

SMARTCITIZEN

This Project report has been submitted in fulfillment of the requirements for the Degree of
Bachelor of Science in Software Engineering



Submitted by

Akash Sheikh

ID: 171-35-2063

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

Supervised by:

Mr. K. M. Imtiaz-Ud-Din

Assistant Professor

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

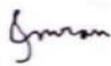
31 May, 2021

Approval

This **Project/Thesis** titled “**SmartCitizen**”, submitted by **Akash Sheikh, ID: 171-35-2063** 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 B.Sc. in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Chairman



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

Internal Examiner 1



Mr. Md. Shohel Arman
Lecturer (Senior Scale)
Faculty of Science and Information Technology
Department of Software Engineering
Daffodil International University

Internal Examiner 2



Asif Khan Shakir
Lecturer (Senior Scale)
Faculty of Science and Information Technology
Department of Software Engineering
Daffodil International University

External Examiner



Dr. Md. Nasim Akhter
Professor
Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology,
Gazipur

PROJECT DECLARATION

The project entitled “**Akash Sheikh**” is done under the supervision **Mr. K. M. Imtiaz-Ud- Din Assistant Professor**, Department of Software Engineering, Daffodil International University. I declare that this project is my original work for the degree of B.Sc. in Software Engineering and that neither the whole work nor any part has been submitted for another degree in this or any other university.

Submitted by

Akash Sheikh

Akash Sheikh

Id: 171-35-2063

Batch: 2th

Department of Software Engineering

Daffodil International University

Certified by:

Imtiaz

Mr. K. M. Imtiaz-Ud-Din

Assistant Professor

Department of Software Engineering

Daffodil International University

Acknowledgement

I am thankful to Allah for giving me the opportunity to complete this project work and learn a lot. I am thanked to my supervisor, Mr. K. M. Imtiaz-Ud-Din, for providing careful guidance starting from selecting the project scope to successfully finalizing the project work. Finally, I like to express my gratitude to Professor Dr. Touhid Bhuiyan, head of the Software Engineering faculty, for inspiring us in all means.

Abstract

This multi-dimensional lifestyle design & developed for the ultra-modern civilization Where nothing could be impossible. That may connect millions of web traffic and perform as a entrance gateway through the world wide web. SmartCitizen alone is not a complete platform. However, SmartCitizen provides us a huge amount of platform. Such like DokanBari built for SmartCitizen.

TABLE OF CONTANT

Approval.....	i
PROJECT DECLARATION	ii
Acknowledgement.....	iii
Abstract.....	iv
TABLE OF CONTANT.....	v
LIST OF FIGURES	ix
LIST OF TABLES	x
Chapter 1 : Introduction	11
1.1 Project Overview	11
1.2 Project Purpose	11
1.2.1 Background	11
1.2.2 Benefits & Beneficiaries.....	11
1.2.3 Goals	12
1.3 Stakeholders.....	12
1.4 Proposed System Model (Block Diagram)	13
1.5 Project Schedule.....	18
1.5.1 Gant Chart	18
1.5.2 Release Plan / Milestone	18
Chapter 2: Software Requirement Specification.....	19
2.1 Functional Requirements.....	19
2.2 Non-Functional Requirements.....	20
2.3 Data Requirements.....	20
2.4 Performance Requirements.....	20
2.4.1 Speed and Latency Requirements	20
2.4.2 Precision or Accuracy Requirements	20
2.4.3 Capacity Requirements	20
2.5 Dependability Requirements	20

2.5.1	Reliability Requirements.....	20
2.5.2	Availability Requirements	20
2.3	Functional Requirement Mapping.....	21
Chapter 3: System Analysis		22
3.1	Use Case Diagram for SmartCitizen.....	22
3.2	Use Case Diagram for DokanBari.....	23
3.3	Identifying Use Cases.....	24
3.4	Identifying Use Cases Description	25
3.5	Activity Diagram	36
3.6	System Sequence Diagram (for each use case)	42
Chapter 4: SYSTEM DESIGN SPECIFICATION		47
4.1	Entity Relationship Diagram	47
4.2	Class Diagram	48
4.3	Development of tools and Technology	49
4.3.1	User Interface Technology	49
4.3.2	Implementation Tools & Platforms	49
Chapter 5: System Testing		50
5.1	Testing Features	50
5.1.1	Features to be tested.....	50
5.1.2	Features to be tested.....	50
5.2	Testing Strategies	50
5.2.1	Testing Approach.....	50
5.2.2	Pass/Fail Criteria.....	51
5.2.3	Suspension and Resumption.....	51
5.2.4	Testing Schedule.....	52
5.3	Test case	53
5.3.1	Test case 1.....	53
5.3.2	Test Case: 02	54
5.3.3	Test Case: 03	55
5.3.4	Test Case: 04	56
5.3.5	Test case 05.....	57

5.3.6. Test Case: 09	58
5.3.7. Test Case: 07	59
5.3.8. Test Case: 08	60
CHAPTER 6: USER MANUAL	61-72
CHAPTER 7:.....	73
7.1. GitHub Link:.....	73
7.2. Limitations:	73
7.3. Obstacles & Achievements:	73
7.4. Future Scope	73

REFERENCES	74-75
Plagiarism Result.....	

LIST OF FIGURES

1.4.1	Block Diagram for SmartCitizen Admin	13
1.4.2	Block Diagram SmartCitizen web Traffic	13
1.5.1	Gant Chart	16
1.5.2	Milestone	16
3.1	Use Case Diagram for SmartCitizen	22
3.2	DokanBari Use Case Diagram	23
3.4.1	Registration	34
3.4.2	Login	35
3.4.3	Home	36
3.4.4	Profile	37
3.4.5	Mange shop	38
3.4.6	Mange Products	38
3.4.7	Orders	39
3.4.8	Cart	40
3.4.9	Payment	41
3.9.1	Registration	42
3.9.2	Login	43
3.9.3	Orders Customer	44
3.9.2	Shop's response	45
3.9.2	Delivery Confirmation	46
4.1	ER Diagram	47
4.2	Class Diagram	48

LIST OF Tables

2.3	Functional Requirement Mapping	21
3.4	Identifying Use Cases Description	25
3.3.1	use case description login	26
3.3.2	use case description home	27
3.3.3	use case description profile	28
3.3.4	use case description mange shop	29
3.3.5	use case description manage product	30
3.3.6	use case description orders	31
3.3.7	use case description cart	32
3.3.8	use case description payment	33
5.2.4	Testing Schedule	52
5.3.1	Test Case 01: Registration	53
5.3.2	Test Case 02: Login	54
5.3.3	Test Case 03: Add to cart	55
5.3.4	Test Case 04: Place Order	56
5.3.5	Test Case 05: Approve Vendor	57
5.3.6	Test Case 06: Approve rider	58
5.3.7	Test Case 07: Confirm Delivery	59
5.3.8	Test Case 08: Logout	60

Chapter 1 : Introduction

1.1 Project Overview

DokanBari is a web application which connecting customer for any kind of shopping purpose. Vendor and Admin can update shop and add products. This can be used in any kind of business purpose where vendor can sell their own products to customer. Admin will have the option to accept or reject shop and products by checking info.

1.2 Project Purpose

To reduce the gap between Vendor and Customer. And make the way to serve quality full products. Where Customer can buy good products. To reduce deception from another online marketplace.

1.2.1 Background

Client want to start a new thing like business. But most of the time people do not know how to start and where to start from. Some time they take wrong dissection. Who can help them and provide them to take good or better suggestions? This system bridges the gap between Vendor and Customer.

1.2.2 Benefits & Beneficiaries

Benefits:

- Easy to create an online shop
- Easy to add products under the shop
- Easy to manage orders
- Easy to get more authorized vendors
- Easy to buy quality full products
- Easy to accept or reject shop and vendor

Beneficiaries:

- General people (Customer)
- Expert people (Admin)
- Expert people (Vendor)
- Expert people (Rider)

1.2.3 Goals

The Goals of DokanBari is very simple. This project Goals are:

- ✓ To reduce the gap between vendor and customer.
- ✓ To serve them quality full products.
- ✓ To reduce deception from another online marketplace.

1.3 Stakeholders

- Admin
- Vendor
- Rider
- Customer

1.4 Proposed System Model (Block Diagram)

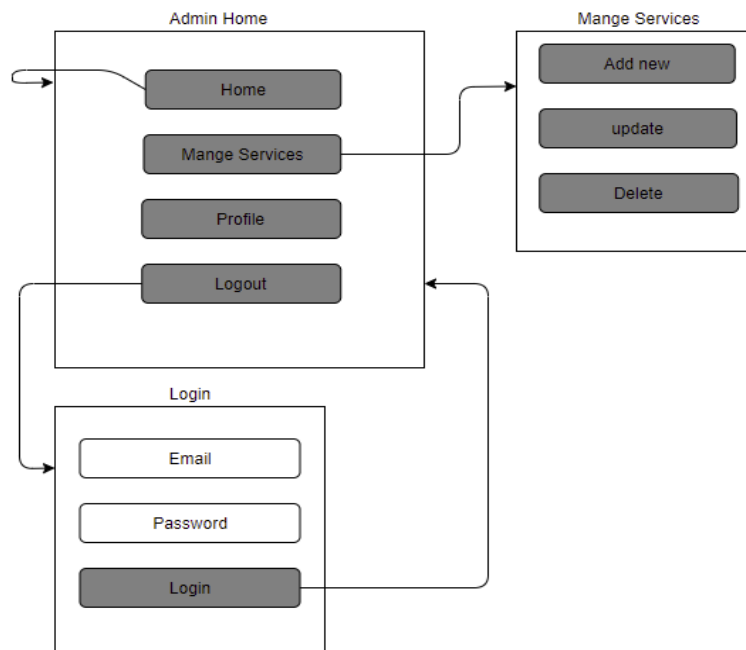


Fig: 1.4.1 Block Diagram for SmartCitizen Admin

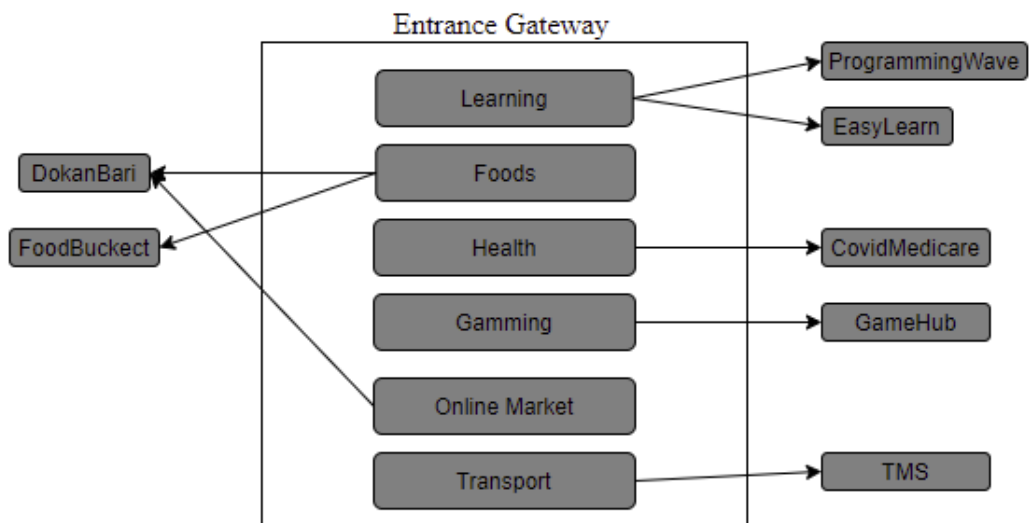


Fig: 1.4.2 Block Diagram for Smartcitizen web Traffic

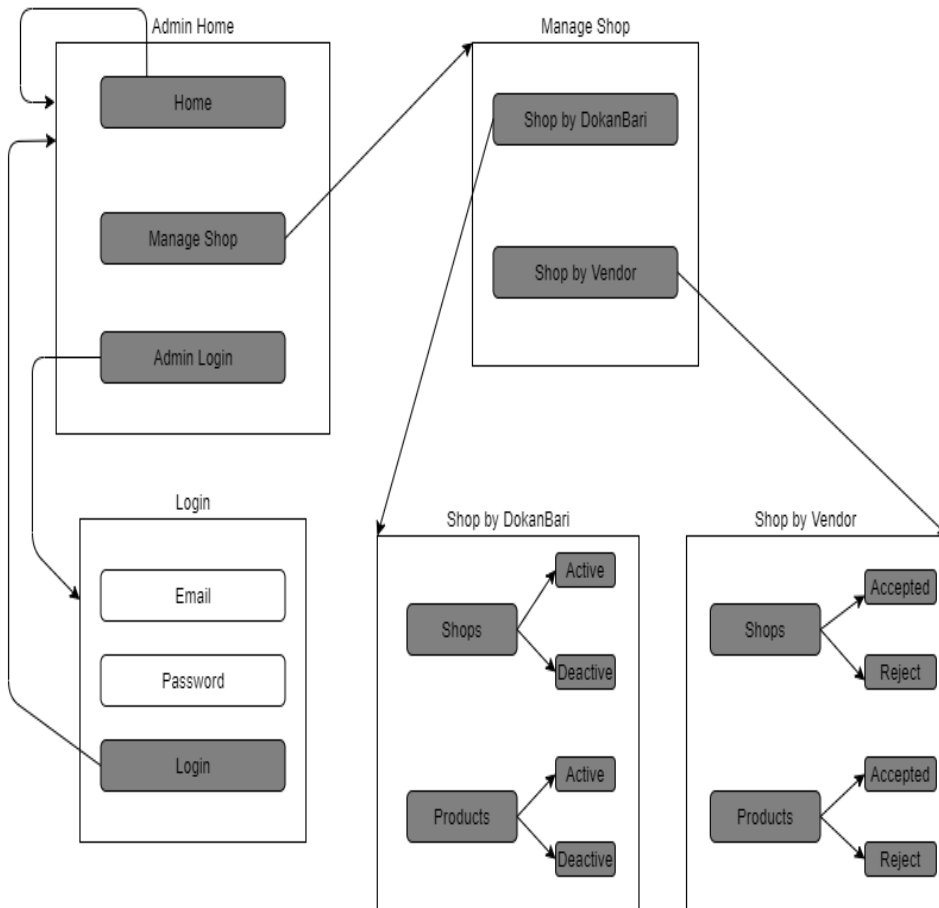


Fig: 1.4.3 Block Diagram for DokanBari Admin

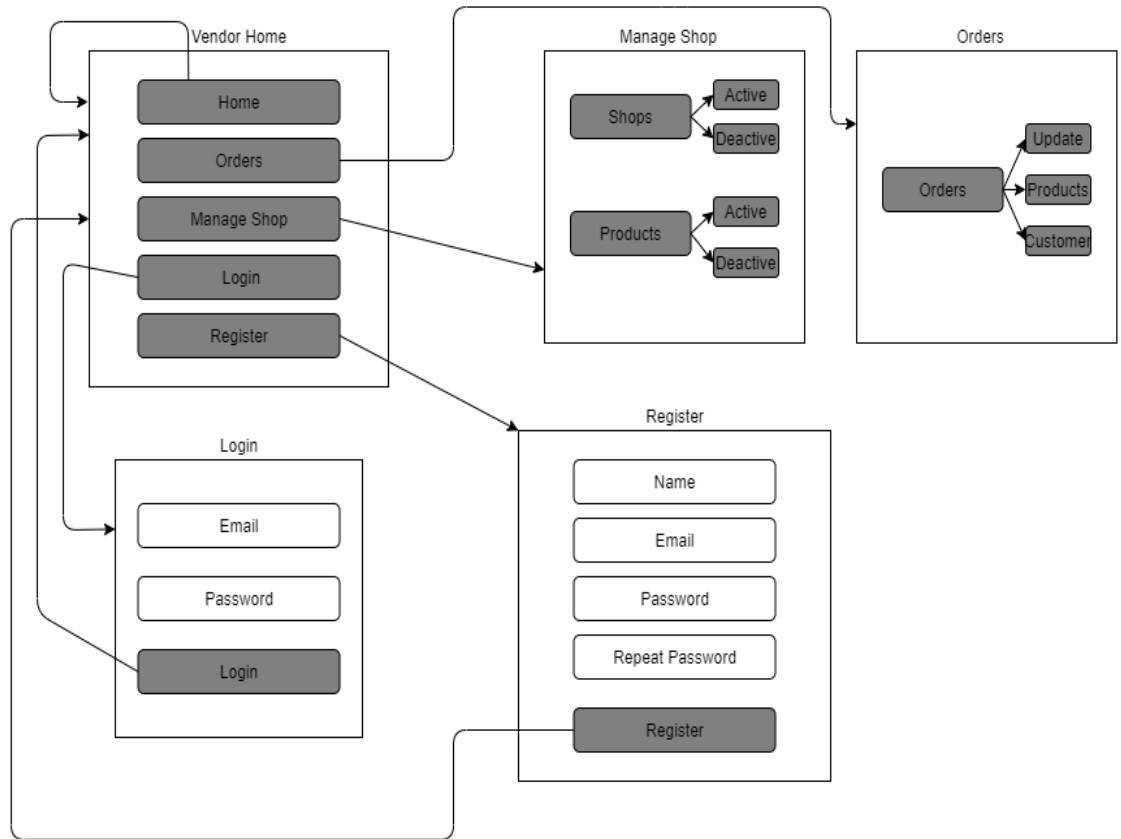


Fig: 1.4.4 Block Diagram for DokaBari Vendor

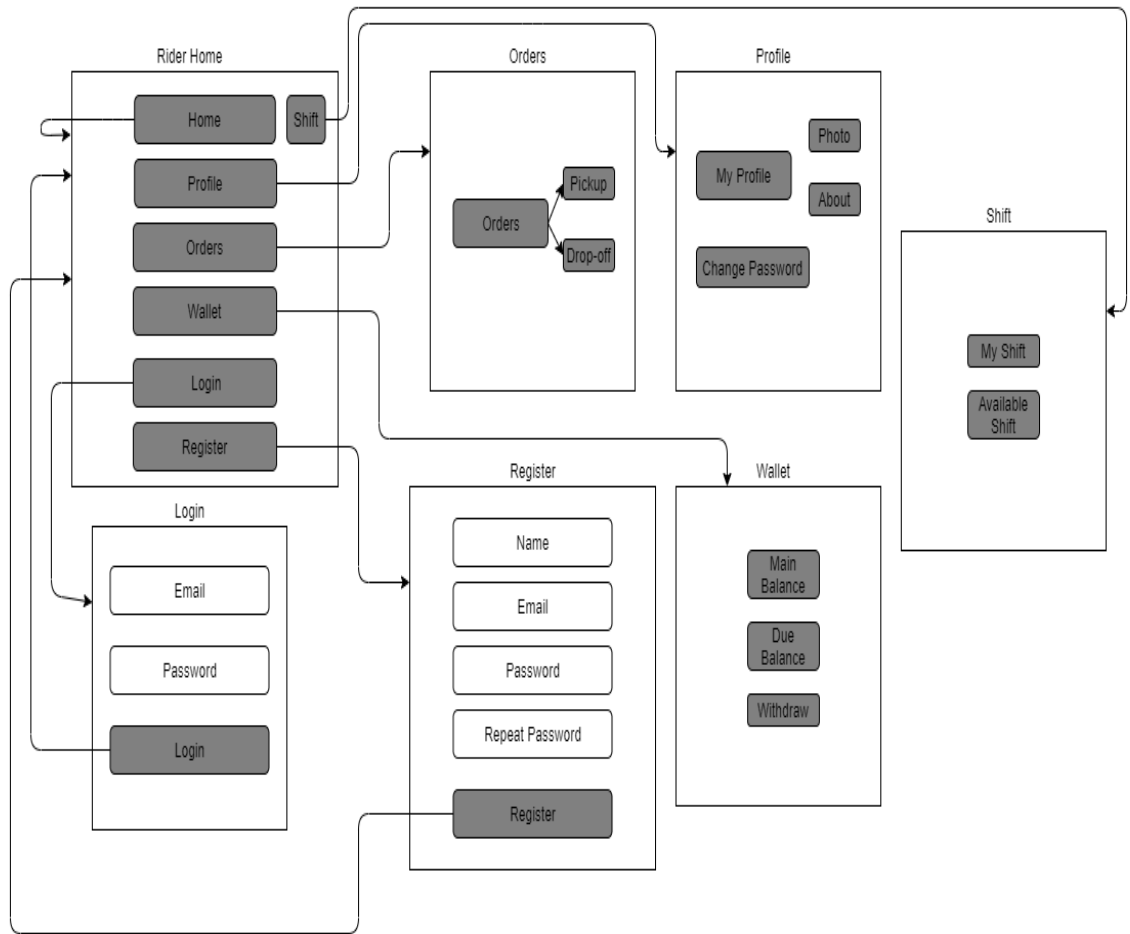


Fig: 1.4.5 Block Diagram for DokanBari Rider

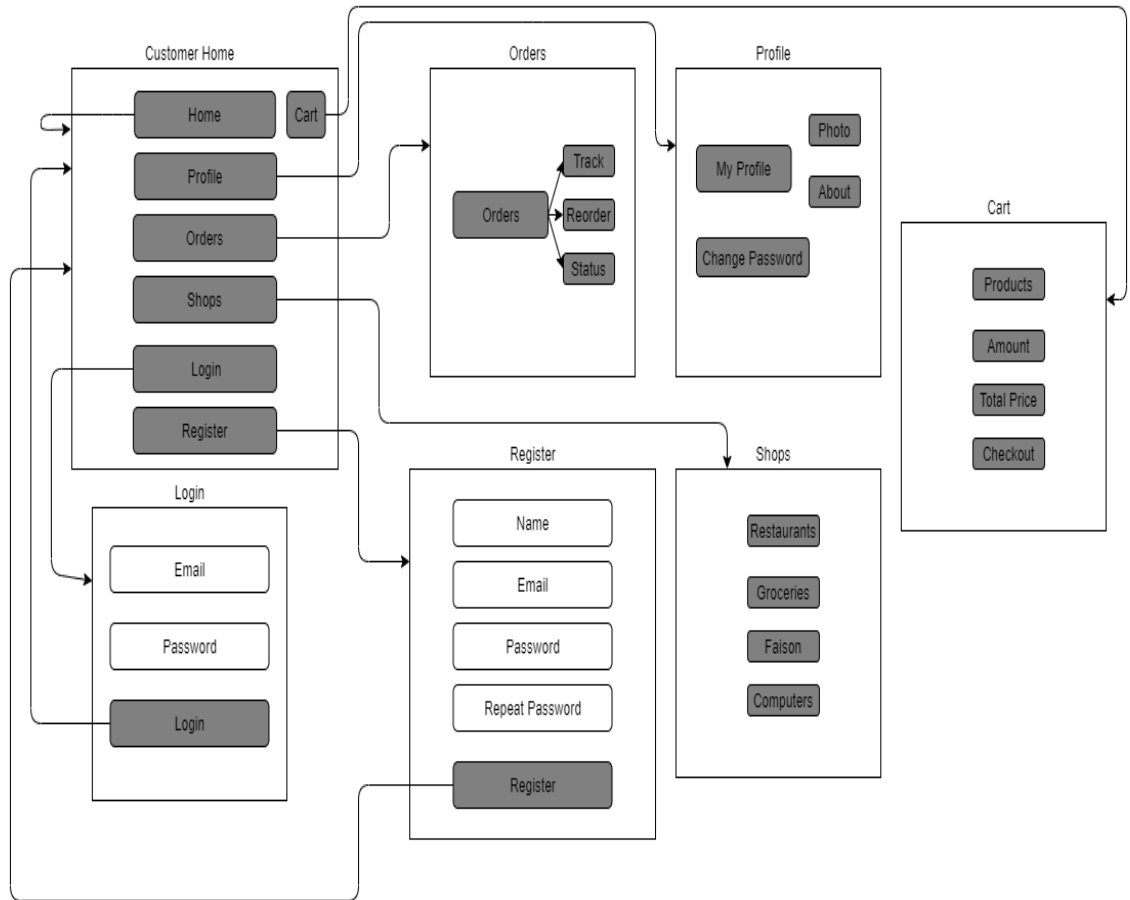


Fig: 1.4.6 Block Diagram for DokanBari Customer

1.5 Project Schedule

1.5.1 Gant Chart for Smart citizen

Task Name	Q1 2020			Q2 2021		
	10 /20	11/20	12/20	1/21	2/21	3/21
Planning	█					
Research		█				
Design			█			
Implementation			█			
Follow up					█	

Table: 1.5.1 Gant Chart

1.5.2 Gant Chart for DokanBari

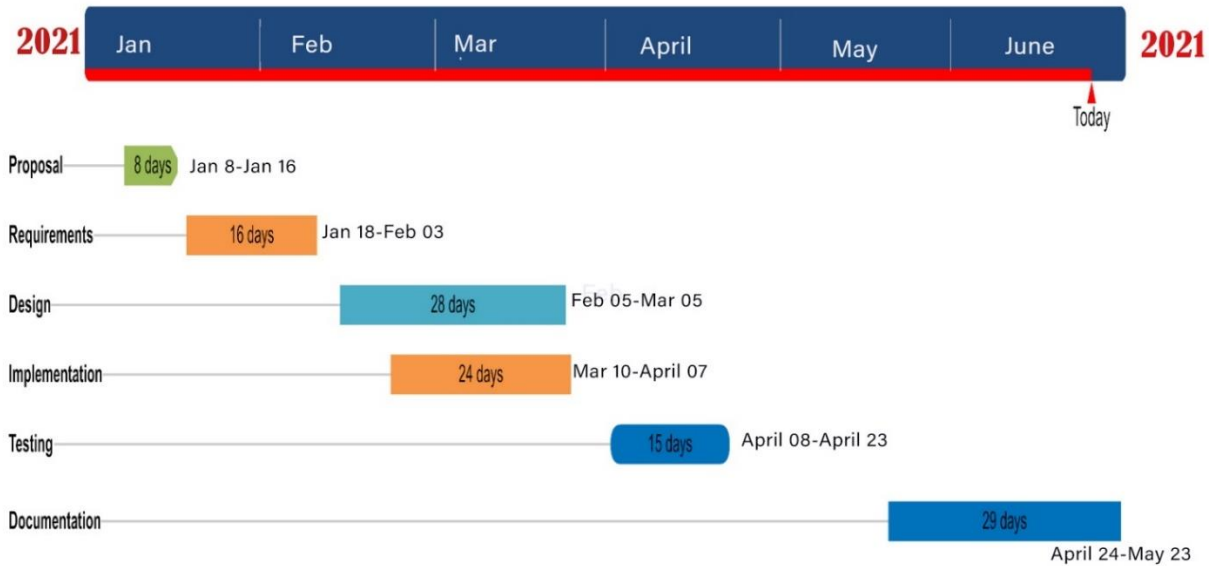


Table: 1.5.1 Gant Chart

Chapter 2: Software Requirement Specification

2.1 Functional Requirements include SmartCitizen

Requirements for Admin Fig: 1.4.1

AFR-1: Admin login
AFR-2: Profile
AFR-3: Manage shop
AFR-4: Manage products

Requirements for Vendor Fig: 1.4.2

VFR-1: Login
VFR-2: Register
VFR-3: Profile
VFR-4: Orders
VFR-5: Update shop
VFR-6: Manage products

Requirements for Rider Fig: 1.4.3

RFR-1: Login
RFR-2: Register
RFR-3: Profile
RFR-4: Orders
RFR-5: Wallet

Requirements for Customer Fig: 1.4.4

CFR-1: Login
CFR-2: Register
CFR-3: Profile
CFR-4: Add to cart
CFR-5: Checkout
CFR-6: Orders

2.2 Non-Functional Requirements

- User should be authenticated by the System for specific features.
- It should have password change option.
- This system Should have Session for purchase products from Sign in to Sign out.
- System should have session for accept or reject shop.
- System should have session for accept or reject product.

2.3 Data Requirements

- Sign in information must be email and password for all users.
- In Sign up and recovery page new password and confirm password must be. same.
- Forgotten email address must be pre-registered email.
- Forgotten account verify code must be matched with sent code to user email.

2.4 Performance Requirements

2.4.1 Speed and Latency Requirements

- Every request should not take more than 1 second.
- Shop management speed as fast as possible.

2.4.2 Precision or Accuracy Requirements

- Shop id must be in hidden mode.

2.4.3 Capacity Requirements

- Every vendor has single shop for serve the product.

2.5 Dependability Requirements

2.5.1 Reliability Requirements

- Shop category must be added by admin.
- Product category must be added by admin.

2.5.2 Availability Requirements

- Update shop must be available for every vendor.
- Pickup drop off should be available for rider.

2.3 Functional Requirement Mapping

F.R.ID	Finalized Functional Requirement
AFR-1	Admin login
AFR-2	Profile
AFR-3	Manage shop
AFR-4	Manage Products
RFR-1	Login
RFR-2	Register
RFR-3	Profile
RFR-4	Orders
RFR-5	Wallet
VFR-1	Login
VFR-2	Register
VFR-3	Profile
VFR-4	Orders
VFR-5	Update Shop
VFR-6	Mange Products
CFR-1	Login
CFR-2	Register
CFR-3	Profile
CFR-4	Add to Cart
CFR-5	Checkout
CFR-6	Order

Chapter 3 : System Analysis

3.1 Use Case Diagram for SmartCitizen

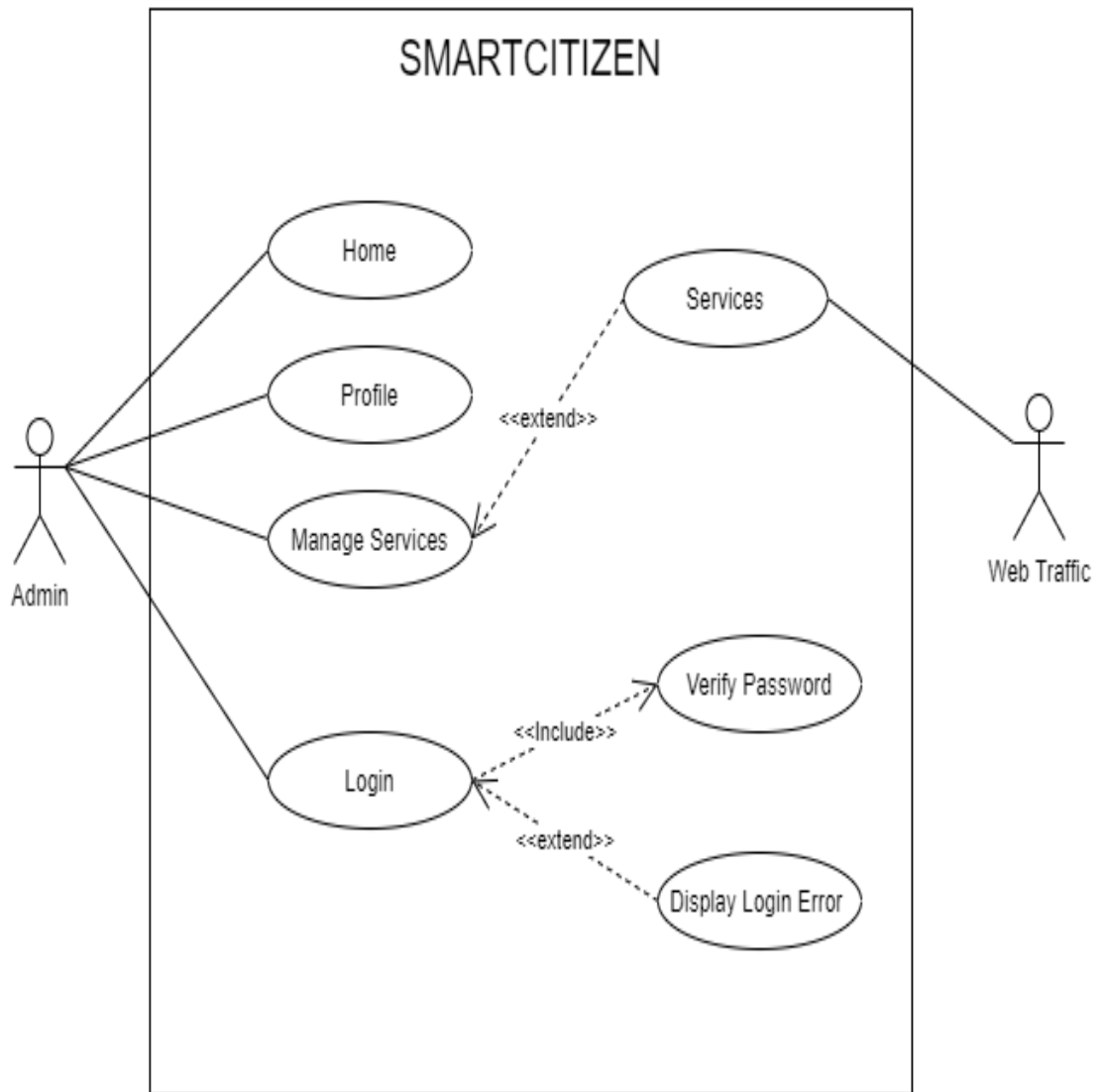


Fig: 3.1 DokanBari Use Case Diagram

3.2 Use Case Diagram for DokanBari

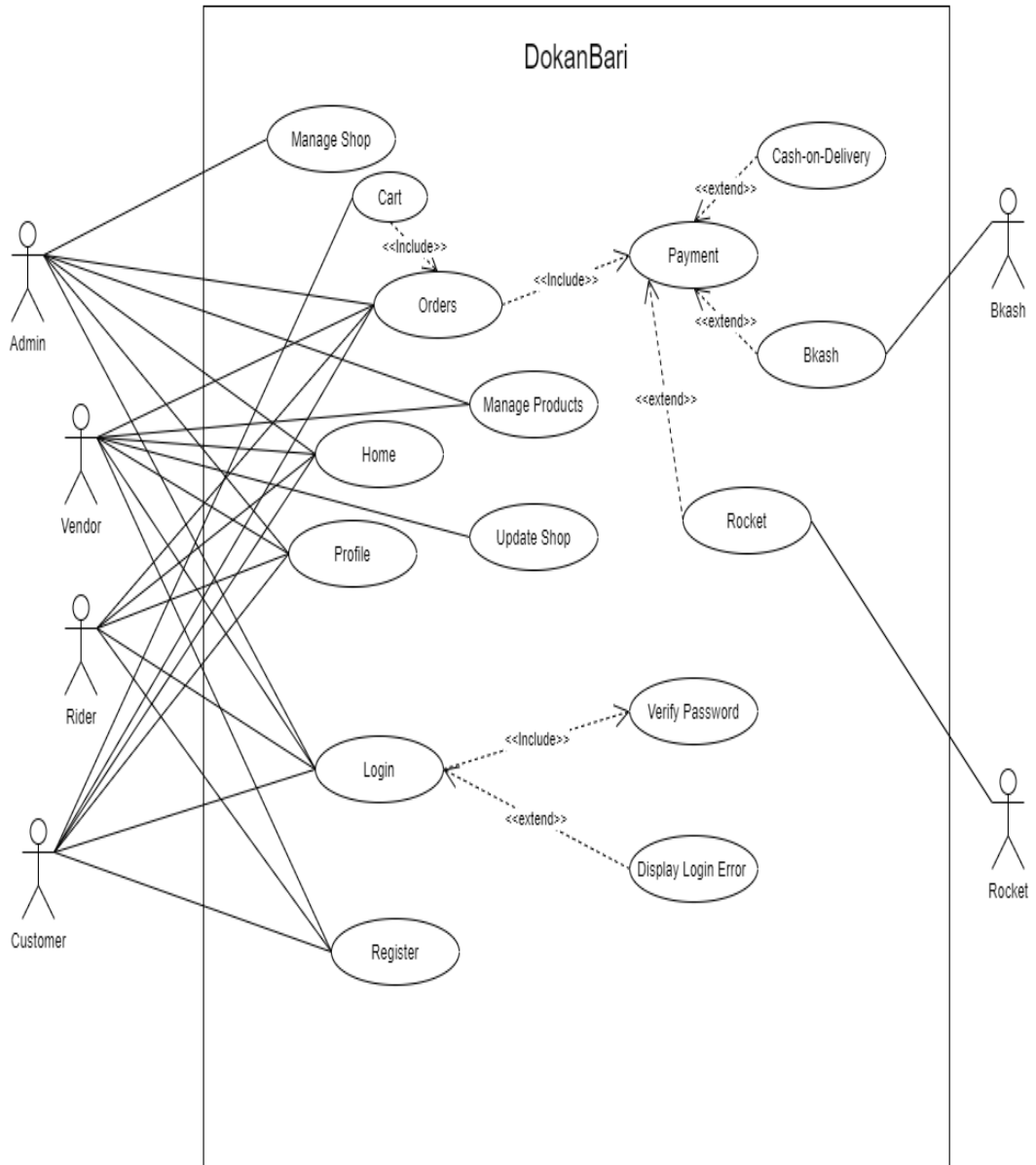


Fig: 3.2 DokanBari Use Case Diagram

3.3 Identifying Use Cases include SmartCitizen

UC-1:	Registration
UC-2:	Login
UC-3:	Home
UC-4:	Profile
UC-5:	Manage Shop
UC-6:	Manage Products
UC-7:	Orders
UC-8:	Cart
UC-9:	Payment
UC-10:	Display login error
UC-11:	Bkash
UC-12:	Rocket
UC-13:	Cash-on-delivery

3.4 Identifying Use Cases Description

Use Case Id	UC-1
Use Case name	Registration
Scenario	User can do Registration to create a new account
Triggering event	If any user want to create a new account and clicked Register button
Brief description	User can create an account, Where it should be provided some basic information about user : Name, Email, Password, and Retype-password
Actors	Vendor, Rider, Customer
Extending use case	None
Included use case	None
Stakeholders	Vendor, Rider, Customer
Pre-Condition	User must have a valid email address for Register the system
Basic path	<ol style="list-style-type: none"> 1. Click on “Register” link on Home page 2. Get Register page 3. Enter the name 4. Enter the email 5. Enter the Password 6. Enter the Retype-Password 7. Click on Register button
Alternative path	None
Exceptional path	None
Post-Condition	All the input values must be validated.

Table: 3.3.1 use case description login

Use Case Id	UC-2
Use Case name	Login
Scenario	User can do Login for enter the system
Triggering event	If any user want to Login for enter the system and clicked Login button
Brief description	User can Login, Where it should be provided some basic information about user : Email and Password
Actors	Admin, Vendor, Rider, Customer
Extending use case	UC-10: Display login error
Included use case	UC-9: Verify password
Stakeholders	Admin, Vendor, Rider, Customer
Pre-Condition	User must have a valid email and password for login the system
Basic path	<ol style="list-style-type: none"> 1. Click on "Sign in" link on Home page 2. Get Sign in page 3. Enter the email 4. Enter the Password 5. Click on Sign in button
Alternative path	None
Exceptional path	None
Post-Condition	All the input values must be match in database record

Table: 3.3.2 use case description home

Use Case Id	UC-3
Use Case name	Home
Scenario	User can land home page in to the system if they have account.
Triggering event	If any user want to go home page to the System and clicked on logo.
Brief description	User can click on logo to go home to the system. If user don't Log in into the system. It will ask for Email and Password, click on Log in button.
Actors	Admin, Vendor, Rider, Customer
Extending use case	None
Included use case	UC-1: Registration, UC-2: Login
Stakeholders	Admin, Vendor, Rider, Customer
Pre-Condition	User must have an account and remember the password for Log in to the System and help land the home page.
Basic path	Click on "Logo" Image
Alternative path	Click on "Home" Menu
Exceptional path	None
Post-Condition	User must Logout for exit the system.

Table: 3.3.3 use case description profile

Use Case Id	UC-4
Use Case name	Profile
Scenario	User can do Edit profile.
Triggering event	If any user want to Edit profile and clicked Edit profile button
Brief description	User can Edit profile, Where it should be provided some information about user : Name, Age, gender, Phone
Actors	Admin, Vendor, Rider, Customer
Extending use case	None
Included use case	None
Stakeholders	Admin, Vendor, Rider, Customer
Pre-Condition	User must enter new information.
Basic path	<ol style="list-style-type: none"> 1. Click on “Edit Profile” link on Profile page 2. Get Edit Profile page 3. Enter new information 4. Click on Update button
Alternative path	None
Exceptional path	None
Post-Condition	All the input values must be validating

Table: 3.3.4

use case description mange shop

Use Case Id	UC-5
Use Case name	Manage Shop
Scenario	User can Mange Shop if they have authorized account.
Triggering event	If any user want manage shop and clicked Shops option.
Brief description	User must have account on system, For mange the shop.
Actors	Admin, Vendor
Extending use case	None
Included use case	None
Stakeholders	Admin, Vendor
Pre-Condition	User must have logged in into the system.
Basic path	<ol style="list-style-type: none"> 1. Click on “Shops” option and manage shop 2. Update shop info 3. Add product under the shop
Alternative path	None
Exceptional path	None
Post-Condition	None

Table: 3.3.5 use case description manage product

Use Case Id	UC-6
Use Case name	Manage Products
Scenario	User can Mange product if they have authorized shop
Triggering event	If any user click on “Products” option.
Brief description	User must have Approved shop on the system, For mange the product.
Actors	Admin, Vendor
Extending use case	None
Included use case	None
Stakeholders	Admin, Vendor
Pre-Condition	User must have Approved shop.
Basic path	<ol style="list-style-type: none"> 1. Click on “products” option and manage product 2. Add new 3. Update 4. delete
Alternative path	None
Exceptional path	None
Post-Condition	None

Table: 3.3.6 use case description orders

Use Case Id	UC-7
Use Case name	Orders
Scenario	Customer can order any type of product, vendor can get orders from customer, rider get delivery request by the shop
Triggering event	Checkout, update status, drop off
Brief description	Customer can add product to cart, also order check out, vendor get instant order request by customer, rider get orders by the shop.
Actors	Admin, Vendor, Rider, Customer
Extending use case	None
Included use case	None
Stakeholders	Admin, Vendor, Rider, Customer
Pre-Condition	User must use cart for make new order
Basic path	<ol style="list-style-type: none"> 1 Click on “Add cart” button add product 2 Click on “Checkout” button confirm order 3 Update status
Alternative path	None
Exceptional path	None
Post-Condition	Customer must select payment option for new order

Table: 3.3.7 use case description cart

Use Case Id	UC-8
Use Case name	Cart
Scenario	User must use cart for make new order
Triggering event	4 Click on “Add cart” button add product
Brief description	Customer can add product to cart, order check out, also clear cart item.
Actors	Customer
Extending use case	None
Included use case	Uc-7 Orders
Stakeholders	Customer
Pre-Condition	Must logged in into the system
Basic path	5 Click on “Add to Cart” button add product
Alternative path	none
Exceptional path	none
Post-Condition	6 Click on “delete” icon delete product

Table: 3.3.8 use case description payment

Use Case Id	UC-9
Use Case name	Payment
Scenario	User must use payment for make new order
Triggering event	7 Click on “Select Payment” option pay subtotal
Brief description	Select cash-on-delivery, Bkash, Rocket, for pay summation of price.
Actors	Customer
Extending use case	Uc-11, Uc-12, Uc-13
Included use case	None
Stakeholders	Customer
Pre-Condition	User must select payment option
Basic path	Click on “Select Payment” option pay order subtotal
Alternative path	Cash-on-delivery
Exceptional path	None
Post-Condition	None

3.5 Activity Diagram

UC-1: Registration

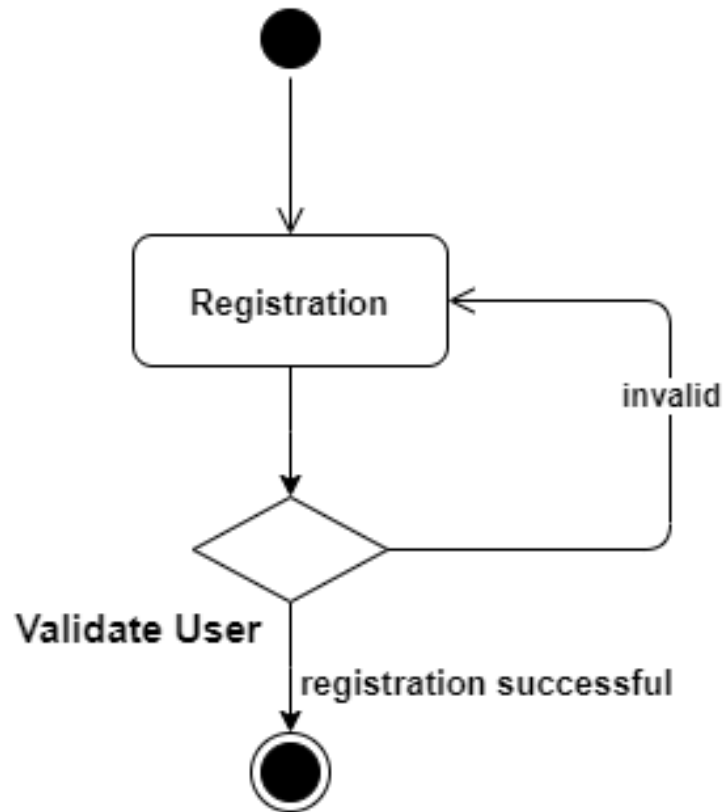


Fig: 3.4.1 Registration

UC-2: Login

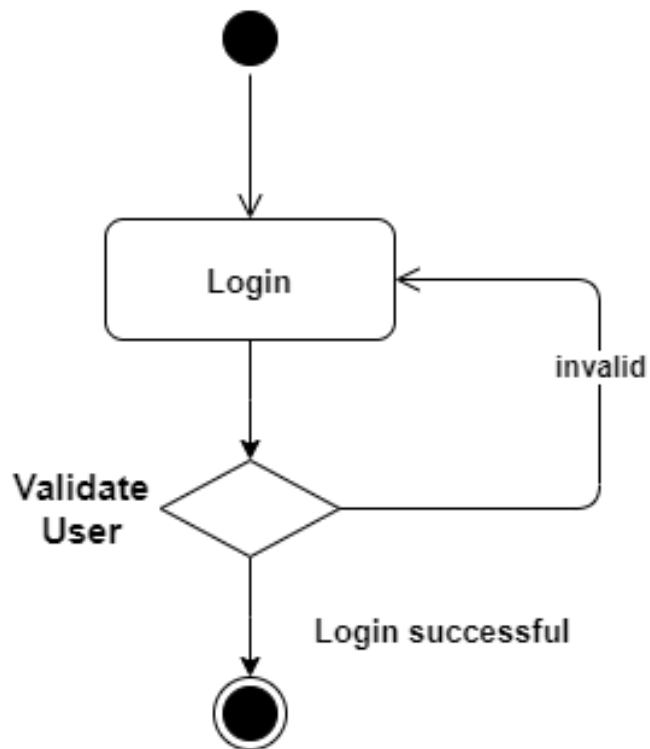


Fig: 3.4.2 Login

UC-3: Home

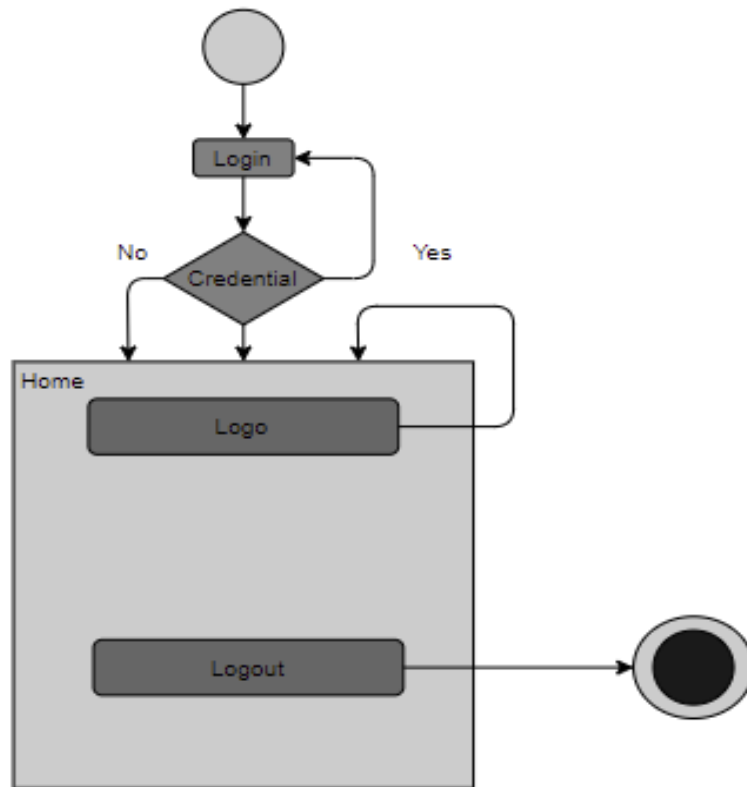


Fig: 3.4.3 Home

UC-4: Profile

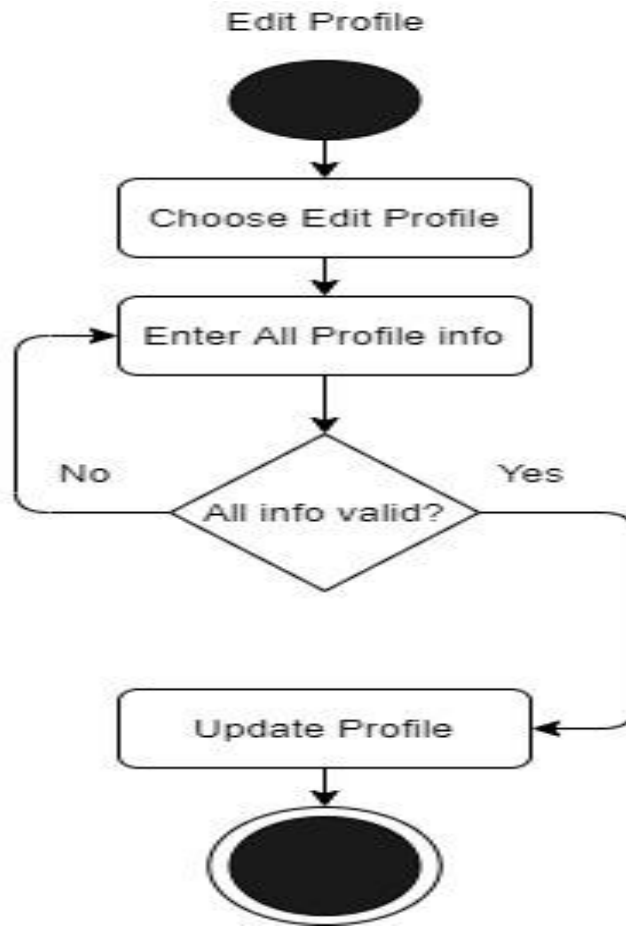


Fig: 3.4.4 Profile

UC-5: Mange shop

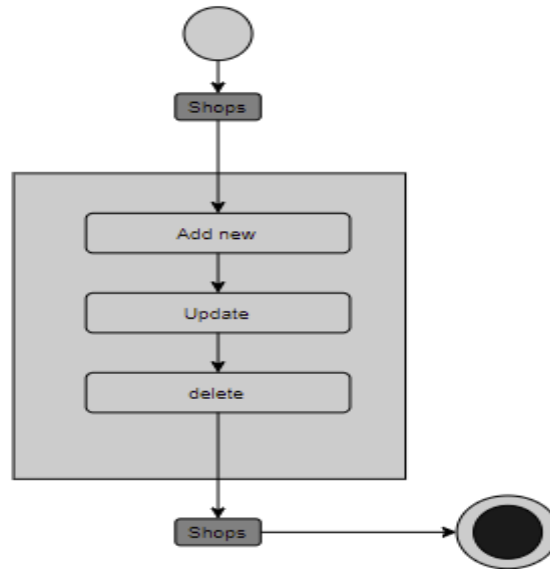


Fig: 3.4.5 Mange shop

UC-6: Mange Products

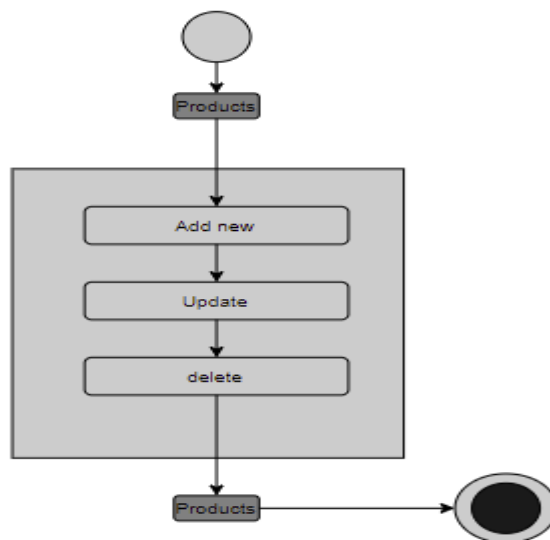


Fig: 3.4.6 Mange Products

UC-7: Orders

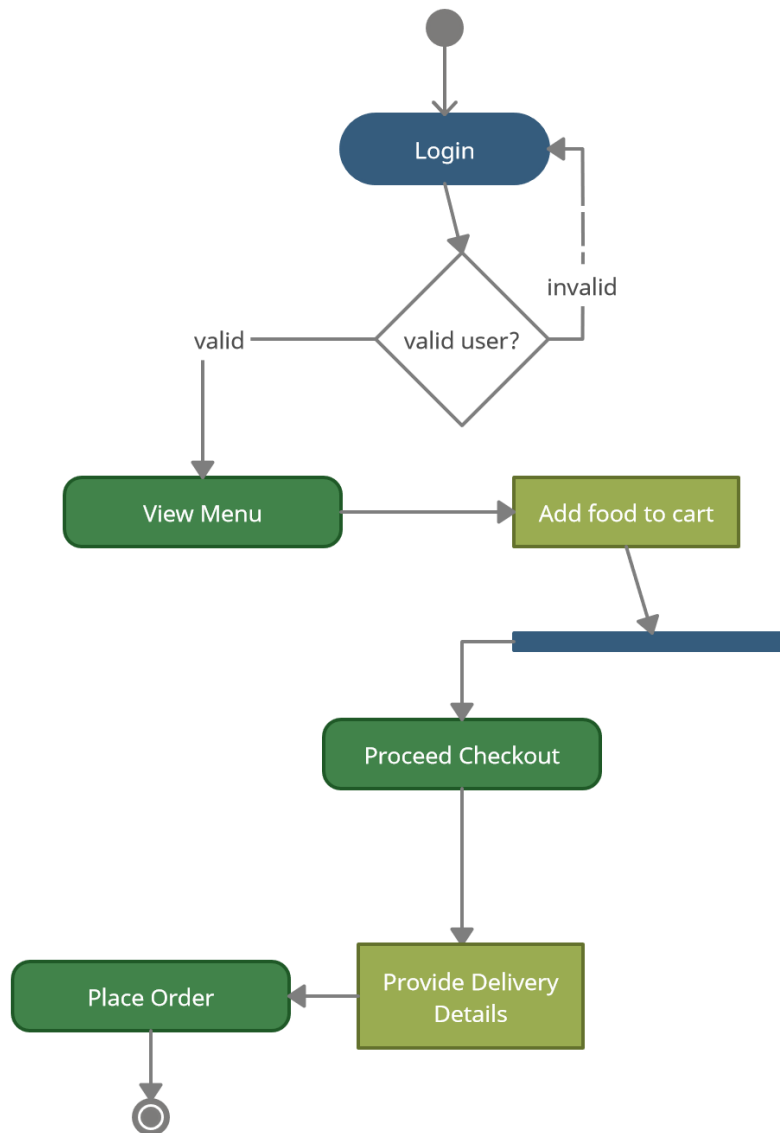


Fig: 3.4.7 Orders

UC-8: Cart

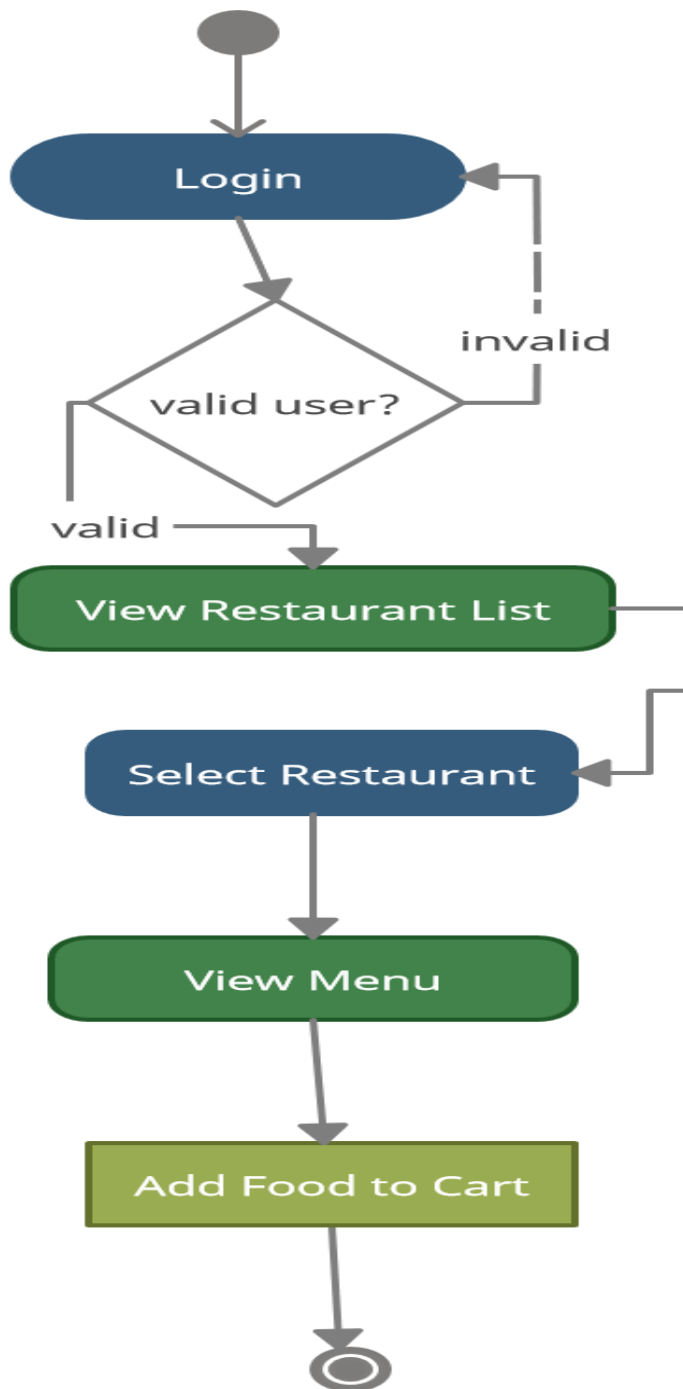


Fig: 3.4.8 Cart

UC-9: Payment

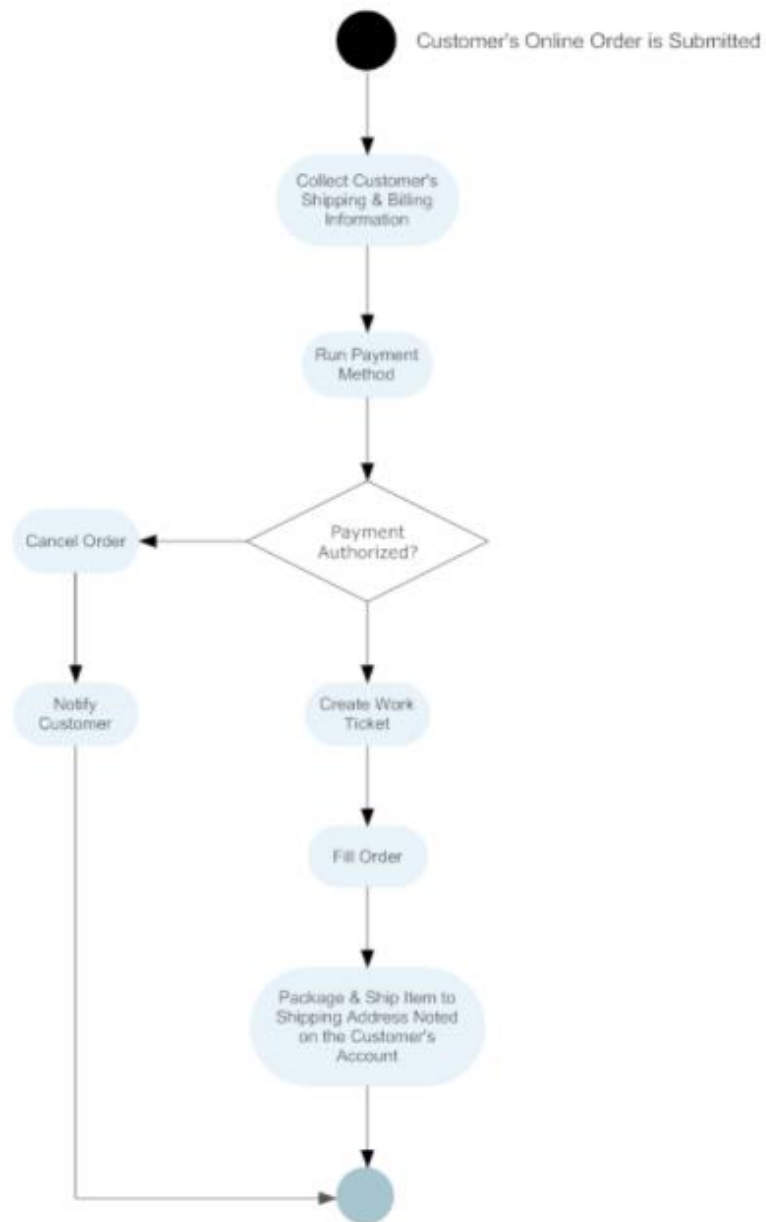


Fig: 3.4.9 Payment

3.6 System Sequence Diagram UC-1: Registration

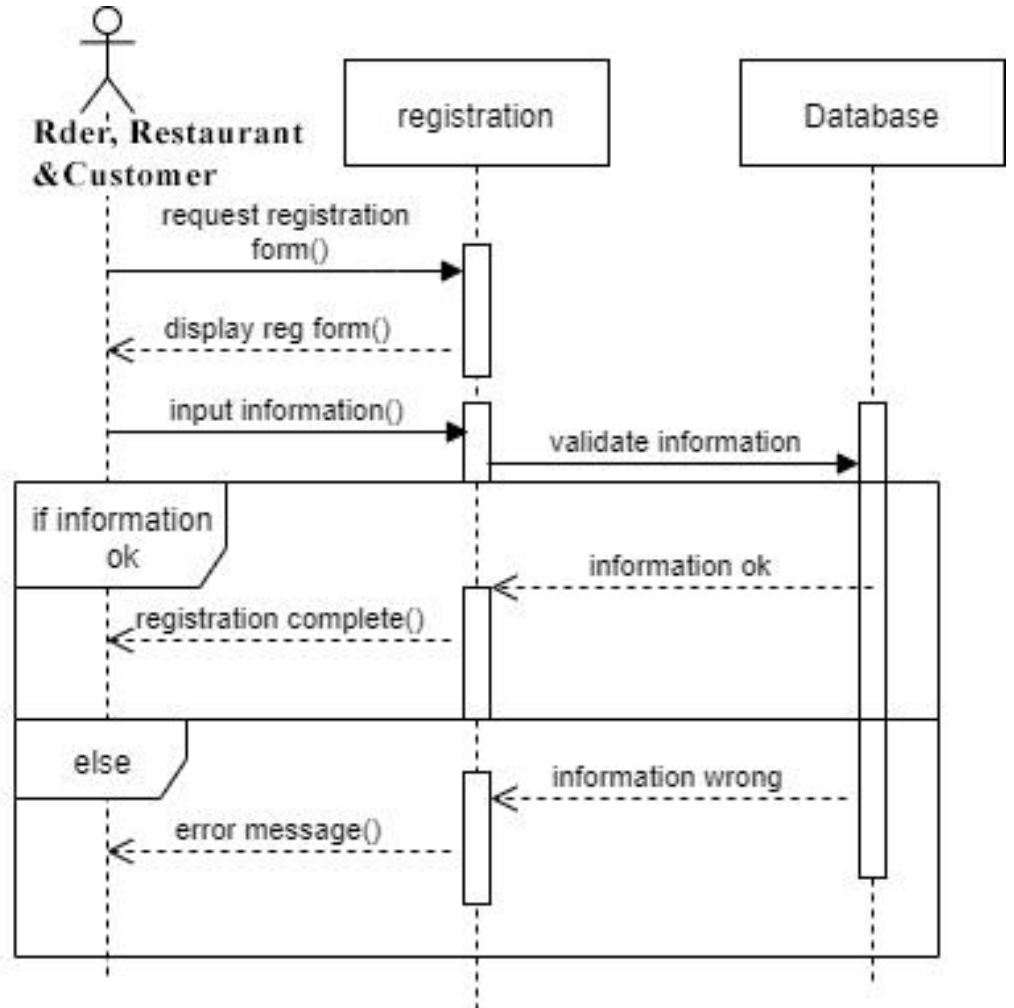


Fig: 3.9.2 Registration

UC-2: Login

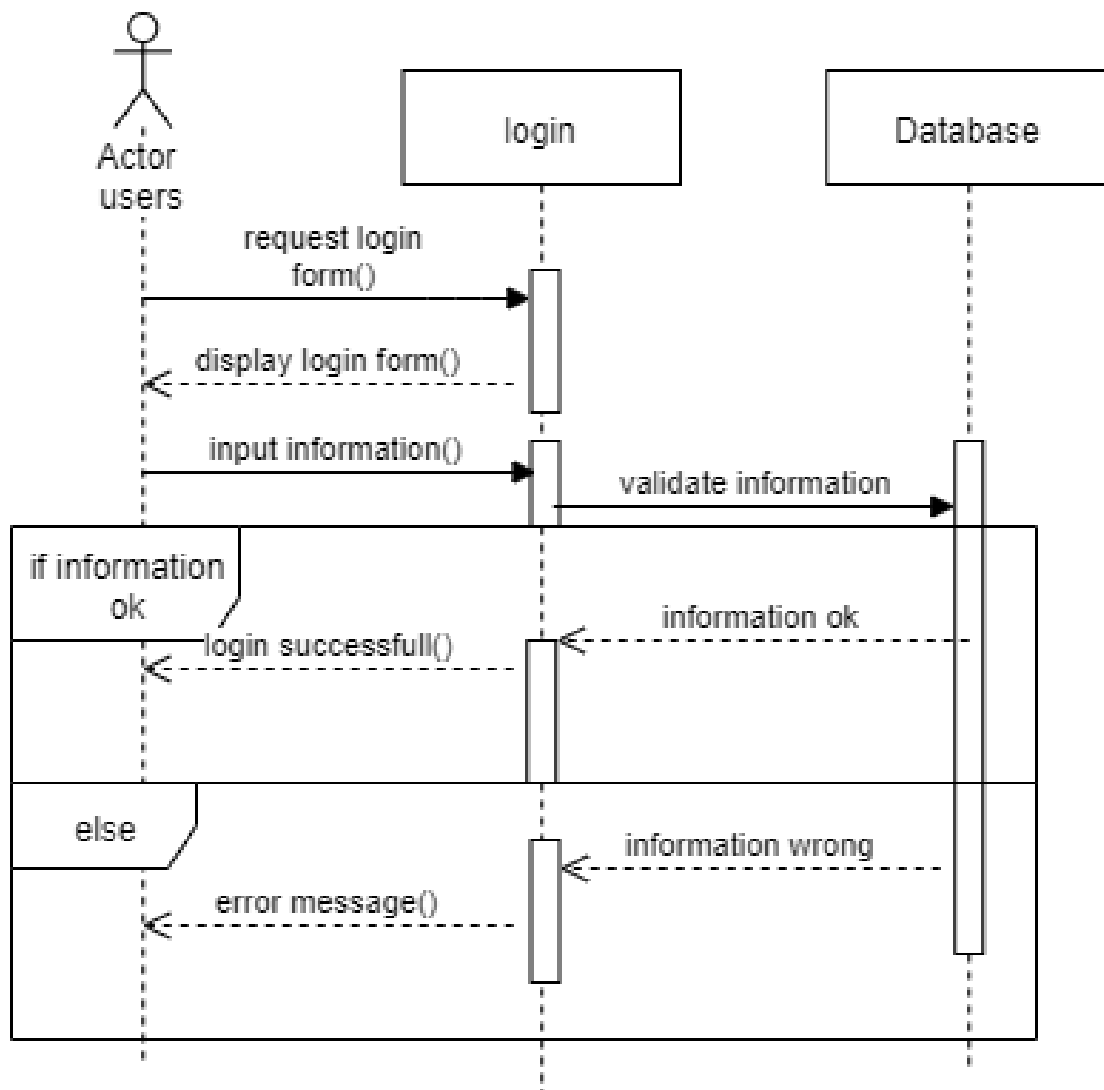


Fig: 3.9.2 Login

UC-7: Orders Customer

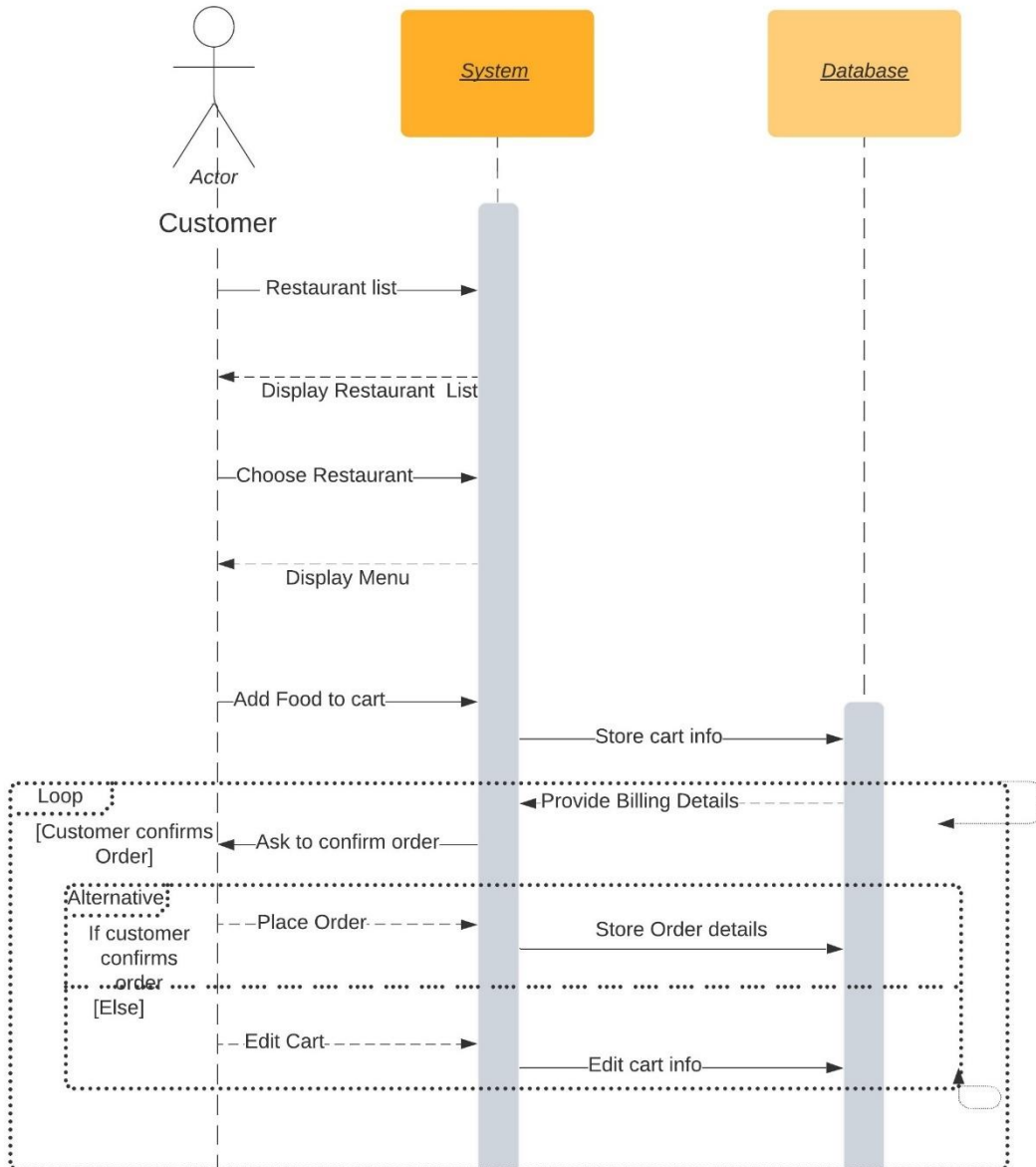


Fig: 3.9.2 Orders Customer

UC-7: Shop's response

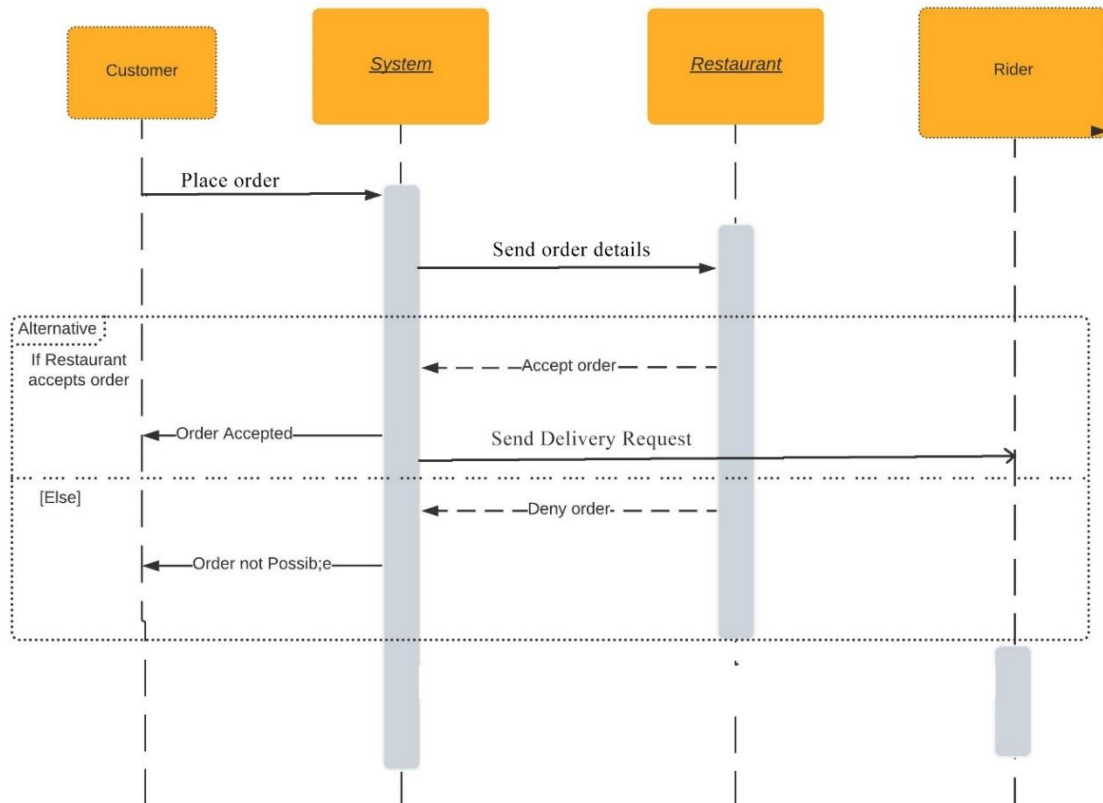


Fig: 3.9.2 Shop's response

UC-7: Delivery Confirmation

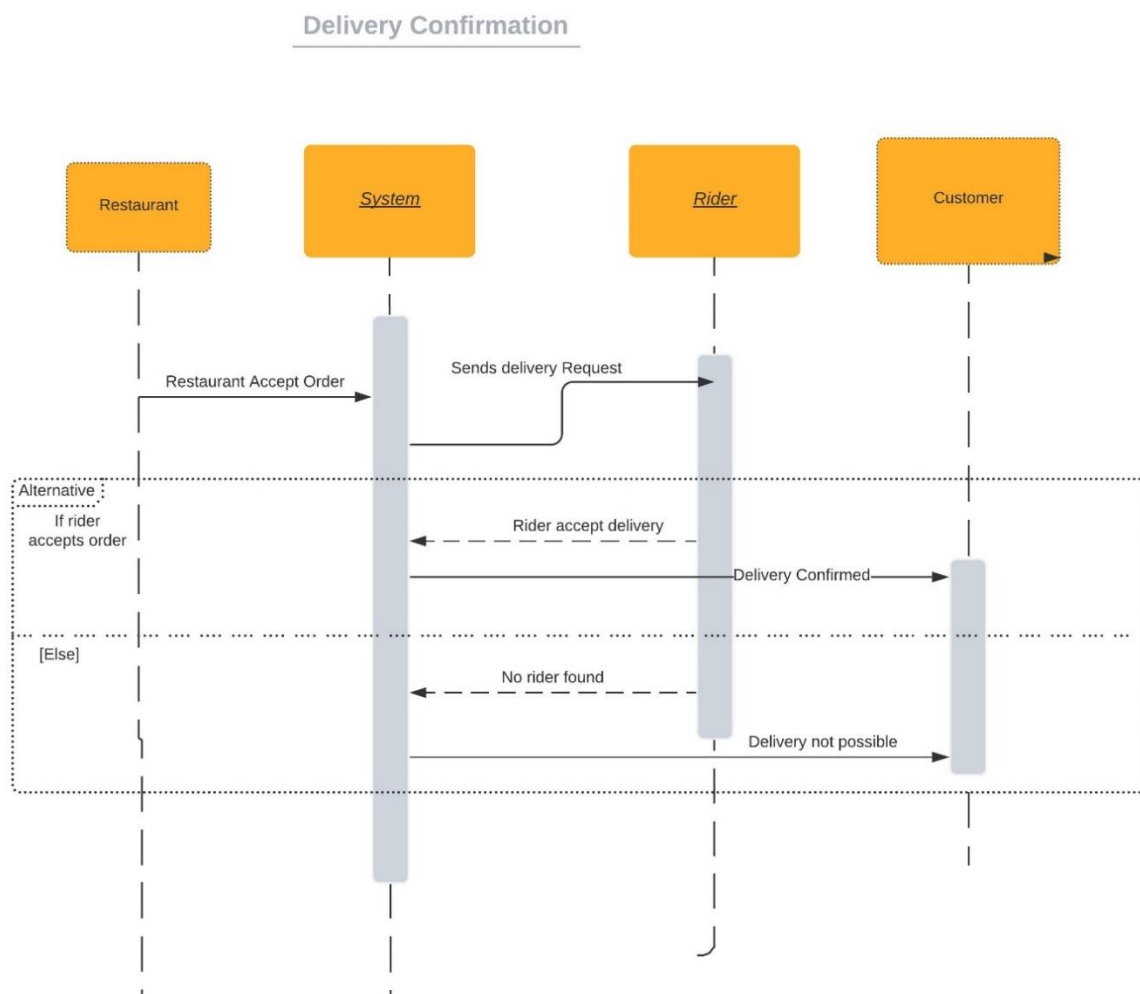


Fig: 3.9.2 Delivery Confirmation

Chapter 4: SYSTEM DESIGN SPECIFICATION

4.1 Entity Relationship Diagram

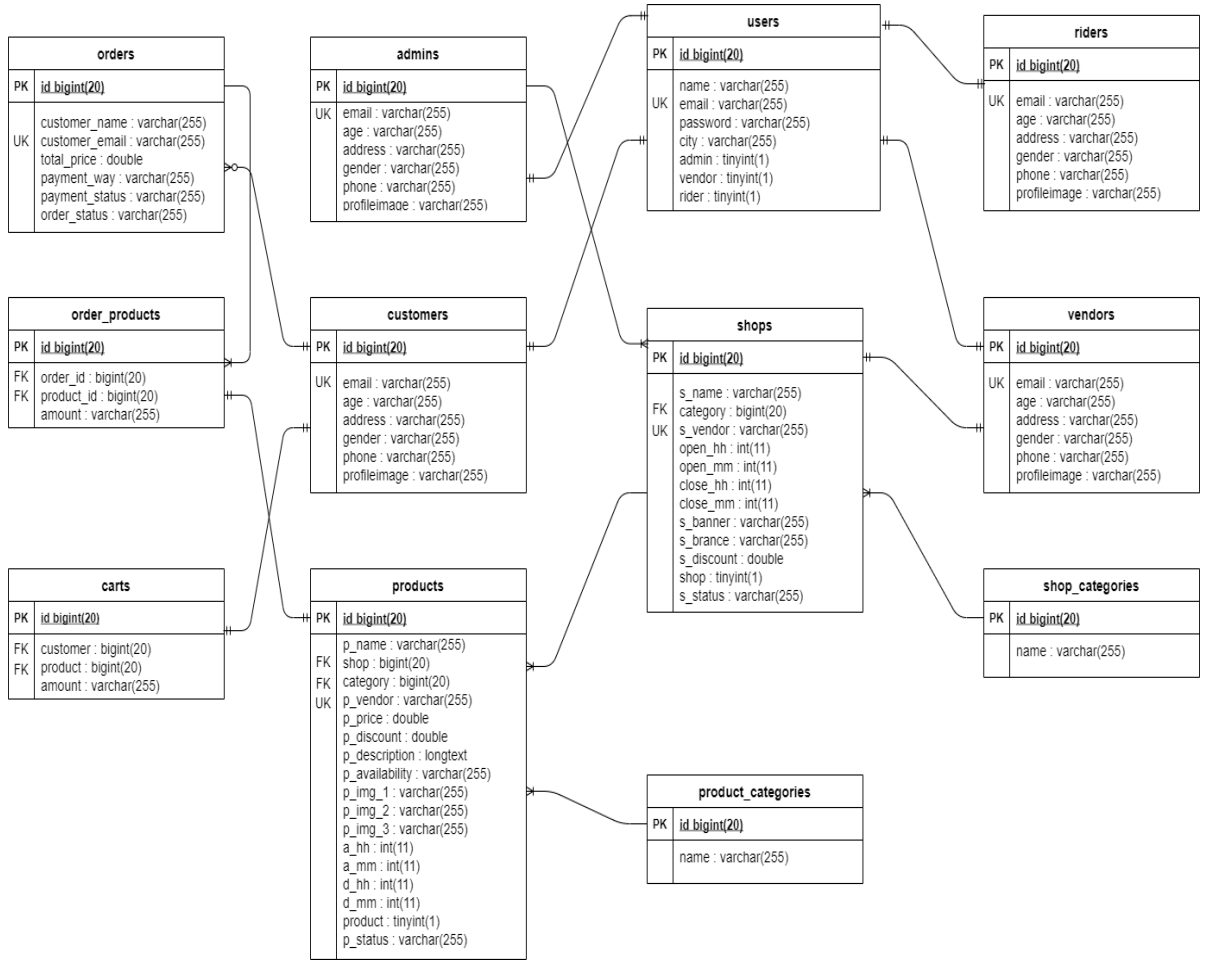


Fig: 4.1 ER Diagram

4.2 Class Diagram

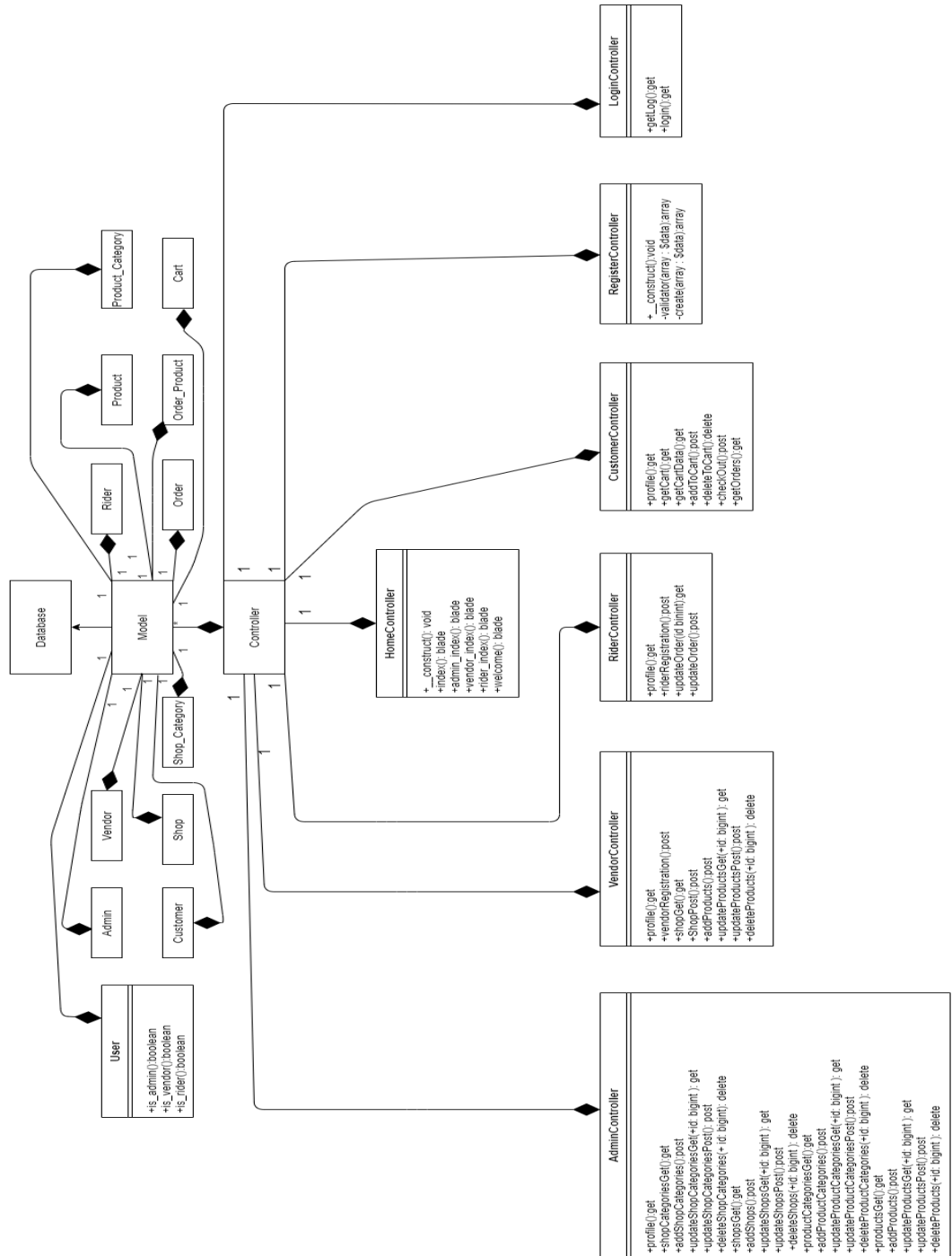


Fig 4.2 Class Diagram

4.3 Development of tools And Technology

4.3.1 User Interface Technology

1. HTML5
2. CSS3
3. Bootstrap 4
4. Framework - Laravel
5. Database - MySQL
6. Server - Apache

4.3.2. Implementation Tools & Platforms

1. Object-Oriented PHP
2. Xampp
3. Visual Code
4. Chrome Browser

Chapter 5: System Testing

5.1 Testing Features

5.1.1 Feature to be tested

1. Registration
2. Login
3. Add to cart
4. Place order
5. Approve Vendor
6. Approve rider
7. Confirm Delivery
8. Logout

5.1.2 Feature not to be tested

1. View Menu
2. Confirm Food Delivered

5.2 Testing Strategies

5.2.1 Test Approach

1. The system will manually be tested.
2. The system testing is based on user acceptancy.

5.2.2 Pass/Fail Criteria

1. Component Pass/Fail criteria – The test will pass if the case meet the object design requirement or fail if not.
2. Integration Pass/Fail criteria – The test will pass if the case meet the object design architecture requirement or fail if not.
3. System Pass/Fail criteria – The test will pass if the case meet the functional and non-Functional requirements or fail if not.

5.2.3 Suspension and Redemption

1. Build Acceptance Test – The system will pass the test if every build is successful if not than try build again.
2. Regression Testing – The system should work properly after each change on the system.
3. System Design Changes – The system should work properly after each change in the design.

5.2.4 Testing Schedule

Test Phase	Time	Owner
Test Plan Creation	1 week	Akash Sheikh
Test Specification Creation	1 week	Akash Sheikh
Test Specification Team Review	2 weeks	Akash Sheikh
Component Testing	2 weeks	Akash Sheikh
Integration Testing	2 weeks	Akash Sheikh
System Testing	3 weeks	Akash Sheikh

Table 14: Testing Schedule

5.3 Test Cases

5.3.1 Test Case 01: Registration

Test Case: 5.3.1		Test Case Name: Registration			
System: Online Marketplace		Subsystem: N/A			
Designed by: Akash Sheikh		Design Date:			
Executed by: Akash Sheikh		Execution Date:			
Description: A user must be registered to order, sell & deliver foods online. To register an account a user must provide valid email address, password and other information.					
Pre-condition: Email address must be unique.					
Step	Email	Password	Response	Pass/Fail	Comment
1	skyfar@gmail.com	1234	Registration successful	Pass	
2	admin@yahoo.com	1234	Invalid email	Fail	
Post-condition: A new user account has been registered successfully					

Table 15: Test Case-01

5.3.2 Test Case 02: Login

Test Case: 5.3.2			Test Case Name: Login		
System: Online Marketplace			Subsystem: N//A		
Designed by: Akash Sheikh			Design Date:		
Executed by: Akash Sheikh			Execution Date:		
Description: Registered user must login to access system. The system will check for authentication and authorization.					
Pre-condition:					
<ul style="list-style-type: none"> • User must input valid email and password. • Let, email be skyfar@gmail.com and password: 1234 					
Step	Email	Password	Response	Pass/Fail	Comment
1	skyfar@gmail.com	1234	Successfully login into the system	Pass	
2	ruyhanur@yahoo.com	1234	Wrong email or password	Fail	
3	sadimuhammad@gmail.com	235677	Wrong email or password	Fail	
4	walidkhan@gmail.com		Password is required	Fail	
5		1234	Email is required	Fail	
Post-condition: Users will successfully log into the system.					

Table 16: Test Case-02

5.3.3 Test Case 03: Add to cart

Test Case: 5.3.3		Test Case Name: Add to cart		
System: Online Marketplace		Subsystem: N//A		
Designed by: Akash Sheikh		Design Date:		
Executed by: Akash Sheikh		Execution Date:		
Description: Registered customers can add food to their cart to order.				
Pre-condition: The customer must be logged in				
Step	Action	Response	Pass/Fail	Comment
1	Customer select the food he/she wants to order by pressing “Add to cart” .	System creates a new cart for the customer and add those selected foods.	Pass	
2	Customer doesn't select any food to cart.	System can't create new cart to add food.	Fail	
Post-condition: A new cart will be created				

Table 17: Test Case-03

5.3.4 Test Case 04: Place Order

Test Case: 5.3.4		Test Case Name: Place Order		
System: Online Marketplace		Subsystem: N//A		
Designed by: Akash Sheikh		Design Date:		
Executed by: Akash Sheikh		Execution Date:		
Description: Customer place order of the foods from the added cart.				
Pre-condition: Cart must have at least one food item .				
Step	Action	Response	Pass/Fail	Comment
1	Customer adds foods to cart and press “Place order”.	System places order for delivery confirmation.	Pass	
2	Customer doesn’t add any food in cart .	The place order button doesn’t response.	Fail	
Post-condition: Order is placed for delivery confirmation.				

Table 18: Test Case-04

5.3.5 Test Case 05: Approve Vendor

Test Case: 5.3.5		Test Case Name: Approve Vendor		
System: Online Marketplace		Subsystem: N//A		
Designed by: Akash Sheikh		Design Date:		
Executed by: Akash Sheikh		Execution Date:		
Description: Admin can Approve new request vendor in the system.				
Pre-condition: Admin must be logged in.				
Step	Action	Response	Pass/Fail	Comment
1	Admin approves new vendor request by clicking 'Approve' button.	System will approve the new Vendor in the system.	Pass	
2	Admin deny new vendor request by clicking 'Deny' button.	Request not approved by administration.	Fail	
Post-condition: Vendor information will get updated				

Table 19: Test Case-05

5.3.6 Test Case 06: Approve rider

Test Case: 5.3.6		Test Case Name: Approve rider		
System: Online Marketplace		Subsystem: N//A		
Designed by: Akash Sheikh		Design Date:		
Executed by: Akash Sheikh		Execution Date:		
Description: Admin can Approve new request rider in the system.				
Pre-condition: Admin must be logged in.				
Step	Action	Response	Pass/Fail	Comment
1	Admin approves new rider request by clicking 'Approve' button.	System will approve the new rider in the system.	Pass	
2	Admin deny new rider request by clicking 'Deny' button.	Request not approved by administration.	Fail	
Post-condition: Rider information will get updated				

Table 20: Test Case-06

5.3.7 Test Case 07: Confirm Delivery

Test Case: 5.3.7		Test Case Name: Confirm Delivery		
System: Online Marketplace		Subsystem: N//A		
Designed by: Akash Sheikh		Design Date:		
Executed by: Akash Sheikh		Execution Date:		
Description: The customer gets delivery confirmation.				
Pre-condition: Rider and Vendor(Restaurant) must accept the order.				
Step	Action	Response	Pass/Fail	Comment
1	Vendor(Restaurant) and a rider press 'Accept order' button.	Food delivery confirmed.	Pass	
2	Vendor(Restaurant) press 'Deny Order' button.	Delivery gets canceled.	Fail	
Post-condition: Food will be delivered to the customer.				

Table 21: Test Case-07

5.3.8 Test Case 08: Logout

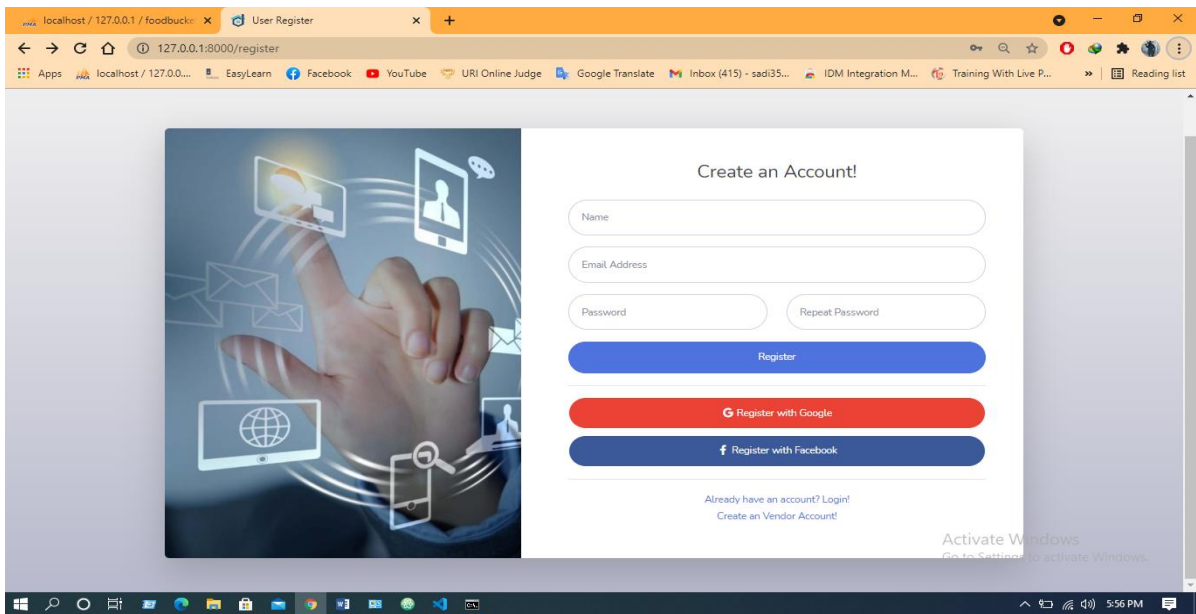
Test Case: 5.3.8		Test Case Name: Logout		
System: Online Marketplace		Subsystem: N//A		
Designed by: Akash Sheikh		Design Date:		
Executed by: Akash Sheikh		Execution Date:		
Description: Registered users can log out from their access and end session				
Pre-condition: The user must be logged in				
Step	Action	Response	Pass/Fail	Comment
1	Users click on “Logout” button	System will end the session and bring the user to the “Login” page	Pass	
Post-condition: User logs out from their access.				

Table 22: Test Case-08

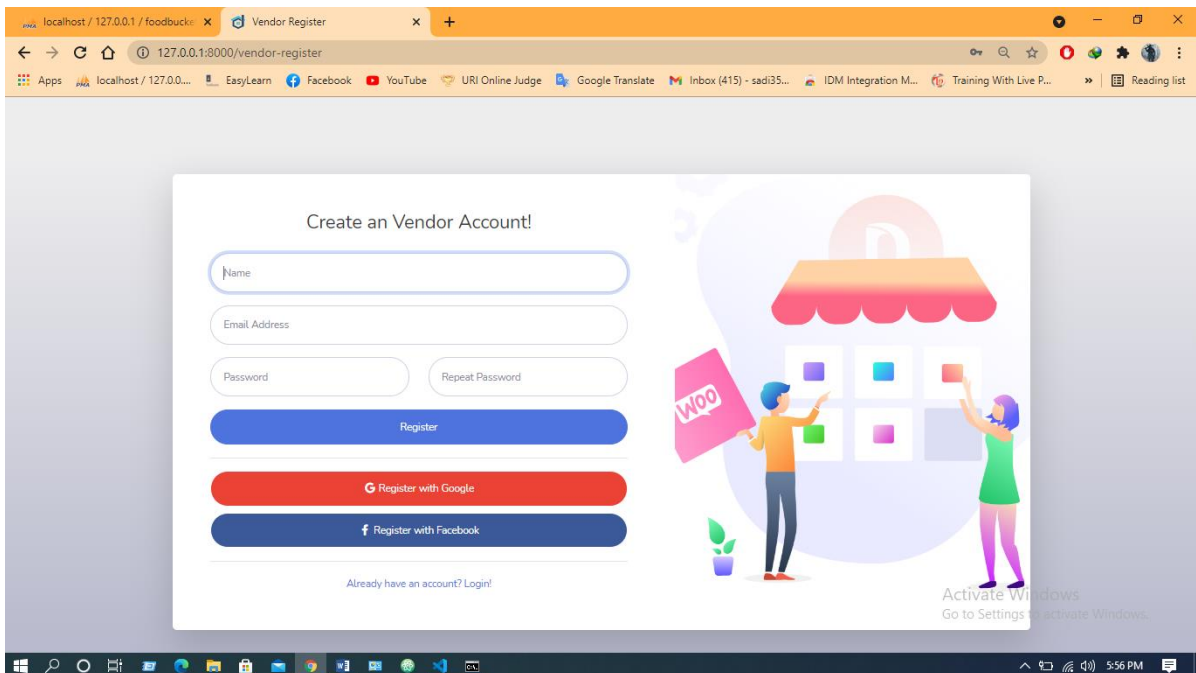
CHAPTER: 06

USER MANUAL

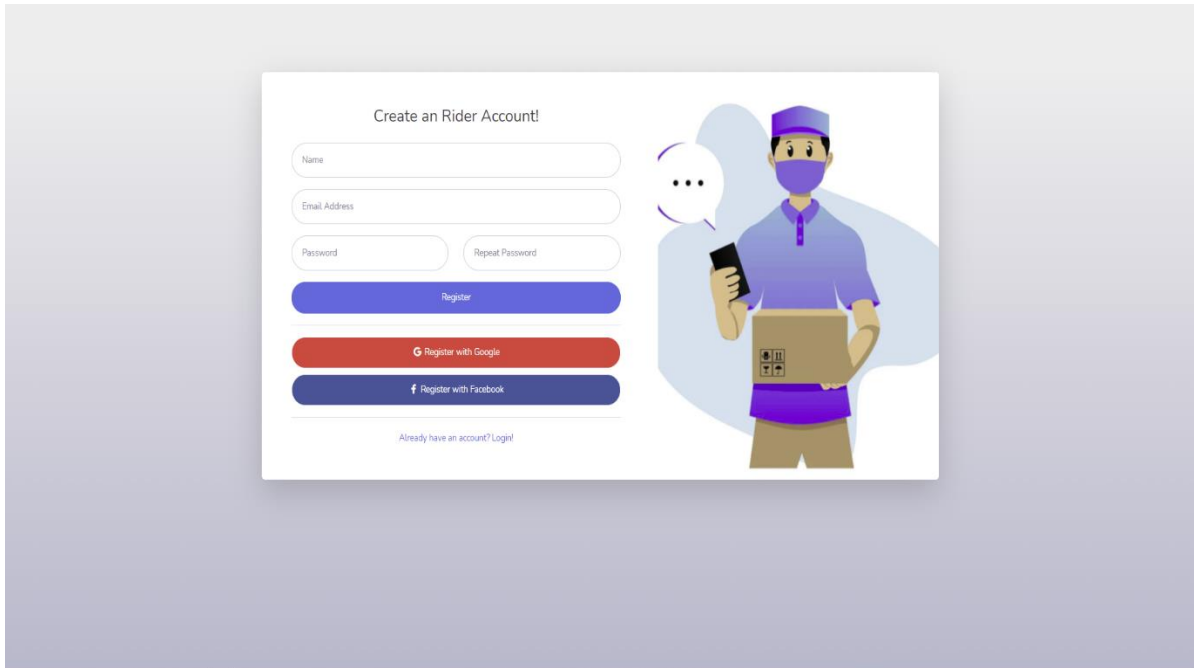
6.1 Registration (Customer)



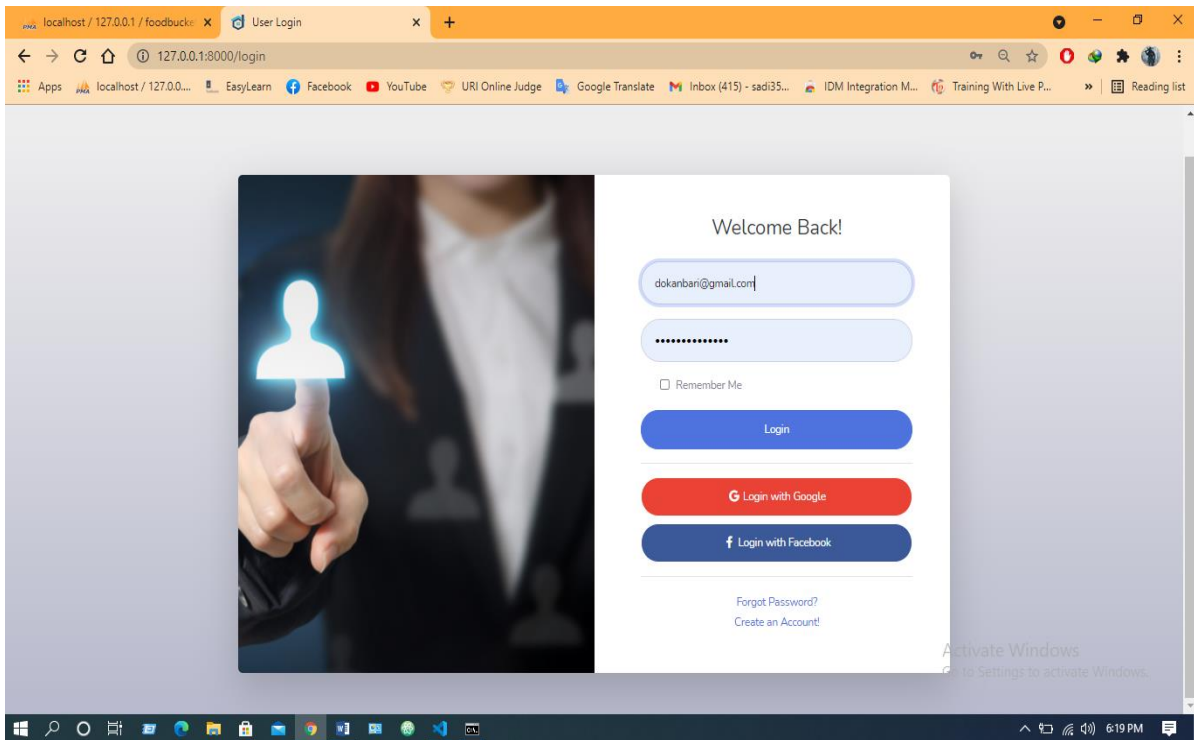
6.2 Registration (Vendor)



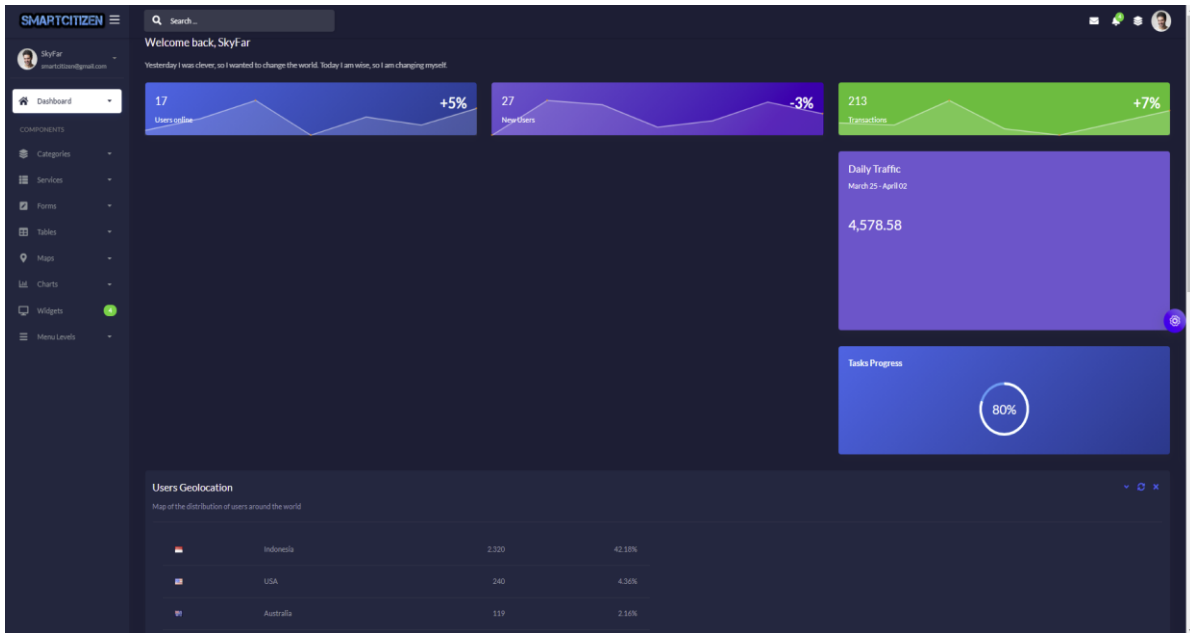
6.3 Registration (Rider)



6.4 Login (All user)



6.5 Dashboard (SmartCitizen Admin)



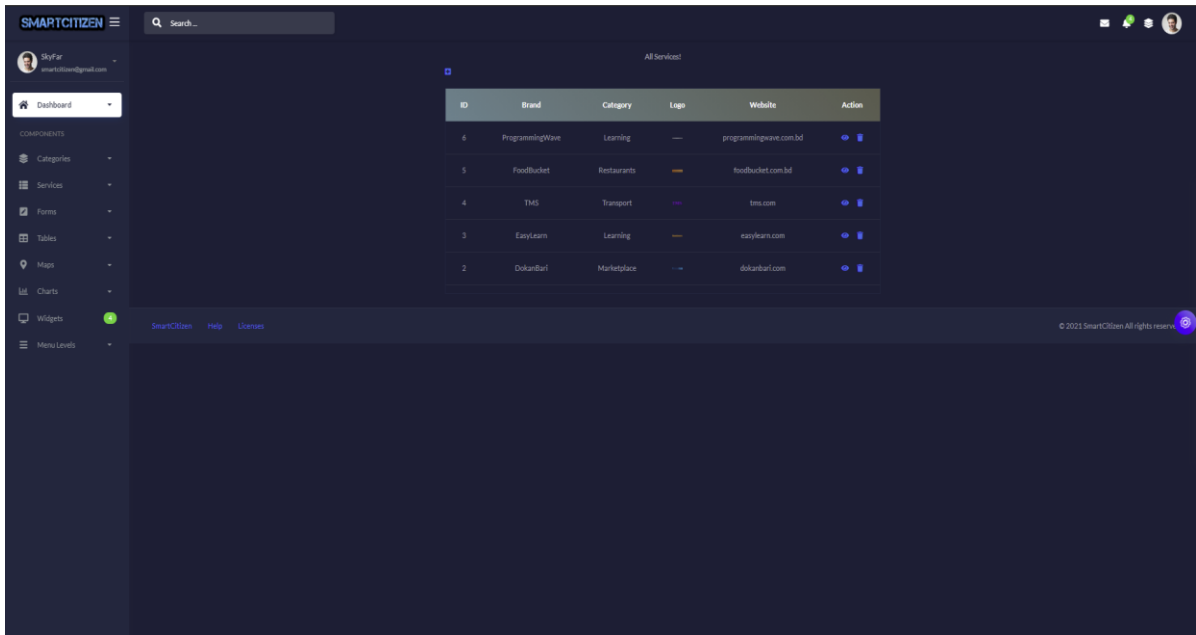
6.6 Service Category (SmartCitizen Admin)

The page displays a table of service categories under the heading "All Categories".

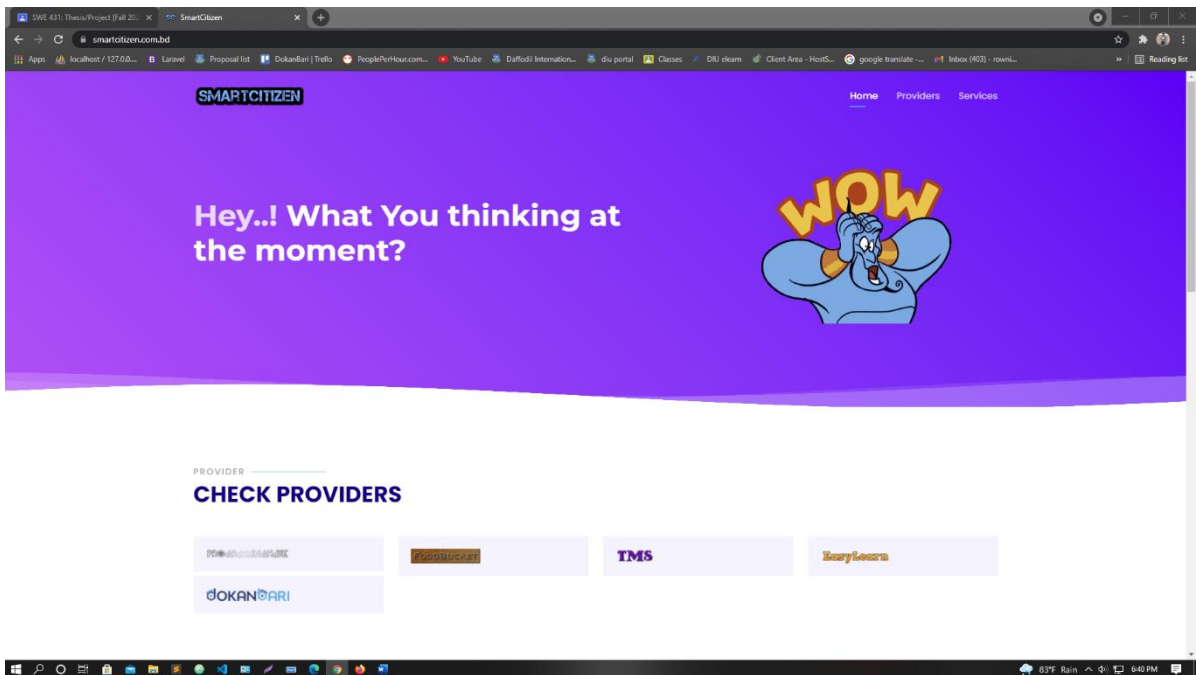
ID	Name	GF	Action
9	Restaurants		
8	Transport		
7	Learning		
6	Marketplace		

Footer: SmartCitizen Help License © 2021 SmartCitizen All rights reserved.

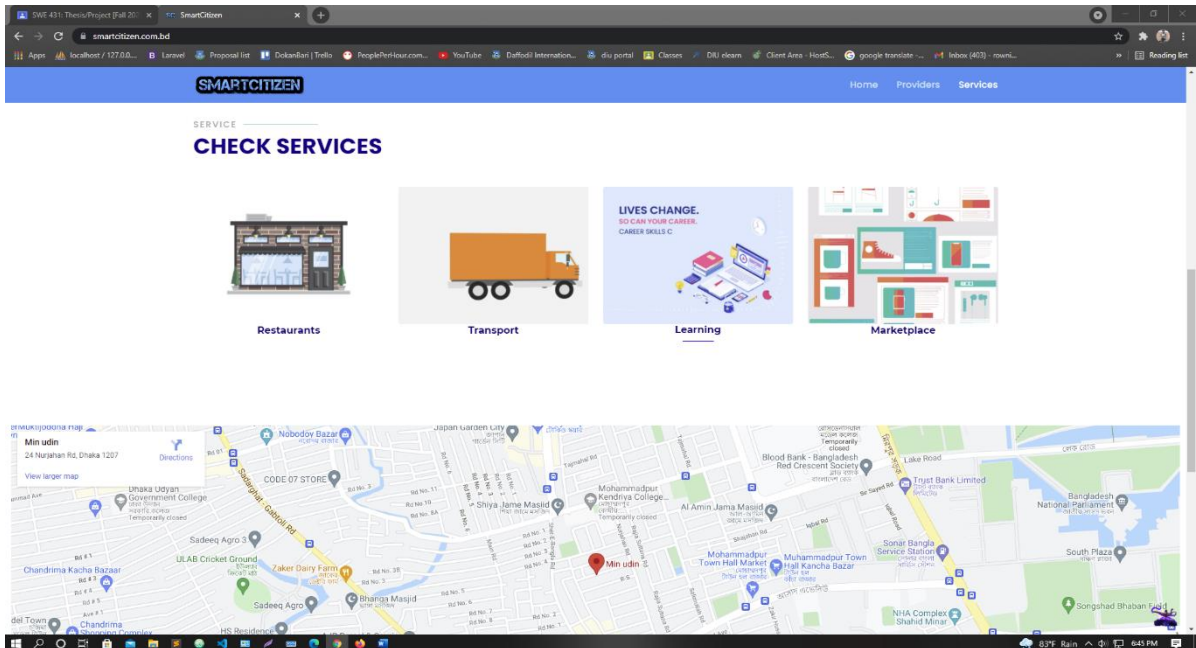
6.7 Services (SmartCitizen Admin)



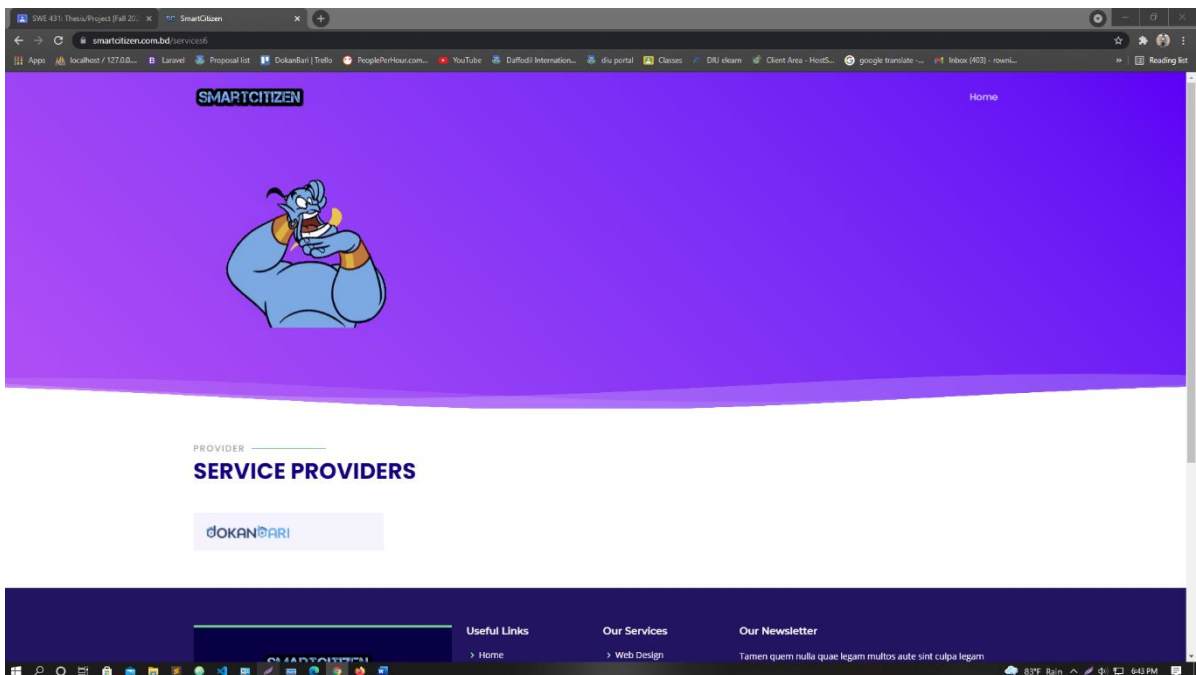
6.8 Gateways (SmartCitizen Admin)



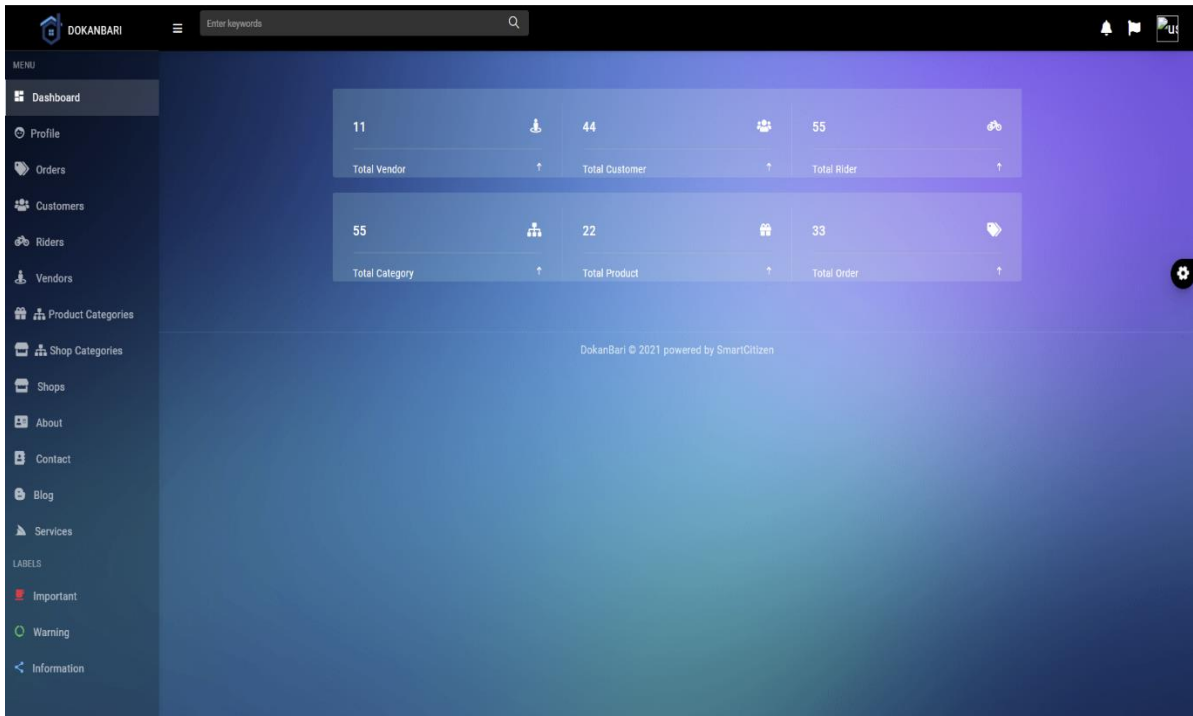
6.9 Get Services



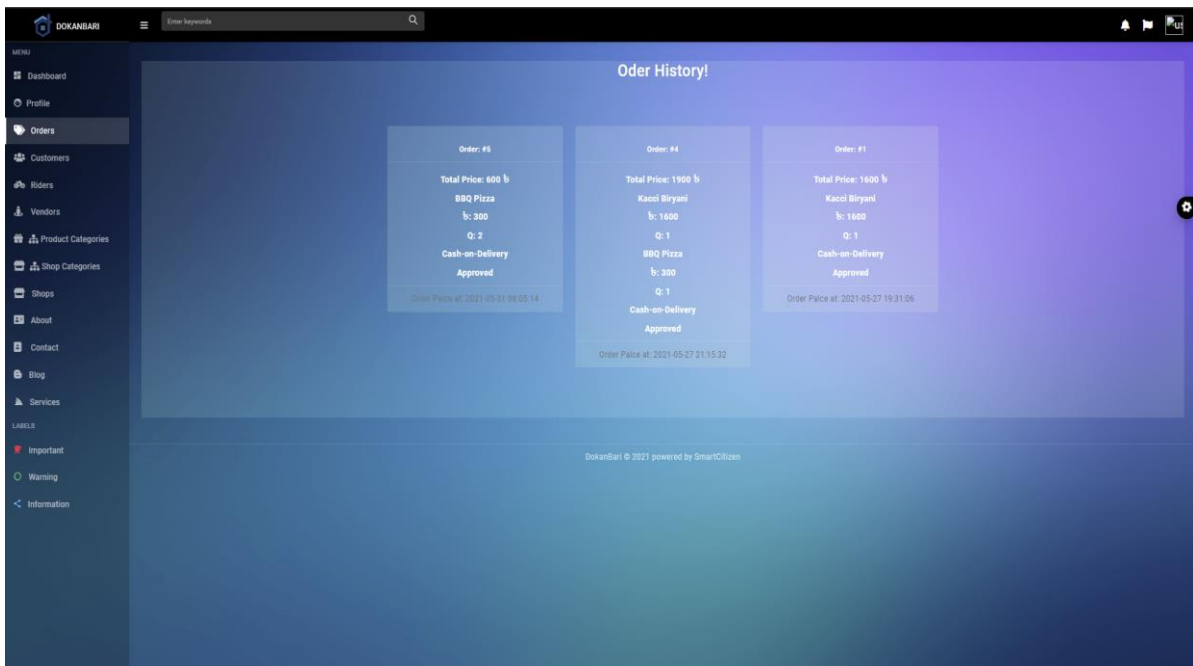
6.10 Service Provider



6.11 Dashboard (DokanBari Admin)



6.12 Orders (DokanBari Admin)



6.13 Shops (DokanBari Admin)

The screenshot shows the 'Shops by DokanBari' section of the admin interface. It features a table with the following data:

ID	NAME	CATEGORY	OPEN	CLOSE	BANNER	BRANCH	DISCOUNT	SHOP	STATUS	ACTION
4	Star Tech	Computer	10-0	23-59		Pangsha	2%	Active	Accepted	
3	Almas Super Shop	Groceries	9-30	23-59		Dhaka	3%	Active	Accepted	
2	The Food Shop	Restaurants	9-0	23-59		Pangsha	5%	Active	Accepted	
1	Hungry Jacks	Restaurants	10-30	23-59		Pangsha	10%	Active	Accepted	

Below this table, there is a section for 'Shops by Vendor!' which contains one entry:

ID	NAME	CATEGORY	OPEN	CLOSE	BANNER	BRANCH	DISCOUNT	SHOP	STATUS	ACTION
5	Agora Super Shop	Groceries	9-20	23-59		Dhaka	0%	Active	Accepted	

The footer of the interface reads 'DokanBari © 2021 powered by SmartClixon'.

6.14 Product by shop (DokanBariAdmin)

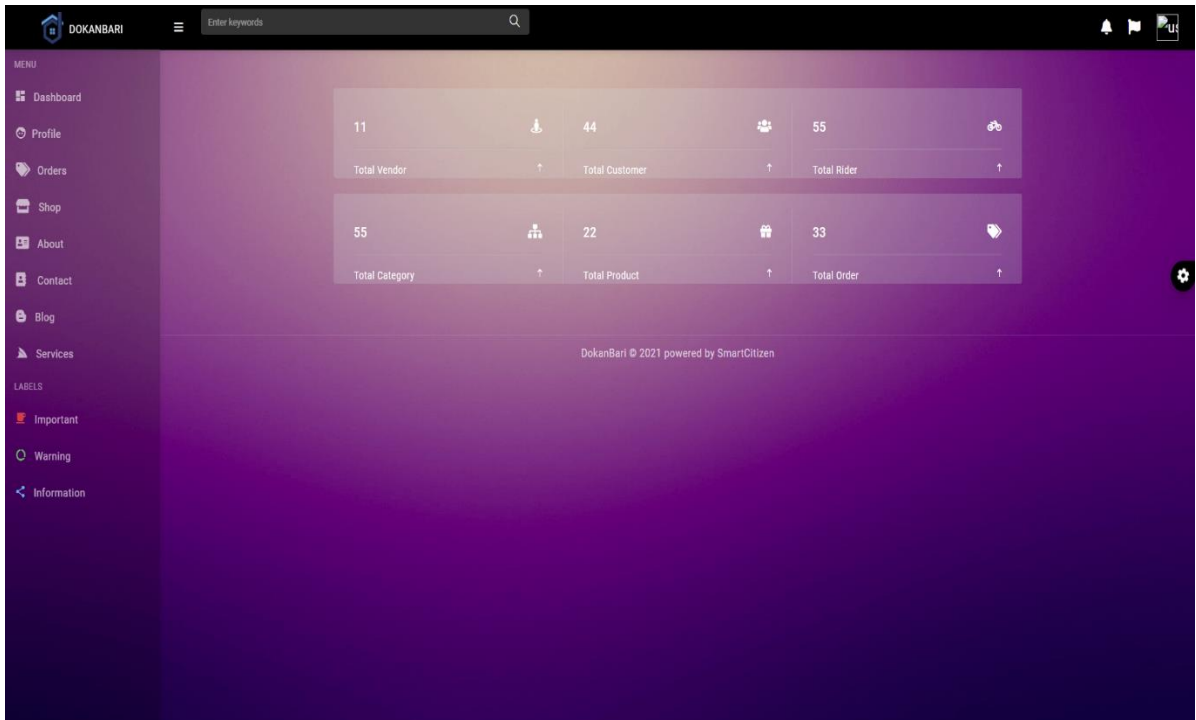
The screenshot shows the 'Update Shop' form in the admin interface. The form fields are as follows:

- SHOP CATEGORY: Computer
- SHOP NAME: Star Tech
- OPEN TIME: HH: 10 MM: 0
- CLOSE TIME: HH: 23 MM: 59
- SHOP DISCOUNT: 2
- SHOP BRANCH: Pangsha
- SHOP: Active
- STATUS: Accepted
- SHOP BANNER: Choose File (No file chosen)

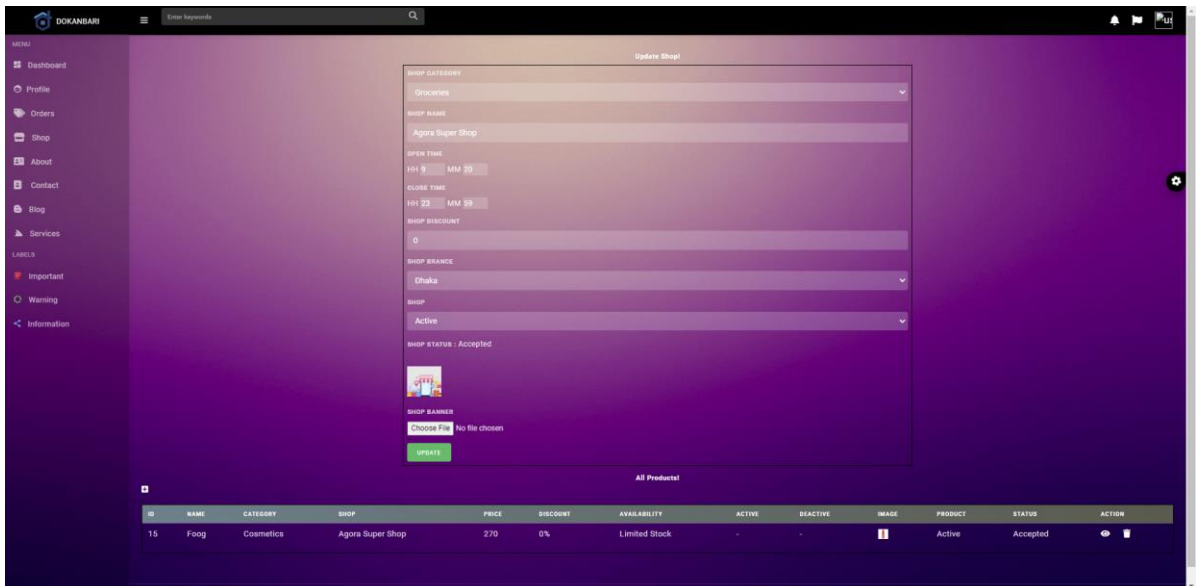
Below the form, there is a section for 'All Products' which contains one entry:

ID	NAME	CATEGORY	SHOP	PRICE	DISCOUNT	AVAILABILITY	ACTIVE	DEACTIVE	IMAGE	PRODUCT	STATUS	ACTION
4	FXF AMD RDNA 2 16GB	Computer Component	Star Tech	150000	0%	In Stock	-	-		Active	Accepted	

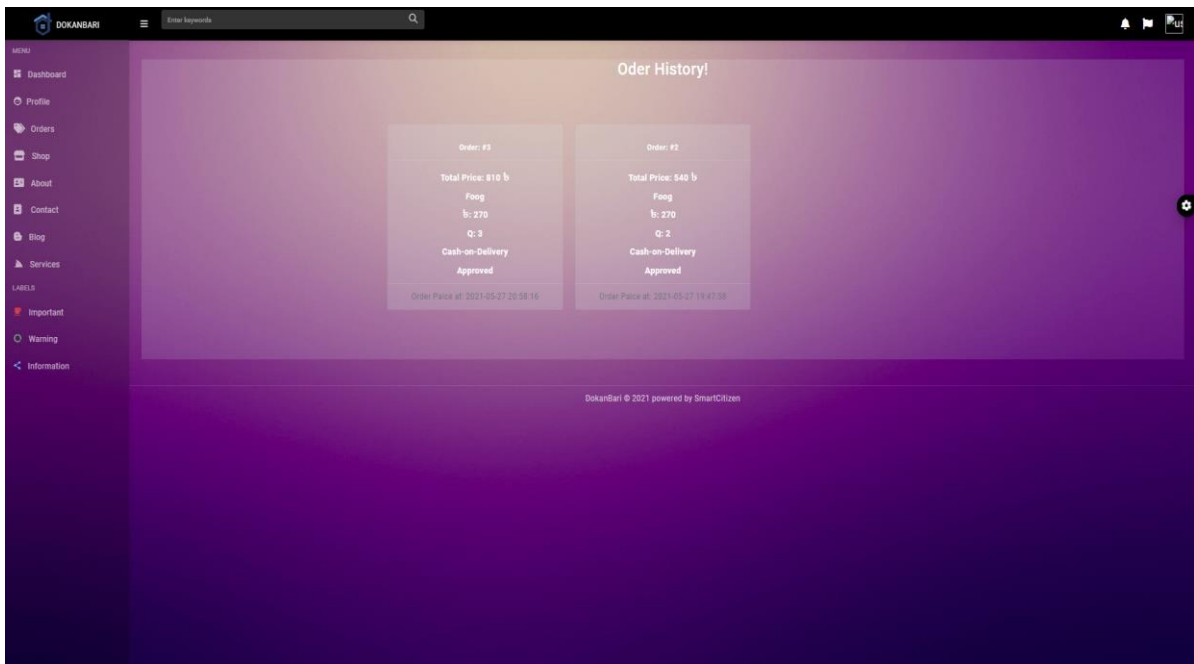
6.15 Dashboard (Vendor)



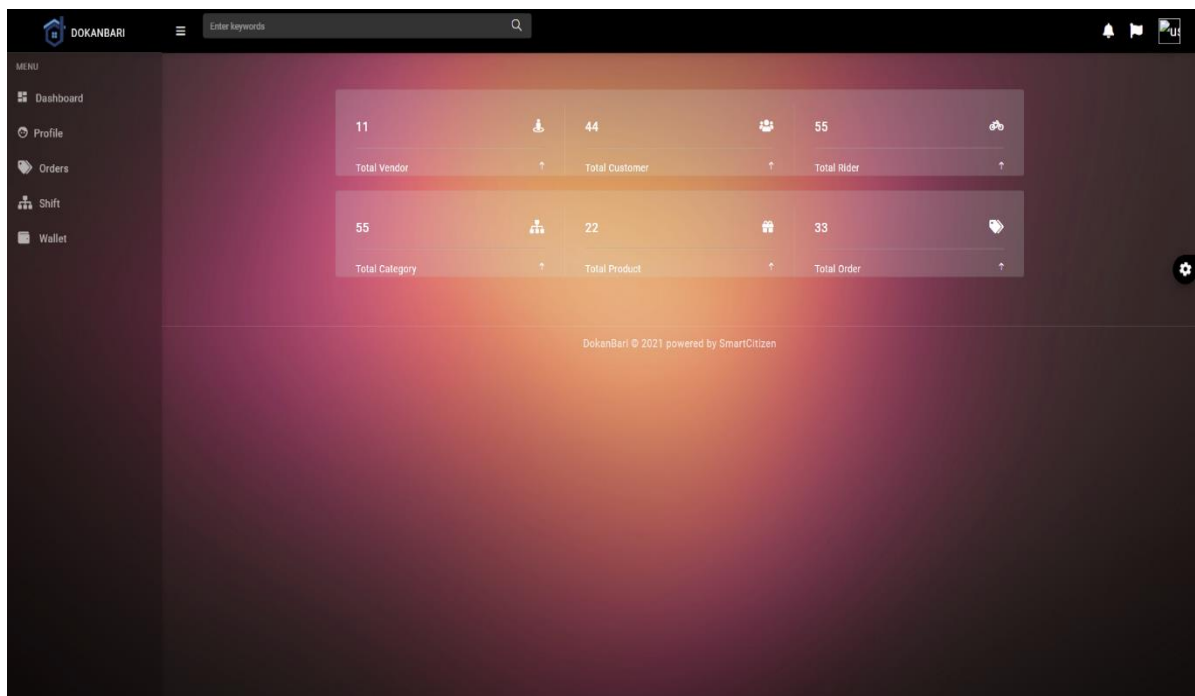
6.16 Vendor Shop Update and Products (Vendor)



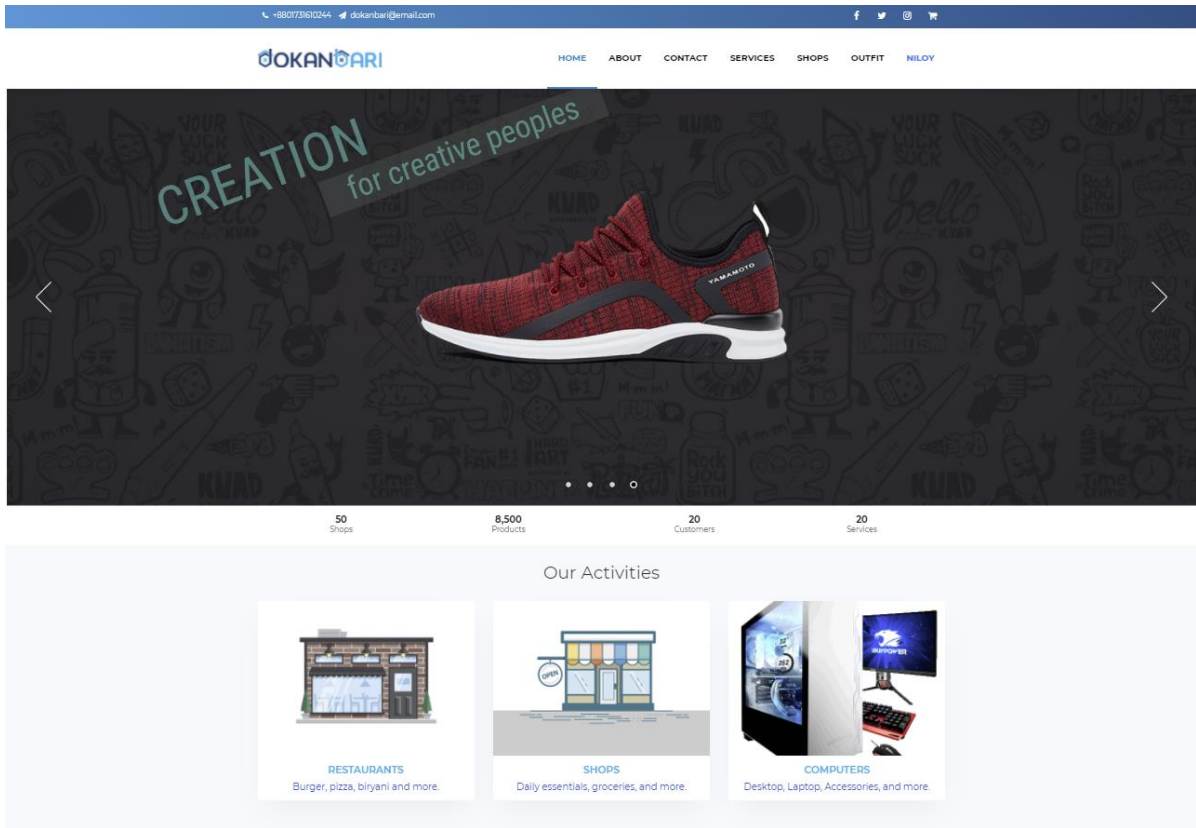
6.17 Vendor Orders (Vendor)



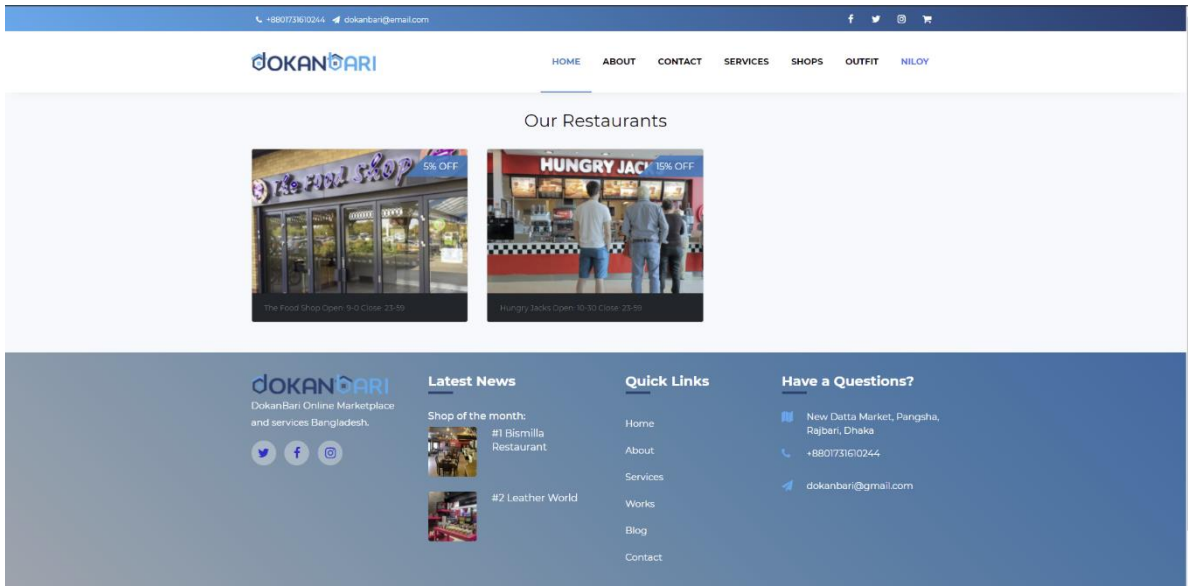
6.18 Dashboard (Rider)



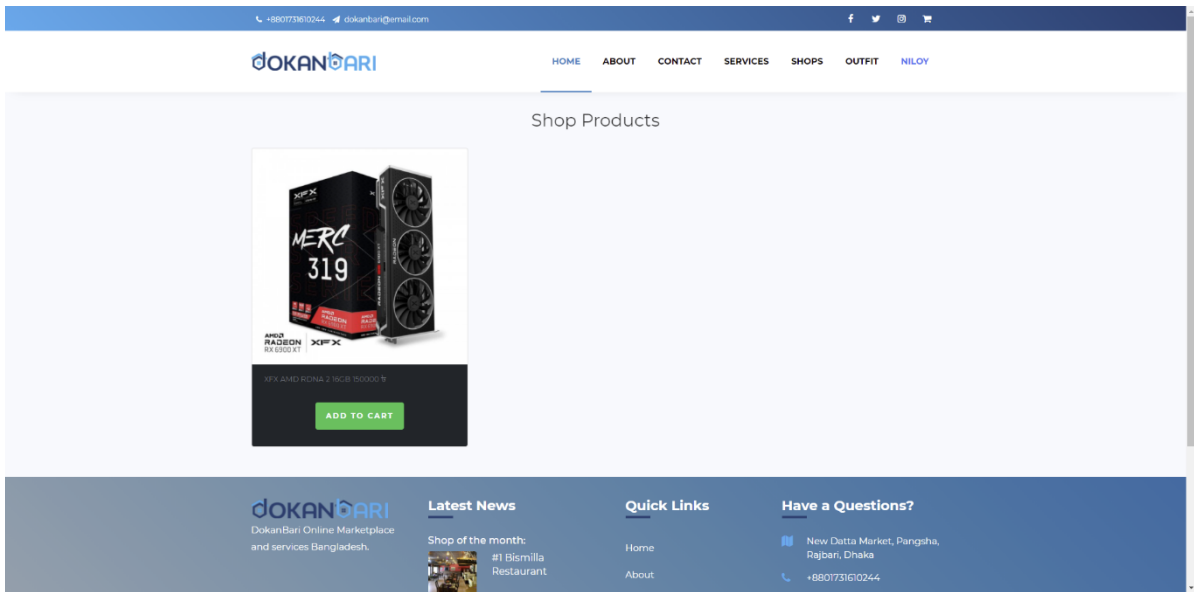
6.19 Home (Customer)



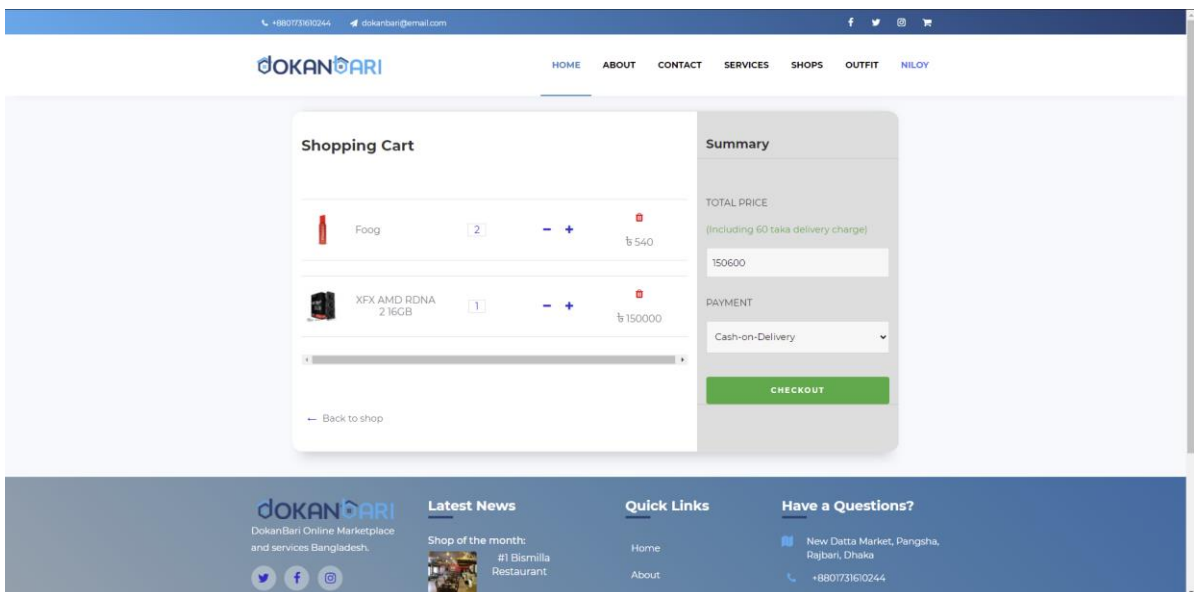
6.20 Restaurants (Customer)



6.21 Products (Customer)



6.22 Carts (Customer)



6.23 Customers Orders (Customer)

The screenshot shows a web application interface for a customer's order history. The page is titled "Oder History!" and displays six orders in a grid. Each order card shows the order number, total price, items, quantities, payment method, and status.

Order #	Total Price	Items	Quantities	Payment Method	Status
Order: #6	150600 ₹	Foog, XFX AMD RDNA 2 16GB	Q: 2, Q: 1	Cash-on-Delivery	Pending
Order: #5	600 ₹	BBQ Pizza	Q: 2	Cash-on-Delivery	Approved
Order: #4	1900 ₹	Kacci Biryani, BBQ Pizza	Q: 1, Q: 1	Cash-on-Delivery	Approved
Order: #3	810 ₹	Foog	Q: 1	Cash-on-Delivery	Pending
Order: #2	540 ₹	Foog	Q: 1	Cash-on-Delivery	Pending
Order: #1	1600 ₹	Kacci Biryani	Q: 1	Cash-on-Delivery	Approved

CHAPTER: 07

7.1. GitHub Link: <https://github.com/AkashSheikh1996/SmartCitizen.git>

7.2. Limitations:

- It cannot make advertisement.
- In the ERD I show the relation between entities.

7.3. Obstacles & Achievements:

I use Laravel to do this project, I have lot of knowledge about this framework functionalities, Also, I have done many projects using this framework, more than seven months' and finally project is done.

7.4. Future Scope

In the future it can be included advertisement for any type of company business etc.

- Their own social platform.
- Also, a release of Android, IOS and windows application.

REFERENCES:

I have gained some knowledge from some platforms. Obviously, I will mention those references.

For making my project successful those resources help me a lot. Now I will mention the names below.

<https://stackoverflow.com/>

<https://getbootstrap.com/>

<https://laravel.com/>

<https://laracast.com/>

<https://udemy.com/>