



Daffodil
International
University

FINAL YEAR PROJECT REPORT

A ONLINE T-SHIRT CUSTOMIZATION WEBSITE "CUSTOMIZE TEE"

SUBMITTED BY

Md. Sabbir Ahmmed Akash

ID: 181-35-2356

Batch: 25th

Department of Software Engineering
Daffodil International University

This Report is Submitted in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Software Engineering.

SUPERVISED BY

Mr. Nuruzzaman Faruqui

Assistant Professor

Department of Software Engineering
Daffodil International University

APPROVAL

APPROVAL

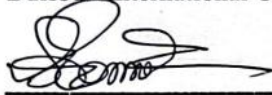
This project titled on "Customize Tee", submitted by **Sabbir Ahmmmed Akash (ID: 181-35-2356)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



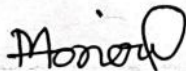
Dr. Imran Mahmud
Associate Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



Nuruzzaman Faruqi
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Md. Monirul Islam
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2



Dr. Md. Sazzadur Rahman
Associate Professor
Institute of Information Technology
Jahangirnagar University

External Examiner

DECLARATION

I affirm that I independently completed this project under the guidance of **Mr. Nuruzzaman Faruqi, Assistant Professor, Department of Software Engineering**, Daffodil International University. Additionally, I confirm that neither this work nor any portion of this project has been presented elsewhere for the attainment of any degree or certification.

Supervised By



16-01-27

Mr. Nuruzzaman Faruqi

Assistant Professor
Department of Software Engineering.
Daffodil International University.

Submitted By



Md. Sabbir Ahmmed Akash

ID: 181-35-2356
Batch: 25th
Department of Software Engineering
Daffodil International University

ACKNOWLEDGEMENT

Firstly, I extend my deepest gratitude to the Almighty God for His divine blessings, which enabled us to successfully complete the final year project.

I am sincerely thankful and express our profound indebtedness to **Mr. Nuruzzaman Faruqui, Assistant Professor, Department of Software Engineering** at Daffodil International University, Dhaka. His extensive knowledge and keen interest in the field of "Web Development" played a crucial role in the execution of this project. The unwavering patience, scholarly guidance, continuous encouragement, energetic supervision, constructive criticism, valuable advice, meticulous reading of numerous drafts, and corrections at every stage were instrumental in the successful completion of this project.

Special thanks are due to Md Khaled Sohel, Dr. Md. Fazla Elahe, Ms. Tapushe Rabaya Toma Mr. Md. Monirul Islam, and all my fellow course mates at Daffodil International University who actively participated in discussions during the course work.

Lastly, I must express sincere appreciation and gratitude for the constant support and patience of my parents.

ABSTRACT

Customize Tee is an innovative and interactive online platform developed to revolutionize the personalized fashion industry. Leveraging modern web technologies, the project offers a dynamic space for users to design, customize, and purchase personalized t-shirts tailored to their unique preferences.

Utilizing a robust tech stack, including a user-friendly front-end powered by React.js, a scalable back-end implemented with Node.js, Express.js facilitating API creation, and MongoDB for seamless data storage, Customize Tee embodies a comprehensive and efficient MERN (MongoDB, Express.js, React.js, Node.js) stack architecture.

The primary objective of Customize Tee is to empower users, including individuals and businesses, to effortlessly express their identity through custom apparel. The platform facilitates a streamlined process for designing and ordering personalized t-shirts, ensuring a delightful and intuitive user experience.

Key Features:

- **User-Friendly Design Tools.** Customize Tee will provide an intuitive and feature rich interface, allowing users to easily design and customize their T-shirts. The platform will offer a comprehensive set of tools that empower users to bring their creative visions to life.

This defense project will delve into the goals, objectives, key features, and the intricacies of the implementation process behind Customize Tee. By combining technology and creativity, Customize Tee aims to redefine personalized fashion, providing a convenient and thorough online storefront for users to explore and purchase a diverse array of customized t-shirts.

TABLE OF CONTENTS

CONTENTS	PAGE
Acknowledgement	III
Abstract	IV
Table of Contents	V-VII
CHAPTER I: INTRODUCTION	1-4
1.1 Project Overview	1
1.2 Project Purpose	1
1.2.1 Motivation	1
1.2.2 Objectives	2
1.2.3 Benefits & Beneficiaries	2
1.2.4 Stakeholders	2
1.2.5 Proposed System Model	3
1.3 Project Schedule	4
1.3.1 Gantt Chart	4
CHAPTER 2 SYSTEM ANALYSIS	5-9
2.1 Feasibility Analysis	5
2.1.1 Technical Feasibility	5
2.1.2 Operational Feasibility	5
2.2 Functional Requirements	6-7
2.3 Non-Functional Requirements	8
2.4 System Requirements	9
2.4.1 Hardware Requirements	9
2.4.2 Software Dependencies	9
CHAPTER 3 SYSTEM DESIGN	10-55
3.1 Development Model	10
3.2 Use Case Diagram	11
3.3 Use Case Descriptions	12-25
3.4 Activity Diagram	26-40
3.5 Sequence Diagram	41-54
3.6 ERD Diagram	55
CHAPTER 4 DEVELOPMENT TOOL & TECHNOLOGY	56-62
4.1 Integrated Development Environment (IDE)	56

4.2 Programming Language	57
4.3 User interface Design	58-59
4.4 Database	59-60
4.5 Deploy and hosting	60-61
4.6 Additional Tools	61-62
4.7 Dependencies	62
CHAPTER 5 SYSTEM TESTING	63-73
5.1 Testing Features	63
5.1.1 Feature to be tested	63
5.2 Testing Strategies	64
5.2.1 Test Approach	64
5.2.2 Pass/Fail Criteria	64
5.2.3 Testing Schedule	64
5.3 Test Cases	65-73
CHAPTER 6 USER MANUAL	74-94
6.1 CUSTOMIZE TEE - Home Page	74-77
6.2 CUSTOMIZE TEE - Login Page	77
6.3 CUSTOMIZE TEE - User Registration Page	78
6.4 CUSTOMIZE TEE - View Products	79
6.5 CUSTOMIZE TEE - Add To Cart	80
6.6 CUSTOMIZE TEE - My Cart	81
6.7 CUSTOMIZE TEE - Manage Cart	82
6.8 CUSTOMIZE TEE - Place Order	83
6.9 CUSTOMIZE TEE - Make Payment	84
6.10 CUSTOMIZE TEE - Design Custom T-shirt	85
6.11 CUSTOMIZE TEE - User Dashboard	86
6.12 CUSTOMIZE TEE - Update Profile	87
6.13 CUSTOMIZE TEE - User Review	88
6.14 CUSTOMIZE TEE - User Order History	89
6.15 CUSTOMIZE TEE - Admin Dashboard	89
6.16 CUSTOMIZE TEE - Add Product	90
6.17 CUSTOMIZE TEE - Manage Product	91
6.18 CUSTOMIZE TEE - Manage Order	92

6.19 CUSTOMIZE TEE - Manage User	93
6.20 CUSTOMIZE TEE - Logout	94
CHAPTER 7 CONCLUSION	95-97
7.1 Project Summery	95
7.2 Limitations	96
7.3 Future Scope	96-97
7.4 Project Link	97
APPENDICES	98-105
REFERENCES	106
PLAGIARISM TEST	107

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Customize Tee is a comprehensive online platform developed for personalized t-shirt design and purchase. It caters to individuals, groups and businesses seeking a unique and creative outlet for expressing their identity through custom apparel. Our project brings creativity and fashion together, offering an innovative online platform for designing, ordering, and customizing t-shirts.

1.2 Project Purpose

1.2.1. Motivation

Nowadays Customize clothing fashion are on demand. Everyone wants to wear their own design clothes like T-shirts, Shirts, Hoodies, Caps e.t.c. Because now a days people have their own taste of choice. And the most using clothing product is T-Shirt.

As we know in Bangladesh we have strength in Garments sector. So we can easily build a fashion brand that's gives people customize T-shirts or Tees. In other countries they also start this kind of project now a days. So Why don't we undertake this project, especially when our garments industry in Bangladesh is exceptionally strong?.

While exploring online platforms for customizing t-shirts in our country, I noticed a common issue – design placement constraints on certain websites. To address this, I envisioned a platform, Customize Tee, where users have complete freedom in placing their designs. This user-centric approach aims to simplify the customization process, providing a hassle-free solution for individuals looking to design t-shirts for personal use, drop-shipping businesses, or events and offering a platform that empowers creativity and simplifies the lives of users.

1.2.2.Objectives

Customize Tee will be a web based system which will resolve the traditional problem of customizing tees. A web base solution will be certainly more accurate than a traditional base solution. Basically the system will have different level of user. The one who is not registered as a user will only be able to view the featured tees and home page with basic information.

The list of operations that the system will provide are-

- Providing Smooth UI based customize section for users.
- Maintain and record user details, designs and orders details of the registered Customers.
- Enabling the Admin to manage products, designs ,orders and user details.
- Enabling the Customer to view designs,manage order and personal details..
- Enabling the Customer to place his design any edge of the t-shirt as desired and design their custom t-shirt with their own designs.
- Solve Bulk Order Problem.

1.2.3 Benefits & Beneficiaries

The system offers customers a user-friendly interface, making it easy for them to navigate. Operating as an online platform, customers have the convenience of browsing products, making purchases, and monitoring orders from the comfort of their homes. The flexibility allows customers to make purchases at any time and from any location without the constraints found in physical stores or garments.

Benefits will be:

- Everything in one platform.
- User will be able to design own t-shirt and make purchase.

Beneficiaries:

- Customize t-shirt audience.
- Small t-shirt customization businesses.

1.2.4 Stakeholders

Stakeholders play a pivotal role in the success and impact of a system or project. In the context of "Customize Tee," various stakeholders contribute to and benefit from the platform.

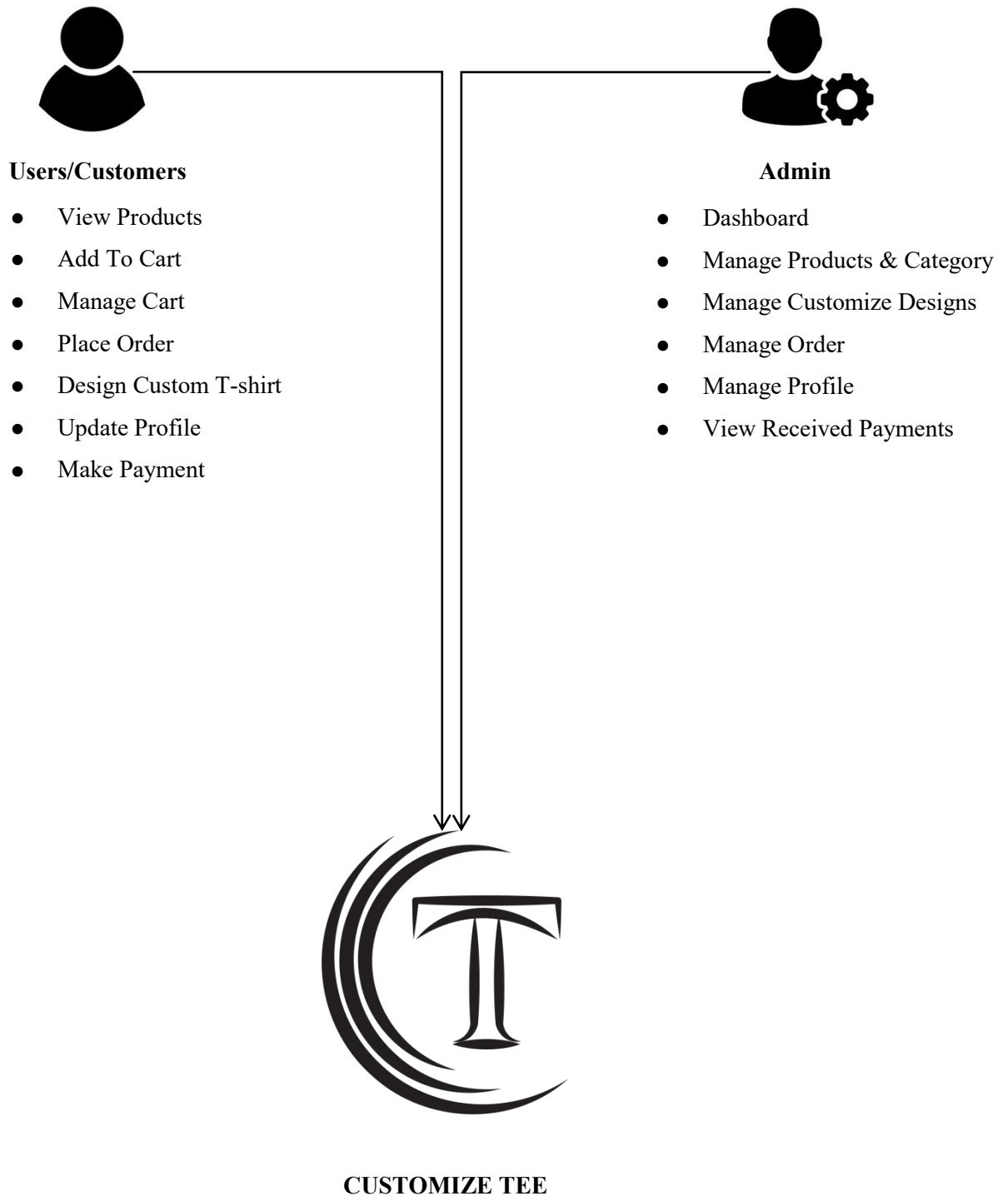
The primary stakeholders include:

- User/Customer.
- Admin.

1.2.5 Proposed System Model

This model aims to simulate the system architecture and is designed to be as user-friendly as possible.

Figure 1.1: Proposed System Model



1.3 Project Schedule

A project schedule outlines the necessary features to be developed, identifies the required resources, and establishes deadlines for project completion. Given the short development time for the system, it is essential to create an effective schedule to ensure timely project completion. This schedule also facilitates communication about tasks that need to be accomplished within the specified time frame.

1.3.1 Gantt Chart

Table 1.1: Gantt Chart

Duration(Week)														
Activities	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14
Research	■	■												
Project Planning		■	■	■										
Requirements Analysis			■	■	■	■								
Designing				■	■	■								
Coding						■	■	■	■					
Testing										■	■	■		
Finalization													■	■

CHAPTER 2

SYSTEM ANALYSIS

2.1 Feasibility Analysis

Feasibility analysis is a crucial evaluation of the system proposal, examining its workability, impact on the organization, ability to meet requirements, and efficient utilization of resources. This analysis is essential to determine whether the proposed system is viable and if it addresses the issues of previous projects effectively. The feasibility study encompasses the following criteria:

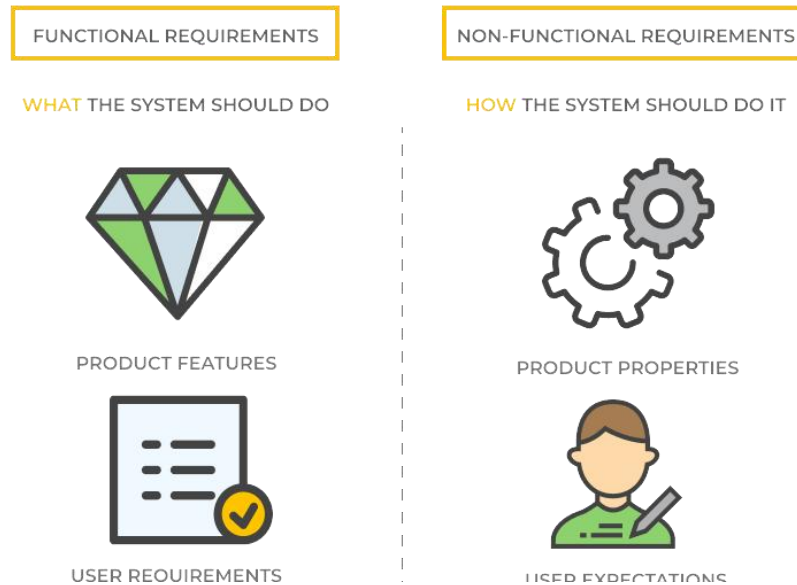
2.1.1 Technical Feasibility

Technical feasibility assesses the availability of resources for the project. It involves an examination of the technical aspects behind the system to determine its feasibility. In the case of the "Customize Tee" project, the technical requirements are relatively basic. The system can be developed using common programming languages such as JavaScript, CSS, and HTML. Therefore, from a technical perspective, the proposed system is deemed feasible.

2.1.2 Operational Feasibility

Operational feasibility studies whether the project can be completed within the given deadline. It assesses whether the project is feasible within the allocated time frame or if it is too extensive to be completed within the stipulated period. The "Customize Tee" project aims to be operationally feasible, with a realistic timeline for completion. This ensures that the project aligns with the specified time constraints and can be successfully executed within the given deadline.

Figure 2.1: Functional & Non-Functional Requirement



2.2 Functional Requirements

Functional requirements are critical features of a project that must be implemented, outlining what the project is designed to accomplish. It is imperative to fulfill all functional requirements. The Functional Requirements for "CUSTOMIZE TEE" include:

For Customer/User

- Registration
- Login & Logout
- View Products
- Add To Cart
- Manage Cart
- Place Order
- Design Custom T-shirt
- Update Profile
- Make Payment

For Admin

- Login & Logout
- Manage Product
- Manage Order
- Manage User
- Receive Payment

Table 2.2: Functional Requirements

SRS No	Name	Descriptions
01	Registration	Customers need to do registration for Login to access the system.
02	Login & Logout	Customers and Admin needed login for using this system.
03	View Products	Customers can see all products and categories in the system.
04	Add To Cart	Customers can add their desire t-shirt from the system in cart.
05	Manage Cart	Customer can manage their cart.Like remove product from cart.
06	Place Order	Customer can place their order with shipping information.
07	Design Custom T-shirt	Customer can Create their own design t-shirt by placing or making own designs in the system and just add the design on any position of a t-shirt.
08	Update Profile	Customer can Update their information.
09	Make Payment	Customers can make their payment using payment gateway.
10	Manage Product	Admin can add,remove,edit products and categories in the system.
11	Manage Order	Admin will confirm or reject placed order by the customer and also able to update the status.
12	Manage User	Admin can see all the users of the system .Can add, remove users, and also can manage role of user in the system.
13	Receive Payment	Admin can see Receive payments for products and then he can update customers order status.

2.3 Non-Functional Requirements

Non-functional requirements are the quality attribute of a system. It defines how a system fulfills the functional requirements. This requirement must be met to ensure the quality. The Non-Functional Requirements for "CUSTOMIZE TEE" include:

- Performance
- Responsibility
- Maintainability
- Security
- Usability
- Availability
- Capacity
- Integrity

Table 2.3: Non-Functional Requirements

SRS No	Name	Descriptions
01	Performance	The system is expected to provide all services without any faults, ensuring a seamless user experience.
02	Responsibility	The system should respond promptly, aiming for minimal response times to enhance user satisfaction.
03	Maintainability	Admins will have the capability to maintain and update user data and other relevant information in a dedicated area, ensuring the system's ongoing adaptability.
04	Security	All data within the system, including the database, will be secured and encrypted. The use of cryptography techniques will bolster the overall security posture.
05	Usability	The system is designed to be user-friendly, offering an intuitive interface for users to easily navigate and utilize its features.
06	Availability	The system is expected to be available at all times, ensuring continuous access for users without interruptions.
07	Capacity	The system should have the capacity to handle a growing user base and an expanding catalog of designs without compromising performance.
08	Integrity	Data integrity is paramount; the system should enforce data consistency, accuracy, and reliability throughout all interactions.

2.4 System Requirements

System requirements define the hardware, software, network, and other specifications needed for the successful development and operation of the proposed system. These requirements provide a comprehensive overview of the necessary resources and infrastructure.

2.4.1 Hardware Requirements

1. Server:

- Processor: Multi-core processor with moderate clock speed (e.g., Quad-core or Hexa-core)
- RAM: 8GB to 16GB DDR4 RAM
- Architecture: 64-bit architecture
- Virtualization Support: Virtualization technology support for efficient resource utilization
- Network Interface: Gigabit Ethernet for high-speed network connectivity

2. Storage:

- Storage Type: SSD (Solid State Drive) or HDD (Hard Disk Drive)
- Capacity: 256GB to 512GB SSD or 1TB to 2TB HDD
- RAID Support: Optional RAID configuration for data redundancy and improved performance
- Read/Write Speed: Balanced read/write speeds for optimal performance

3. Network:

- Network Connectivity: Gigabit Ethernet (10/100/1000 Mbps)
- Wireless: Optional Wi-Fi support for flexibility in network connectivity
- Network Security: Support for standard security protocols (WPA3 for Wi-Fi, firewall settings)
- Bandwidth: Adequate bandwidth to handle network traffic efficiently

2.4.2 Software Dependencies

1. Web Hosting:

- Firebase Hosting

2. Database:

- MongoDB

3. Programming Languages:

- Front-end Development- React (JavaScript library for building user interfaces)
- Back-end Development- Node.js (JavaScript runtime) and Express.js (web application framework for Node.js)

CHAPTER 3

SYSTEM DESIGN

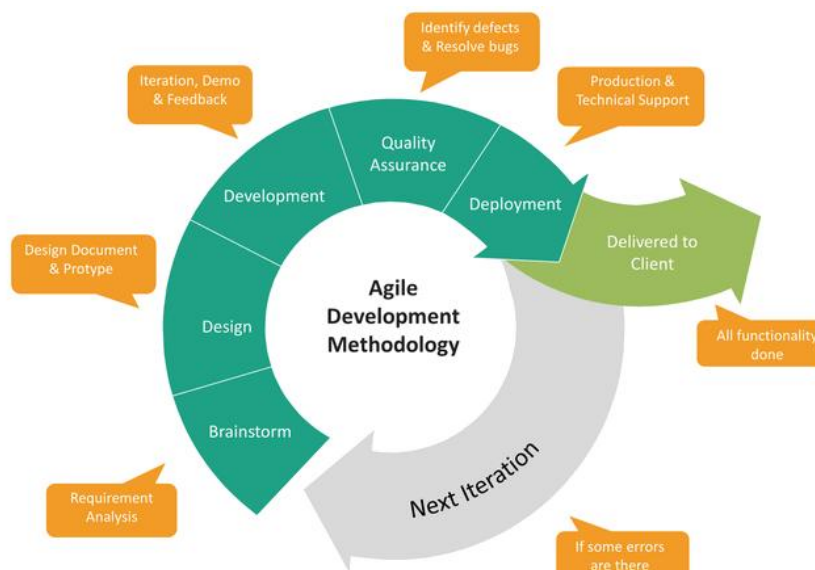
3.1 Development Model

The development of Customize Tee adopts the Agile Software Development Life Cycle (SDLC) model. This approach combines elements of iterative and incremental processes, emphasizing adaptability and customer satisfaction through the rapid delivery of a functional and evolving software product. The adoption of Agile SDLC for Customize Tee is geared towards creating a dynamic, responsive, and customer-centric development environment, ultimately delivering a successful and user-satisfying e-commerce platform.

Key Characteristics of Agile SDLC for "CUSTOMIZE TEE" :

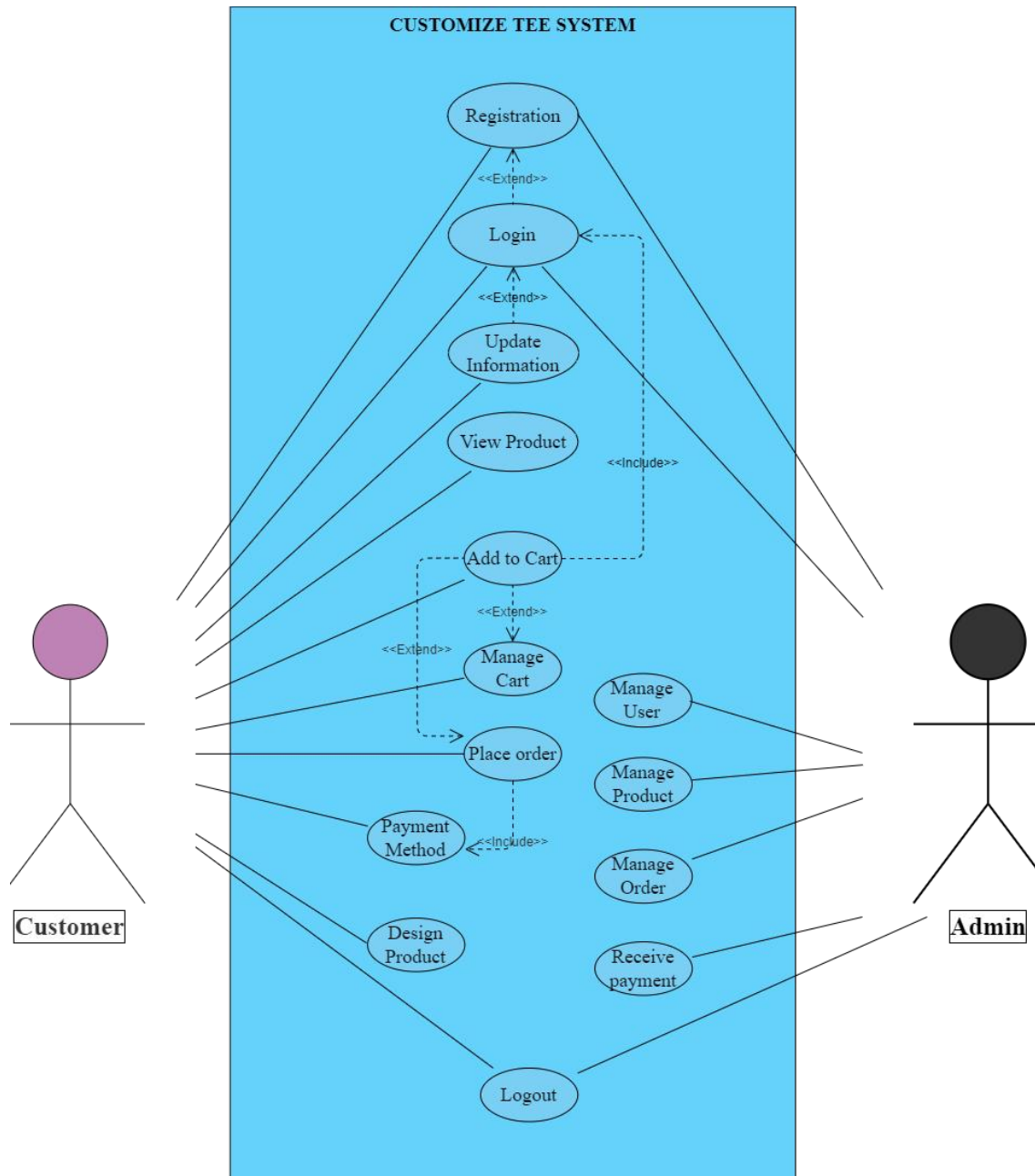
- Iterative and Incremental Development
- Rapid Adaptation to Changes
- Customer Centric Focus
- Collaborative and Cross-Functional Teams

Figure 3.1: Agile Development Methodology



3.2 Use Case Diagram

Figure 3.2: Use Case Diagram For "CUSTOMIZE TEE"



3.3 Use Case Descriptions

3.3.1 Registration

Table 3.3.1: Use Case Description For Registration

Use Case	Registration	
Goal	Customers can Register & create profile	
Precondition	Must fill valid information for registration	
Success End Condition	Customer successfully create profile, and can log into the system using same email and password	
Failed End Condition	Customer can't register An error message will be shown and a option will be provided.. "Incorrect email & password" and a "Try again"	
Primary Actors: Secondary Actors:	Customer	
Trigger	Customer requests to register and create profile	
Description / Main success scenario	Step	Action
	1	System will provide a register form
	2	Customer will fill up the necessary field with email and password correctly
	3	Hit register button
	4	System will verify user with given information provided by user
	5	Registration completed and profile created successfully
Alternative Flows	Step	Branching Action
	2A	Message "Error: please fill up the necessary field correctly and Try again"
	4A	Message "Error: please give correct Email and Password in required field"
	6A	Message "Login Error: please give correct Email and Password in required field and Try again"
Quality Requirements	Easy to interact and Fast responsive system	

3.3.2 Login

Table 3.3.2: Use Case Description For Login

Use Case	Login	
Goal	Customers can log into the system Admin can login to the system	
Precondition	They must fill correct information for login	
Success End Condition	Customer successfully log into the system & Admin successfully log into the system	
Failed End Condition	Customer and Admin can't log in into the system An error message will be shown and a option will be provided.. "Incorrect email & password" and a "Try again"	
Primary Actors: Secondary Actors:	Customer Admin	
Trigger	Customer requests to Login and Admin requests to Login	
Description / Main success scenario	Step	Action
	1	System will provide a Login form
	2	Customer will fill up the necessary field with email and password correctly and Admin will fill up the email and password field
	3	Hit Login button
	4	System will verify user with given information provided by user
	5	Login completed successfully
	6	Log into the system
Alternative Flows	Step	Branching Action
	2A	Message "Error: please fill up the necessary field correctly and Try again"
	4A	Message "Error: please give correct Email and Password in required field"
	6A	Message "Login Error: please give correct Email and Password in required field and Try again"
Quality Requirements	Easy to interact and Fast responsive system	

3.3.3 View Products

Table 3.3.3: Use Case Description For View Products

Use Case	View Product	
Goal	Everyone can view products and categories	
Precondition		
Success End Condition	Customers can view products then they can pick their desire product and place order	
Failed End Condition	An error message will be shown and a “Try again” option will be provided	
Primary Actors: Secondary Actors:	Customer	
Trigger	Customer requests to view all product and categories through the See more button and Categories section in the system	
Description / Main success scenario	Step	Action
	1	System will display products and categories
	2	Hit See more button
	3	System will shows all products
	4	Click Categories section
	5	The Categories will be displayed
Alternative Flows	Step	Branching Action
	3A	Message “Error: Something went wrong. Try again”
	5A	Message “Error: Something went wrong. Try again”
Quality Requirements	View all products and categories with optimized Images	

3.3.4 Add To Cart

Table 3.3.4 : Use Case Description For Add To Cart

Use Case	Add to cart	
Goal	Customer can choose and add desire product to the cart	
Precondition	Need to Login with a register profile	
Success End Condition	Products will be added to the cart successfully	
Failed End Condition	An error message will be shown and a Try again option will be provided Message “Error: Something went wrong. Try again”	
Primary Actors: Secondary Actors:	Customer	
Trigger	Customer requests to add their desire product to their cart	
Description / Main success scenario	Step	Action
	1	System will display the product
	2	Customer will choose the product
	3	Customer will hit the add to cart button
	4	Products added to the cart successfully
Alternative Flows	Step	Branching Action
	3A	Message “Error, Please Try again”
	4A	Message “Error: Something went wrong. Try again”
Quality Requirements	Product added to the cart very smoothly	

3.3.5 Manage Cart

Table 3.3.5: Use Case Description For Manage Cart

Use Case	Manage Cart	
Goal	Customers can add more products And remove products from their cart into the system	
Precondition	Customer must select their desire product	
Success End Condition	Customer successfully manage their cart	
Failed End Condition	An error message will be shown and a Try again option will be provided Message “Error: Something went wrong. Try again”	
Primary Actors: Secondary Actors:	Customer	
Trigger	Customer requests to add or remove product from their cart	
Description / Main success scenario	Step	Action
	1	System will display Cart
	2	Customer select the desire product
	3	Hit increase,decrease,remove,add button
	4	System will successfully increase,decrease,remove,add product as customer want
Alternative Flows	Step	Branching Action
	1A	Message “Error: please wait. Try again”
	4A	Message “Error: please select product”
Quality Requirements	Manage cart very smoothly	

3.3.6 Place Order

Table 3.3.6: Use Case Description For Place Order

Use Case	Place order	
Goal	Customers can place order from their cart	
Precondition	Customer must login into the system and select product	
Success End Condition	Order placed successfully	
Failed End Condition	Order not placed An error message will be shown and a Try again”option will be provided “Error: Something went wrong. Try again”	
Primary Actors: Secondary Actors:	Customer	
Trigger	Place order Button	
Description / Main success scenario	Step	Action
	1	System will display Cart
	2	Customer select the product from the cart
	3	Hit place order button
	4	Order placed successfully
Alternative Flows	Step	Branching Action
	2A	Message “Error: please select at least one product .Try again”
	3A	Message “Error: please try again”
	4a	Message “Error: Something went wrong. Try again”
Quality Requirements	Place order within seconds	

3.3.7 Design Custom T-shirt

Table 3.3.7: Use Case Description For Design Custom T-shirt

Use Case	Design Custom T-shirt	
Goal	Customers can Make their own design into the system	
Precondition	Customer must need to login into the system	
Success End Condition	Customer successfully Make Customize design	
Failed End Condition	Can't make design at this moment An error message will be shown and a Try again option will be provided “ Error: Can't make design at this moment .Try again”	
Primary Actors: Secondary Actors:	Customer	
Trigger	Customize t-shirt	
Description / Main success scenario	Step	Action
	1	System will display Customize t-shirt section
	2	Customer hit Create Design button
	3	Display t-shirt mockup and other tools
	4	Customer will select desire t-shirt color and upload designs in front or back side of the t-shirt and make placements
	5	Hit save button
	6	Customize t-shirt design successfully
Alternative Flows	Step	Branching Action
	2A	Message “Error: Please Try again”
	3A	Message “Error: Please reload”
	4A	Message “Error: Please wait .Try again”
	5A	Message “Error: Please wait .Try again”
Quality Requirements	Customizing page need user friendly ,tools,upload design and placement need to work smoothly	

3.3.8 Update Profile

Table 3.3.8: Use Case Description For Update Profile

Use Case	Update Profile	
Goal	Customer can Modify/Update their profile	
Precondition	Customer must need to login into the system	
Success End Condition	Customer update their profile successfully	
Failed End Condition	Customer can't update their profile	
Primary Actors: Secondary Actors:	Customer	
Trigger	Update Profile Button	
Description / Main success scenario	Step	Action
	1	System will display profile section
	2	Hit Update Profile Button
	3	Give update information in required field
	4	System will display update information in profile as user want
	5	Update Profile successfully
Alternative Flows	Step	Branching Action
	2A	Message "Error: Please Try again"
	3A	Message "Error: please try again"
	4A	Message "Error: please try again"
Quality Requirements	System will smoothly display update information in user profile	

3.3.9 Make Payment

Table 3.3.9: Use Case Description For Make Payment

Use Case	Payment method	
Goal	Customers can Make Payment for their order into the system	
Precondition	Customer must need to login into the system And place order	
Success End Condition	Customer successfully Make Payment for their orders	
Failed End Condition	Can't Make Payment at this moment An error message will be shown and a Try again option will be provided “ Error: Can't Make Payment at this moment .Try again”	
Primary Actors: Secondary Actors:	Customer	
Trigger	Make Payment Button	
Description / Main success scenario	Step	Action
	1	System will display Placed order section
	2	Customer select the Payment method
	3	Hit Make Payment button
	4	System will verify user information that given
	5	Make Payment successfully
	6	Get a notification and receipt of the order & payment
Alternative Flows	Step	Branching Action
	2A	Message “Error: Please select a payment method. Try again”
	4A	Message “Error: Please give correct information”
	5A	Message “Error: Payment Fail. Try again”
Quality Requirements	Payment must be Done instantly & Get a receipt of the order & payment as quickly as possible	

3.3.10 Manage Product

Table 3.3.10: Use Case Description For Manage Product

Use Case	Manage Product	
Goal	Admin can Add/Remove/Modify all Products	
Precondition	Admin must need to login into the system	
Success End Condition	Admin can Add/Remove/Modify Products successfully	
Failed End Condition	Can't Add/Remove/Modify Products	
Primary Actors: Secondary Actors:	Admin	
Trigger	Add/Remove/Modify Button	
Description / Main success scenario	Step	Action
	1	System will display all products
	2	Admin select the product for modify and remove,and upload product and its details for add product
	3	Hit Add/Remove/Modify Products button
	4	System will hide removed product or display modify products or Add new products as admin want
	5	Add/Remove/Modify Products successfully
Alternative Flows	Step	Branching Action
	2A	Message "Error: please select product. Try again"
	3A	Message "Error: please try again"
	4A	Message "Error: please try again"
Quality Requirements	System will smoothly hide removed product or display modify products or Add new products as admin want	

3.3.11 Manage Order

Table 3.3.11: Use Case Description For Manage Order

Use Case	Manage order	
Goal	Admin can approve & reject order using the system	
Precondition	Must need to login into the system and must have pending order	
Success End Condition	Manager can approve & reject order successfully	
Failed End Condition	Product stock out at this time	
Primary Actors: Secondary Actors:	Admin	
Trigger	Manage Order Button	
Description / Main success scenario	Step	Action
	1	System will display all pending orders
	2	Manager select the order
	3	Hit approve or reject order button
	4	System will notify customer for approve & reject order
	5	Manage (approve & reject) order successfully
Alternative Flows	Step	Branching Action
	3A	Message “Error: please select order. Try again”
	5A	Message “Error: please try again”
Quality Requirements	System need to give customer a notification after approval or decline	

3.3.12 Manage User

Table 3.3.12: Use Case Description For Manage User

Use Case	Manage User	
Goal	Admin can Add/Remove/Modify user profiles	
Precondition	Admin must need to login into the system	
Success End Condition	Admin can Add/Remove/Modify user profile successfully	
Failed End Condition	Admin can't Add/Remove/Modify user profiles	
Primary Actors: Secondary Actors:	Admin	
Trigger	Manage Button	
Description / Main success scenario	Step	Action
	1	System will display all User profiles
	2	Admin select the profile for modify or remove,or add profile or its details for modify
	3	Hit Add/Remove/Modify Products button
	4	System will removed profile or display modify profile or Add new profile as admin want
	5	Add/Remove/Modify profile successfully
Alternative Flows	Step	Branching Action
	2A	Message "Error: please select profile . Try again"
	3A	Message "Error: please try again"
	4A	Message "Error: please try again"
Quality Requirements	System will smoothly hide removed profile or display modify profile or Add new profile as admin want	

3.3.13 Receive Payment

Table 3.3.13: Use Case Description For Receive Payment

Use Case	Receive Payment	
Goal	Admin can see Receive payments for products and then he can update customers order status.	
Precondition	Admin must need log into the system	
Success End Condition	Admin successfully get order confirmation and get alert for receiving payment	
Failed End Condition	Admin didn't get order confirmation or alert	
Primary Actors: Secondary Actors:	Admin	
Trigger	Receive payment Section	
Description / Main success scenario	Step	Action
	1	System will display invoices of received payment with details
	2	Get a invoice of the order & payment
	3	Received Payment successfully
Alternative Flows	Step	Branching Action
	1a	Message : "Can't display.Please Try again"
	2a	Message: "No payment received yet"
Quality Requirements	Get Invoices of the order as soon as possible	

3.3.14 Logout

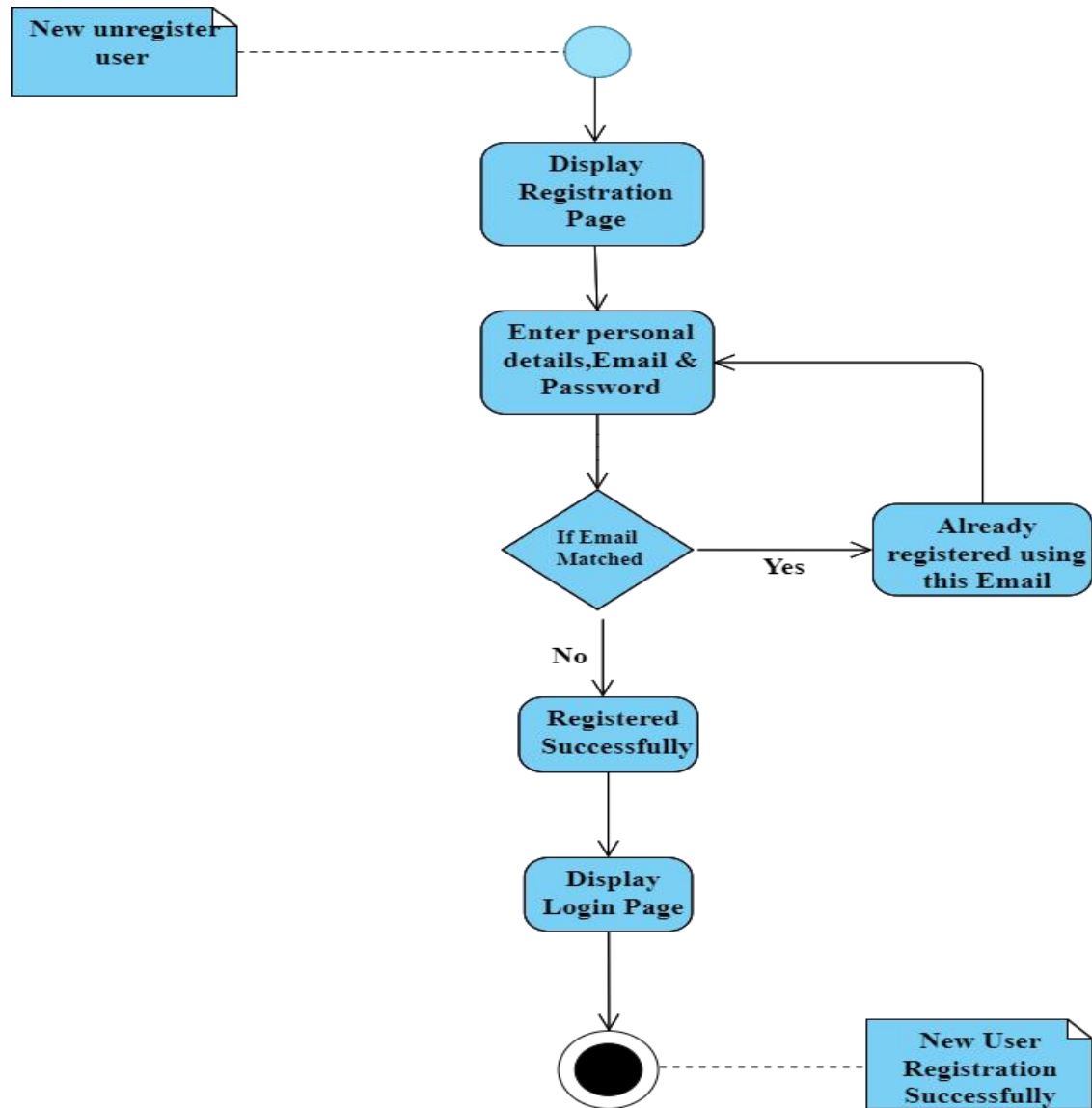
Table 3.3.14: Use Case Description For Logout

Use Case	Logout	
Goal	Customers and Admin both can Logout from the system	
Precondition	Must need to log in to the system	
Success End Condition	Customer and Admin successfully logout from the system	
Failed End Condition	Not logged in into the system	
Primary Actors: Secondary Actors:	Customer Admin	
Trigger	Customer And Admin requests to Logout from the system	
Description / Main success scenario	Step	Action
	1	System will provide a Logout option
	2	Hit Logout button
	3	System will log out the user
	4	Logout successfully
Alternative Flows	Step	Branching Action
	2A	Message “Error: Please Try again”
Quality Requirements	Fast responsive system	

3.4 Activity Diagram

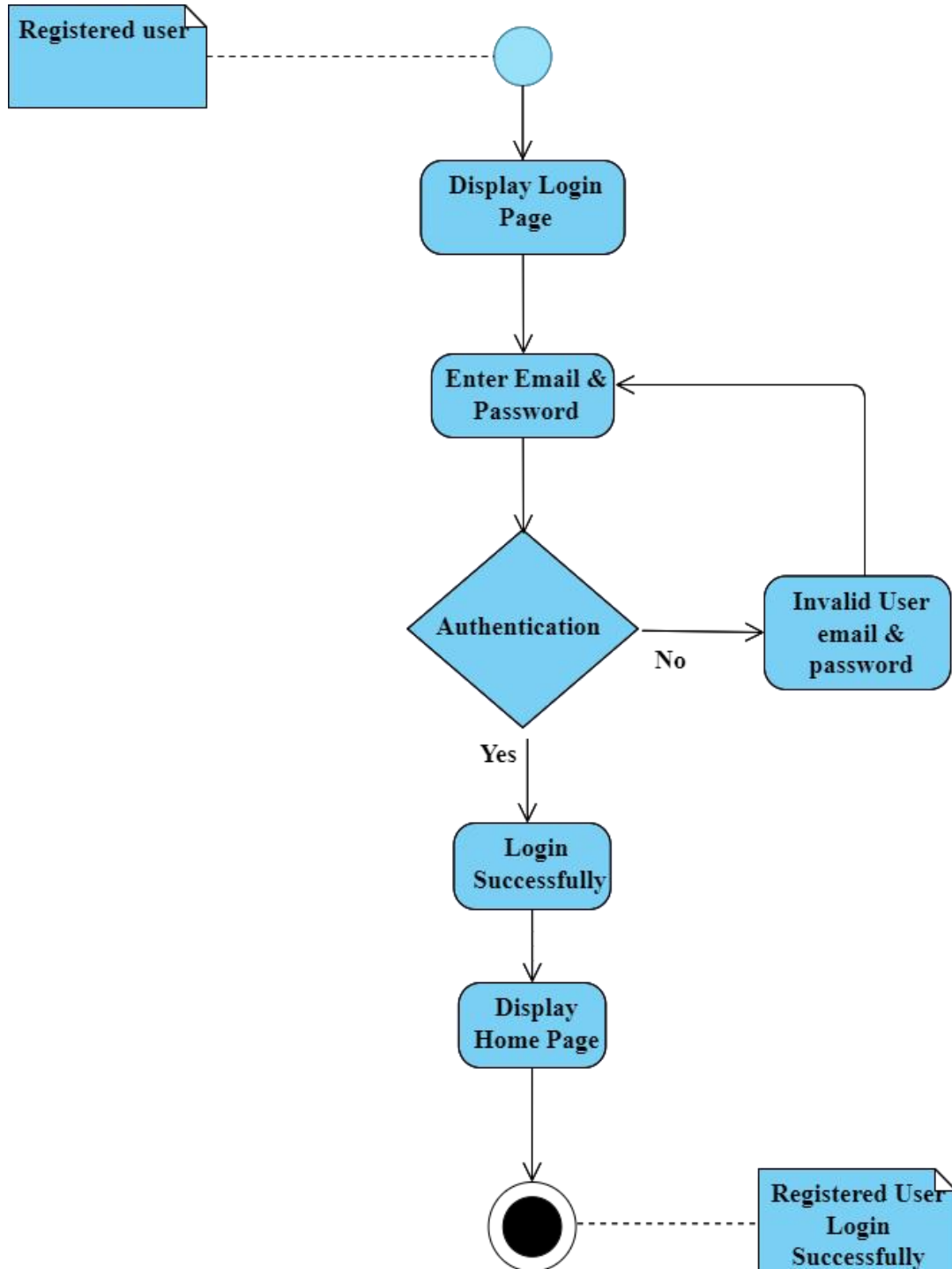
3.4.1 Registration

Figure 3.4.1: Activity Diagram For Registration



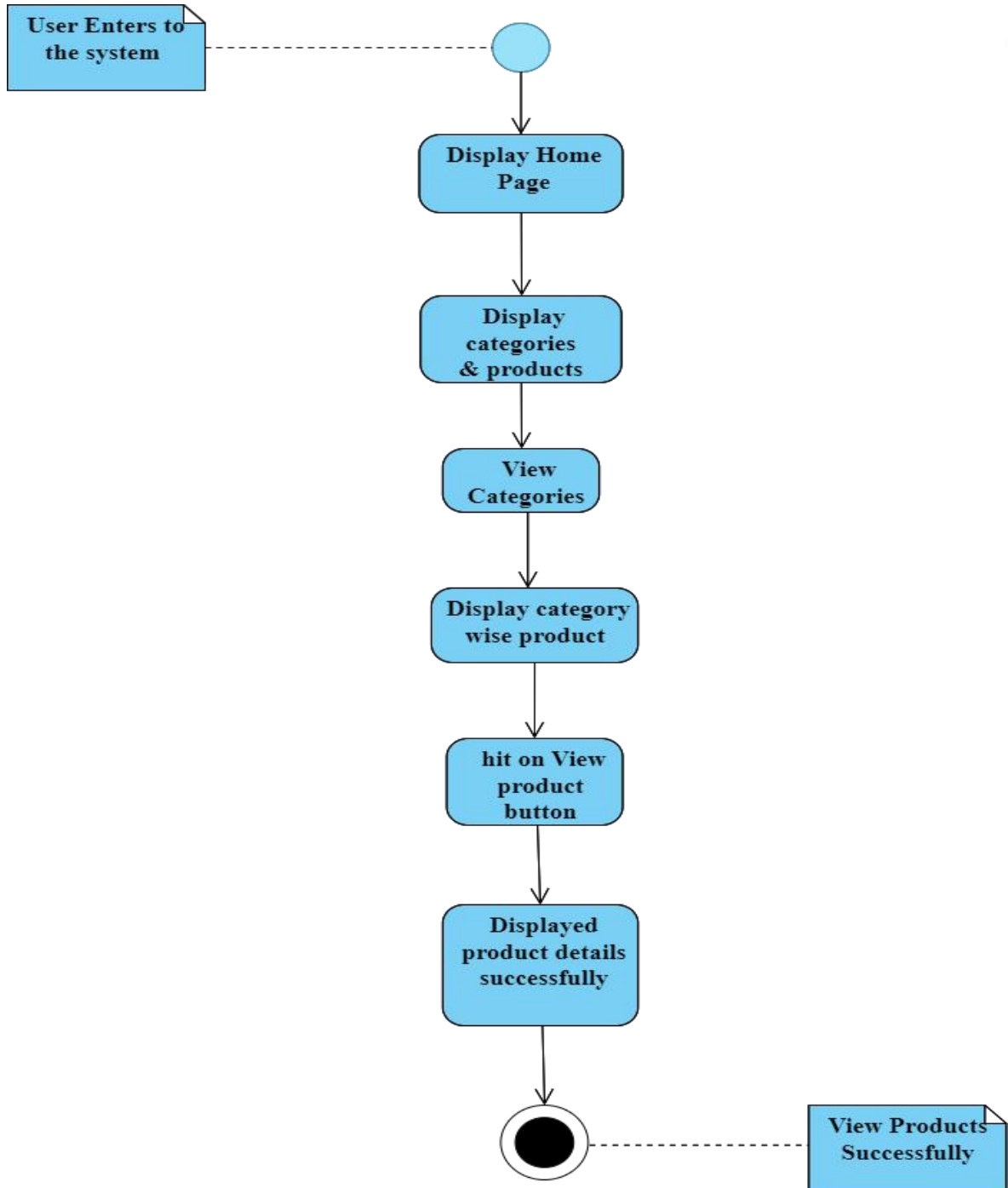
3.4.2 Login

Figure 3.4.2: Activity Diagram For Login



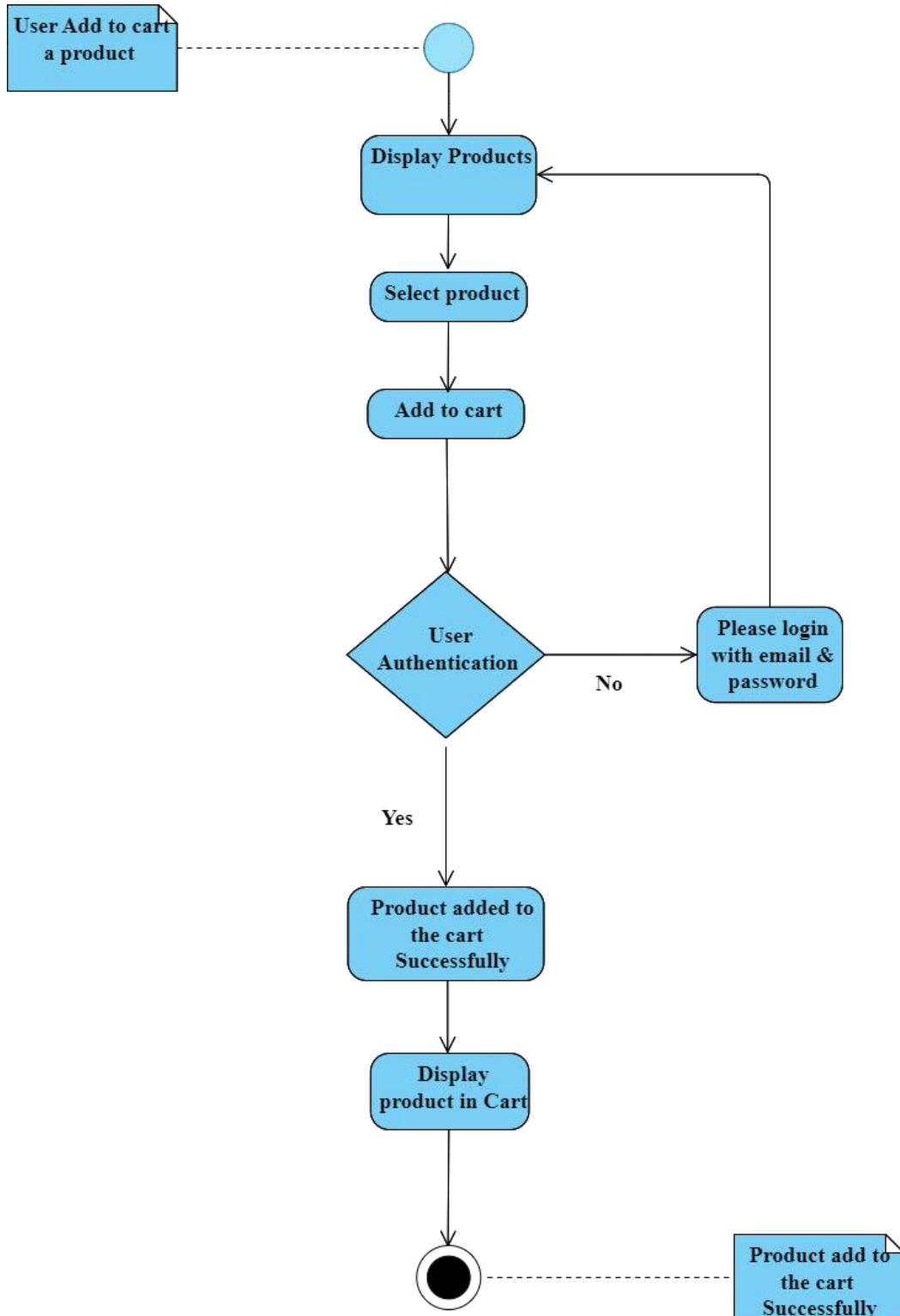
3.4.3 View Products

Figure 3.4.3: Activity Diagram For View Products



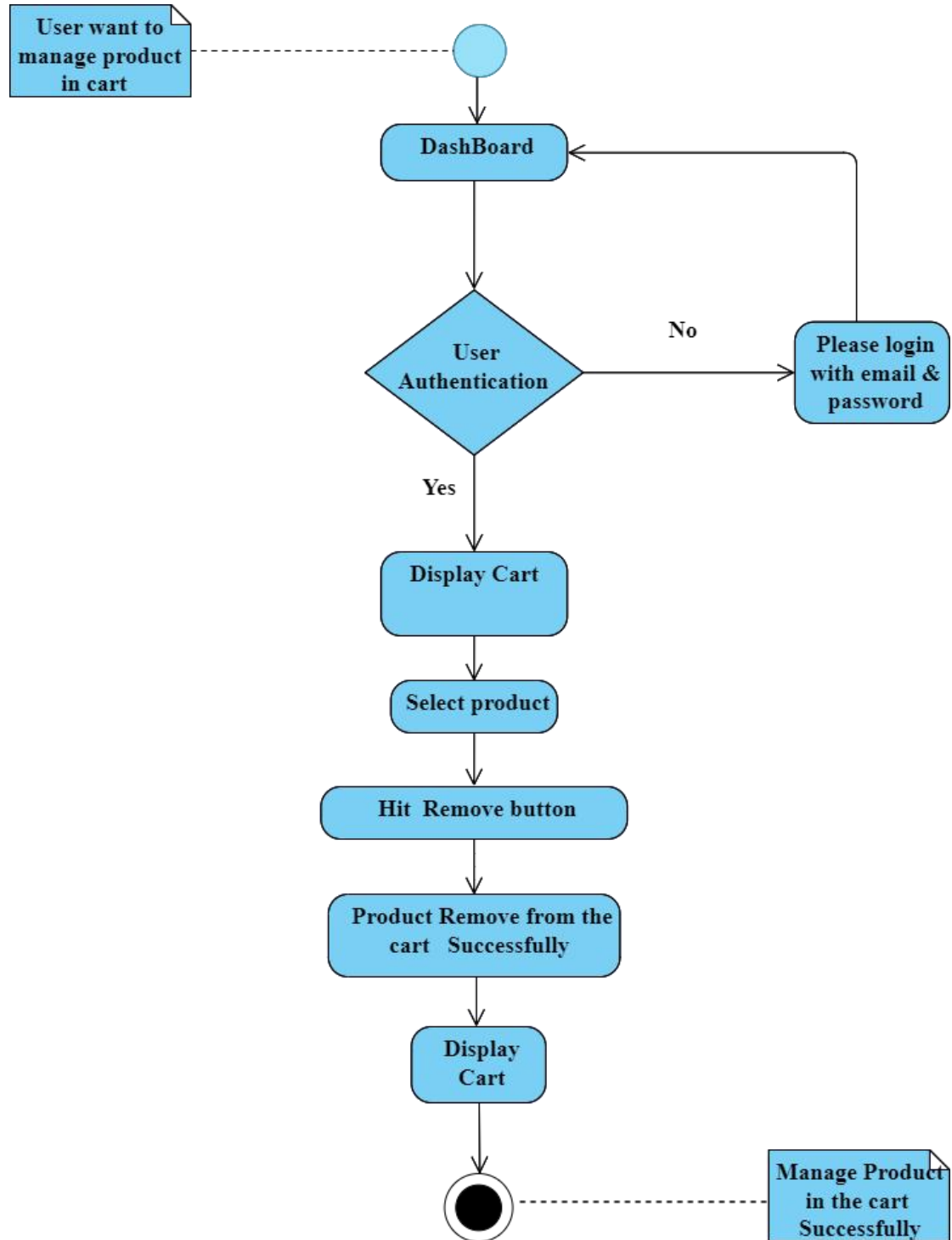
3.4.4 Add To Cart

Figure 3.4.4: Activity Diagram For Add To Cart



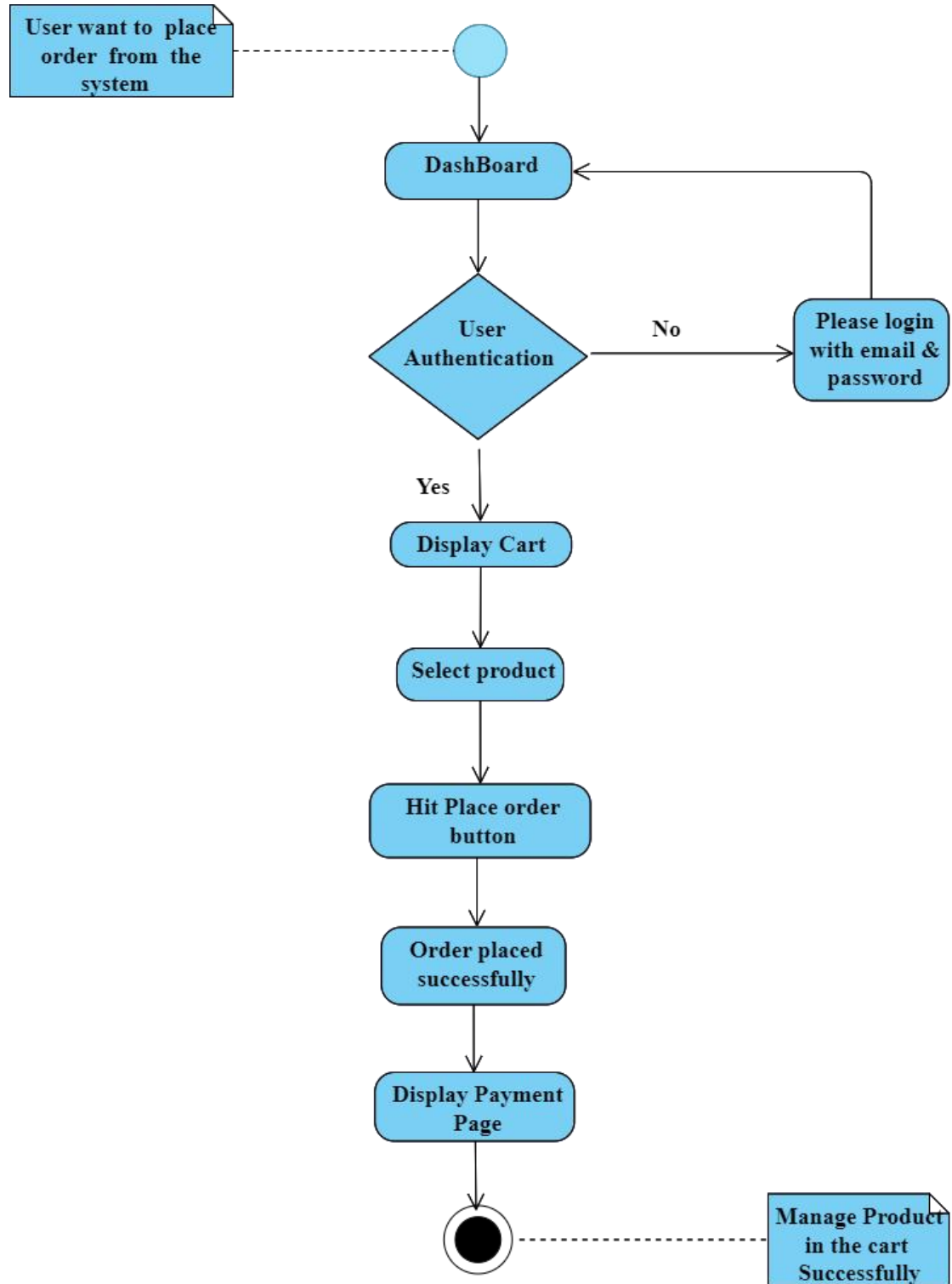
3.4.5 Manage Cart

Figure 3.4.5: Activity Diagram For Manage Cart



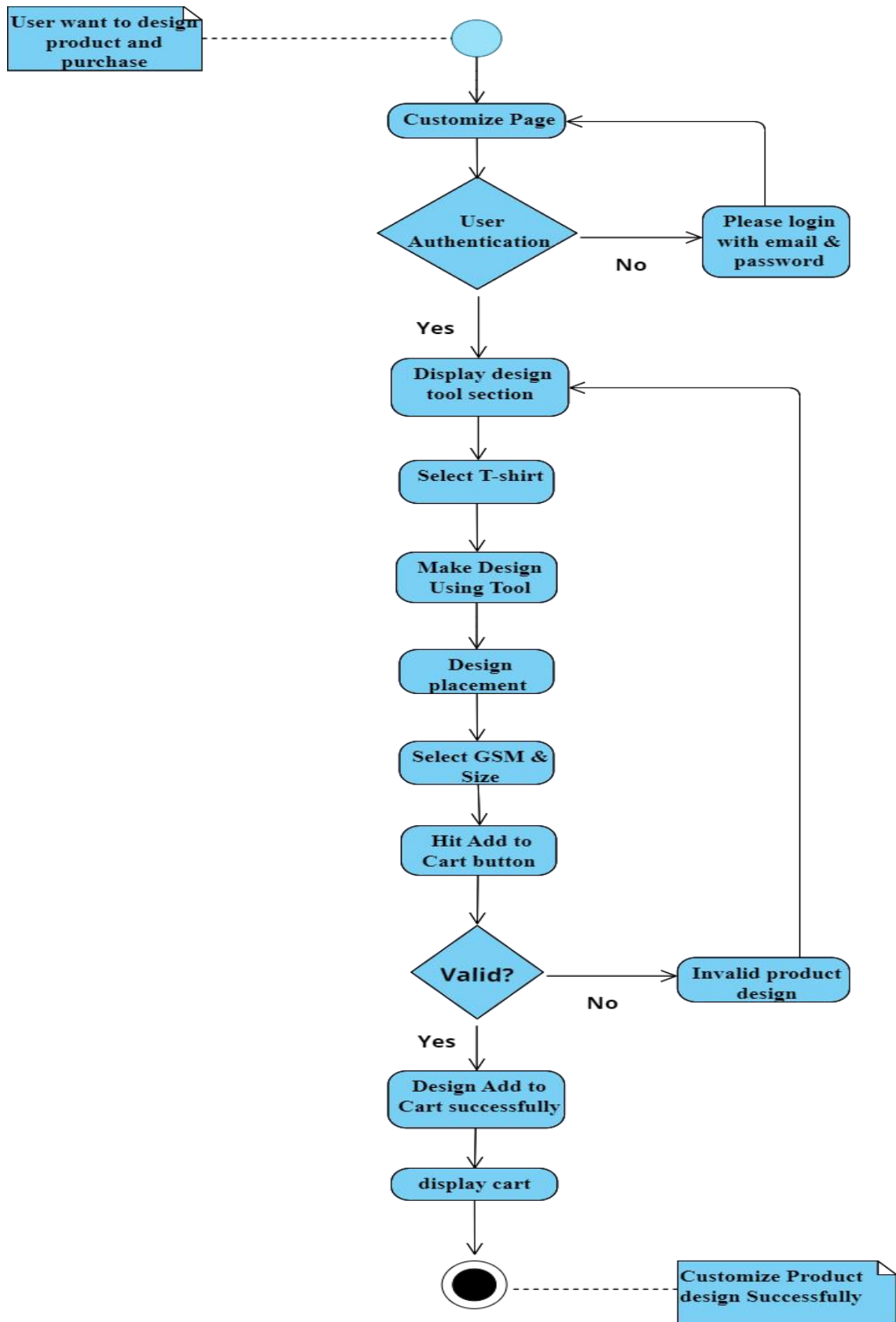
3.4.6 Place Order

Figure 3.4.6: Activity Diagram For Place Order



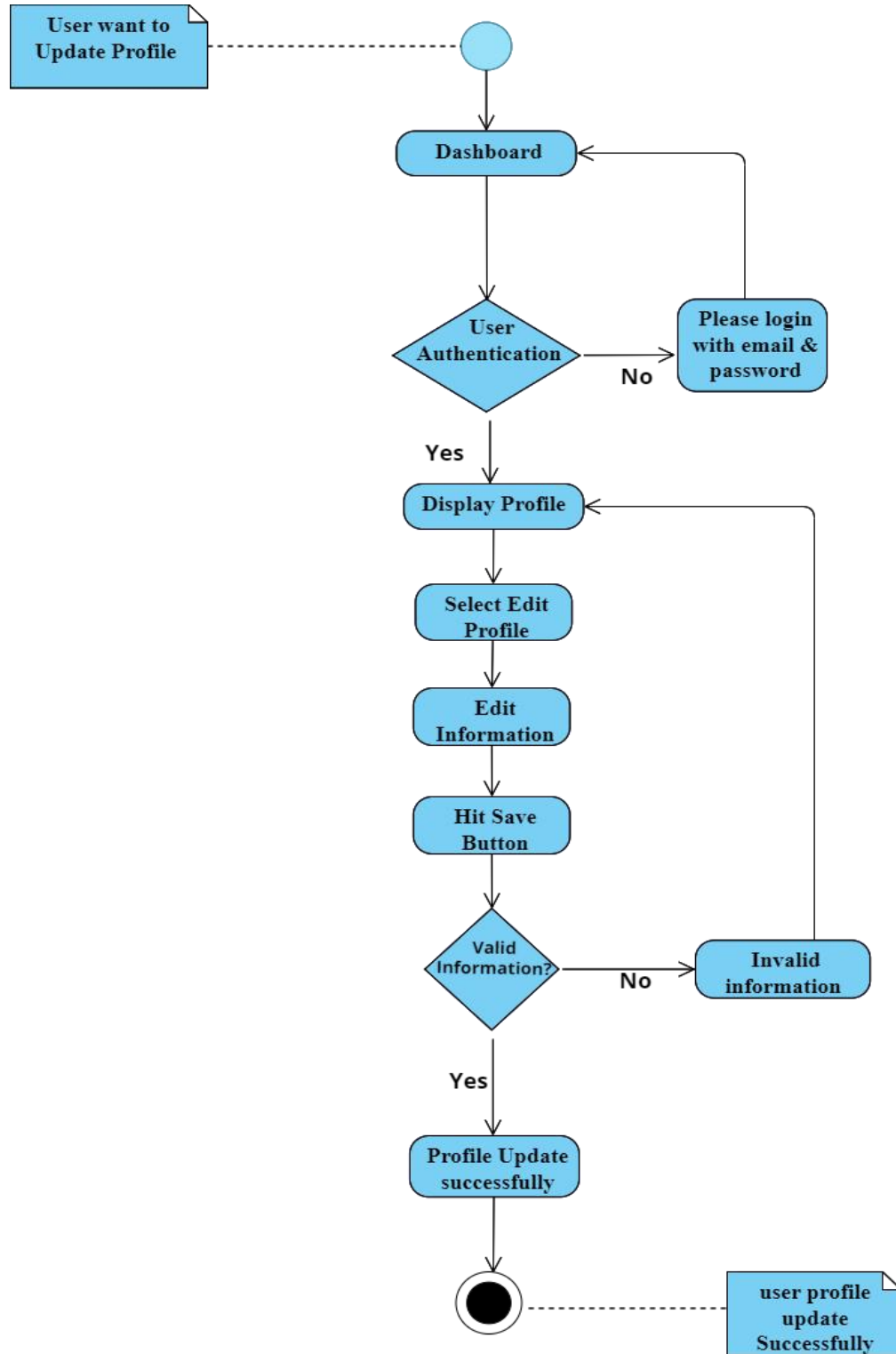
3.4.7 Design Custom T-shirt

Figure 3.4.7: Activity Diagram For Design Custom T-shirt



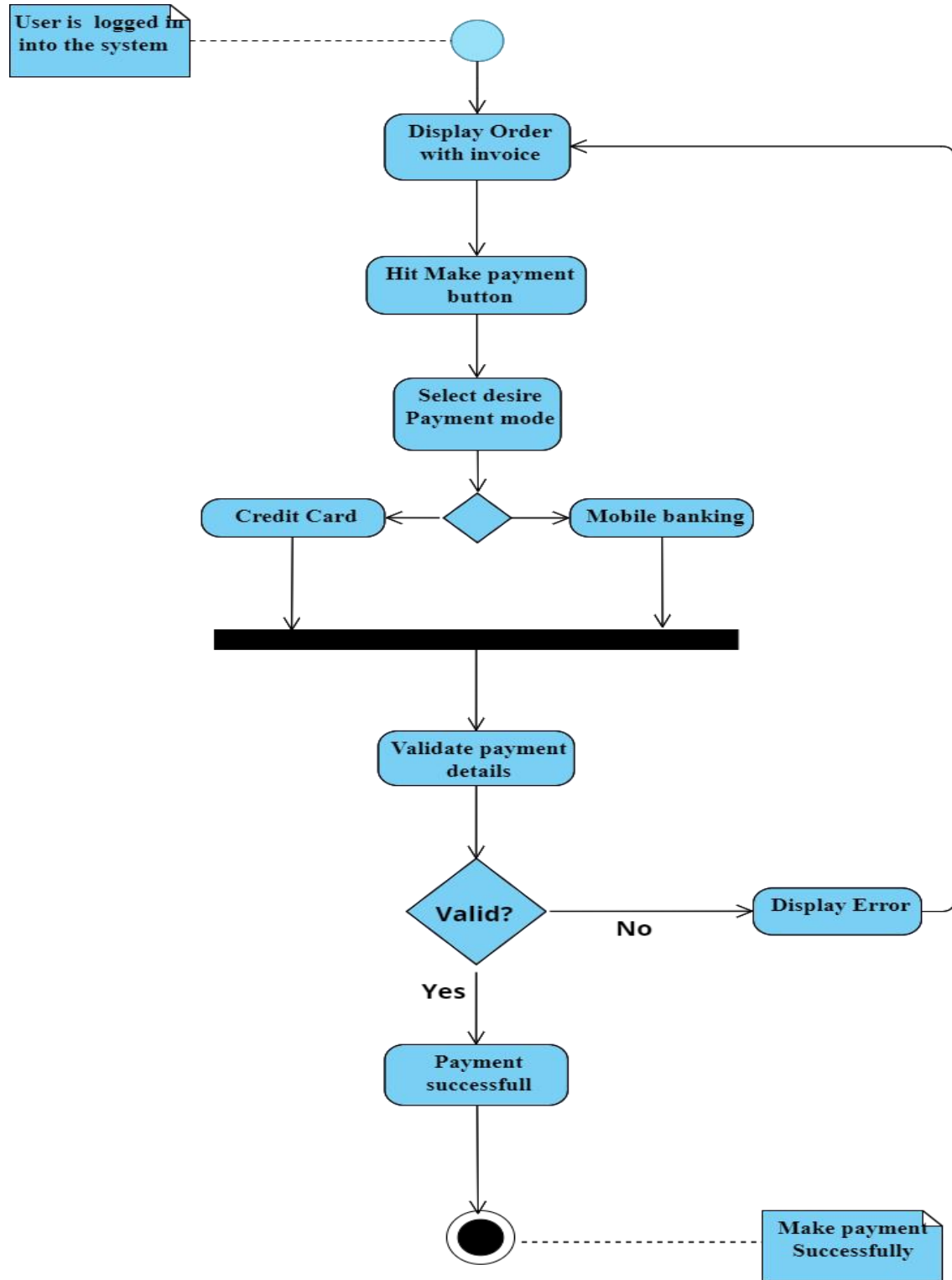
3.4.8 Update Profile

Figure 3.4.8: Activity Diagram For Update Profile



3.4.9 Make Payment

Figure 3.4.9: Activity Diagram For Make Payment



3.4.10 Manage Product

Figure 3.4.10: Activity Diagram For Add Product

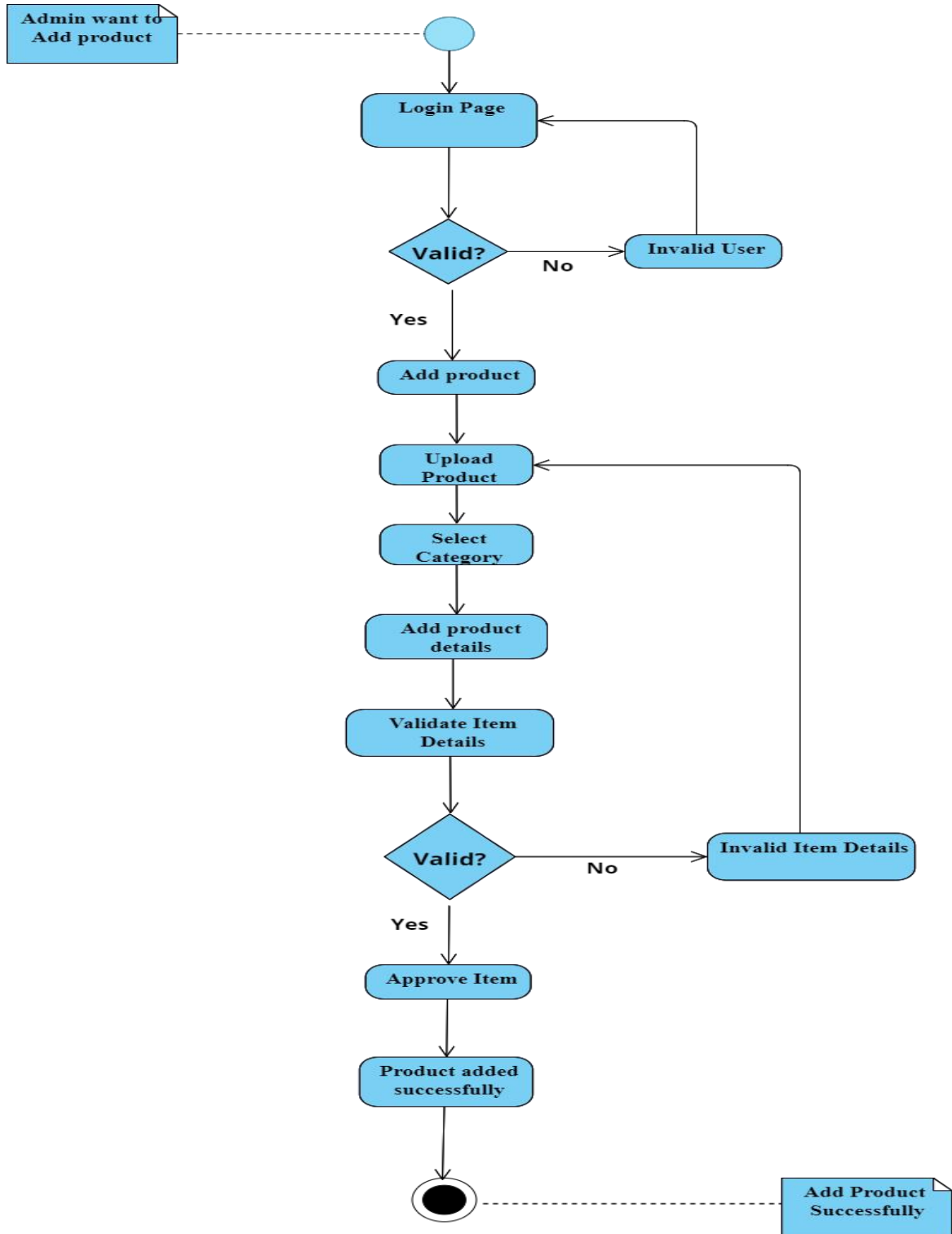
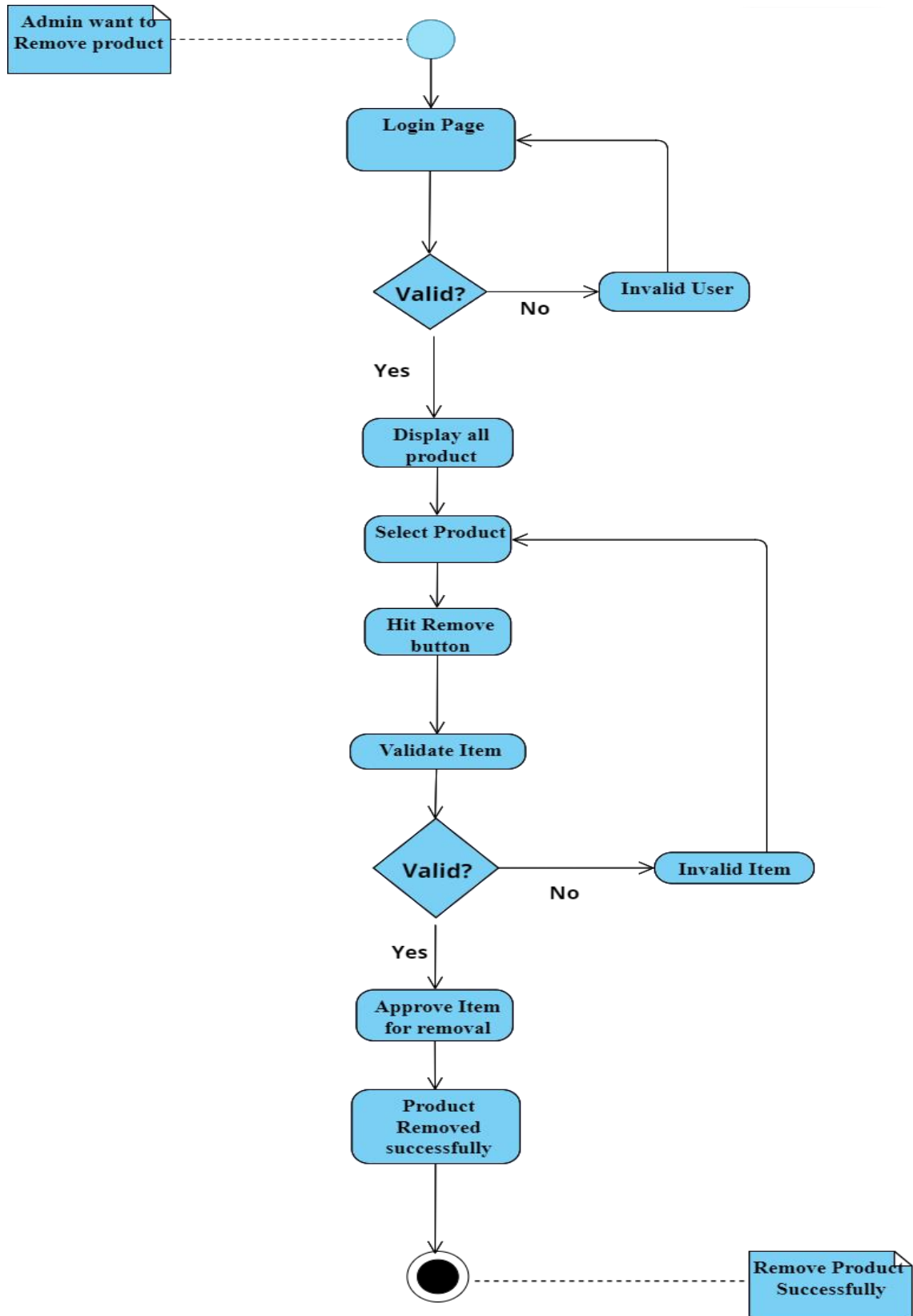
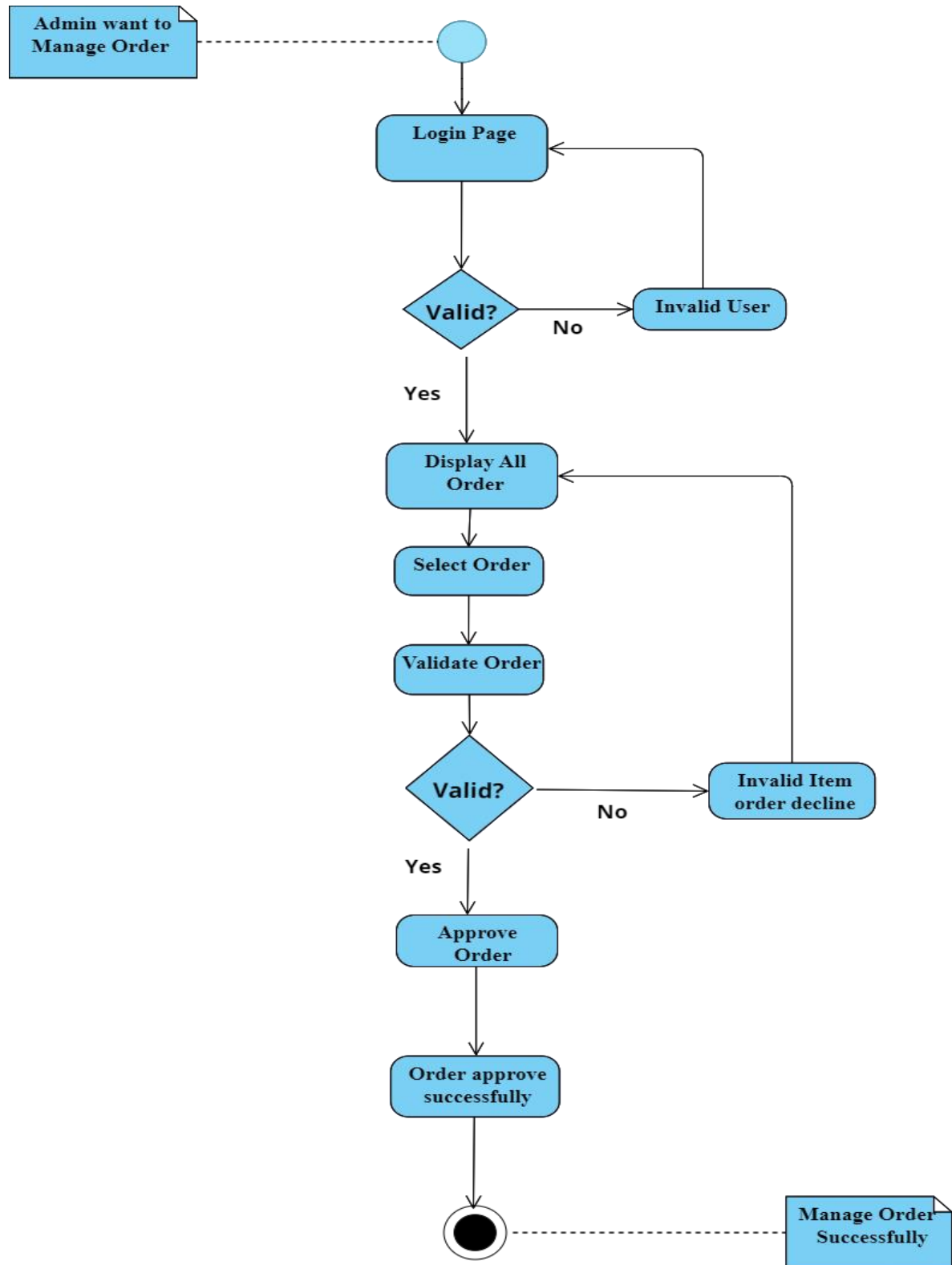


Figure 3.4.11: Activity Diagram For Remove Product



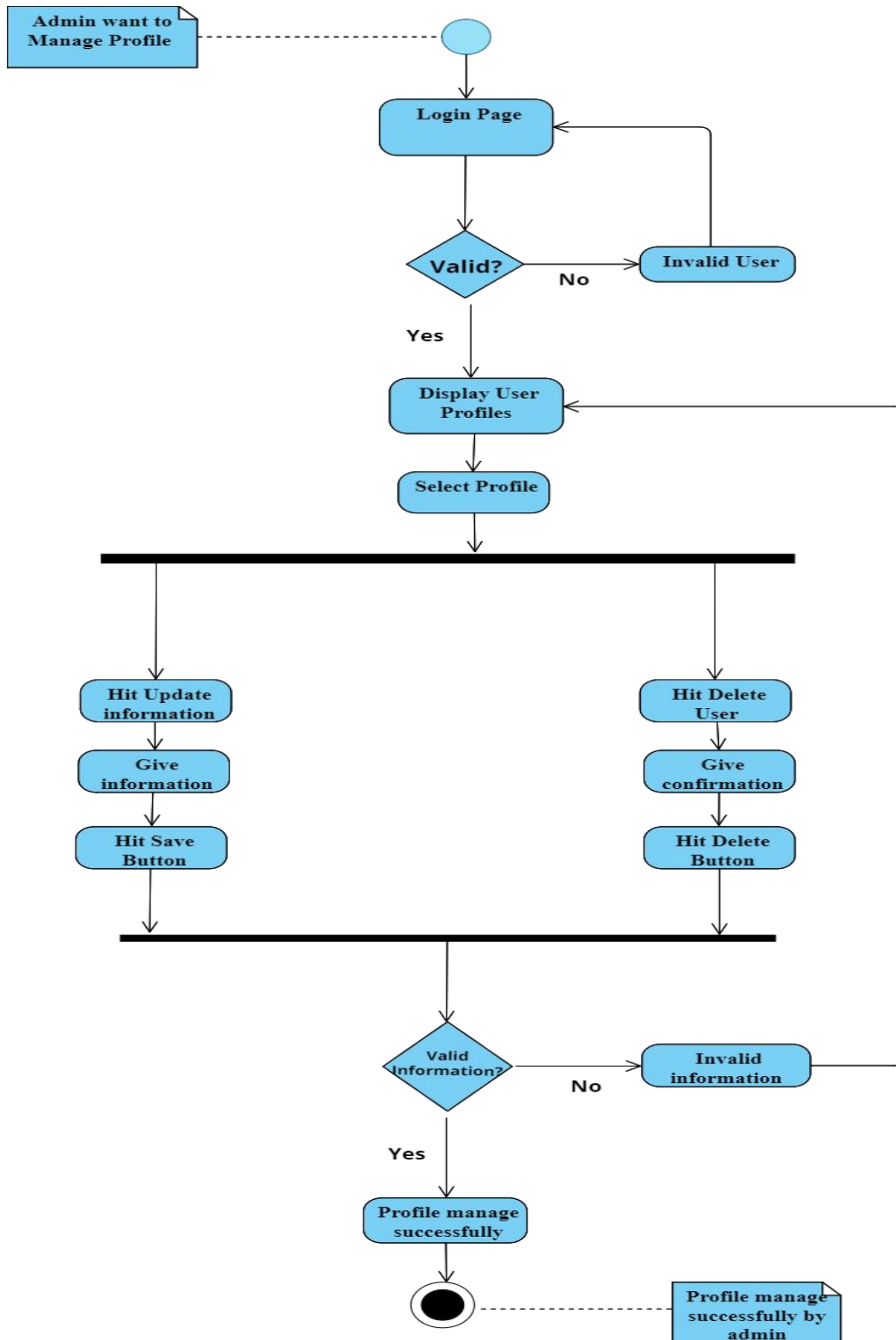
3.4.11 Manage Order

Figure 3.4.12: Activity Diagram For Manage Order



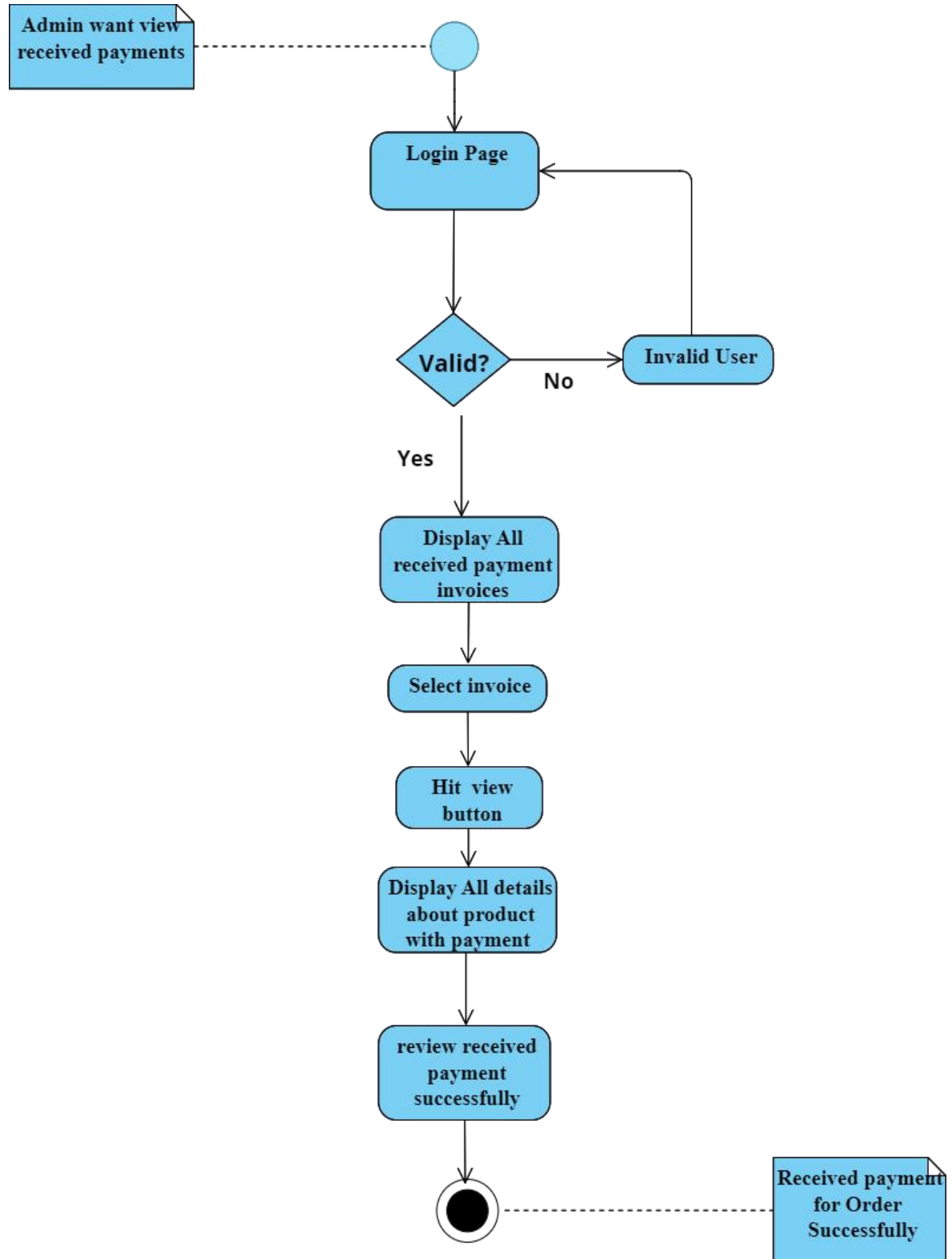
3.4.12 Manage User

Figure 3.4.13: Activity Diagram For Manage User



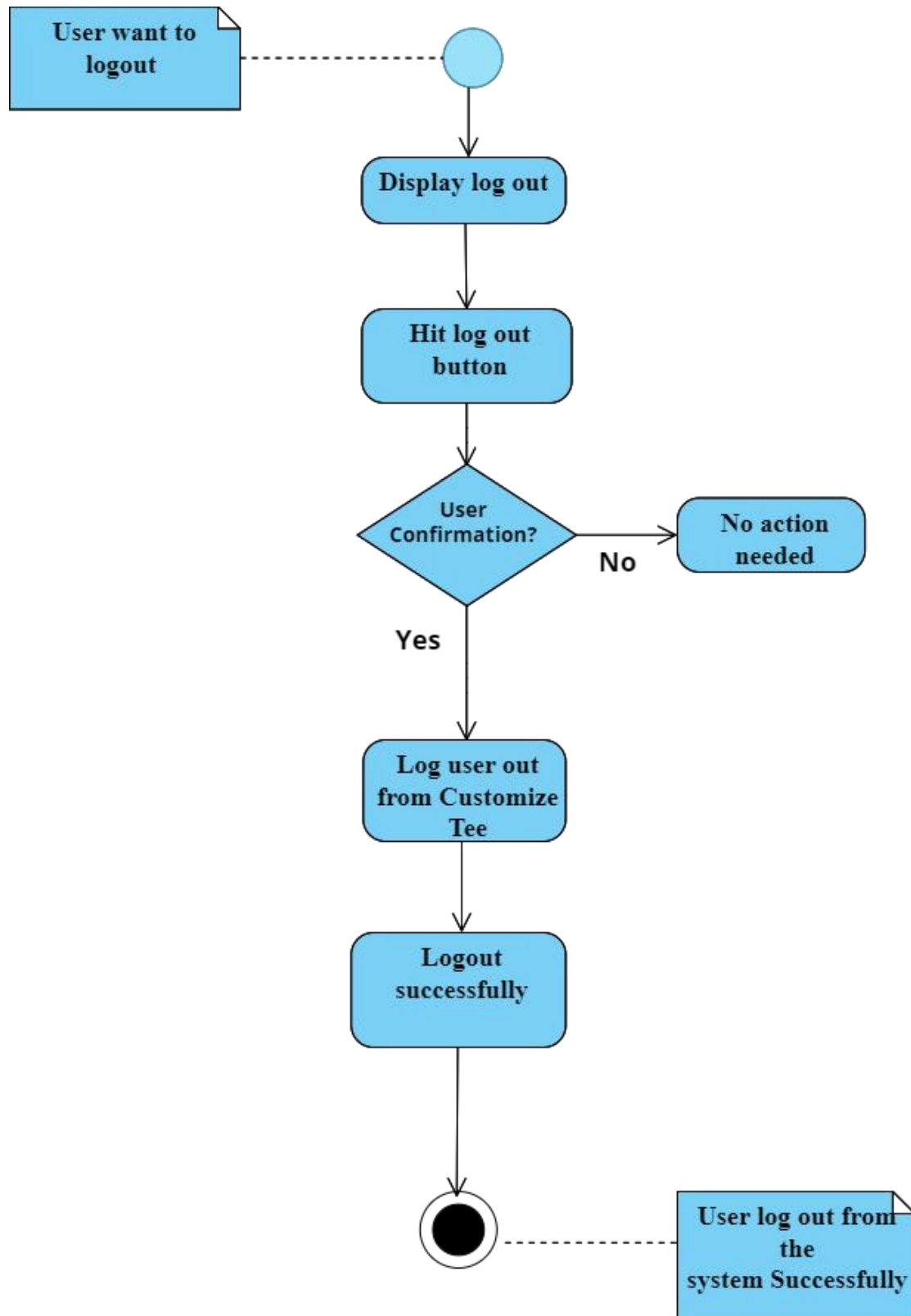
3.4.13 Receive Payment

Figure 3.4.14: Activity Diagram For Receive Payment



3.4.14 Logout

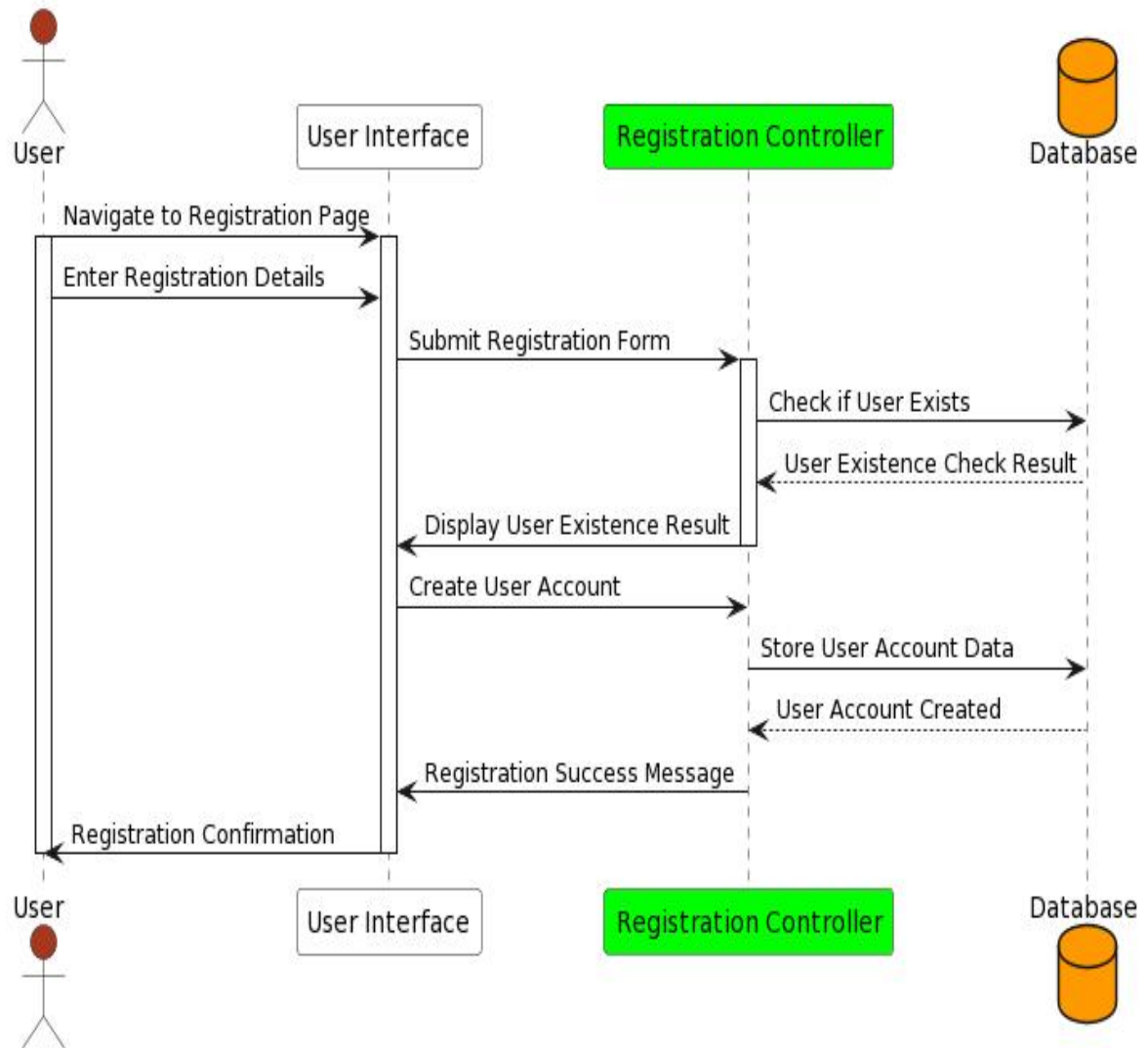
Figure 3.4.15: Activity Diagram For Logout



3.5 Sequence Diagram

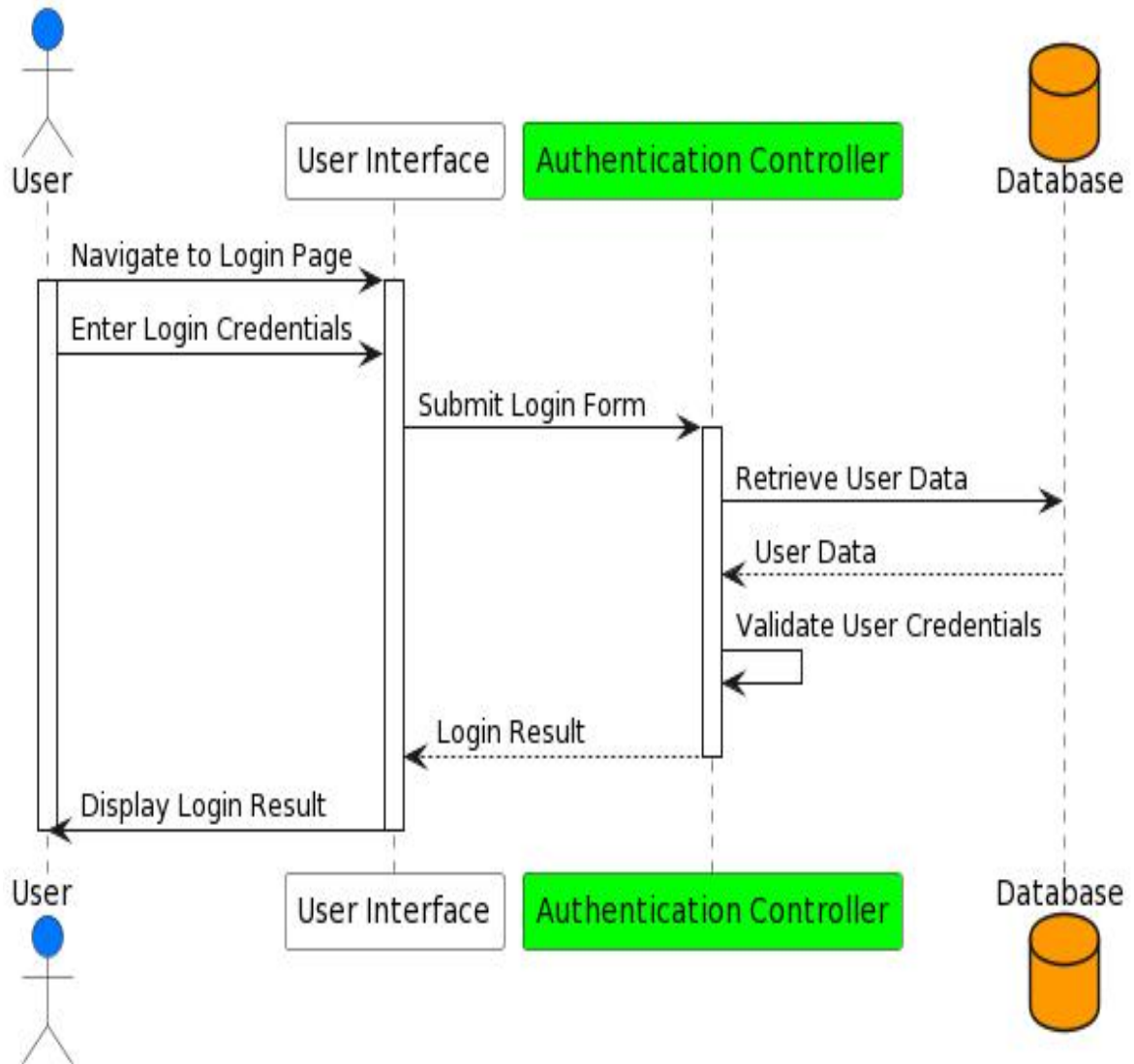
3.5.1 Registration

Figure 3.5.1: Sequence Diagram For Registration



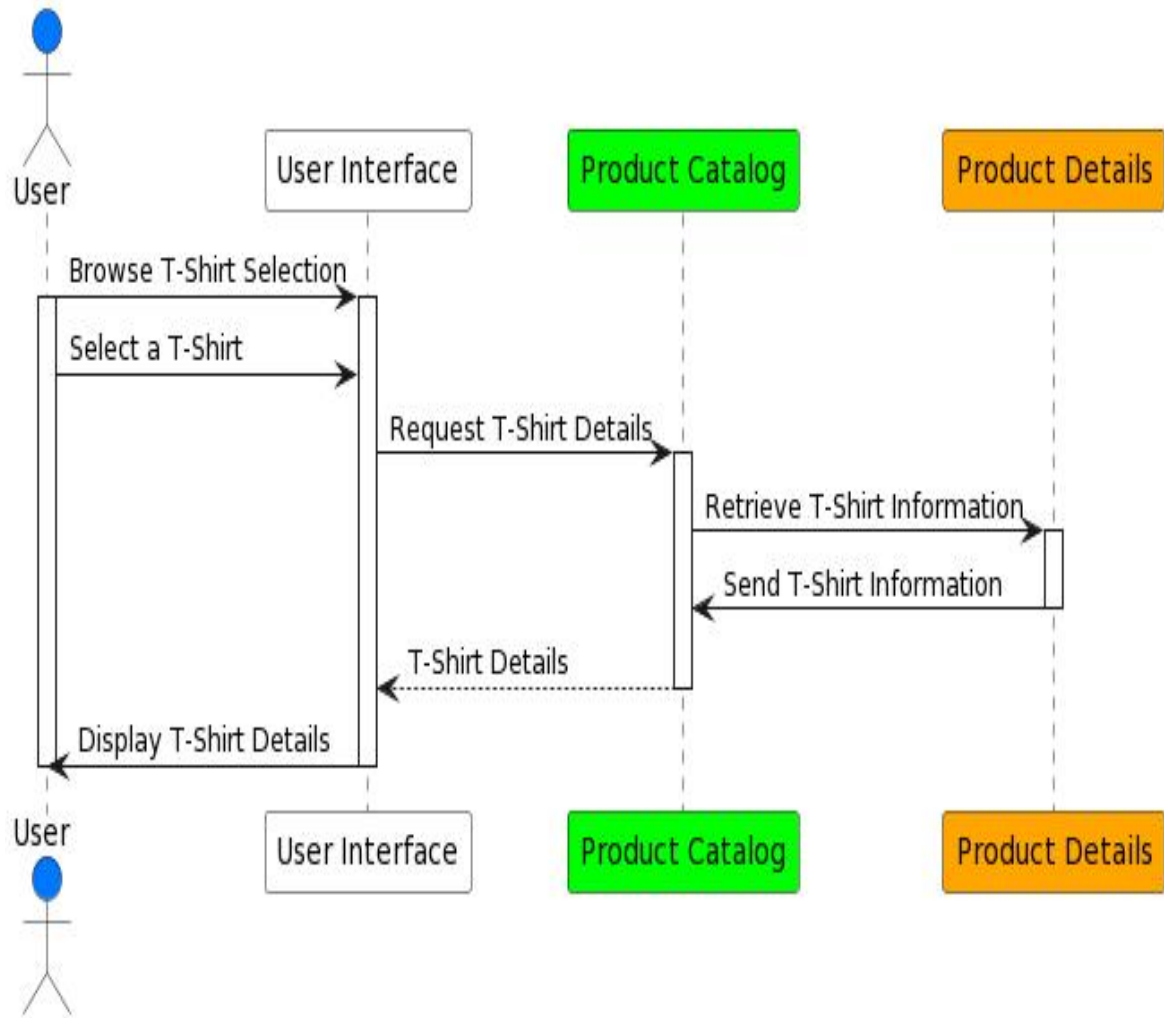
3.5.2 Login

Figure 3.5.2: Sequence Diagram For Login



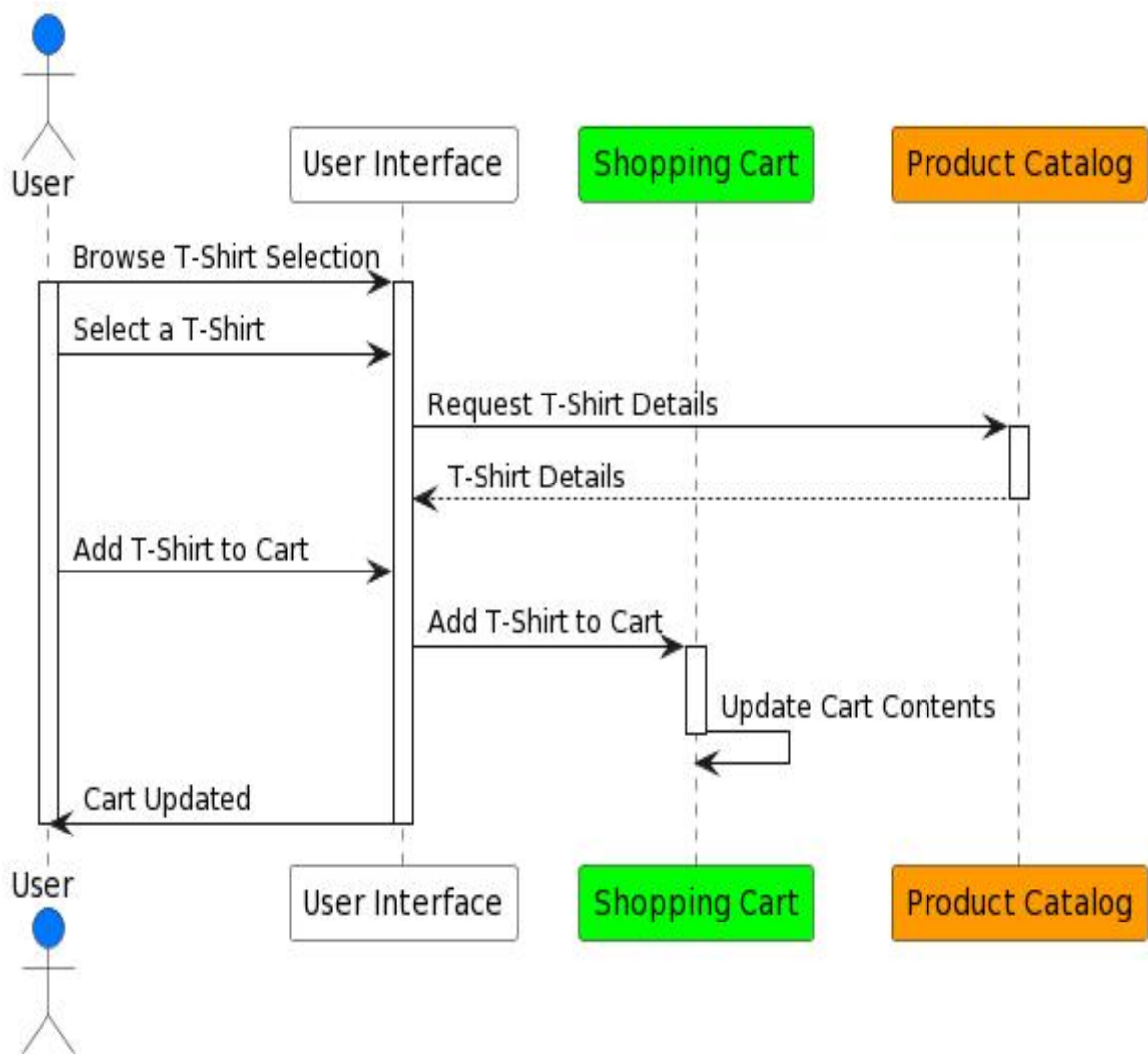
3.5.3 View Products

Figure 3.5.3: Sequence Diagram For View Products



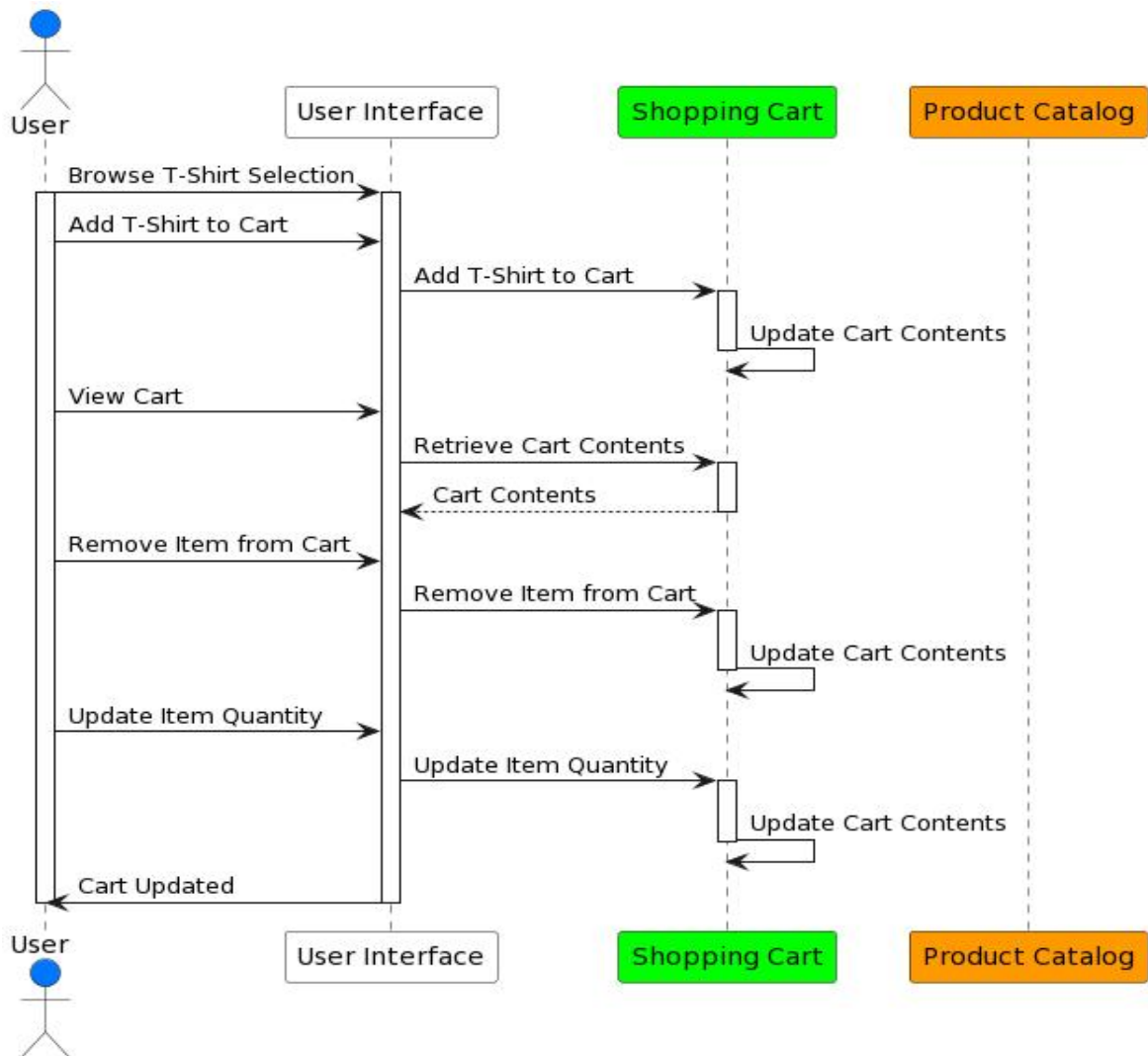
3.5.4 Add To Cart

Figure 3.5.4: Sequence Diagram For Add To Cart



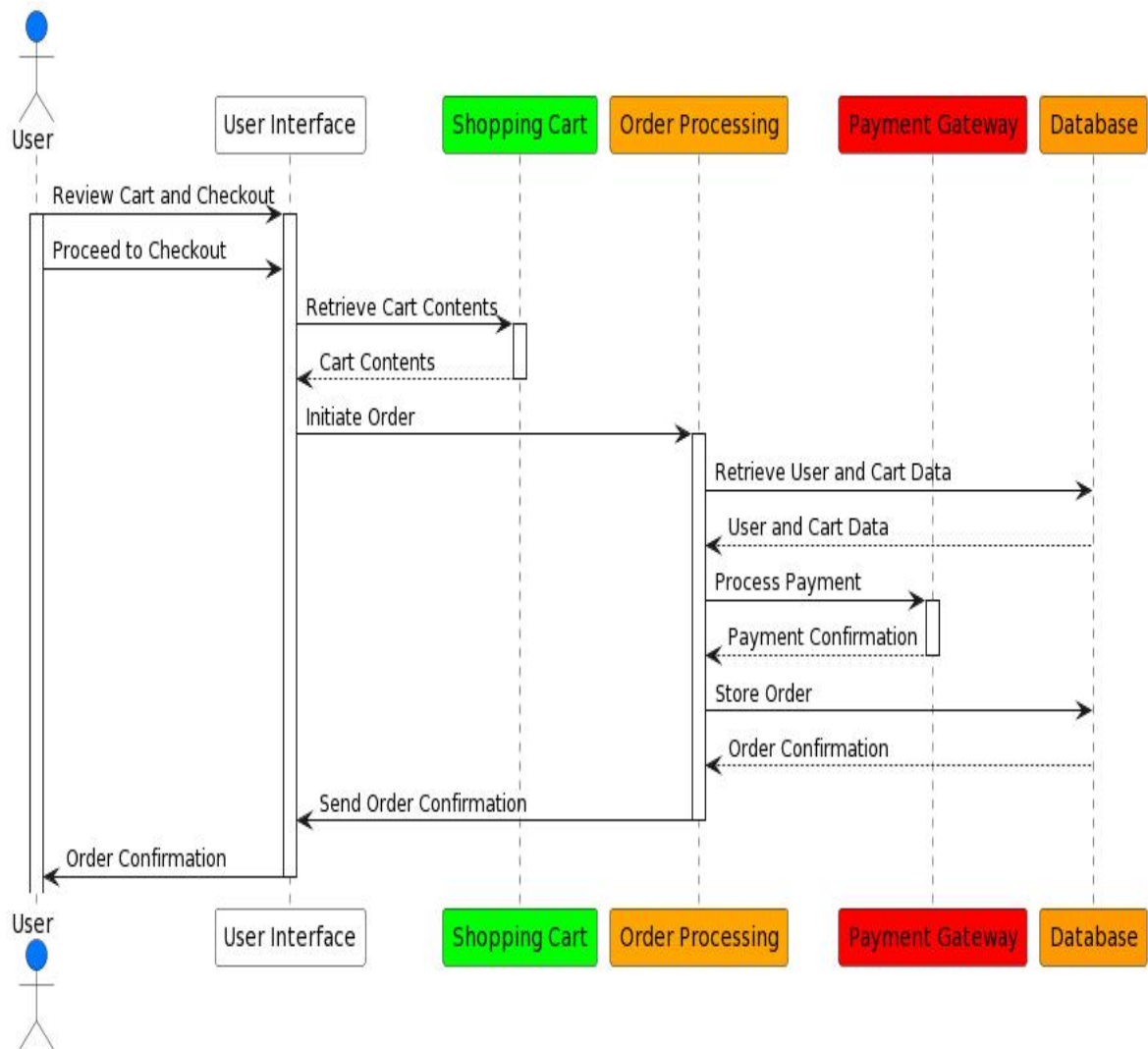
3.5.5 Manage Cart

Figure 3.5.5: Sequence Diagram For Manage Cart



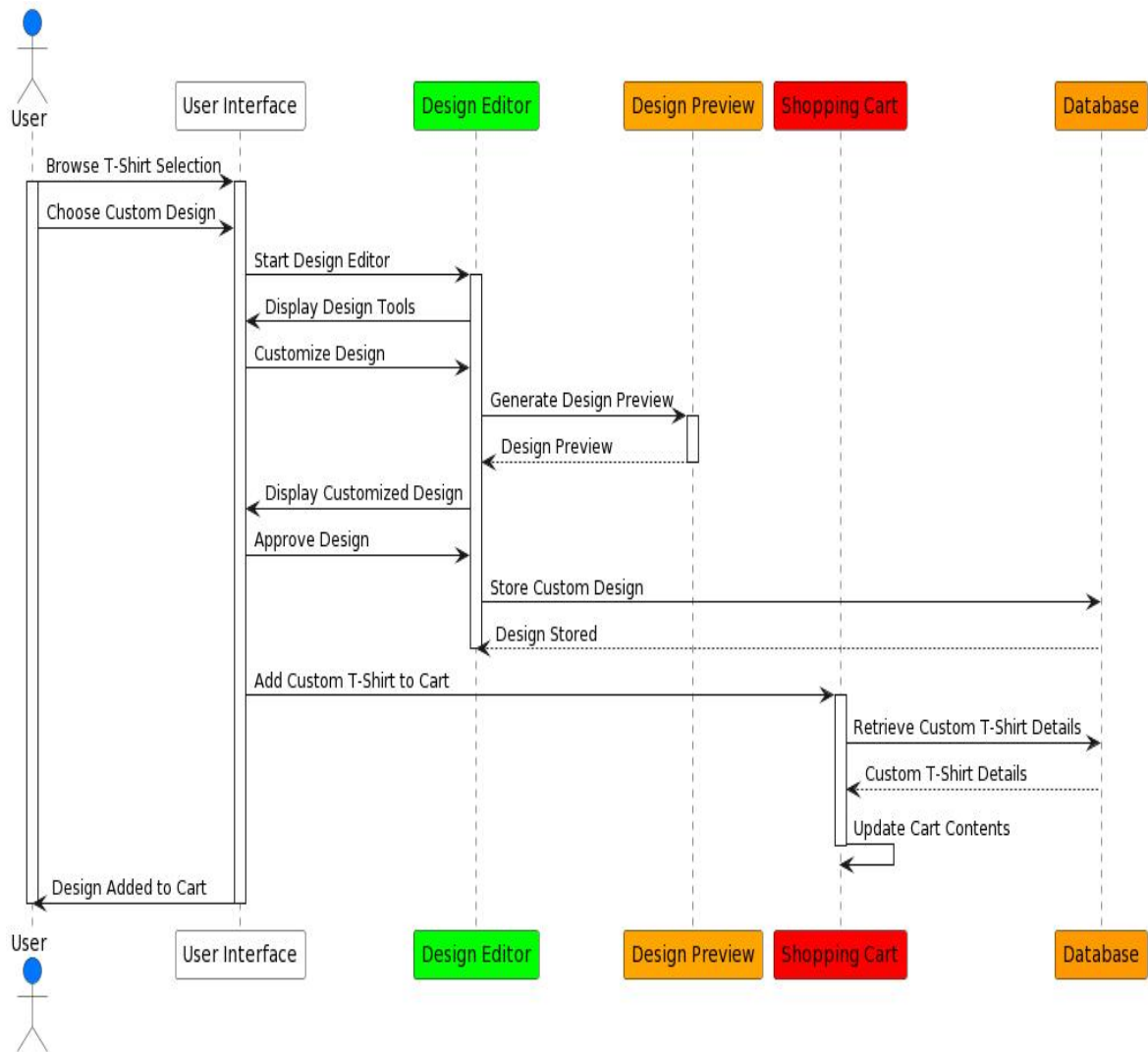
3.5.6 Place Order

Figure 3.5.6: Sequence Diagram For Place Order



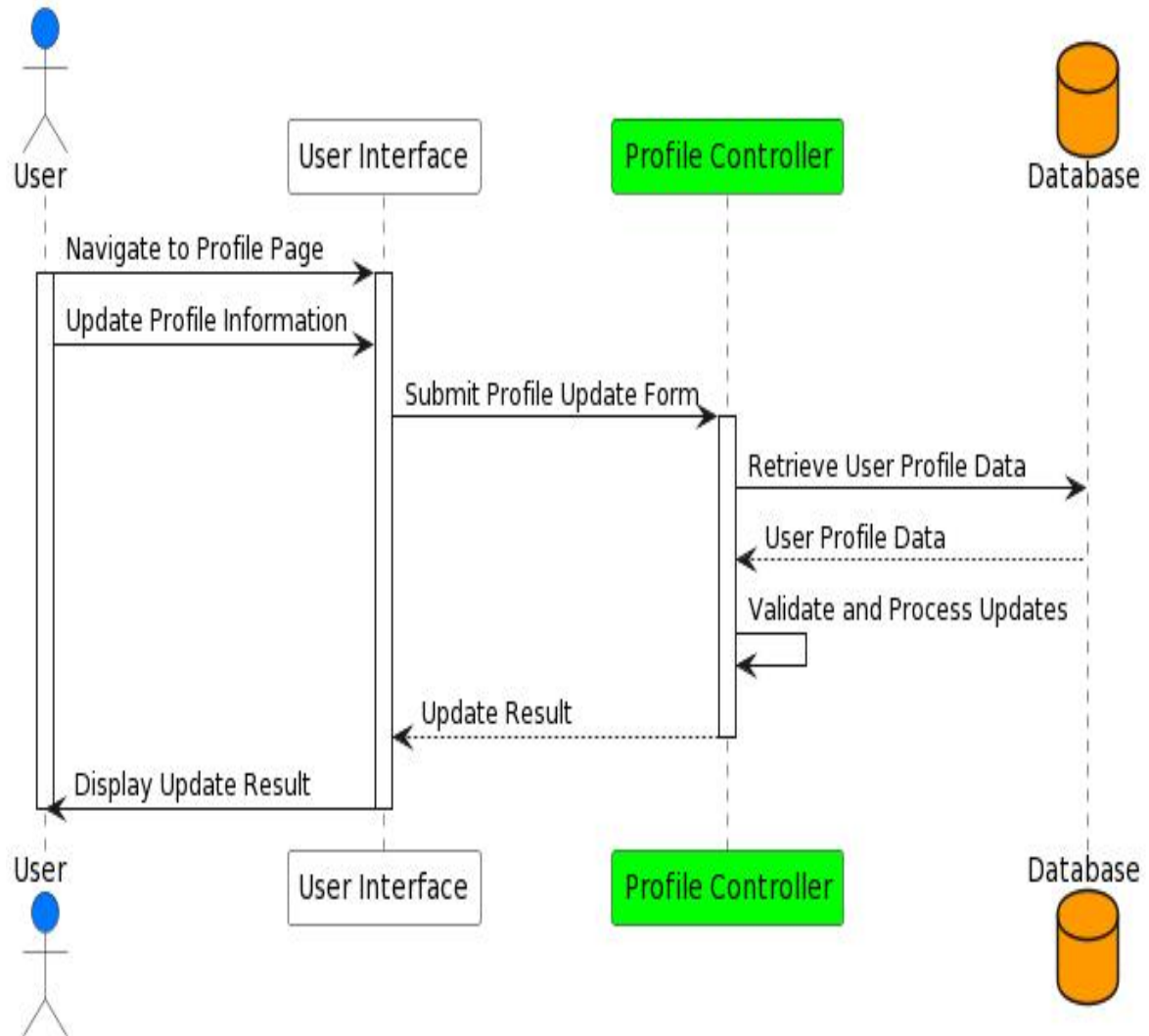
3.5.7 Design Custom T-shirt

Figure 3.5.7: Sequence Diagram For Design Custom T-shirt



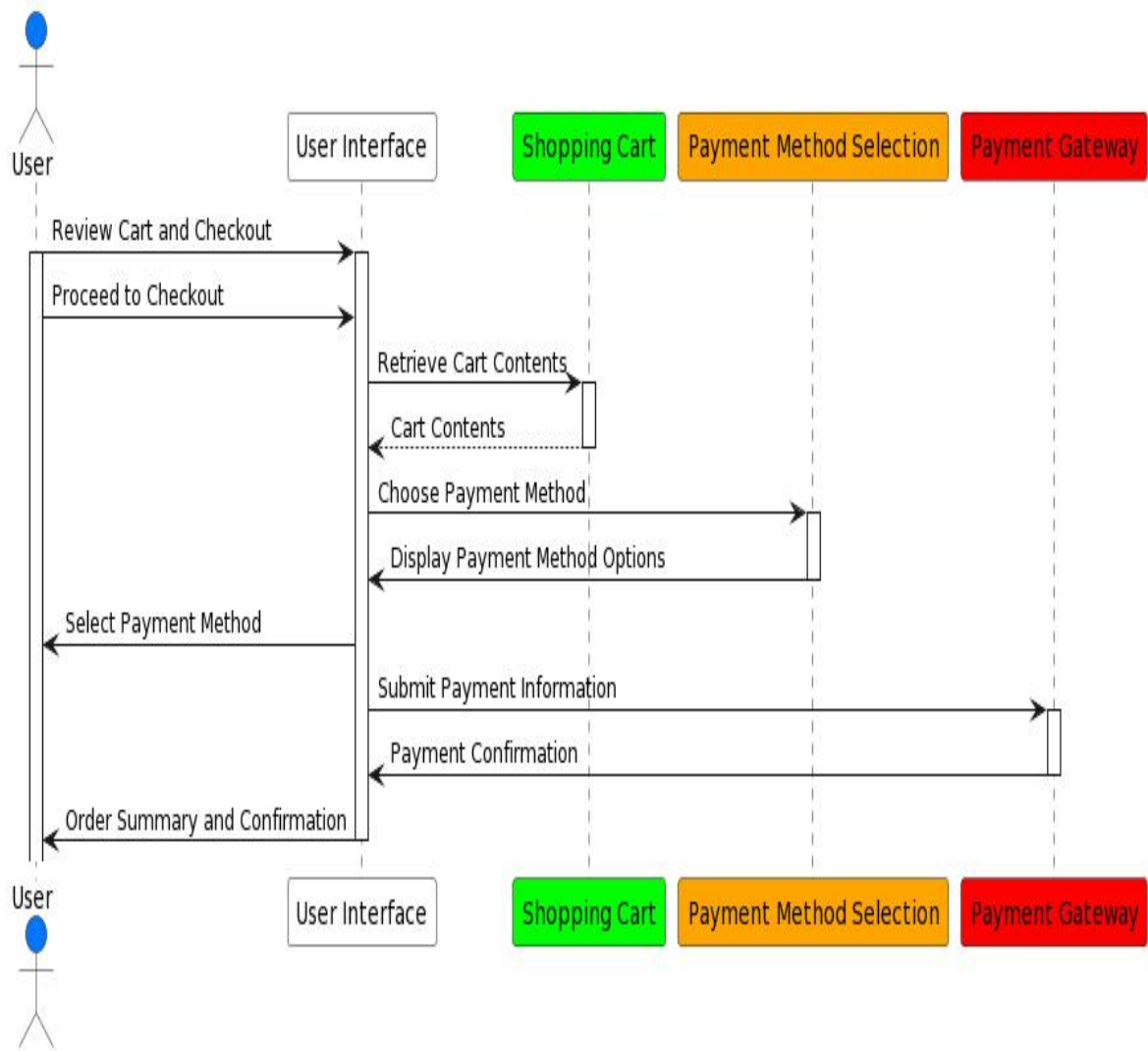
3.5.8 Update Profile

Figure 3.5.8: Sequence Diagram For Update Profile



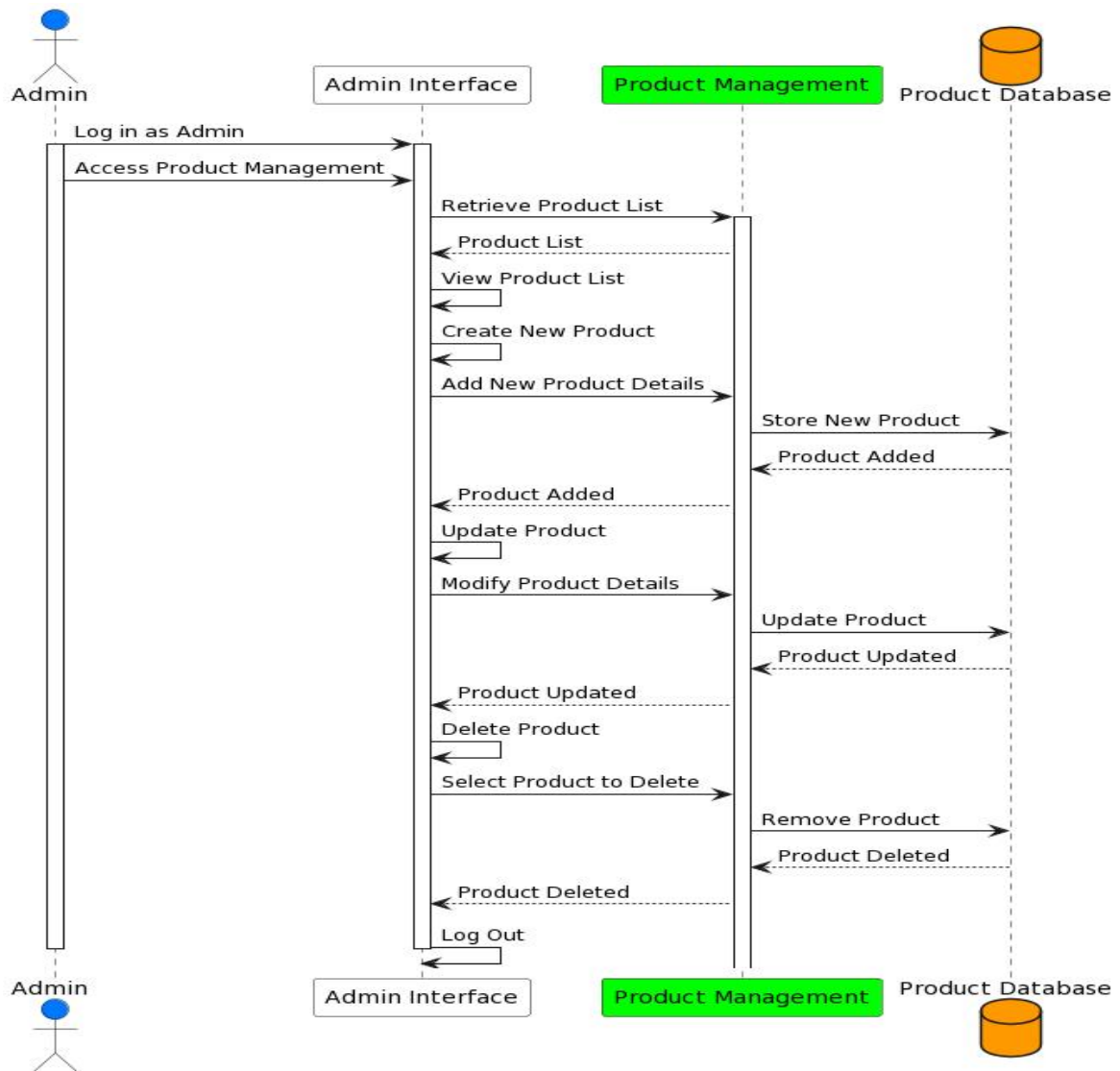
3.5.9 Make Payment

Figure 3.5.9: Sequence Diagram For Make Payment



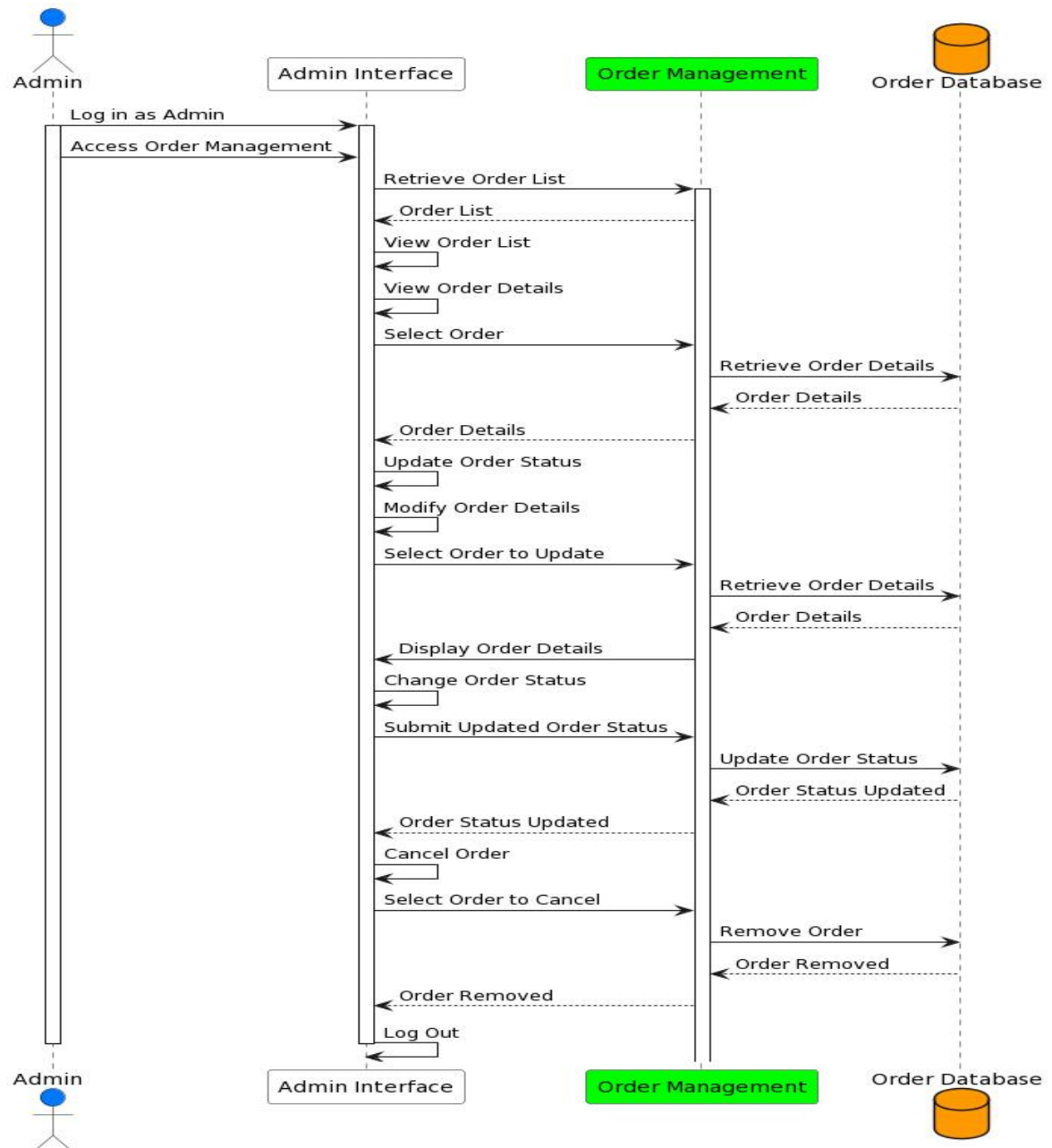
3.5.10 Manage Product

Figure 3.5.10: Sequence Diagram For Add Product



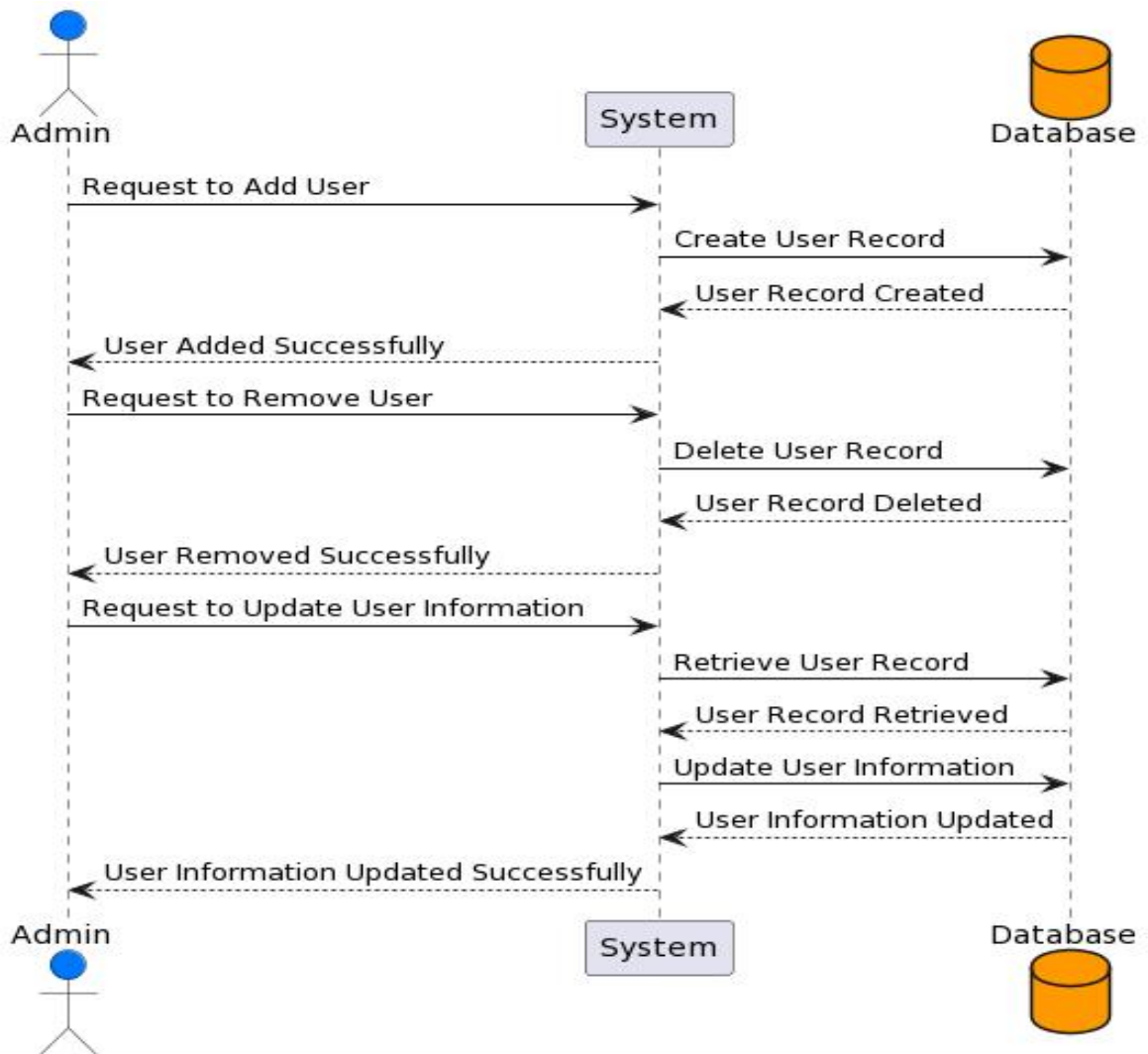
3.5.11 Manage Order

Figure 3.5.11: Sequence Diagram For Manage Order



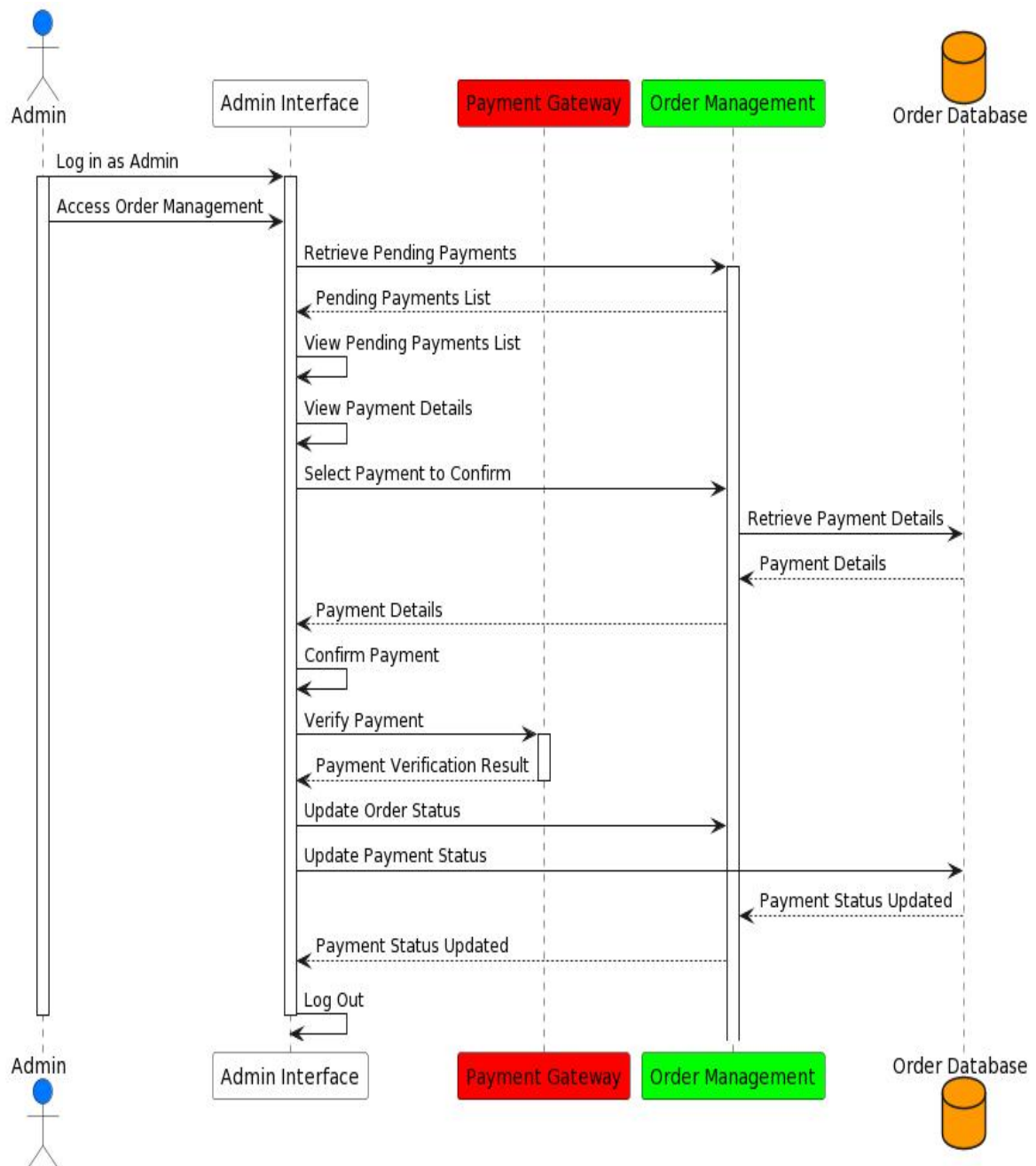
3.5.12 Manage User

Figure 3.5.12: Sequence Diagram For Manage User



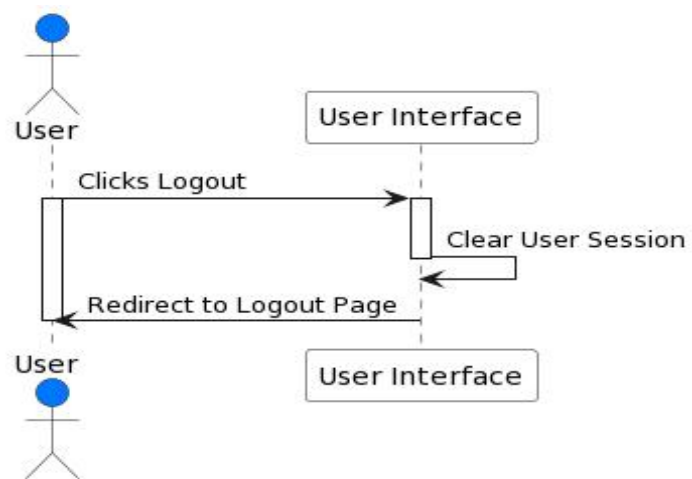
3.5.13 Receive Payment

Figure 3.5.13: Sequence Diagram For Receive Payment



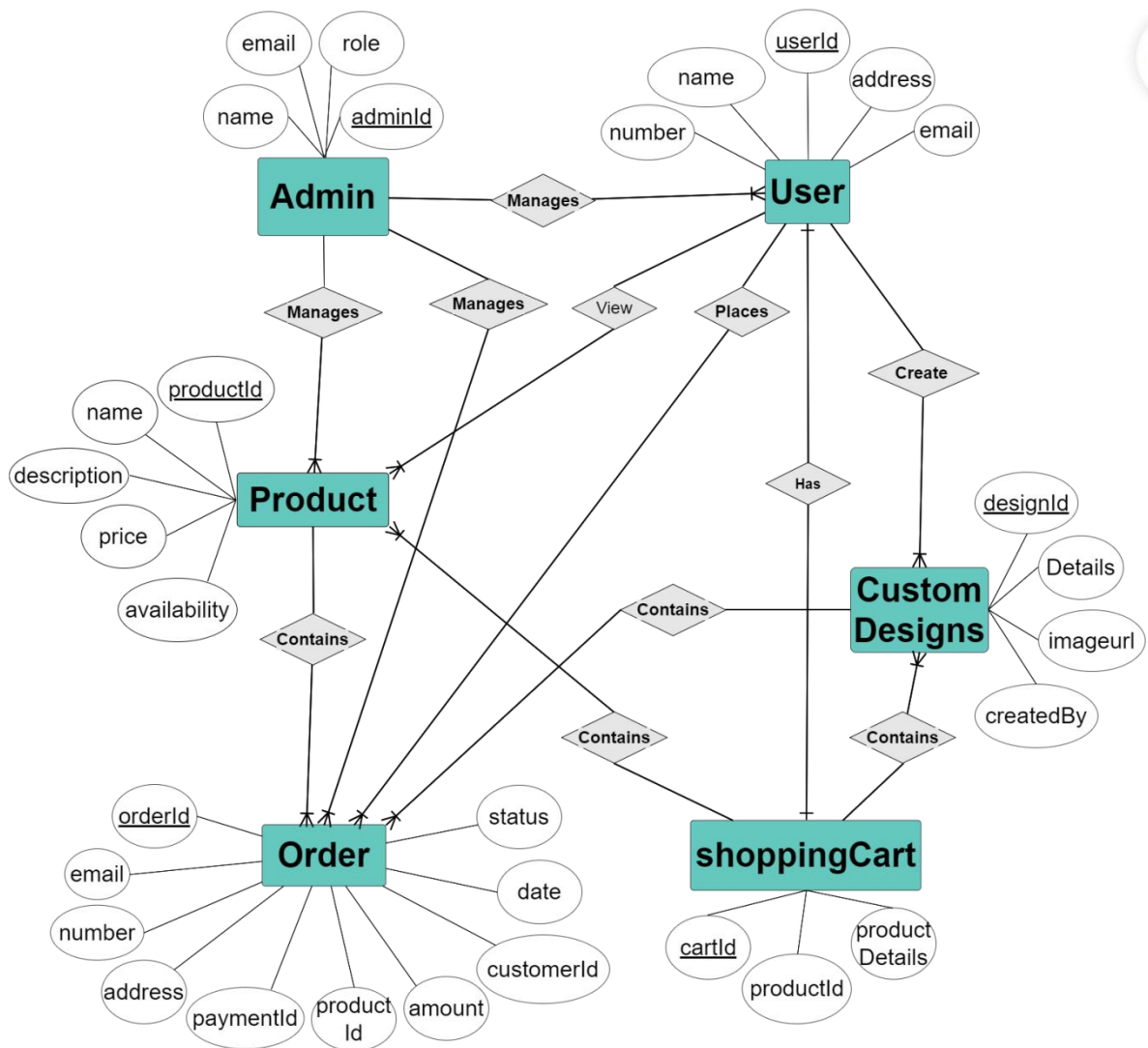
3.5.14 Logout

Figure 3.5.14: Sequence Diagram For Logout



3.6 Entity Relationship Diagram

Figure 3.6 Entity Relationship Diagram For "CUSTOMIZE TEE"



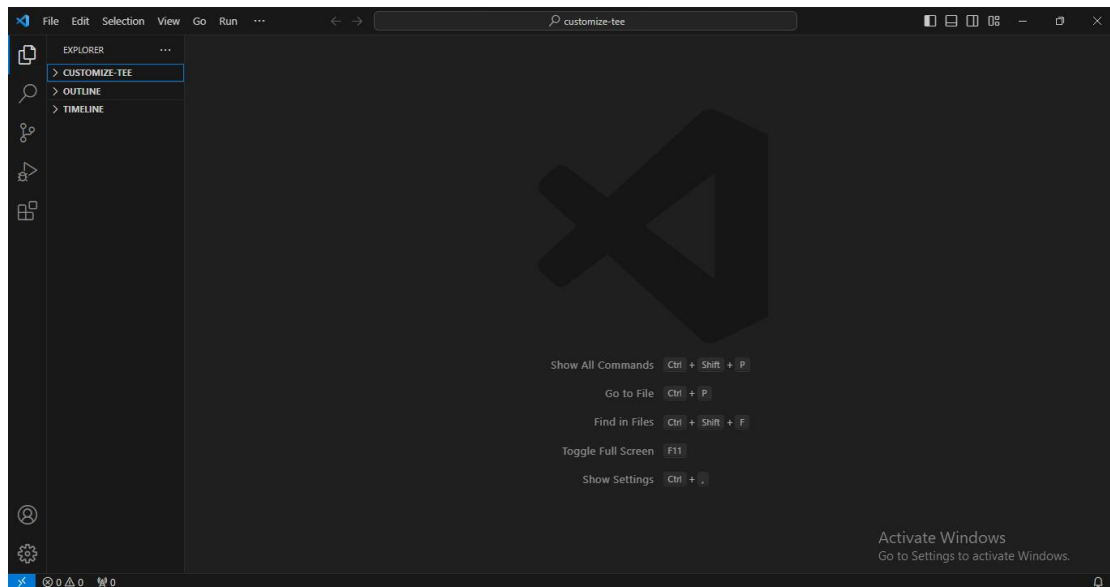
CHAPTER 4

DEVELOPMENT TOOL & TECHNOLOGY

4.1 Integrated Development Environment (IDE)

For the development of Customize Tee, I opted for the widely acclaimed and versatile Integrated Development Environment (IDE) - Visual Studio. Visual Studio provides a comprehensive suite of tools, including advanced code editing features, powerful debugging capabilities, and seamless version control integration.

Figure 4.1 Integrated Development Environment (IDE) - Visual Studio

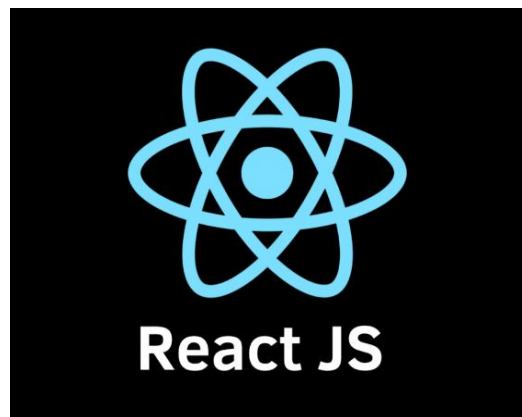


4.2 Programming Language

React

The front-end of Customize Tee is developed using the React framework, a powerful JavaScript library for building dynamic and interactive user interfaces. React's component based architecture facilitates modular development and seamless state management.

Figure 4.2.1 React JS Logo



Node.js

Node.js is a JavaScript runtime that enables server-side execution of JavaScript code outside the browser. It facilitates the development of server-side applications using JavaScript and offers a variety of packages and modules from the npm (Node Package Manager) ecosystem. Node.js is employed on the server side, providing a scalable and efficient runtime environment for executing server-side code.

Figure 4.2.2 Node.js Logo



4.3 User interface Design

Tailwind CSS

Tailwind CSS is employed for styling and theming the user interface. Its utility-first approach and pre-designed components allow for rapid development and customization of the UI, ensuring a sleek and responsive design.

Figure 4.3.1 Tailwind CSS Logo



DaisyUI

DaisyUI, an extension for Tailwind CSS, further enhances the UI design capabilities. It provides additional components and utilities to streamline the customization process, resulting in a polished and cohesive user interface

Figure 4.3.2 DaisyUI Logo



Firestore

Firestore Console is used for managing and configuring Firestore services, including authentication and real-time database functionalities. It serves as a centralized tool for overseeing various aspects of the application's front-end and back-end components.

Figure 4.3.3 Firestore Logo



4.4 Database

MongoDB

MongoDB, a NoSQL database, is chosen for its flexibility and scalability. Its document oriented structure allows for efficient storage and retrieval of complex data, making it an ideal choice for managing user profiles, product details, and order information.

Figure 4.4.1 MongoDB Logo



Firestore Realtime Database

Firestore Realtime Database is utilized for its real-time synchronization capabilities, enabling seamless and instant updates across connected clients. This is particularly beneficial for features such as real-time order tracking.

Figure 4.4.2 Firestore Realtime Database Logo



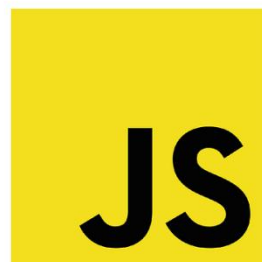
4.5 Deployment and hosting

Express.js

The server-side application is developed using Express.js, a web application framework for Node.js. Express.js facilitates the creation of robust and scalable server-side applications, ensuring optimal performance.

Figure 4.5.1 Express.js Logo

Express



Firestore Hosting

Firestore Hosting is leveraged for deploying and hosting the front-end of Customize Tee. With features like automatic SSL certificates and CDN delivery, Firestore Hosting provides a secure and globally distributed hosting solution.

Figure 4.5.2 Firestore Hosting Logo



4.6 Additional Tools

GitHub

GitHub is utilized for version control and collaborative development. It provides a centralized repository for code storage, version tracking, and collaboration among team members.

Figure 4.6.1 GitHub Logo



Chrome DevTools

Chrome DevTools are used for real-time testing, debugging, and optimizing the frontend code. These tools provide insights into the performance and behavior of the application during development.

Figure 4.6.2 Chrome DevTools Logo



4.7 Dependencies

16. tailwind CSS
17. daisyUI
18. headlessui
19. Heroicons
20. sweetalert2
21. Swiper
22. material-ui
23. stripe-js
24. tanstack react-query
25. axios
26. firebase
27. localforage
28. match-sorter
29. Polotno
30. Recharts
31. react-icons
1. react
2. react-dom
3. react-helmet-async
4. react-hook-form
5. react-responsive-carousel
6. react-router-dom
7. react-simple-captcha
8. react-tabs
9. sort-by
10. jsonwebtoken
11. mongoose
12. Nodemon
13. Cors
14. dotenv
15. express

CHAPTER 5

SYSTEM TESTING

System testing is an integral phase in the development process aimed at ensuring the functionality, performance, and reliability of the Customize Tee.

5.1 Testing Features

5.1.1 Feature To Be Tested

Table 5.1.1 Feature To Be Tested

Features	Priority	Description
Registration	1	Customers must register for login to access the system.
Login & Logout	1	Customers and Admin must log in to use the system.
View Products	1	Customers can view all products and categories in the system.
Add To Cart	1	Customers can add their desired T-shirt from the system to the cart.
Manage Cart	2	Customers can manage their cart, including removing products.
Place Order	1	Customers can place orders with shipping information.
Design Custom T-shirt	1	Customers can create their own design T-shirt and add the design on any position of a T-shirt.
Update Profile	2	Customers can update their information.
Make Payment	1	Customers can make payments using a payment gateway.
Manage Product	1	Admin can add, remove, and edit products and categories.
Manage Order	1	Admin can confirm or reject placed orders by customers and update the order status.
Manage User	1	Admin can view all users, add, remove users, and manage user roles.
Receive Payment	1	Admin can view received payments for products and update customer order status.

5.2 Testing Strategies

5.2.1 Test Approach

The testing approach involves a combination of manual and automated testing methodologies:

Manual Testing:

- User acceptance testing (UAT) for end-to-end scenarios.
- Exploratory testing for uncovering potential issues.
- Regression testing to ensure new features do not impact existing functionalities.

Automated Testing:

- Integration testing for API endpoints.
- Unit testing for critical functions.
- Performance testing for responsiveness and scalability.

5.2.2 Pass/Fail Criteria

Pass Criteria:

- All critical features must function as intended.
- No critical bugs or security vulnerabilities.
- Compliance with design specifications.

Fail Criteria:

- Critical features are dysfunctional.
- Security vulnerabilities compromise user data.
- Non-compliance with design specifications.

5.2.3 Testing Schedule

Table 5.2.3 Time Schedule

Test Phase	Time
Create Test Plan	1 Week
Unit Testing	During Development
Component Testing	During Development
User Interface Testing	2 Week
Performance Testing	2 Week
Accessibility Test	3 Week

5.3 Test Cases

5.3.1 Registration

Table 5.3.1 Test Case For Registration

Test Id	Test Case	Test Description	Executed By	Date
01	Registration	This scenario oversees the functionality related to user registration.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Click on the Register Button	Required Message	Required Message	Pass
2	Put name, password, phone, address and wrong email	Required Error Message	No Required Message	Fail
3	Put valid email, and blank name, password, phone, address	Required Error Message	No Required Message	Fail
4	Put valid name, password, phone, address, and email	Registered	User Registered Successfully	Pass

5.3.2 Login

Table 5.3.2 Test Case For Login

Test Id	Test Case	Test Description	Executed By	Date
02	Login	This scenario oversees the functionality related to user login.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Click on Login Button	Required Message	Required Message	Pass
2	Put valid email and wrong password	Required Error Message	No Required Message	Fail
3	Blank email and password field	Required Error Message	No Required Message	Fail
4	Put valid email and password	Login	Login Successfully	Pass

5.3.3 View Products

Table 5.3.3 Test Case For View Products

Test Id	Test Case	Test Description	Executed By	Date
03	View Products	This scenario oversees the functionality related to View Products.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Navigate to the "View Products" section	Product list is displayed	The user can see all available products and categories	Pass
2	Click on a specific category	Products filtered by the selected category are displayed	Only products from the selected category are shown	Pass
3	Click on a product	Product details page is opened	Detailed information about the selected product is shown	Pass
4	Add the product to the cart	Product is added to the cart	The user receives a confirmation message, and the cart is updated	Pass
5	Navigate to the cart	Cart page is displayed	The user can see the added product in the cart	Pass
6	Remove a product from the cart	Product is removed from the cart	The cart is updated, and the removed product is no longer visible	Pass

5.3.4 Access Control

Table 5.3.4 Test Case For Access Control

Test Id	Test Case	Test Description	Executed By	Date
04	Access Control	This scenario validates the access control mechanism for user and admin roles.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Attempt to access admin features with a regular user account	Access denied	The system should restrict access to admin-specific features for regular users	Pass
2	Log in as an admin	Admin dashboard is displayed	The admin should have access to admin-specific features and information	Pass
3	Navigate to user-specific pages as an admin	Access denied	The admin should be restricted from accessing user-specific pages	Pass
4	Log out	User is logged out	The system should properly log out the user or admin	Pass
5	Attempt to access pages without logging in	Redirected to the login page	Unauthenticated users should be prompted to log in	Pass
6	Log in with incorrect credentials	Authentication failed message	Users should be notified of authentication failure	Pass
7	Log in with correct credentials	Dashboard is displayed	Authenticated users should be directed to their respective dashboards	Pass

5.3.5 Placing an Order

Table 5.3.5 Test Case For Placing an Order

Test Id	Test Case	Test Description	Executed By	Date
05	Placing an Order	This scenario verifies the process of a customer placing an order.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Navigate to the cart page	Cart contents displayed	The system should show the selected products in the user's cart	Pass
2	Verify cart contents	Correct products and quantities	The displayed cart contents should match the products and quantities selected by the user	Pass
3	Click on the "Proceed to Checkout" button	Redirected to the checkout page	The system should guide the user to the checkout page for order processing	Pass
4	Review order summary	Correct order details displayed	The system should present an order summary for user confirmation	Pass
5	Select payment method	Payment method chosen	Users should be able to choose their preferred payment method	Pass
6	Click on the "Place Order" button	Order confirmation message	The system should confirm the successful placement of the order	Pass
7	Verify order status	Order status updated	The order status should reflect as "Pending" or a similar status indicating the order has been placed	Pass

5.3.6 Custom Design

Table 5.3.6 Test Case For Custom Design

Test Id	Test Case	Test Description	Executed By	Date
06	Custom Design	This scenario verifies the process of Custom Design.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Navigate to the "Custom Design" section	Customization tools displayed	The system should show design tools such as color palette, design elements, and placement options	Pass
2	Choose T-shirt color	Selected color displayed	The system should update the T-shirt preview with the chosen color	Pass
3	Add design elements	Elements added to the T-shirt	The system should allow users to add graphics, text, or other design elements to the T-shirt	Pass
4	Adjust design placement	Design placement modified	Users should be able to move, resize, or rotate the added design elements	Pass
5	Add the custom design to the cart	Design added to the cart	The system should allow users to add the customized T-shirt to their shopping cart	Pass
6	Proceed to checkout	Redirected to the checkout page	Users should be able to proceed to checkout with their customized T-shirt in the cart	Pass

5.3.7 Manage Product

Table 5.3.7 Test Case For Manage Product

Test Id	Test Case	Test Description	Executed By	Date
07	Manage Product	This scenario verifies the process of Manage Product.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Navigate to "Manage Product" section	Product list displayed	The system should show a list of existing products with options for management	Pass
2	Add a new product	Product added successfully	The system should allow the admin to add a new product with details such as name, description, price, and category	Pass
3	Edit an existing product	Product details updated	The admin should be able to edit the details (e.g., name, description, price) of an existing product	Pass
4	Remove a product	Product removed successfully	The system should allow the admin to remove an existing product from the inventory	Pass
6	Manage product categories	Categories updated	The admin should be able to add, edit, or remove product categories	Pass

5.3.8 Manage Order

Table 5.3.8 Test Case For Manage Order

Test Id	Test Case	Test Description	Executed By	Date
08	Manage Order	This scenario verifies the process of Manage Order.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Navigate to "Manage Order" section	Order list displayed	The system should show a list of existing orders with options for management	Pass
2	View order details	Order details displayed	The admin should be able to view detailed information about a specific order, including customer details, items, and status	Pass
3	Confirm an order	Order status updated to "Confirmed"	The admin should be able to confirm a pending order, indicating that it is ready for processing	Pass
4	Reject an order	Order status updated to "Rejected"	The admin should be able to reject a pending order, indicating that it cannot be fulfilled	Pass
5	Update order status	Order status changed successfully	The admin should be able to update the status of an order (e.g., "Processing," "Shipped," "Delivered")	Pass

5.3.9 Manage User

Table 5.3.9 Test Case For Manage User

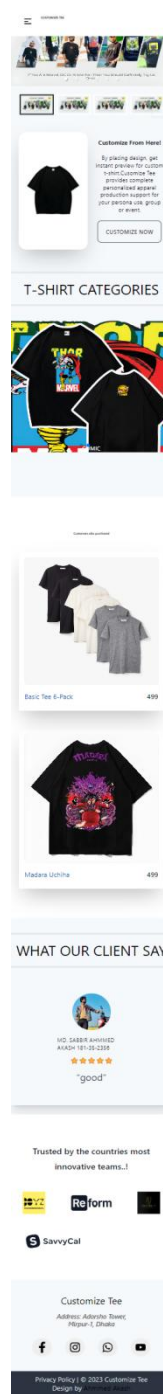
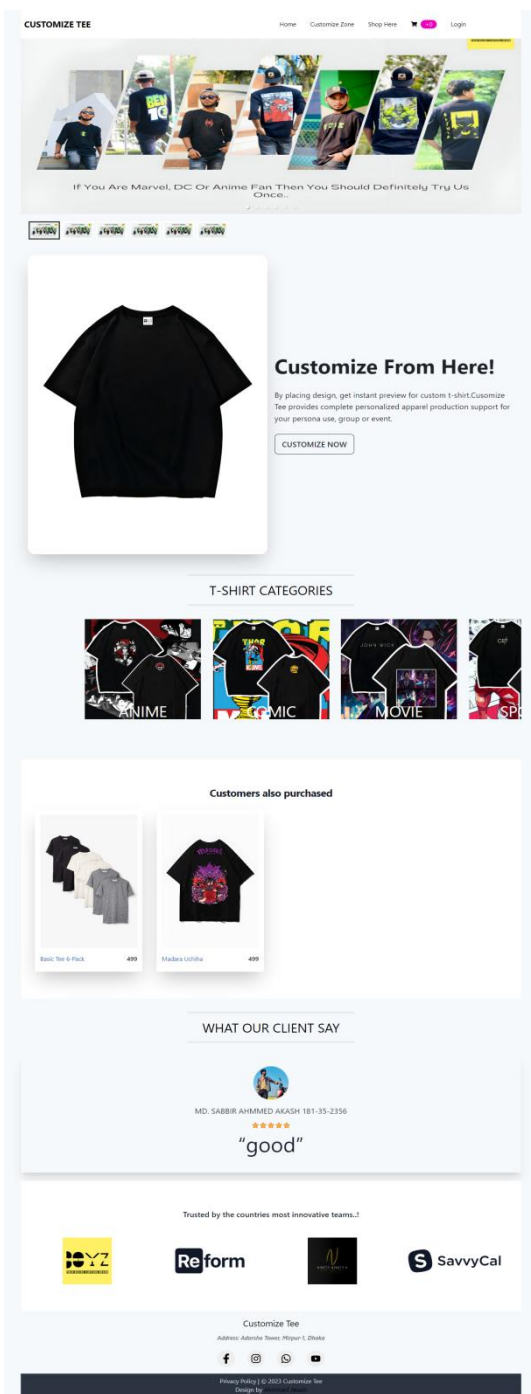
Test Id	Test Case	Test Description	Executed By	Date
09	Manage User	This scenario verifies the process of Manage User.	Md. Sabbir Ahmmed Akash	27/12/2023

Steps	Action	Action Result	Expected System Response	Pass/Fail
1	Navigate to "Manage User" section	User list displayed	The system should show a list of registered users with options for management	Pass
2	View user details	User details displayed	The admin should be able to view detailed information about a specific user, including profile data and role	Pass
3	Add a new user	User added successfully	The admin should be able to add a new user to the system, specifying user details and role	Pass
4	Remove a user	User removed successfully	The admin should be able to remove an existing user from the system	Pass
5	Update user information	User information updated successfully	The admin should be able to modify user details, such as name, email, or role	Pass
6	Change user role	Role changed successfully	The admin should be able to modify the role of a user (e.g., change from customer to admin)	Pass

CHAPTER 6 USER MANUAL

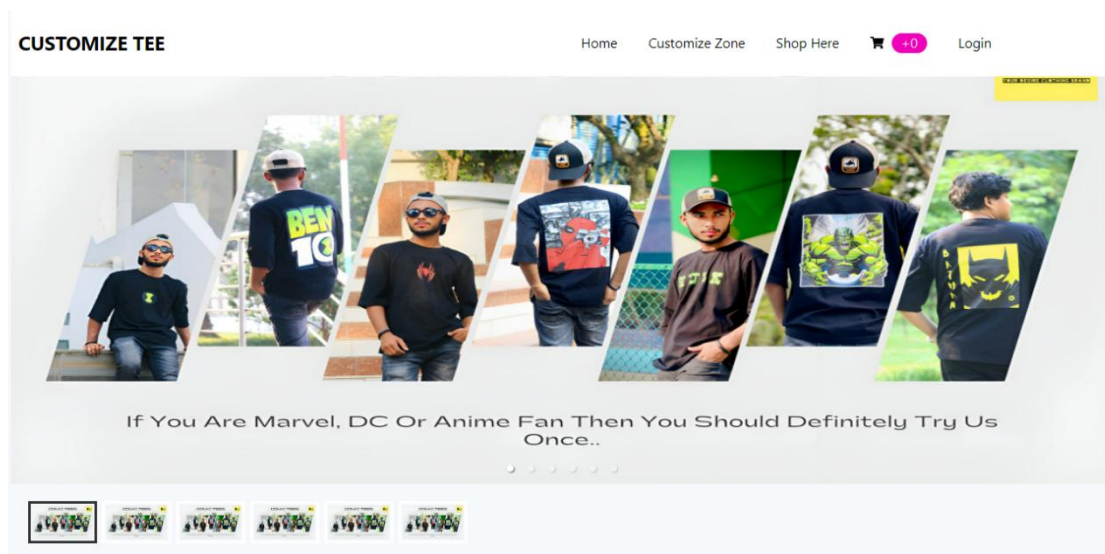
6.1 CUSTOMIZE TEE - Home Page

Figure 6.1 Responsive Home Page



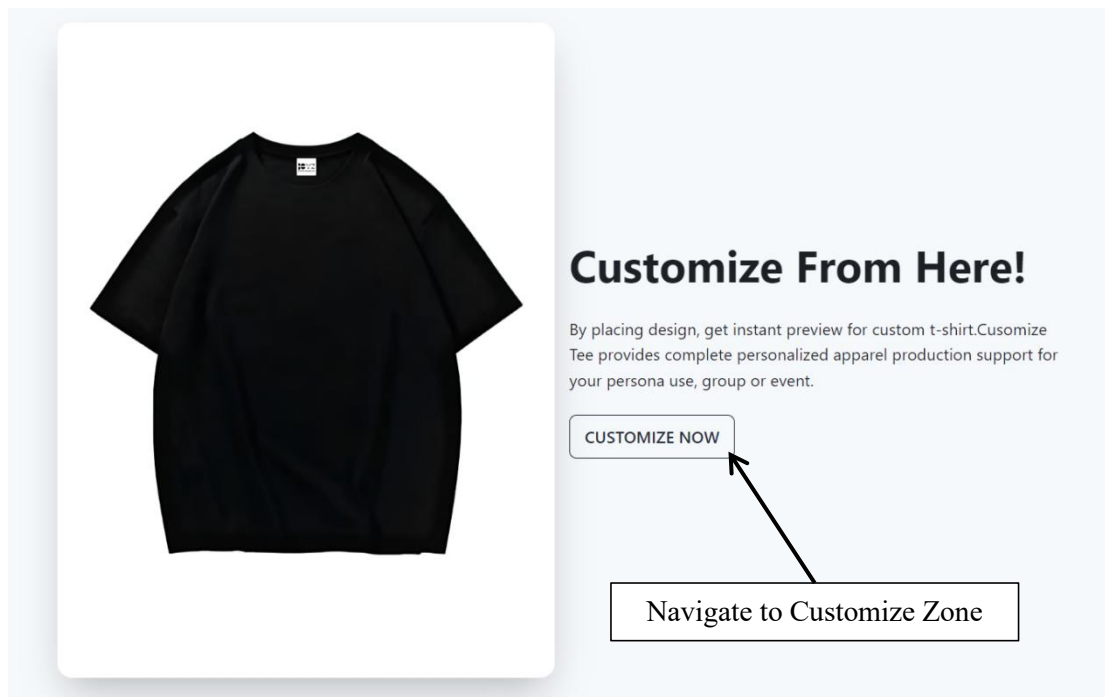
6.1.1 Home Page Banner

Figure 6.1.1 Banner Section



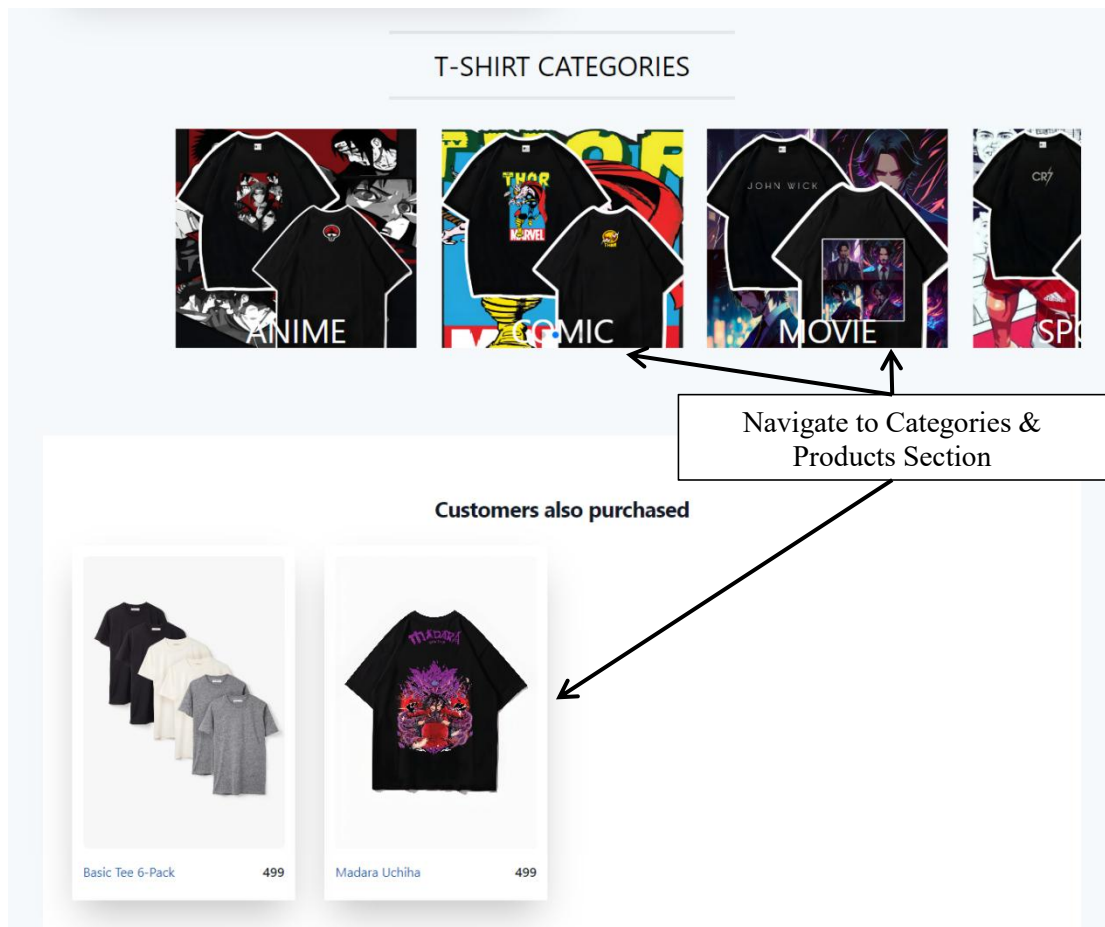
6.1.2 Home Page Customize Section

Figure 6.1.2 Customize Section



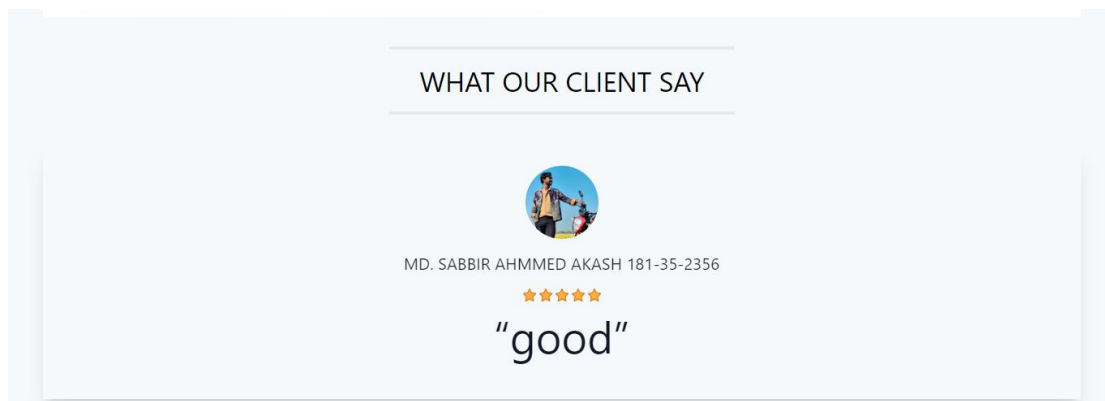
6.1.3 Home Page Categories & Products Section

Figure 6.1.3 Categories & Products Section



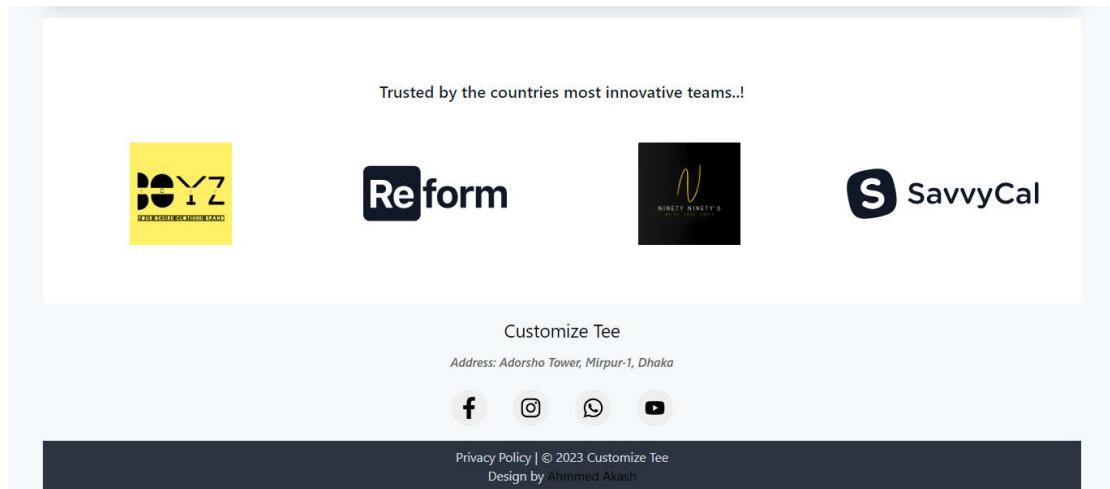
6.1.4 Home Page Reviews Section

Figure 6.1.4 Reviews Section



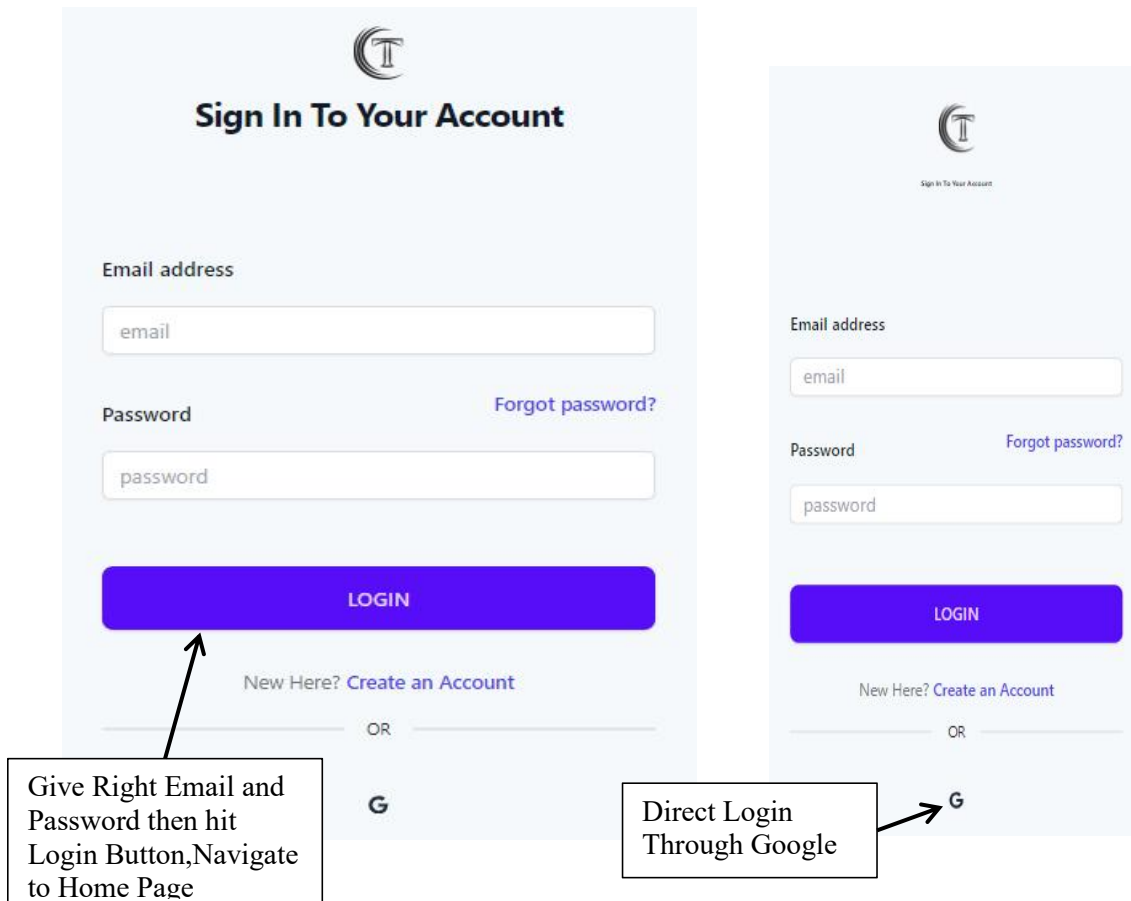
6.1.5 Home Page Footer Section

Figure 6.1.5 Footer Section



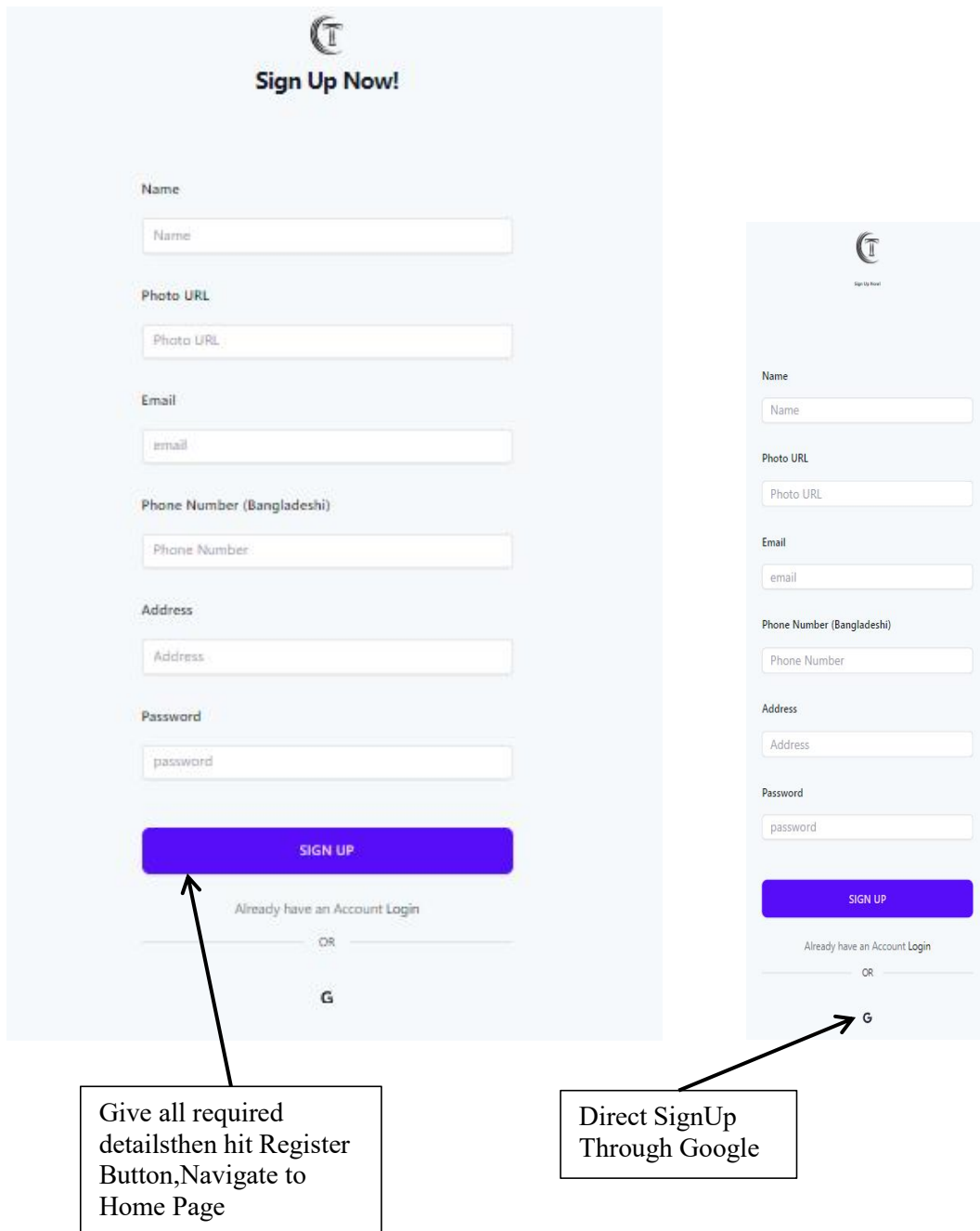
6.2 CUSTOMIZE TEE - Login Page

Figure 6.2 Responsive Login Page



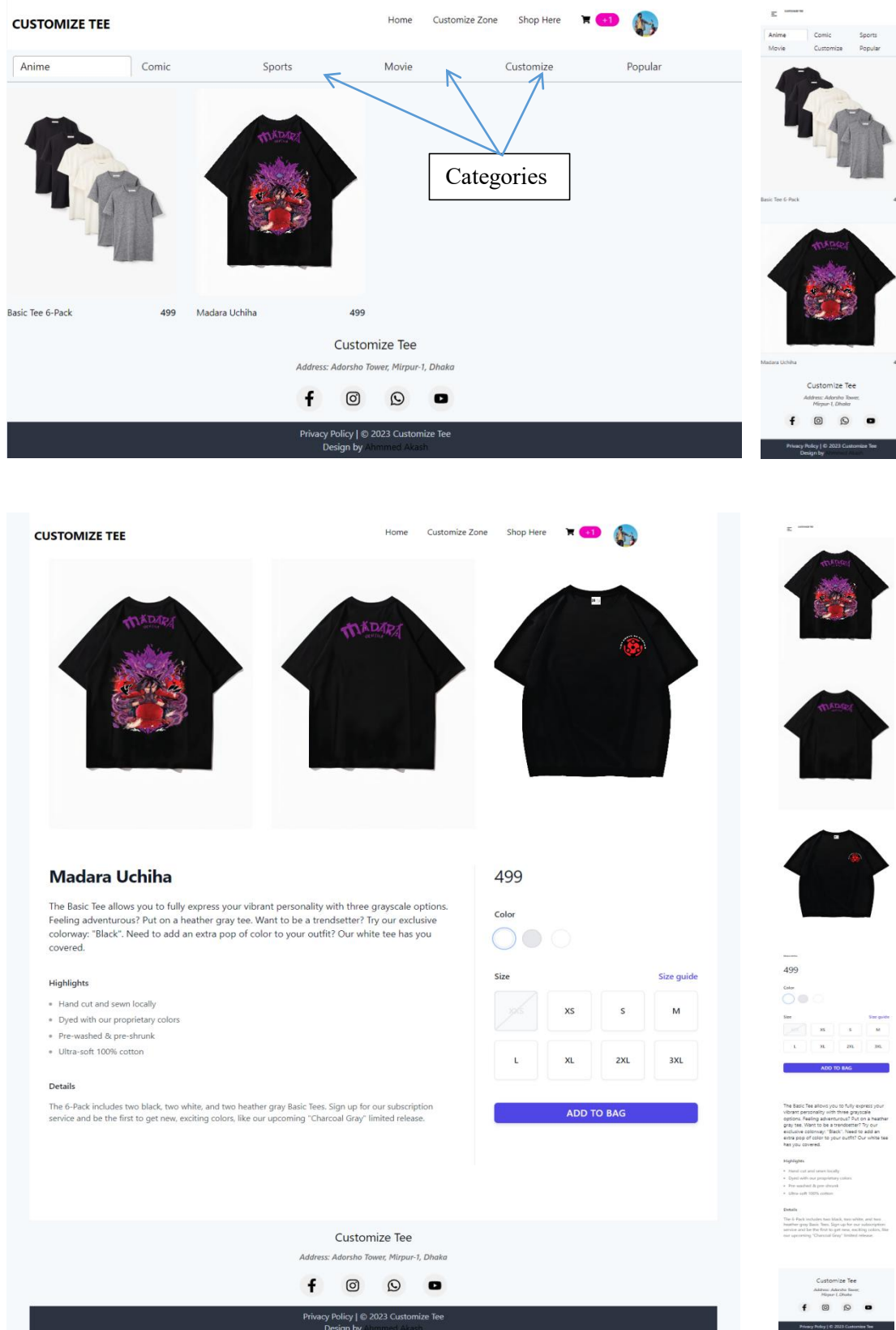
6.3 CUSTOMIZE TEE - User Registration Page

Figure 6.3 Responsive Registration Page



6.4 CUSTOMIZE TEE - View Products

Figure 6.4 Responsive Categories & Product Preview Page



6.5 CUSTOMIZE TEE - Add To Cart

Figure 6.5 Add To Cart

CUSTOMIZE TEE

Home

Madara Uchiha

The Basic Tee allows you to fully express your vibrant personality with three grayscale options. Feeling adventurous? Put on a heather gray tee. Want to be a trendsetter? Try our exclusive colorway: "Black". Need to add an extra pop of color to your outfit? Our white tee has you covered.

Highlights

- Hand cut and sewn locally
- Dyed with our proprietary colors
- Pre-washed & pre-shrunk
- Ultra-soft, 100% cotton

Details

The 6-Pack includes two black, two white, and two heather gray Basic Tees. Sign up for our subscription service and be the first to get new, exciting colors, like our upcoming "Charcoal Gray" limited release.

499

Color

Size

Size guide

XS S M

L XL 2XL 3XL

ADD TO BAG

Add to Cart Successfully

User select t-shirt color ,size then click add to bag

Customize Tee

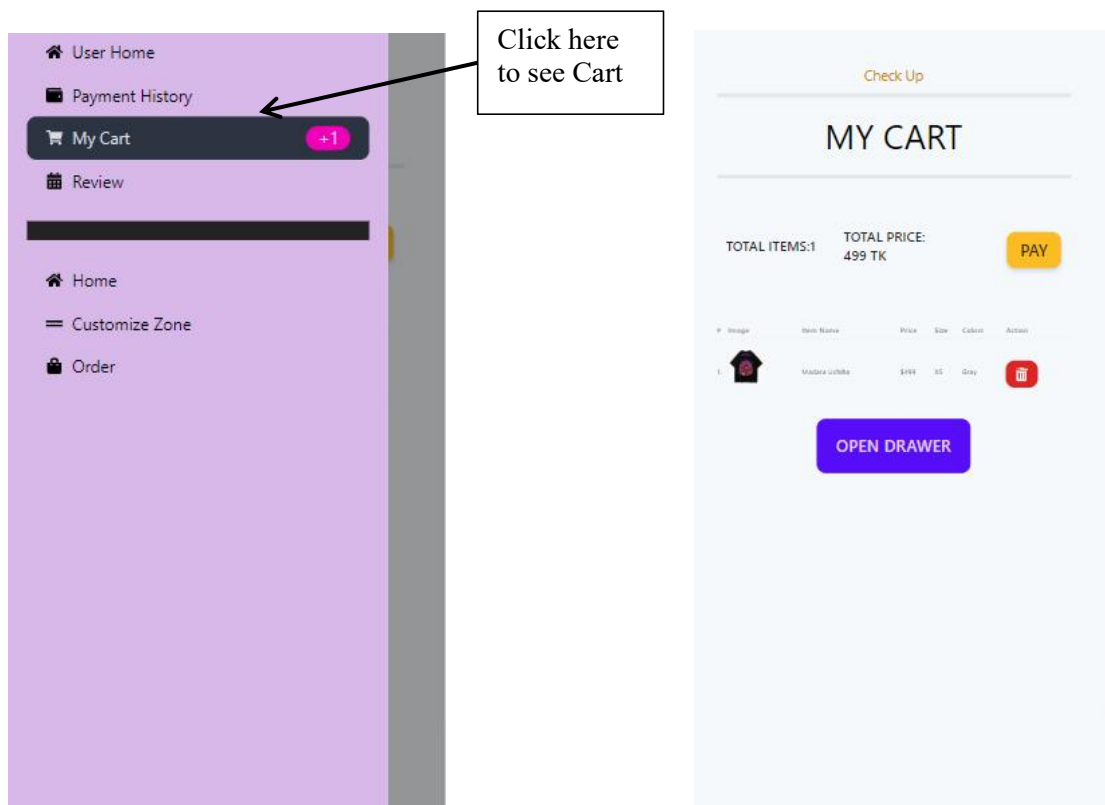
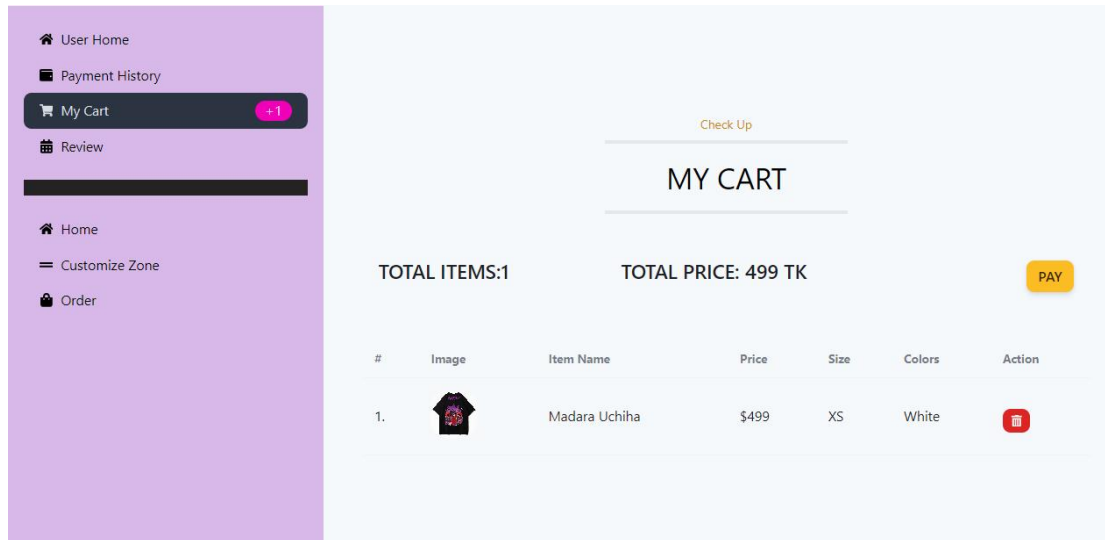
Address: Adorsho Tower, Mirpur-1, Dhaka

f @

Privacy Policy | © 2023 Customize Tee
Design by Ahmmed Akash

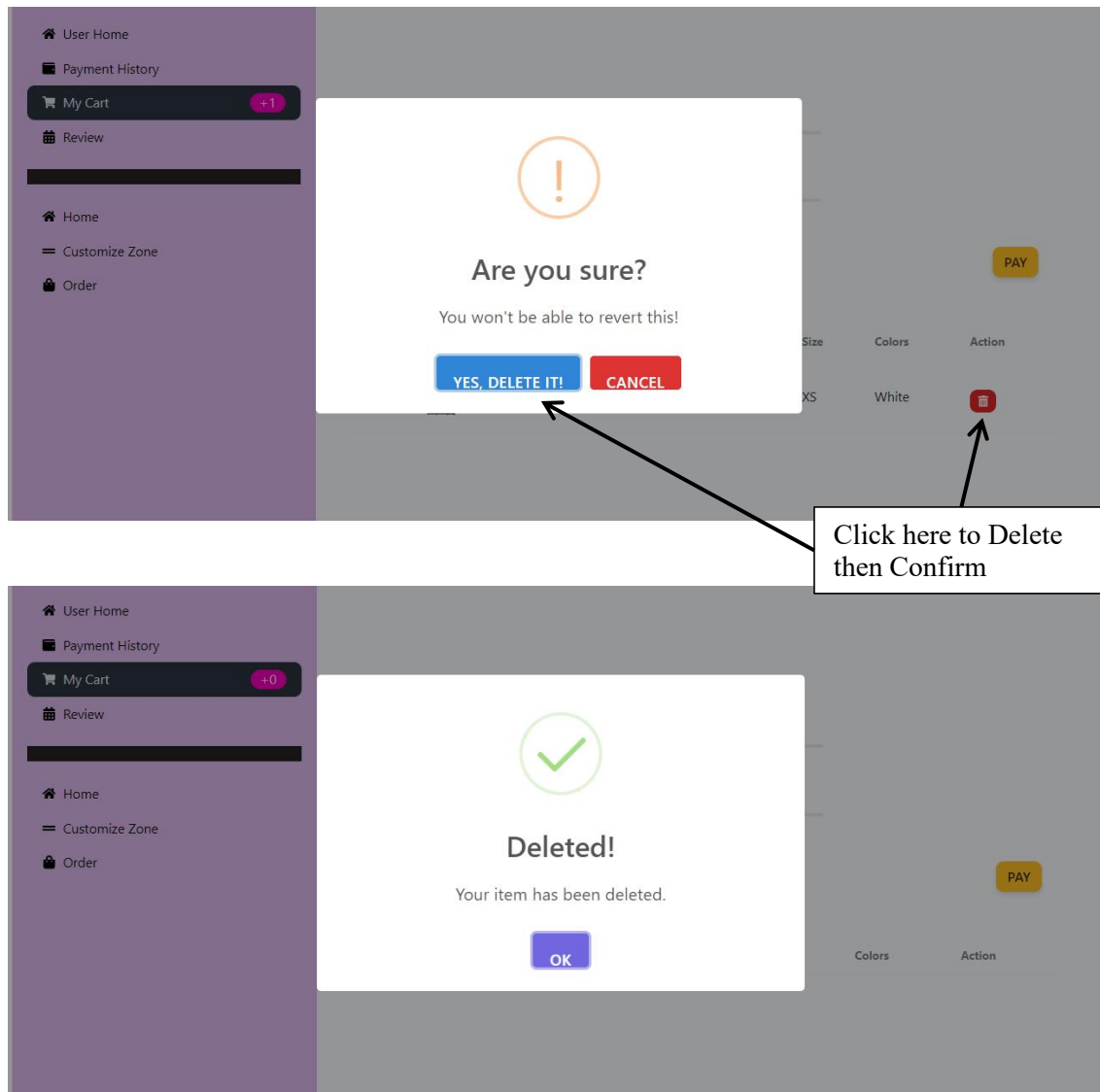
6.6 CUSTOMIZE TEE - My Cart

Figure 6.6 Responsive My Cart



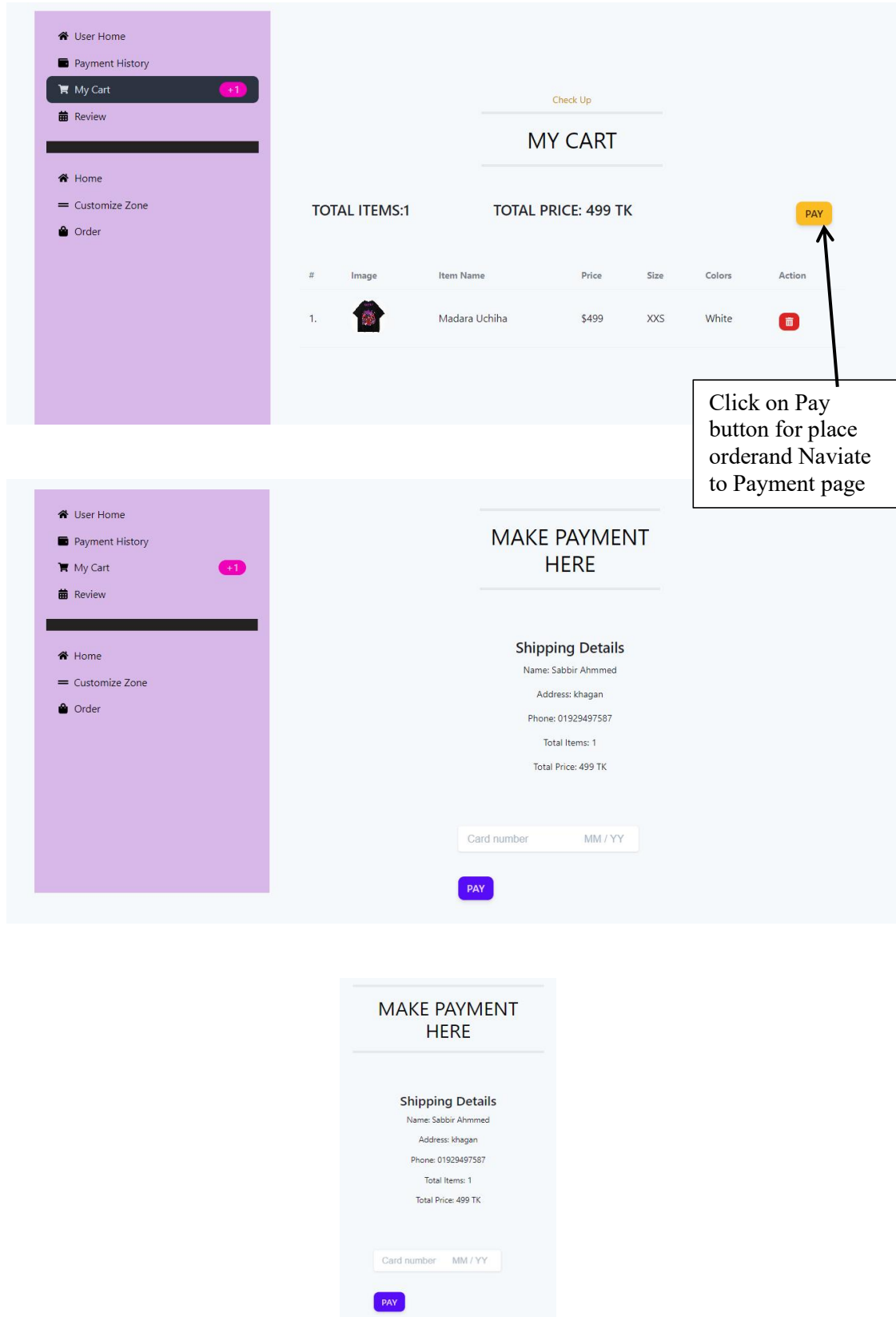
6.7 CUSTOMIZE TEE - Manage Cart

Figure 6.7 Manage Product From My Cart



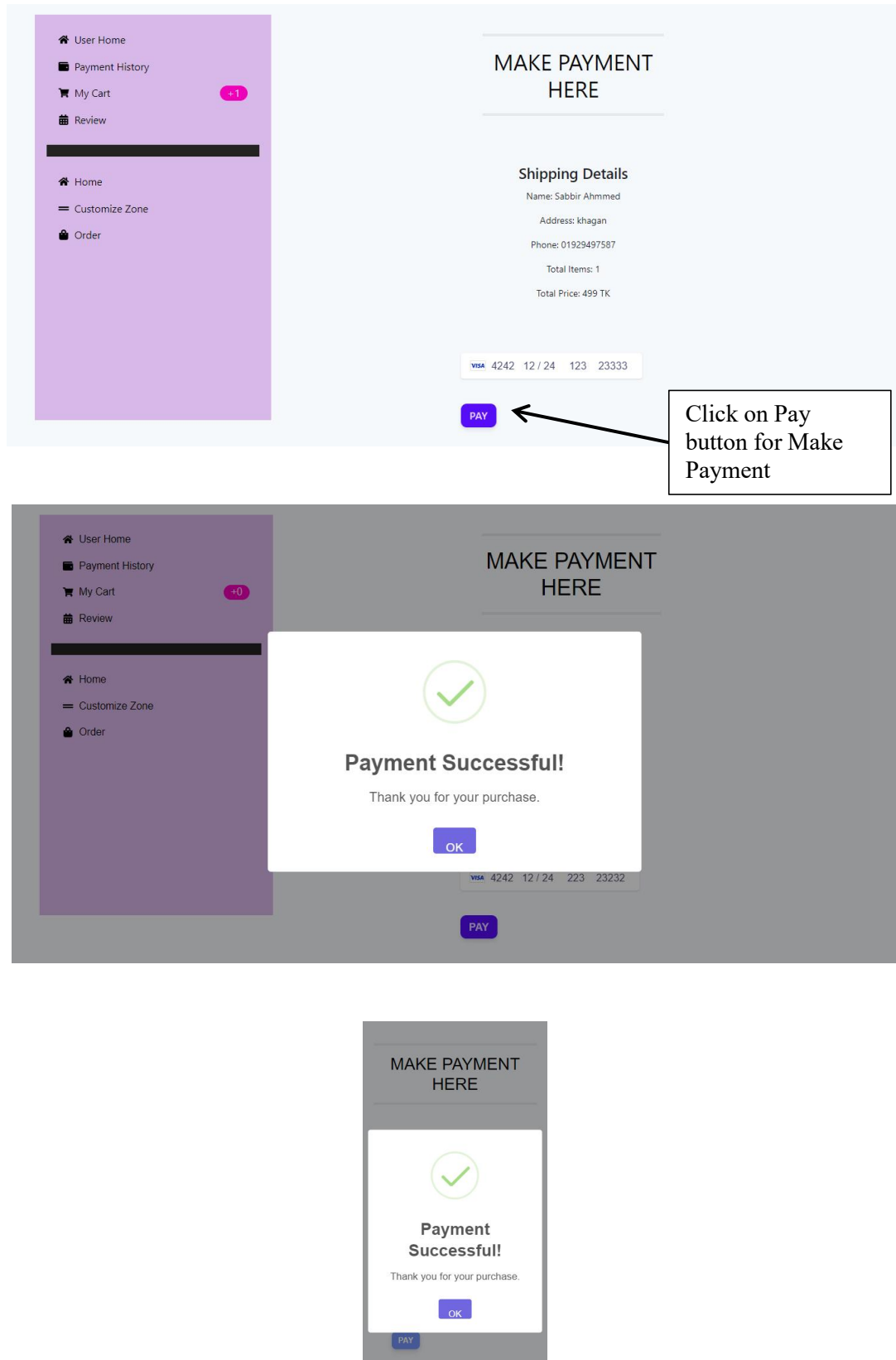
6.8 CUSTOMIZE TEE - Place Order

Figure 6.8 Responsive Place Order From My Cart



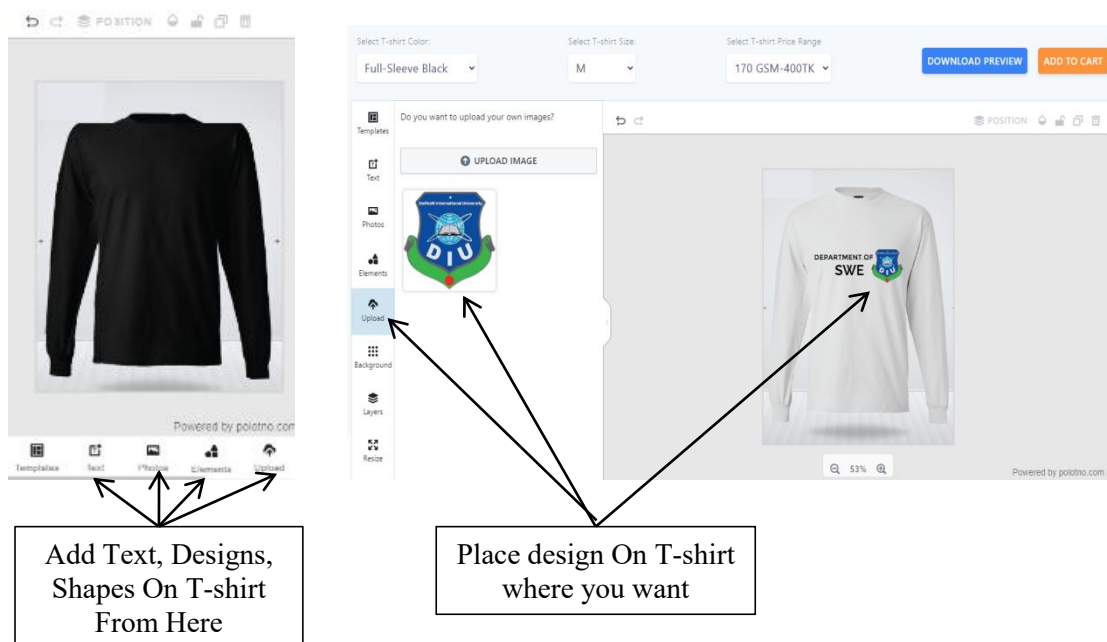
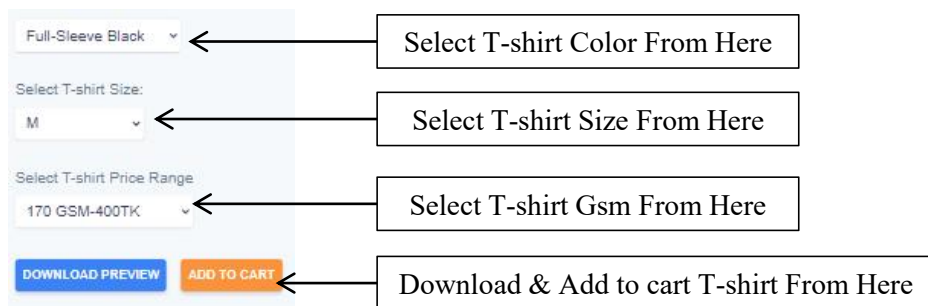
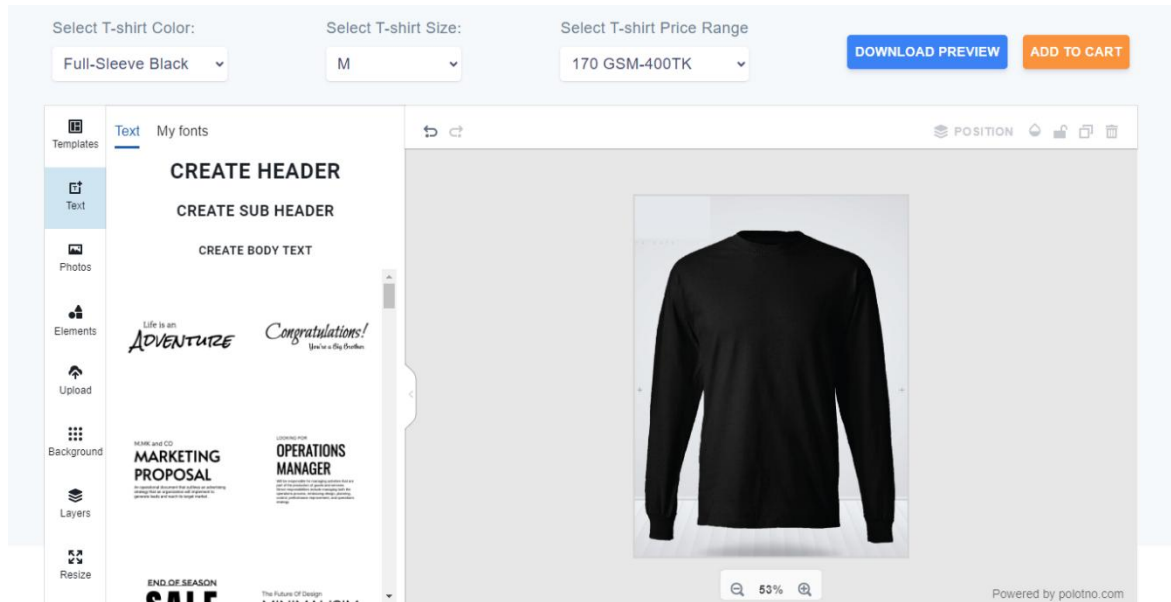
6.9 CUSTOMIZE TEE - Make Payment

Figure 6.9 Responsive Make Payment For Order



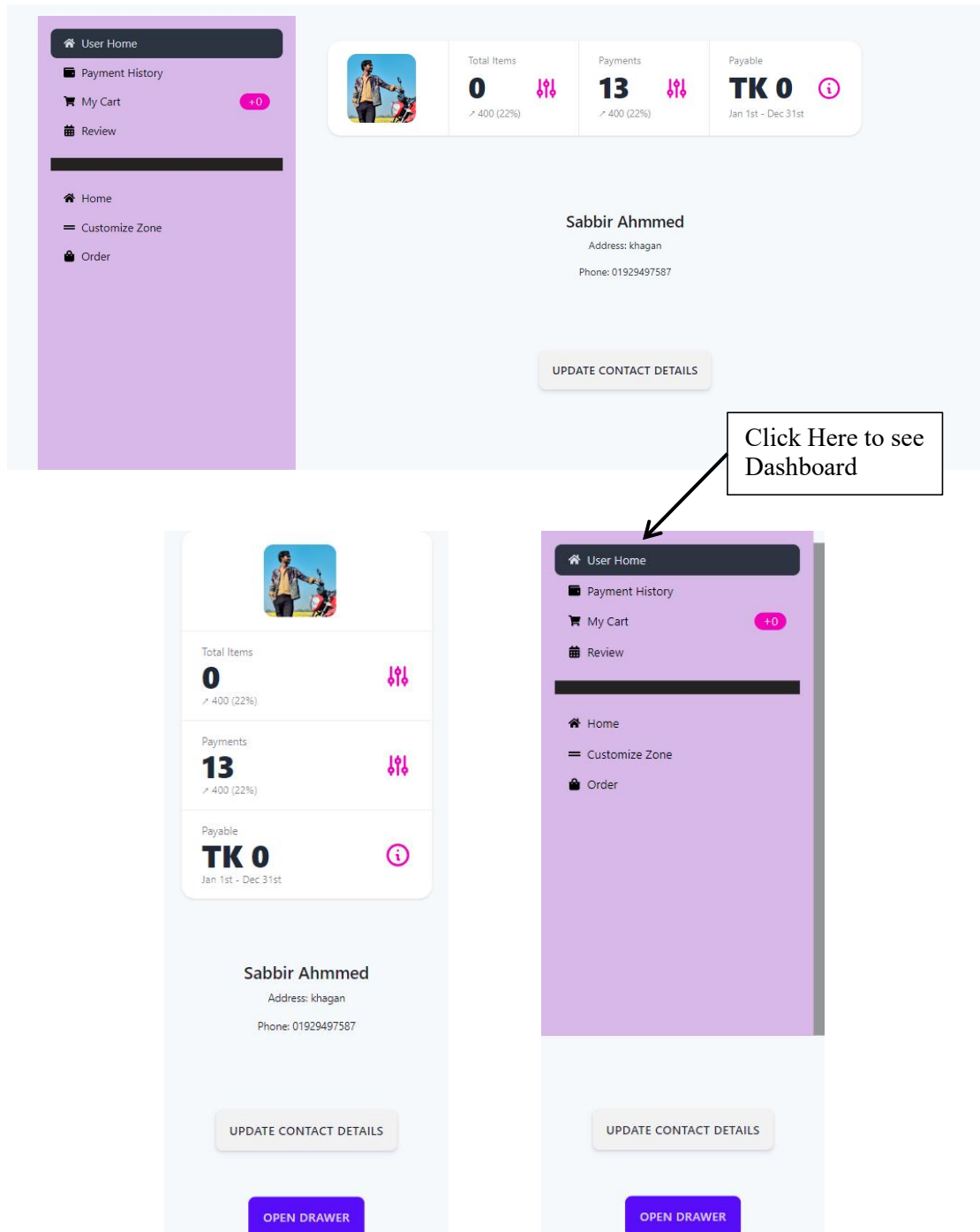
6.10 CUSTOMIZE TEE - Design Custom T-shirt

Figure 6.10 Responsive Design Custom T-shirt



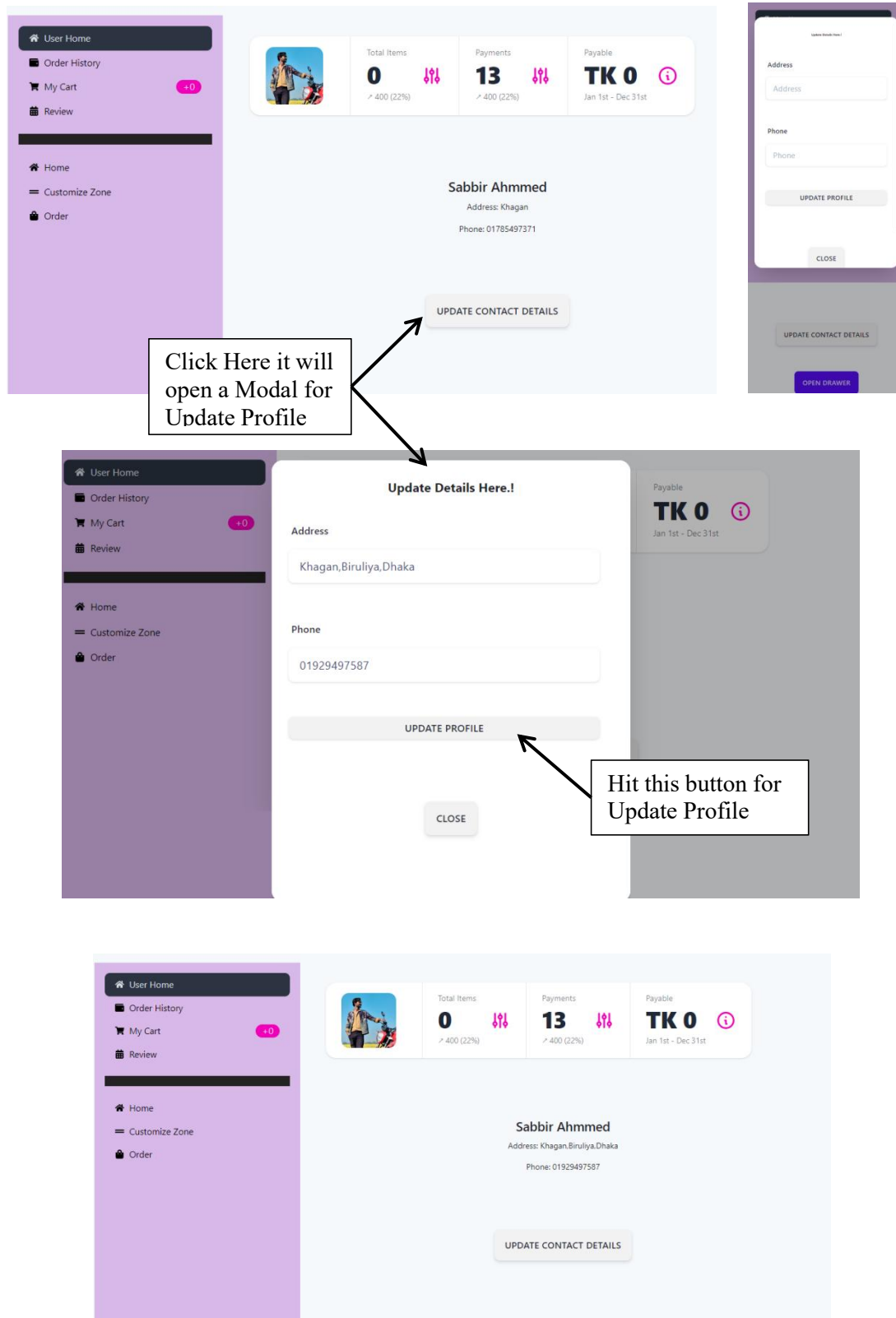
6.11 CUSTOMIZE TEE - User Dashboard

Figure 6.11 Responsive User Dashboard



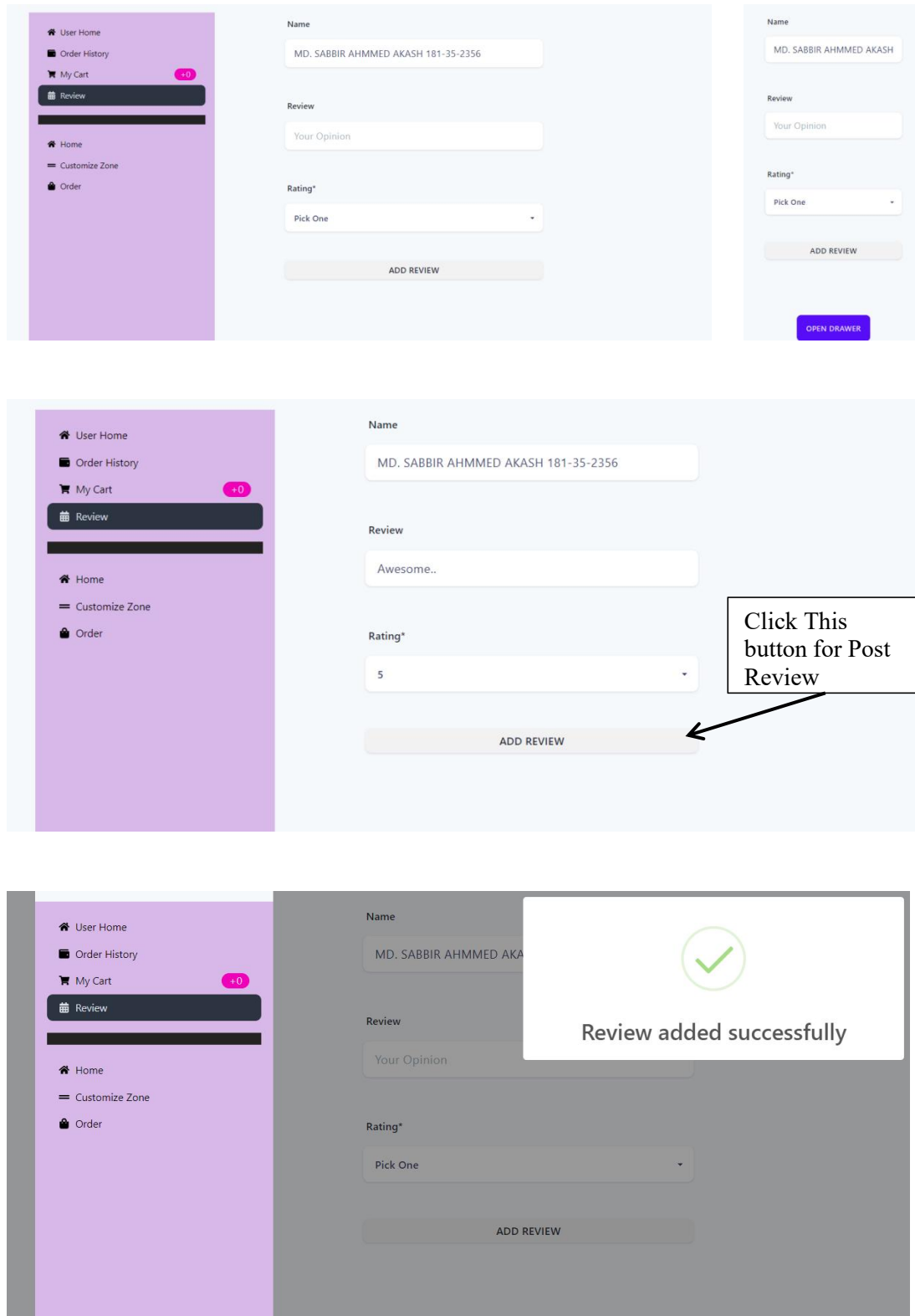
6.12 CUSTOMIZE TEE - Update Profile

Figure 6.12 Responsive Update Profile



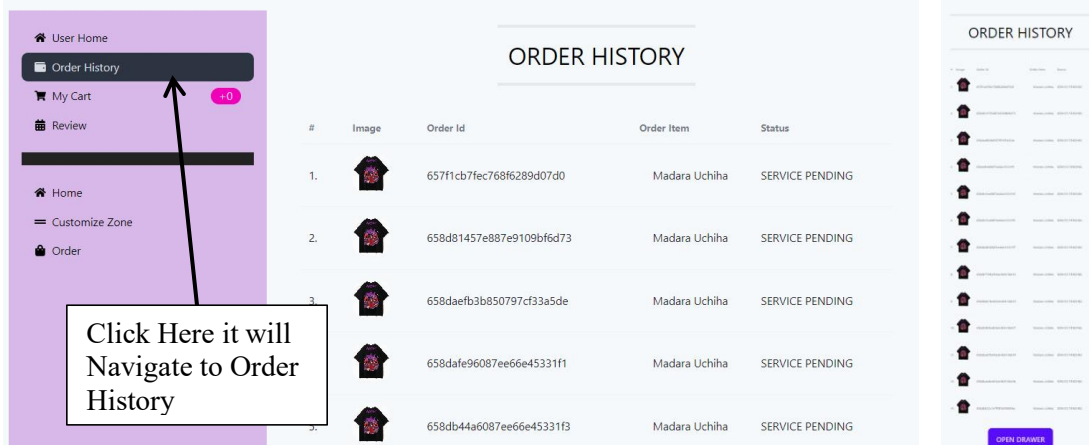
6.13 CUSTOMIZE TEE - User Review

Figure 6.13 Responsive User Review



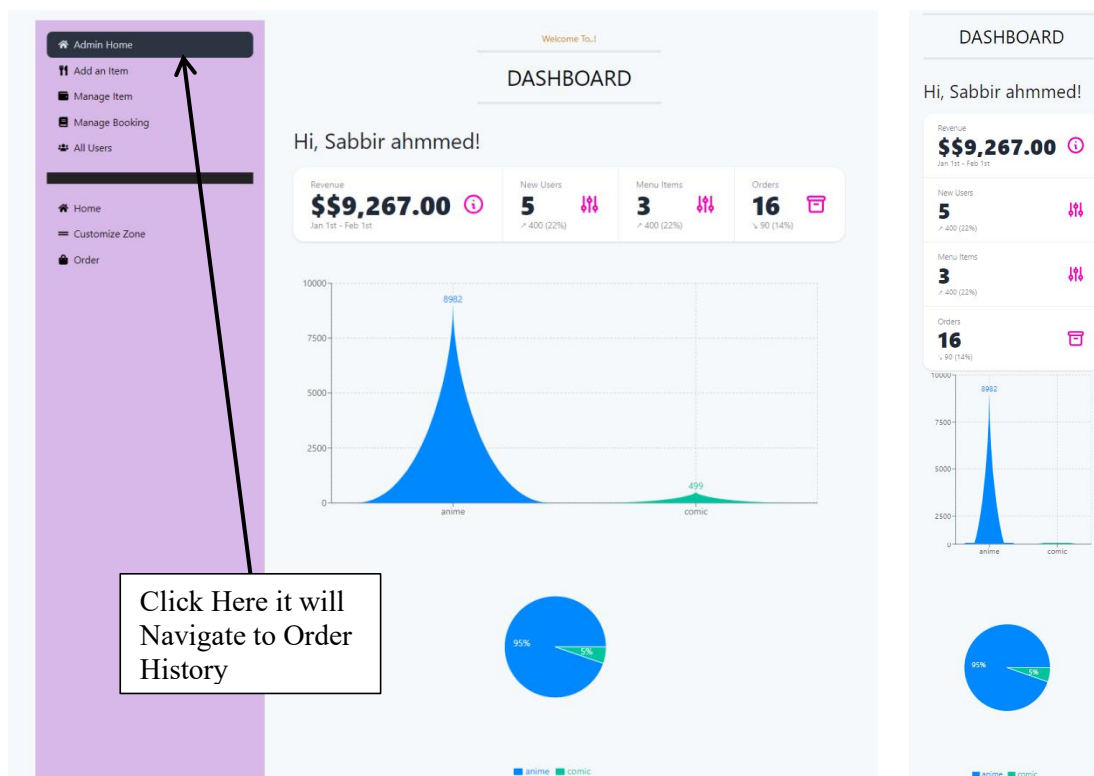
6.14 CUSTOMIZE TEE - User Order History

Figure 6.14 Responsive User Order History



6.15 CUSTOMIZE TEE - Admin Dashboard

Figure 6.15 Responsive Admin Dashboard



6.16 CUSTOMIZE TEE - Add Product

6.16 Responsive Add Product

The image displays two versions of the 'ADD AN ITEM' form side-by-side. The left version shows the form with empty input fields for Product Name, Category, Size, Price, Product Details, and Mockup. The right version shows the form with the same fields, but with a purple 'OPEN DRAWER' button at the bottom right.

The image shows a screenshot of the 'ADD AN ITEM' form with a callout box pointing to the 'ADD ITEM' button. The callout box contains the text 'Click Add Item Button to add Product'. The form fields are filled with the following data:

- Product Name*: Itachi Uchiha
- Category*: anime
- Size*: M,L,XL,XXL
- Price*: 499
- Product Details: 100% Cotton
- Mockup: CHOOSE FILE Untitled-6.png
- T-shirt Frontside Image: CHOOSE FILE Itachi Uchiha.png
- T-shirt Backside Image: CHOOSE FILE Itachi Uchiha.png

6.17 CUSTOMIZE TEE - Manage Product

Figure 6.17 Responsive Manage Product

The figure illustrates the 'Manage All Items' interface in three states:

- Top Screenshot:** Shows the 'Manage All Items' page with a table of products. The sidebar on the left includes 'Admin Home', 'Add an Item', 'Manage Item', 'Manage Booking', 'All Users', 'Home', 'Customize Zone', and 'Order'. The table lists three items: 'Basic Tee 6-Pack', 'Bas Tee 6-Pack', and 'Madara Uchiha', each with a price of \$499 and delete buttons.
- Middle Screenshot:** Shows a confirmation dialog box with the text 'Are you sure? You won't be able to revert this!' and two buttons: 'YES, DELETE IT!' and 'CANCEL'.
- Bottom Screenshot:** Shows the 'Manage All Items' page with a callout box containing the text 'Click Delete Button to Remove a Product then just confirm it'. An arrow points from the callout to the delete button of the first item in the table. Below the table is an 'OPEN DRAWER' button.

#	Item	Category	Price	Update	Delete
1.	Basic Tee 6-Pack	anime	\$499		
2.	Bas Tee 6-Pack	comic	\$499		
3.	Madara Uchiha	anime	\$499		

6.18 CUSTOMIZE TEE - Manage Order

Figure 6.18 Responsive Manage Order

#	Order Id	Order Item	Status	Update	Delete
1.	657f1cb7fec768f6289d07d0	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
2.	657f38bf7c6a465e500e74b5	Basic Tee 6-PackBasic Tee 6-PackMadara UchihaCustomizeMadara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
3.	657f5246fb3ff72f8eb76b97	Basic Tee 6-Pack	service pending	<input type="checkbox"/>	<input type="checkbox"/>
4.	657f52a8fb3ff72f8eb76b99	Bas Tee 6-Pack	service pending	<input type="checkbox"/>	<input type="checkbox"/>
5.	658d81457e987e9109bf6d73	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
6.	658daefb3b850797cf33a5de	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
7.	658dafa96087ee66e45331f1	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
8.	658db44a6087ee66e45331f3	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>

Click Delete or Edit Button to Update status or Reject order then Confirm it.

#	Order Id	Order Item	Status	Update	Delete
1.	657f1cb7fec768f6289d07d0	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
2.	657f38bf7c6a465e500e74b5	Basic Tee 6-PackBasic Tee 6-PackMadara UchihaCustomizeMadara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
3.	657f5246fb3ff72f8eb76b97	Basic Tee 6-Pack	service pending	<input type="checkbox"/>	<input type="checkbox"/>
4.	657f52a8fb3ff72f8eb76b99	Bas Tee 6-Pack	service pending	<input type="checkbox"/>	<input type="checkbox"/>
5.	658d81457e987e9109bf6d73	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
6.	658daefb3b850797cf33a5de	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
7.	658dafa96087ee66e45331f1	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
8.	658db44a6087ee66e45331f3	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>
10.	658db6856087ee66e45331f7	Madara Uchiha	service pending	<input type="checkbox"/>	<input type="checkbox"/>

Are you sure?
You won't be able to revert this!

YES, CONFIRM IT! **CANCEL**

6.19 CUSTOMIZE TEE - Manage User

Figure 6.19 Responsive Manage User

The screenshot shows a responsive web interface for managing users. On the left is a purple sidebar with navigation options: Admin Home, Add an Item, Manage Item, Manage Order, All Users (highlighted), Home, Customize Zone, and Order. The main content area is titled 'MANAGE ALL USERS' and displays 'Total Users: 5'. Below this is a table with 5 rows of user data. Each row has columns for #, Name, Email, Role, and Action. The Action column contains a red trash icon for deletion and an orange person icon for making a user an admin.

#	Name	Email	Role	Action
1.	Sabbir ahmmed	sahmmed974@gmail.com	admin	
2.	Boyz Zone	bozone007@gmail.com		
3.	Sabbir Ahmmed	sabbir35-2356@diu.edu.bd		
4.	uchihaobito	sa85497371@gmail.com		
5.	Itachi Uchiha	nintyninty@gmail.com		

This screenshot shows the same 'MANAGE ALL USERS' interface as above, but with a white confirmation dialog box overlaid in the center. The dialog box has a yellow warning icon and the text 'Are you sure? You won't be able to revert this!'. It contains two buttons: 'YES, CONFIRM IT!' (blue) and 'CANCEL' (red). A text box on the right says 'Click Delete or Make Admin Button to delete user or make admin and Confirm it.' with arrows pointing to the 'Make Admin' and 'Delete' buttons in the table below.

Are you sure?
You won't be able to revert this!

YES, CONFIRM IT! **CANCEL**

Click Delete or Make Admin Button to delete user or make admin and Confirm it.

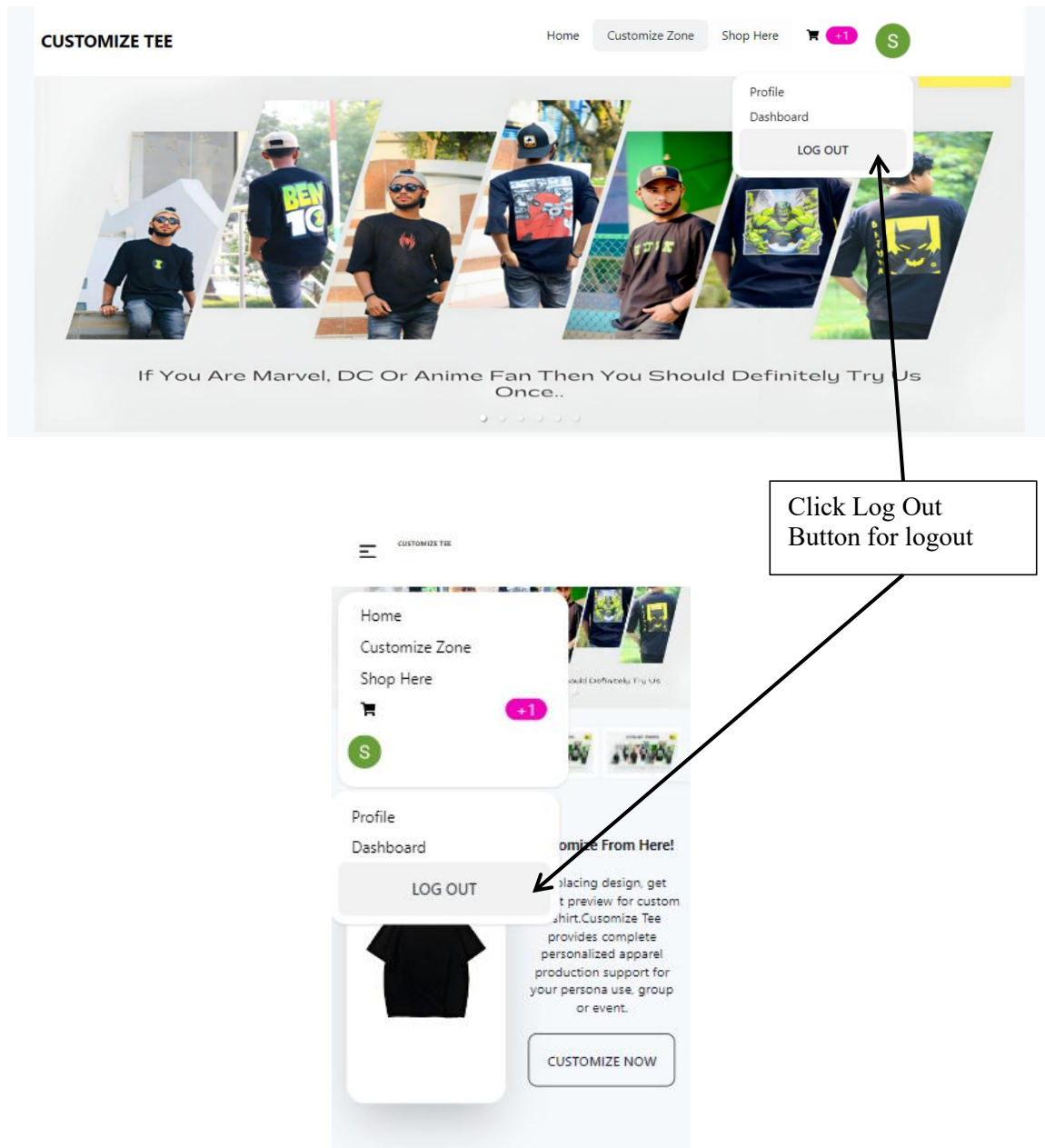
This screenshot shows the 'MANAGE ALL USERS' interface in a mobile view. The title 'MANAGE ALL USERS' and 'Total Users: 5' are at the top. Below is a table with 5 rows of user data, similar to the desktop view. At the bottom of the page is a blue button labeled 'OPEN DRAWER'.

#	Name	Email	Role	Action
1.	Sabbir ahmmed	sahmmed974@gmail.com	admin	
2.	Boyz Zone	bozone007@gmail.com		
3.	Sabbir Ahmmed	sabbir35-2356@diu.edu.bd		
4.	uchihaobito	sa85497371@gmail.com		
5.	Itachi Uchiha	nintyninty@gmail.com		

OPEN DRAWER

6.20 CUSTOMIZE TEE - Logout

Figure 6.20 Responsive Logout



CHAPTER 7

CONCLUSION

7.1 Project Summery

The Customize Tee project embarked on a mission to revolutionize the e-commerce landscape, with a particular focus on the dynamic realm of customized t-shirts. The central goal was to establish a dynamic and user-centric online platform dedicated to providing unparalleled personalization for customers. Simultaneously, the initiative aimed to equip administrators with efficient tools for seamless product management.

At the core of this project is a commitment to developing a comprehensive and efficient online marketplace for t-shirts, addressing the diverse requirements of our user base. Through the integration of user-friendly design tools, robust e-commerce features, and secure payment processing, the platform strives to deliver a seamless and enjoyable shopping experience. The overarching goal is to catalyze growth and success within the competitive t-shirt sales market. In essence, Customize Tee is not merely an e-commerce platform, it stands as a creative hub where individuality and fashion seamlessly converge. The intuitive interface, an array of customization tools, and streamlined order processing collectively redefine the landscape of personalized fashion. As we embark on this journey of innovation, we extend an invitation for you to join our community, where we continuously push boundaries, enrich the realm of custom apparel, and explore the limitless possibilities of print-on-demand services. Together, let's shape the future of personalized fashion.

This platform is more than just a marketplace for personalized t-shirts, it serves as a canvas for individuality, a space where technology and creativity converge to redefine fashion experiences. As we chart our course forward, the Customize Tee project envisions itself as a dynamic force, staying ahead of trends and setting new standards for personalized fashion in the digital age.

In conclusion, Customize Tee is not merely a project, it is an ongoing exploration, a commitment to innovation, and a celebration of diversity in style. We invite you to be part of this exciting journey, where technology meets self-expression, and together, we shape the future of fashion.

7.2 Limitations

While the Customize Tee project strives to deliver a seamless and innovative customization experience, it is essential to acknowledge certain limitations:

- Users are unable to cancel or modify orders after making a payment.
- Users cannot preview or download the invoice, they can only view certain details in the order history. In this demo project, the available payment options are limited.
- The current product range is limited, restricting choices for users.
- Customization tools may have a learning curve for some users.
- In the customization tool, the 3D mock-up of the t-shirt is not available.
- Only the front side mock-up of t-shirts is available in the customization tool.

7.3 Future Scope

The Customize Tee project has demonstrated significant potential in providing users with a unique and personalized shopping experience for t-shirts. However, several limitations have been identified during the development of this demo project. These limitations pave the way for future enhancements and expansions to elevate the platform to new heights. The following areas represent key aspects of the future scope for the Customize Tee project:

- Implement a feature allowing users to cancel or modify orders even after making a payment. This ensures a more flexible and user-friendly shopping experience.
- Introduce functionality for users to preview and download detailed invoices. Providing comprehensive invoices enhances transparency and allows users to have a record of their transactions.
- Expand the available payment options to cater to a broader user base. Integrating additional payment gateways will accommodate diverse preferences and regional methods.
- Increase the product range by adding more t-shirt designs, styles, and categories. A diverse product catalog will provide users with a wider array of choices, encouraging engagement.
- Simplify and enhance the customization tools to reduce the learning curve for users. An intuitive interface and user-friendly design will make the customization process more accessible to a wider audience.
- Integrate a 3D mock-up feature in the customization tool. This enhancement allows users to visualize their customized t-shirts from multiple angles, providing a more immersive and realistic preview.

- Extend the customization tool to display both front and back sides of the t-shirt in the mock-up. This feature enhances the customization experience by offering a complete view of the designed product.

This project holds potential future opportunities that could prove beneficial. The future scope of the Customize Tee project revolves around addressing the identified limitations and further refining the platform to deliver an enriched and seamless user experience. By incorporating these enhancements, the project aims to stay at the forefront of personalized fashion, providing users with an innovative and satisfying shopping journey.

7.4 Project Link

APPENDICES

A. Literature Review

A.1 Existing Solutions

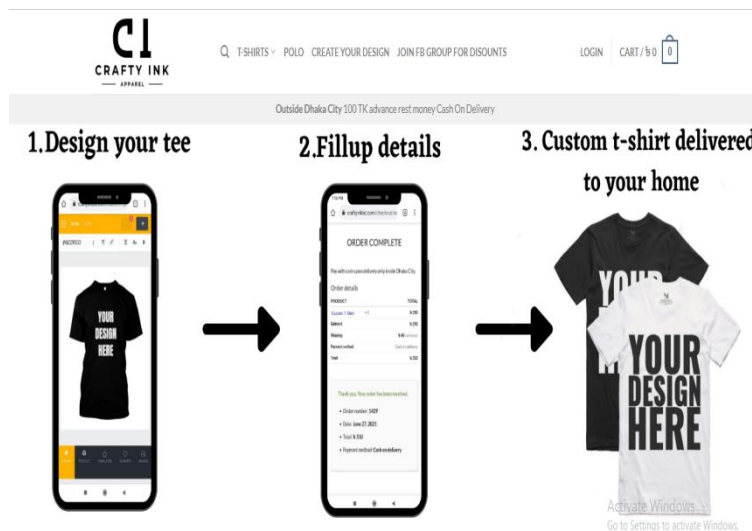
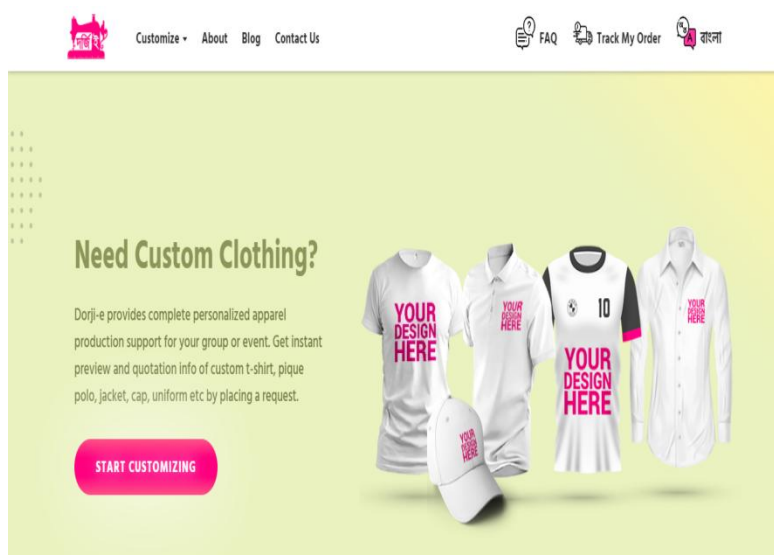
Dorji-e

- No way for adding Own Design
- Design placement issue

Crafty Ink

- No way for adding Own Design
- Hard to use customize section

Figure 8.1 Dorji-e & Crafty Ink



B Code Snippets

B.1 Front-end Development

The front-end development process of the "Customize Tee" project played a pivotal role in shaping user interactions and the overall functionality of the platform. Leveraging the React library and the utility-first CSS framework Tailwind CSS, we aimed to create a dynamic and visually appealing online platform.

1. Design Implementation & Tailwind CSS Configuration

Tailwind CSS, a utility-first CSS framework, played a central role in simplifying the styling process and maintaining clean, scalable code. DaisyUI Plugin Integrated for additional UI components and theme management. The configuration file, as shown below, provided customization options for themes and extended default settings.

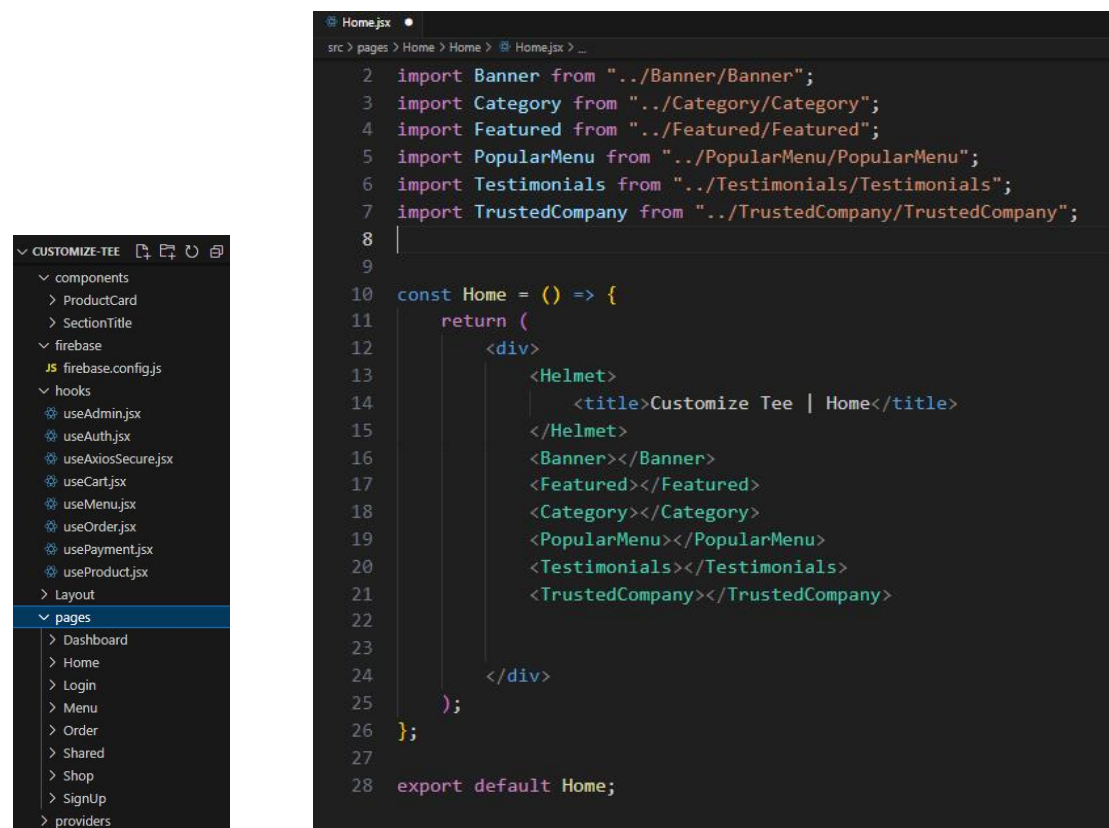
Figure 8.2.1 Tailwind CSS Configuration

```
JS tailwind.config.js x
JS tailwind.config.js > [default] > daisyui
1  /** @type {import('tailwindcss').Config} */
2  export default {
3    content: [
4      "./index.html",
5      "./src/**/*.{js,ts,jsx,tsx}",
6    ],
7    theme: {
8      extend: {},
9    },
10
11   daisyui: {
12     themes: ["light"],
13   },
14   plugins: [
15     require("daisyui"),
16     require('@tailwindcss/aspect-ratio'),
17   ],
18 }
```

2. React Component Creation

Frontend development for the "Customize Tee" project centers on the use of React components to shape the user interface. The website's structure benefits from React's component-based architecture, allowing seamless organization and management of various elements. Each component encapsulates specific functionalities, such as headers, product displays, and checkout forms, contributing to an intuitive and cohesive user experience.

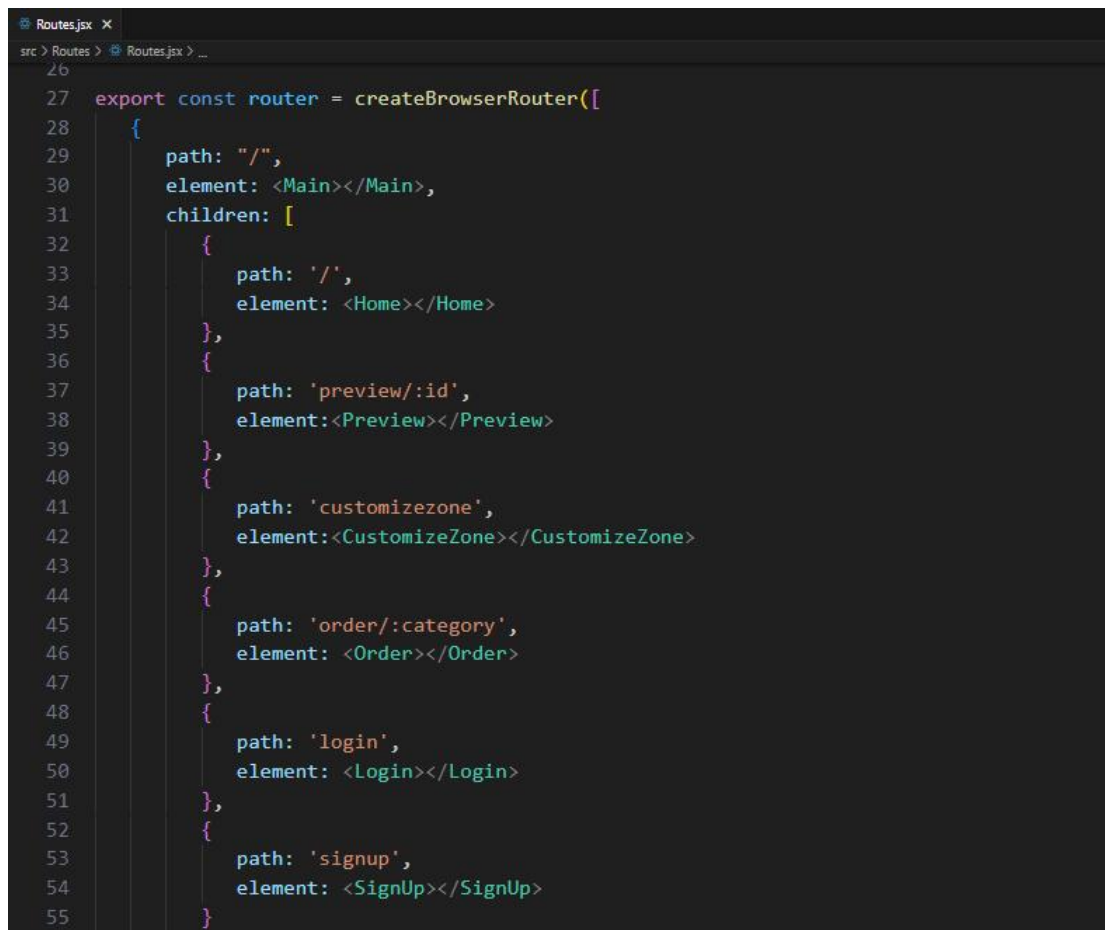
Figure 8.2.2 React Components



3. React Routing

React Router serves as a widely adopted solution for managing client-side routing within React applications. Its purpose is to facilitate navigation between distinct sections of the site without requiring a full page reload. Utilizing routes allows users to seamlessly transition from one part of the website to another, providing a user experience akin to a conventional multi-page application. However, in reality, it operates as a single-page application.

Figure 8.2.3 Routes



```
27 export const router = createBrowserRouter([
28   {
29     path: "/",
30     element: <Main></Main>,
31     children: [
32       {
33         path: '/',
34         element: <Home></Home>
35       },
36       {
37         path: 'preview/:id',
38         element: <Preview></Preview>
39       },
40       {
41         path: 'customizezone',
42         element: <CustomizeZone></CustomizeZone>
43       },
44       {
45         path: 'order/:category',
46         element: <Order></Order>
47       },
48       {
49         path: 'login',
50         element: <Login></Login>
51       },
52       {
53         path: 'signup',
54         element: <SignUp></SignUp>
55       }
56     ]
57   }
58 ])
```

4. Integration of API

The front-end retrieves information from the back-end server through asynchronous HTTP calls, employing tools such as Axios for this purpose. API integration encompasses obtaining diverse data types, including details about products, user identities, order specifics, and more. When a visitor accesses the site, product descriptions are retrieved from the back-end database and presented on the front-end

Back-end Development.

Figure 8.2.4 Integration of API

```
useCart.jsx X
src > hooks > useCart.jsx > useCart
1 import { useQuery } from '@tanstack/react-query'
2 import useAxiosSecure from './useAxiosSecure';
3 import useAuth from './useAuth';
4
5 const useCart = () => {
6   const { user, loading } = useAuth();
7   const [axiosSecure] = useAxiosSecure()
8   // const token = localStorage.getItem('access-token')
9   const { refetch, data: cart = [] } = useQuery({
10    queryKey: ['carts', user?.email],
11    enabled: !loading,
12    queryFn: async () => {
13      const res = await axiosSecure(`/carts?email=${user?.email}`)
14      // console.log('res from axios', res)
15      return res.data;
16    },
17  })
18
19  return [cart, refetch]
20
21 }
22 export default useCart;
```

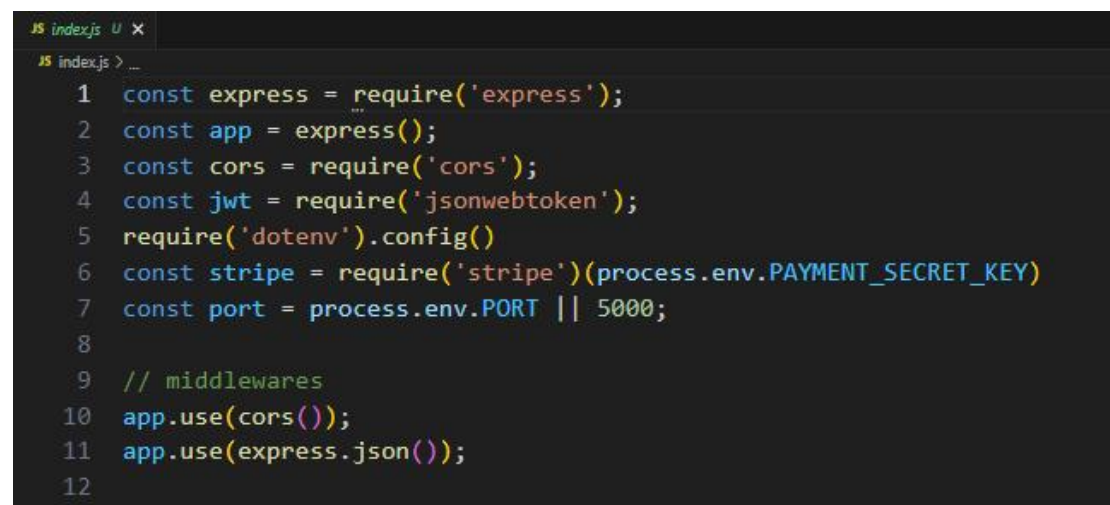
B.2 Back-end Development

The back-end serves as the foundation of the platform, handling crucial functionalities such as user authentication, data management, API endpoints, and component interactions. MongoDB serves as a robust database for storing essential data like user profiles, product details, orders, and reviews. Security measures, including JWT-based user verification, are implemented to safeguard user accounts and financial transactions. Here is an overview of the back-end development process for the "Customize Tee" project, highlighting key components and processes

1. Server Setup

The back-end server setup commenced with the utilization of Node.js and Express.js, establishing a framework to manage incoming HTTP requests and responses.

Figure 8.2.5 Server Setup



```
JS index.js U x
JS index.js > ...
1  const express = require('express');
2  const app = express();
3  const cors = require('cors');
4  const jwt = require('jsonwebtoken');
5  require('dotenv').config()
6  const stripe = require('stripe')(process.env.PAYMENT_SECRET_KEY)
7  const port = process.env.PORT || 5000;
8
9  // middlewares
10 app.use(cors());
11 app.use(express.json());
12
```

2. Database Integration

MongoDB serves as the chosen database system for storing versatile data, encompassing information about products, users, orders, reviews, and more. Mongoose, an Object Data Modeling (ODM) module, is employed to facilitate communication between the program and the MongoDB back-end.

Figure 8.2.5 Database Integration

```
index.js | x
index.js | run | app.get('/users/email') callback
37
38 const { MongoClient, ServerApiVersion, ObjectId } = require('mongodb');
39 const uri = `mongodb+srv://${process.env.DB_USER}:${process.env.DB_PASS}@cluster0.e0e015q.mongodb.net/?retryWrites=true&
w=majority`;
40
41 // Create a MongoClient with a MongoClientOptions object to set the Stable API version
42 const client = new MongoClient(uri, {
43   serverApi: {
44     version: ServerApiVersion.v1,
45     strict: true,
46     deprecationErrors: true,
47   }
48 });
49
```

3. API Endpoints Creation

The integration between the "Customize Tee" website's front-end and back-end server is established through API endpoints. These endpoints define the paths through which the front-end communicates with the back-end to perform specific actions or retrieve data. Each API route is associated with a unique URL path and an HTTP method (GET, POST, PUT, or DELETE) that corresponds to the intended action.

Figure 8.2.6 API Endpoints Creation

```
index.js | x
index.js | run | app.get('/users/email') callback
177
178 // products apis
179 app.get('/products', async (req, res) => {
180   const result = await productCollection.find().toArray();
181   res.send(result);
182 })
183 app.get('/products/:id', async (req, res) => {
184   const productId = req.params.id;
185
186   try {
187     const result = await productCollection.findOne({ _id: new ObjectId(productId) });
188
189     if (result) {
190       res.json(result);
191     } else {
192       res.status(404).json({ message: 'Product not found' });
193     }
194   } catch (error) {
195     console.error('Error fetching product details by ID:', error);
196     res.status(500).json({ message: 'Internal server error' });
197   }
198 });
199 app.delete('/products/:id', verifyJWT, verifyAdmin, async (req, res) => {
200   const id = req.params.id;
201   const query = { _id: new ObjectId(id) }
202   const result = await productCollection.deleteOne(query);
203   res.send(result);
204 })
205
```


4. User Authentication and Authorization

Authentication in the "Customize Tee" project is implemented using JSON Web Tokens (JWT) to ensure secure management of user sessions and safeguard login processes. User roles and permissions are configured to grant appropriate access rights, allowing admins to manage their products and users to view their orders. This system ensures a secure and controlled user experience on the platform.

Figure 8.2.7 User Authentication and Authorization

```
AuthProvider.jsx x
src > providers > AuthProvider.jsx > ...
44
45  useEffect(() => {
46    const unsubscribe = onAuthStateChanged(auth, currentUser => {
47      setUser(currentUser);
48      console.log('current user', currentUser);
49
50      // get and set token
51      if (currentUser){
52        axios.post('http://localhost:5000/jwt', { email: currentUser.email })
53          .then(data => {
54            // console.log(data.data.token)
55            localStorage.setItem('access-token', data.data.token)
56            setloading(false);
57          })
58      }
59      else {
60        localStorage.removeItem('access-token')
61      }
62
63    });
64    return () => {
65      return unsubscribe();
66    }
67  }, []);
68
69
```

REFERENCES

Dorjie. (n.d.). Retrieved December 30, 2023, from <https://dorjie.com/>

Crafty Ink BD. (n.d.). Retrieved December 30, 2023, from <https://craftyinkbd.com/>

Visual Studio Code. (2015, April 29). Retrieved December 30, 2023, from <https://code.visualstudio.com/>

GitHub. (2008). Retrieved December 30, 2023, from <https://github.com/>

Tailwind CSS. (n.d.). Retrieved December 30, 2023, from <https://tailwindcss.com/>

React. (2013, May 29). Retrieved December 30, 2023, from <https://react.dev/>

MongoDB Atlas. (2009, February 11). Retrieved December 30, 2023, from <https://www.mongodb.com/atlas/database>

Node.js. (2009, May 27). Retrieved December 30, 2023, from <https://nodejs.org/en>

Express.js. (2010, November 16). Retrieved December 30, 2023, from <https://expressjs.com/>

Polotno. (n.d.). Retrieved December 30, 2023, from <https://polotno.com/>