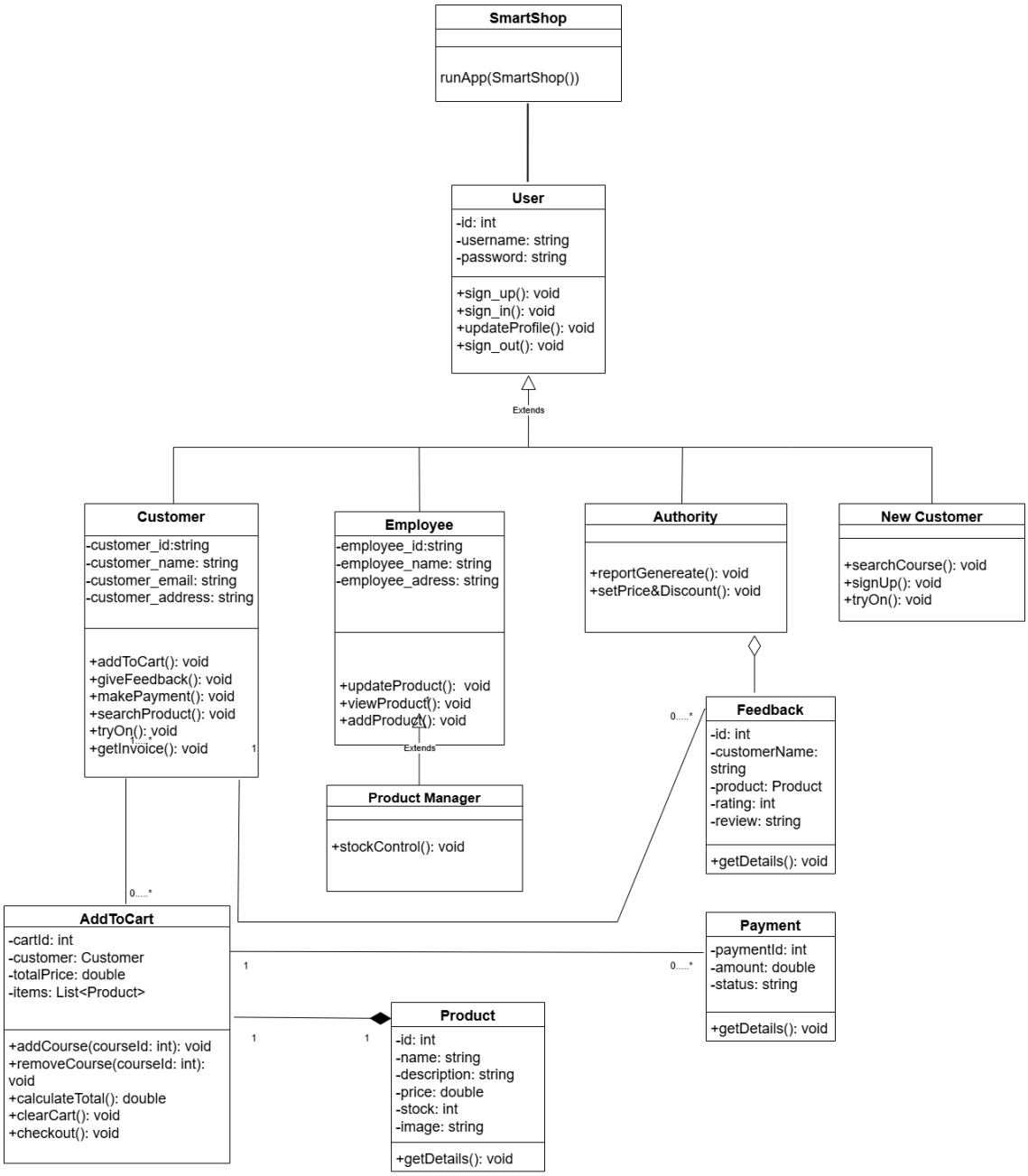


Figure 4: Class Diagram

2.4.6 ER Diagram

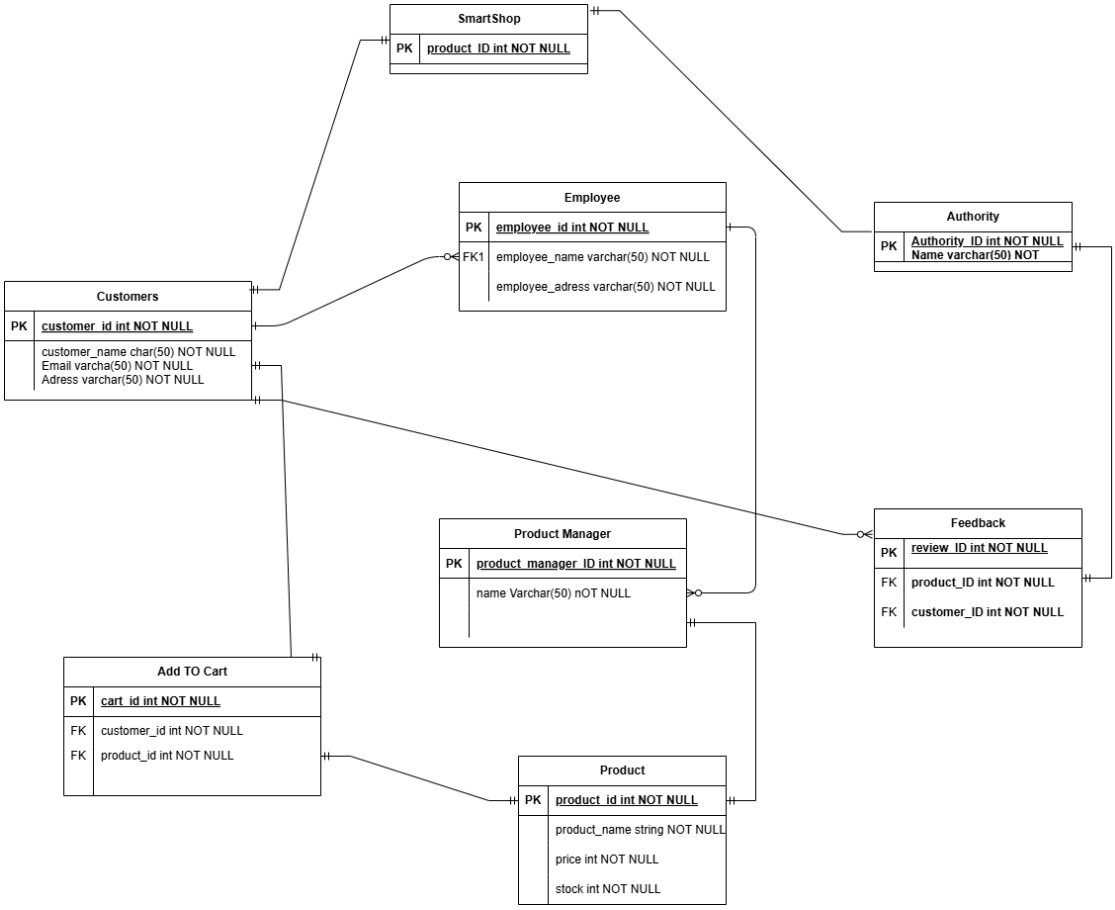


Figure 5: ER Diagram

2.5 Coding: Appendix A

The Coding part is shown at the end.

2.6 Summary

This chapter defines the structure of the whole project.

Functional requirements, non-functional requirements, Object oriented system design using UML diagrams which includes:

1. Use case diagram
2. Case description
3. Activity diagram

4. Sequence diagram

5. Class diagram

6. ER diagram

These diagrams give us a very clear picture of the project functionalities. Use case diagram lets us know the project functionalities, case descriptions lets us know the descriptions of all the functionalities, Activity diagrams defines the activity of each functions, the sequence diagrams provide the sequence of how the functions are going to work, the class diagram pictures out the whole apps with stakeholders functionalities and relativity , and the er diagram gives us the idea of how the database would look like of the project.

And at the end of the chapter some coding examples are given which shows a clear picture of how the project is being developed.

Chapter 3

Software Testing

3.1 Introduction

The Smartshop system serves as an online shopping platform aimed at streamlining the purchasing and product management processes for both users and employees. It features an intuitive interface that allows customers to register, log in, explore products, add items to their cart, and complete their purchases. Besides the user features, the system provides access to the employees, which allows them to perform product management, dashboard awareness, and account-related operations. The main idea of the system is to provide the users with easy communication and interaction with the store, secure authentication, efficient data management, and without any difficulties, navigation of the dashboards. The service values reliability, security, usability, and efficiency, which cause the users and employees not to have problems with the complexities of using the system.

3.2 Testing Features

1.1.10 3.2.1 Feature to Be Tested

- **a.** User Registration
- **b.** User Login

c. Employee Registration
d. Employee Login
e. User Dashboard
f. Employee Dashboard
g. Add to cart
h. User Logout
i. Employee Logout
j. Update Profile

3.3 Testing Strategies

3.3.1 Test Approach (Enhanced Version)

Tests should involve an approach that is more effective and efficient (enhanced version). The Smartshop system testing methodology is based on the systematic and structured system testing practices. All features of the application are tested to establish that all the functional components of the application work properly in different input conditions. Test cases are designed on the positive and negative cases to test the behavior of a system where the system is provided with valid, invalid, and boundary values. Manual testing is done to simulate actual user experiences whereby inputs, outputs and system reaction are recorded and evaluated. The primary objective of this practice is to identify the functional failures, approve the meeting of the requirements, and ensure the system stability, security and reliability before the final deployment. This is because problems are resolved at a tender age and this improves the overall performance, usability and quality of the software.

3.3.2 Pass/Fail Criteria

A test case is considered a Pass when the real result of the system is the expected one without any errors or irregularities. Conversely, a test case is considered as a Fail when system response is other than what is expected, accepts an invalid data, functions improperly, and leads to the occurrence of unforeseen event like crashes or faulty navigation. Only on passing all the tests associated with a feature, a feature is approved and accepted. When all the key modules are passed, the entire system will be deemed fit to be deployed and this will ensure that the software satisfies its functional, operational, and quality requirements.

Test Case 1: User Registration

Test Case: 3.4.1	**Test Case Name:** **User Registration**
System: SmartShop	**Subsystem:**

Test Case: 3.4.1

Test Case Name:

User Registration

User Authentication

Design

Designed by: Jubaer Islam Nahin

Date: Jubaer Islam Nahin

Execution

Executed by: Jubaer Islam Nahin

Date: _____

Description: Checking the ability of a user to register with the right information.

Pre-condition: The user must access the registration page.

Step	Name	Email	Password	Retype Password	Response	Pass/Fail	Comment
1	Valid Name	user@gmail.com	123456	123456	Registration Successful	Pass	All fields valid
2	*Blank*	user@gmail.com	123456	123456	Name Empty	Fail	Name required
3	User	*Blank*	123456	123456	Email Empty	Fail	Email required
4	User	user@gmail.com	*Blank*	123456	Password Field Empty	Fail	Password required

Test Case: 3.4.2 Test Case Name: User Sign in

System: Smartshop **Subsystem:** User Authentication

Step	Name	Email	Password	Retype Password	Response	Pass/Fail	Comment

1	Valid Name	user@gmail.com	123456	123456	Sign in Successful	Pass	Valid credentials
2	*Blank*	user@gmail.com	123456	123456	Name Empty	Fail	Name required
3	User	*Blank*	123456	123456	Email Empty	Fail	Email required
4	User	user@gmail.com	*Blank*	123456	Password Field Empty	Fail	Password required

Test Case: 3.4.3 **Test Case Name: Employee Sign Up**

System: Smartshop **Subsystem:** Employee Authentication

Step	**Name**	**Email**	**Password**	**Retype Password**	**Response**	**Pass/Fail**	**Comment**
1	Employee 1	employee@gmail.com	123456	123456	Sign Up Successful	Pass	Valid details
2	*Blank*	user@gmail.com	123456	123456	Name Empty	Fail	Name required
3	Employee	*Blank*	123456	123456	Email Required	Fail	Email required

Test Case: 3.4.4 Test Case Name: Employee Login

System: Smartshop **Subsystem:** Employee Authentication

Step	Name	Email	Password	Retype Password	Response	Pass/Fail	Comment
1	Valid Name	user@gmail.com	123456	123456	Sign Up Successful	Pass	All fields valid
2	*Blank*	user@gmail.com	123456	123456	Name Empty	Fail	Name required

Test Case: 3.4.5 Test Case Name: User Dashboard

System: Smartshop Subsystem: User Panel

Step	Action	Expected Result	Response	Pass/Fail	Comment
1	Sign in as user	Dashboard loads	Displayed	Pass	Dashboard works well
2	Access dashboard without sign in	Access denied	Redirect to sign in	Pass	Secured

Test Case: 3.4.6 Test Case Name: Employee Dashboard

System: Smartshop Subsystem: Employee Panel

Step	Action	Expected Result	Response	Pass/Fail	Comment

1	Sign in as employee	Dashboard loads	Displayed	Pass	Access granted
2	Access dashboard without sign in	Access denied	Redirect to sign in	Pass	Secured

Test Case: 3.4.7 **Test Case Name: Add to cart**

System: Smartshop **Subsystem: Cart Management**

Step	Item	Quantity	Response	Pass/Fail	Comment
1	Product selected	1	Added successfully	Pass	Item added
2	No item selected	-	Error displayed	Fail	Must choose product

Test Case: 3.4.8 **Test Case Name: User Sign out**

System: Smartshop **Subsystem: User Authentication**

Step	Action	Expected result	Response	Pass/Fail	Comment
1	Click Sign out	Session closed	Sign out successfully	Pass	Secure termination

2	Click Sign out	Session closed	Something went wrong	Fail	Server overload

Test Case: 3.4.9 **Test Case Name: Employee Sign out**

System: Smartshop **Subsystem: Employee Authentication**

Step	Action	Expected result	Response	Pass/Fail	Comment
1	Click Sign out	Session closed	Sign out successfully	Pass	Secure termination
2	Click Sign out	Session closed	Something went wrong	Fail	Server overload

Test Case: 3.4.10 **Test Case Name: Update Profile**

System: Smartshop **Subsystem: Account Management**

Step	Name	Email	Action	Response	Pass/Fail	Comment
1	Updated name	user@gmail.com	save	Profile Updated	Pass	Valid Update
2	Blank	user@gmail.com	save	Name required	Fail	Can not leave empty

3.5 Summary

This chapter provided a general review of the tests that were carried out on the Smartshop system. It described the important features of the system including user and employee functionality of the system and provided a methodology of testing that was systematic in a bid to ensure that performance, functionality, and reliability were satisfied. The test cases were used to test each feature using known test cases and expected outcomes to determine pass or fail criteria. The system test cases demonstrated the way the application would handle valid and invalid inputs in different modules. To conclude, the chapter reaffirms that the system meets its functional requirements and is stable, user friendly and ready to get deployed.

Chapter 4

Deployment and Maintenance

4.1 Introduction

The Deployment and Maintenance phase ensures successful delivery of Smartshop application to end users and its dependability in the long-term. The key goals of this stage are to prepare the system to be released into production, deploy the application to the selected platforms, monitor the performance of the application and make any necessary updates or improvements. It also involves the process of rectification, optimization of the system and ensuring the application is able to meet the evolving needs of the users and the technological changes.

4.2 Attempt to pursue SRLC (software release life cycle). Software Release Life Cycle (SRLC) describes the stages through which Smartshop goes through since its inception to a fully operational and developed product. Smartshop follows a systematic SRLC to ensure quality, stability and continued growth.

1. Pre-Alpha Stage

Designing the database structure, user interface wireframes and architecture. Setting up processes, systems, and development instruments. Developing such key elements as the back-end architecture, the approach to the integration of AI, product list, and authentication.

2. Alpha Stage

Primary preparation of ML model inclusion (visual search, recommendations). Simple user interface of the Flutter application has been applied. Testing done by developers to identify serious bugs. Attributes are not always stable.

3. Beta Stage

The application is fully functional but can use some improvement. It has added features of Artificial Intelligence such as size prediction, recommendation engine, and chatbot. Extensive testing has been done including functional, usability and performance test. It has released a beta version to a small number of people such as team members and testers. Feedback is also obtained to enable improvements, user interface optimization and correction of bugs.

4. Release Candidate (RC)

All the main features are applied and tried. Serious bugs are fixed, still, small problems might be present. Optimizations have been performed (loading speed, API response time and UI improvements). The system is now ready to be deployed on production servers and application stores. An ultimate documentation, configurations, and security verifications have been done.

5. Production Release

Smartshop is already available in the Google Play Store, App Store, and/or web based. The backend is deployed on cloud infrastructure like Firebase, AWS or Python backend server. Actual users begin to use the application. Monitoring tools are used to monitor user behavior, crashes and performance metrics.

6. Maintenance Stage

Regular updates on, Corrections to errors, Security improvements, User interface mobilizations Introduction of new features retraining or updating of AI models ensure that the application remains stable, secure, and relevant.

7. End-of-Life (Optional Future Stage)

In the case Smartshop reaches a mature stage and can be replaced by a newer technology or a newer version can be offered, an end-of-life strategy can be considered. This includes the termination of updates, providing migration support or initiating a major redesign.

Chapter 5

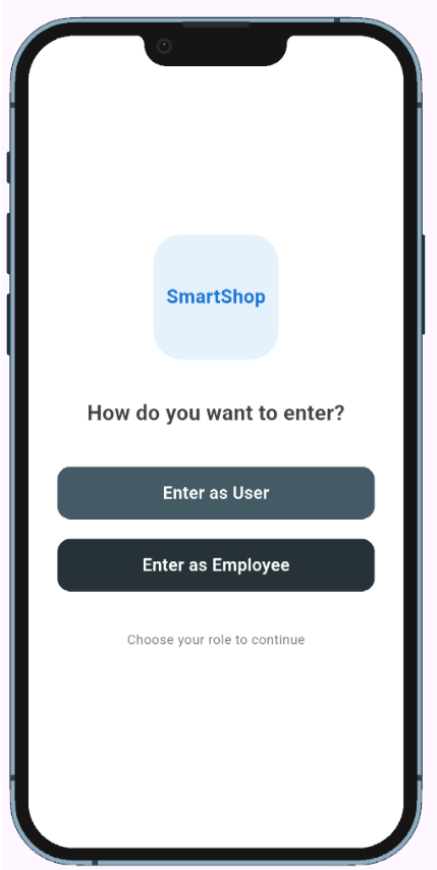
User Manual

5.1 Introduction

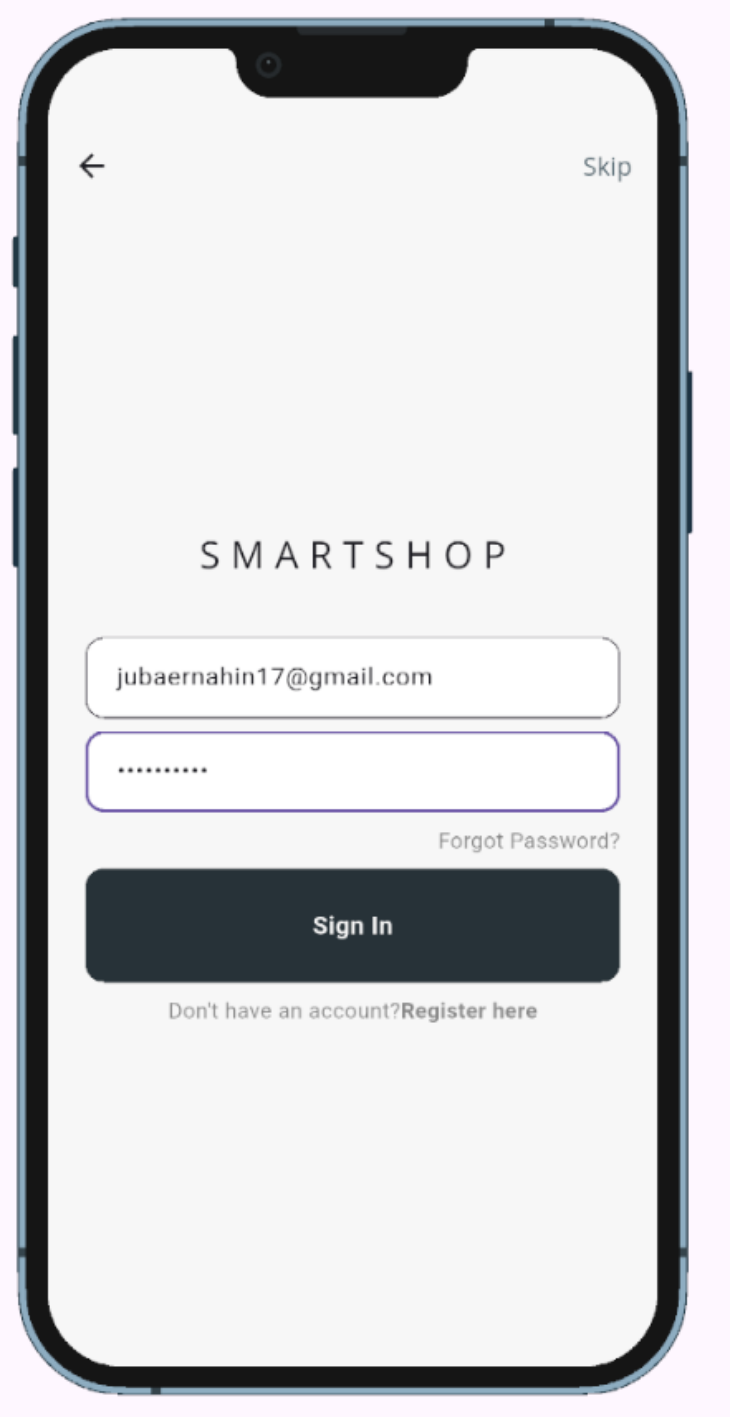
Smartshop App is a state-of-the-art, easy to use, shopping management application that will streamline the processes of product browsing, ordering, and buying to both the customers and the shop owners. This user manual is designed to be a step-by-step version to understand how to use all the features of the application so that clients could find it easy to navigate the interface, manage their accounts, explore products and finalize transactions without having to go through the manual too many times. This manual can guide both new and seasoned users on how to maximize the use of the easy-to-use tools and easy shopping experience of Smartshop.

5.2 Project Functionalities

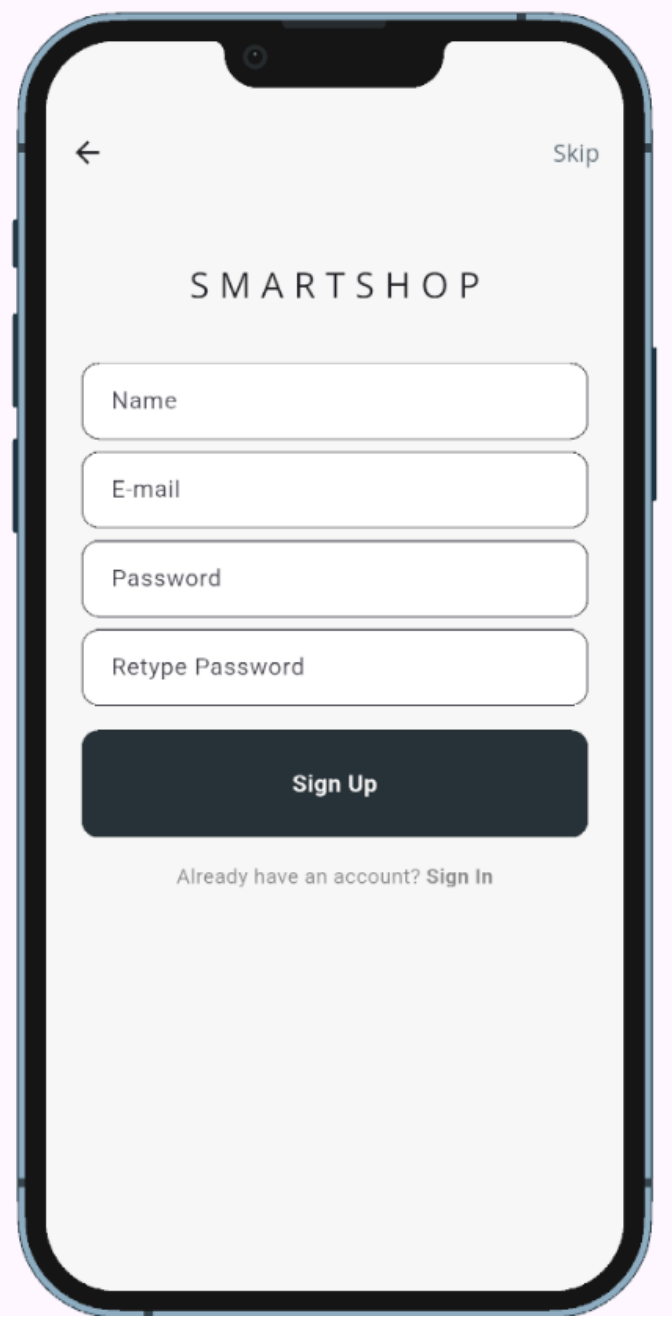
User Role Checker



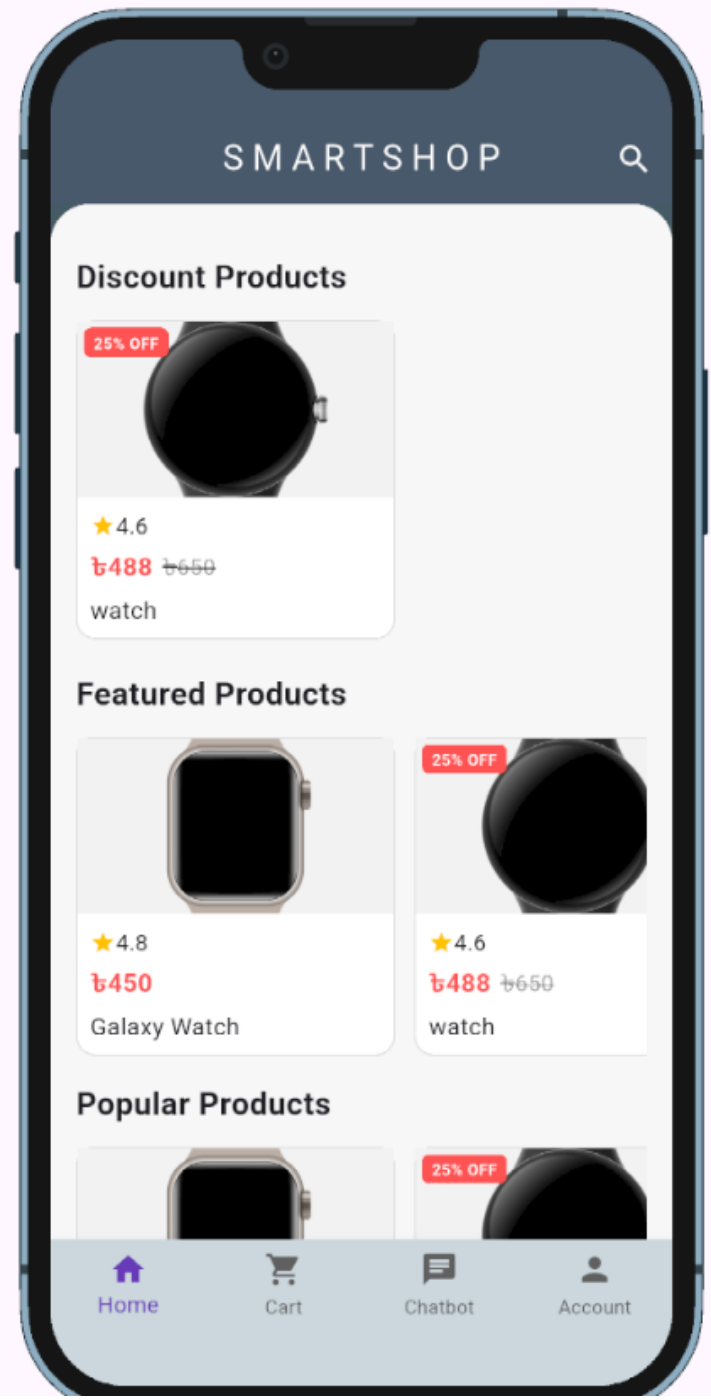
Sign In Screen



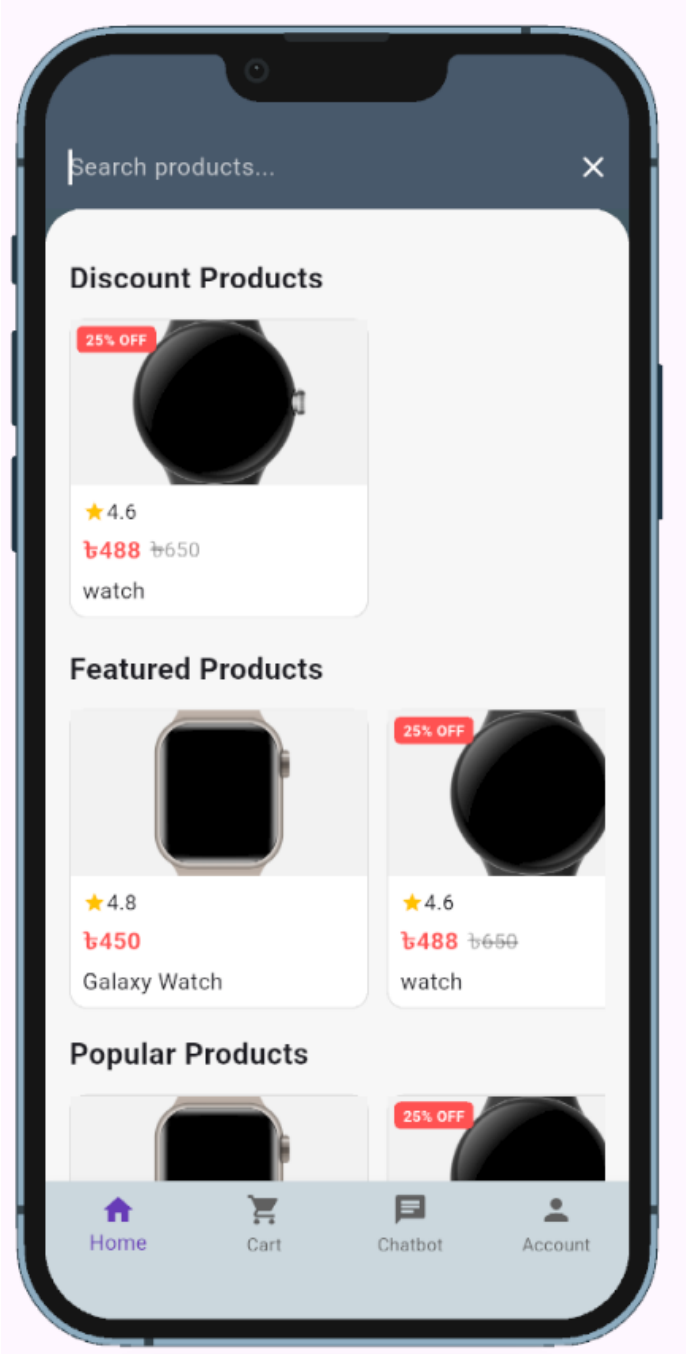
Sign Up Screen



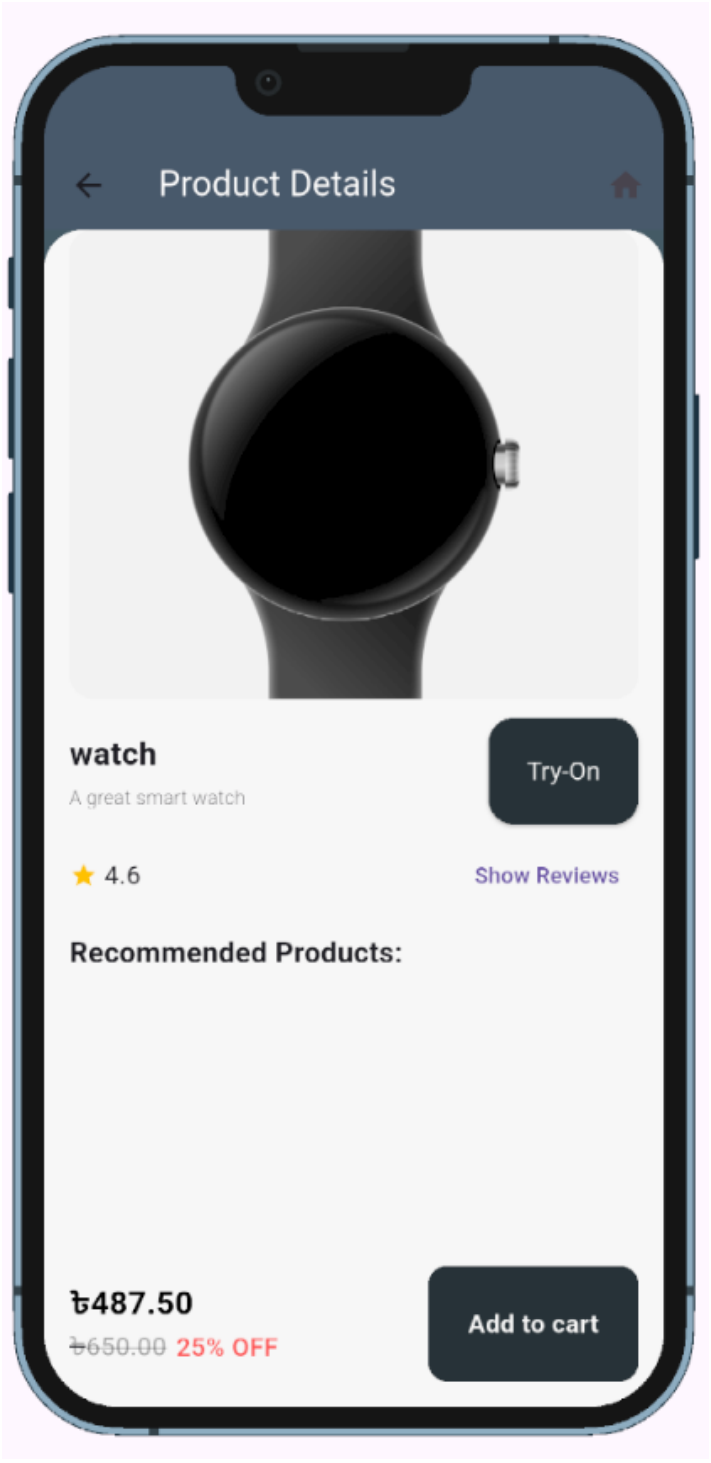
Dashboard Screen



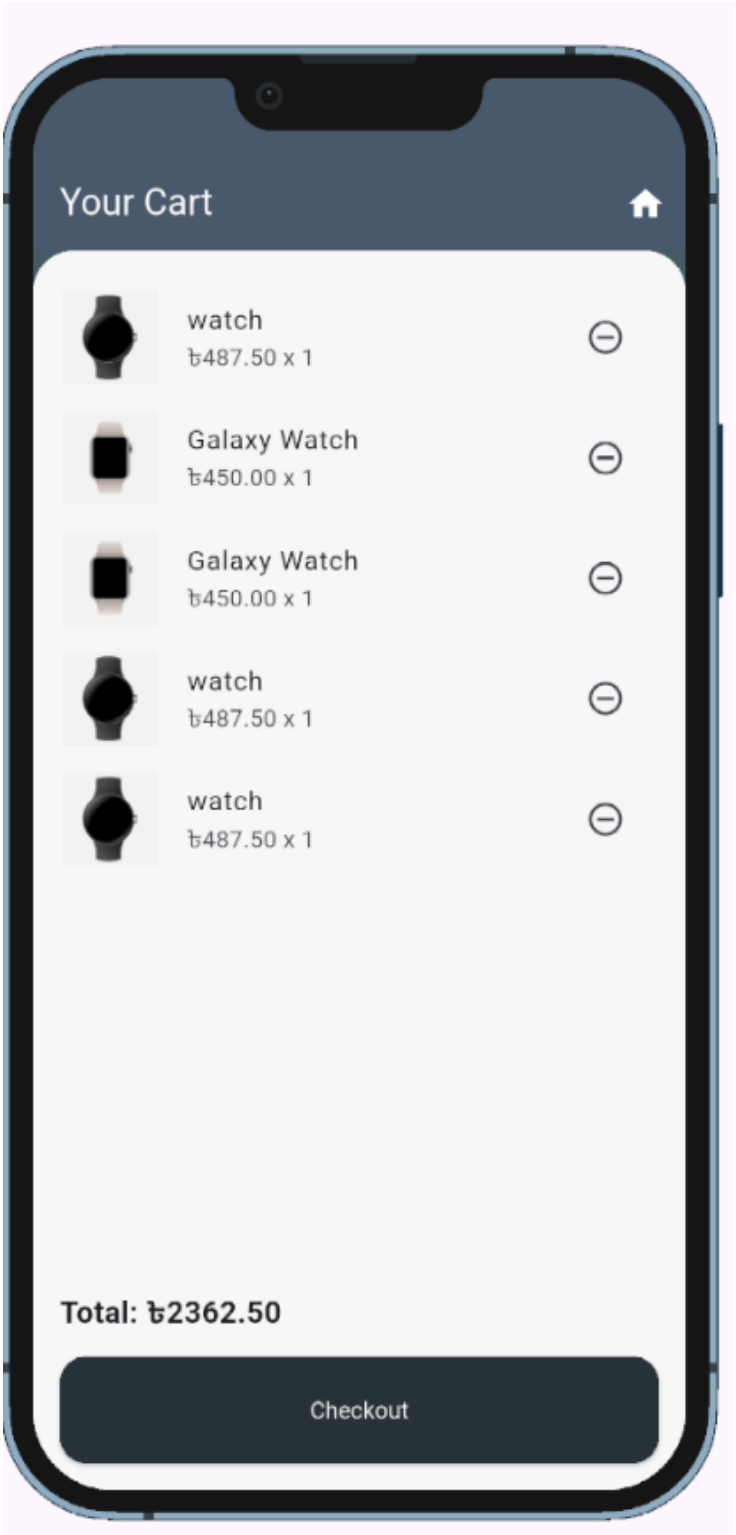
Search Products



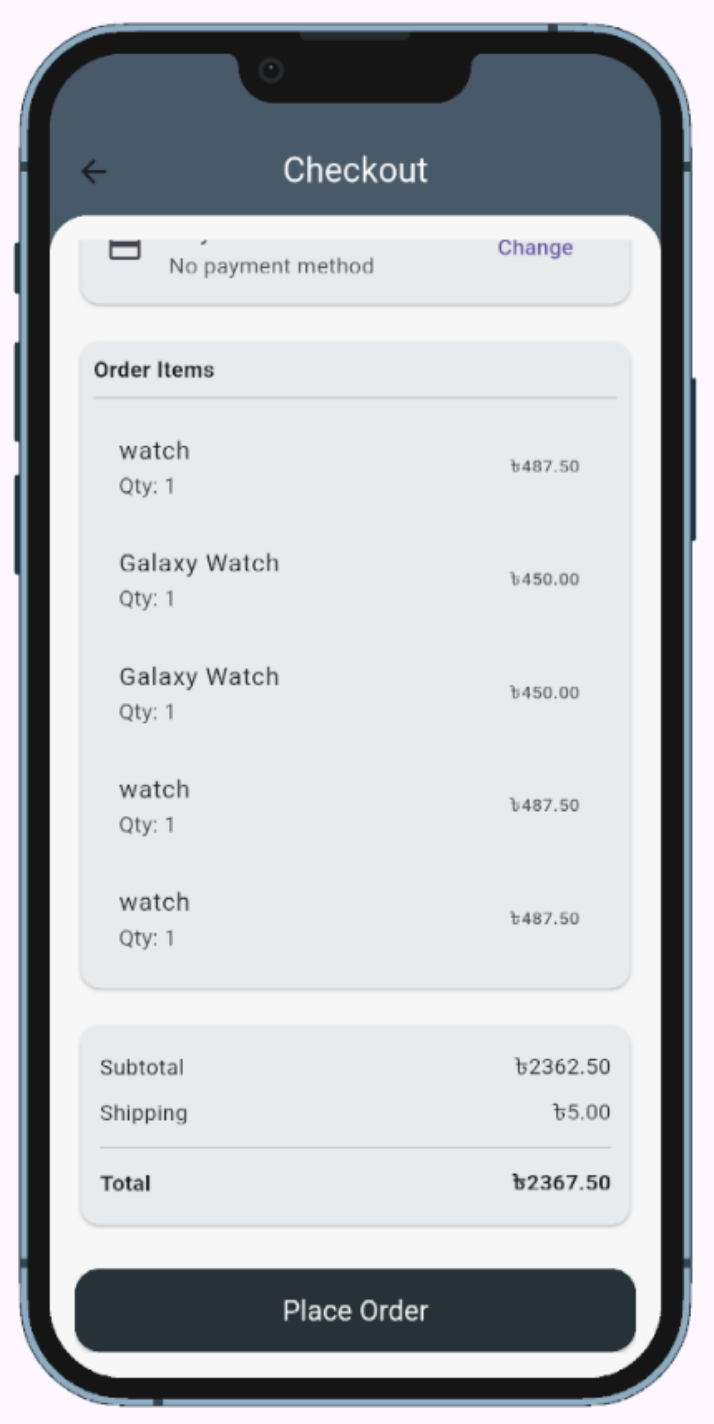
Product Details Screen



Cart Screen

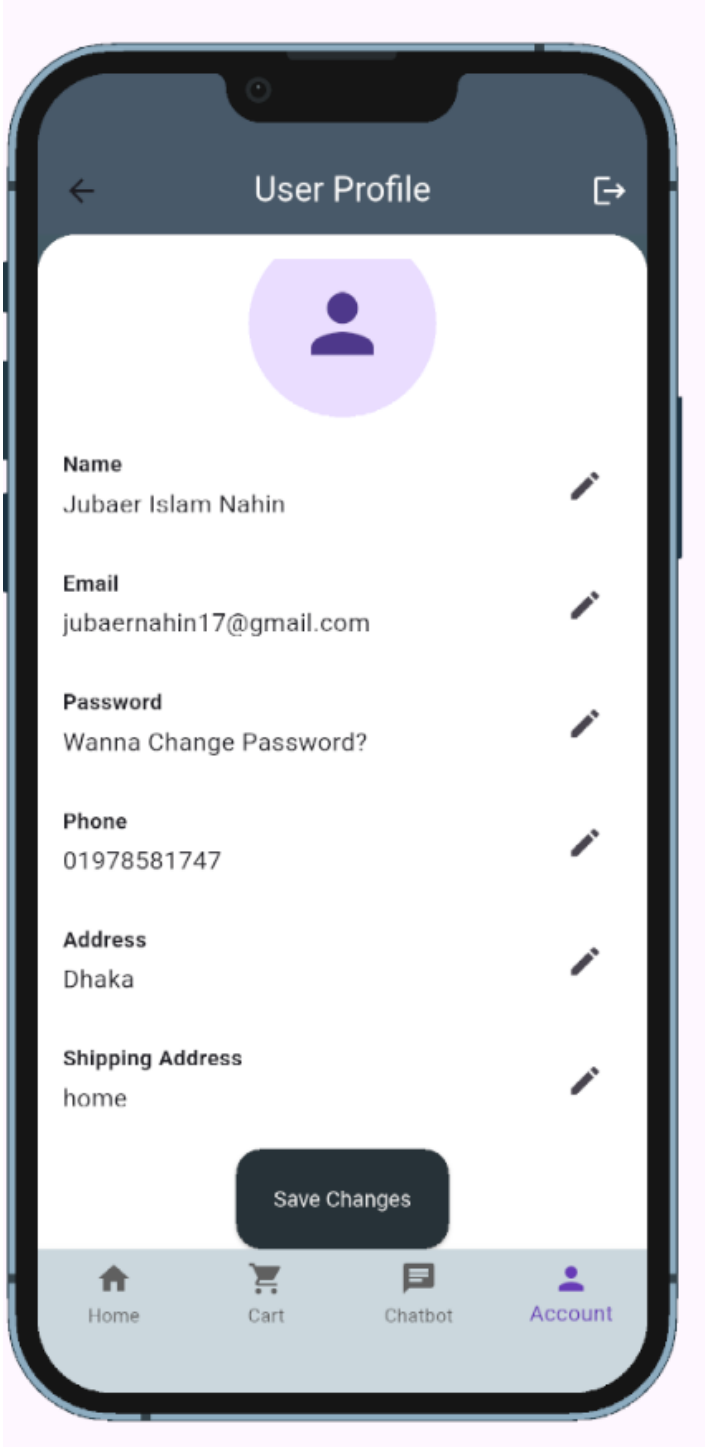


Checkout Screen



Chatbot Screen

User Profile Screen



5.3 Summary

In this chapter the project functionalities are shown. The pictures above are the introduction to the screens that will be provided in the app for the users.

Anyone can easily understand by just looking at the screens how things are going to work as the Ui is kept very much user friendly.

Chapter 6: Project Summary

6.1 Introduction

This chapter provides a summary of the Smartshop project, highlighting its limitations, scope, potential future enhancements, and overall accomplishments.

6.2 Project Limitations

The limited availability of time and resources hindered the complete development of advanced AI functionalities.

Constraints related to cloud and GPU resources impacted the accuracy and performance of the machine learning models.

Certain anticipated features—such as comprehensive virtual try-on capabilities and sophisticated admin analytics—are still in progress.

Limitations in the dataset have adversely affected the performance of the recommendation system and size prediction functionalities.

6.3 Scope

Included:

1. User authentication
2. Product catalog and details
3. Cart, wishlist, and checkout
4. Basic AI functionalities (visual search, simple recommendations)
5. Chatbot integration
6. Flutter-based mobile user interface with backend API support

Excluded:

1. Comprehensive AR and 3D virtual try-on
2. Advanced admin dashboard

3. Multi-vendor marketplace
4. Real-time delivery tracking
5. AI-driven pricing or analytics

6.4 Future Work

Accomplish complete AR/AI virtual try-on and 3D body modelling. Increase the precision of the prediction system and size prediction. Add a multi-vendor support, analytics and a seller dashboard. Add real-time tracking of the delivery process and loyalty. Enhance performance, security and machine learning model training.

6.5 Conclusion

Smartshop is an appropriate solution to offer a modern, AI-enhanced and e-commerce platform with basic shopping features and first AI-related features. Despite time and resource constraints that lead to the incompleteness of some of the advanced functionalities, the project sets a strong base and grounds the future developments and demonstrates the possibilities of integrating Flutter with AI technologies.

REFERENCES

1. Personalization in E-Commerce Applications Anna Goy, Liliana Ardissono, and Giovanna Petrone Department di Informatica, University's di Torino Corso Svizzera 185, Torino, Italy

2. Research on E-commerce Platform of Online Shopping Consumers Yidan Wang CLA College of Arts and Science, University of Minnesota, Twin Cities, Minneapolis, Minnesota, 55455, United State Corresponding author. Email: gaoming@cas-harbour.org

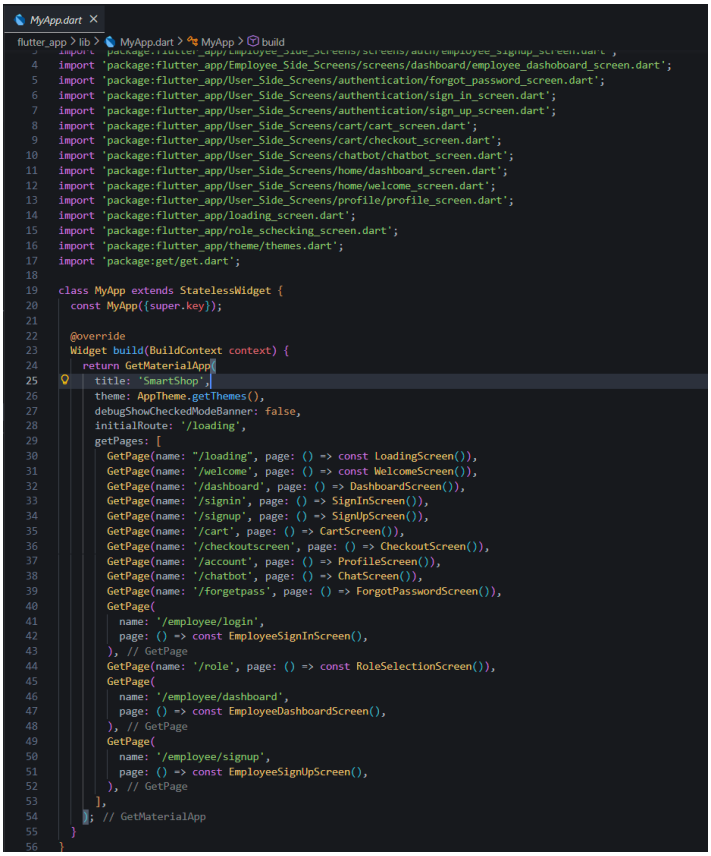
3. Dart Documentation
4. Flutter Docs
5. Flutter Packages
6. Firebase Documentation
7. TensorFlow Lite

APPENDICES

Appendix A: Coding

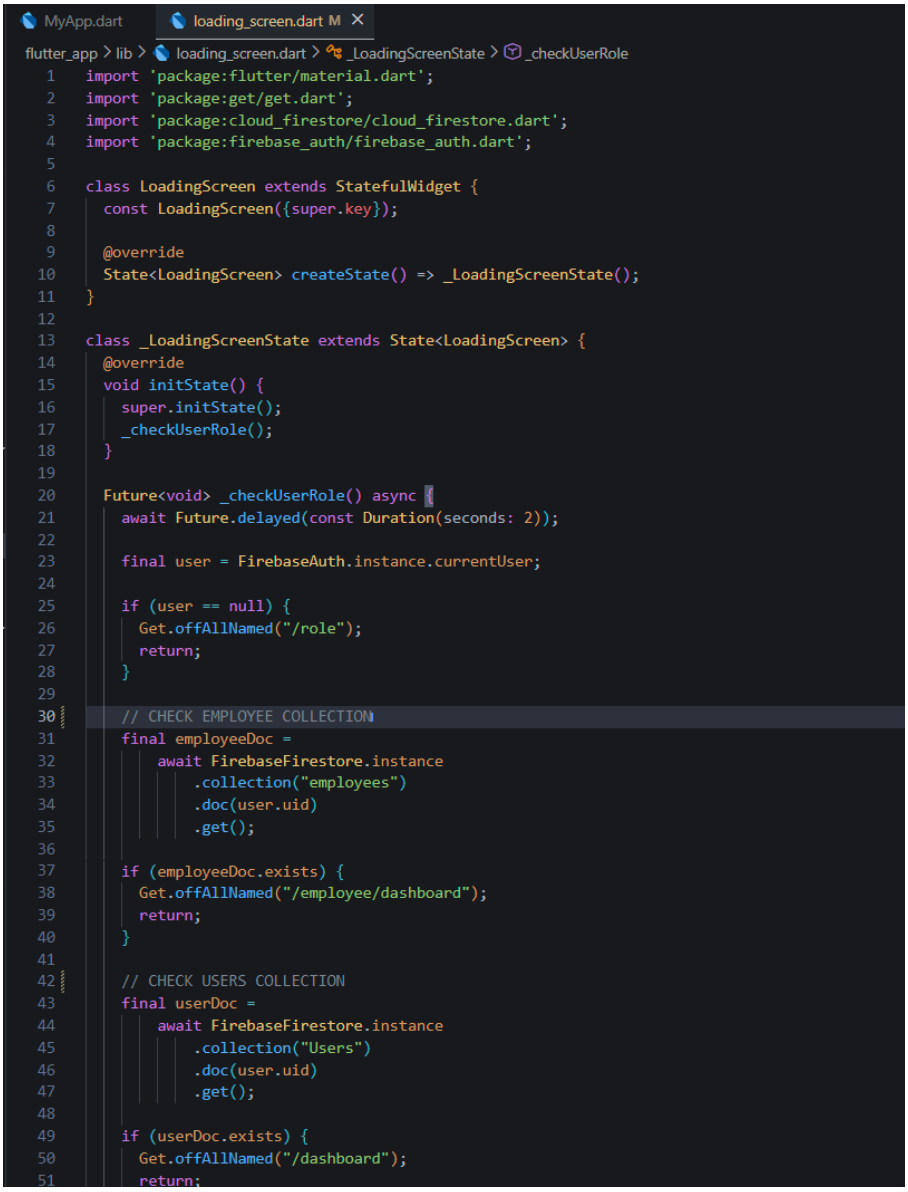
This main() function here is the core function which is the main entry point to ran the code.

This MyApp() function is defined mainly for managing all the routes and by default it provides app theme through GetMaterialApp() function as I used getx for state management, navigation and dependency injection.

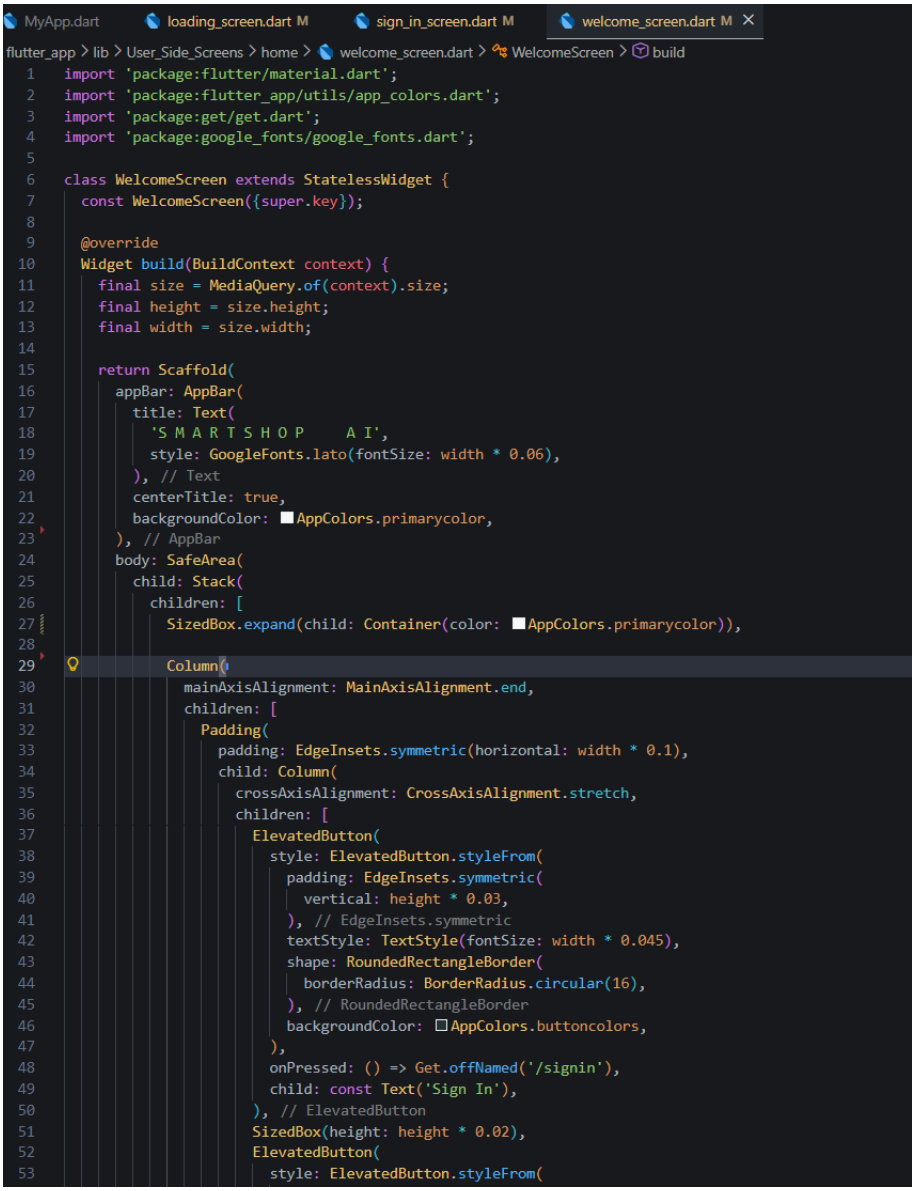


```dart
import 'package:flutter_app/Employee_Side_Screens/screens/dashboard/employee_dashoboard_screen.dart';
import 'package:flutter_app/User_Side_Screens/authentication/forgot_password_screen.dart';
import 'package:flutter_app/User_Side_Screens/authentication/sign_in_screen.dart';
import 'package:flutter_app/User_Side_Screens/authentication/sign_up_screen.dart';
import 'package:flutter_app/User_Side_Screens/cart/cart_screen.dart';
import 'package:flutter_app/User_Side_Screens/cart/checkout_screen.dart';
import 'package:flutter_app/User_Side_Screens/chatbot/chatbot_screen.dart';
import 'package:flutter_app/User_Side_Screens/home/dashboard_screen.dart';
import 'package:flutter_app/User_Side_Screens/home/welcome_screen.dart';
import 'package:flutter_app/User_Side_Screens/profile/profile_screen.dart';
import 'package:flutter_app/loading_screen.dart';
import 'package:flutter_app/role_schecking_screen.dart';
import 'package:flutter_app/theme/themes.dart';
import 'package:get/get.dart';

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'SmartShop',
      theme: AppTheme.getThemes(),
      debugShowCheckedModeBanner: false,
      initialRoute: '/loading',
      getPages: [
        GetPage(name: "/loading", page: () => const loadingScreen()),
        GetPage(name: '/welcome', page: () => const WelcomeScreen()),
        GetPage(name: '/dashboard', page: () => DashboardScreen()),
        GetPage(name: '/signin', page: () => SignInScreen()),
        GetPage(name: '/signup', page: () => SignUpScreen()),
        GetPage(name: '/cart', page: () => CartScreen()),
        GetPage(name: '/checkoutscreen', page: () => CheckoutScreen()),
        GetPage(name: '/account', page: () => ProfileScreen()),
        GetPage(name: '/chatbot', page: () => ChatScreen()),
        GetPage(name: '/forgetpass', page: () => ForgotPasswordScreen()),
        GetPage(
          name: '/employee/login',
          page: () => const EmployeeSignInScreen(),
        ), // GetPage
        GetPage(name: '/role', page: () => const RoleSelectionScreen()),
        GetPage(
          name: '/employee/dashboard',
          page: () => const EmployeeDashboardScreen(),
        ), // GetPage
        GetPage(
          name: '/employee/signup',
          page: () => const EmployeeSignUpScreen(),
        ), // GetPage
      ],
    ); // GetMaterialApp
  }
}
```

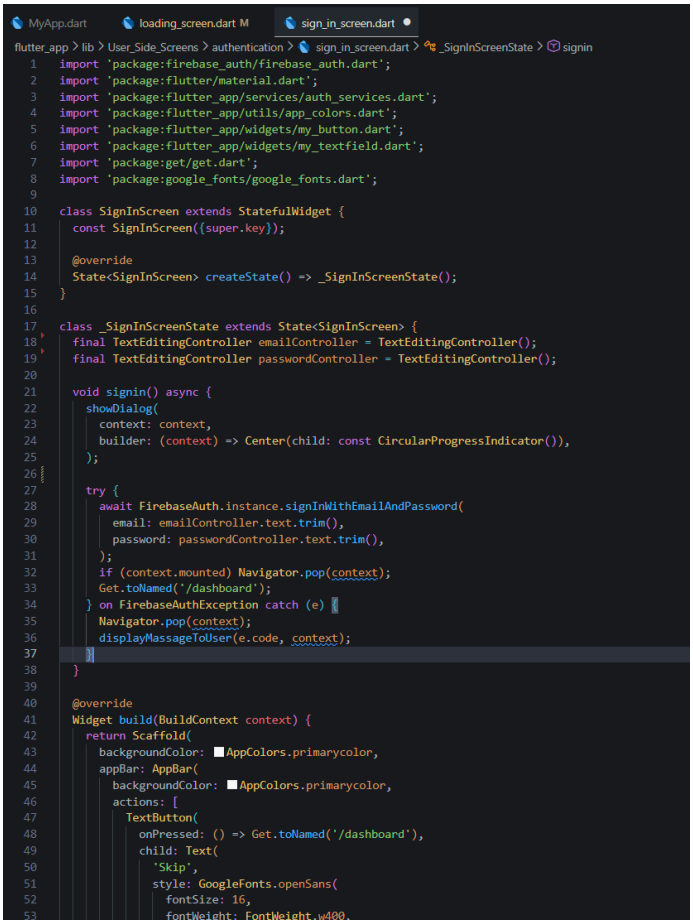
This is the loading screen which checks if any user customer or employee is already logged in or not if not logged in it takes them to the welcome screen.



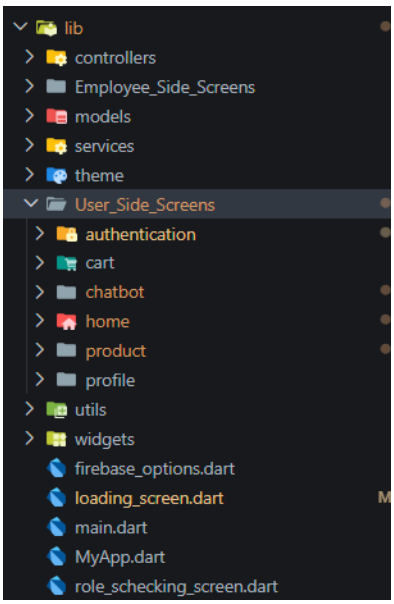
Welcome Screen let's the user decide If they want to sign in or signup.



The Sign in screen let's user sign in to the app.
If not signed in user can also sign up, and if the user also want to enter the app just for visiting they can also do it as there is a skip button which will directly lead them to the dashboard.

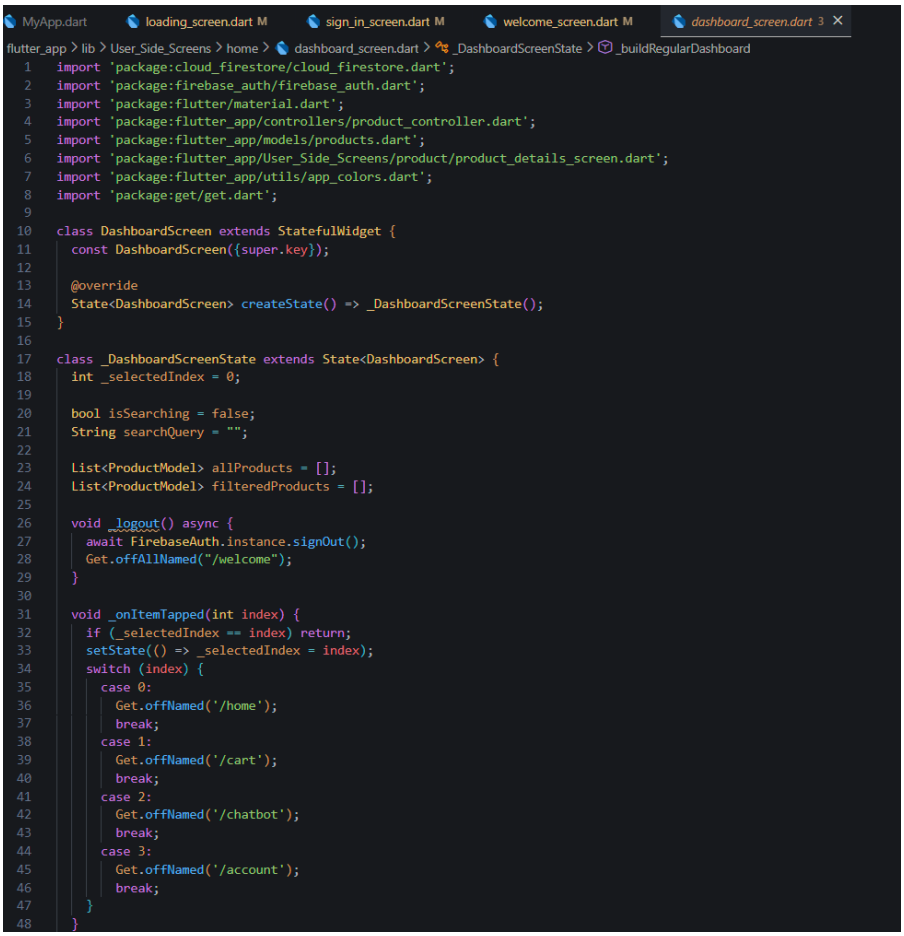


Stakeholder's screens are separated in the project structure for better code readability



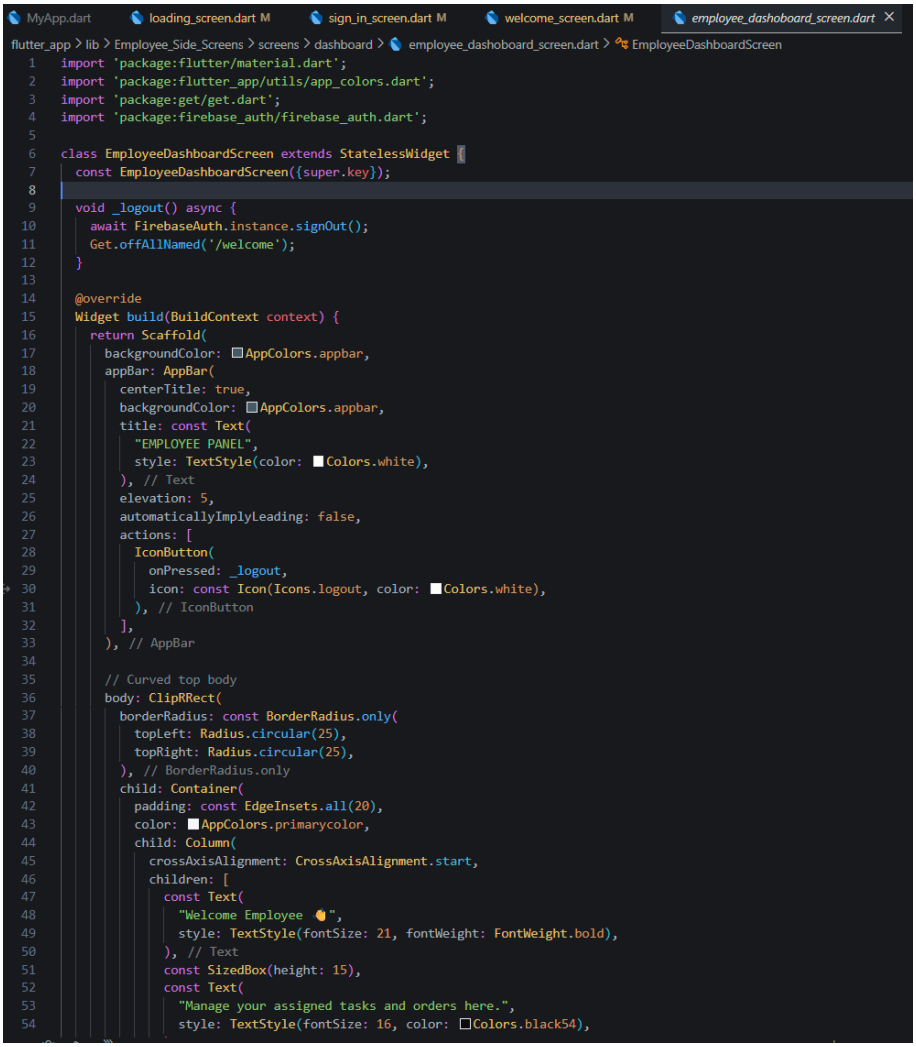
Dashboard for user and employee are also different below here the coding examples of both side dashboards are given:

This is the User side dashboard:

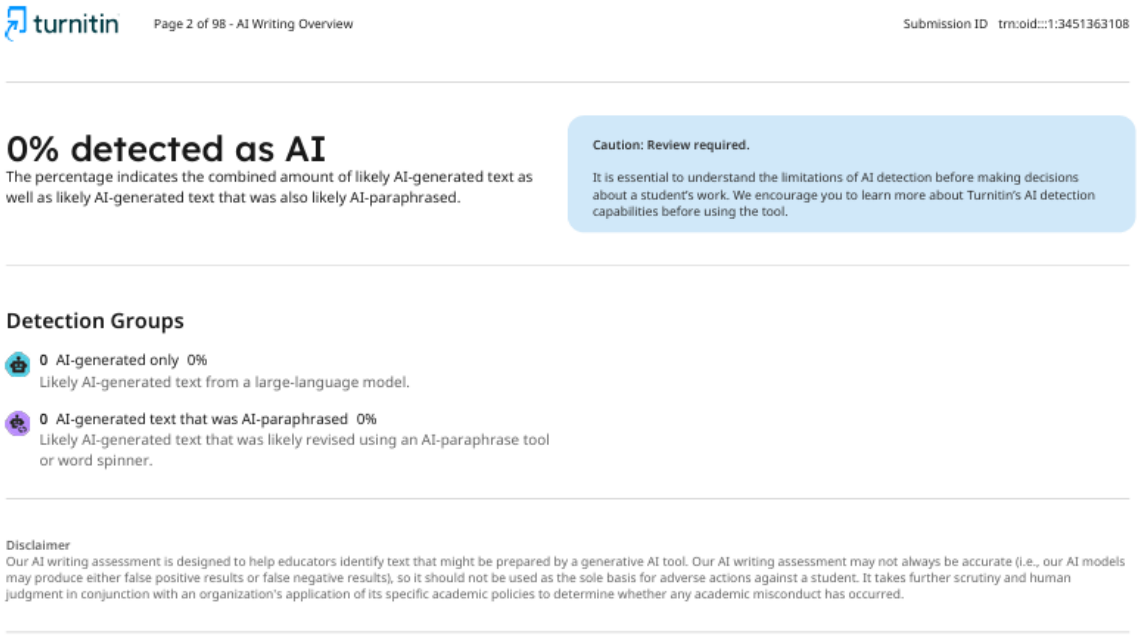


```dart
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_app/controllers/product_controller.dart';
import 'package:flutter_app/models/products.dart';
import 'package:flutter_app/User_Side_Screens/product/product_details_screen.dart';
import 'package:flutter_app/utils/app_colors.dart';
import 'package:get/get.dart';

class DashboardScreen extends StatefulWidget {
  const DashboardScreen({super.key});

  @override
  State<DashboardScreen> createState() => _DashboardScreenState();
}

class _DashboardScreenState extends State<DashboardScreen> {
  int _selectedIndex = 0;

  bool isSearching = false;
  String searchQuery = "";

  List<ProductModel> allProducts = [];
  List<ProductModel> filteredProducts = [];

  void _logout() async {
    await FirebaseAuth.instance.signOut();
    Get.offAllNamed("/welcome");
  }

  void _onItemTapped(int index) {
    if (_selectedIndex == index) return;
    setState(() => _selectedIndex = index);
    switch (index) {
      case 0:
        Get.offNamed('/home');
        break;
      case 1:
        Get.offNamed('/cart');
        break;
      case 2:
        Get.offNamed('/chatbot');
        break;
      case 3:
        Get.offNamed('/account');
        break;
    }
  }
```

This is the employee side dashboard:



```dart
import 'package:flutter/material.dart';
import 'package:flutter_app/utils/app_colors.dart';
import 'package:get/get.dart';
import 'package:firebase_auth/firebase_auth.dart';

class EmployeeDashboardScreen extends StatelessWidget {
  const EmployeeDashboardScreen({super.key});

  void _logout() async {
    await FirebaseAuth.instance.signOut();
    Get.offAllNamed('/welcome');
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: AppColors.appbar,
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: AppColors.appbar,
        title: const Text(
          "EMPLOYEE PANEL",
          style: TextStyle(color: Colors.white),
        ), // Text
        elevation: 5,
        automaticallyImplyLeading: false,
        actions: [
          IconButton(
            onPressed: _logout,
            icon: const Icon(Icons.logout, color: Colors.white),
          ), // IconButton
        ],
      ), // AppBar

      // Curved top body
      body: ClipRRect(
        borderRadius: const BorderRadius.only(
          topLeft: Radius.circular(25),
          topRight: Radius.circular(25),
        ), // BorderRadius.only
        child: Container(
          padding: const EdgeInsets.all(20),
          color: AppColors.primarycolor,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const Text(
                "Welcome Employee 👋",
                style: TextStyle(fontSize: 21, fontWeight: FontWeight.bold),
              ), // Text
              const SizedBox(height: 15),
              const Text(
                "Manage your assigned tasks and orders here.",
                style: TextStyle(fontSize: 16, color: Colors.black54),
```



turnitin Page 2 of 98 - AI Writing Overview Submission ID trn:oid:::1:3451363108

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

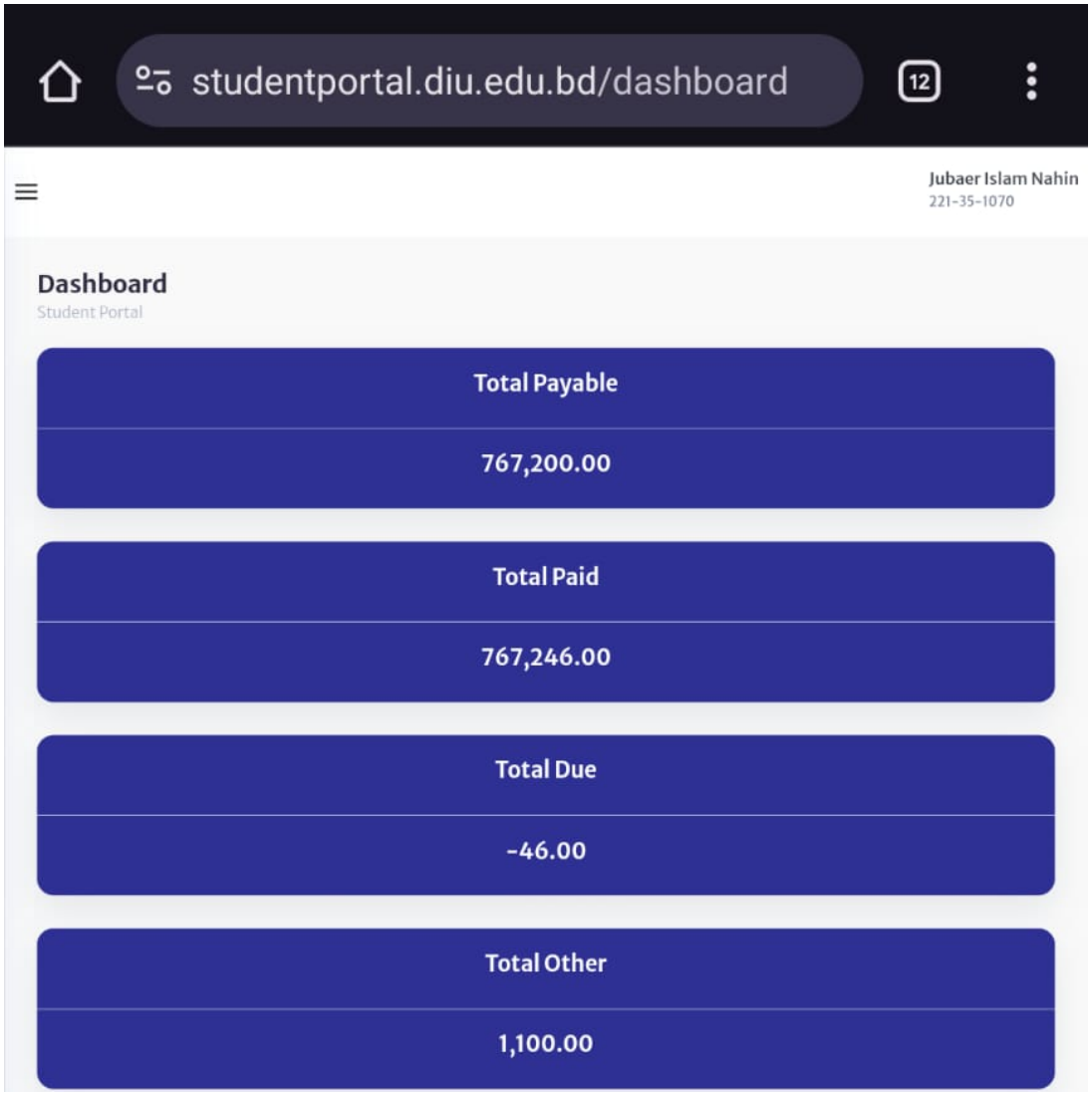
It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

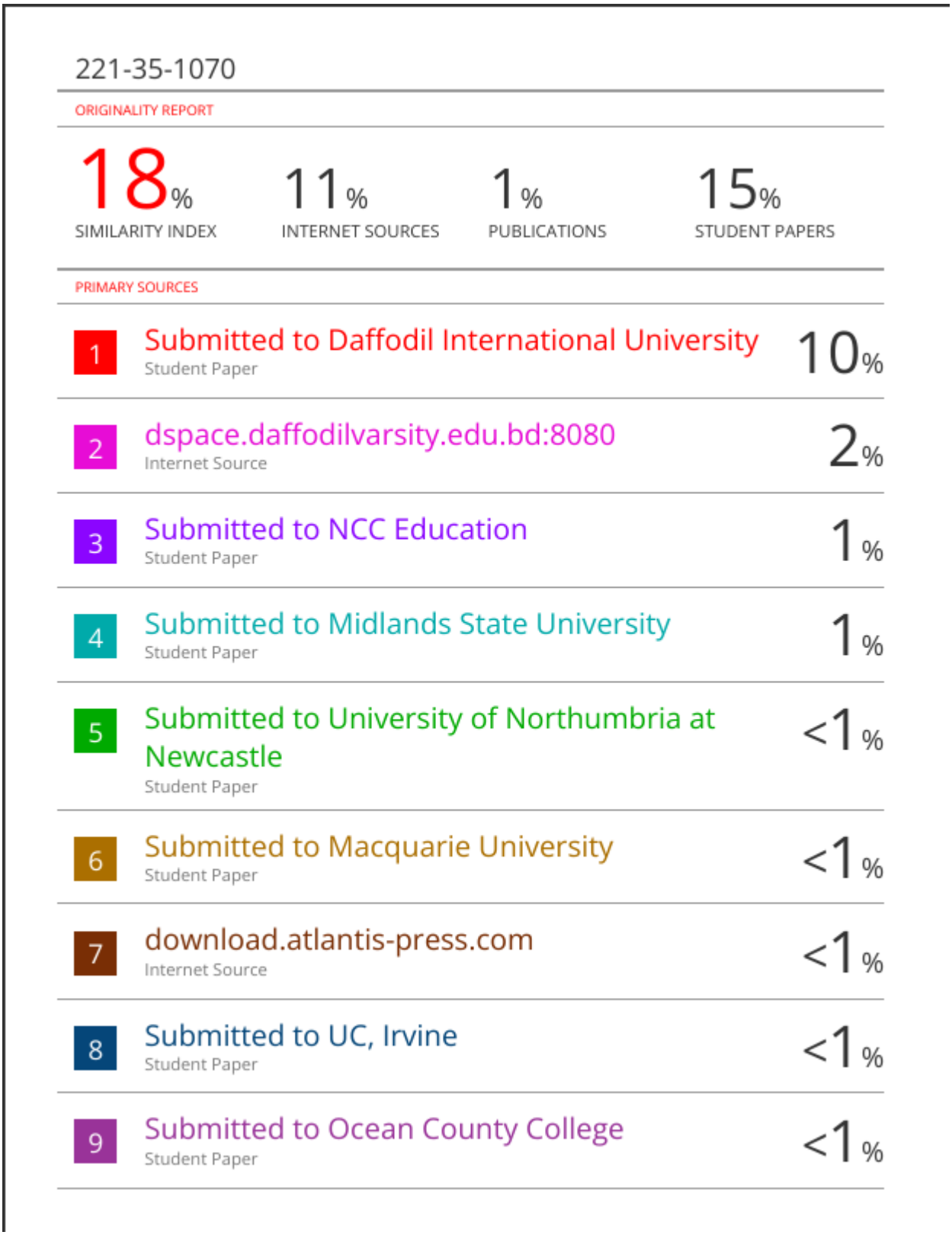
Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.





221-35-1070

ORIGINALITY REPORT

18%	11%	1%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	10%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	Submitted to NCC Education Student Paper	1%
4	Submitted to Midlands State University Student Paper	1%
5	Submitted to University of Northumbria at Newcastle Student Paper	<1%
6	Submitted to Macquarie University Student Paper	<1%
7	download.atlantis-press.com Internet Source	<1%
8	Submitted to UC, Irvine Student Paper	<1%
9	Submitted to Ocean County College Student Paper	<1%