THE PROJECT TITLE

TopOne.com

Submitted By

Md Jahid Hassan

(Id: 171-35-1976)

Supervised By

Ms. Tapushe Rabaya Toma

Lecturer (Senior scale)

Faculty of Science and Information Technology

Department of Software engineering

Daffodil International University

Fall-2020

A project (SWE-431) submitted in fulfillment of

The Requirements for the degree of BSc in

Software Engineering

APPROVAL

This thesis/project/internship titled on “Title”, submitted by Student_Name, ID: 172-35-abc 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

Imran

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

Chairman

Md Anwar Hossen
Assistant Professor
Department of Software Engineering
Daffodil International University

Internal Examiner 1

Shakir

Asif Khan Shakir
Senior Lecturer
Department of Software Engineering
Daffodil International University

Internal Examiner 2

Professor Dr M Shamim Kaiser
Institute of Information Technology
Jahangirnagar University

External Examiner

DECLARATION

I am the author; hereby declare that the project becomes the property of Daffodil International University (DIU). I give permission to the DIU that they can change in whole or in part for the purpose of research or academic exchange only.

Supervised by

23.06.21

Ms. Tapushe Rabaya Toma
Lecturer (Senior Scale)
Daffodil International University
Department of Software Engineering

Submitted by

23/06/2021

Md. Jahid Hassan
ID: 171-35-1976
Daffodil International University
Department of Software Engineering

ACKNOWLEDGEMENT

Firstly, I want to thank the almighty Allah who give me the opportunity to complete the project by keeping me continuously healthy in terms of physically and mentally, I also want to thank my parents who are always inspiring me to continue my studies as well as the complex the project during the difficult situations in covid-19. I want to thank all of the teacher who thought me the achieve the required skills to complete this project .Basically my supervisor “MS. Tapushe Rabaya Toma” she also help me to make my project perfectly and give my mistake in time so for that it is possible to make my project in time.

PROJECT SUMMARY

This is a web base application and I only develop this web application for business purpose. This is “TopOne.com”. In this project I implement for customer satisfaction like easy to order for getting good service and quality full product provide. In this case it is very essential web application for life easier. Every kind of class people can use this web application/website for buy their choice-able product .This is a clothes base website/ web application. All kinds of people can buy their needed clothes from our online store. Customer can visit website and can order product and add to cart with give them some additional information and location with fill up registration form then they need to login and checkout. This case customer can save their time and suffering. Usefulness to visit my web application there are a lot of product they will get. Customer can customize design and they will get this after fulfill up their requirement.so I assume that this will very important roll to go ahead one step in digital world.

Table of Contents

1. CHAPTER-01 INTRODUCTION

1.1 Project Overview

This web application is very effective now digital world. Totally this is developing for business purpose. Here customer must be satisfied because of this there are a lot of products for them. Customer can visit this website/web application and then they choice their best product and they need to add cart. If this is first time to add product then they need to registration with their email and they need to login before add to cart and then they can easily add their product into their cart and they can show price of their cart. After adding product to cart and they need to pay. There are some way to pay their price one is cash on delivery and another is online pay method. Only they can choice one way at a time. If they want to pay online they need to select online pay system then come one interface of website then the select to mobile banking system then they can pay money by bkash mobile banking system. After this their product will store and Admin will control and provide clearance by email then customer will get one invoice and after a few day customer will get their product.

1.2 Project Overview

1.2.1 Background

This web application developed for customer satisfaction and good quality full product provide. Here product will show and customer can easily buy to stay home and can save their important time. The main theme of this is getting best product without hassle.

1.2.2 Usefulness and Benefits

- Customer easily choice product
- Hassle free product
- Quality full products
- Can find necessary product
- Can online pay if they
- Multiple product add to cart

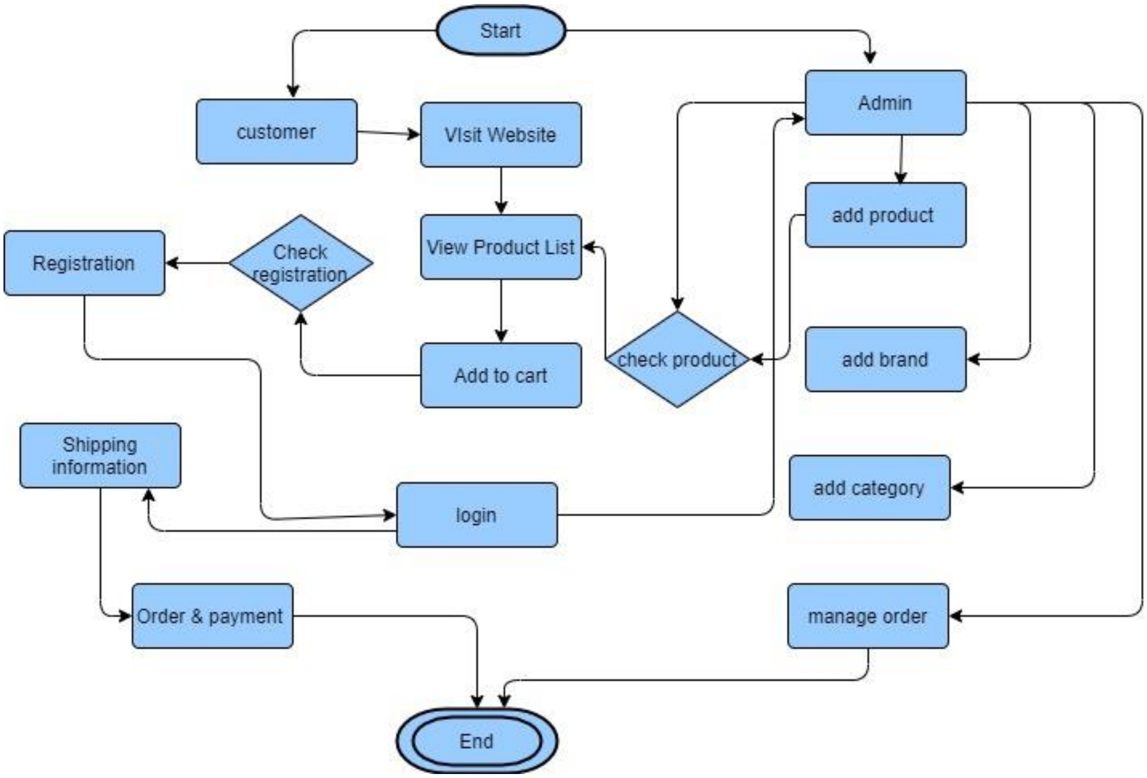
1.2.3 Goals

- Hassle free order
- Product details check
- Provide low cost
- Easily choice product
- Maintain time to services
- Multiple product add to cart

1.3 Stockholders

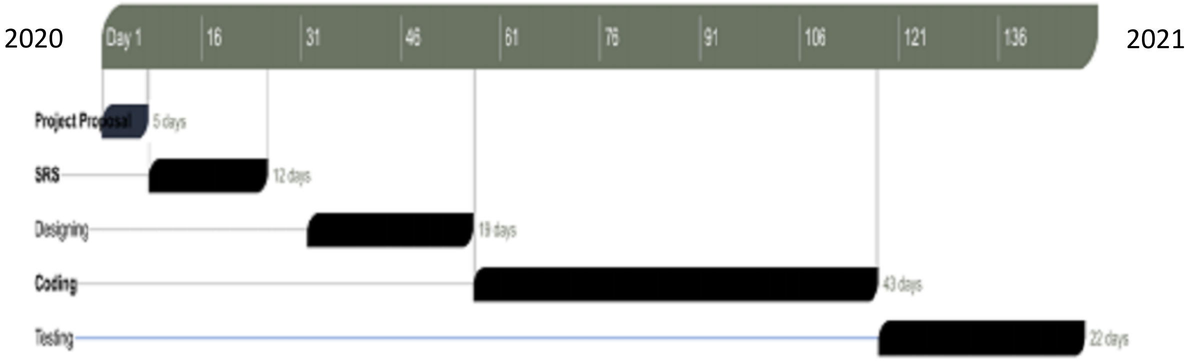
- Customer
- Admin
- Guest

1.4 The Model of Proposed System (Block Diagram)



1.5 Project Schedule

1.5.1 Gant Chart



2. CHAPTER-02 SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requirements

Here I notice all of Functional Requirements below my web application

Table 2.1 list down Functional Requirements

ID	Name	Description	
FRQ-1	**Can visit website**	This website is open field here any kind of people and parson can visit. Here no need to any kind of hassle to visit. No term and condition to visit.	Customer Admin Guest
FRQ-2	**Add cart**	All kind of user can use this but before add this cart customer must be visit this website and finally thus are able to add to cart	Customer
FRQ-3	**Remove cart**	Another functional requirements this. It is very important cause without any customer cannot remove cart before this customer must be add minimum product to his cart then he can delete /remove cart	Customer
FRQ-4	**Update cart**	A customer are can update this product after add any one product into his cart he think that he need to another product choice able to get this another then can update and he can increase his quantity to one to many. So this is	Customer

		very easy to add this one.	
FRQ-5	**Shipping**	Shipping is another important functional requirement for a customer. If any customer wants to shipping his product he must fill up a requirement field to get shipping/ order without fill up this form any customer cannot shipping.	Customer
FRQ-6	**Order**	This is the main theme of here .it is mandatory for a customer to get any product to his on location. First customer visit my web site to find his choice-able product from here and then he think that need for him then he order it by giving some information to get it. It is big part of my project because it major part of customer part.	Customer
FRQ-7	**View product list**	Every one of the world can visit my website to get their own choice. If any customer wants they can view my site any time in a week, month and year daily.	Customer Guest
FRQ-8	**View product details**	After view product list. If anyone interested to view details and they can	Customer Guest

		do this job. To visit my site and view product list anybody no need to give any kind of information to view this.	
FRQ-9	**payment**	Payment is the big part of my project. If any customer wants to buy product he must be pay for this order. Without payment clearly customer never will get any kind of product. If there any kind of customer buy multiple product for our sit then must need to order product then the need to checkout and then give some shipping information to get order product before payment. After complete order.	Customer
FRQ-10	**Registration**	Registration is a common part of my site. When customer add product to cart he need to registration. Here he will fill up customer name, customer email, and customer set a password and then the mobile number, address.	Customer Admin
FRQ-11	**Login**	Customer must be registration and then can go another stapes for complete	Customer Admin

		his order. So he also remembers user name or email and password.	
FRQ-12	**Can add category**	It is very important part for an admin portion because a admin can add category to view customer site product category add so that an admin must be add category to attractive and quality full product to view page.	Admin
FRQ-13	**Can add manage category**	Here a admin can manage add category because after period admin will add new category and some old category will remove so that's why admin need to manage category to add.	Admin
FRQ-14	**Can brand add**	It is very important part for an admin portion because a admin can add brand to view customer site product brand category add so that an admin must be add brand category to get new brand product to choice-able to customer site.	Admin
FRQ-15	**Manage order**	Admin only can manage order. Admin must login this or dashboard	Admin

		and then he can manage order delete, update, edit and invoice of this manage order part. So this is very important for an admin.	
FRQ-16	**Post new product**	Customer can view thus kind of product list, category and brand all kind of work done by admin portion of new product portion site. Here admin fill up product category, brand, product code, product name, product price, stock amount, short description, long description, image, sub-image etc.	Admin
FRQ-17	**Can see order details**	Here admin can see order details for this customer site. Which customer place orders their product? Thus customer information can see admin of customer order details.	Admin
FRQ-18	**can update product**	It also another part of admin here admin can update product. Such as at the first time admin up load product but is price will increase so that he can add easily update product.	Admin
FRQ-19	**Edit product info**	After uploading product any time admin can change	Admin

		such as he can mistake to give input null bull or missing then he can edit all kind of information.	
FRQ-20	**Download invoice**	There are two kind of invoice report generate of this one is admin and one is customer.	Customer Admin
FRQ-21	**Manage brand**	Admin can manage also their brand. Update, delete and edit.	Admin

2.2 Functional Requirements

Here I notice all of Nonfunctional requirements below my web application

Table 2.2 list down Non-Functional Requirements

ID	Name	Description	Non-Functional Requirements	Priority
NFRQ-1	Availability	This availability is seven day and twenty four hour.	Non-Functional	High
NFRQ-1	Accuracy	Here is highly accuracy. Developer mode and systematic implemented so that is accuracy must concern.	Non-Functional	Medium
NFRQ-1	Security	It is very security ensure application because I implement this is very latest framework authentication so that is secure 100%	Non-Functional	High
NFRQ-1	maintenance	Easily user can use it and maintaining section is more comfortable for maintain that’s reason it is very user friendly.	Non-Functional	Medium

2.3 Performance Requirements

2.3.1 Speed and Latency Requirements

- Her dataset and migration file are well structure into MYSQL and LARAVWL php artisan command form with migration maintained
- I also try to marge larravel versioning controls with migration file so that I sometimes roll back migration file

2.3.2 Precision and Accuracy Requirements

- This is more and more accuracy maintains my full concern for developer mood.
- Customer easily can change their registration form shipping information accurately.

2.3.3 Capacity Requirements

- This web application has a lot of ability to accurate data serve into server and gain full control Customer.
- Admin can give or add product then customer can easily show view page and can order capacity high of customer.

2.4 Dependability Requirements

2.4.1 Reliability Requirements

- When customer will order he must registration add login to add card and additional information.
- Customer want to complete order he must be shipping information
- Also need to customer payment to add any product into cart
- If any product and any category want to add Admin must be login into his dashboard and must be adding this item.

2.4.2 Amiability Requirements

- Goal to interactive xampp server for the exchanging data to database and handling the request and sent back the customer.
- Any time authorized user can access system module and can change all kind of information
- It can use chrome and Microsoft edge

2.5 Maintainability and Supportability Requirements

2.5.1 Maintenance Requirements

- Here authority user or admin will handle this system.
- Except the authority no one can access of this system
- Only can maintain admin to all kind of product section panel
- No one can change the privacy access without admin

2.5.2 Supportability Requirements

- If customer fetch any kind of problem they can email or alternative way to inform to us
- After someday cross they our service provide will check all kind of problem issue and try to solve

2.5.3 Adaptability Requirements

A customer can order any kind of product so this is not difficult for them to buy this product. But customer can solve their all kind of difficulty to informing admin site.

2.5.4 Scalability or Extensibility Requirements

- Admin has been must be consider by the side of developer panel.
- This kind of developed by technical team of expert level people. They have the suggestions must be ensure scalability module of this admin system.

2.6 Security Requirements

2.6.1 Access Requirements

- Without Authenticated user no one cannot enter their panel
- Customer and Gest type customer can visit web application and product details
- Customer can registration other wish any kind user cannot registration and no allow to registration
- Admin user can only allow to all kind of access.

2.6.2 Integrity Requirements

Here system has been navigated by customer, guest and admin

2.6.3 Privacy Requirements

This system is sensitive information are control by the admin and also protected. Any kind of user and security specialist cannot able to inter into admin panel. Third party cannot enter our site.

2.7 Usability and Human Interaction Requirements

Table 2.3 Usability and Human Interaction Requirements

UH-01	**User Friendly**
Description	This web application is user friendly of easiest features.
Stakeholders	Customer, Guest, Admin

2.7.1 Ease of Use Requirements

The system is easy to product collect and products add for admin and customer.

Table 2.4 Ease of Use Requirements

UH-02	The User sit must be usable for customer and Guest site.
Description	The system indicates the several possibilities customer and admin and guest
Stakeholders	Customer, Guest, Admin

2.7.2 Understand Ability and Politeness Requirements

Table 2.5 Understand Ability and Politeness Requirements

UH-03	This system has made only for local customer of Bangladesh
Description	Since, the local customer can try best to make system more understandable. I implement the payment get call ssl Commerce for thinking about paymet method online payment menthe and bkash mobile banking it is very easy to use and collaborate.
Stakeholders	Customer, Guest, Admin

2.7.3 Understand Ability and Politeness Requirements

- Login customer before order
- Registration customer if before no registration
- Login Admin before dashboard
- Select must be payment method before success order

2.7.4 User Documentation Requirements

Table 2.6 User Documentation Requirements

UH-03	This system has made only developer documentation particularly
Description	Build up this project there is particularly requirement of applicant documentation. Here used some terms and condition to create project documentation.
Stakeholders	Admin developer

2.8 Look and Feel Requirements

Here all clear that customer can order complete to give their major information before payment. On the other side admin must be login and add category, brand and product also. So that every customer can find their necessary product in time to view my website and easily can order product.

2.8.1 Appearance Requirements

This is clear to Admin, Customer and Guest to find their own activity as like buy product, pay product price and clearance. On the other had admin can add product his portion.

Table 2.7 Appearance Requirements

LF-01	Labels of All field are very sensitive and bold with font "time new roman".
Description	There are many kind of input field as like order product, shipping information and add information. And another is product add form her many kind of filed must be fill up an admin.
Stakeholders	Customer, Guest, Admin

2.8.2 Style Requirements

Table 2.8 Style Requirements

LF-02	Here I implement this project separately and high priorities when focusing on design using css and bootstrap
Description	To build up this web application following some instruct because of I need to some instruction as like frame work structure css-4 and bootstrap-4 for this user interface view very well and look like structured.
Stakeholders	Customer, Admin

2.9 Operational and Environmental Requirements

This all requirements are focus on Customer, Admin and Guest how they are actually rules and how they are able to satisfy of their own site. I want to tall again that this is a web application to provide customer quality full product and successfully sell product is admin main goal.

2.9.1 Expected Physical Requirements

This is actually not physically visible but there are some kind of requirements as like to physically when customer order product admin can view this who order product and then admin check customer status and clear customer status and then customer can view his order totally completed and give admin access to customer download invoice and another invoice store admin to delivery report attachment to delivery.

2.10 Legal Requirements

This web application requirements has no violence of rules because of there are some kind of input field and accessibility so that any third party cannot inter and violence Customer can order and admin can add product so this is secure for both side and easily can perform.

2.10.1Compliance Requirements

I think There are no satisfaction to give any kind of compliance requirements because of customer will must be satisfaction to new product and site did not load a log time because I use here latest all kind of framework and php version.

2.10.2 Standard Requirements

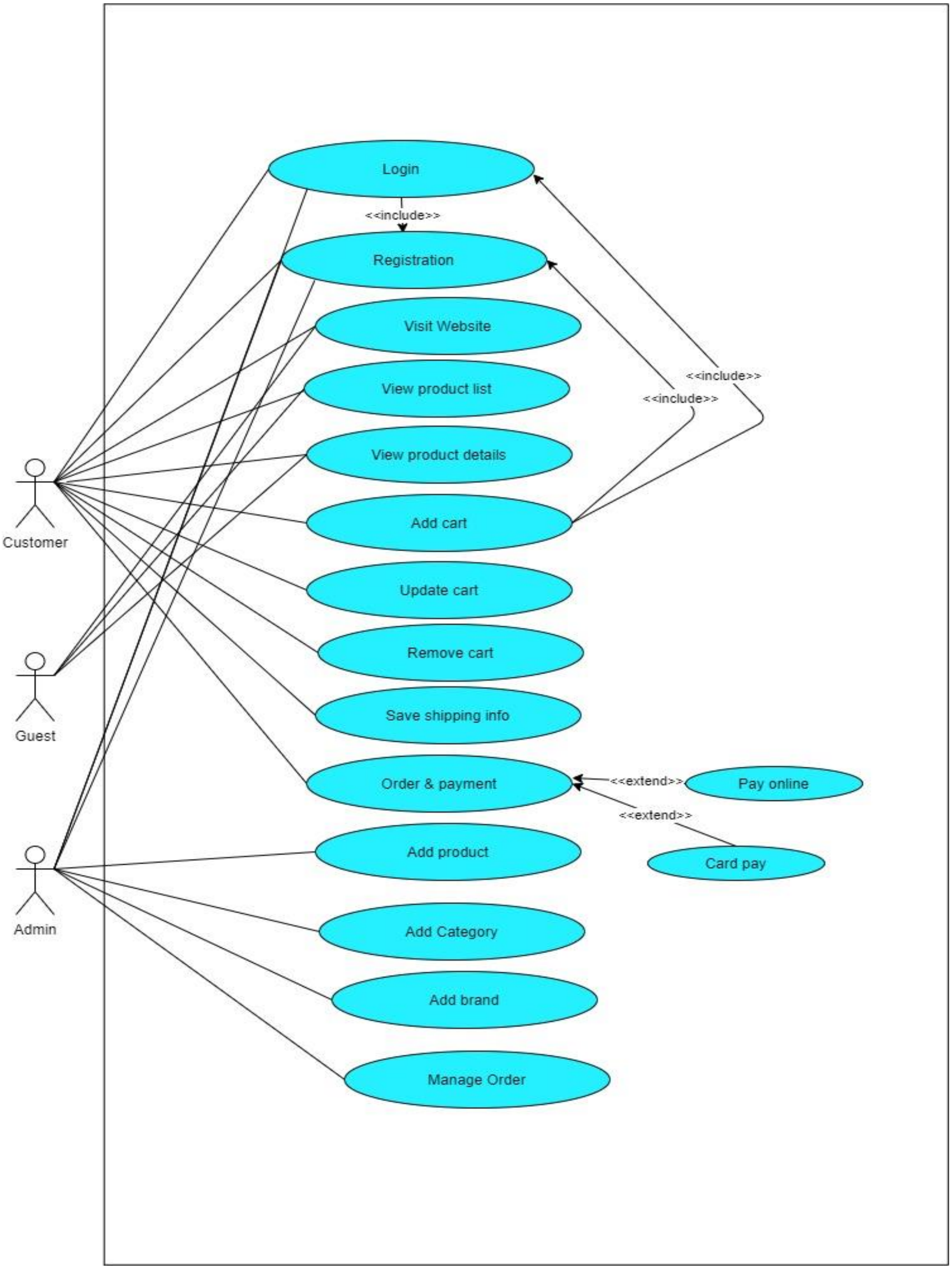
Here some new and standard requirements.

Table 2.9 Standard Requirements

LF-01	To add order into cart and payment for customer
Description	This is considered as public website as like Arrong, Easy all kind of brand.
Stockholders	Customer

3. CHAPTER-03 SYSTEM ANALYSIS

3.1 Use Case



3.2 Use Case Description

3.2.1 Login

Table 3.1 Login

Use case	Login	
Goal <a longer statement of the goal in context if needed >	Customer & admin cannot enter system without login that reason admin and customer must be login to inter into system for admin and buy product to customer.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	Customer can order to give login and admin can login system successfully. After that they perform their actions as expected successfully.	
Failed End Condition <the state of the world if goal abandoned >	If customer is getting fail they cannot enter order and perform fail their action as accepted successfully.	
Primary Action	Customer, Guest. Admin	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	Login request come and can login	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	After giving all input field as like email and click on login button then action will successfully done to order page.
	2	Wrong email the wrong password this submission button redirect to login page again. So that we need to give accurate email and password.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	The customer gives their credentials including email & password. Store from Post request process into encrypted process.
Quality Requirements	Step	Requirement
	1	Requirements are store email and password completion the full process.

3.2.2 Registration

Table 3.2 Registration

Use case	Registration	
Goal <a longer statement of the goal in context if needed >	Customer & admin cannot enter system without registration that reason customer must be registration to inter into system Successfully.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	Customer before order must need to registration then can possible to add to cart and able to enter the system successfully. After that they perform their actions as expected successfully.	
Failed End Condition <the state of the world if goal abandoned >	If customer is getting fail they cannot enter order or shipping and perform fail their action as accepted successfully.	
Primary Action	Customer	
Secondary Actor	Admin	
Trigger<a longer statement of the goal in context if needed >	Registration request through and registration come in	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	After giving all input field as like email, name, mobile, address, password set and click on login button then action will successfully done to order page.
	2	Wrong type data submission button redirect to registration page again. So give valid data .
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	**Step**	**Branching action**
	a1	The customer gives their credentials including accurate data. Store from Post request process into database.
Quality Requirements	**Step**	Requirement
	1	Requirements are store input field data completion the full process.

3.2.3 Visit Website

Table 3.3 Visit Website

Use case	Visit Website	
Goal <a longer statement of the goal in context if needed >	Customer, Guest can visit website to without any kind of hassle no suffering. Search the website and easily can visit and view product.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	Customer, Guest can visit web site easily and successfully. After that they perform their actions as expected successfully.	
Failed End Condition <the state of the world if goal abandoned >	If customer is getting fail to find out web site they cannot enter or cannot visit web site. Order and perform fail their action as accepted successfully.	
Primary Action	Customer, Guest	
Secondary Actor	Admin	
Trigger<a longer statement of the goal in context if needed >	Throw Url and possible to product list and visit web site	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Search to browser and easily can visit successfully into here
	2	Wrong searching and find key word will not possible to find out my site.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	The customer and guest Searching and finding url process then they are able to find web site.
Quality Requirements	Step	Requirement
	1	Must be search right key word then it is possible to find our page or web site.

3.2.4 View Product List

Table 3.4 View Product list

Use case	View Product list	
Goal <a longer statement of the goal in context if needed >	Customer, Guest can visit website to without any kind of hassle no suffering. Then they can view all kind of product list with category wise.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	Customer and guest can view all product list of this system successfully. After that they perform their actions as expected successfully.	
Failed End Condition <the state of the world if goal abandoned >	If customer is wrong input type they will fail they view product list and fail their action as accepted successfully.	
Primary Action	Customer, Guest	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	Accurate instruction is possible to view product list	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	After finding category wise or list of view they can successfully view product list and can order.
	2	Wrong process of the request and selection the customer must be fail 100% successfully
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	If customer give their system base selection they must successful to view product list and can get any kind of order submission
Quality Requirements	Step	Requirement
	1	He must conscious about field structure add can view product list.

3.2.5 View Product Details

Table 3.5 View Product Details

Use case	View Product Details	
Goal <a longer statement of the goal in context if needed >	Customer and Guest all kind of access of the frontend design or product list and product details of this site. Customer can any thin to start to check product list view and details view of this system	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	When customer check product quality and details he must be select specific product of list then he can successfully go to details page.	
Failed End Condition <the state of the world if goal abandoned >	If customer and guest not select see product view he cannot show product details view he must be fail successfully	
Primary Action	Customer, Guest	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	Customer will through order page to order product	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Customer want to view product details view he must a button request post data then they are able to successful.
	2	Wrong selection of product details view customer cannot view product detail view so that he cannot all thinks of product appropriately
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	If customer give their system base selection they must successful to view product list and can get any kind of order submission
Quality Requirements	Step	Requirement
	1	Must be select stapes to view product list toughly

3.2.6 Add Cart

Table 3.6 Add cart

Use case	Add cart	
Goal <a longer statement of the goal in context if needed >	A customer can easily order. Only he needs to give some information into shipping section.	
Preconditions <what we expect is already the state of the words >	Shipping information	
Success End Condition <the state of the world upon successfully complete >	After viewing product any customer can select this choice-able product to his cart and he is able to add product into his card and she can change his cart quantity successfully.	
Failed End Condition <the state of the world if goal abandoned >	If he fail to add to cart he is not to able his product and again to try for fail	
Primary Action	Customer	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	Redirect to shipping page	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Viewing product any customer can select this choice-able product to his cart and he is able to add product into his card and she can change his cart quantity successfully.
	2	If he fail to add to cart he is not to able his product and again to try for fail and wrong type of quantity selection and without product selection he cannot add to cart finally fail.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	After select add to cart customer can buy this order to give shipping.
Quality Requirements	Step	Requirement
	1	Completing his add to cart customer are fully satisfying

3.2.7 Update Cart

Table 3.7 Update cart

Use case	Update cart	
Goal <a longer statement of the goal in context if needed >	Customer must be product adds to cart then he is able to update product quantity. And finally he can update button successfully done.	
Preconditions <what we expect is already the state of the words >	Add at less one product into cart.	
Success End Condition <the state of the world upon successfully complete >	Customer can update if he want to more and more product into his cart then he can increase and update product.	
Failed End Condition <the state of the world if goal abandoned >	No change quantity he no need to add product into his cart and it is fail.	
Primary Action	Customer	
Secondary Actor	Admin	
Trigger<a longer statement of the goal in context if needed >	Another page will come font of customer.	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Customer can add addition product and increase product quantity easily and successfully.
	2	
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	After select add to cart customer can buy this order to give shipping can change for update issue.
Quality Requirements	Step	Requirement
	1	It maintain all product selection and informing

3.2.8 Remove Cart

Table 3.8 Remove Cart

Use case	Remove Cart	
Goal <a longer statement of the goal in context if needed >	Customer has not all update product need or extra product select .for this he need to remove his product from cart so that he can select Remove action and finally it will be done.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	A customer can anytime any product can remove after section and before payment that's reason it is successfully done.	
Failed End Condition <the state of the world if goal abandoned >	If customer forget to remove extra product to his payment will extra product it goes to fail condition	
Primary Action	Customer	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	Directly go to payment option	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Customer has ability to change tis and any time he can change to remove cart at all.
	2	Customer forget to remove extra product to his payment will extra product it goes to fail condition and extra charge for him with vat.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	Customer has an alternation way to remove his quantity and his product to in change before shipping information or payment option
Quality Requirements	Step	Requirement
	1	All products are store appropriately.

3.2.9 Save Shipping Info

Table 3.9 Save Shipping Info

Use case	Save Shipping Info	
Goal <a longer statement of the goal in context if needed >	After add to cart customer can all kind of shipping information set into Save shipping information and must be conscious about field validation input because it must be customer receive location important to delivery product.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	Must be save shipping information because here any kind on change of customer reviving location and order delivery address and mobile number.	
Failed End Condition <the state of the world if goal abandoned >	If no select this page it is very harmful and pathetic for customer and customer will not get his product.	
Primary Action	Customer	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	To transfer another page where payment get way selection.	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Customer must be save shipping information because here any kind on change of customer reviving location and order delivery address and mobile number. So that customer can get his product his exact location.
	2	Customer wrong information will probability to loss this product and can be fail
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	There Is no alternative way to solution for this
Quality Requirements	Step	Requirement
	1	Ensure his appropriate selection it will helpful and back to his new selection

3.2.10 Order and Payment

Table 3.10 Order and payment

Use case	Order and payment	
Goal <a longer statement of the goal in context if needed >	This features or functional Requirement more important for customer because it complete order and payment to get product of a customer success.	
Preconditions <what we expect is already the state of the words >	Add cart	
Success End Condition <the state of the world upon successfully complete >	Customer when clearly order and payment completely his cleaner or invoice report generate to admin site and one request submit into server side successfully.	
Failed End Condition <the state of the world if goal abandoned >	If customer fails order and payment then again he need to add cart product and go back page again and select product again its fail condition.	
Primary Action	Customer	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	Directly go to payment option and will open mobile banking and cash on delivery.	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Customer will also get clearly order and payment completely his cleaner or invoice report generate to admin site and one request submit into server side with shipping information successfully.
	2	Here customer fails order and payment then then he goes through fail condition.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	The alternative way again to select product and then he will able to order and payment
Quality Requirements	Step	Requirement
	1	There is no change to any kind of success full quality at all

3.2.11 Add Product

Table 3.11 Add Product

Use case	Add Product	
Goal <a longer statement of the goal in context if needed >	Admin can add product with some information including some field.so that the customer can view all kind of product after adding admin product list and can order this product finally	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	After adding product customer can buy and their all kind of information can show into admin panel.	
Failed End Condition <the state of the world if goal abandoned >	If admin miss take add product it will not view add in customer site	
Primary Action	Admin	
Secondary Actor	Customer, Guest.	
Trigger<a longer statement of the goal in context if needed >	It will be redirect add category page	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Admin will add product customer can buy and their all kind of information can show into admin panel it is the successful work
	2	If admin no add addition product it will not visualize client's site or customer site.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	Here no alternation way to add product of admin panel.
Quality Requirements	Step	Requirement
	1	Every day satisfaction good providing includes.

3.2.12 Add Category

Table 3.12 Add Category

Use case	Add Category	
Goal <a longer statement of the goal in context if needed >	Here add category is important part. It also depend on admin .otherwise customer side on category will not show.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	After add category it look very structure in customer site and it help to awesome and attractive to there and more product show add customer will satisfy to see more product and will more order.	
Failed End Condition <the state of the world if goal abandoned >	If admin no add category then it will be looking poor customer site and customer will not satisfy.	
Primary Action	Admin	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	It will redirect to Manage category page	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Admin must add category because customer site and it help to awesome and attractive to there and more product show add customer will satisfy to see more product and will more order.
	2	An admin no add category then it will be looking poor customer site and customer will not satisfy. They will not visit this site and it will fall achieve goal.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	No way to cover this scenario
Quality Requirements	Step	Requirement
	1	The entire try including issue to quality maintenance.

3.2.13 Add Brand

Table 3.13 Add Brand

Use case	Add Brand	
Goal <a longer statement of the goal in context if needed >	Here add brand is important part. It also depend on admin .otherwise customer side on category will not show.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	After add brand it look very structure in customer site and it help to awesome and attractive to there and more product show add customer will satisfy to see more product and will more order.	
Failed End Condition <the state of the world if goal abandoned >	If admin no add category then it will be looking poor customer site and customer will not satisfy.	
Primary Action	Admin	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	It will redirect to Manage brand page	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Admin must add brand because customer site and it help to awesome and attractive to there and more product show add customer will satisfy to see more product and will more order.
	2	An admin no add brand then it will be looking poor customer site and customer will not satisfy. They will not visit this site and it will fall achieve goal.
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	No way to cover this scenario
Quality Requirements	Step	Requirement
	1	It makes sensitive and quality maintenance.

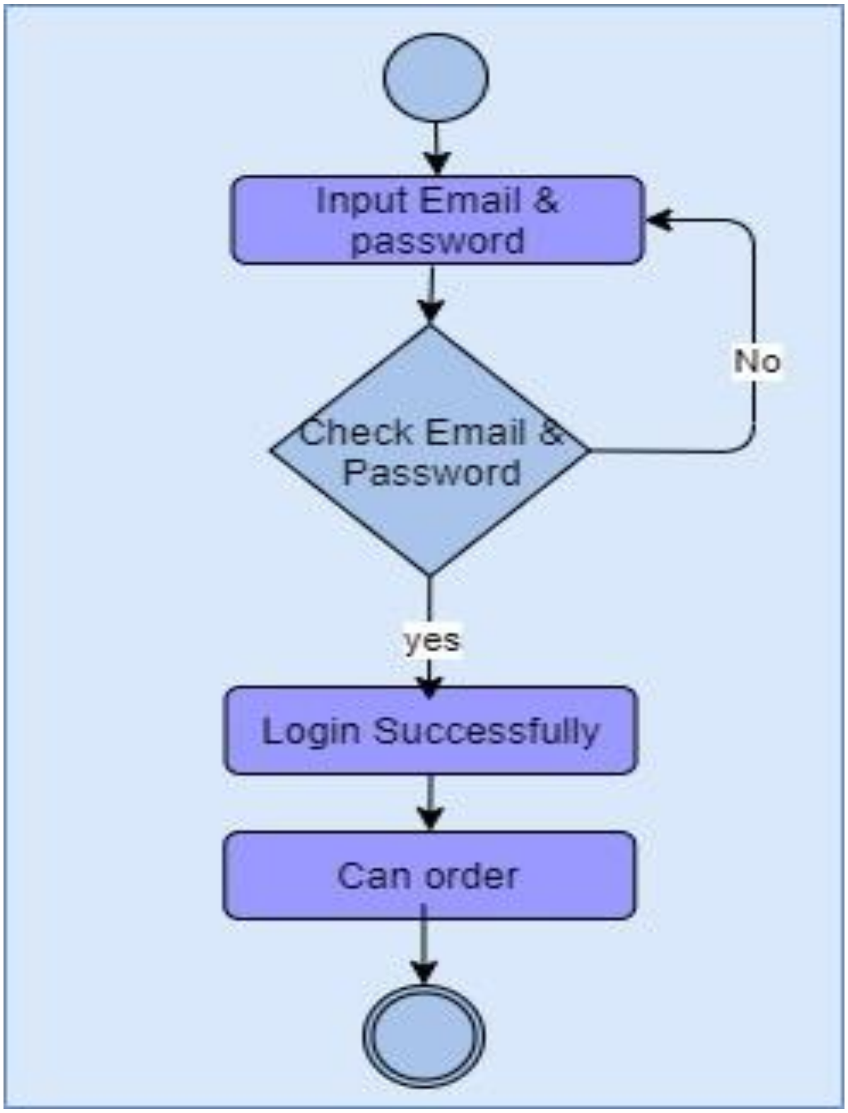
3.2.14 Manage Order

Table 3.14 Manage Order

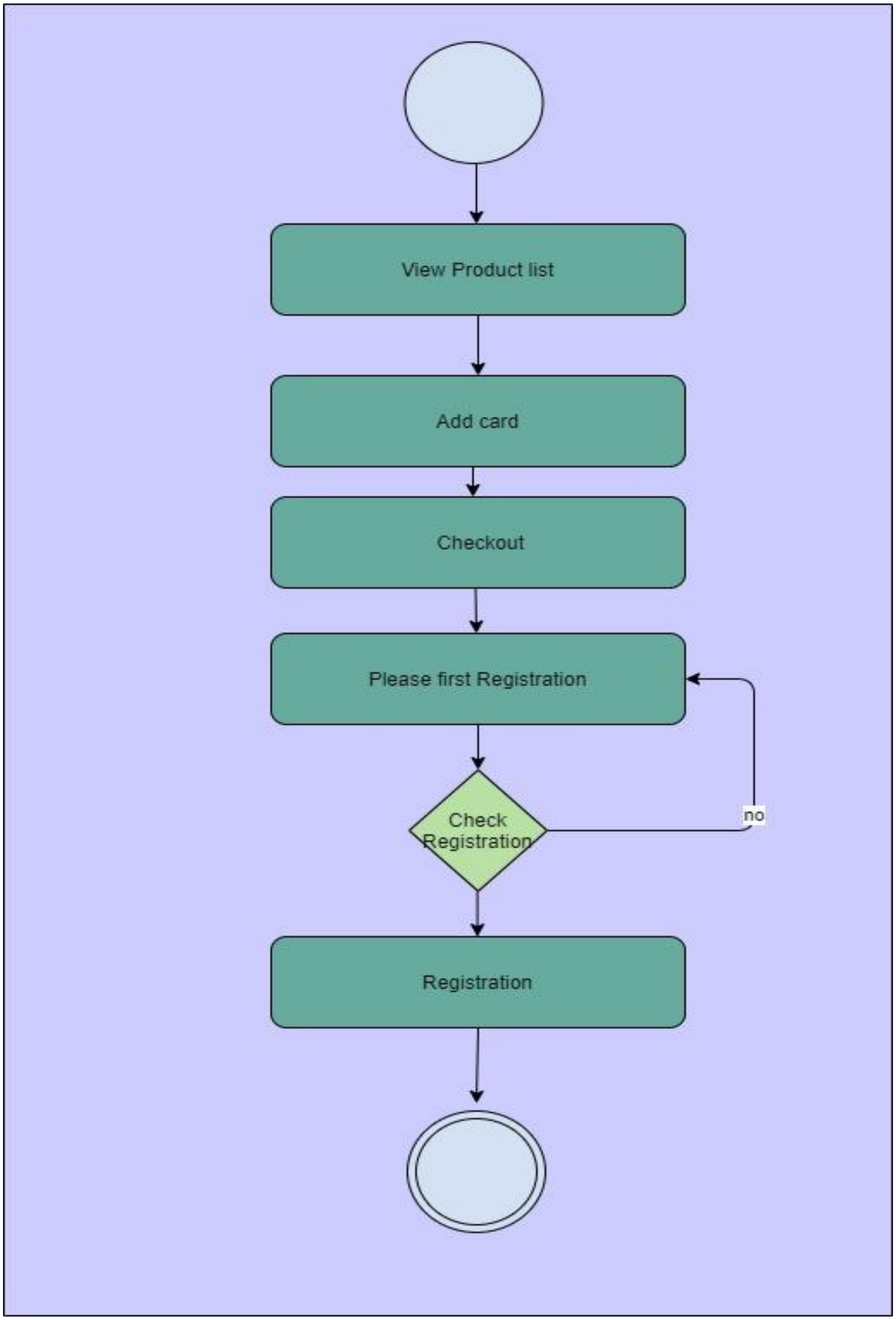
Use case	Manage Order	
Goal <a longer statement of the goal in context if needed >	Manage order is a major part of this project all kind of information order, payment create instance view in this part and control here.	
Preconditions <what we expect is already the state of the words >	N/A	
Success End Condition <the state of the world upon successfully complete >	Here all kind of outcome show and delivery report invoice and another data check here and finally delivery report generate from here.	
Failed End Condition <the state of the world if goal abandoned >	If there is something wrong product delivery will not appropriate serve from her and customer will fall in suffering.	
Primary Action	Admin	
Secondary Actor	N/A	
Trigger<a longer statement of the goal in context if needed >	It will redirect to this page and no other back.	
Description/main Success <the steps of the scenario from the trigger to a goal delivery and any clean up after >	Step	Action
	1	Complete information data and all kind of information provide from here and again all kind of success depends of dis functionality.
	2	Miss behave also can damage data site and customer information in this section so fail issue can take danger harmful .
Alternative Flows <a: condition causing branching > <a1: action or name of sub use case >	Step	Branching action
	a1	No way to recovery and plane wise develop
Quality Requirements	Step	Requirement
	1	It 100% includes and find the quality requirements to successful collection

3.3 ACTIVITY DIAGRAM

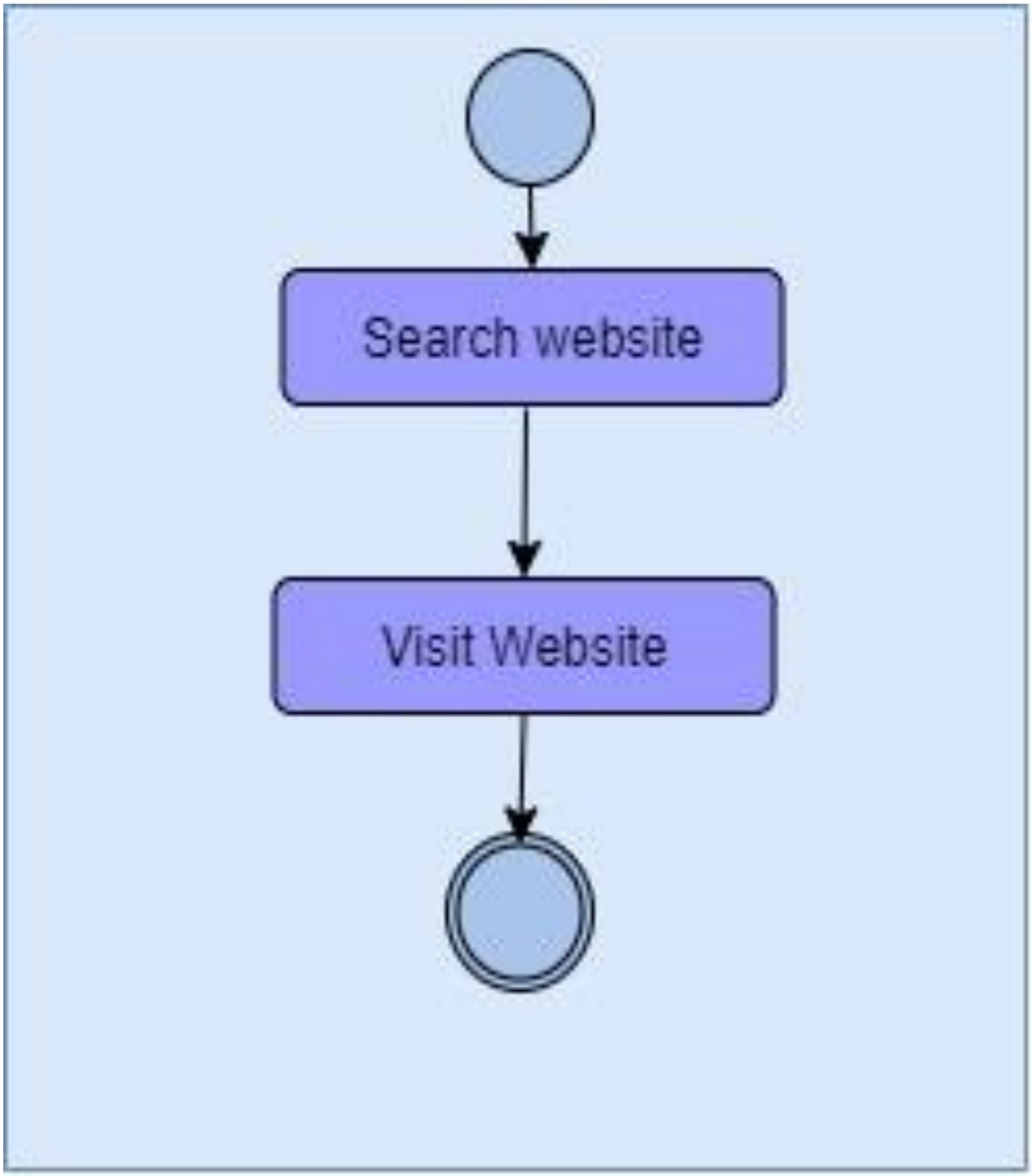
3.3.1 LOGIN



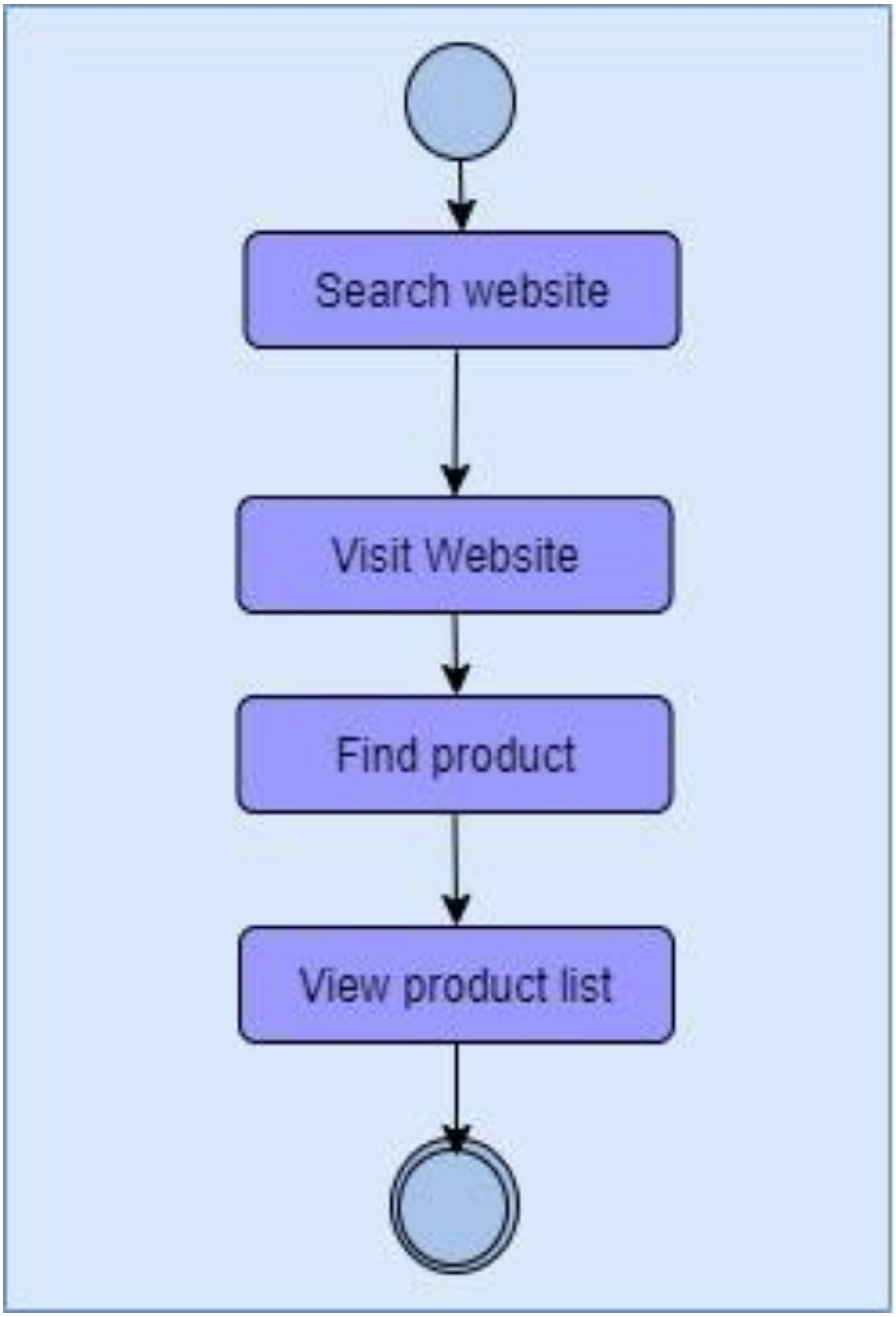
3.3.2 REGISTRATION



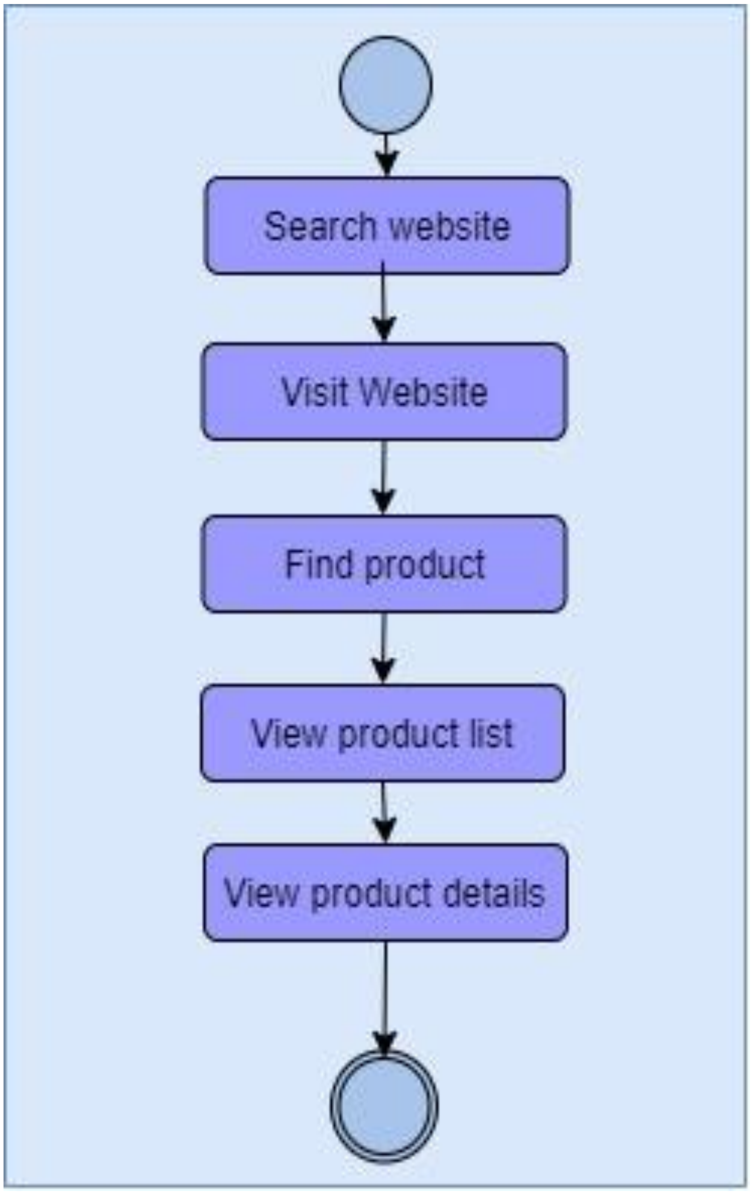
3.3.3 VIST WEBSITE



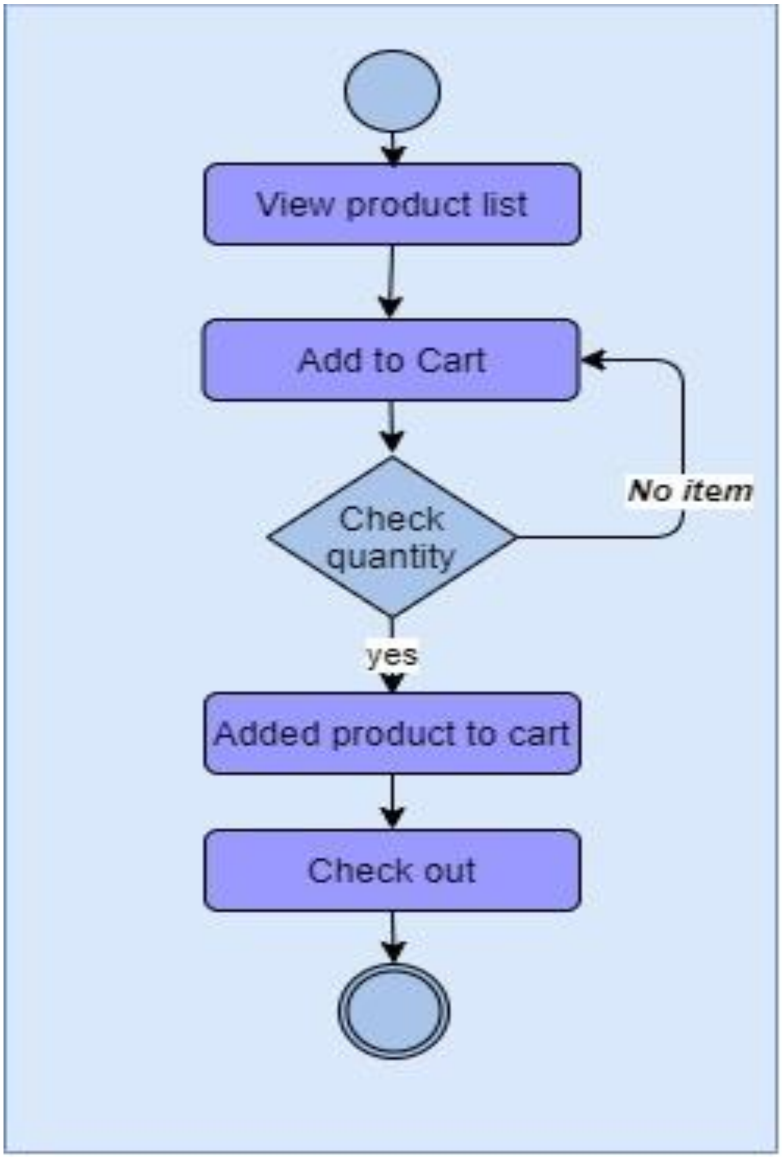
3.3.4 VIST PRODUCT LIST



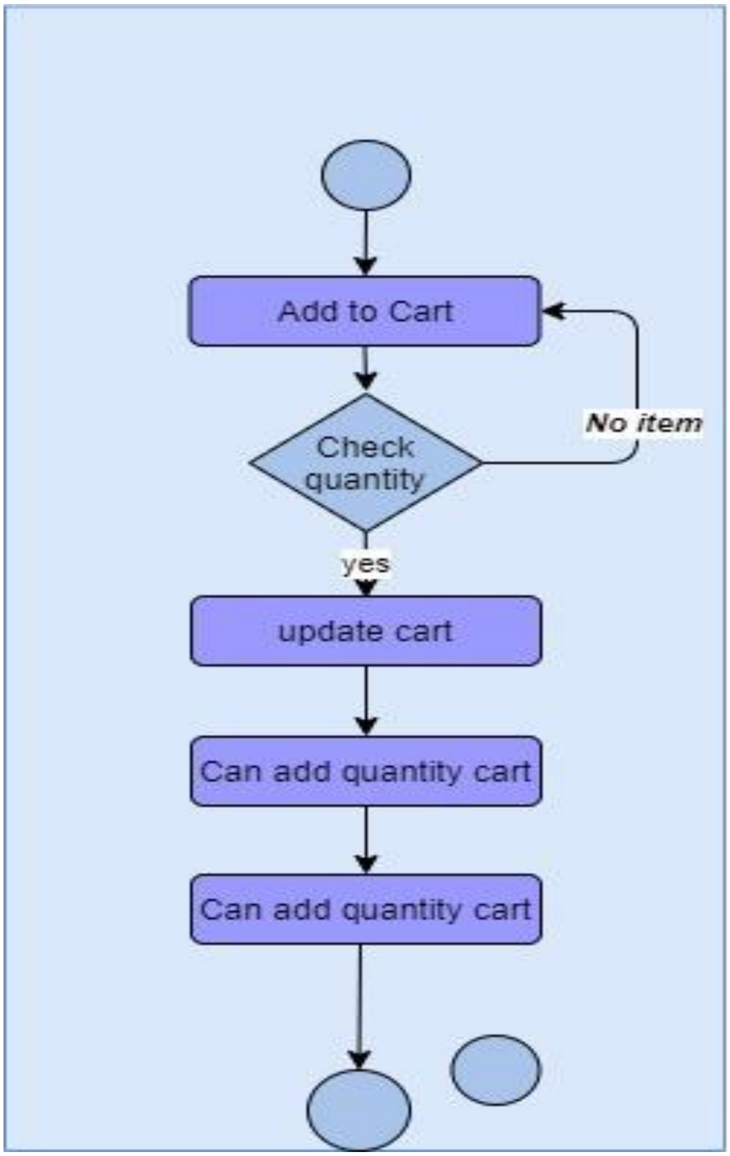
3.3.5 VIST PRODUCT DETAILS



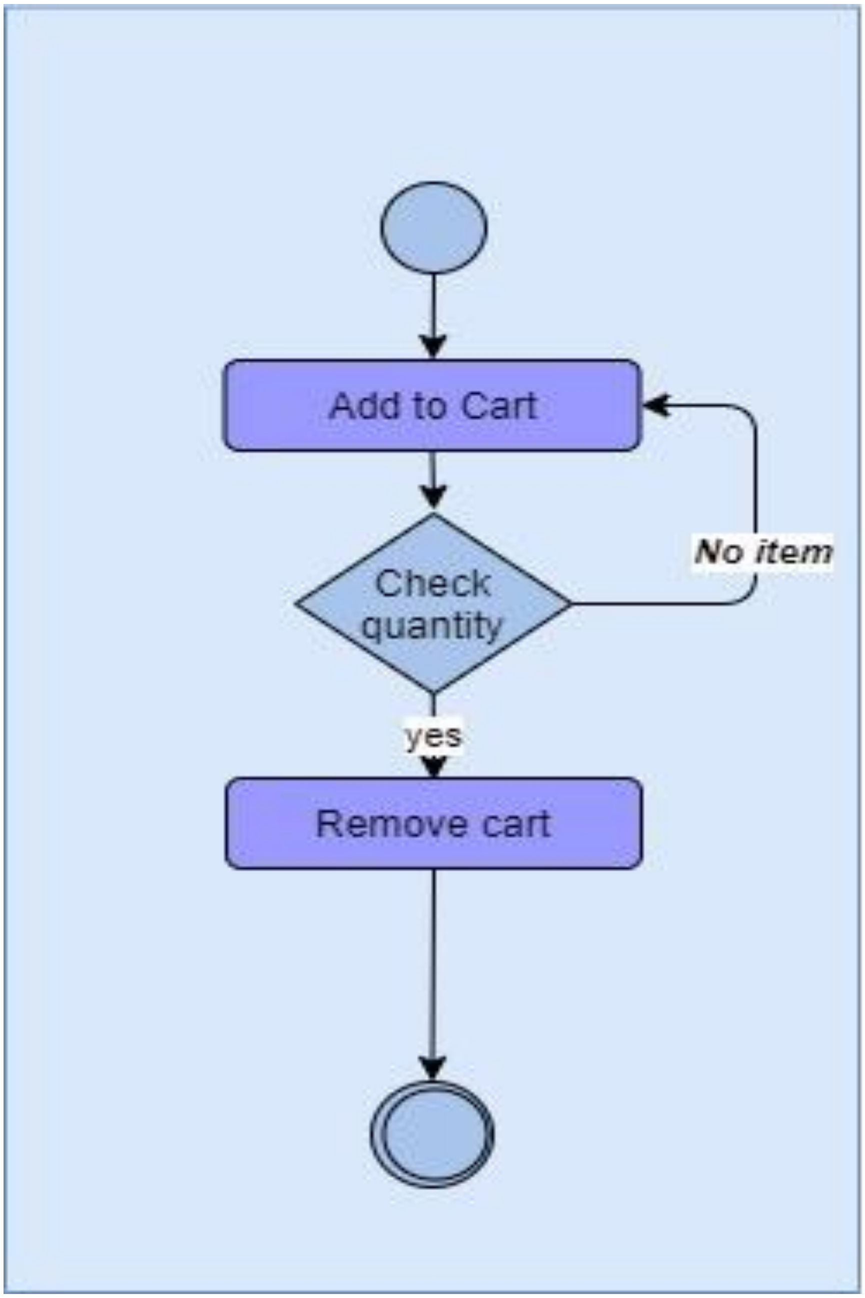
3.3.6 ADD CART


View product list
Add to Cart
Check quantity
No item
yes
Added product to cart
Check out

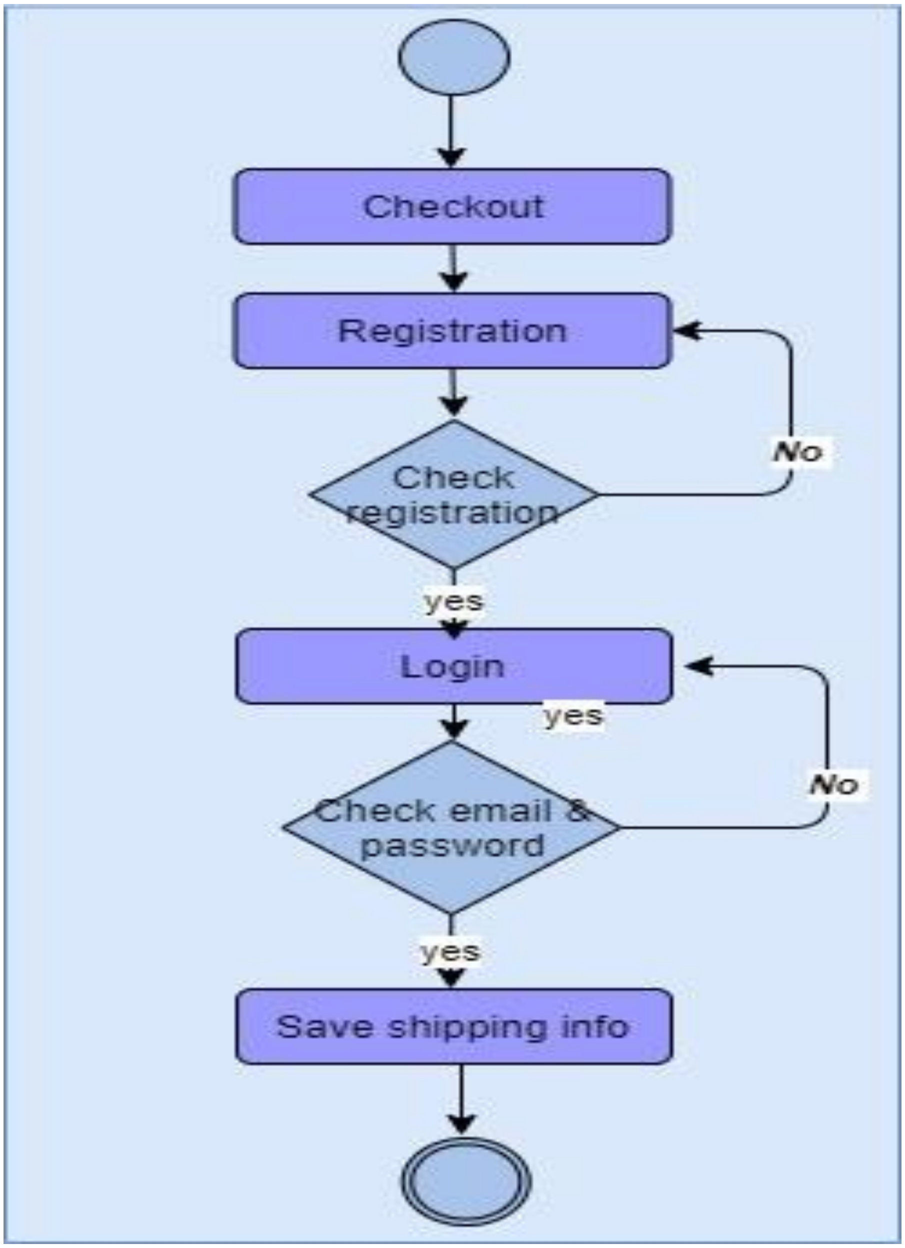
3.3.7 UPDATE CART



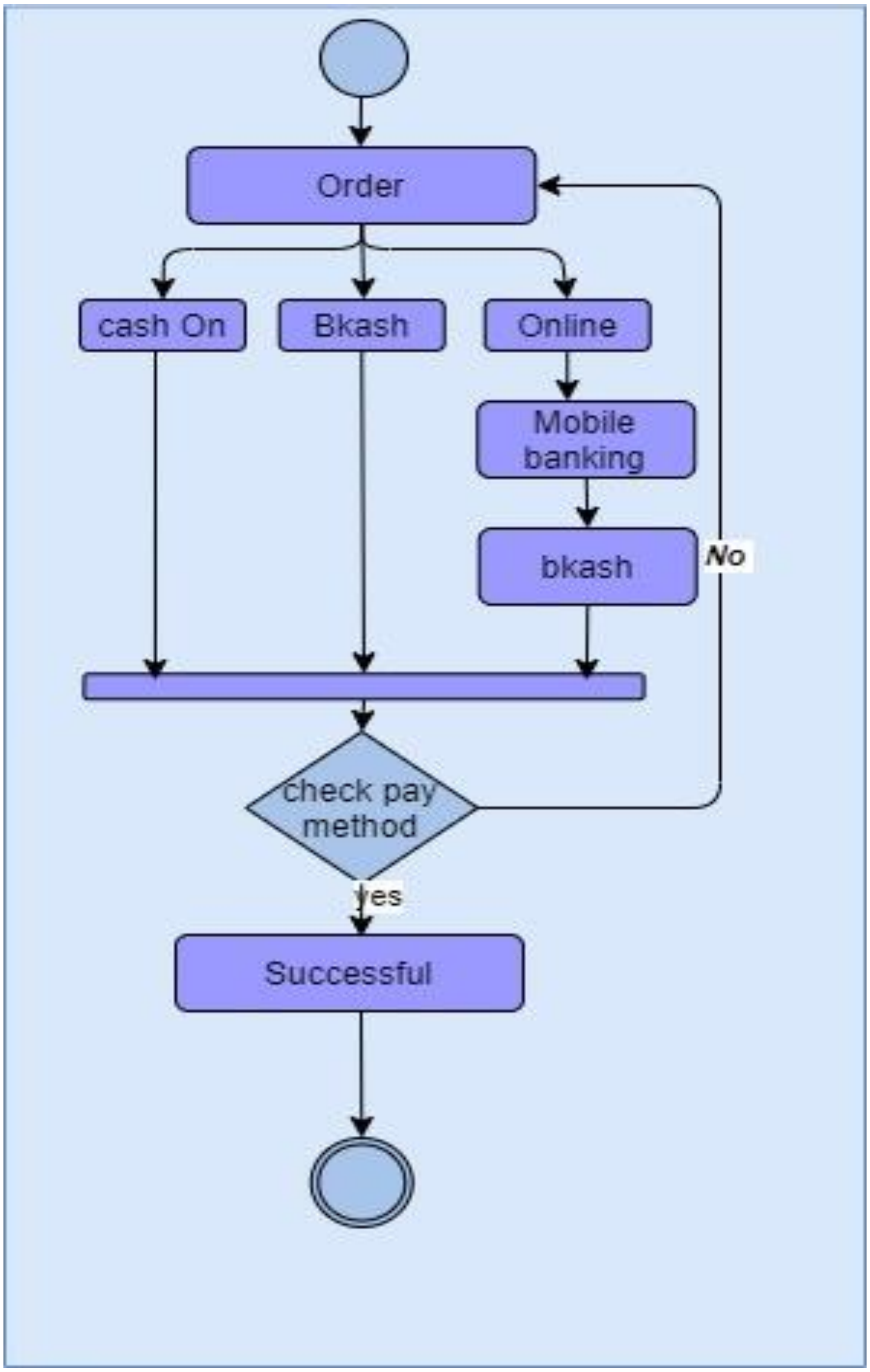
3.3.8 REMOVE CART



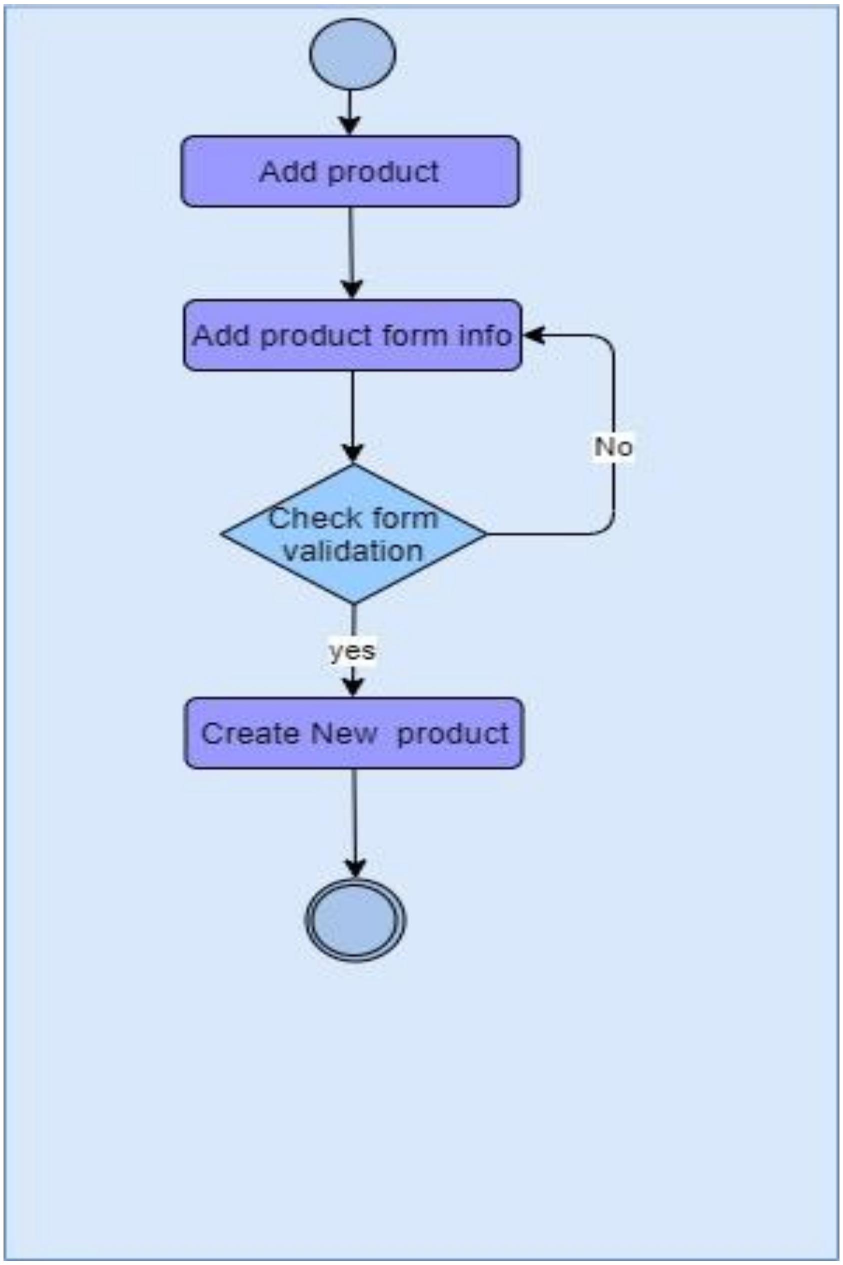
3.3.9 SAVE SHIPPING INFO



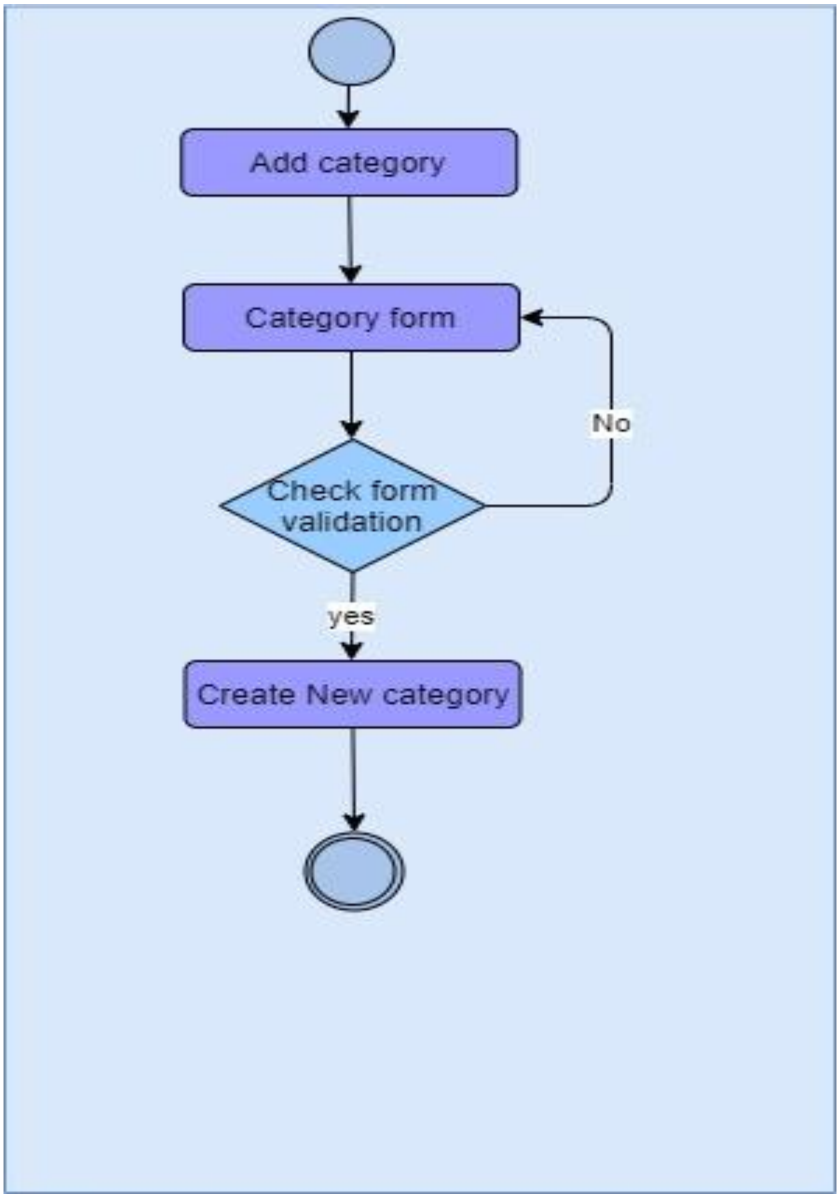
3.3.10 ORDER & PAYMENT



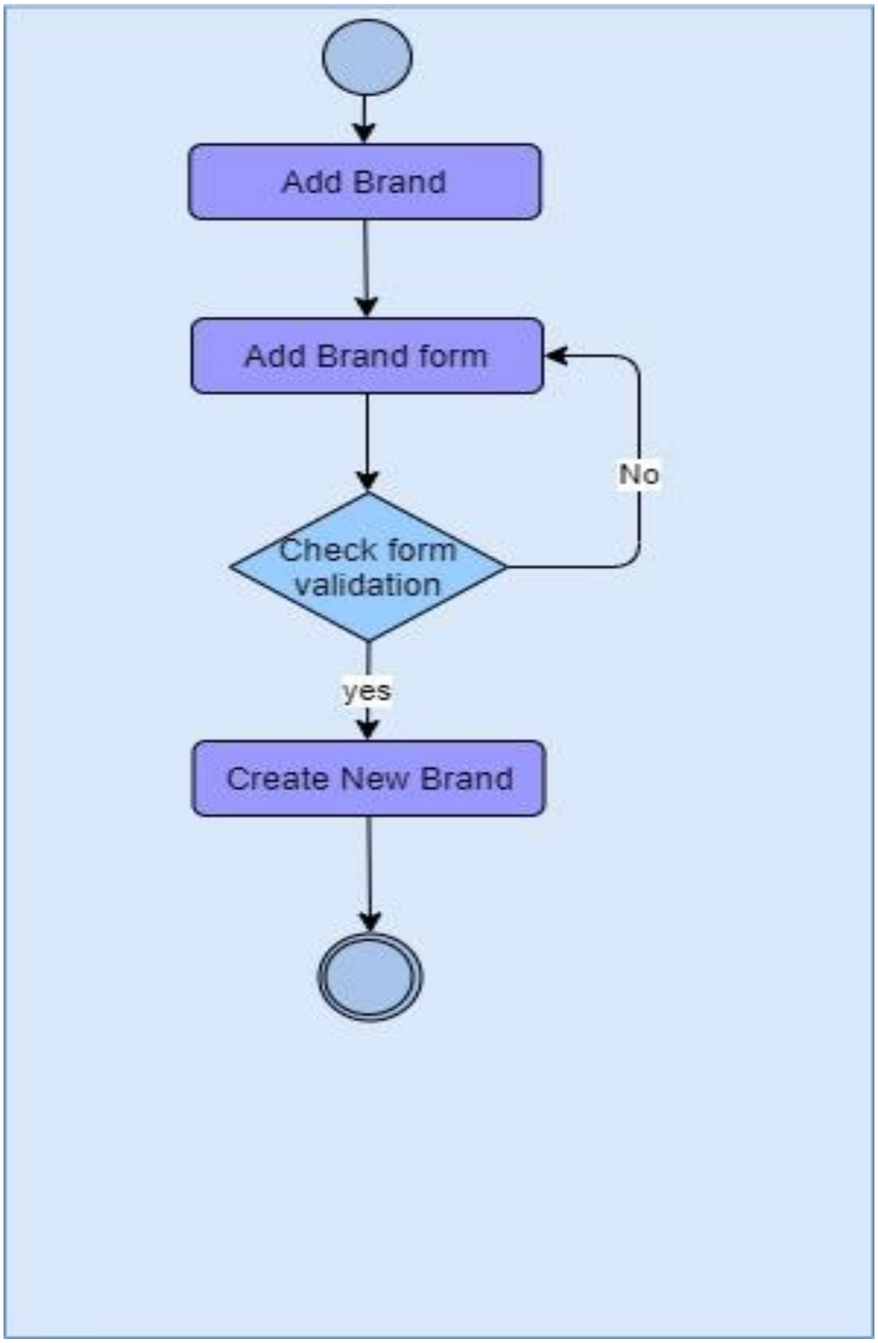
3.3.11 ADD PRODUCT



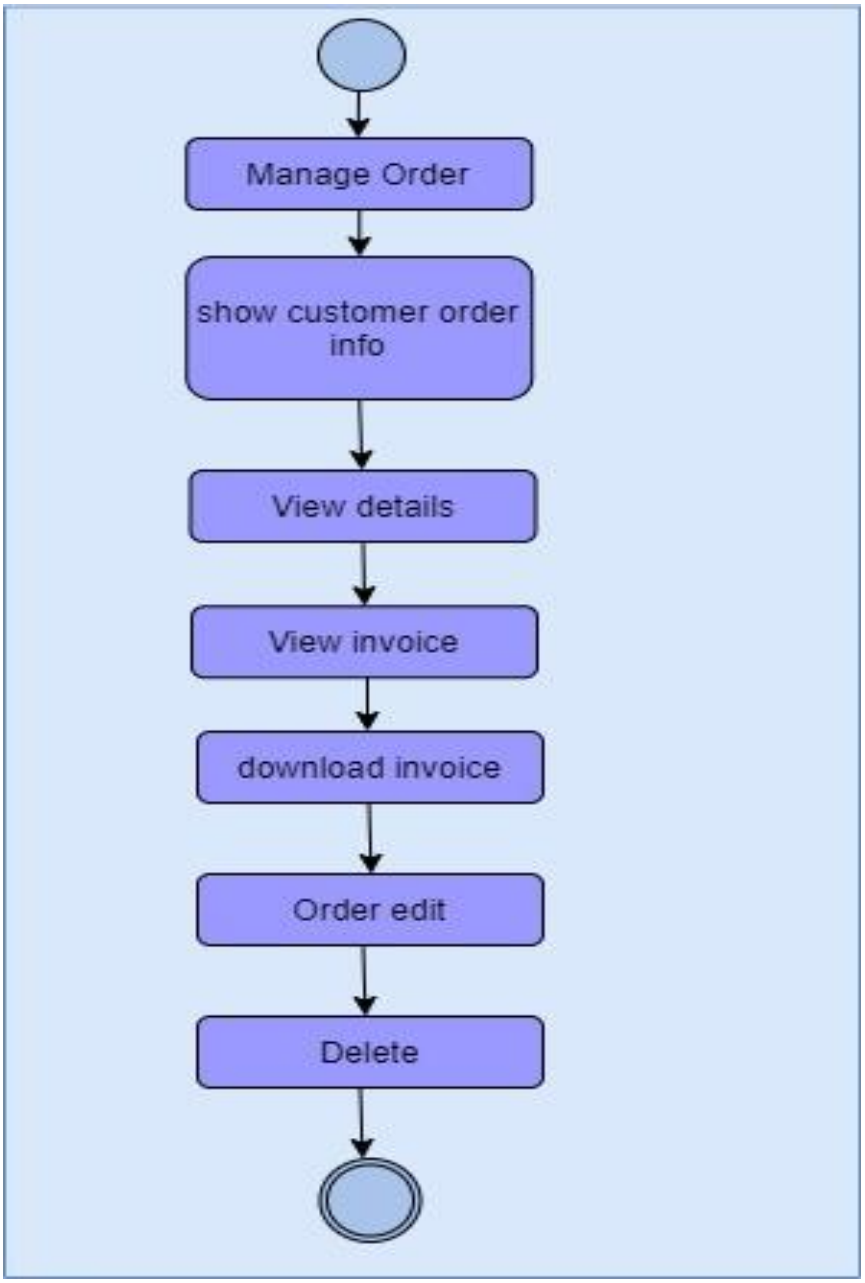
3.3.12 ADD CATEGORY



3.3.13 ADD BRAND

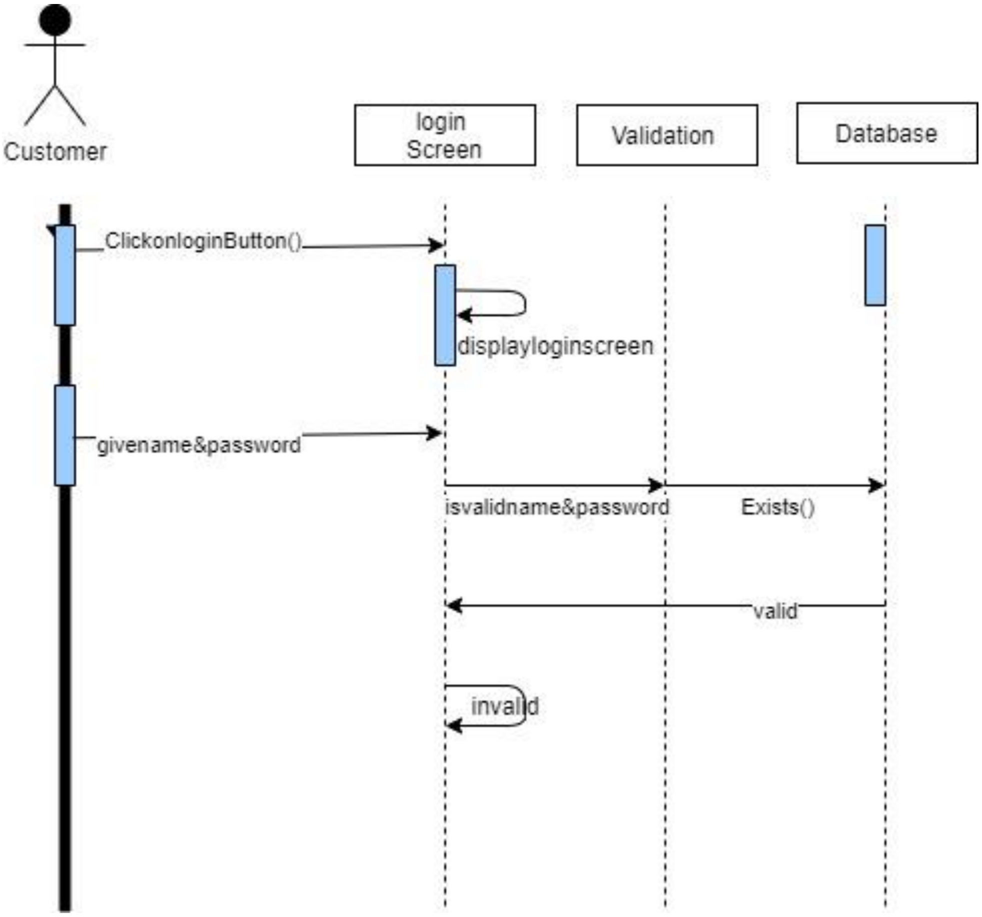


3.3.14 MANAGE ORDER

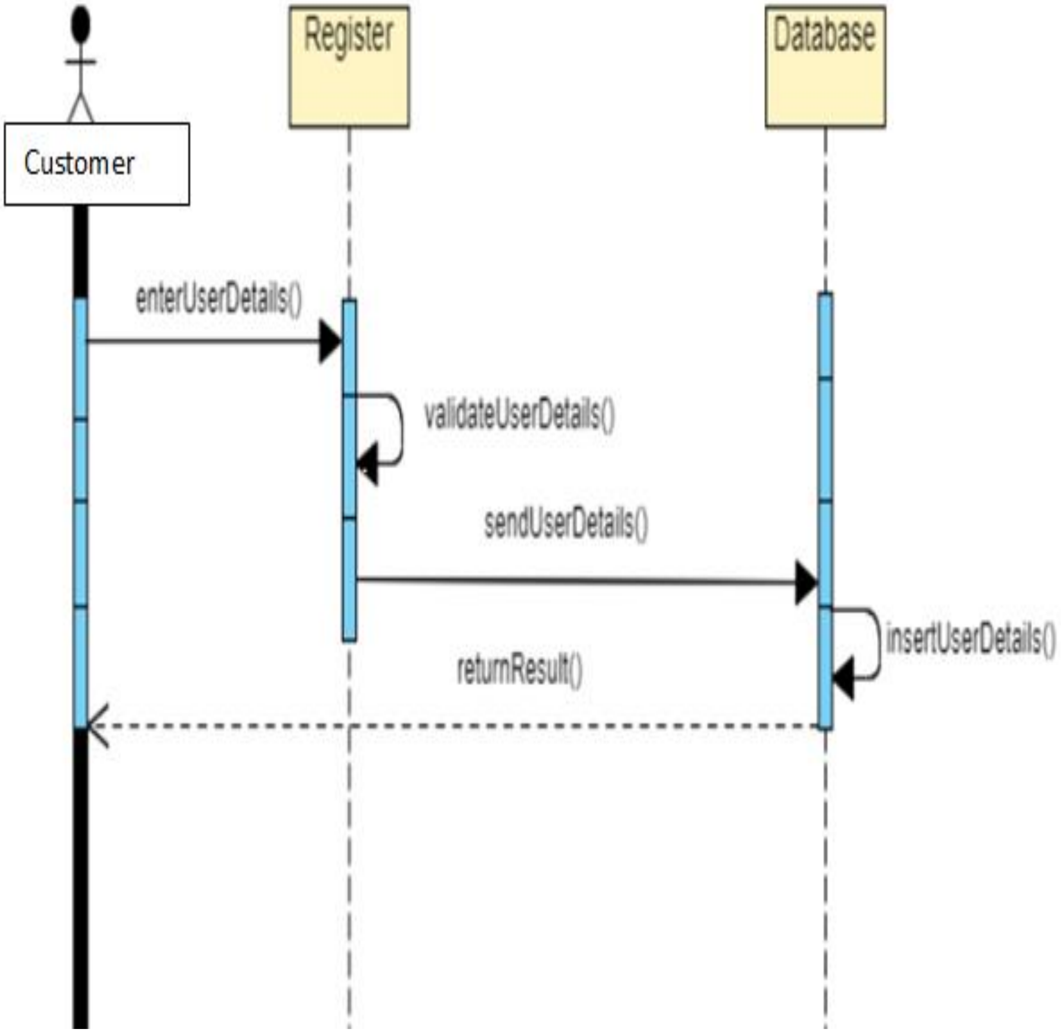


3.4 SEQUENCE DIAGRAM

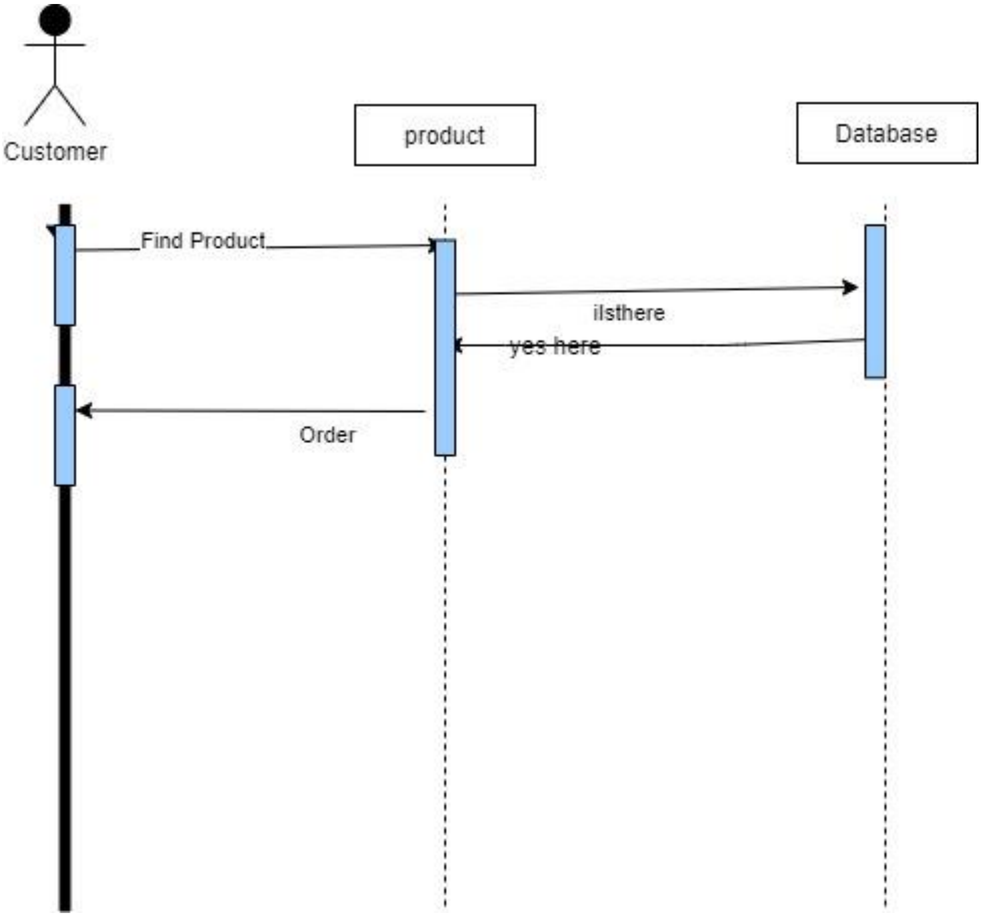
3.4.1 Login Sequence Diagram



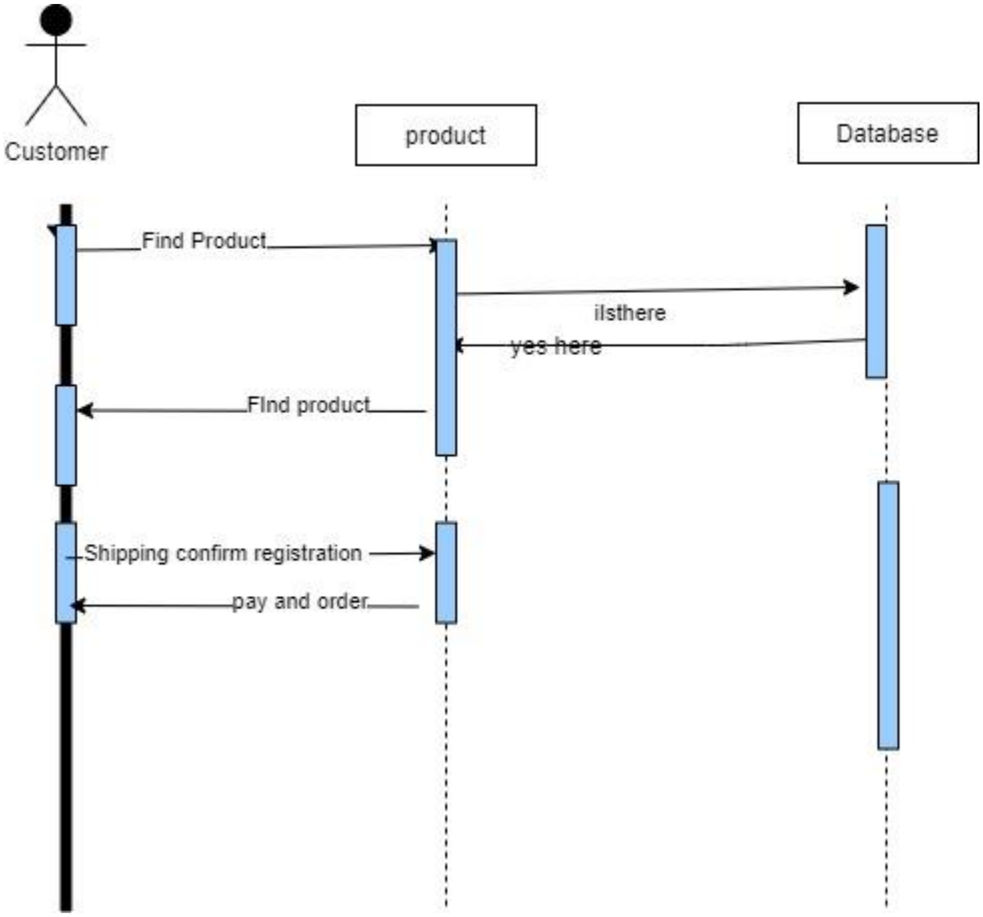
3.4.2 Registration Sequence Diagram



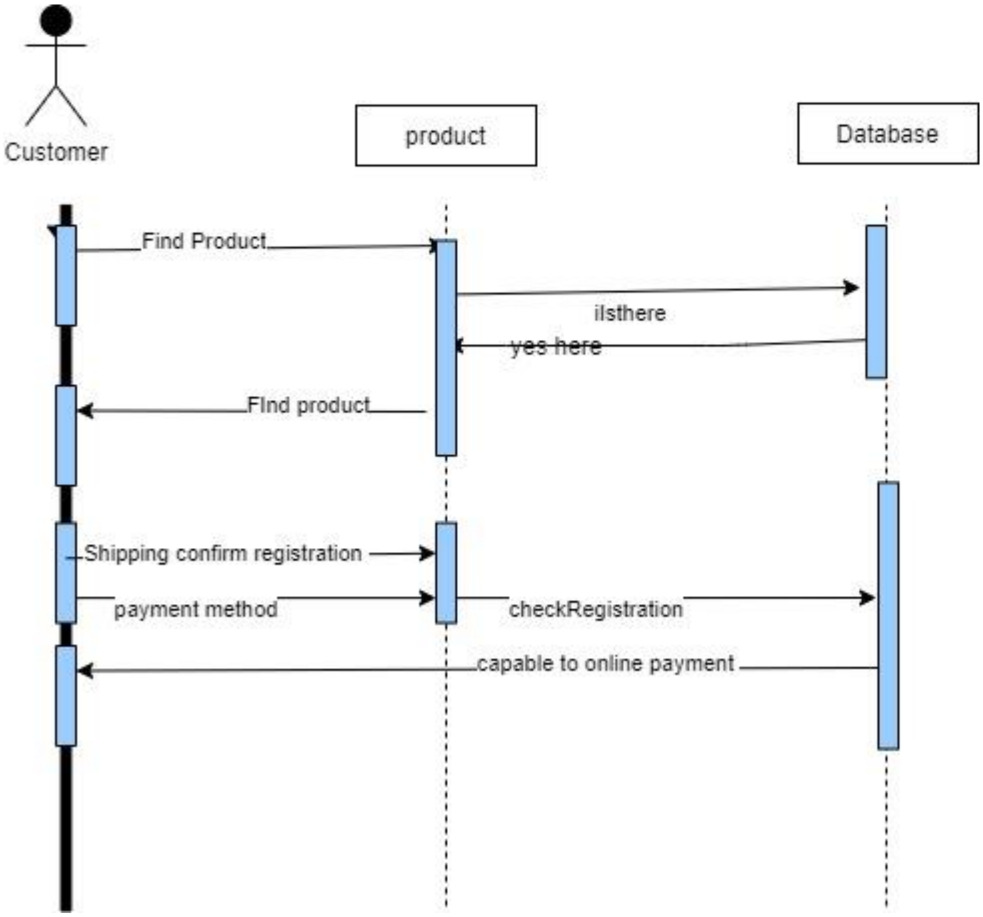
3.4.3 Product List Sequence Diagram



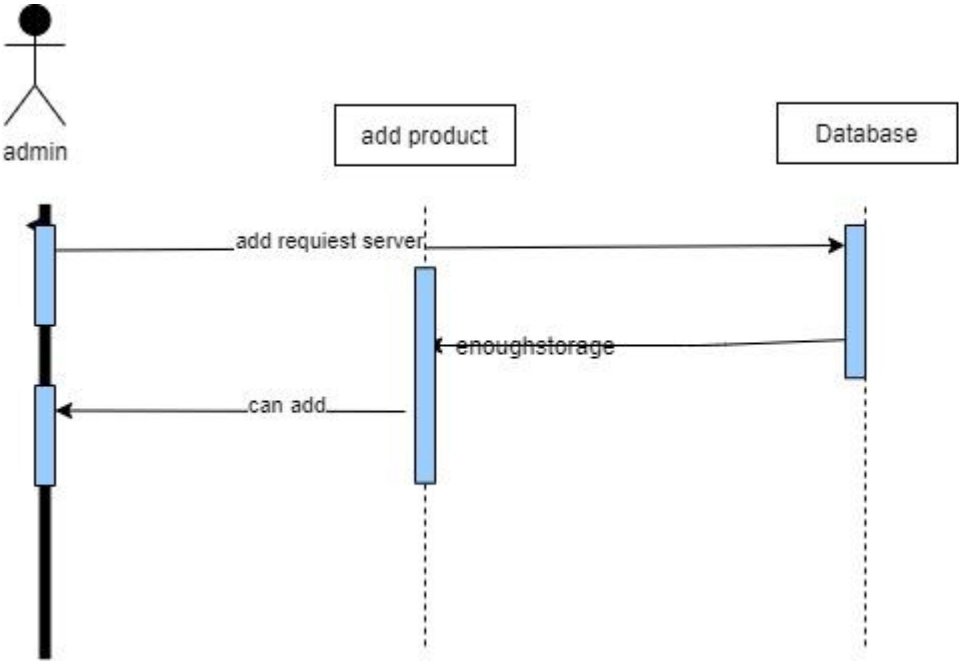
3.4.4 Payment Sequence Diagram



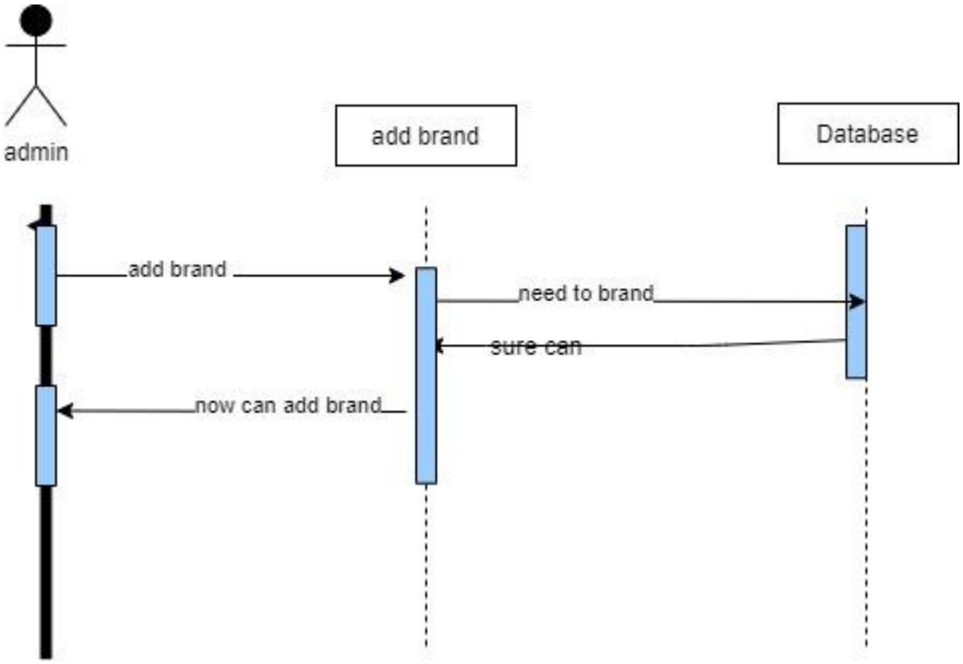
3.4.5 Payment Method Sequence Diagram



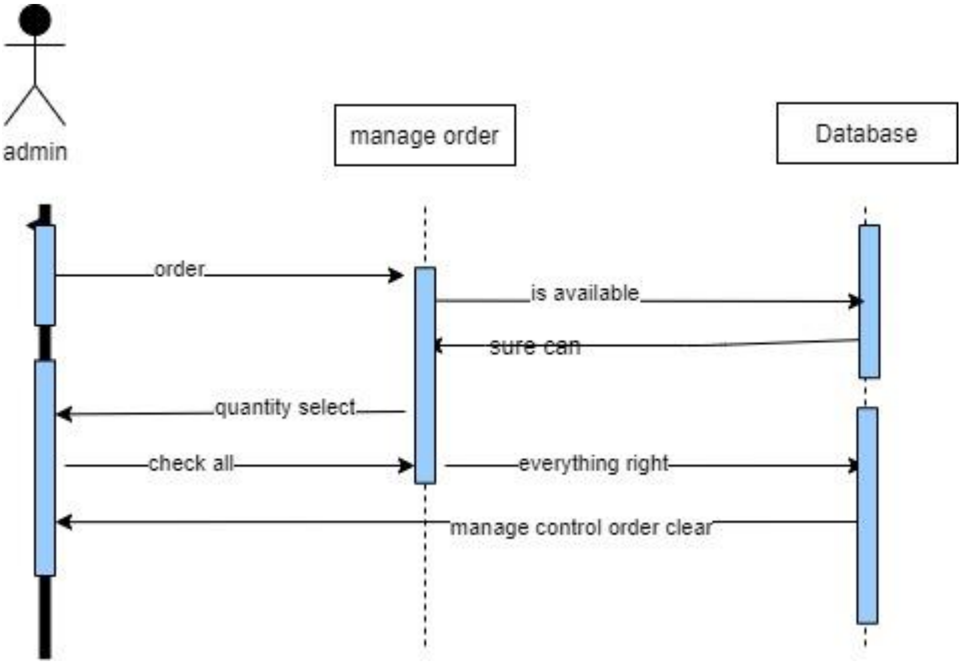
3.4.6 Add Product Sequence Diagram



3.4.7 Brand Product Sequence Diagram

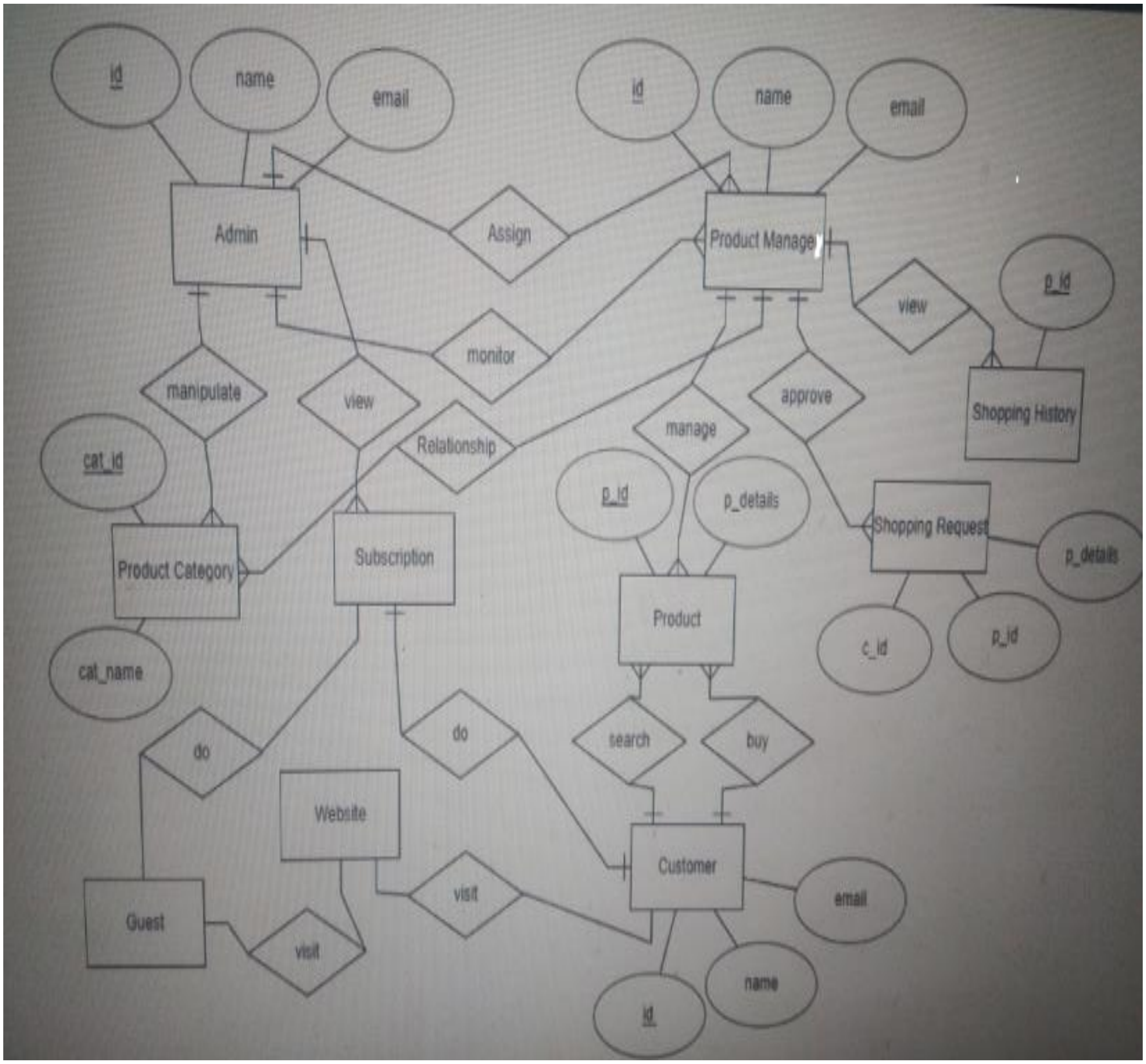


3.4.8 Manage Order Sequence Diagram

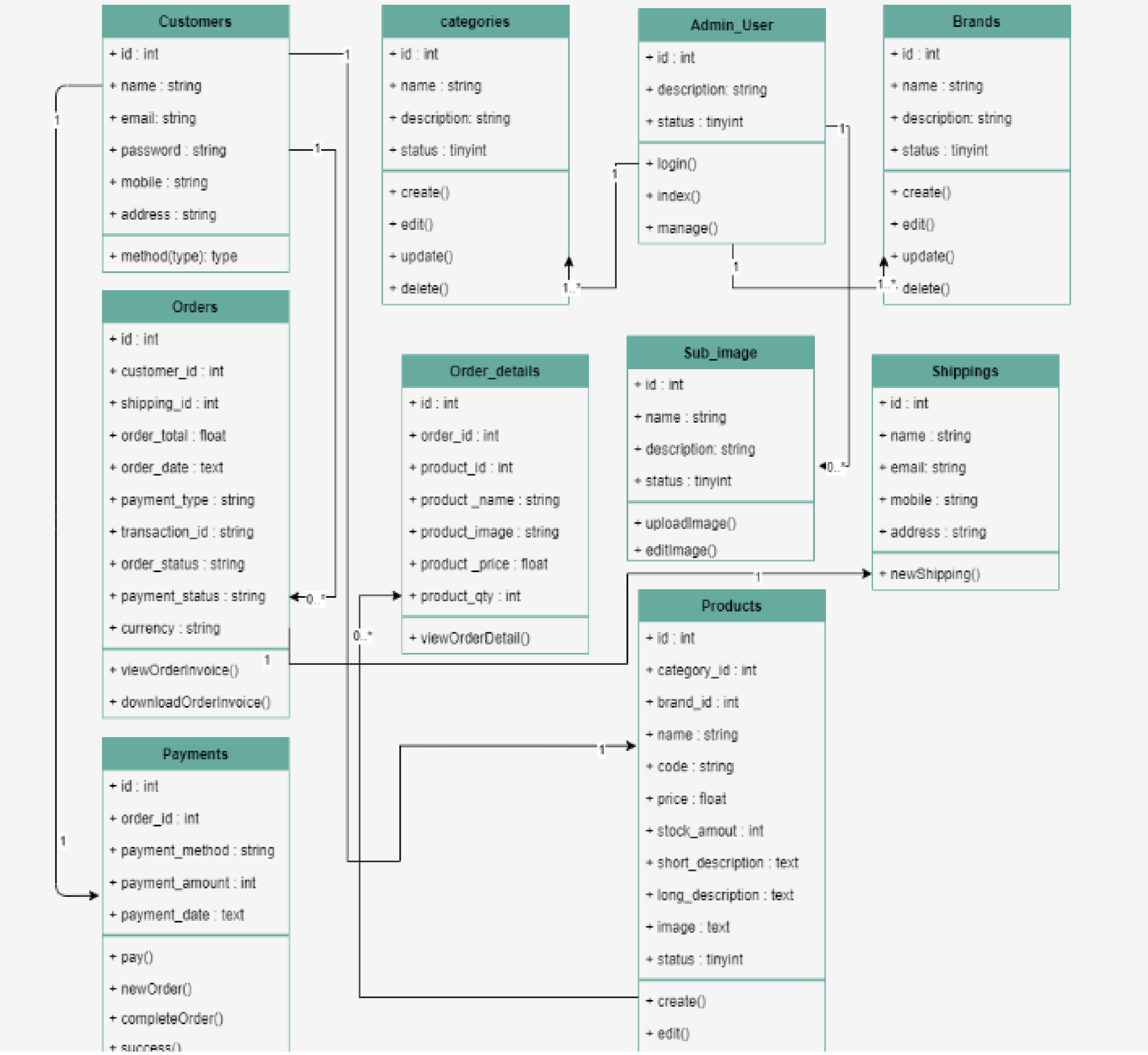


4. CHAPTER-04 SYSTEM DESIGN SPECIFICATION

4.1 ENTITY RELATIONSHIP DIAGRAM



4.2 CLASS DIAGRAM



4.3 Development Tools and Technology

4.3.1 User Interface Technology

- HTML5
- CSS3
- JAVASCRIOT
- BOOTSTRAP

4.3.2 Implementation Tools and Platforms

- PHP7
- Laravel8
- XAMPP
- PHPSTORM

5. CHAPTER-05 SYSTEM TESTING

5.1 Testing Features

5.1.1 Features to be tested

- Login
- Registration
- Order
- Payment
- Manage product
- Manage order

5.1.2 Features not to be tested

- Add category
- Add brand
- Shipping

5.2 Testing Strategies

5.2.1 Test Approach

- The author of the system and admin work done by testing approaches.
- Above all Features will apply testing approaches

5.2.2 Pass or Fail Criteria

- If the condition report match then we are consider the status as Pass
- If the condition report does not match then we are consider the status as Fail

5.2.2 Testing Schedule

Here I consider that the whole project into small tasks and add the schedule for each part of given below

Table 5.1: Testing Schedule

Test phase	**Time**
To Create test plan	2 Weeks
The test Specification test plan	1 Week
Test Specification Team	1 Week
Component Test	2 Weeks
Integration Test	1 Week
System Testing	4 Weeks

5.3 TEST CASE

5.3.1. TEST CASE 1

Table 5.2: Test Case1 (Login)

TEST CASE ID	TEST CASE NAME	TEST SCENARIO	TEST STEPS	TEST DATEA	EXPECTED RESULTS	PASS/ FAIL
TU01	Login	Enter customer login with valid data	1.First need visit and add cart product 2.Registration and then email and password enter 3. Submit button	Email=”jahidhassanen@gmail.com” Password =”1323456”	The customer successfully can add to product into cart	Pass
TU02	Login	Enter Customer login with valid data	1.First need visit and add cart product 2.Registration and then email and password enter 3. Submit button	Email=”jahidhassanen@gmail.com” Password =123456	The customer should not login successfully	Pass

5.3.2. TEST CASE 2

Table 5.3: Test Case2 (Registration)

TEST CASE ID	TEST CASE NAME	TEST SCENARIO	TEST STEPS	TEST DATEA	EXPECTED RESULTS	PASS/ FAIL
TU01	Registration	Enter Customer Name, Email, Password, Address, Mobile, Locations valid data	1.First need visit and add cart product 2.Registration and then email and password enter 3. Submit button	Email=”jahidhassanen@gmail.com” Password =”1323456” Address=”Banani” Mobile=”01765686264” Location=”Dhaka”	Customer successfully register into the system	Pass
TU02	Registration	Enter Customer	1.First need visit and add cart product	Email=”jahidhassanen@gmail.com	Customer enter false	Pass

		Name, Email, Password, Address, Mobile, Locations valid data	2.Registration and then email and password enter 3. Submit button	” Password =”1323456” Address=”Banani” Mobile=”01765686264” Location=”Dhaka”	input he cannot order successfully	

5.3.3. TEST CASE 3

Table 5.4: Test Case3 (Order)

TEST CASE ID	TEST CASE NAME	TEST SCENARIO	TEST STEPS	TEST DATEA	EXPECTED RESULTS	PASS/ FAIL
TU01	Order	Add product into his card and then he can order select	1.Need to add cart 2.then Registration 3.Shipping information 4.after this submit order button	add order into submission then it give the product all information final it come into main outcome	The customer successfully can add complete product order into cart	Pass
TU02	Order	Add product into his card and then he can order select	1.Need to add cart 2.then Registration 3.Shipping information 4.after this submit order button	add order into submission then it give the product all information final it come into main outcome	The customer fail order he cannot successfully order fail	Pass

5.3.4 TEST CASE 4

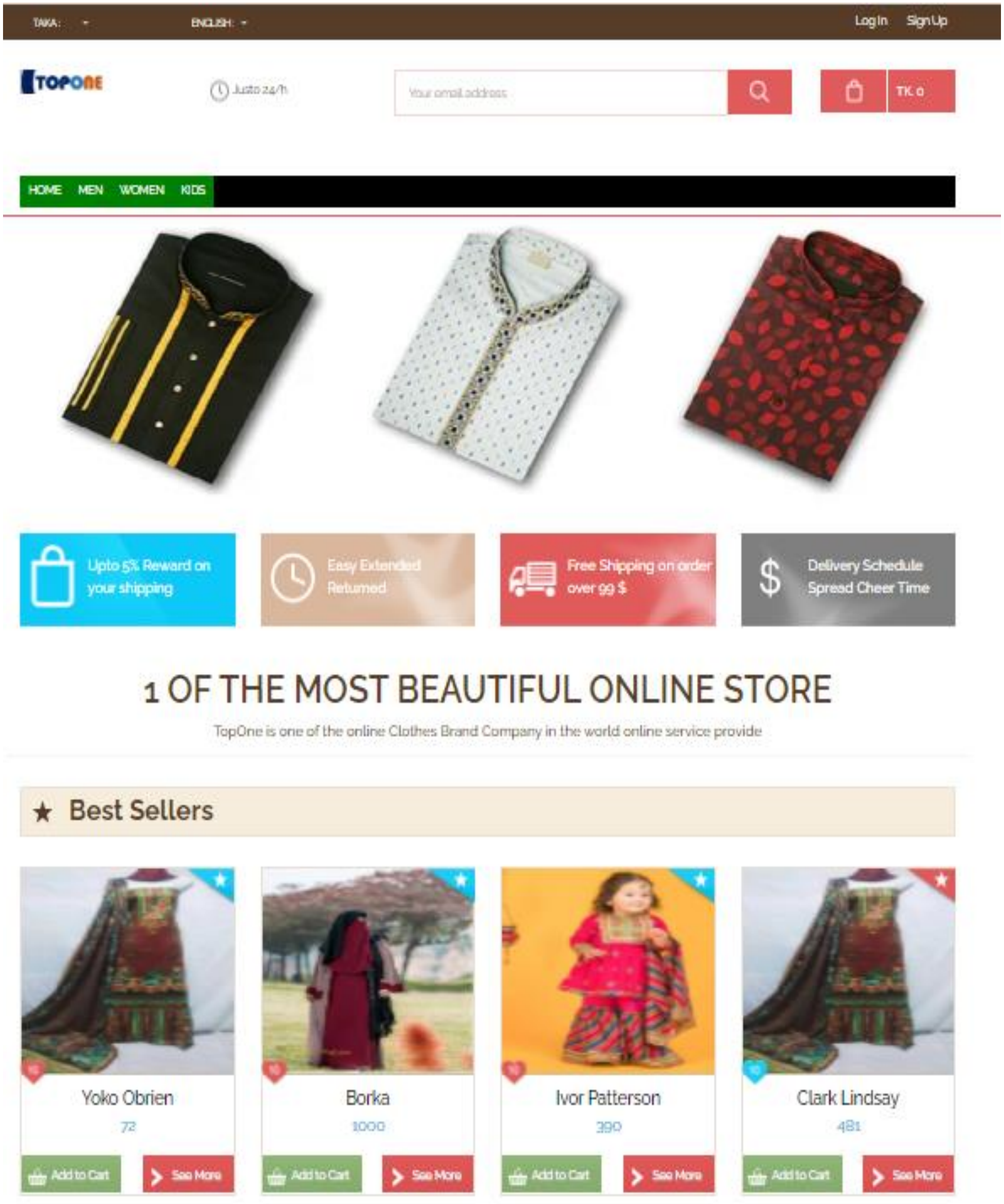
Table 5.5: Test Case4 (Payment)

TEST CASE ID	TEST CASE NAME	TEST SCENARIO	TEST STEPS	TEST DATEA	EXPECTED RESULTS	PASS/ FAIL
TU01	payment	After add product it will goes to payment option	1.First need visit and add cart product 2.Registration and then email and password enter 3. order product 4.then submit payment	Amout=4000 Name=Jahid Email=jahidhassan@gmail.com	Payment has been successfully completed	Pass
TU02	payment	After add product it will goes to payment option	1.First need visit and add cart product 2.Registration and then email and password enter 3. order product 4.then submit payment	Amout=4000 Name=Jahid Email=jahidhassan@gmail.com	Before click to the payment button make sure than customer authorize	Pass

6. CHAPTER-6 USER INTERFACE

6.1 CUSTOMER

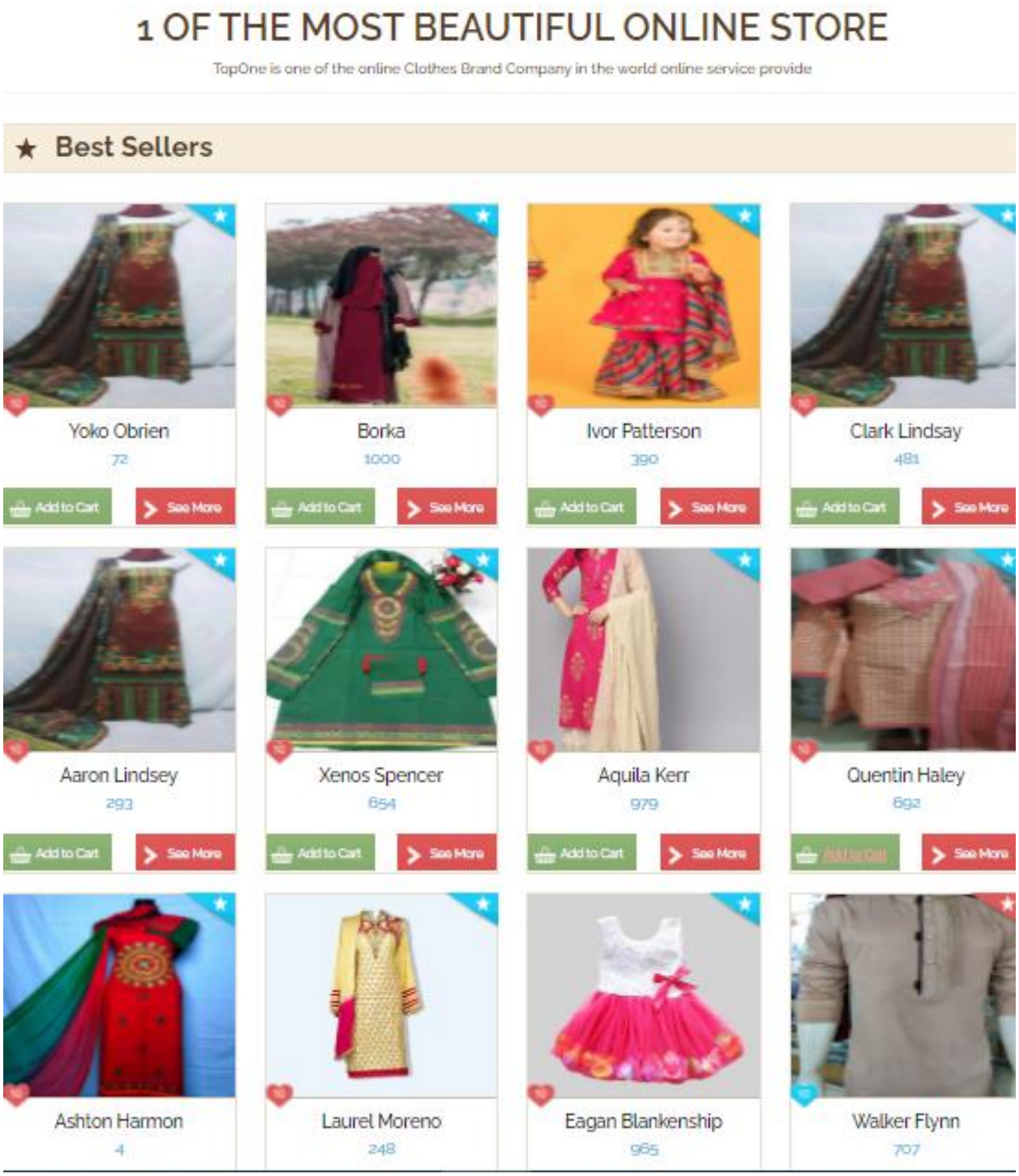
1. View and find customer page



1 OF THE MOST BEAUTIFUL ONLINE STORE

TopOne is one of the online Clothes Brand Company in the world online service provide

2. View product list page



1 OF THE MOST BEAUTIFUL ONLINE STORE

TopOne is one of the online Clothes Brand Company in the world online service provide

★ Best Sellers

Yoko Obrien
72

Borka
1000

Ivor Patterson
390

Clark Lindsay
481

Aaron Lindsey
293

Xenos Spencer
654

Aquila Kerr
979

Quentin Haley
692

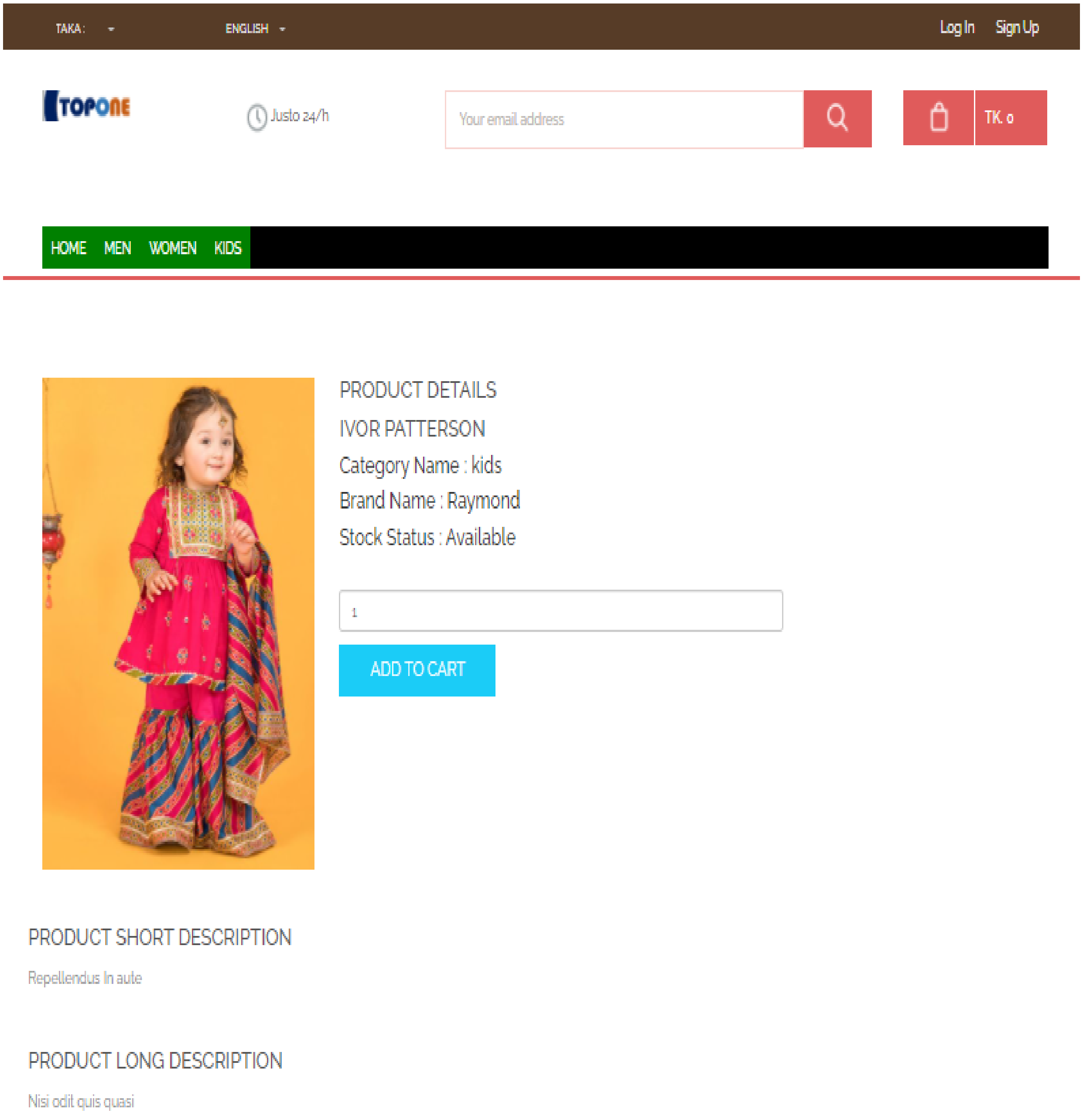
Ashton Harmon
4

Laurel Moreno
248

Eagan Blankenship
965

Walker Flynn
707

3. View product details page



PRODUCT DETAILS

IVOR PATTERSON

Category Name : kids

Brand Name : Raymond

Stock Status : Available

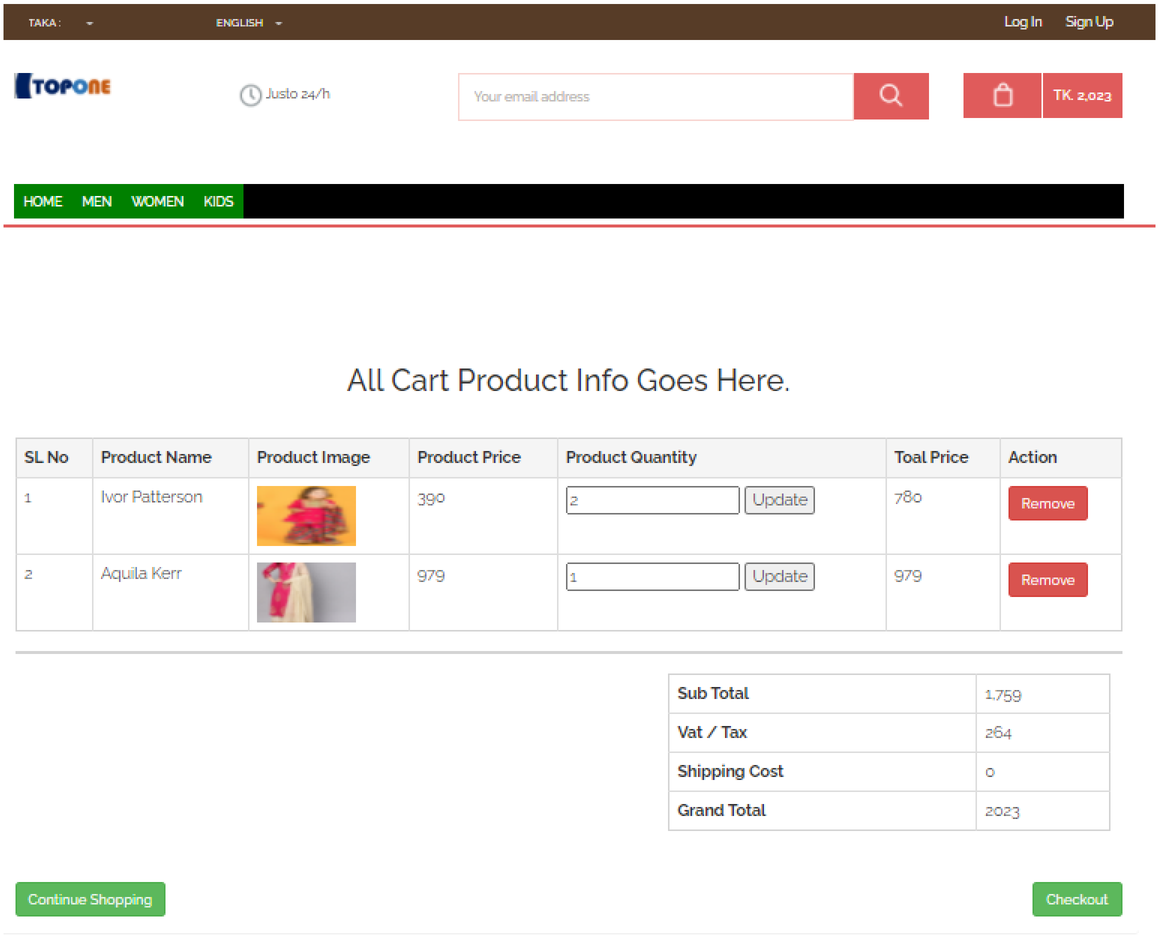
PRODUCT SHORT DESCRIPTION

Repellendus In aute

PRODUCT LONG DESCRIPTION

Nisi odit quis quasi

4. Add cart page

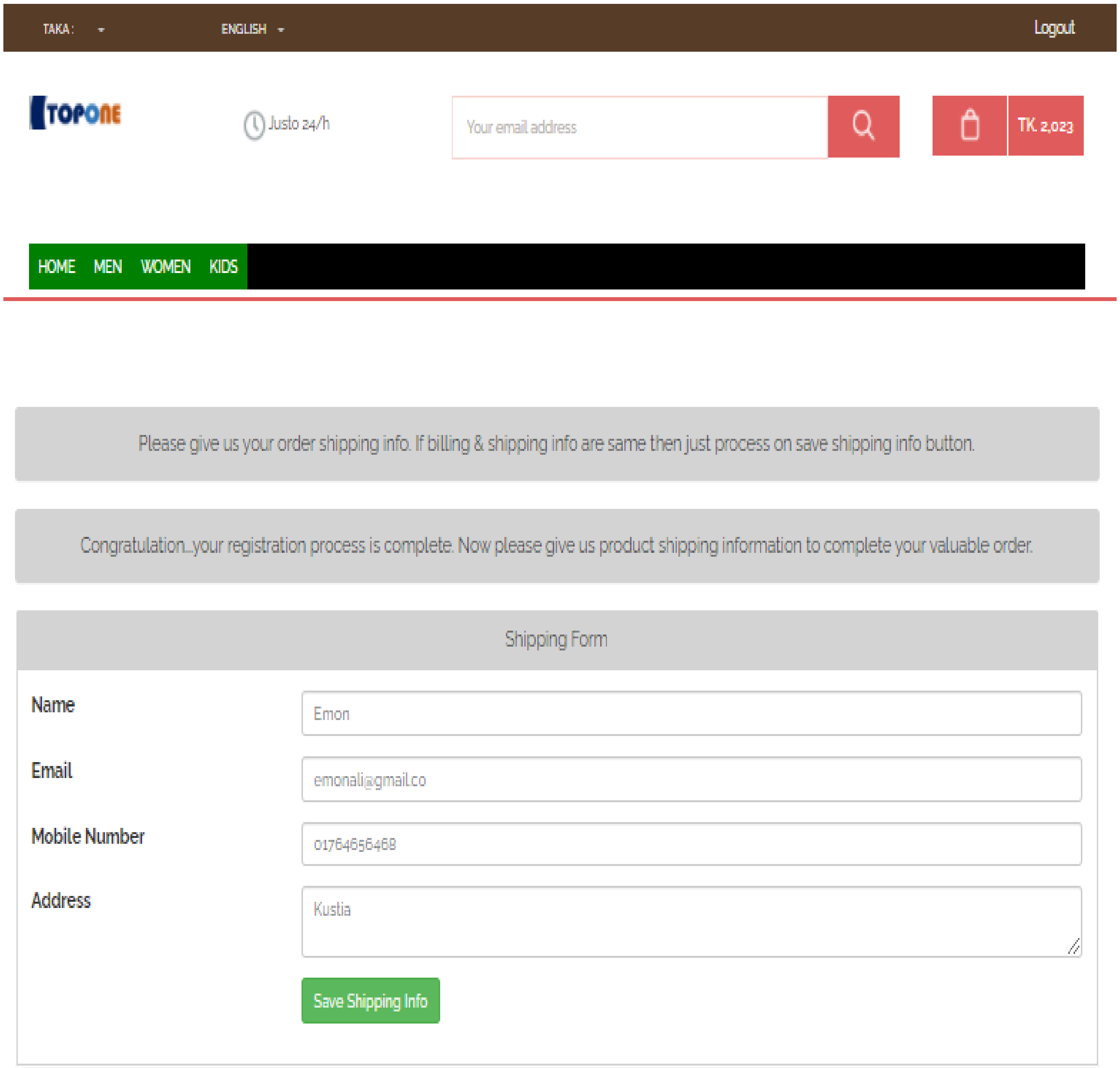


All Cart Product Info Goes Here.

SL No	Product Name	Product Image	Product Price	Product Quantity	Toal Price	Action
1	Ivor Patterson		390	2 Update	780	Remove
2	Aquila Kerr		979	1 Update	979	Remove

Sub Total	1,759
Vat / Tax	264
Shipping Cost	0
Grand Total	2023

5. Registration page



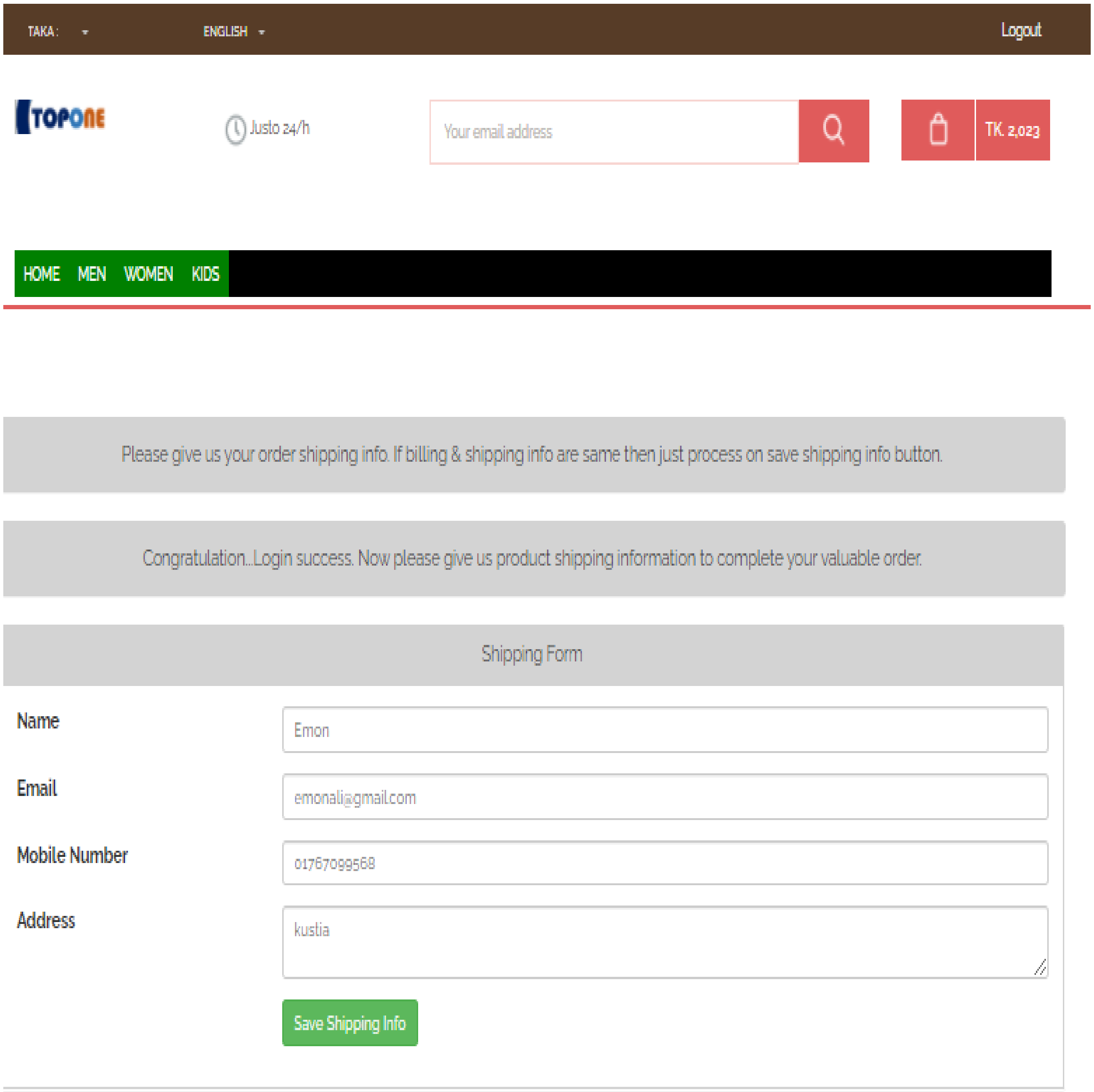
Please give us your order shipping info. If billing & shipping info are same then just process on save shipping info button.

Congratulation..your registration process is complete. Now please give us product shipping information to complete your valuable order.

Shipping Form

Name	Emon
Email	emonali@gmail.co
Mobile Number	01764656468
Address	Kustia

6. Shipping information page page



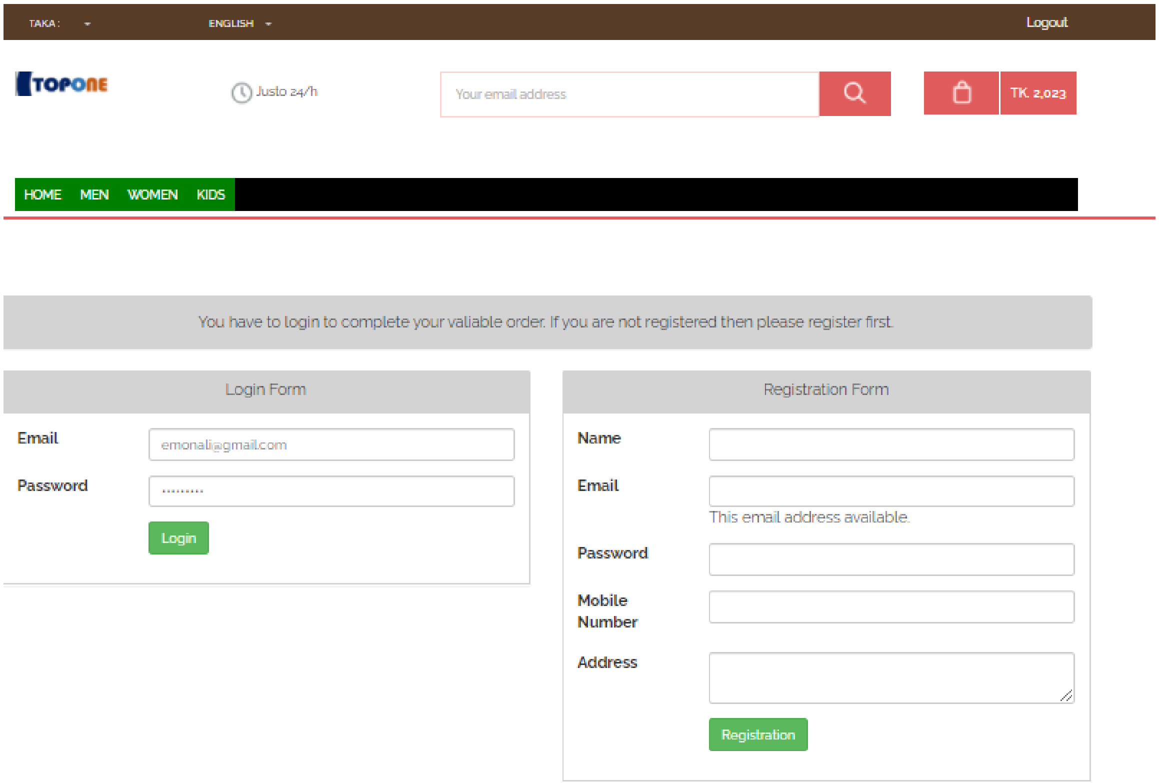
Please give us your order shipping info. If billing & shipping info are same then just process on save shipping info button.

Congratulation...Login success. Now please give us product shipping information to complete your valuable order.

Shipping Form

Name	Emon
Email	emonali@gmail.com
Mobile Number	01767099568
Address	kustia

7. Login page



You have to login to complete your valiable order. If you are not registered then please register first.

Login Form

Email: emonali@gmail.com

Password: ………

Registration Form

Name

Email

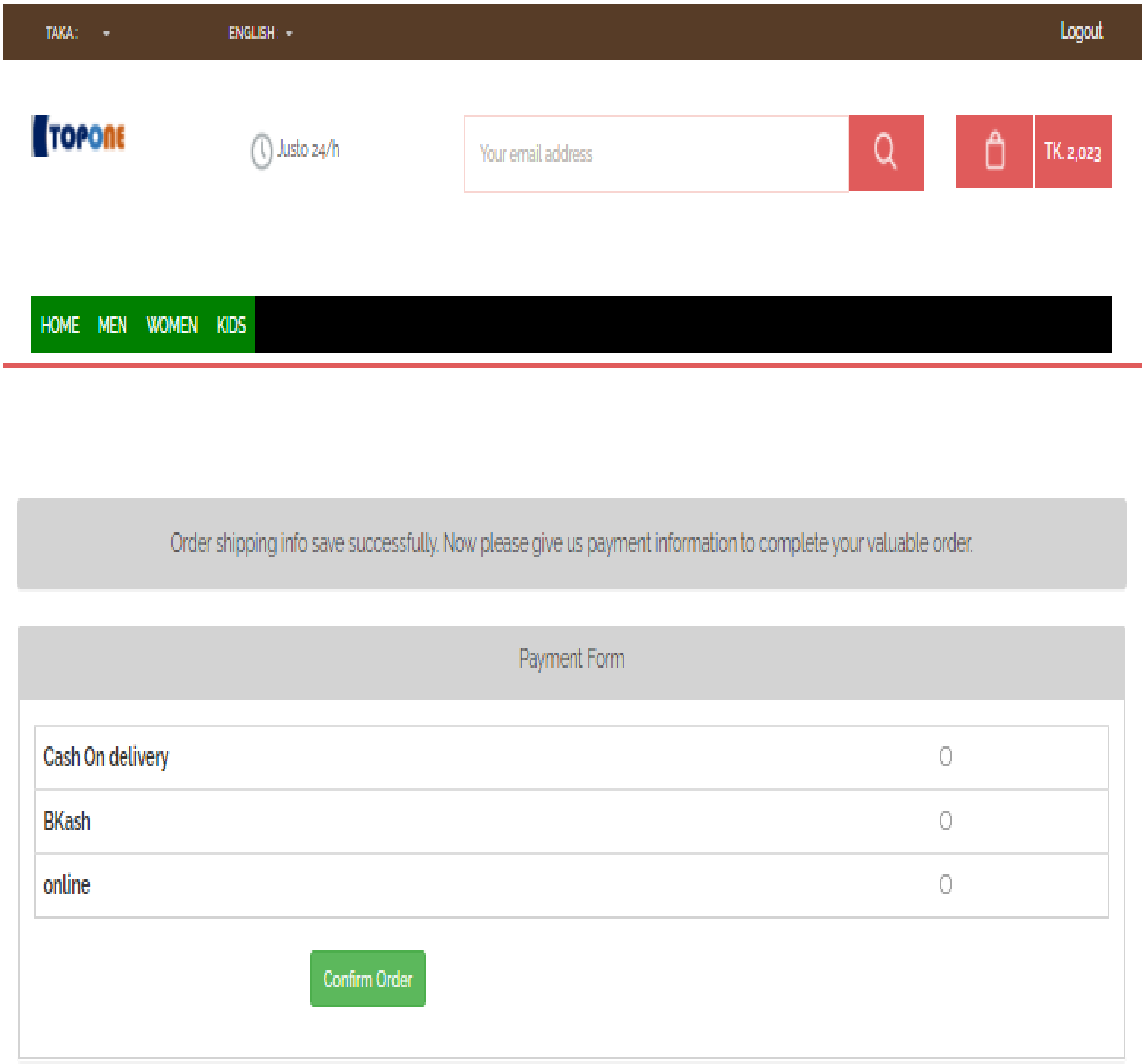
This email address available.

Password

Mobile Number

Address

8. Order and payment page

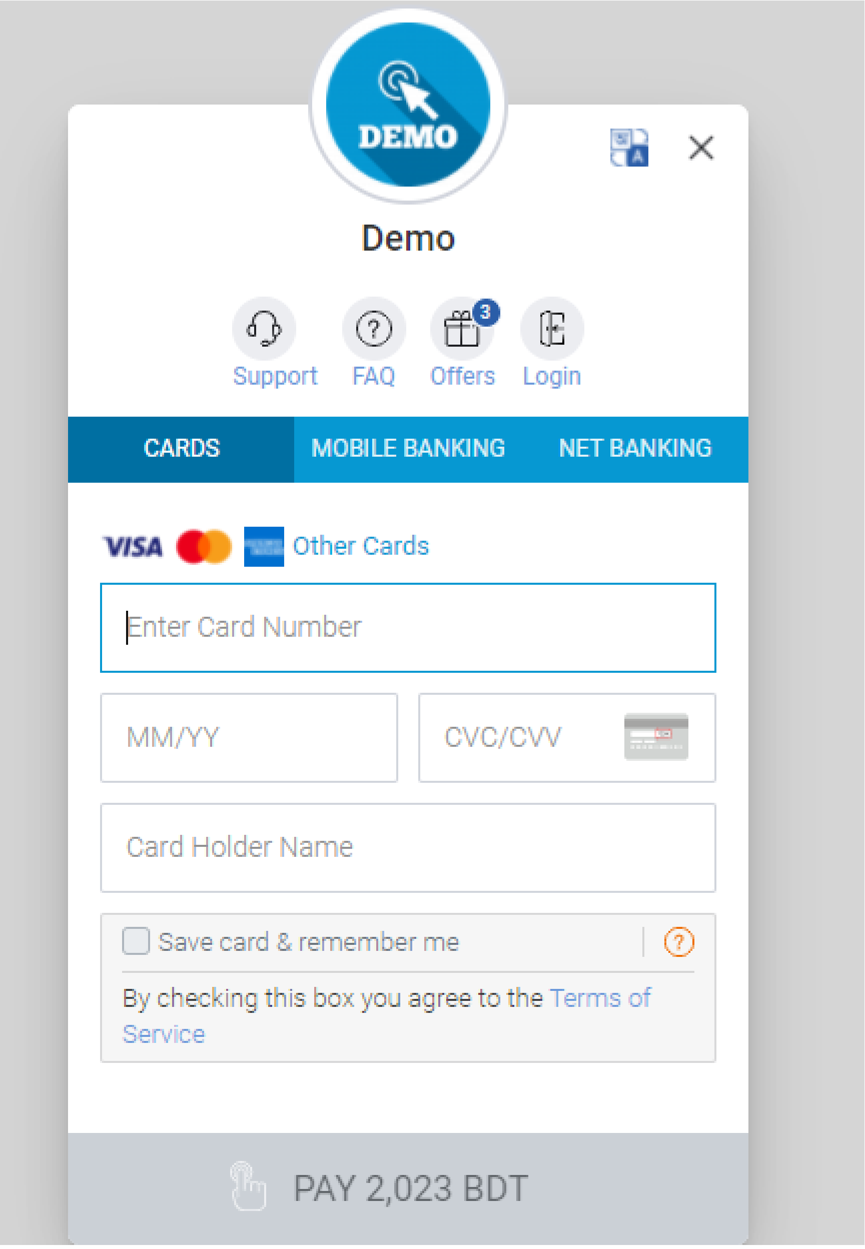


Order shipping info save successfully. Now please give us payment information to complete your valuable order.

Payment Form

Cash On delivery	○
BKash	○
online	○

9. Payment method



Demo

Support FAQ Offers Login

CARDS MOBILE BANKING NET BANKING

VISA Other Cards

Enter Card Number

MM/YY

CVC/CVV

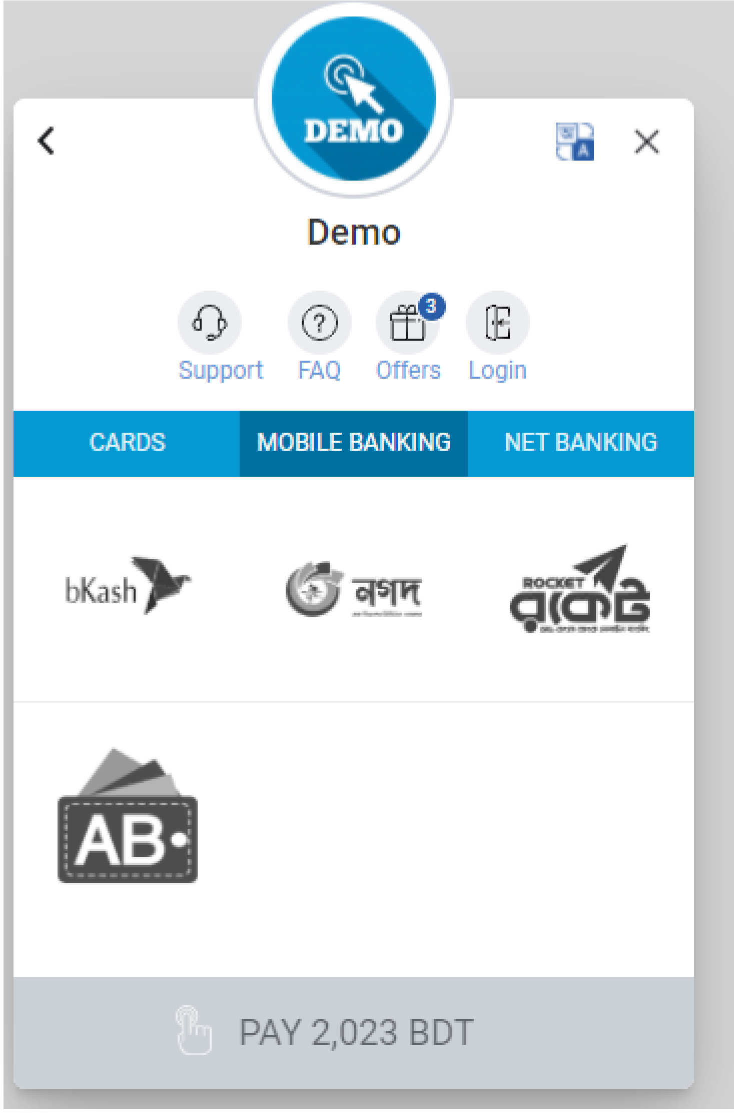
Card Holder Name

□ Save card & remember me

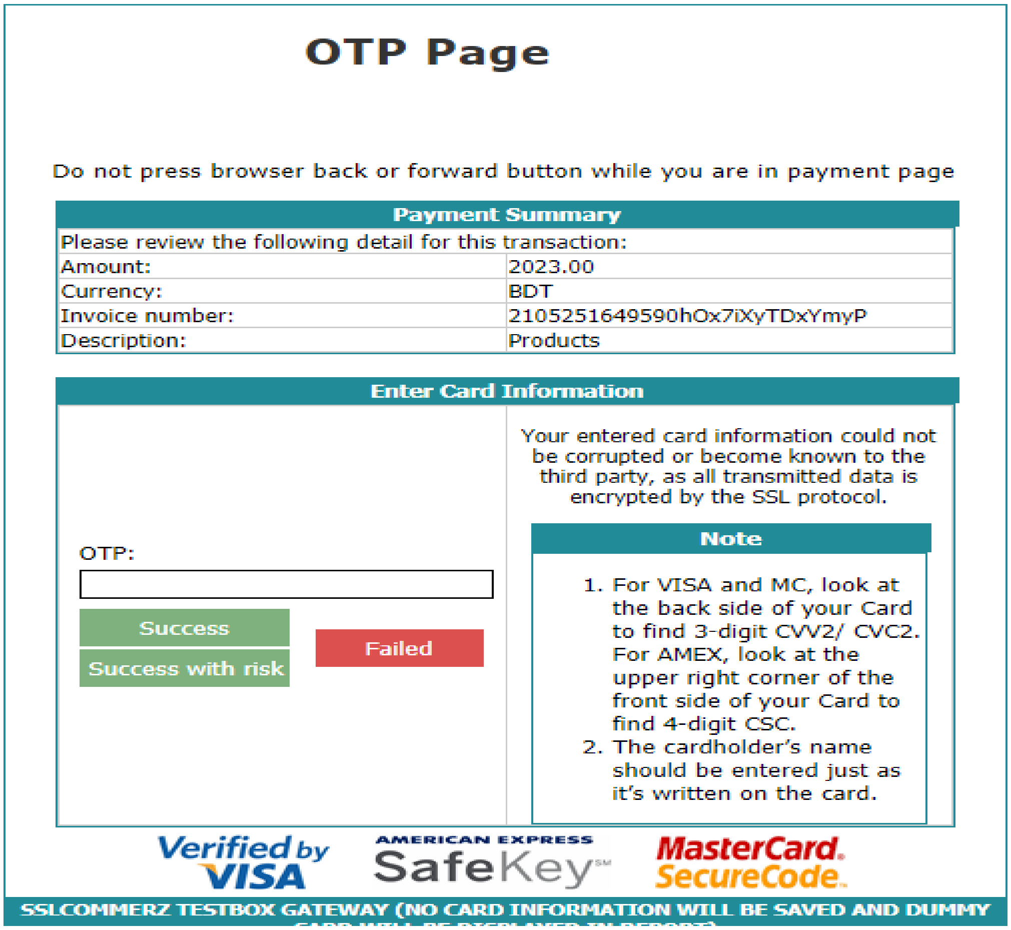
By checking this box you agree to the Terms of Service

PAY 2,023 BDT

10. Online mobile banking



11. Confirm order with online banking



OTP Page

Do not press browser back or forward button while you are in payment page

Payment Summary

Please review the following detail for this transaction:	
Amount:	2023.00
Currency:	BDT
Invoice number:	2105251649590hOx7iXyTDxYmyP
Description:	Products

Enter Card Information

Your entered card information could not be corrupted or become known to the third party, as all transmitted data is encrypted by the SSL protocol.

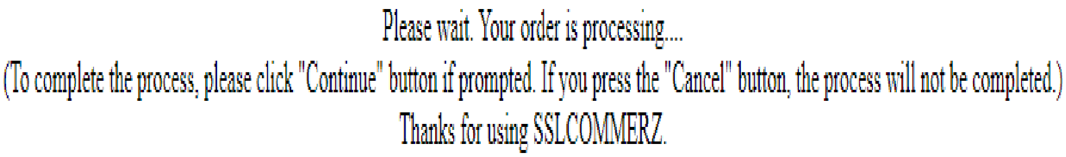
Note

1. For VISA and MC, look at the back side of your Card to find 3-digit CVV2/ CVC2. For AMEX, look at the upper right corner of the front side of your Card to find 4-digit CSC.
2. The cardholder's name should be entered just as it's written on the card.

OTP:

SSLCOMMERZ TESTBOX GATEWAY (NO CARD INFORMATION WILL BE SAVED AND DUMMY CARD WILL BE DISPLAYED IN REPORT)

12. Successful order and transaction



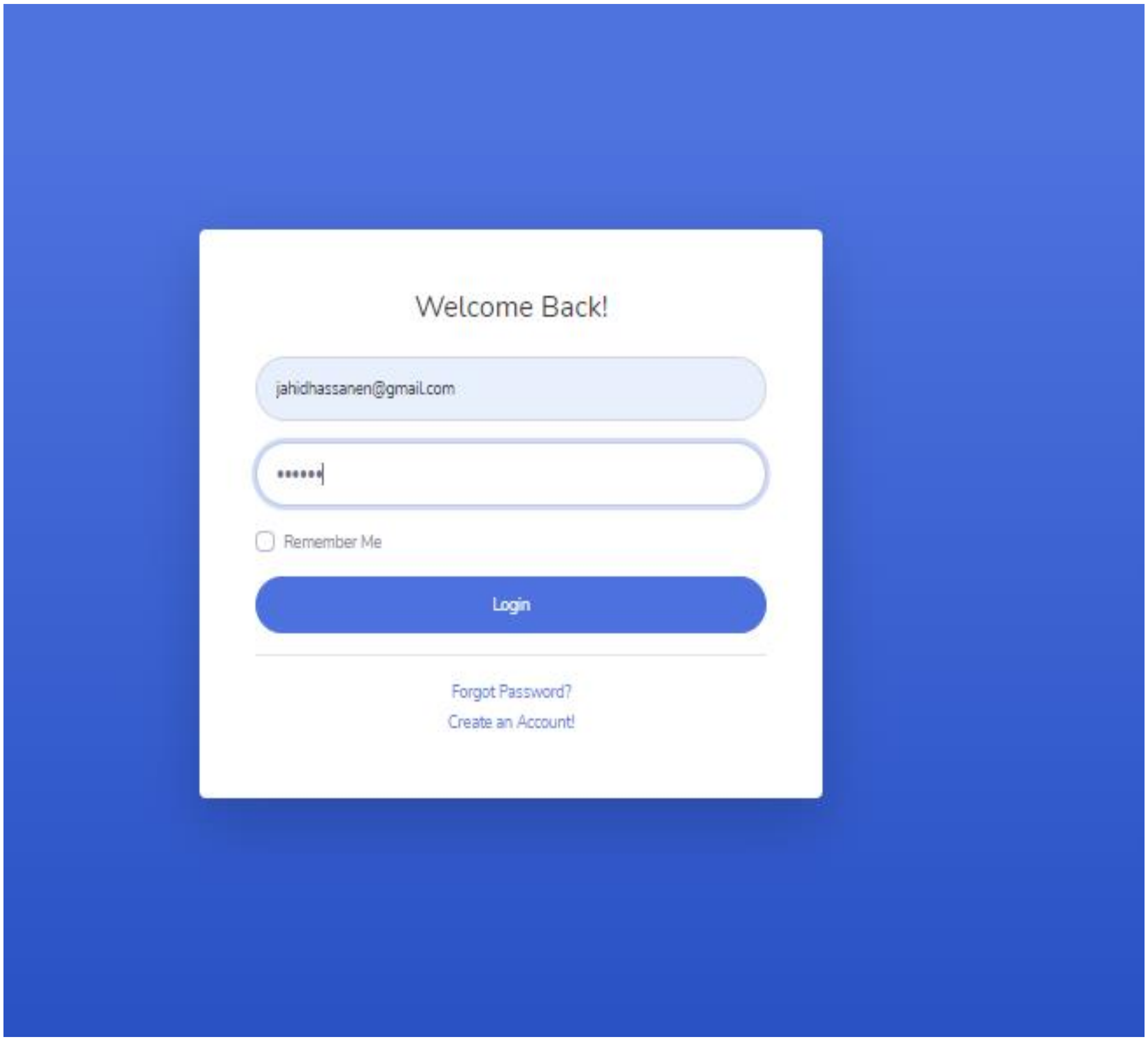
Please wait. Your order is processing....

(To complete the process, please click "Continue" button if prompted. If you press the "Cancel" button, the process will not be completed.)

Thanks for using SSLCOMMERZ.

6.2 ADMIN DASHBOARD

1. Admin login



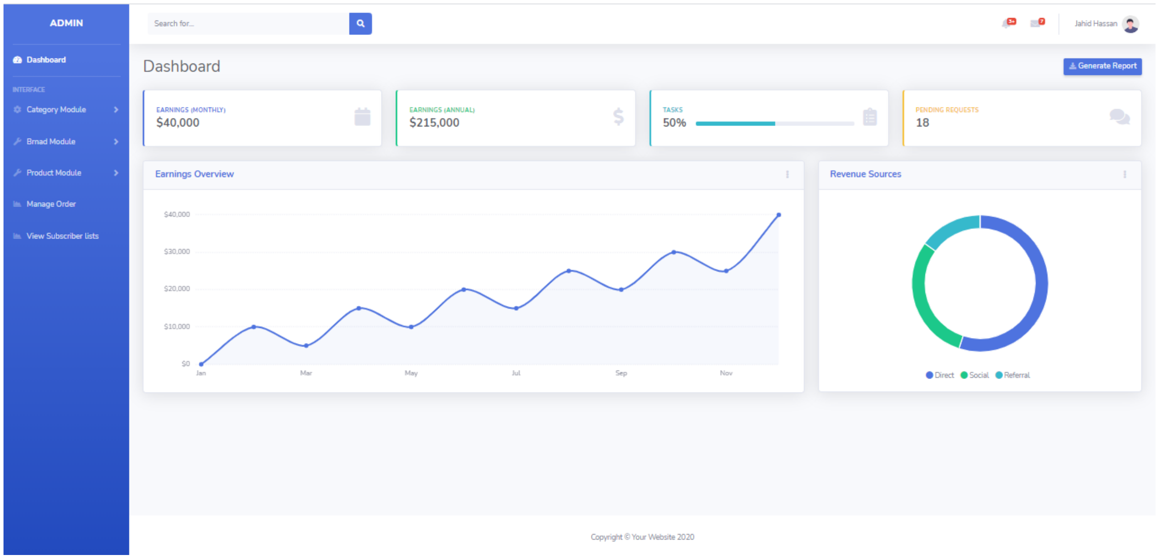
Welcome Back!

jahidhassanen@gmail.com

••••••

□ Remember Me

2. Admin dashboard



Dashboard

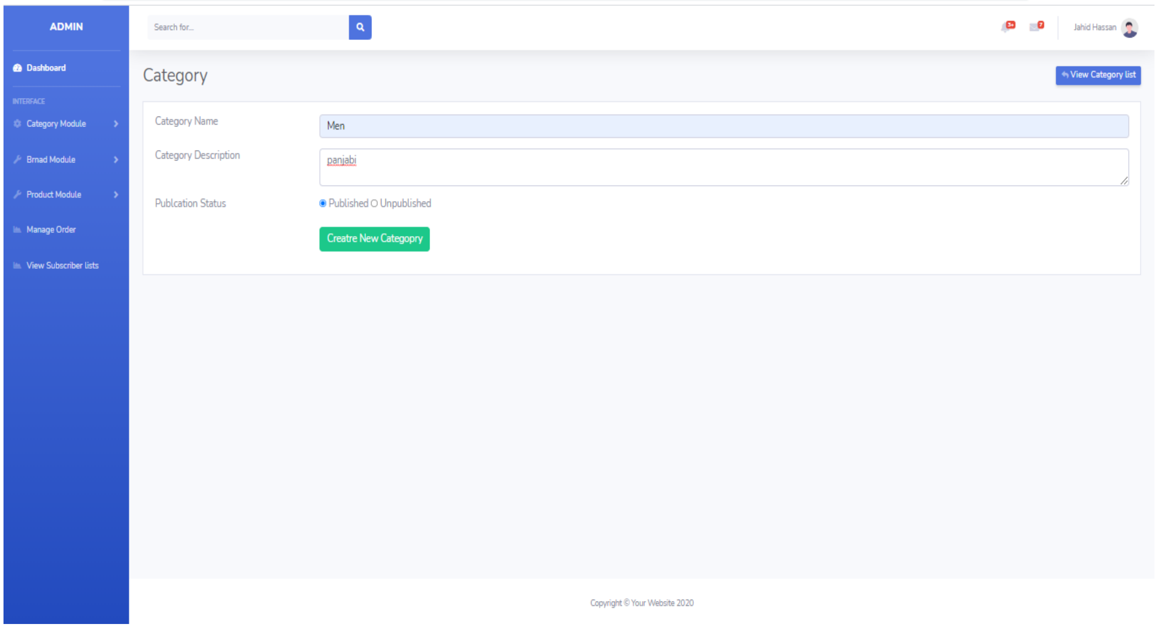
EARNINGS (MONTHLY) $40,000

EARNINGS (ANNUAL) $215,000

TASKS 50%

PENDING REQUESTS 18

3. Add category



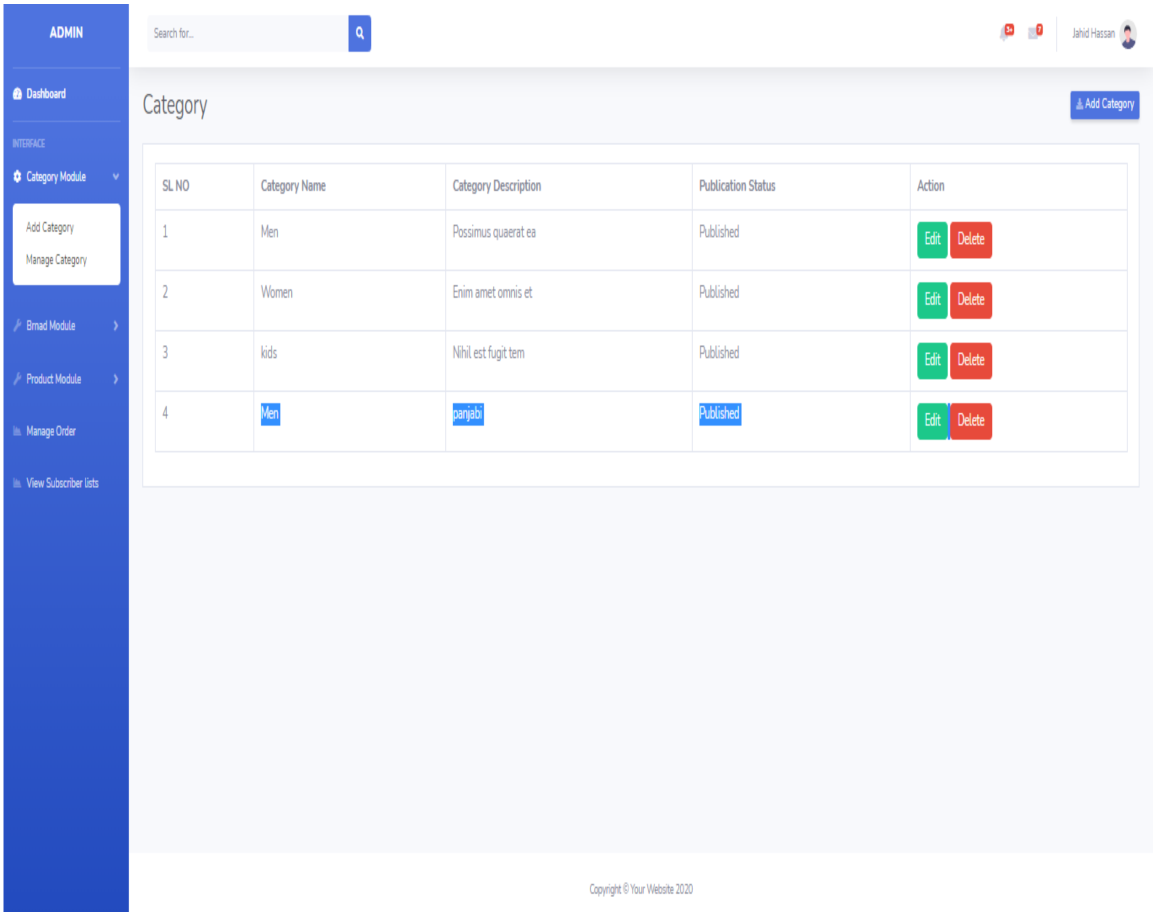
Category

Category Name: Men

Category Description: panjabi

Publcation Status: ● Published ○ Unpublished

4. Admin adds manage category

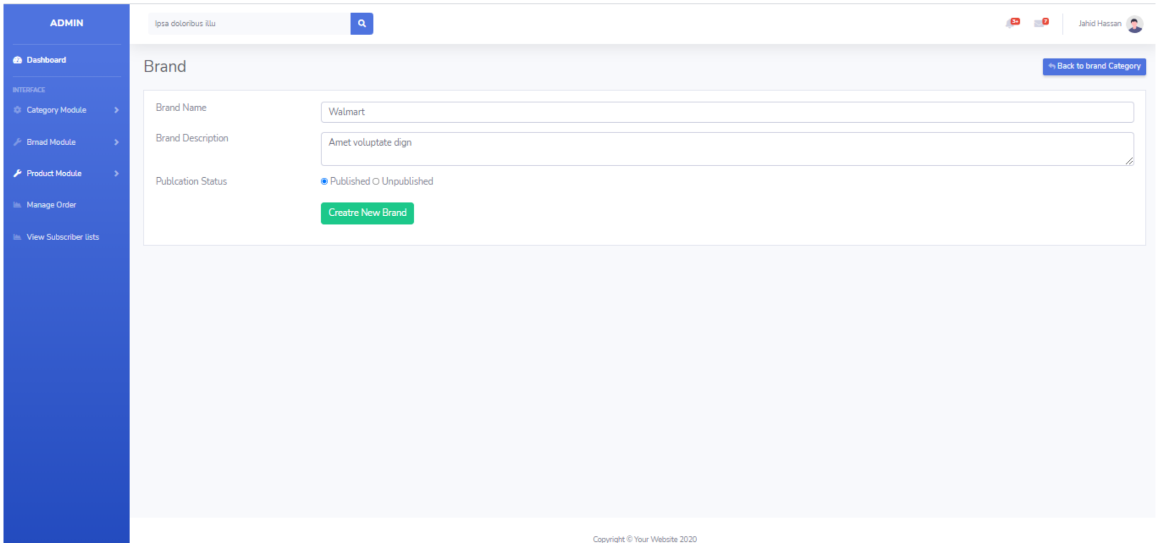


Category

SL NO	Category Name	Category Description	Publication Status	Action
1	Men	Possimus quaerat ea	Published	Edit Delete
2	Women	Enim amet omnis et	Published	Edit Delete
3	kids	Nihil est fugit tem	Published	Edit Delete
4	Men	panjabi	Published	Edit Delete

Copyright © Your Website 2020

5. Admin add Brand Category



Ipsa doloribus illu

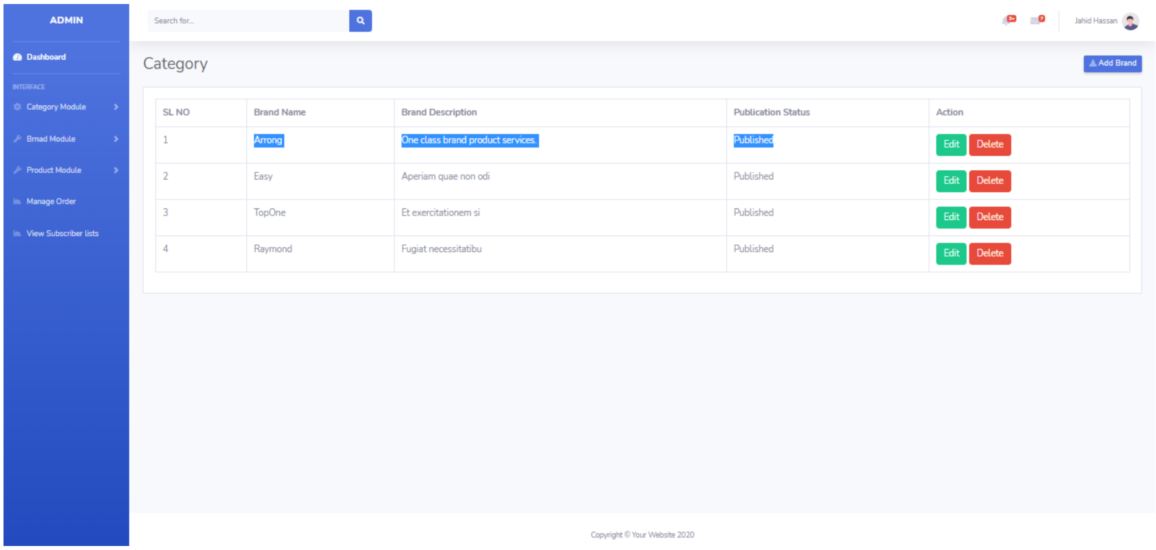
Brand

Brand Name	Walmart
Brand Description	Amet voluptate dign
Publcation Status	● Published ○ Unpublished

Creatre New Brand

Copyright © Your Website 2020

6. Admin Manage Brand

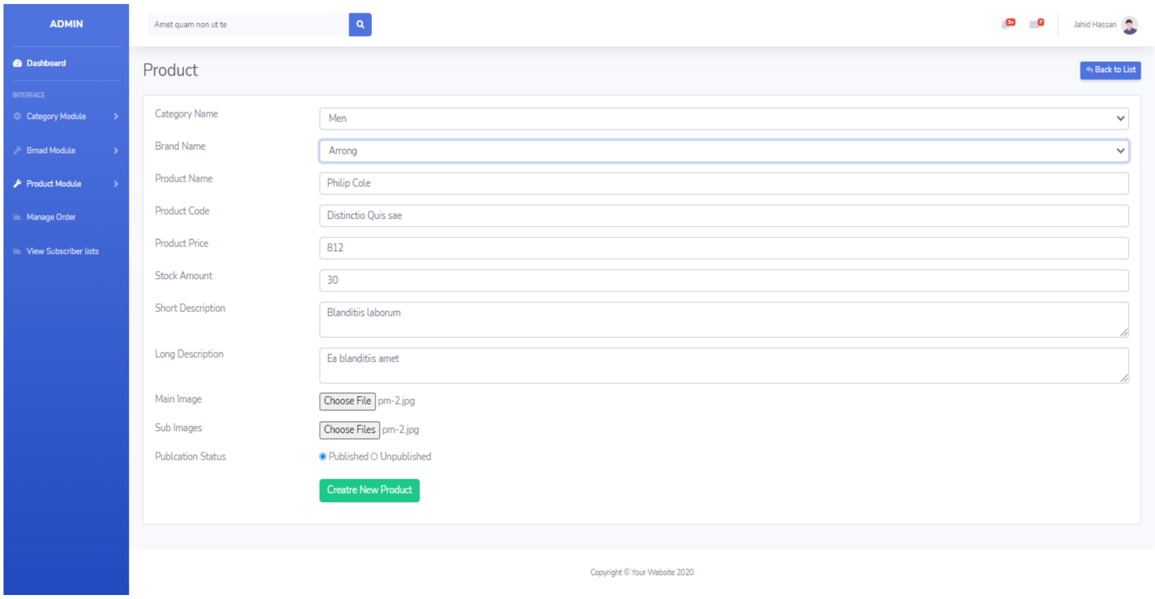


Category

SL NO	Brand Name	Brand Description	Publication Status	Action
1	Arrong	One class brand product services.	Published	Edit Delete
2	Easy	Aperiam quae non odi	Published	Edit Delete
3	TopOne	Et exercitationem si	Published	Edit Delete
4	Raymond	Fugiat necessitatibu	Published	Edit Delete

Copyright © Your Website 2020

7. Admin Add Product

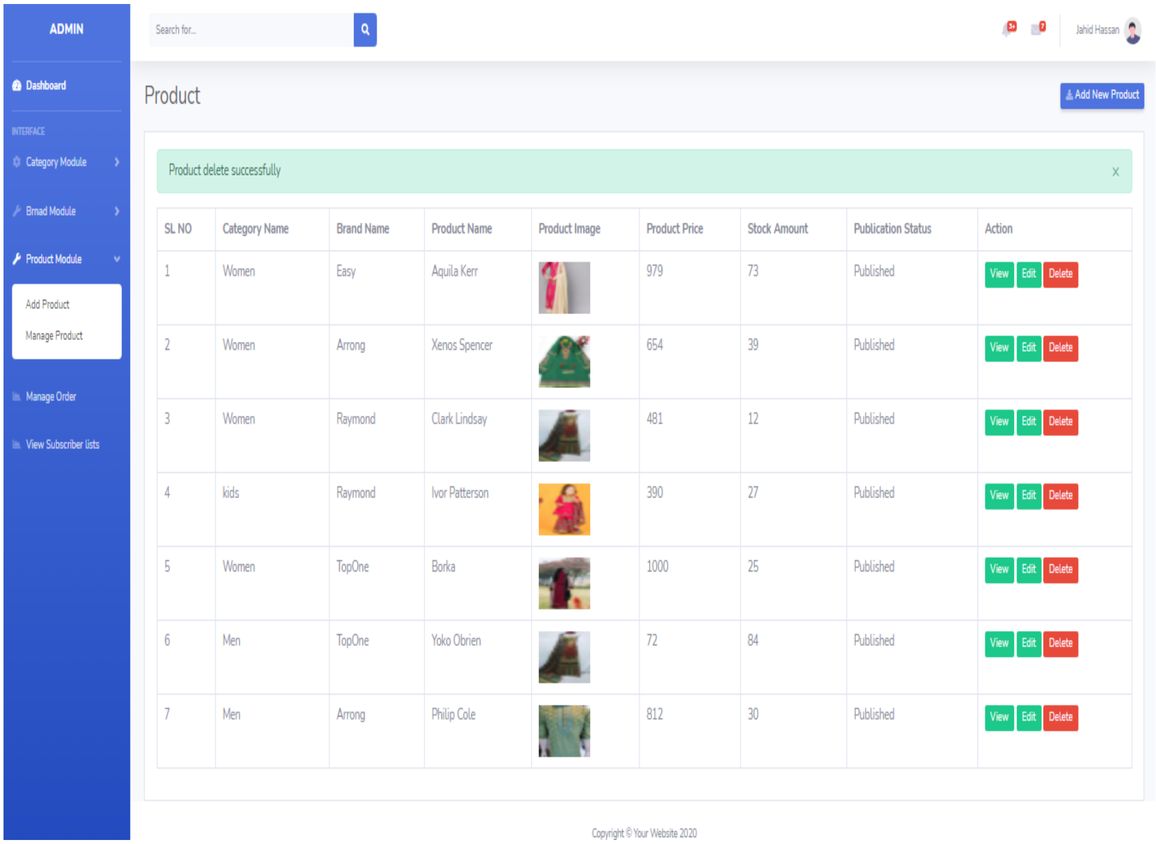


Product

Category Name	Men
Brand Name	Arrong
Product Name	Philip Cole
Product Code	Distinctio Quis sae
Product Price	812
Stock Amount	30
Short Description	Blanditiis laborum
Long Description	Ea blanditiis amet
Main Image	Choose File pm-2.jpg
Sub Images	Choose Files pm-2.jpg
Publcation Status	● Published ○ Unpublished

Creatre New Product

8. Admin Manage Product



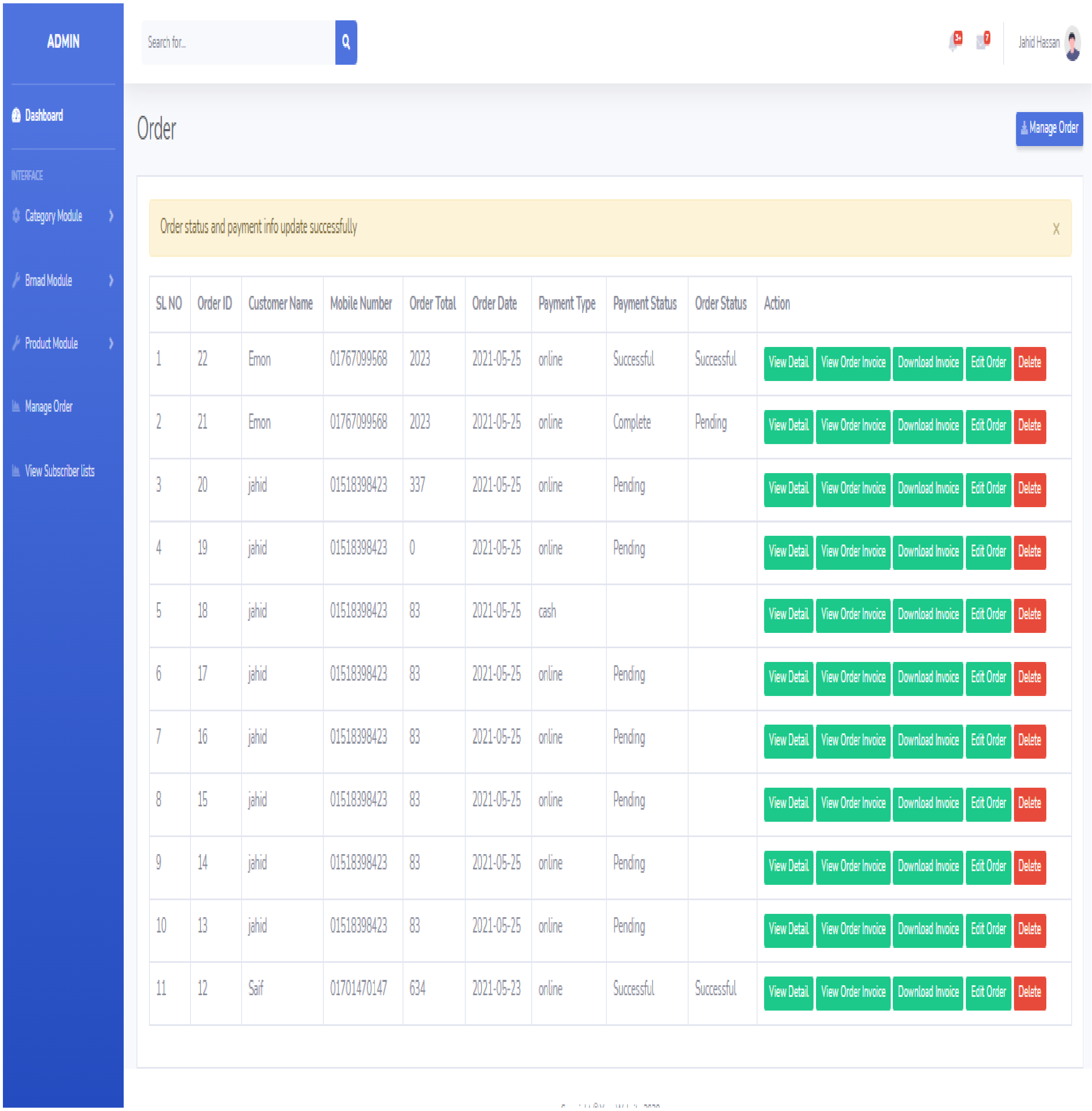
Product

Product delete successfully

SL NO	Category Name	Brand Name	Product Name	Product Image	Product Price	Stock Amount	Publication Status	Action
1	Women	Easy	Aquila Kerr		979	73	Published	View Edit Delete
2	Women	Arrong	Xenos Spencer		654	39	Published	View Edit Delete
3	Women	Raymond	Clark Lindsay		481	12	Published	View Edit Delete
4	kids	Raymond	Ivor Patterson		390	27	Published	View Edit Delete
5	Women	TopOne	Borka		1000	25	Published	View Edit Delete
6	Men	TopOne	Yoko Obrien		72	84	Published	View Edit Delete
7	Men	Arrong	Philip Cole		812	30	Published	View Edit Delete

Copyright © Your Website 2020

9. Admin panel customer order view detail



Order

Order status and payment info update successfully

SL NO	Order ID	Customer Name	Mobile Number	Order Total	Order Date	Payment Type	Payment Status	Order Status	Action
1	22	Emon	01767099568	2023	2021-05-25	online	Successful	Successful	View Detail View Order Invoice Download Invoice Edit Order Delete
2	21	Emon	01767099568	2023	2021-05-25	online	Complete	Pending	View Detail View Order Invoice Download Invoice Edit Order Delete
3	20	jahid	01518398423	337	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
4	19	jahid	01518398423	0	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
5	18	jahid	01518398423	83	2021-05-25	cash			View Detail View Order Invoice Download Invoice Edit Order Delete
6	17	jahid	01518398423	83	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
7	16	jahid	01518398423	83	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
8	15	jahid	01518398423	83	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
9	14	jahid	01518398423	83	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
10	13	jahid	01518398423	83	2021-05-25	online	Pending		View Detail View Order Invoice Download Invoice Edit Order Delete
11	12	Saif	01701470147	634	2021-05-23	online	Successful	Successful	View Detail View Order Invoice Download Invoice Edit Order Delete

7. CHAPTER-07 PROJECT SUMMARY

7.1 GIT HU LINK: https://github.com/Jahid-SWE/TopOne.com

7.2 LIMITATIONS

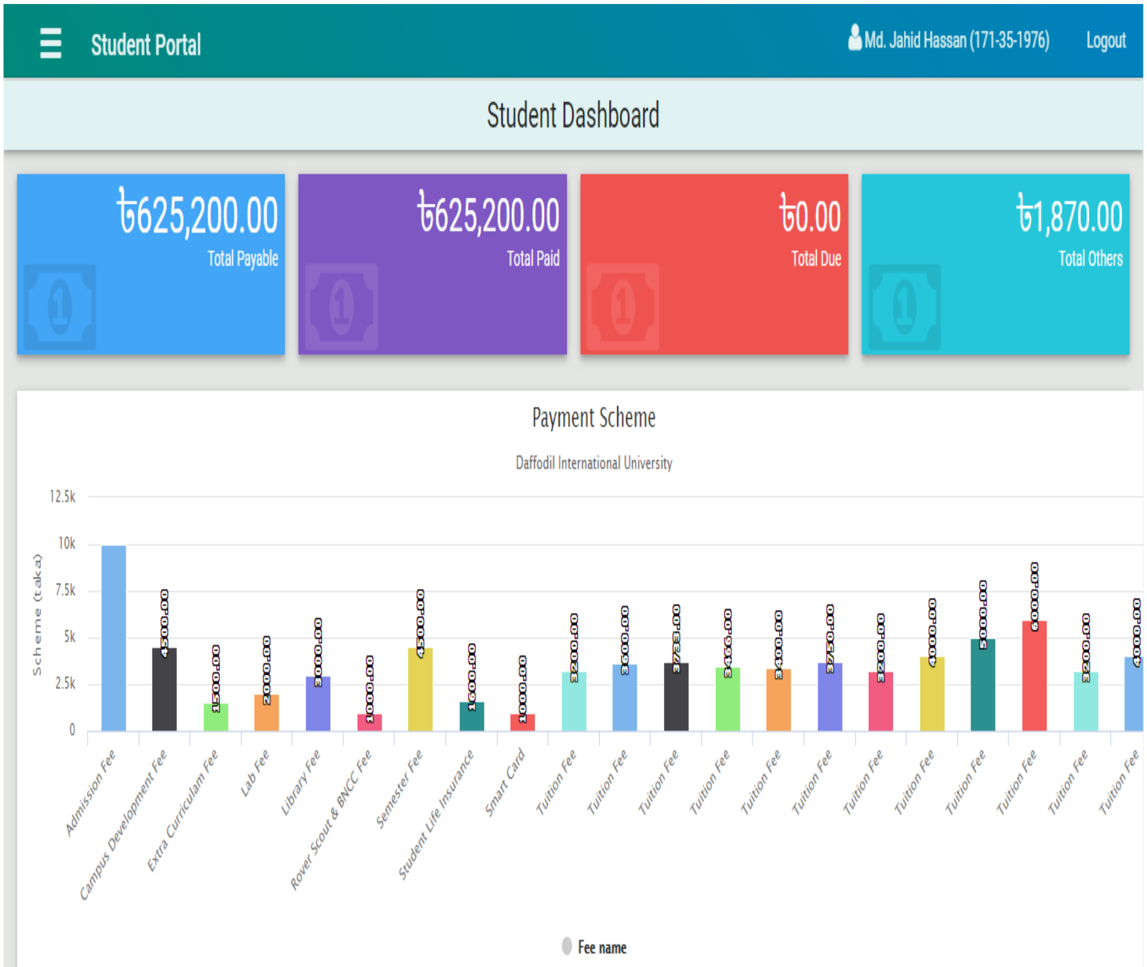
Table 7.1 Limitations of my systems

Performance	1.Less performance
Speed	1.Medium speed
Features	1. This web application has more feature as like order, payment, shipping add cart etc. This all are important for the customer and give here 2. Admin site has a lot of features between product add to order invoice.
Ability	1. More than thousand customer can hit at a time.

7.3 FUTURE WORK PLAN

- Voice searching
- AI include (Artificial intelligence)
- Complain option

Accounts clearance



Student Dashboard

৳625,200.00 Total Payable

৳625,200.00 Total Paid

৳0.00 Total Due

৳1,870.00 Total Others

Payment Scheme

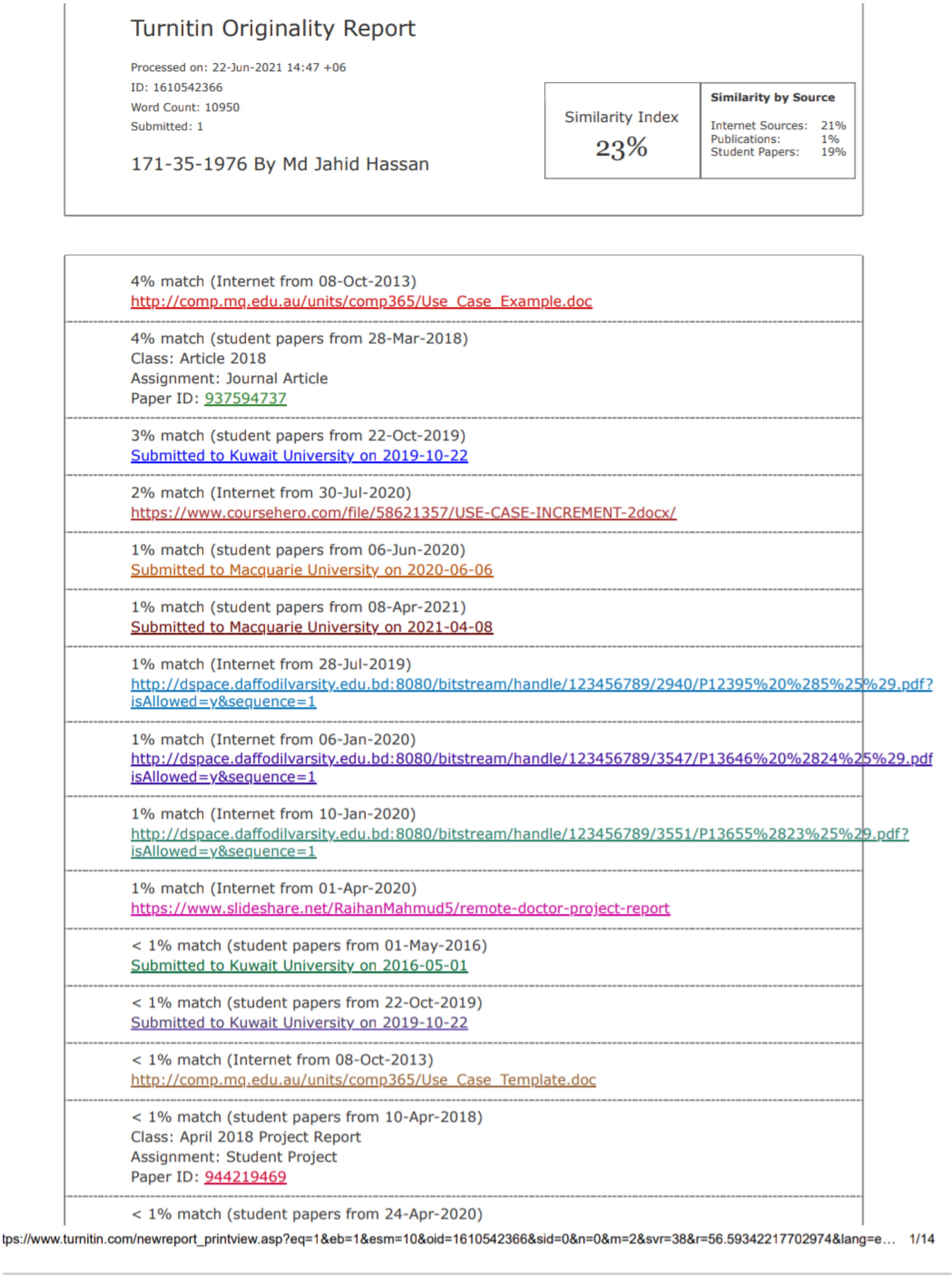
Daffodil International University

Appendix

```
namespace App\Http\Controllers;

use App\Models\Customer;
use App\Models\Order;
use App\Models\OrderDetail;
use App\Models\Payment;
use App\Models\Shipping;
use PDF;
use Illuminate\Http\Request;

class AdminOrderController extends Controller
{
    public function index()
    {
        return view('admin.order.manage', [
            'orders' => Order::orderBy('id', 'desc')->get()
        ]);
    }

    public function viewOrderDetail($id)
    {
        $order = Order::find($id);
        return view('admin.order.order-detail', [
            'order'     => $order,
            'products'  => OrderDetail::where('order_id', $order->id)-
>get(),
            'customer'  => Customer::find($order->customer_id),
            'shipping'  => Shipping::find($order->shipping_id),
            'payment'   => Payment::where('order_id', $order->id)-
>first(),
        ]);
    }

public function updateOrder(Request $request)
{
    $order = Order::find($request->id);
    $order->order_status    = $request->order_status;
    $order->payment_status  = $request->payment_status;
    $order->save();

    if($request->order_status == 'Complete')
    {
        $payment = Payment::where('order_id', $request->id)->first();
        $payment->payment_amount    = $request->amount;
        $payment->payment_date      = date('Y-m-d');
        $payment->save();
    }
    return redirect('/manage-order')->with('message', 'Order status and
payment info update successfully');
}
class CartController extends Controller
{
    public function index(Request $request)
    {
        $product = Product::find($request->id);
        Cart::add([
            'id'        => $product->id,
            'name'      => $product->name,
            'qty'       => $request->qty,
            'price'     => $product->price,
            'weight'    => 0,
            'options'   => [
                'image' => $product->image
            ]
        ]);
        return redirect('/show-cart');
    }

    public function directAddToCart($id)
    {
      $product = Product::find($id);
      Cart::add([
          'id'        => $product->id,
          'name'      => $product->name,
          'qty'       => 1,
          'price'     => $product->price,
          'weight'    => 0,
          'options'   => [
              'image' => $product->image
          ]
      ]);
      return redirect('/show-cart');
    }

    public function show()
    {
        $cartProducts = Cart::content();
        return view('front.cart.show', [
            'categories'     => Category::where('status', 1)->get(),
            'cartProducts'   => $cartProducts

        ]);
    }

    public function update(Request $request)
    {
        Cart::update($request->rowId, $request->qty);
        return redirect('/show-cart')->with('message', 'cart product
info update successfully.');
    }

    public function remove($rowId)
    {
        Cart::remove($rowId);
        return redirect('/show-cart')->with('message', 'cart product
info remove successfully.');
    }
}
class CheckoutController extends Controller
{
    public function index()
    {
        return view('front.checkout.checkout', [
            'categories'    => Category::where('status', 1)->get(),
        ]);
    }

    public function checkEmailAddress()
    {
        $email = $_GET['email'];

        $customer = Customer::where('email', $email)->first();
        if($customer)
        {
            $date = [
              'success' => false,
              'message' => 'Sorry, this email address already exists.
Please try another one.'
            ];
        }
        else
        {
            $date = [
                'success' => true,
                'message' => 'This email address available.'
            ];
        }
        return json_encode($date);
    }


    public function newCustomer(Request $request)
    {
      $customer = new Customer();
      $customer->name    = $request->name;
      $customer->email   = $request->email;
      $customer->password = bcrypt($request->password);
      $customer->mobile    = $request->mobile;
      $customer->address   = $request->address;
      $customer->save();

      Session::put('customer_id', $customer->id);
      Session::put('customer_name', $customer->name);

      return redirect('/show-shipping')->with('message',
'Congratulation...your registration process is complete. Now please
give us product shipping information to complete your valuable
order.');
    }

    public function customerLogout(Request $request)
    {
      Session::forget('customer_id');
      Session::forget('customer_name');

      return redirect('/')->with('message', 'You are successfully
logout.');
```

```
    }

    public function customerLogin(Request $request)
    {
      $customer = Customer::where('email', $request->email)->first();
      if($customer) {
        if (password_verify($request->password, $customer->password)) {
              Session::put('customer_id', $customer->id);
              Session::put('customer_name', $customer->name);
              return redirect('/show-shipping')->with('message',
'Congratulation...Login success. Now please give us product shipping
information to complete your valuable order.');
        } else {
            return redirect()->back()->with('message', 'Sorry Your
password is wrong.');
        }
      }
      else
      {
        return redirect()->back()->with('message', 'Sorry Your email
address is wrong.');
      }
    }


    public function shipping()
    {
      return view('front.checkout.shipping', [
          'categories'    => Category::where('status', 1)->get(),
          'customer'      =>
Customer::find(Session::get('customer_id'))
      ]);
    }

    public function newShipping(Request $request)
    {
      $shipping = new Shipping();
      $shipping->name   = $request->name;
      $shipping->email  = $request->email;
      $shipping->mobile   = $request->mobile;
      $shipping->address  = $request->address;
      $shipping->save();

      Session::put('shipping_id', $shipping->id);

      return redirect('/show-payment')->with('message', 'Order shipping
info save successfully. Now please give us payment information to
complete your valuable order.');
    }

    public function payment()
    {
      return view('front.checkout.payment', [
          'categories'    => Category::where('status', 1)->get(),
      ]);
    }
    public function newOrder(Request $request)
    {
      $paymentMethod = $request->payment_method;
      if($paymentMethod == 'cash')
      {
        $order = new Order();
        $order->customer_id = Session::get('customer_id');
        $order->shipping_id = Session::get('shipping_id');
        $order->order_total = Session::get('grand_total');
        $order->order_date  = date('Y-m-d');
        $order->payment_type= $paymentMethod;
        $order->save();


        $cartProducts = Cart::content();
        foreach($cartProducts as $cartProduct)
        {
          $orderDetail = new OrderDetail();
          $orderDetail->order_id      = $order->id;
          $orderDetail->product_id    = $cartProduct->id;
          $orderDetail->product_name  = $cartProduct->name;
          $orderDetail->product_image = $cartProduct->options->image;
          $orderDetail->product_price = $cartProduct->price;
          $orderDetail->product_qty   = $cartProduct->qty;
          $orderDetail->save();

          Cart::remove($cartProduct->rowId);

          $product = Product::find($cartProduct->id);
          $product->stock_amount = ($product->stock_amount -
$cartProduct->qty);
          $product->save();
        }

        $payment = new Payment();
        $payment->order_id = $order->id;
        $payment->payment_method = $paymentMethod;
        $payment->payment_amount = 0;
        $payment->payment_date = date('Y-m-d');
        $payment->save();

        Session::put('grand_total', 0);

        return redirect('/complete-order')->with('message', 'Your order
info save successfully. We will contact with you soon.');
      }
      else if($paymentMethod == 'bkash')
      {

      }
      else if($paymentMethod == 'online')
      {
        return view('front.checkout.online-payment',[
            'categories'    => Category::where('status', 1)->get(),
            'customer'      =>
Customer::find(Session::get('customer_id')),
            'tran_id'       => rand(1, 100000000),
            'payment_method'=>$paymentMethod,
        ]);
      }
    }

    public function confirmOnlinePayment(Request $request)
    {
        return $request->all();
    }

    public function completeOrder()
    {
        return view('front.checkout.complete-order',[
            'categories'    => Category::where('status', 1)->get(),
        ]);
    }
    public function pay(Request  $request)
    {
        # Here you have to receive all the order data to initate the
payment.
        # Let's say, your oder transaction informations are saving in a
table called "orders"
        # In "orders" table, order unique identity is "transaction_id".
"status" field contain status of the transaction, "amount" is the order
amount to be paid and "currency" is for storing Site Currency which
will be checked with paid currency.

        $post_data = array();
        $post_data['total_amount']= Session::get('grand_total');
        $post_data['currency'] = "BDT";
        $post_data['tran_id'] = uniqid(); // tran_id must be unique
        # CUSTOMER INFORMATION
        $post_data['cus_name'] =$request->input('cus_name');

        $post_data['cus_email'] =$request->input('cus_name');

        $post_data['cus_add1'] = $request->input('cus_add1');
        $post_data['cus_add2'] = "";
        $post_data['cus_city'] = $request->input('city');
        $post_data['cus_state'] = $request->input('state');
        $post_data['cus_postcode'] = $request->input('zip');
        $post_data['cus_country'] = $request->input('country');
        $post_data['cus_phone'] = $request->input('customer_mobile');
        $post_data['order_total']= Session::get('grand_total');
        $post_data['order_date']= date('Y-m-d');
        $post_data['customer_id'] = Session::get('customer_id');
        $post_data['shipping_id'] = Session::get('shipping_id');
        $post_data['cus_fax'] = "";

        # SHIPMENT INFORMATION
        $post_data['ship_name'] = "Store Test";
        $post_data['ship_add1'] = "Dhaka";
        $post_data['ship_add2'] = "Dhaka";
        $post_data['ship_city'] = "Dhaka";
        $post_data['ship_state'] = "Dhaka";
        $post_data['ship_postcode'] = "1000";
        $post_data['ship_phone'] = "";
        $post_data['ship_country'] = "Bangladesh";

        $post_data['shipping_method'] = "NO";
        $post_data['product_name'] = "Computer";
        $post_data['product_category'] = "Goods";
        $post_data['product_profile'] = "physical-goods";

        # OPTIONAL PARAMETERS
        $post_data['value_a'] = "ref001";
        $post_data['value_b'] = "ref002";
        $post_data['value_c'] = "ref003";
        $post_data['value_d'] = "ref004";

        #Before  going to initiate the payment order status need to
insert or update as Pending.
        $update_product = DB::table('orders')
            ->where('transaction_id', $post_data['tran_id'])
            ->updateOrInsert([
//                'name' => $post_data['cus_name'],
//                'email' => $post_data['cus_email'],
//                'phone' => $post_data['cus_phone'],
//                'address' => $post_data['cus_add1'],
                'payment_status' => 'Pending',
                'payment_type' => $request->payment_method,

                'transaction_id' => $post_data['tran_id'],
                'currency' => $post_data['currency'],
                'order_total'=>$post_data['order_total'],
                'order_date'=>$post_data['order_date'],
                'customer_id' => $post_data['customer_id'],
                'shipping_id'=> $post_data['shipping_id']
            ]);
        $id = DB::getPdo()->lastInsertId();

        $cartProducts = Cart::content();
        foreach($cartProducts as $cartProduct)
        {
            $orderDetail = new OrderDetail();
            $orderDetail->order_id      =  $id;
            $orderDetail->product_id    = $cartProduct->id;
            $orderDetail->product_name  = $cartProduct->name;
            $orderDetail->product_image = $cartProduct->options->image;
            $orderDetail->product_price = $cartProduct->price;
            $orderDetail->product_qty   = $cartProduct->qty;
            $orderDetail->save();

            Cart::remove($cartProduct->rowId);

            $product = Product::find($cartProduct->id);
            $product->stock_amount = ($product->stock_amount -
$cartProduct->qty);
            $product->save();

        }


        $sslc = new SslCommerzNotification();
        # initiate(Transaction Data , false: Redirect to SSLCOMMERZ
gateway/ true: Show all the Payement gateway here )
        $payment_options = $sslc->makePayment($post_data, 'hosted');

        if (!is_array($payment_options)) {
            print_r($payment_options);
            $payment_options = array();
        }
    }
    public function success(Request $request)
    {

//        echo "Transaction is Successful";
        $tran_id = $request->input('tran_id');
        $amount = $request->input('amount');
        $currency = $request->input('currency');
        $sslc = new SslCommerzNotification();

        #Check order status in order tabel against the transaction id
or order id.
        $order_detials = DB::table('orders')
            ->where('transaction_id', $tran_id)
            ->select('transaction_id', 'payment_status', 'currency',
'order_total')->first();

        if ($order_detials->payment_status == 'Pending') {
//            $validation = $sslc->orderValidate($tran_id, $amount,
$currency, $request->all());
            $validation = true;
            if ($validation == TRUE) {
                /*
                That means IPN did not work or IPN URL was not set in
your merchant panel. Here you need to update order status
                in order table as Processing or Complete.
                Here you can also sent sms or email for successfull
transaction to customer
                */
                $update_product = DB::table('orders')
                    ->where('transaction_id', $tran_id)
                    ->update(['payment_status' =>
'Successful','order_status'=>'Successful']);
                 return redirect()->route('home');

//                return redirect('/candidateHome')
//                    ->with('transaction','Transaction is successfully
completed');

            } else {
                /*
                That means IPN did not work or IPN URL was not set in
your merchant panel and Transation validation failed.
                Here you need to update order status as Failed in order
table.
                */
                $update_product = DB::table('orders')
                    ->where('transaction_id', $tran_id)
                    ->update(['payment_status' => 'Failed']);
                echo "validation Fail";
            }
        } else if ($order_detials->payment_status == 'Processing' ||
$order_detials->payment_status == 'Successful') {
            echo 'transaction is successfull';
            /*
             That means through IPN Order status already updated. Now
you can just show the customer that transaction is completed. No need
to udate database.
             */


//            return redirect('/candidateHome')-
>with('success','Transaction is successfully completed');
        } else
        {
            #That means something wrong happened. You can redirect
customer to your product page.
            echo "Invalid Transaction";
        }




    }
    public function fail(Request $request)
    {
        $tran_id = $request->input('tran_id');

        $order_detials = DB::table('orders')
            ->where('transaction_id', $tran_id)
            ->select('transaction_id', 'status', 'currency', 'amount')-
>first();

        if ($order_detials->status == 'Pending') {
            $update_product = DB::table('orders')
                ->where('transaction_id', $tran_id)
                ->update(['status' => 'Failed']);
            echo "Transaction is Falied";
        } else if ($order_detials->status == 'Processing' ||
$order_detials->status == 'Complete') {
            echo "Transaction is already Successful";
        } else {
            echo "Transaction is Invalid";
        }

    }

    public function cancel(Request $request)
    {
        $tran_id = $request->input('tran_id');

        $order_detials = DB::table('orders')
            ->where('transaction_id', $tran_id)
            ->select('transaction_id', 'status', 'currency', 'amount')-
>first();
```

```
        if ($order_detials->status == 'Pending') {
            $update_product = DB::table('orders')
                ->where('transaction_id', $tran_id)
                ->update(['status' => 'Canceled']);
            echo "Transaction is Cancel";
        } else if ($order_detials->status == 'Processing' ||
$order_detials->status == 'Complete') {
            echo "Transaction is already Successful";
        } else {
            echo "Transaction is Invalid";
        }


    }

    public function ipn(Request $request)
    {
        #Received all the payement information from the gateway
        if ($request->input('tran_id')) #Check transation id is posted
or not.
        {

            $tran_id = $request->input('tran_id');

            #Check order status in order tabel against the transaction
id or order id.
            $order_details = DB::table('orders')
                ->where('transaction_id', $tran_id)
                ->select('transaction_id', 'status', 'currency',
'amount')->first();

            if ($order_details->status == 'Pending') {
                $sslc = new SslCommerzNotification();
                $validation = $sslc->orderValidate($tran_id,
$order_details->amount, $order_details->currency, $request->all());
                if ($validation == TRUE) {
                    /*
                    That means IPN worked. Here you need to update
order status
                    in order table as Processing or Complete.
                    Here you can also sent sms or email for successful
transaction to customer
                    */
                    $update_product = DB::table('orders')
                        ->where('transaction_id', $tran_id)
                        ->update(['status' => 'Processing']);

                    echo "Transaction is successfully Completed";
                } else {
                    /*
                    That means IPN worked, but Transation validation
failed.
                    Here you need to update order status as Failed in
order table.
                    */
                    $update_product = DB::table('orders')
                        ->where('transaction_id', $tran_id)
                        ->update(['status' => 'Failed']);

                    echo "validation Fail";
                }

            } else if ($order_details->status == 'Processing' ||
$order_details->status == 'Complete') {

                    #That means Order status already updated. No need
to udate database.

                    echo "Transaction is already successfully
Completed";
            } else {
                #That means something wrong happened. You can redirect
customer to your product page.

                echo "Invalid Transaction";
            }
        } else {
            echo "Invalid Data";
        }
    }
}
```



Turnitin Originality Report

Processed on: 22-Jun-2021 14:47 +06
ID: 1610542366
Word Count: 10950
Submitted: 1

171-35-1976 By Md Jahid Hassan

Similarity Index	Similarity by Source	
23%	Internet Sources:	21%
	Publications:	1%
	Student Papers:	19%

4% match (Internet from 08-Oct-2013)
http://comp.mq.edu.au/units/comp365/Use_Case_Example.doc

4% match (student papers from 28-Mar-2018)
Class: Article 2018
Assignment: Journal Article
Paper ID: 937594737

3% match (student papers from 22-Oct-2019)
Submitted to Kuwait University on 2019-10-22

2% match (Internet from 30-Jul-2020)
https://www.coursehero.com/file/58621357/USE-CASE-INCREMENT-2docx/

1% match (student papers from 06-Jun-2020)
Submitted to Macquarie University on 2020-06-06

1% match (student papers from 08-Apr-2021)
Submitted to Macquarie University on 2021-04-08

1% match (Internet from 28-Jul-2019)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2940/P12395%20%285%25%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 06-Jan-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3547/P13646%20%2824%25%29.pdf
isAllowed=y&sequence=1

1% match (Internet from 10-Jan-2020)
http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3551/P13655%2823%25%29.pdf?isAllowed=y&sequence=1

1% match (Internet from 01-Apr-2020)
https://www.slideshare.net/RaihanMahmud5/remote-doctor-project-report

< 1% match (student papers from 01-May-2016)
Submitted to Kuwait University on 2016-05-01

< 1% match (student papers from 22-Oct-2019)
Submitted to Kuwait University on 2019-10-22

< 1% match (Internet from 08-Oct-2013)
http://comp.mq.edu.au/units/comp365/Use_Case_Template.doc

< 1% match (student papers from 10-Apr-2018)
Class: April 2018 Project Report
Assignment: Student Project
Paper ID: 944219469

< 1% match (student papers from 24-Apr-2020)

tps://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=10&oid=1610542366&sid=0&n=0&m=2&svr=38&r=56.59342217702974&lang=e… 1/14

References

IAEME Publication. (2021, April 4). *DATA SCIENCE AND MACHINE LEARNING APPROACH TO IMPROVE E-COMMERCE SALES PERFORMANCE ON SOCIAL WEB*. Retrieved May 15, 2021, from Academia: https://www.academia.edu/49000248/DATA_SCIENCE_AND_MACHINE_LEARNING_APPROACH_TO_IMPROVE_E_COMMERCE_SALES_PERFORMANCE_ON_SOCIAL_WEB

Roozemond, B. (2020, 10 31). How draw.io diagrams help you tackle the challenges of Covid-19. Retrieved from draw.io: https://drawio-app.com/how-drawiodiagrams-help-you-tackle-the-challenges-of-covid-19/ Tryfanava, D. (2020, 7 28). Functional vs Non-functional Requirements:

Examples and Types. Retrieved from vironIT: https://vironit.com/functional-vs-nonfunctional-requirements

Bartlett, J. (2020, 8 14). How to Write Test Cases for Software (with a sample). Retrieved from TestLodge: https://blog.testlodge.com/how-to-write-test-casesfor-software-with-sample/

Mark. (2010, 6 3). Systems Analysis and Design. Retrieved from Mark's IT Blog