



Daffodil
International
University

Shop Easy

Submitted by:

Mostafijur Rahman

ID: 172-35-2183

This Project report has been submitted in finishing of the requirements

for the Degree of

Bachelor of Science in Software Engineering.

Department of Software Engineering

Daffodil International University

Supervised by:

Asif Khan Shakir

Lecturer (Senior Scale)

Department of Software Engineering

Daffodil International University

Approval

This Project/Thesis titled “Shop Easy”, submitted by Mostafijur Rahman, ID: 172-35-2183 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.



Asif Khan Shakir

Lecturer (Senior Scale)

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University



Mostafijur Rahman

ID: 172-35-2183

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Acknowledgement

At first, I would like to thank my respectable Dr. Imran Mahmud, Associate Professor (Head In-Charge), Department of SWE, Daffodil International University. It would not be possible to come so far without him.

I want to applause and appreciate my respectable supervisor Asif Khan Shakir, Lecturer (Senior Scale). Without his kind help, it would not be possible to finish the project.

I want to thank all the faculty members of the Software Engineering Department of Daffodil International University. Especially my advisors Ms. Fatama Binta Rafiq, Lecturer, Department of Software Engineering, And Ms. Nusrat Jahan, Assistant Professor, Department of Software Engineering. Without their continuous support and valuable wisdom, it would not be possible to make it so far.

Finally, I want to thank the StackOverflow community. It would be very hard to go ahead without their help.

Declaration

I hereby declare that I have taken this project under the supervision of

Asif Khan Shakir, Lecturer (Senior Scale), Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree

Mostafijur Rahman

ID: 172-35-2183

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:

Asif Khan Shakir, Lecturer (Senior Scale)

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Abstract

People are getting busy day by day. Every day go to the market and buy your daily needs are time-consuming. Shop Easy would be a better solution to eliminate the problem. The goal of this project is to create an online marketplace for all categories of goods. People can buy all categories of goods in one place. Alongside this, It poses an opportunity for people who wants to run a business online.

Table of Contents

Approval.....	ii
Acknowledgement	iii
Declaration.....	iv
Abstract.....	v
Table of content.....	vi
List of Figures.....	ix
List of Tables.....	x
Chapter 1: Project Initiation	1
1.1 Introduction	1
1.2 Background of the project.....	1
1.3 Problem with current system	1
1.4 Purpose and scope	1
1.5 Stakeholders.....	2
1.6 Conclusion.....	2
Chapter 2: Pre Analysis	2
2.1 Introduction.....	2
2.2 foodpanda.com.bd.....	2
2.3 chaldal.com.....	3
2.4 swapno.com.....	3
2.5 Conclusion.....	3
Chapter 3: Feasibility Study	3
3.1 Introduction.....	3
3.2 Economical feasibility.....	3

3.3 Technical feasibility.....	4
3.4 Conclusion.....	4
Chapter 4: System Analysis	4
4.1 Functional requirements.....	4
4.2 Non Functional requirements.....	10
4.2.1 Performance requirements.....	10
4.2.2 Capacity requirements.....	10
4.2.3 Availability requirements	10
4.2.4 Security requirements.....	11
4.2.4.1 Access requirements.....	12
4.3 Use Case Diagram.....	13
4.4 Use Case Description.....	14
Chapter 5: System Design.....	21
5.1 Introduction.....	21
5.2 High level design.....	21
5.3 Low level design.....	22
5.3.1 Introduction.....	22
5.3.2 Database schema Diagram.....	23
5.3.3 ERD Diagram.....	24
5.3.4 Sequence Diagram.....	25
Chapter 6: Development.....	29
6.1 Introduction.....	29
6.2 Coding platform.....	29
6.3 Reason behind Choosing	29
6.3.1 In general	29

6.3.2 On the project perspective	29
6.4 Conclusion	30
Chapter 7: System UI.....	30
7.1 Home Page.....	30
7.2 Delivery Man Dashboard	31
7.3 Delivery Man Orders.....	31
7.4 Login.....	32
7.5 Sign Up.....	32
7.6 My shops	33
7.7 Shop dashboard.....	33
Chapter 8: Critical Appraisal	34
8.1 Introduction.....	34
8.2 Strength of the system.....	34
8.3 weakness of the system.....	34
8.4 Conclusion.....	34
Reference.....	35

List of Figures

4.3.1	Use case diagram for Shop Easy	13
5.3.1	Class Diagram	22
5.3.2	Database Schema Diagram	23
5.3.3	ERD Diagram	24
5.3.4	Sequence Diagram for Add product	25
5.3.5	Sequence Diagram for Cancel order	25
5.3.7	Sequence Diagram for Change Order status	26
5.3.8	Sequence Diagram for Check my orders	26
5.3.9	Sequence Diagram for Create shop	27

List of Tables

4.1.1	Customer Registration	4
4.1.2	Seller Registration	4
4.1.3	Delivery man Registration	5
4.1.4	Search Shop	5
4.1.5	Filter Shop	5
4.1.6	Filter product	6
4.1.7	Remove product from shopping cart	6
4.1.8	Remove product from shopping cart	6
4.1.9	Check orders	7
4.1.10	Accept order	7
4.1.11	Create shop	7
4.1.12	List my shop	8
4.1.13	Mark an order as delivered	8
4.1.14	Mark an order as pending	8
4.1.15	Add product	8
4.1.16	Edit a product	8
4.1.17	Delete product	9
4.1.18	Accept delivery status update	9
4.1.19		9
4.4.1	Use Case Description of Login	14
4.4.2	Use Case Description of place order	14
4.4.3	Use Case Description of cancel order	15
4.4.4	Use Case Description of Sign up	15
4.4.5	Use Case Description of Create Shop	16
4.4.6	Use Case Description of Add product	16
4.4.7	Use Case Description of delete product	17

Chapter 1: Project Initiation

1.1 Introduction

Most of the E-commerce platform of our country does not support multiple categories of the shop. Some of them are starting to support multiple categories of shops. But what they are missing is a wide range of filtering systems. People do not need to see all kinds of shops when they visit the system. By adding some filtering options they can filter out what they do not need can quickly speed up their buying process.

I realize that the application needs to handle multiple types of customers, so the application needs to manage an extensive amount of users at a time. Therefore I selected Nodejs as the backend framework. I followed MVC architecture for developing the application. As the application does not need complex set of queries and needs to handle an extensive amount requests, I used MongoDB for database which is really great for handling huge amount of requests.

1.2 Background of The Project:

Most of the E-commerce platforms are running in a back-dated way in Bangladesh. Their delivery systems are not fully digitalized, and for that, they often deprive of business insights. Most of them do not support multiple categories of shops. People love to get all goods in one place. This need is greatly solved by various mega malls. But still, there is no complete E-commerce solution in Bangladesh. Shop Easy aims to solve those existing problems.

1.3 Problem with Current Systems

E-commerce platforms that currently exist most of them do not support multiple categories of shops. Although few of them do support multiple categories, they failed to provide rich filtering options.

1.4 Purpose and Scope:

E-commerce platforms that currently exist most of them do not support multiple categories of shops. Although few of them do support multiple categories, they failed to provide rich filtering options.

People are moving to E-commerce day by day. Existing E-commerce platforms are rapidly failing to provide for people's needs. Shop Easy will provide a easy way of buying different kinds of products and a smooth delivery process. After all, this will create big opportunity for sellers and people who want to work as a delivery man.

1.5 Stakeholders:

- i. Company owner
- ii. Customer
- iii. Delivery man
- iv. Seller

1.6 Conclusion:

Finally, I believe Shop Easy will fill a vital corner of the E-commerce system by providing an easy way for shopping. Shop Easy will create opportunities for people who want to run an online business and for delivery men.

Chapter 2: Pre-analysis

2.1 Introduction

Before starting the project, I researched few E-commerce platforms in Bangladesh. Most of them are incomplete to provide an easy way of shopping.

2.2 foodpanda.com.bd:

Foodpanda is a popular food and grocery seller in Bangladesh. They support two categories of shops. It is easy to use. But where they are failing is providing a way to filter out what types of shop user do not need.

2.3 chaldal.com:

Chaldal is another popular grocery seller in Bangladesh. The UI of Chaldal is complex to use and does not provide easy way to of filtering.

2.4 shwapno.com:

Shwapno is a popular multiple category of product seller in Bangladesh. The UI of Shwapno is very complex to use.

2.5 Conclusion:

After analyzing these three platforms, I reached none of them are providing rich filtering options or has complex UI

Chapter 3: Feasibility Study

3.1 Introduction:

The trust in E-commerce platforms is increasing day by day. Developing an E-commerce platform that will provide an easy way of shopping will be beneficial for the owner of the application and the sellers

3.2 Economical Feasibility:

As this is going to be a huge project the development cost will be huge. The owner has to calculate a big amount for marketing. There are so many people out there who want to start a business but cannot because of the extra costs like shop rent, security bill, electricity bill, etc. But Shop Easy can eliminate that extra cost by running a business online. So it is expected that the platform will get more sellers than other E-commerce platforms. The platform needs to charge sellers a little for creating shops. Every order needs to charge a particular amount.

3.3 Technical Feasibility:

To develop the project, I selected Nodejs, Javascript, MongoDB Reactjs, Mapbox, Nestjs, Socket.io as the technology stack. I have sound knowledge of the above technologies. So it can be achieving in time.

3.4 Conclusion:

After analyzing the above scenario, we can come at that building the system will be feasible and sustain on the market.

Chapter 4: System Analysis

Software Requirement Specification:

4.1 Functional Requirements:

FR-01	Customer Registration
Description	Without completing registration a user cannot order in the system.
Stakeholders	Customer

Table 4.1.1: Customer Registration

FR-02	Seller Registration
Description	A user must need to register himself/herself as a Seller to use seller functionality of the system.
Stakeholders	Seller

Table 4.1.2: Seller Registration

FR-03	Delivery Man Registration
Description	A user must need to register himself/herself as a delivery man to use delivery man functionality of the system.
Stakeholders	Delivery Man

Table 4.1.3: Delivery Man Registration

FR-04	Search Shops
Description	A user can search shop of an area by providing the location.
Stakeholders	Customer

Table 4.1.4: Search Shop

FR-05	Filter Shops
Description	A user can filter out shops when searching shops by location by deselecting shop categories. This will help user to filter out unnecessary categories of shops
Stakeholders	Customer

Table 4.1.5: Filter Shop

FR-06	Filter Products
-------	-----------------

Description	A user can filter out products of a shop by selecting filter criteria. This will help user to filter out unnecessary products of a shop he/she see.
Stakeholders	Customer

Table 4.1.6: Filter Products

FR-07	Remove Product From Shopping cart
Description	Customer can remove a product or reduce the quantity of a product. The Customer must have at least one product in his/her shopping cart to use this functionality.
Stakeholders	Customer

Table 4.1.7 Remove product from shopping cart

FR-08	Add Product to Shopping cart
Description	Customer can add product to shopping cart as much as he/she want.
Stakeholders	Customer

Table 4.1.8: Add product to shopping cart

FR-09	Check Orders
Description	A logged in customer can view his/her order history. This will help the customer to see his/her previous orders.
Stakeholders	Customer

Table 4.1.9: Check orders

FR-10	Accept Order
Description	A logged in customer can accept his/her pending order from respected delivery man
Stakeholders	Customer

Table 4.1.10: Accept order

FR-11	Create New Shop
Description	Logged in seller can create new shop of any category as much as he/she want
Stakeholders	Seller

Table 4.1.11: Create New Shop

FR-12	List My Shops
Description	Logged in seller can list his/her shops. This will help seller to list his/her shops.
Stakeholders	Seller

Table 4.1.12: List my shops

FR-13	Mark An Order As Delivered
Description	Seller can mark an order as delivered. This will send a request to respected delivery man to update the order status for that shop. The order must need be in pending status.
Stakeholders	Seller

FR-14	Mark An Order As Pending
Description	If seller mistakenly set an orders status to delivered, then he/she can update order status to pending. This will send a request to the respected delivery man he/she need to accept the request.
Stakeholders	Seller

FR-15	Add Product
Description	Seller can add product to a particular shop by using this functionality. Seller need to be logged in to use this functionality
Stakeholders	Seller

FR-16	Edit A Product
Description	Seller can edit products of a shop. To use this feature user need to logged in as a seller.
Stakeholders	Seller

FR-17	Delete Product
-------	----------------

Description	Seller can delete product from his/her shops
Stakeholders	Seller

FR-18	Accept Delivery Status Update
Description	A delivery man can accept status update request of an order of a shop. After Accepting status update request the status will be delivered. This will help delivery man to identify which product are delivered to him/her of a shop of an order.
Stakeholders	Delivery Man

FR-19	Decline Delivery Status Update
Description	A delivery man can decline delivery status update.
Stakeholders	Delivery Man

FR-20	Mark An Order As Reached
Description	A delivery man can update the delivery status of an order to Reached. This will send a request to the respected customer. The customer need to accept the request to update the status.
Stakeholders	Delivery Man

4.2 Non Functional Requirements:

4.2.1 Performance Requirements:

PR-01	Processing An Order
Description	This depends on the infrastructure. On a local computer this application can process (receive and send to seller and delivery man) 30 order in a second. This also depends on the user internet connection.
Stakeholders	Customer, Delivery Man, Seller

PR-02	Search Shop
Description	This also depends on the user internet connection. On a 1mbps internet connection it can take up to 1 second.
Stakeholders	Customer

4.2.2 Capacity Requirements:

The system is able to manage all the information about Customer, Seller, Shop, Products, Delivery man, and Application.

PR-03	The system will contain all the Seller, Customer, Shop, Delivery Man information
Description	The system will store all the dynamic data in to the database and static data into disk storage.
Stakeholders	Customer, Seller, Delivery man

4.2.3 Availability Requirements:

The system must need to available 8AM to 10PM.

DR-01	Must be available the system at 8AM to 10PM
Description	The system must be available at the business hours
Stakeholders	System Developer

4.2.4 Security Requirements:

This system has some security requirements. Like-

- i. Username/Password
- ii. Validation
- iii. Authentication
- iv. Authorization

4.2.4.1 Access Requirements:

SR-01	Access To Seller Module
Description	The system will check that the authenticated user is a valid seller before provide access to the Seller module of the system.
Stakeholders	Seller

SR-02	Access To Customer Module
-------	---------------------------

Description	The system will check that the authenticated user is a valid Customer before provide access to the Customer module of the system.
Stakeholders	Customer

SR-03	Access To Delivery Man Module
Description	The system will check that the authenticated user is a valid delivery man before provide access to Delivery Man the module of the system.
Stakeholders	Delivery Man

SR-04	Validate User Data
Description	The system will check all the user's data before processing.
Stakeholders	System Developer

4.3 Use Case Diagram:

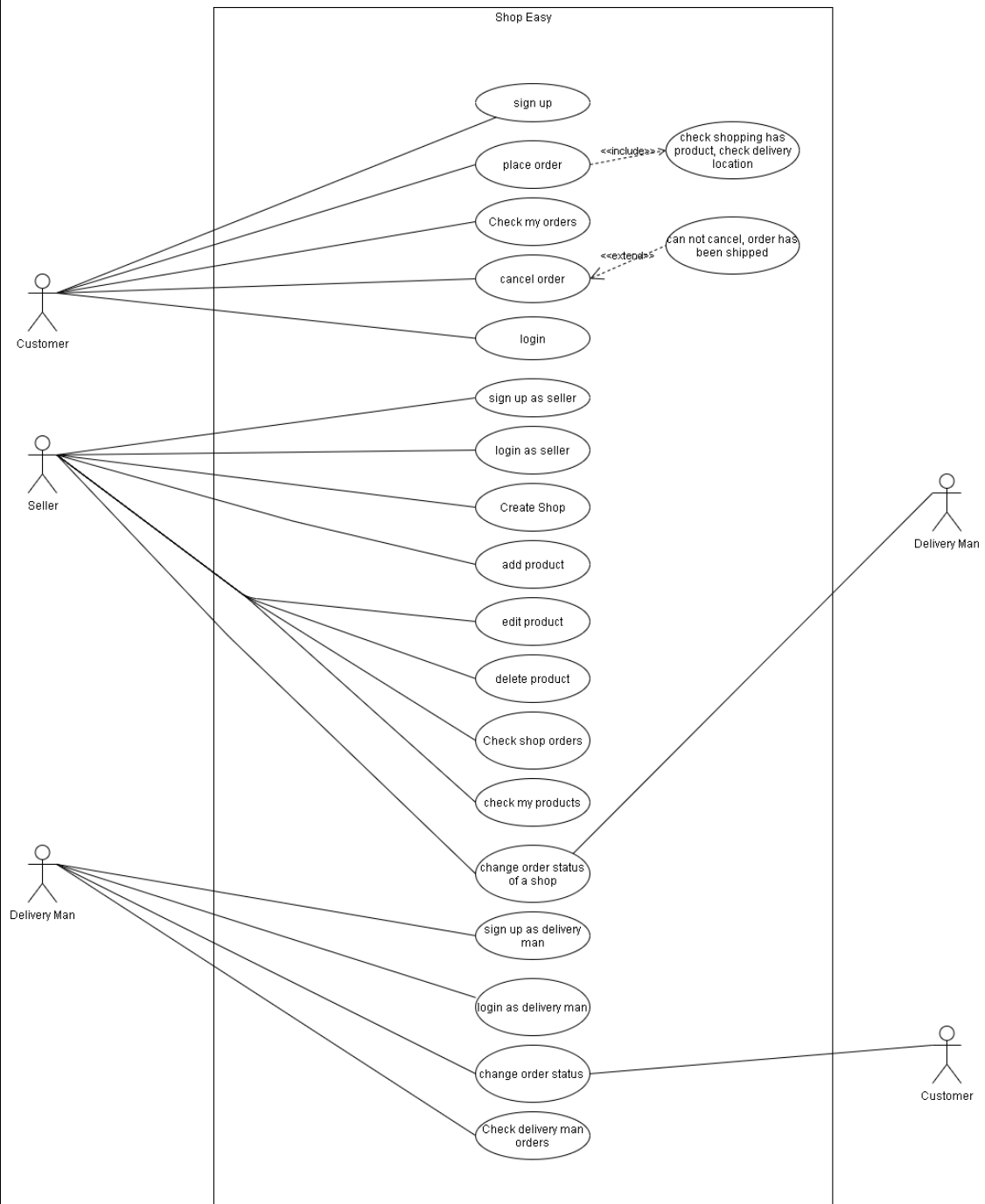


Figure 4.3.1 Use Case Diagram for Shop Easy

4.4 Use Case Description:

Use Case Id	UC-01	
Use Case Name	Login	
Scenario	Customer can login to access customer privileged modules.	
Preconditions	Customer must need to be registered	
Extending use case	None	
Included use case	None	
Actors	Customer	
Triggering event	Click in Login button	
Basic path	Step	Action
	1	Input email and password
	2	Click Login button
Alternative Flows	N/A	
Post Condition	After successful login customer will be redirected to the home page	

Table 4.4.1: Use Case Description Login

Use Case Id	UC-02
Use Case Name	Place Order
Scenario	Customer create new order
Preconditions	Customer must need to be logged in. Shopping cart must have at least one product and delivery location

Extending use case	None	
Included use case	Check Shopping cart has at least on product and delivery location	
Actors	Customer	
Triggering event	Click in place order button	
Basic path	Step	Action
	1	Customer add some product in the shopping cart
	2	Customer selects delivery location
	3	Click place order button
Alternative Flows	N/A	
Post Condition	After successfully order placed customer will see a message “order has been placed”	

Table 4.4.2 Use Case Description Place Order

Use Case Id	UC-03	
Use Case Name	Cancel Order	
Scenario	Customer can cancel order.	
Preconditions	Customer must need to be logged in	
Extending use case	Cannot cancel order	
Included use case	None	
Actors	Customer	
Triggering event	Click in Cancel order button	
Basic path	Step	Action

	1	Click My Orders Link
	2	Click Cancel Order Button
Alternative Flows	N/A	
Post Condition	After successfully Canceling order Customer will see a message “order has been canceled”	

Table 4.4.3 Use Case Description for Cancel Order

Use Case Id	UC-04	
Use Case Name	Sign Up	
Scenario	User can register against the system	
Preconditions	User email address must be new to the system	
Extending use case	None	
Included use case	None	
Actors	Customer, Seller, Delivery Man	
Triggering event	Click in Sign Up button	
Basic path	Step	Action
	1	Input email, name, password, confirm password
	2	Click Sign Up button
Alternative Flows	N/A	
Post Condition	After successful login customer will be redirected to the home page	

Table 4.4.4 Use Case Description for Sign Up

Use Case Id	UC-05	
Use Case Name	Create Shop	
Scenario	Seller will create shop	
Preconditions	Seller must need to be logged in	
Extending use case	None	
Included use case	None	
Actors	Seller	
Triggering event	Click in the Create Shop Link	
Basic path	Step	Action
	1	Open navigation menu
	2	Click Create shop link
Alternative Flows	N/A	
Post Condition	After successfully shop creation seller will be redirect to my shop page	

Table 4.4.5 Use Case Diagram for Create Shop

Use Case Id	UC-06
Use Case Name	Add Product
Scenario	Seller Will be able to add products to shop
Preconditions	Seller must need to be logged in. Seller must need to have shop.
Extending use case	None
Included use case	None

Actors	Seller	
Triggering event	Click in Login button	
Basic path	Step	Action
	1	Go to Shop Dashboard
	2	Click Add product button
	3	Fill required information
	4	Click Submit button
Alternative Flows	N/A	
Post Condition		

Table 4.4.6 Use Case Description for Add Product

Use Case Id	UC-07	
Use Case Name	Delete Product	
Scenario	Seller will be able to delete product from shop.	
Preconditions	Seller must need to have logged in. Seller need to have product in shop	
Extending use case	None	
Included use case	None	
Actors	Seller	
Triggering event	Click in Delete button	
Basic path	Step	Action
	1	Go to dashboard
	2	Click Products link

	3	Click Delete Icon on the product
Alternative Flows	N/A	
Post Condition		

Table 4.4.7: Use Case Description for Delete Product

Use Case Id	UC-08	
Use Case Name	Check Shop Orders	
Scenario	Seller will be able to get his/her shop orders	
Preconditions	Seller Must need to be logged in. Seller must need to have shop.	
Extending use case	None	
Included use case	None	
Actors	Seller	
Triggering event	Click in Orders link	
Basic path	Step	Action
	1	Go to dashboard
	2	Click orders link
Alternative Flows	N/A	
Post Condition	After successfully getting orders seller will be redirected to the orders page	

Table 4.4.8: Use Case Description for Check Shop Orders

Use Case Id	UC-09
-------------	-------

Use Case Name	Change Order Status of a Shop	
Scenario	Seller will be able to change order status to delivered or pending	
Preconditions	Seller must need to be registered	
Extending use case	None	
Included use case	None	
Actors	Seller, Delivery Man	
Triggering event	Click in Change Status button	
Basic path	Step	Action
	1	Go to Orders
	2	Click Mark as delivered
Alternative Flows	N/A	
Post Condition	After successfully Changing status seller will be notified that.	

Table 4.4.9: Use Case Description for Change Order Status of a Shop

Use Case Id	UC-10
Use Case Name	Change Order Status
Scenario	Delivery man will be able to change status of an order
Preconditions	Delivery man must need to be registered
Extending use case	None
Included use case	None
Actors	Delivery man, Customer
Triggering event	Click in mark as reached button

Basic path	Step	Action
	1	Go to dashboard
	2	Click mark as reached button
Alternative Flows	N/A	
Post Condition	After successfully Changing status Delivery man will be notified that.	

Table 4.3.10 Use Case Description Change Order Status

Chapter 5: System Design

5.1 Introduction:

This chapter consist of two main section. High level design and Low level design.

High Level Design section consist of Component Diagram and Low Level Design section consist of ERD Diagram, Database Schema Diagram, Class Diagram, Sequence Diagram.

5.2 High Level Design:

5.3 Low Level Design:

5.3.1: Class Diagram:

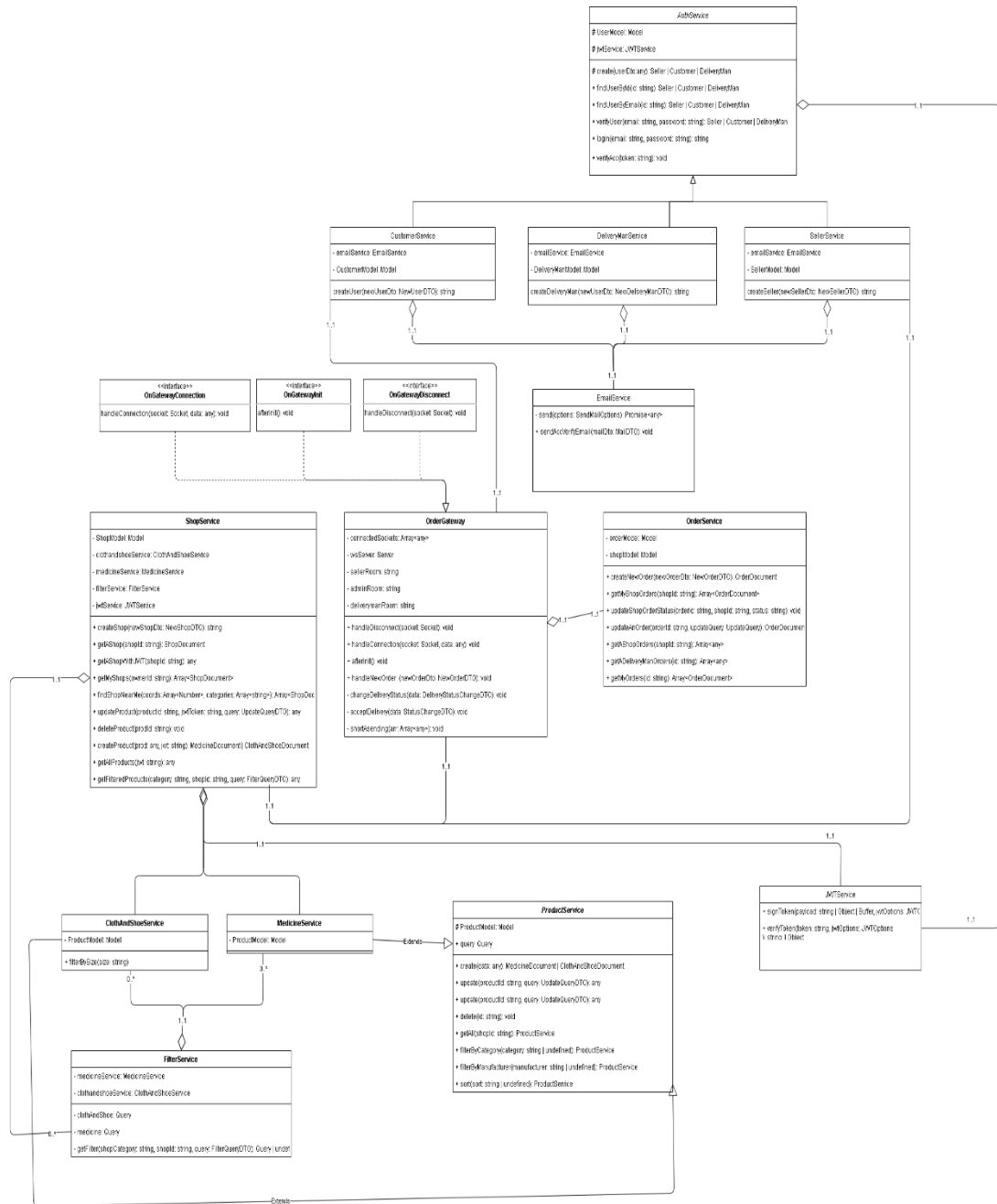


Figure: 5.3.1 Class Diagram for Shop Easy

5.3.2 Database Schema Diagram:

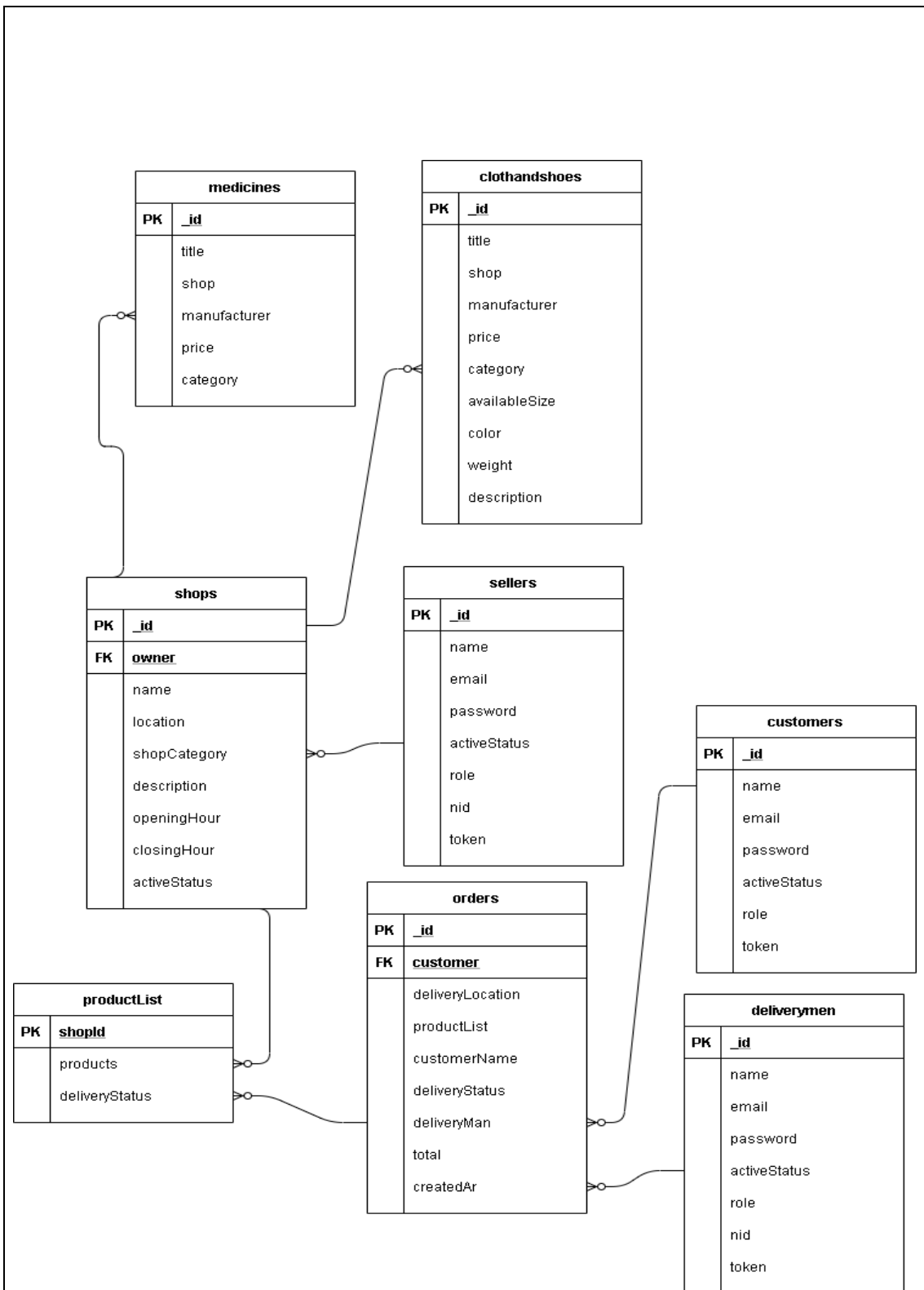


Figure 5.3.2 Database Schema Diagram for Shop Easy

5.3.3 ERD Diagram:

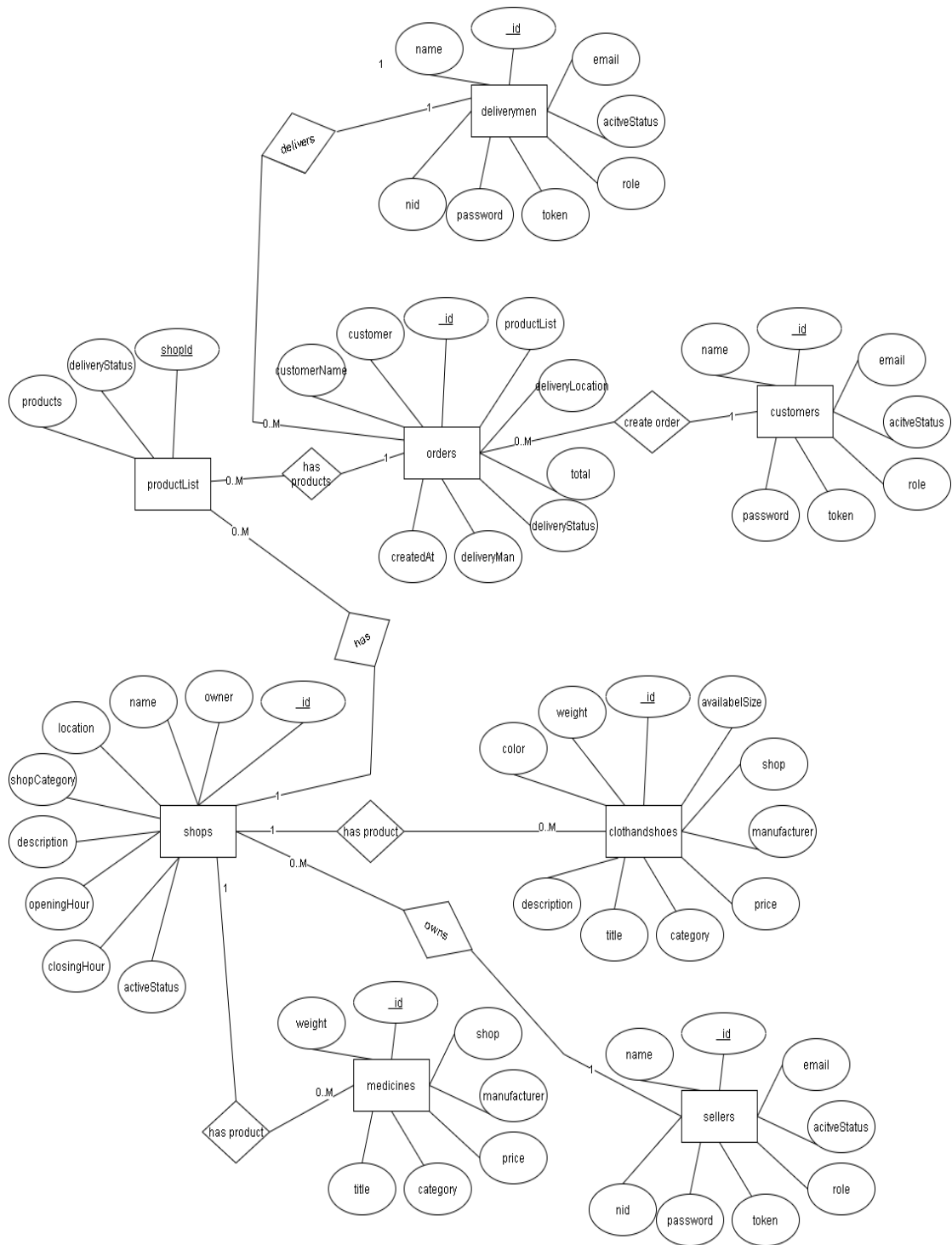


Figure 5.3.3 ERD Diagram for Shop Easy

5.3.4 Sequence Diagram:

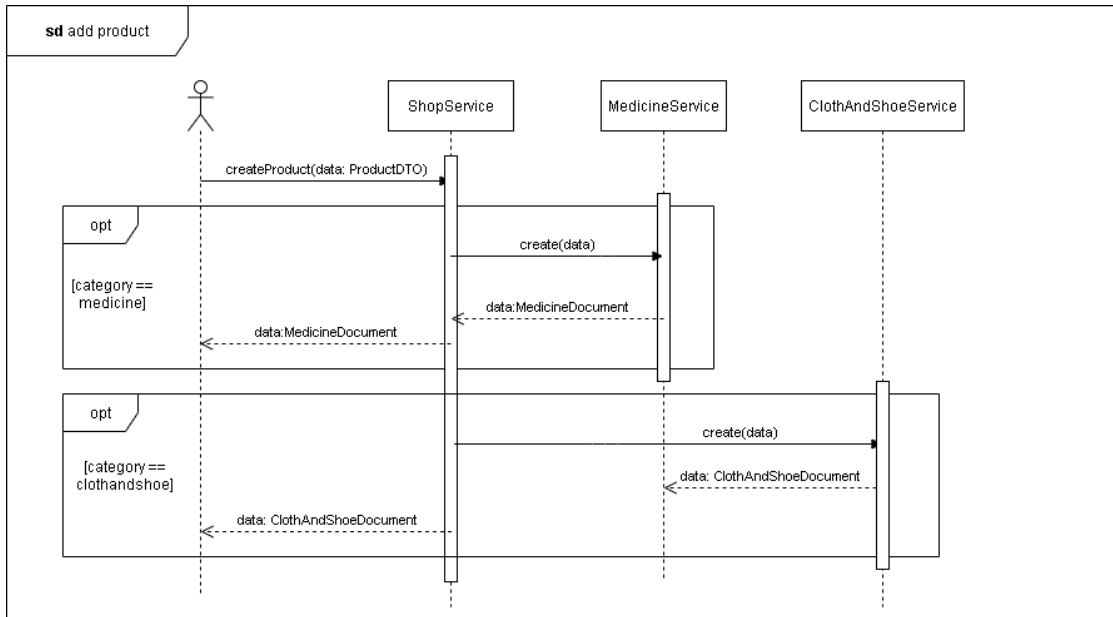


Figure 5.3.4 Sequence Diagram for Add Product

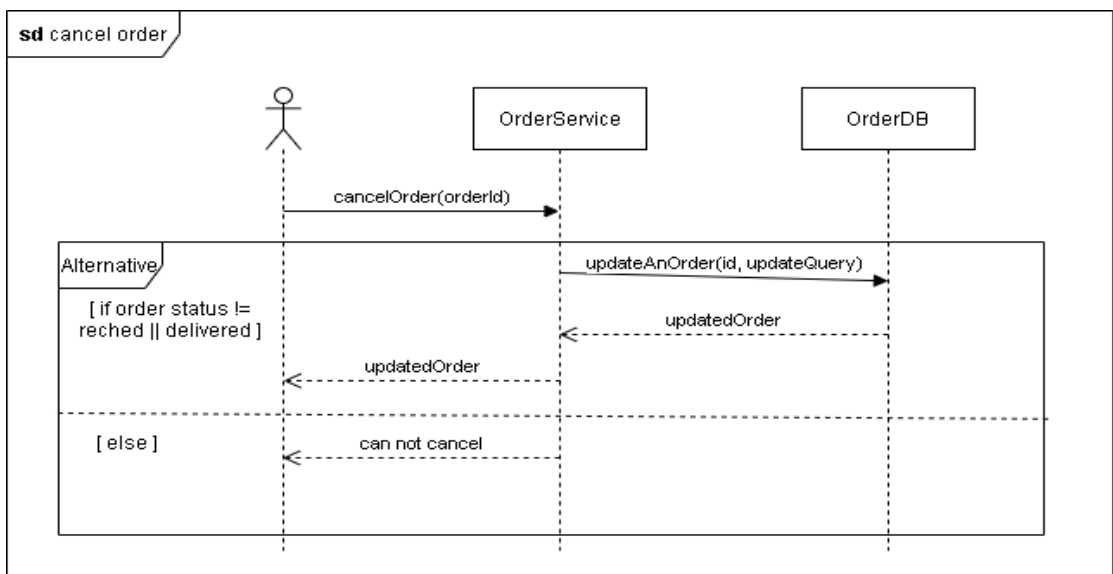


Figure 5.3.5 Sequence Diagram for Cancel Order

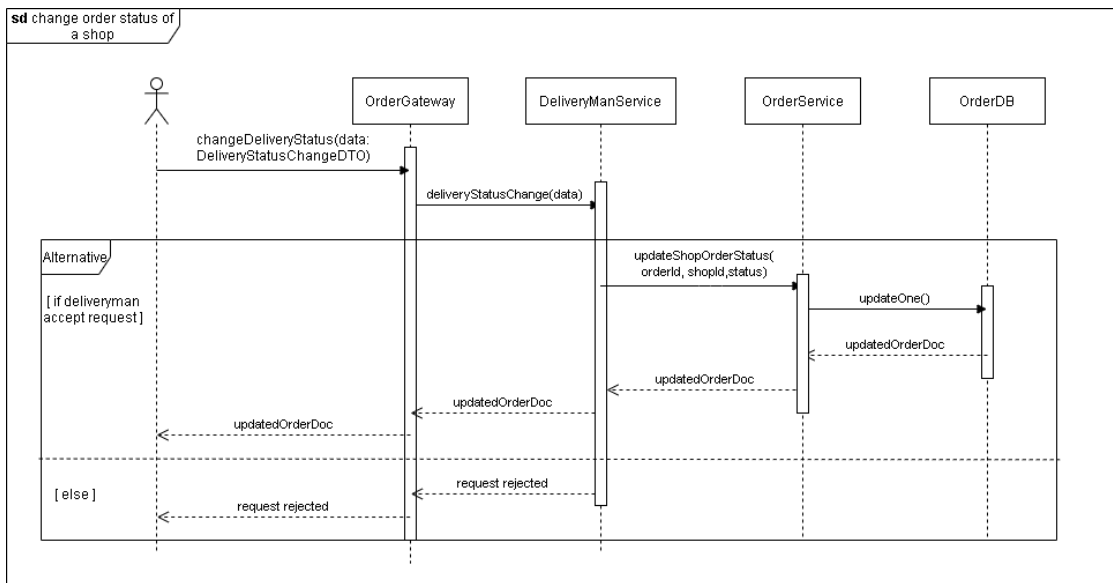


Figure 5.3.6 Sequence Diagram for Change Order Status of a Shop

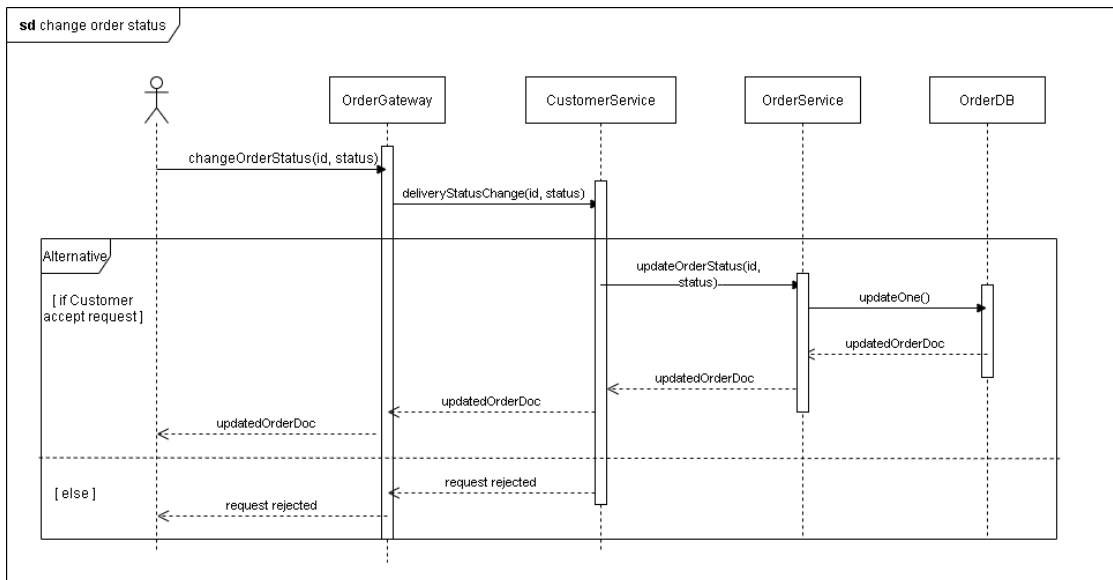


Figure 5.3.7 Sequence Diagram for Change Order Status

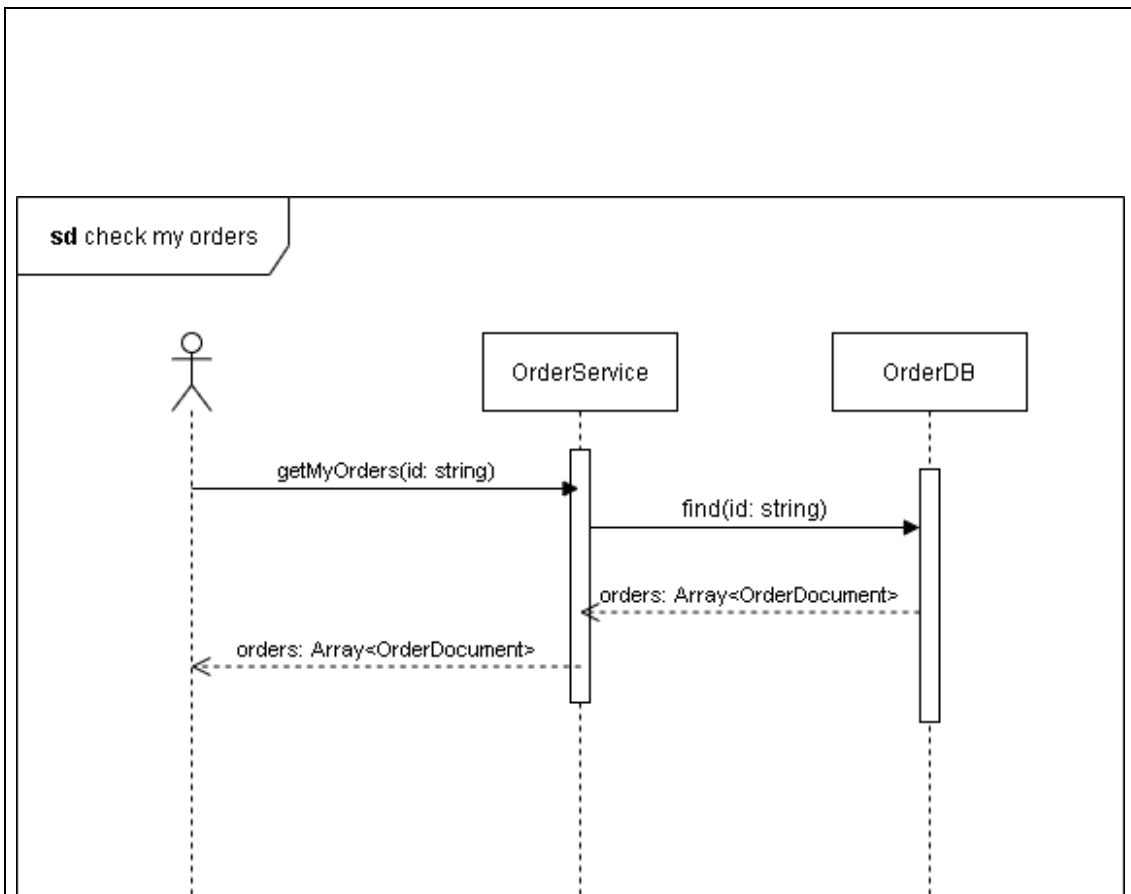


Figure 5.3.8 Sequence Diagram for Check My Orders

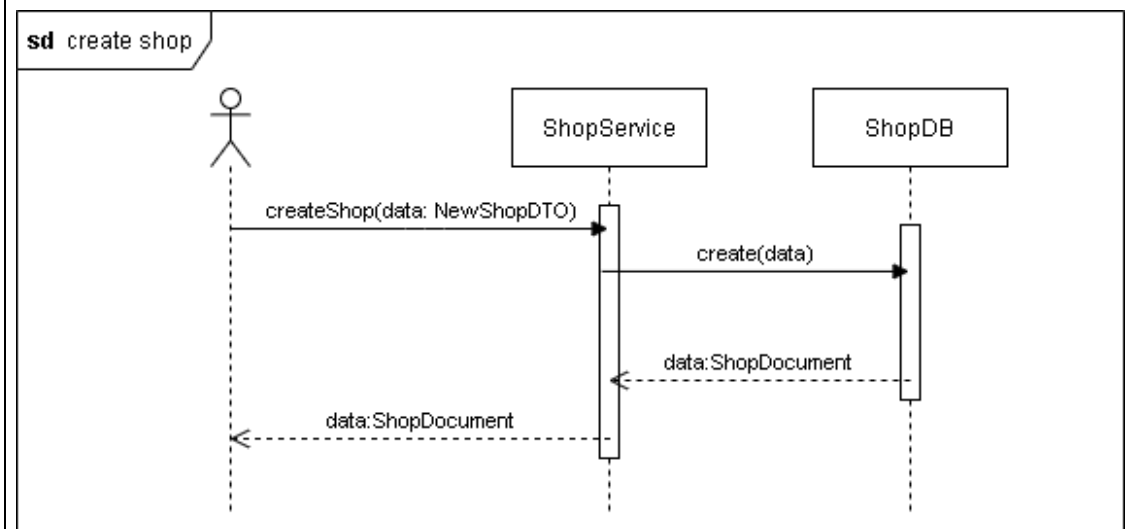


Figure 5.3.9 Sequence Diagram for Create Shop

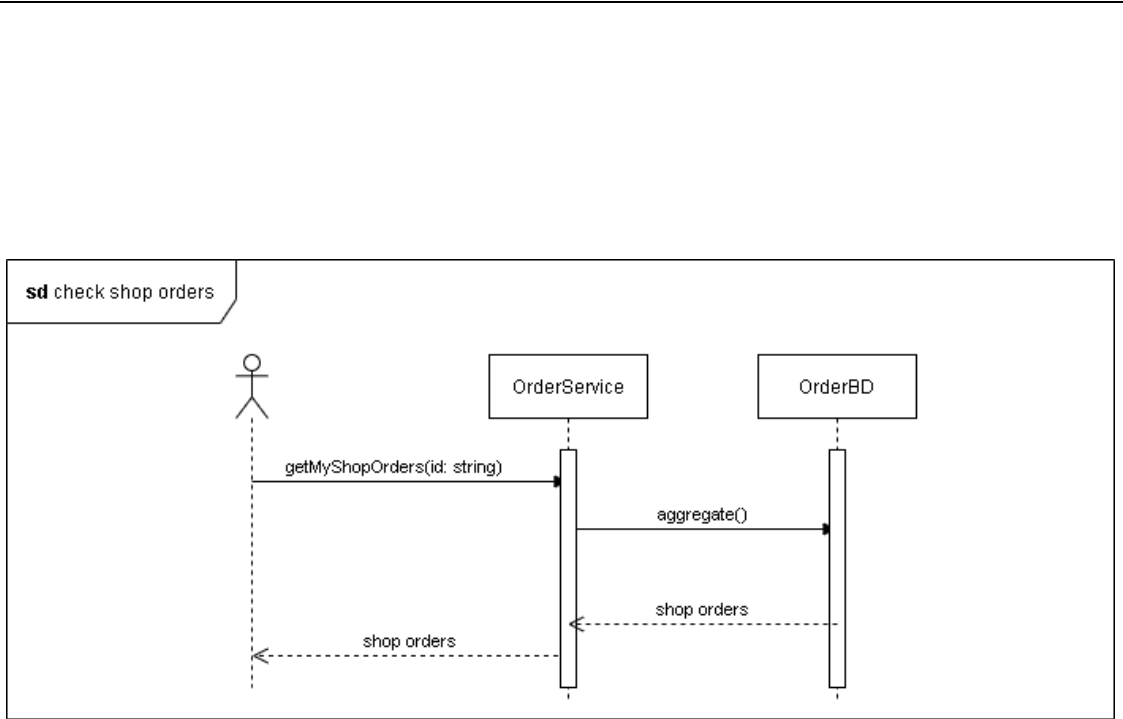


Figure 5.3.10 Sequence Diagram for Check Shop Order

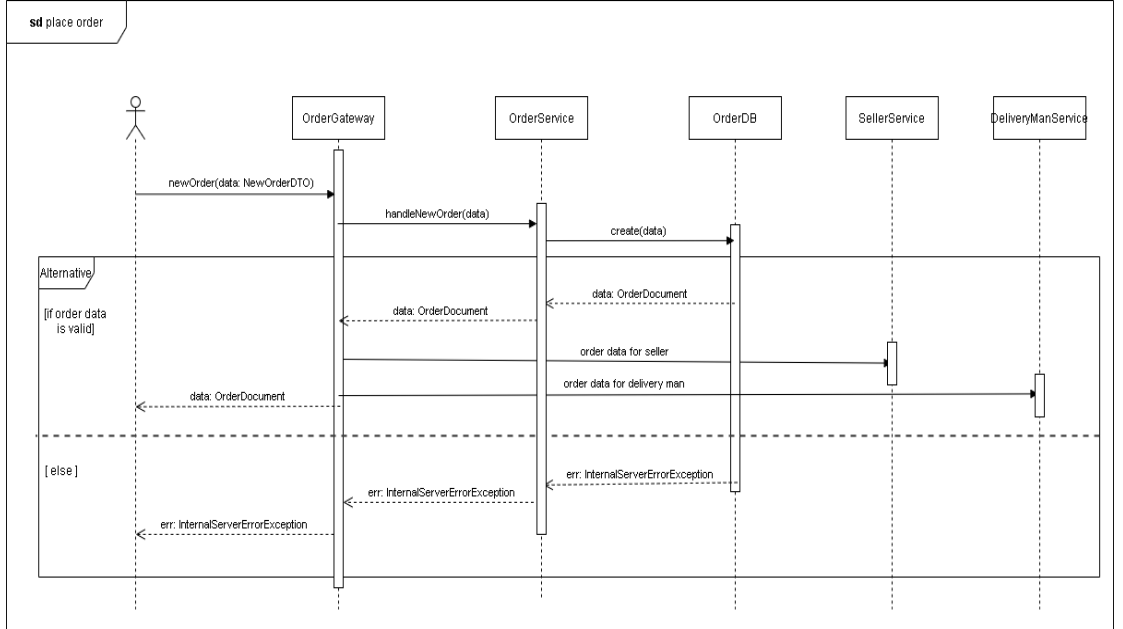


Figure 5.3.11 Sequence Diagram for Place Order

Chapter 6: Development

6.1: Introduction:

Before start developing the project I choose my programming language, backend framework, frontend framework, database.

6.2: Coding Platforms:

For this project I am using Javascript as the primary programming language, Nodejs (14.17.4) as my backend framework, Reactjs (17.0.2) as my UI rendering library, For Database I am using MongoDB. HTML5, CSS3, Material-UI

6.3 Reason Behind Choosing:

6.3.1: In General:

Javascript is the most popular programming language for web application development. It can be used with frontend applications as well as server applications. On the other hand, Nodejs is the most popular open-source runtime for Javascript. Big companies like Netflix, NASA, PayPal, Trello are using Nodejs. MongoDB is the most popular and easy-to-use document database. Reactjs is an open-source UI rendering library created by Facebook. It is greatly monitored by the company and continuously improving the library. Big companies like Uber Eats, Netflix, Airbnb is using Reactjs.

6.3.2 On The Perspective of This Project:

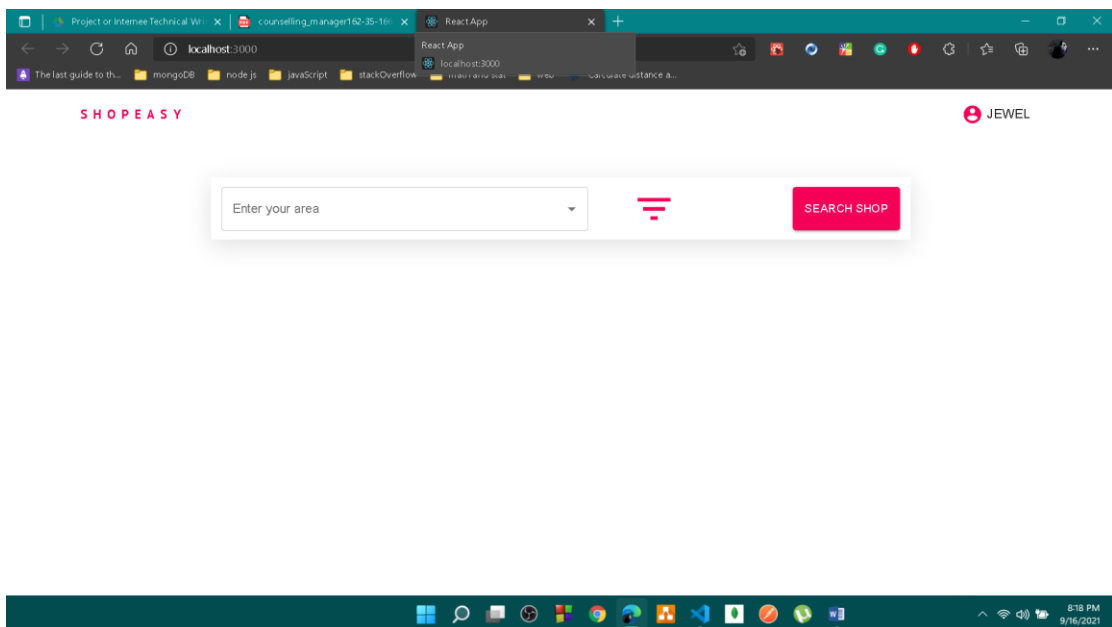
Javascript is a single-threaded programming language That is great for scaling applications. Nodejs uses Javascript as the programming language. This project needs to handle an extensive amount of users, so the application needs to scale well. MongoDB is a document database that scales well and is flexible. This application needs to handle an extensive amount of read, write operations to the database, and MongoDB is great for handling extensive amount of request.

6.4 Conclusion:

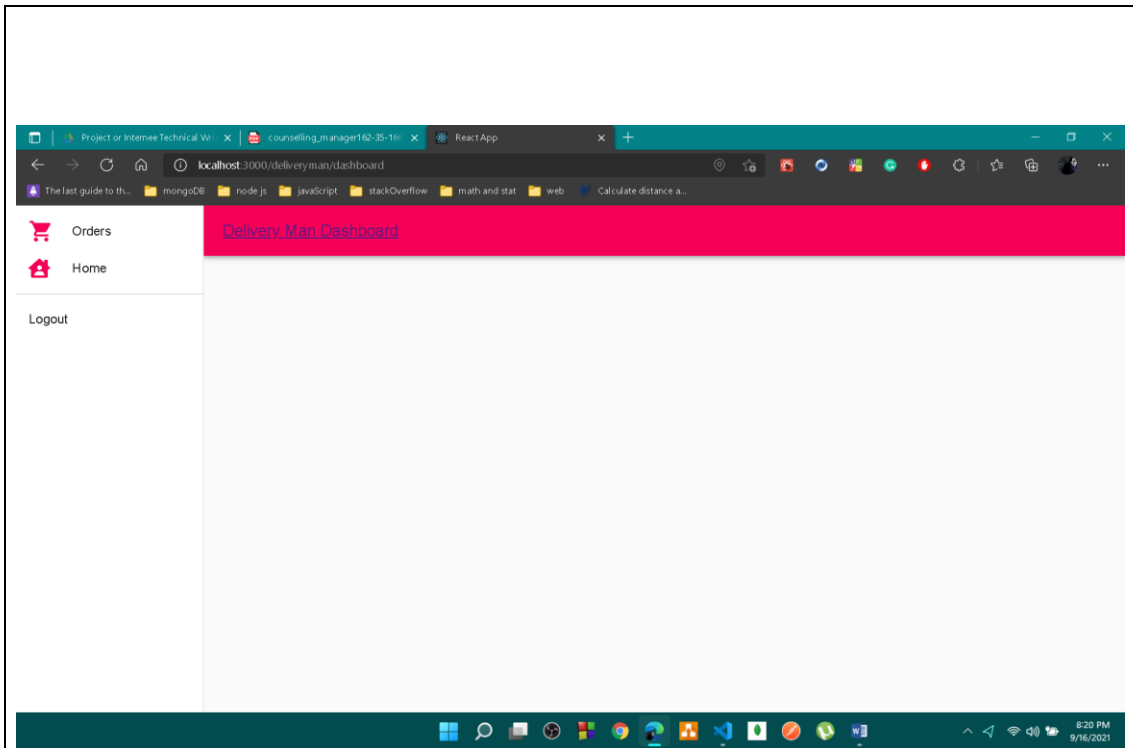
On the initial research I found the above technologies will be best for developing this project. In farther research if something needs to change it can be done.

Chapter 7: System UI

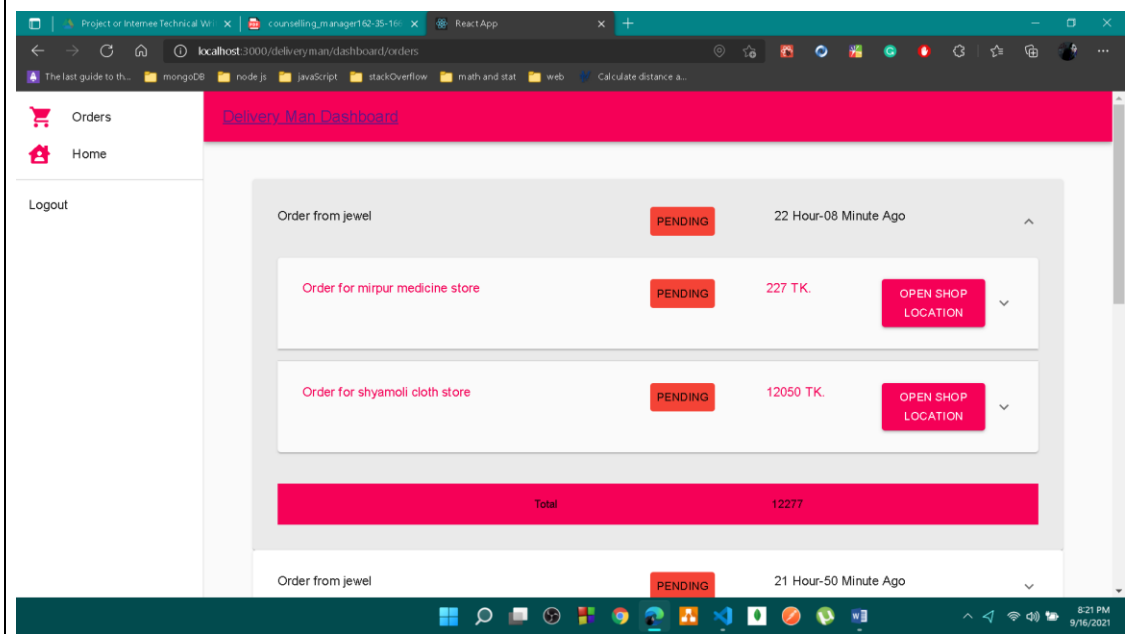
7.1 Home page



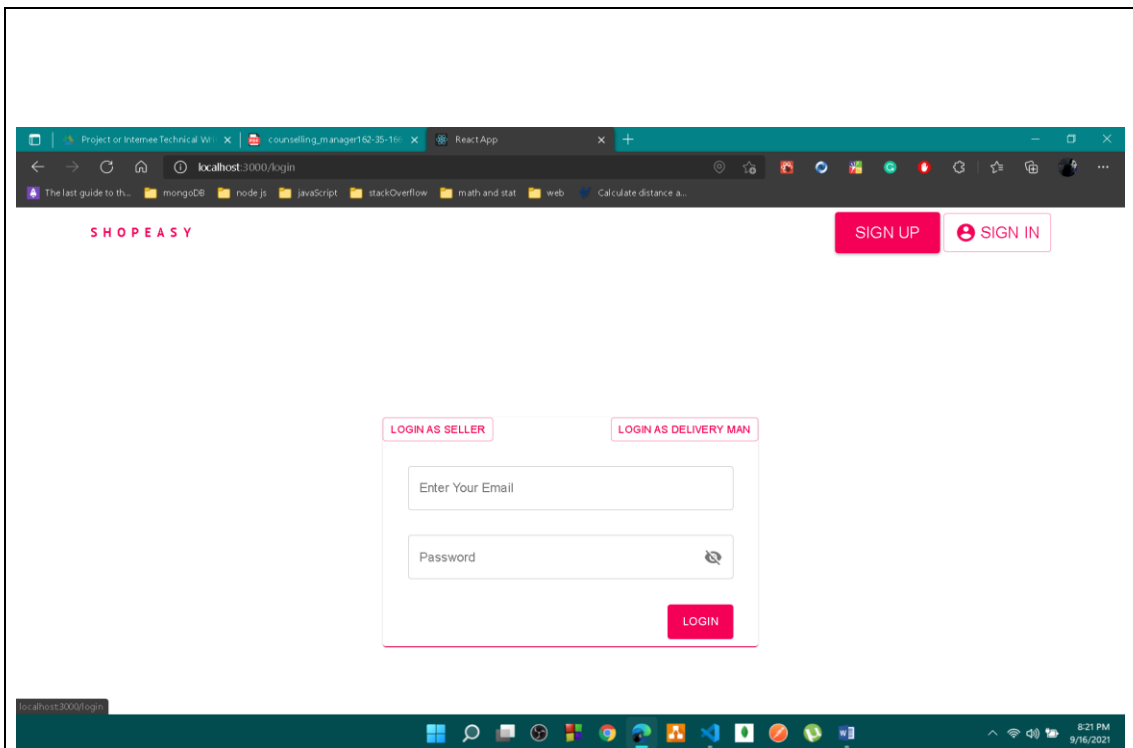
7.2 Delivery Man Dashboard



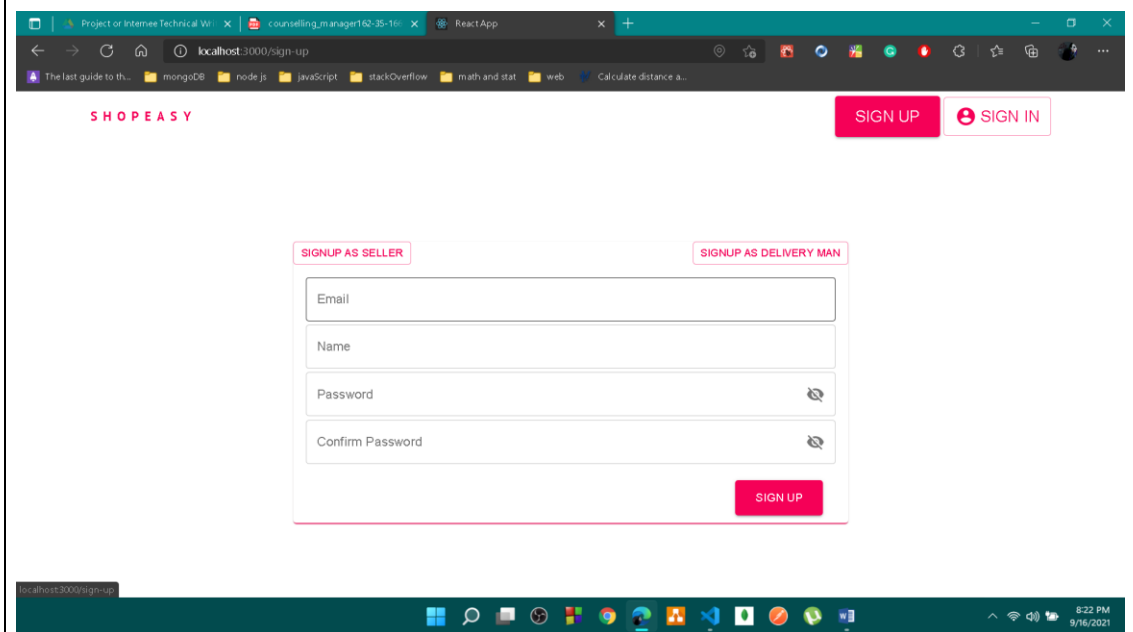
7.3 Delivery man orders



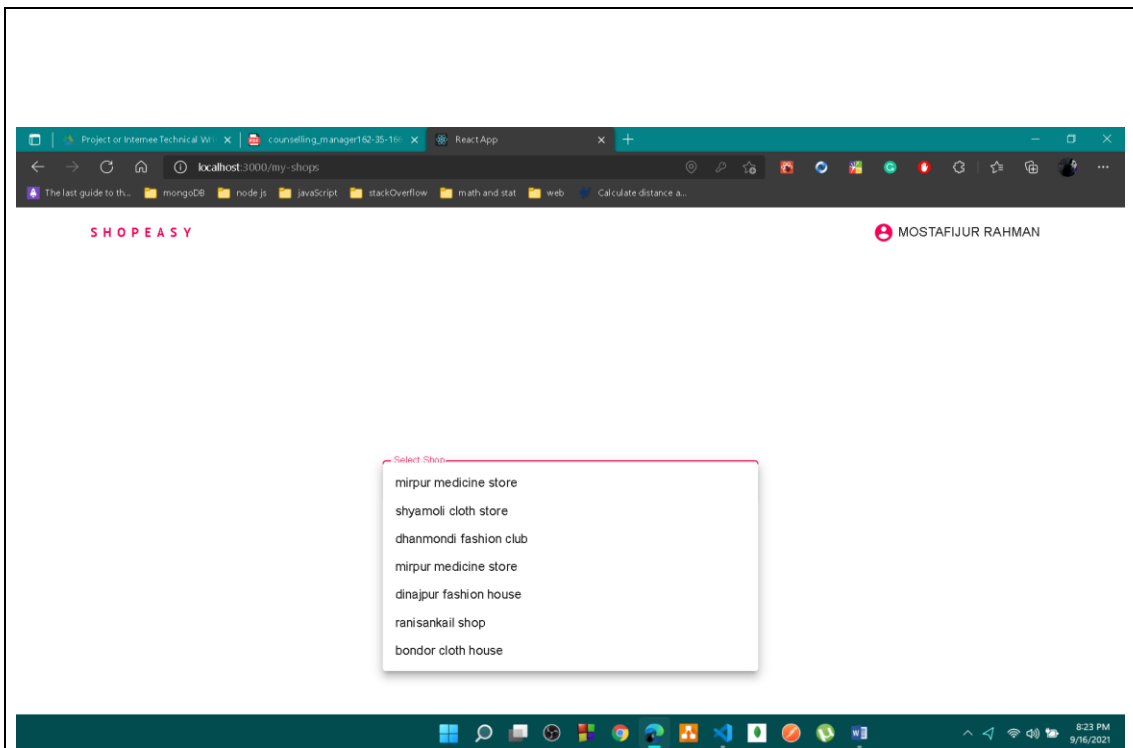
7.4 Login:



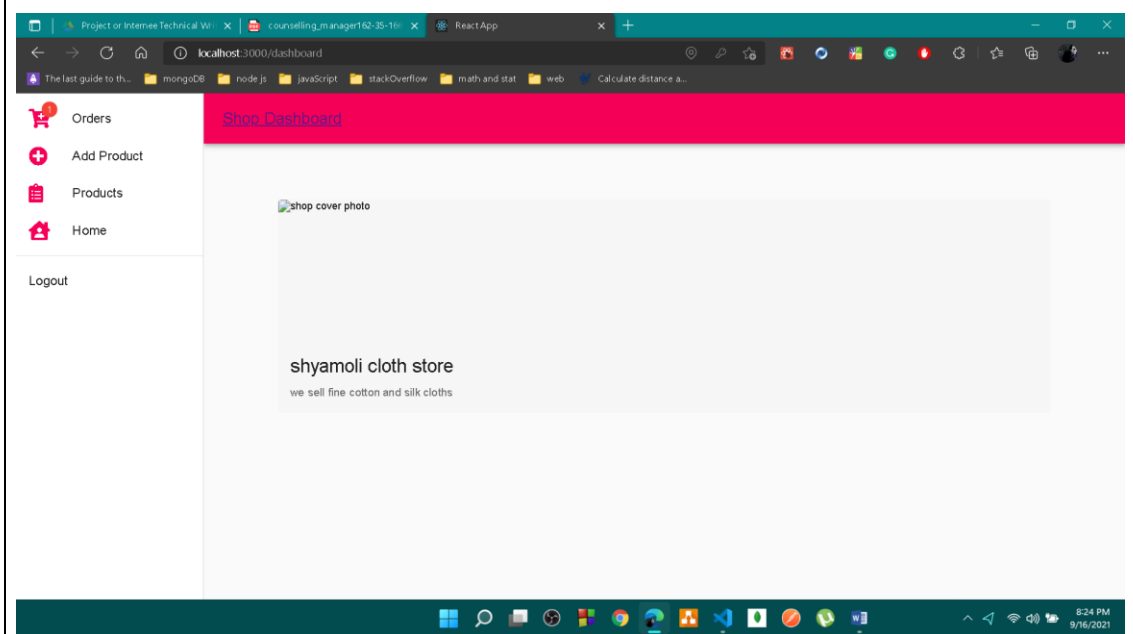
7.5 Sign Up:



7.6 My shops



7.7 Shop Dashboard



Chapter 8: Critical Appraisal

8.1 Introduction:

This project is aim solve a small but very critical lacking in current e-commerce platforms in Bangladesh. At this moment this project greatly solving the problem.

8.2 Strength of The System:

- i. **Scaling:** This system need to handle extensive amount of request. So this system is aim to scale well.
- ii. **Flexibility:** This system is fully flexible and easy to change.
- iii. **Security:** This system is build security in mind. Great amount of authorization and authentication is added. NoSql injection is prevented.

8.3 Weakness of The System

- i. **Not Responsive UI:** Currently this UI of this system is not responsive.
- ii. **Not Fully Featured:** This system need more features.
- iii. This system is using a free plan of Mapbox and Mailgun, so the more than 5000 request cannot sent to for map and email.

8.4 Future Updates:

- i. **More Shop Category:** In future this system will support more shop categories.
- ii. **Responsive UI:** The UI will be responsive in next update.
- iii. **Admin:** In future update an admin monitoring system will be added

References:

omg.org, 2007. OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2