Department of Computing and Information System (CIS)

Title of the Project:

Vegor Organic Food Supply

Course Code: CIS 499

Spring- 2022

Supervised By:

Mr. Abdullah Bin Kasem Bhuiyan

Lecturer of Computing & Information System

Daffodil International University

Submitted By:

Farjana Ahamed

ID: 182-16-320

Submission Date: 07-08-2022

ACKNOWLEDGMENT

Firstly, I want to explicit my gratitude and appreciation to Almighty Allah for helping me to finish the project successfully. After that, I want to thank my mother and siblings who showed me time for their support, courage, and empathy. The whole course without their sincere cooperation, I could not be able to finish my project successfully. I would love to explicit my special gratitude to all my teachers and most importantly my supervisor **Mr. Abdullah Bin Kasem Bhuiyan** of Daffodil International University, without their direction and feedback it is impossible to complete this system.

Finally, I would love to remember my late father who supported me to reach my goal. My friends also help me whenever I face many types of problems. I am so grateful to all of them.

DEDICATION

I have dedicated my project effort to my family & many friends. Especially, thanks to my caring parents, their encouragement and determination help me to reach my goal. I would like to dedicate my work to my father who died before my graduation. If he were alive, he could very happy to see my hard work. Besides, my sister **Faria Ahamed** was by my side and was very special. I have dedicated the work of this project to many of my friends and family who have supported me throughout the process. I will always praise them for what they have done**. Mehedi Hasan Rinku** in particular helped me master the leader points with my technical skills and many hours of proofreading.

©Daffodil International University

EXECUTIVE SUMMARY

Vegor Organic Food Supply (VOFS) is an academic project. The goal of deliver food products to particular customers. This project is for **Palal Group** where **Vegor Agri Complex Ltd.** is delivering their organic food products which they grow on their farms. They have some elite customers, and they deliver their grocery food products to the customer in a manual way. The customers' order products through a phone call, there is no online system. As there is no online system, the customer is not feeling free to order many types of products through a phone call, besides, the staffs manually keep the data, so it is problem and time-consuming to retrieve or search any customer data, other works, etc. So, **Vegor Organic Food Supply (VOFS)** is going to give a solution to **Vegor Agri Complex Ltd.** to their facing problems. Through the VOFS system, the customers will easily order food products online, view category and product details, add products to their cart, order and manage their order, pay through online-offline both, and request food for purchase in a high quantity. The system is only for particular users of Vegor Agri Complex Ltd.

©Daffodil International University

TABLE OF CONTENTS

CONTENTS **PAGE**

Acknowledgement……..……..……..i

Dedication…….……..…i

Executive Summary…….…….....ii

CHAPTER

CHAPTER 1 - INTRODUCTION 1-3

1. 1 Introduction.....1

1. 2 Document Contents 2

CHAPTER 2 – INITIAL STUDY.....3-6

2. 1 Project Proposal..... 3

2. 2 Background of the project 3

2. 3 Problem Areas.....6

2. 4 Possible Solutions.....6

CHAPTER 3 – LITERATURE REVIEW 7-10

3.1 Discussion on the Problem Domain.....7

3.2 Discussion on the Problem Solution: 8

3.3 Comparison among the leading solutions.....10

3.4 Recommended Approach.....10

CHAPTER 4 – METHODOLOGY.....11-16

4.1 What to Use 15

4.2 Why to Use.....15

4.3 Sections of Methodology..... 16

4.4 Implementation Plans..... 16

CHAPTER 5 – PLANNING 17-20

5.1 Project Plan.....17

5.1.1 Work Breakdown Structure 17

©Daffodil International University

5.1.2 Resource Allocation 18
5.1.3 Time Boxing 19
5.1.4 Gantt Chart 20
5.2 TEST PLAN 20-23
5.2.1 Testing Against the Time Boxes..... 21
5.2.2 Required Test..... 22
5.2.3 Test Case..... 22
5.2.4 User Acceptance Test Plan 23
5.3 RISK MANAGEMENT 24-26
5.3.1 Risk Identification & Assessment..... 25
5.3.2 Risk Precaution / Action Plan.....26
5.3.3 Steps Taken for Possible Risks..... 26
5.4 CHANGE MANAGEMENT.....27-28
5.4.1 Factors that Might Cause Change..... 27
5.4.2 DSDM Welcome Change..... 27
5.4.3 Considering Business Priority..... 28
5.4.4 Change Workshop..... 28
5.4.5 Changes that are allowed.....28
5.4.6 Key Decision Taker of Change 28
5.**5 QUALITY MANAGEMENT.....…29**
5.5.1 Rules Applied to Maintain Quality..... 29
5.5.2 DSDM Standard Quality Measures..... 29
5.5.3 Quality Plan and Measuring Meter..... 29
CHAPTER 6 – FEASIBILITY..... 30-34
6.1 All Possible Types of Feasibility..... 32
6.2 Cost Benefit Analysis..... 34
6.3 Is DSDM Good or Bad for this Project..... 34
CHAPTER 7 – FOUNDATION 35-40

©Daffodil International University

7.1 The Problem Area Identification..... 35
7.1.1 Interview..... 35
7.1.2 Observations..... 35
7.1.3 Questionnaires..... 36
7.2 Rich Picture..... 37
7.3 Specific Problem Area Identification..... 37
7.4 Possible Solutions..... 37
7.5 Overall Requirement List..... 38
7.6 Technology to be implemented..... 40
7.7 Recommendation and Justification 40
CHAPTER 8 – EXPLORATION..... 41-55
8.1 Old System Use Case..... 41
8.2 Activity Diagram 42
8.3 Full System Use Case..... 43
8.4 Full System Activity Diagram..... 44
8.5 Requirements Catalogue..... 49
8.6 Prioritized Requirements List (PRL)..... 50
8.7 Prototype of the new system..... 55
CHAPTER 9 – ENGINEERING 56-69
9.1 New System Modules 59
9.2 Use Case Diagram of the VOFS 60
9.3 Class Diagram of the VOFS..... 61
9.4 Entity Relationship Diagram..... 62
9.5 Sequence Diagram 63
9.6 Component Diagram of the VOFS..... 64
9.7 Deployment Diagram..... 65
9.8 System Interface Design..... 69
CHAPTER 10 – DEPLOYMENT..... 70-74

©Daffodil International University

10.1 Core Module Coding Sample..... 73

10.2 Possible Problem Breakdown..... 74

CHAPTER 11 – TESTING.....75-86

11.1 Test Plan Acceptance..... 76

11.2 Test Case.....76

11.3 Unit Testing.....79

11.4 Module Testing.....81

11.5 Integration Testing.....83

11.6 Security Testing.....84

11.7 Accessibility Testing..... 86

CHAPTER 12 – IMPLEMENTATION..... 87

12.1 Training.....87

12.2 Implementation Scheme..... 87

12.3 Scaling..... 87

12.4 Load Balancing..... 87

CHAPTER 13 – CRITICAL APPRAISAL AND EVALUATION..... 88-89

13.1 Objective could be met..... 88

13.2 Objective that totally don't meet of touched.....89

CHAPTER 14 LESSONS LEARNED..... 90-91

14.1 Pre-Project-Review-closing..... 90

14.2 What I have learnt..... 90

14.3 The problems I have faced..... 90

14.4 What Solution Occurred.....91

CHAPTER 15 – CONCLUSION..... 92-93

15.1 Summary of the Project..... 92

15.2 Goal of the project.....92

15.3 Success of the Project..... 92

15.4 What I have done in the documentation..... 93

©Daffodil International University

15.5 Value of the Project..... 93

15.6 My Experience..... 93

REFERENCES..... 94

LIST OF FIGURES

Figure 1: The overview of the site-1 ……....…… 8

Figure 2: The overview of the site-2 …….…… 9

Figure 3: The overview of the site-3 …….…… 9

Figure 4: Dynamic Systems Development Method (DSDM)…… …. 11

Figure 5: Structured System Analysis and Design Method (SSADM) or Waterfall Model…….13

Figure 6: Rapid Application development Model …….……. 14

Figure 7: Project file cycle Gantt chart …….…..… 20

Figure 8: Rich picture…….……..….36

Figure 9: The legends of rich picture …….……..39

Figure 10: Client-server model ……...….…41

Figure 11: Use case diagram ……42

Figure 12: User activity diagram of VOFS …….…… 43

Figure 13: Full Use case of the purpose of VOFS …….…..… 44

Figure 14: Admin activity diagram for VOFS …… 45

Figure 15: Manager activity diagram for VOFS …….…… 46

Figure 16: Staff activity diagram for VOFS …….……..…… 47

Figure 17: Customer panel prototype…….……..… 51

Figure 18: Admin panel prototype……..… 51

Figure 19: User login prototype ……...…52

Figure 20: Food category prototype ……53

Figure 21: Product page prototype ……54

Figure 22: Cart prototype …….……... 55

Figure 23: SSLCommerz prototype…….…….…...… 60

Figure 24: Use case diagram for VOFS …….…….…… 61

©Daffodil International University

Figure 25: Class diagram of VOFS …… 62
Figure 26: Entity Relationship Diagram of VOFS ……..63
Figure 27: Sequence Diagram of VOFS ……64
Figure 28: Component diagram of the VOFS …… 65
Figure 29: Deployment Diagram……66
Figure 30: Home page interface……67
Figure 31: Food category interface …… 68
Figure 32: Admin panel interface …… 68
Figure 33: Add product by admin interface ……69
Figure 34: Customer profile interface …… 70
Figure 35: Log in page ……71
Figure 36: Logout sample ……..71
Figure 37: Add to cart system ……72
Figure 38: Cart view system ……73
Figure 39: Payment system ……..... 77
Figure 40: Unit test case-1 results ……78
Figure 41: Unit test case-2 results …… 79
Figure 42: Unit test case-3 results ……80
Figure 43: Modular test-1 result ……..81
Figure 44: Modular test-2 result ……..82
Figure 45: User login integrate test ……83
Figure 46: User login integrate test results……84
Figure 46: User security test results ……85

©Daffodil International University

LIST OF TABLES

Table 1: Cost Estimation of the Website-based solution ……..…...…….…… 5

Table 2: Cost Estimation of the Tab-based solution ……...……5

Table 3: Cost Estimation of the Desktop-based solution ……..….…… 6

Table 4: Work Breakdown Structure of DSDM …….. 17

Table 5: Resource Allocation list …….……... 18

Table 6: List of the time boxes …….19

Table 7: Sample of testing against the time box …….. 21

Table 8: Sample Test case …….…….. 22

Table 9: User Acceptance Test plan …….…….……..….……23

Table 10: Risk identification Table …….……...…..24

Table 11: Risk Precaution …….….…….. 25

Table 12: Risk dealing steps……. 26

Table 13: Total cost Estimation for the project……... 33

Table 14: Earning estimation for the project ……...…….33

Table 15: Estimation Revenue on a five-year scale …….……34

Table 16: Requirement catalog for login registration …….…….…… 48

Table 17: Requirement catalog for choosing category and product ……..…….……..49

Table 18: Requirement catalog for Pre-booking ……..….……49

Table 19: Login module …….……. 56

Table 20: View food category and food module…….……. 56

Table 21: Wishlist module…….…….…….. 57

Table 22: Cart module…….…….…… 57

Table 23: Order module……..…….…….…… 57

Table 24: Report module…….……...…….…… 58

Table 25: Staff module …… 58

Table 26: Food Price Prediction module…….……..58

Table 27: Payment module ……..…….57

©Daffodil International University

Table 28: Unit test case-1 ……...……..… 77

Table 29: Unit test case-2 …… 78

Table 30: Unit test case-3 …….……79

Table 31: Modular test-1 …….….80

Table 32: Modular test-2 ……..……...……81

Table 33: Integration testing …….……82

Table 34: User security test …….… 84

Table 35: Customer controller accessibility test …….…….…….85

Table 36: Admin controller accessibility test-2 ……...…….……86

Table 37: User training …….…87

CHAPTER 1

Introduction

Today's world is full of technology. So, people want to be more comfortable and save time. That is why they are always looking for better services. The customers of Palal Group are using a manual system for purchasing food, they have no automated or online system to order the food products. In this technical era, the customers and the company (Palal Group) want to make their system online food ordering system. So, the Vegor Organic Food Supply (VOFS) is providing the facilities to make their food delivering and ordering easy.

1.2 Documents contents

This paper will include the provided chapters for documenting project work.

Chapter-1: Introduction

The presentation of the system in detail.

Chapter-2: Initial Phase

In this section, the detailed initial part of the system with its targeted goals, objectives, problem fields, alternative solving way, & project background.

Chapter-3: Methodology

This section discusses in detail the domain of problems, solutions, which method will be good for the project, etc.

Chapter-5: Planning

This chapter is about the planning to complete the project with work breakdown structure, resource allocation, time boxing, Gantt chart, test case, etc.

Chapter-6: Feasibility

This section discusses the feasible ways of the project with different platform cost-benefit analyses.

©Daffodil International University

Chapter-7: Foundation

In this section, the requirements gathering ways, problem solutions, and functional & non-functional requirements will be discussed.

Chapter-8: Exploration

This section contains the necessary catalogs, including some old systems and basic UML diagrams and prototypes of the new system.

Chapter-9: Engineering

This section defines the major modules, particularly with appropriate diagrams.

Chapter-10: Deployment

Here, some coding scripts and development problems are discussed including appropriate screenshots.

Chapter-11: Testing

Testing the whole system with functional & non-functional testing approaches.

Chapter-12: Implementation

Implementation methods, training models, and related issues are discussed here. Here are discussed the implementation methods, tanning, and implementation scheme.

Chapter-13: Critical Appraisal and Evaluation

Evaluation of the total work, requirements reach its goal or not, why not, etc. are discussed.

Chapter-14: Lessons Learned

The learnings and the facing problems-solution are explained in detail in this section.

Chapter-15: Conclusion

The goal of the project and gathered experiences are explained here.

©Daffodil International University

CHAPTER 2

Initial Study

2. 1 Project Proposal

Vegor Organic Food Supply is planning to develop a centralized online-based food ordering and delivery system for particular customers. The customer's order products by calling the staff, which makes them hassle as their wishes can be changed to order food products. So, the VOFS is providing them with the easiest and most simple way to order food as their wish. They don't need to call and waste money on-call services. They will just open the website and order foods as their wish. Moreover, they can request extra-large quantity foods if they need for any occasion as the Vegor Agri Complex Ltd. has the firm to grow organic products.

2. 2 Background Study

I am always concerned about food and human health. The people of Dhaka city cannot find fresh and healthy food, so they eat adulterate food which makes their health corrupted, those adulterated food makes very serious diseases like cancer. In our world, over 660 million people get affected then sick from eating adulterated & unhealthy food where die 400 thousand people every day. [3] So, I always to be do something about the issue. My target was working to deliver fresh food to the people of Dhaka. Then, I find Vegor Agri Complex Ltd. where they have farms, grow organic and fresh products, and deliver their products to some elite customers in a manual way. So, I decided to make an online-based food delivery system for them as well as I am fulfilling my wish too.

Although there are many online shopping websites in our country to order products, the foods they provide are not fresh and organic, sometimes they got rotten food, and sometimes they didn't get products timely. But the customers of VOFS are free from those occurrences. Even they will request food in a high quantity if they need it. They will simply login into the system through provided username and password, and then they will available to choose categories and products. The system is very simple, easy and user-friendly, even a new user will understand what to do on the website

.

2. 3 Problem Areas

Description of the Proposed System

The Vegor Organic Food Supply (VOFS) system is a web-based application that allows particular customers to order food online from Vegor Agri Complex Ltd. Also, the staff will allow managing customers, products, orders, etc.

Feasibility Study

- **Operational Feasibility**

  The system is for some elite customers so the customers need to log in first with their username and password. After that, the customer will be able to view categories and details of grocery foods. Then, they have to pre-order that food. After ordering, they will receive their order in 2-3 days, because those grocery products need to pluck or collect from the firm. After that, the customers need to pay online first. They will also get notified about their delivery time and date. A delivery pick-up will arrive in Dhaka with orders with a fixed category of food like vegetables and fruits, meat and fish, and agro products (rice, corn, etc.)

- **Technical Feasibility**

  The technical feasibility test condemns that **Vegor Organic Food Supply (VOFS)** users can complete their activities more easily than the previous manual activities. I am going to use XAMMP server, SQLlite Database Server, Django framework, and for development and documentation, I have used Microsoft Office. This project is designed & developed by the lowest & most trendy web technology so that the system is adaptable to any web browser with just an internet connection. The project has an approach-control sequence to manage and protect access to system data. The application is web-based and platform-independent and its selected development platforms & resources are budgeted and costly so technically it is possible to do.

- **Economic Feasibility**

  We know, many types of options for developing any proposed system. I am including web-based applications, tab applications, and desktop applications.

- **Website-Based Application:** Complete applications & data are supplied on a remote server and can be penetrated through the Internet anytime, anywhere, without the trouble of installation.

Equipment	Cost per unit	Cost
With extranet network VPN	2,000 BDT only per month	2,000 BDT
Desktop pc (core i5 7th gen, 4.8 GHz processor, 16 GB DDR4 RAM, 512GB SSD)	1,32456 BDT total	1,32456 BDT total
Website, Files, and Email servers	10,500 BDT only per month	10,500 BDT
	Total cost =	144,956 BDT

Table 1: Cost Estimation of the Website-based solution

- **Tab-Based Application:** Each tab containing the system can browse to stay on the device which shouldn't be a bit of a hassle.

Equipment	Cost per unit	Cost
Web, email, file Servers.	10,500 BDT per month	10,500 BDT
Desktop pc (core i5 7th gen, 4.8 GHz processor, 16 GB DDR4 RAM, 512GB SSD)	1,32456 BDT total	1,32456 BDT
	Total cost =	144,956 BDT

Table 2: Cost Estimation of the Tab-based solution

- **Desktop-Based Application:** Every desktop computer requires web browsing and the data is hosted on a remote server. No need to install it.

Equipment	Cost per unit	Cost
Web, email, file Servers.	5,000 BDT per month	5,000 BDT
Desktop pc (core i5 7th gen, 4.8 GHz processor, 16 GB DDR4 RAM, 512GB SSD)	1,32456 BDT total	1,32456 BDT
	Total cost =	137,456 BDT

Table 3: Cost Estimation of the Desktop-based solution

A domain name and hosting service are required to make the system live. I will create a website application for the customers of Vegor Agri Complex Ltd. So that the customers can easily access it through the mobile, tab, laptop, etc.

2. 4 Problem Solution

The customers of Vegor Agro Complex Ltd. are ordering products manually now. So, they are unable to choose products by viewing details, choosing a quantity, online payment, online invoice, special request for food, etc. Even the company has to keep customers' data manually, so this is so time lengthy process if they need to locate any customer. Manually, they keep data that is not consistence with the online era. So, my system is going to give the best solution to the problems.

CHAPTER 3

Literature Review

3.1 Discussion of the problem domain

Vegor Organic Food Supply (VOFS) focuses on food ordering, the transaction process system, and the management information system. As we know, the Vegor Agri Complex Ltd. is providing manual services now, so they are unable to connect online, and make their system run automatically. Some of the main problems are discussed below:

1. All information keeps in a manual way
2. Customers can not view product details
3. Customers cannot change carts as their wish
4. Customers can not rate products and view reviews
5. Customers pay money manually
6. Customers can not request food products for any occasion
7. Customers can not view order history
8. Customers do not get any invoice after payment
9. The company is unable to track customer information in an automated way

3.2 Discussion of the Problem Solution

Vegor Organic Food Supply is a system where customers can buy food products online as their wish. Now a day's technology is everywhere. Some solutions should favor the issue discussed below:

- **Automated system:**
  Here, information about customers, orders, transactions, etc. is kept in the manual system in Vegor Agri Complex Ltd. In the VOFS system, all the information is stored in the database, so there is no possible way to lose data. Also, data will be found easily if needed at any time.

- **Easy interface:**

  Currently, customers can not view product details properly, they even not be alerted if any product arrives newly. So, VOFS is providing a simple ordering system where a new user can purchase the product.

- **Online payment:**

  Customers are paying money after purchases manually. Only cash on delivery is available now. But VOFS is providing the online payment facilities through SSLCommerz.

3.3 Comparison among the leading solutions:

The environment is growing day by day and new automation platforms are coming, updating & making the easiest life. Nowadays the world is very tiny like a suburb because of the use of the internet. Many solutions to the same problem in diverse ways. Although **Vegor Organic Food Supply (VOFS)** is a solution for a particular company, some ongoing systems running now. Here in this section, I will analyze the comparison of the well-known & famous current websites. And this lists weaknesses, and strengths.

- https://www.shwapno.com/ (Local)
- https://chaldal.com/ (Local)
- https://www.walmart.com/ (International)

Figure of Shwapno

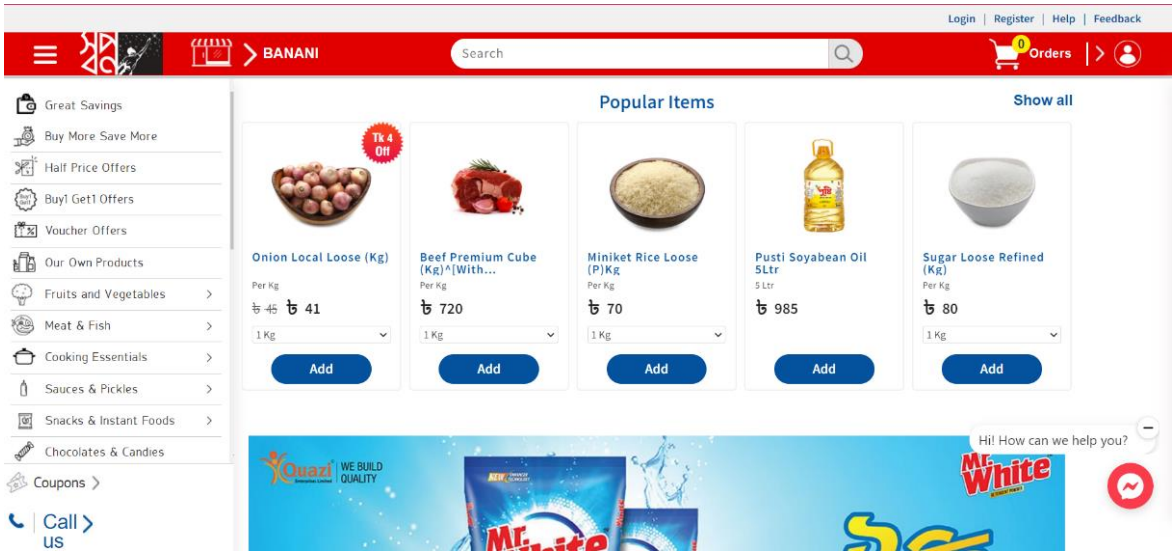


Figure 1: The overview of the site-1

Figure of Chaldal

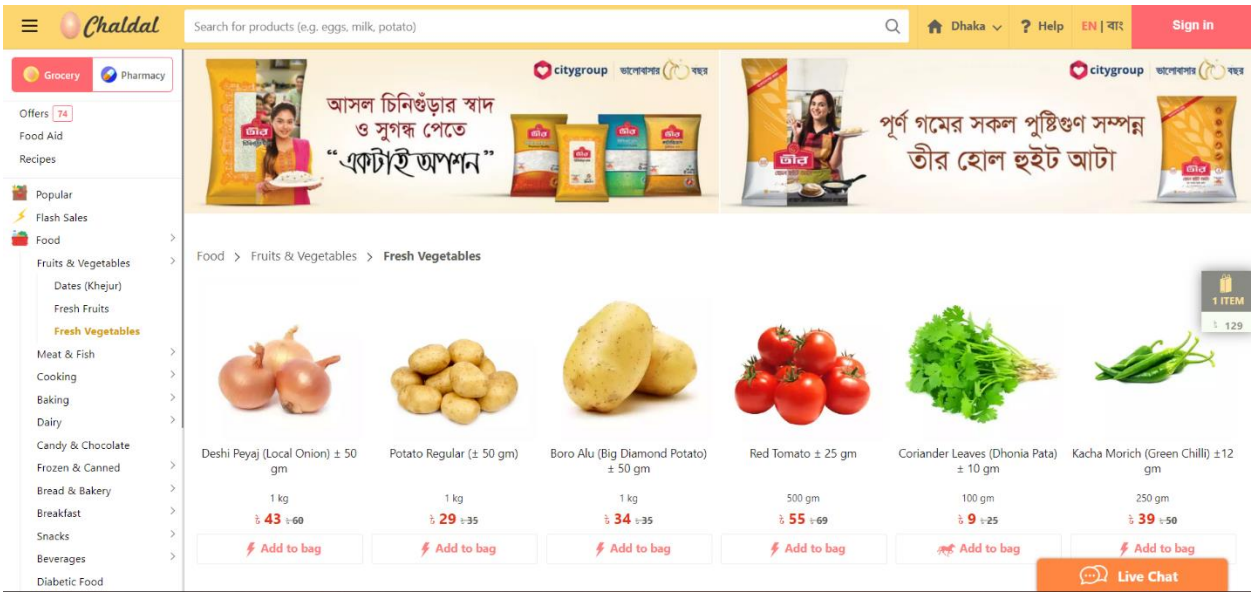


Figure 2: The overview of the site-2

Figure of Walmart

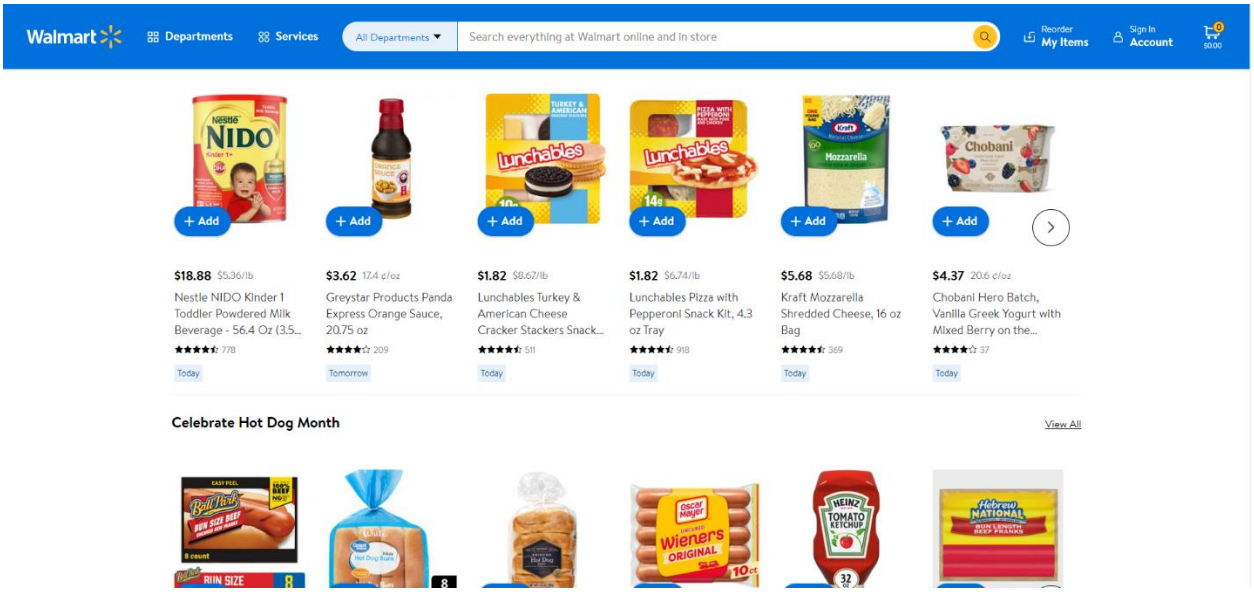


Figure 3: The overview of the site-3

Best Features

- ✓ Very well organized and wonderful design
- ✓ Provide many choices options to users
- ✓ Have an organized program for ordering products

Limitations

- ✓ No special request options
- ✓ Not showing automation video of products
- ✓ Too many ads

3.4 Recommended Approach

We can see that there is no need for existing website features it clarifies the functionality and a new system needs to be considered for consideration. The new system aspects listed here should be provided-

- ➢ The system has a user-friendly interface and it's easy to use.
- ➢ Restriction for accessing user information.
- ➢ Auto notifying system
- ➢ Identity verification and approval process.

CHAPTER 4

Methodology

All methods have their strengths and weaknesses and they depart for their particular reasons. The method chosen for development depends often on the success rate of development. In this chapter, I will discuss in detail the selected method and the proper reason why I choose this method.

4.1 What to use

We know, that software development methods play an important role. Many types of software development methods like waterfall model, prototype model, agile software development, rapid application development, dynamic system development model, spiral model, joint application development, etc. are used for software development purposes. Now, I am going to describe 3 methods with their merits and demerits.

Dynamic Systems Development Method (DSDM)

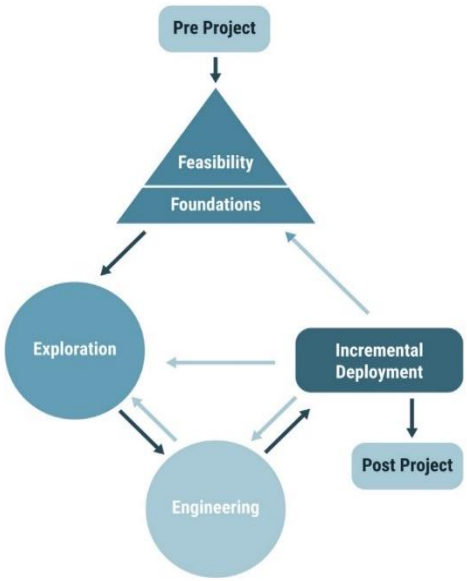


Figure 4: Dynamic Systems Development Method (DSDM)

The Dynamic Systems Development Method (DSDM) is a method that is based on the Rapid Application Development Method. The major preference of this approach is concurrent user engagement through the whole iterative & incremental approach. Always I pursue DSDM to create a quality project with customer feedback.

Advantages of Dynamic Systems Development Method (DSDM)

- The DSDM model prioritizes business, ensuring that the business value of any project they deliver is important
- We can provide common product features at a speedy pace
- Developers can freely reach their clients
- The system is easy to stay on a fixed budget
- Development & progress happen fast

Disadvantages of Dynamic Systems Development Method (DSDM)

- Sometimes there are big management overheads
- Implementation can be costly as it may be impractical for a small company
- DSDM does not boost developer's creativity
- Systems follow a certain standard, even if other more practical options are available

Structured System Analysis and Design Method (SSADM) or Waterfall Model

One of the most familiar models for software development is the methods of software development. This approach is treated to be consistent with a simple way of software development. This model method follows a continuous progression which cites the phrase that any phase will begin working only when the previous stage is completed. This development process does not go to the previous stage if any change is required to meet the requirements.

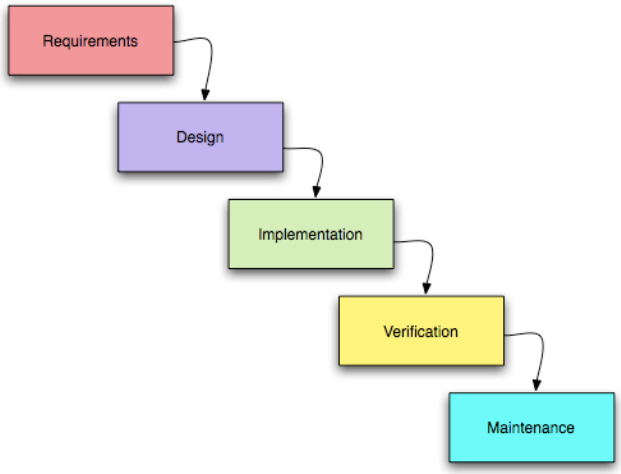


Figure 5: Structured System Analysis and Design Method (SSADM) or Waterfall Model

Advantages of Structured System Analysis & Design Method (SSADM) or Waterfall Model

- Every stage needs to be finished before developing the next phase.
- Applicable for short projects that all requirements are clear.
- Before completing each stage, their quality-assurance test (verification-validation) should be done
- At each stage of the software development cycle, expanded documentation is done.
- With minimal client interference, the project is dependent on the project team
- During the process, any adjustment to the system is made.

Disadvantages of Structured System Analysis and Design Method (SSADM) or Waterfall Model

- During this phase, faults can only be solved
- This method is recommended for complex projects where requirements change intermittently

- In the developmental process, the duration of the test comes quite late
- For developers & testers, documentation takes a lot of time.
- In the ongoing development phase, helpful feedback from clients cannot be included
- Slight changes or bugs in the finished software can cause many obstacles

Rapid Application development

Rapid Application Development or RAD is an adequate method that provides faster expansion and quality work than other software development methods. This method is designed in a way that it gives credit for the peak benefits of software development in a simple way. The key to this approach is to accelerate the entire software.

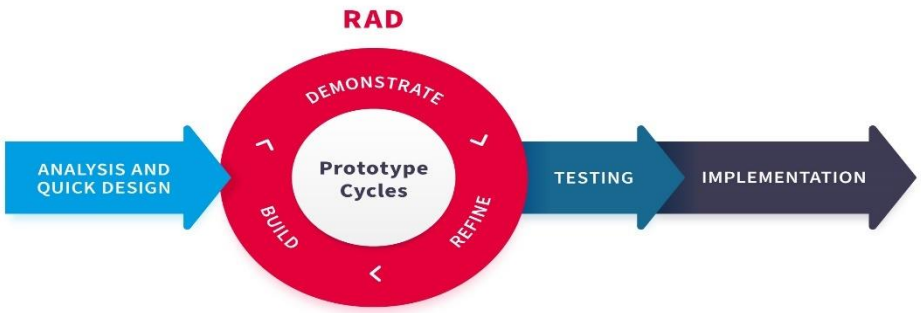


Figure 6: Rapid Application Development Model

Advantages of Rapid Application development

- Malleable and versatile to change
- This is advantageous when we need to lower the overall project risk
- Distributable transfers are easy as scripts, powerful abstraction, and transitional code are used
- Manual coding has declined as a result of code generators and code reuse
- There is less chance of error considering the prototyping in a variety

Disadvantages of Rapid Application development

- It does not apply to small projects
- All applications are not appropriate for RAD
- This is not appropriate when the technical risk is high
- If the developers are not pledged to submit software timely, RAD projects can fall
- Functionalities take in lower due to time boxing, with functionalities being pushed to the next variation to complete a discharge in a shorter time

Choosing Methodology

To consider the above criteria, I have chosen the DSDM method for the Vegor Organic Food Supply System project. DSDM will be the perfect method and easily applicable option for Vegor Organic Food Supply (VOFS) project. The reason for choosing this method is that DSDM always ensures product & project quality in the period and it is malleable for use in any company. During the development process, clients may switch or shift their requirements if they feel that any changes are required for their project.

4.2 Why to Use

The project needs to pursue a procedure to strongly meet the provided time frame & budget requirements. By following its authorized predetermined phases, a method helps to achieve the major objectives of the project. Choosing the perfect one is hard though. The methodology of such an academic project will go a long way in completing the project and providing output.

4.3 Sections of Methodology

There are some sections of the DSDM system that the developer team needs to pursue. There are:

- **Pre-project Stage**

  The initial project concept includes terms, budget, and basic requirements in this section, the

- **Feasibility Study Stage**

  The technical, economic & operational feasibility of the planned project is defined in this stage measures.

- **Requirement Gathering Phase**

  Here, functional & nonfunctional requirements of the system are gathered by applying different techniques.

- **Requirement Analysis and Prioritization Phase**

  Collected requirements are tested and arranged using a prioritization technique like Moscow

- **Search & Engineering Stage**

  In this stage, the core iterative section of the method was used to analyze the requirements & make the iterative solution.

- **Review Stage**

  Establish deliverables are evaluated with users and returned to the earlier episode if the change is required.

4.4 Implementation Plans

The project is in the final stages of work, where advanced applications are discharged for use. Making the new system accessible for use if there are any issues associated with it detected & resolved. Release criteria, configuration, & planning are decided in this section. Then if everything is OK the recent system is released.

CHAPTER 5

Planning

5.1 Project Plan

It shows the way to plan the culmination of the project. The project is broken down into several individual parts and following the plan, all the work will be done within a specified time which is provided in this plan.

5.1.1 Work Breakdown Structure

It proposes that the system can be divided into smaller tasks & the project can be done promptly and more simply and efficiently. This pattern gives us an estimate of time and work. Completing the project could be more complicated without this framework. Thus, in WBC, the proposed system is divided into sections and sub-sections by a chart below:

Serial	Task Name	Duration	Start Date	End Date
1	User Authentication System	4 Days	Mon 24/01/2022	Thu 27/01/2022
2	Food Category Manipulation	8 Days	Fri 28/01/2022	Fri 04/02/2022
3	Food Availability and Stock Management	5 Days	Sat 05/02/2022	Wed 09/02/2022
4	Cart Manipulation	8 Days	Thu 10/02/2022	Thu 17/02/2022
5	Pre-Ordering System	10 Days	Sun 20/02/2022	Tue 01/03/2022
6	Delivery Status Notification System	5 Days	Sat 05/03/2022	Wed 09/03/2022
7	Wishlist Manipulation	4 Days	Thu 10/03/2022	Sun 13/03/2022
8	Online Payment System	20 Days	Sun 20/03/2022	Fri 08/04/2022
9	Food-product Price Prediction System Management	15 Days	Sun 10/04/2022	Sun 24/04/2022
10	Report Observing System	5 Days	Tue 26/04/2022	Sat 30/04/2022
11	Validity and verification in different user aspects (staff, admin, manager)	5 Days	Mon 02/05/2022	Fri 06/05/2022
12	Discounts Management	2 Day	Sun 08/05/2022	Mon 09/05/2022
13	Food Quality Rating and Review System	2 Day	Sun 15/05/2022	Mon 16/05/2022
14	Testing	5 Day	Fri 20/05/2022	Tue 24/05/2022
15	Totals	98 Days		

Table 4: Work Breakdown Structure of VOFS

5.1.2 Resource Allocation

All resources are apportioned and maintained for the culmination of the proposed project in a formed manner. In project planning, resource allotment is the most imperative part. Since this is an academic system, I have to play particular roles at particular times because I am responsible for the whole system. The allocation of resources for the VOFS project to meet the pre-determined task delivery deadline is as follows:

Time box	Task	Resource
TB 1	User Authentication System	User, Developer
TB 2	Food Category Manipulation	Developer
TB 3	Food Availability and Stock Management	Developer
TB 4	Cart Manipulation	Developer, User
TB 5	Pre-Ordering System	Developer, User
TB 6	Delivery Status Notification System	Developer, User
TB 7	Wishlist Manipulation	Developer, User
TB 8	Online Payment System	Developer, User
TB 9	Food-product Price Prediction System Management	Developer
TB 10	Report Observing System	Developer
TB 11	Validity and verification in different user aspect (staff, admin, manager)	Developer, User
TB 12	Discounts Management	Developer
TB 13	Food Quality Rating and Review System	Developer, User
TB 14	Testing	Tester

Table 5: Resource Allocation list

5.1.3 Time Boxing

This is another important part of DSDM project planning, as DSDM divides tasks into time boxes to meet targets ahead of schedule. In this section, all tasks are divided into several time boxes with a duration. These tasks must be completed within the time allotted for the repetitive procedure.

SL	Task	Duration	Resource
1	User Authentication System	4	User, Developer
2	Food Category Manipulation	8	Developer
3	Food Availability and Stock Management	5	Developer
4	Cart Manipulation	8	Developer, User
5	Pre-Ordering System	10	Developer, User
6	Delivery Status Notification System	5	Developer, User
7	Wishlist Manipulation	4	Developer, User
8	Online Payment System	20	Developer, User
9	Food-product Price Prediction System Management	15	Developer
10	Report Observing System	5	Developer
11	Validity and verification in different user aspect (staff, admin, manager)	5	Developer, User
12	Discounts Management	2	Developer
13	Food Quality Rating and Review System	2	Developer, User
14	Testing	5	Tester

Table 6: Time-box table

5.1.4 Gantt Chart

The Gantt chart is a distinct depiction of the project activity timetable. It shows a progress bar duration from beginning to the last date rather than day. Gantt chart for Vegor Organic Food Supply attached below:

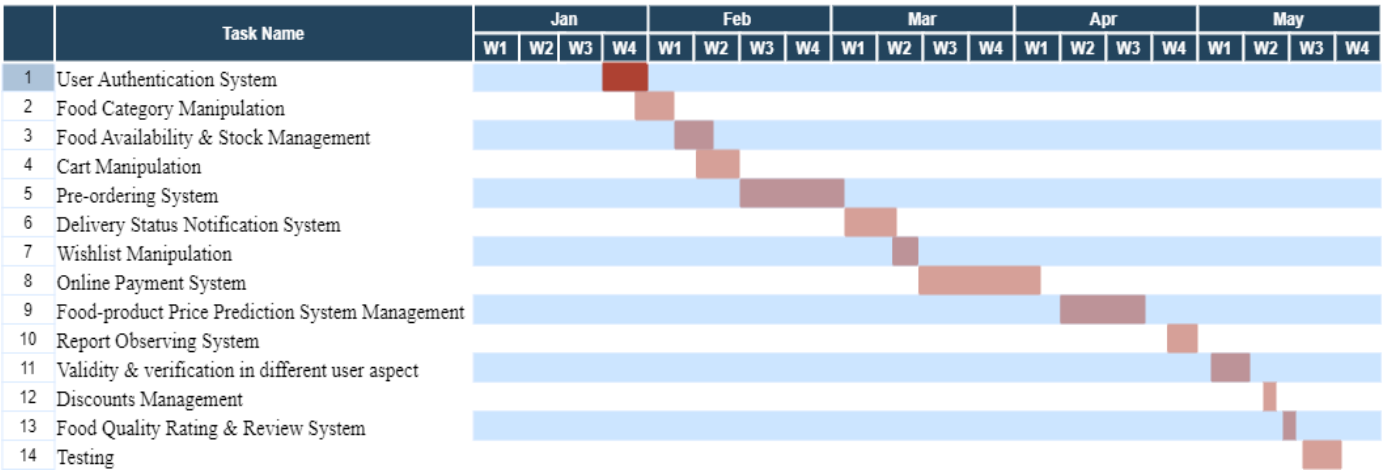


Figure 7: Project file cycle Gantt chart

5.2 Test Plan

The test is planned based on the difference between the given input and the desired output of the individual system. Software development has been tested during the verification & validation process

5.2.1 Testing against the time boxing

The time-box process is based on estimating a specific and maximum single time for an individual department. Test against time box:

User name	Example	Role	Example
Time box ID			
Time box content			

Test Type	Test steps	Expected result	Actual Result	Comment
Unit test				
Integration Test				
System Test				
Acceptance Test				
Security Test				
Usability Test				
Reliability rest				

Table 7: Sample of testing against the time box

5.2.2 Required Test

Some testing sectors exist but functional & non-functional tests are the 2 main types of a system:

Functional Testing:

- **Unit testing:** It is only usable on a short part of the software design. In this phase, I divided units & test those by creating them individually or as a group of units. This is accomplished by the developer concerned with using common input and monitoring the predicted output.

- **Integration testing:** This unit works with tested elements and creates a programmatic framework that is endorsed by design. This segment combines groups of elements to create the desired output based on the input provided by the programmer. Black box & white box were included here experimental.

- **System Testing:** Including Windows and Linux, it works for some operating systems. It only works with the black-box testing technique. It receives the required input and displays the desired output with internal work.

- **Acceptance Testing:** Users interact directly with this test. This is done after the above test is over. It acts on user expectations and vindication as well as meeting business needs.

Non-Functional Testing:

- **Security Testing:** This testing is done externally & through security testing. An application includes a web application. It verifies illegal users or entry and protects against external & internal attacks, including SQL injection.

- **Usability Testing:** This provides that the system makes them user-friendly and those users can easily communicate & access services. This kind of review and validation has happened in this usability test.

- **Reliability testing:** The users can collaborate precisely with this test. This assures the authenticity of the operation of the project. It examines & confirms a variety of system failures.

5.2.3 Test Case

A test case consists of test case no, type, description, & steps.

Test case no			
Test type			
Test Description			
Test Steps	Expected Result	Actual Result	Comment

Table 8: A sample Test case

5.2.4 User Acceptance Test Plan

This is the last step in the user acceptance test segment. It specifies and tests the effectiveness of the test by the user concerned.

Test case no			
Test type			
Test Description			
Precondition of testing			
Users Name			
Act as			
Test Steps	Expected Result	Actual Result	Comment

Table 9: User Acceptance Test plan

5.3 Risk Management

This is the process of classifying, evaluating, & responding to any risk which can occur in an ongoing project in the life cycle of the project development. To track remaining projects and meet their goals and expected outcomes. The risks of a project may include project timetable, budget as well as project effectiveness. Different types of risk management involved with **Risk assessment, Risk identification, Steps taken for possible risks & Risk precaution.**

5.3.1 Risk Identification & Assessment

The most important section & process of risk management planning is Risk identification. The initial step in risk identification is to identify and analyze the risks that may occur and affect the project, as well as to identify & raise its features along with the features of the document. According to this system, some risk identification criteria will be explained which are provided below:

- Risk-track with potential risk identification and documentation
- The identified risk details include the actual cause
- Implications including future outcomes and risk factors

The details of potential risk identification and record of the risks of this system are provided below:

Type of Risks	Causes	Impact & consequence
Scheduling	Many types of causes to appear scheduling risks included: ▪ Inadvertently late reason of Corona Pandemic situation ▪ Individual segments fail with data losses of Users and admins.	▪ Late submission of project
Database Error and Destroy	In this project two panels here, admin and user. So sometimes it's tough to manage	The user could not order products properly
Unauthorized Access	A person trying to access this system without permission. This can make it difficult to include risks ▪ Users may order with cash on delivery without an invalid address	Confidential information can be hacked as well as changed
Technical problem ❖ **Hardware**	▪ Lacking required velocity ▪ Lacking required conduct ▪ Lacking required structure	▪ Organizing may failure
Project Backup	▪ Lacking the system backup on GitHub or a similar backup system	▪ Actions on expenditure ▪ Loses the total system
Network Failure	▪ Lack of approach to some integrated API and Internet connection required for email management	

Table 10: Risk identification Table

5.3.2 Risk Precaution

The risk action is planned after the completion of risk identification and risk assessment. There are some steps you can take to begin the process of preparation for mediation

- Occurred and to prevent potential risks
- Occur and reduce potential risks
- To address those that have raised potential risks

Types of risks	Action	Action Taken By	Action Required
Scheduling	For making definite time evaluation by using Gantt chart etc. technology use: Hive, Forecast, etc.	Involved developer	Before starting project development
Database Error	▪ To make the necessary normalization. ▪ To create the perfect necessary relationship at the box	Involved - developer	At the time of Data management
Unauthorized Access	▪ To perform on code while managing code smelling ▪ To launch multi authentication system for this project .	Involved - developer	When setting the authentication segment
Technical problem ❖ **Hardware**			
Project Backup	▪ Lacking system backup on GitHub or a similar backup system	Involved developer	Update frequently
Network Failure	▪ Lack of access to some integrated API and Internet connection required for email management	Involved developer	At the time of the project development

Table 11: Risk Precaution

5.3.3 Steps Taken for Possible Risk

After analyzing and checking potential risks, the responsible person takes the necessary stages. These are

Types of risks	Description	Likelihood	Impact	Steps Taken
Scheduling	The most common risks are the probability of missing project schedules & deadlines.	Rare	High	During this system, time boxing and Gantt chart created charts & work breakdown structures.
Database Error	Probability of redundant entry, invalid relationship.	Likely	High	Normal form normalization and entity-relationship images have been used and data dictionaries have been created.
Unauthorized Access	Illegal access to the system.	Unlikely	High	Implement a proper authentication system and access authorization.
Technical Problem ❖ **Hardware**	Hardware crashes can cause unresponsiveness.	Frequent	Medium	Backup and formal maintenance and managing have been ensured.
Network Failure	Internet connection failure.	Frequent	Low	The backup server is configured and high-speed bandwidth is appropriated.

Table 12: Risk dealing steps

5.4 Change Management

5.4.1 Factors that might cause change

Different types of changes are needed for this system. Some are included: functional requirements, an adjustment in the scheduling portion, different modules of the user interface, security handling portions, rating and review portions of the patient and donner portions.

5.4.2 DSDM Welcome Change

Sometimes the life-cycle of a project development should be changed based on the needs of the project. For this reason, we should pursue some procedures that take up different types of protests based on the required system. So here DSDM is one of the acceptable, stable, accurate executable methods. In this system, we pursue the procedure to manage & perform modifications during development. It works in several approaches:

- Establishing a system for tracking any module updates Method
- Taking reviews from some users when changing categories, such as customer details, and email automation changes.
- Performing after the computerization of changes
- Assuring goals of the security
- Assuring the accuracy as well as extensibility

5.4.3 Considering Business Priority

The modifications are treated to meet the desired goals & objectives with the organization's objectives. That's why we need to prioritize the effectiveness of the change as well as the requirements that will result in the strong benefits expected from the users involved. Prioritization is:

Area	Level of Priority
Scheduling segments	6
Functional requirements	5
User interface	7
Rating and review segments	6
Security handling segments	5

5.4.4 Change Workshop

As well as concerning the profiles of the team members, the user interaction will be firmly established. Developers may be involved in making necessary changes and additions do a workshop by creating questionnaires on what will be needed to make changes and additions to this system.

5.4.5 Changes that are allowed

Modifications need to be adapted based on the priority level of the change Modifications need to be made based on several resources, costs, schedules, quality, and risk During the development of a certain part of this system, we need to allow changes according to our priorities.

5.4.6 Key Decision Taker of Change

The key decision-makers of the change will be taken by various experts in their area. This project includes individual decision-makers for change here: Requirements Analyst, Developer & Tester.

The analyst can change the functional requirements as needed and the developer may decide to change the relevant code as needed.

5.5 Quality Management

The management of quality is maintaining the standard that the user is granted to accept during the system's inception. This management depends on different aspects including-

- To track customers, patients
- Conversation among team members and users
- Endless modifications in requirement

5.5.1 Rules Applied to Maintain Quality

The guideline for maintaining standards is implemented from system to system and from user to user. Some rules need to be followed while maintaining quality including quality control and quality assurance.

5.5.2 DSDM Standard Quality Measures

Solution quality: The solution quality has been assured which is based on desired business needs and customer expectations by recording users' requirements. To complete this procedure 2 types of prioritizations have been applied. One is Moscow & another one is Time Boxing.

These two methods have been finished with managing the time.

Process quality: The process quality has been assured which is based on the focused company. 2 types of methods have been applied to finish this process quality. One is DSDM & another is CMMI.

5.5.3 Quality Plan and Measuring Meter

Different types of quality plans have been managed in this phase including

- Execution of scheduling & resource allocation
- After the completion of modules, testing has been an execution in all portion
- For more proceeding execution, when the total modifications have been executed then need to record and reported all the changes completely.

CHAPTER 6

Feasibility

6.1 All possible types of Feasibility

Operational Feasibility:

The system is for some elite customers so the customers need to log in first with their username and password. After that, the customer will be able to view categories and details of grocery foods. Then, they have to pre-order those food. After ordering, they will receive their order in 2-3 days, because those grocery products need to pluck or collect from the firm. After that, the customers need to pay online first. They will also get notified about their delivery time and date. A delivery pick-up will arrive in Dhaka with orders with a fixed category of food like vegetables and fruits, meat and fish, and agro products (rice, corn, etc.)

Technical Feasibility:

The technical feasibility test condemns that **Vegor Organic Food Supply (VOFS)** users can maintain their actions more easily than the earlier manual system. This system is composed & expanded by the lowest & most famous web technology so that the project is suitable for any web browser with a lower internet connection. This project has an entry control sequence to manage a secure approach to the data of the system. The web-based application is platform-independent and its selected development platforms and resources are cheap and costly so it is technically possible. Therefore, apparently in this system perspective, the technical aspects are provided down:

Hardware:

- ASUS Laptop (i3 10th gen)
- Router (Xiaomi)

Software:

- PyCharm
- Microsoft office
- Google chrome
- Windows 11
- XAMPP

Database:

- SQLite

Technology:

- HTML
- CSS
- JavaScript
- Bootstrap

Server Side:

- Python
- Django

Economic Feasibility:

Several substitutes to create the proposed system include desktop applications, web-based applications, etc.

- Web-based application costs are given below:

Equipment	Cost per unit	Cost
Extranet network with VPN	2000 BDT per month	2000 BDT
Desktop pc (core i5 7th gen, 4.8 GHz processor, 16 GB DDR4 RAM, 512GB SSD)	132456 BDT	132456 BDT
Web, File, and Email servers	10500 BDT per month	10500 BDT
Total:		144,956 BDT

- Desktop Application cost:

Equipment	Cost per unit	Cost
Web, email, file Servers.	5,000 BDT per month	5,000 BDT
Desktop pc (core i5 7th gen, 4.8 GHz processor, 16 GB DDR4 RAM, 512GB SSD)	1,32456 BDT total	1,32456 BDT
Total:		137,456 BDT

To make this system live, a domain & hosting facility is required.

Market Research Analysis Based on the Feasibility Factors:

There is an online food ordering system named Shwapno which is currently available in our country for all people. People can order different kinds of products through that system. My system is similar to the following system but it is not for all people of our country, it is for only some elite customers who are already engaged with the Palal group manually.

6.2 Cost-Benefit Analysis

The cost-benefit analysis focuses mainly on estimating earnings and expenses. The total expense & income are measured to get prosperity. Cost-benefit analysis for this system is given below:

Total Cost:

SL no	Equipment	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Web-based application cost	1,32456 BDT	-	-	-	-	144,956 BDT
2	Desktop Application cost	137,456 BDT	-	-	-	-	137,456 BDT
3	Domain and hosting Cost	20000 BDT	20000 BDT	20000 BDT	20000 BDT	20000 BDT	100000BDT
4	Employee Expenses	40000 BDT	40000 BDT	40000 BDT	40000 BDT	40000 BDT	200000BDT
5	Other costs	20000 BDT	20000 BDT	20000 BDT	20000 BDT	20000 BDT	100000BDT
6	Total Cost	310000 BDT	80000 BDT	80000 BDT	80000 BDT	80000 BDT	630000 BDT

Table 13: The total cost Estimation for the system

Total Earn:

SL No	Earn Sector	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Govt. tax 1	150000 BDT	180000 BDT	210000 BDT	240000 BDT	270000 BDT	1050000 BDT
2	Govt. Tax 2	200000 BDT	10000 BDT	240000 BDT	260000 BDT	300000 BDT	1210000 BDT
	Total	350000 BDT	390000 BDT	450000 BDT	500000 BDT	570000 BDT	2260000 BDT

Table 14: The earning estimation for the system

Total Revenue:

SL No	Sectors	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Total Earning	350000 BDT	390000 BDT	450000 BDT	500000 BDT	570000 BDT	2260000 BDT
2	Total Equipment Cost	402000 BDT	80000 BDT	80000 BDT	80000 BDT	80000 BDT	7220000 BDT
	Total Revenue	52000 BDT	310000 BDT	370000 BDT	420000 BDT	490000 BDT	1538000 BDT

Table 15: The estimation of revenue on a 5-year scale

Simply, the promotion of this project is the government. By collecting VAT and taxes for different purposes, a huge amount of money will be earned in a year. Revenue will increase year by year. Therefore, this system will be beneficial for the government & the people too.

6.3 Is DSDM Good or Bad for this Project

It is a scholastic project & needs to conclude in a fixed time with all major features. The system may require to get changes during its development, that's why it should develop constantly. VOFS assures continuous development action and appoints tough rules & restrictions. Also, it assures the quality of the activities within the time limit. Furthermore, it assures user engagement all over the system's lifetime. So, I can say DSDM is the perfect technique to pursue Vegor Organic Food Supply.

CHAPTER 7

Foundation

7.1 The problem area identification

The great improvement for any project is finding out the problem & that phase can be implemented by the customers because generally, they will use the system. So, some of the issues collected from the user:

7.1.1 Interview

This interview is a perfect choice whether we want to collect data or find out the real issue or the real goal. The real solution can be realized by finding the problem. For the proposed project, some questions from the user are fixed to be interviewed which are given down:

1. **For User:**
    - Can get notification
    - Can pay money online

2. **For Admin**
    - See the user information
    - Fix any kind of problem of user

7.1.2 Observation

In collecting user specifications and knowing the real issue, observations are also very useful. Essentially, observation methods are used to get to the workplace & find solutions for the users to accept. Thus, the main points of observation are:

- View product details
- Users can request especially for food
- Admin can see the user data

7.1.3 Questionnaires

Questionnaires are the best strategy for effectively gathering data and asking questions to customers and other people. Some queries are fixed for the customer and can be short queries or MCQ. Thus, the question is given below:

	Question for problem identification
Question - 1	Why do people need an online food delivery system?
Answer	
Question - 2	Tell us your thought about Vegor Organic Food Supply
Answer	
Question -3	How do you feel to order food online?
Question -4	What should be the special requesting system for food?
Answer	

7.2 Rich Picture

The Rick image is a top view or bird's eye view of the project's user activity. Further, envisions clashing issues, collaboration, & business procedures.

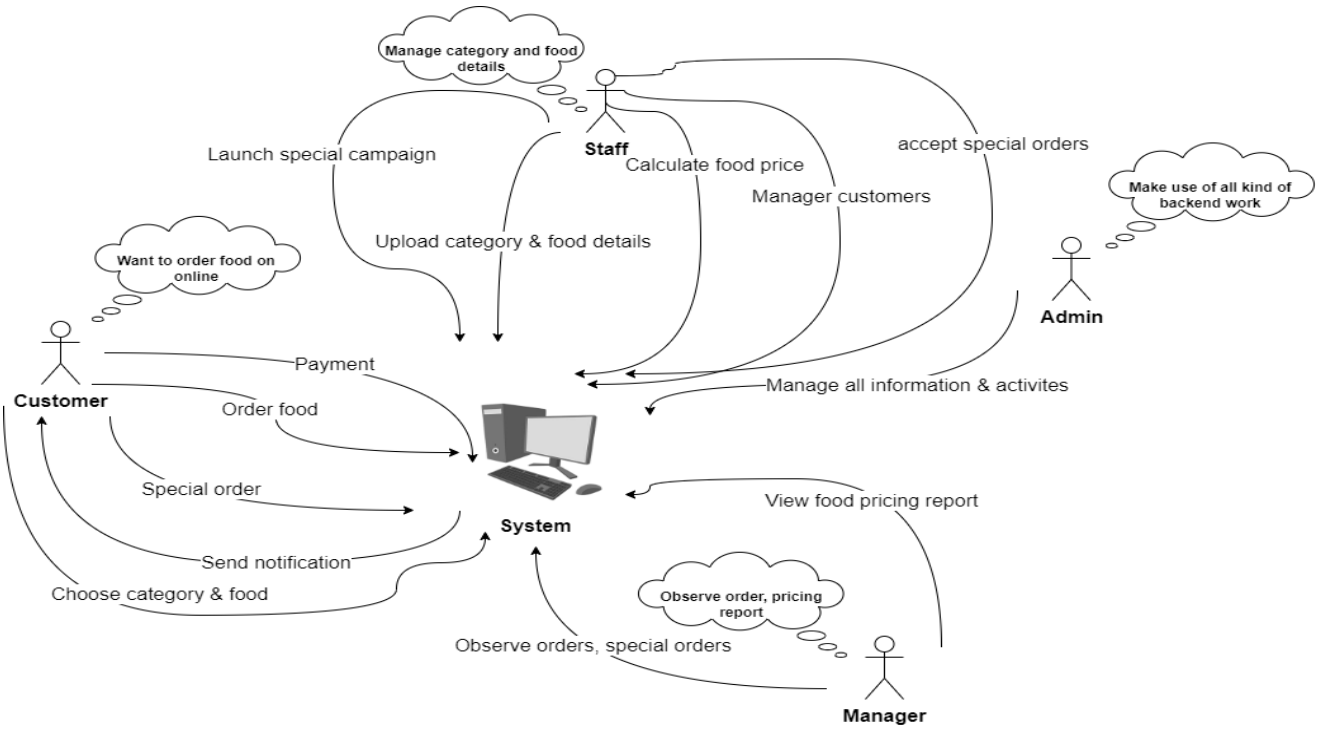


Figure 8: Rich picture

Key Actors

There are many actors in VOFS. The most important actors are

- Admin
- Customer
- Staff
- Manager

Short description of rich picture:

This figure displays that the administrator can control & maintain each process and manage clashes. Also. it shows the difference between the client & the admin. It also shows that many problems can occur.

7.3 Specific Problem Area Identification

Several issues are detected in the problem area identification section using data collection techniques. And rich illustrations illustrate the complete system procedures of VOFS by reviewing these I have detected some definite issues which are given down-

- There is no way to identify and verify the users.
- Location-based order
- Ensure user information
- Online payment system

7.4 Possible solution

The planned solution to the detected problem is

- The new centric project can be launched means VOFS.
- Security and privacy concerns should resolve.
- An automated notification system should develop.
- Using SSLCommerz payment system

7.5 Overview Requirement list

The ultimate whole specification list is given down:

Functional requirements:

- User Authentication System
- Food Category Manipulation
- Food Availability and Stock Management
- Cart Manipulation
- Pre-Ordering System
- Special Ordering System
- Online Payment System
- Food-product Price Prediction System
- Wishlist Manipulation
- Report Observing System
- Food Quality Rating and Review System

Non-functional requirements

- Delivery Status Notification System
- Assure the data and resources are backup safely
- Easy, simple & user-friendly interface
- Validity & Verification in the different user aspects
- Discounts Management or Special Campaign

7.6 Technology to be implemented

Several application technology choices for implementing the planned system. To reach a goal successfully, choosing the proper technology is very important. Those options are below:

Client-server Application Technology

It is a desktop-based technology, where applications are required to be installed on customer devices and data information gathered on isolated servers. Request application to the isolated server for client request data information and server response.

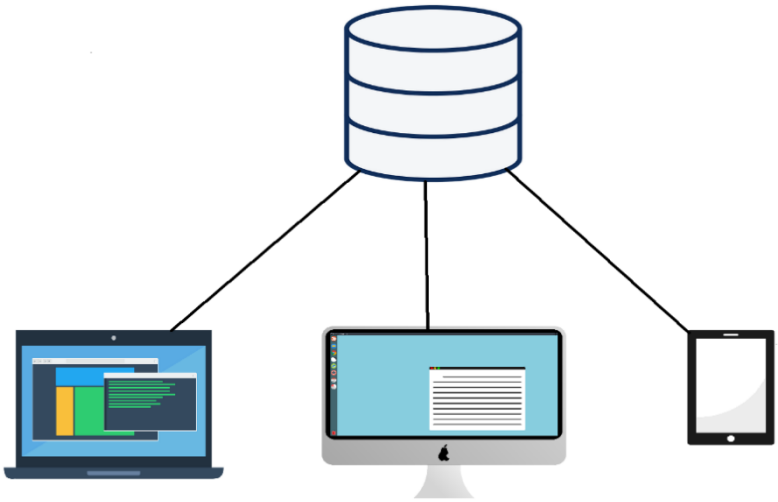


Figure 9: Client-server model

The main functionalities of the client-server application are recorded below:

- The application needs to be installed physically
- Need server & client applications need to be maintained
- Expensive for users to buy apps.
- Not lightweight and always usable.

Web application

The web applications are hosted on isolated servers and it is not required to install something on the customer's device. The application is invaded via the Internet & requires both types of scripting in a web browser, these are client-side & server-side. The major features of Scripting (Kohan, 2019) a web-based application is:

- Not needed installation
- Invaded via internet & web browser on the cloud.
- Comparably low expenses.
- Usable from any place at any time.

7.7 Recommendation and justification

The recommended Vegor Organic Food Supply needs to be available from Dhaka city at all times. Even the system should be usable from any internet-enabled device e.g, smartphones, laptops, desktops, & tablets. It should contain a medium number of users at the same time. This is not feasible using client-server application technology, so the web application is recommended for the planned system. All the people of Dhaka city will be able to order food through the project.

CHAPTER 8

Exploration

8.1 Old system Use Case

The use case diagrams illustrate the progress of work performance of a system. Now I will attach the use case as the old system use case.

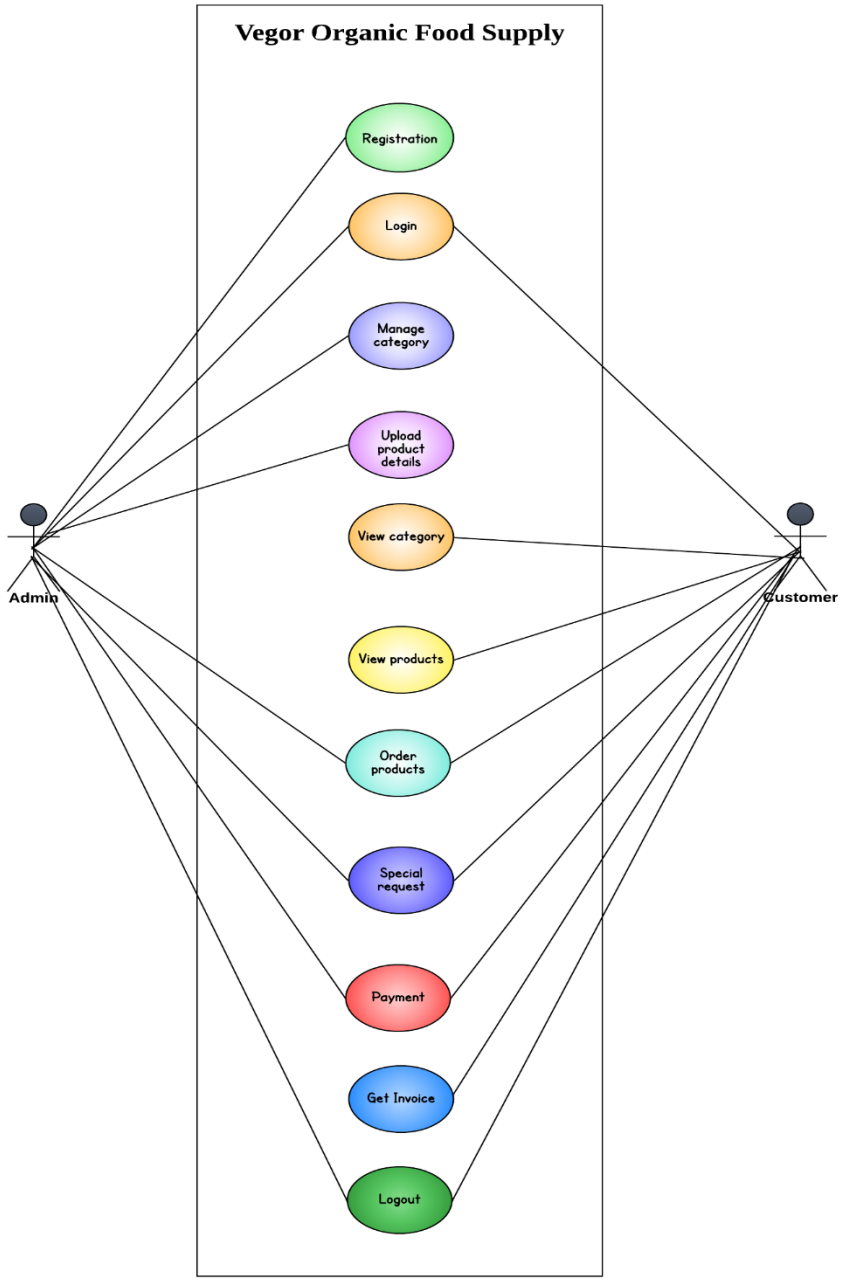


Figure 10: Use case diagram

8.2 Activity Diagram

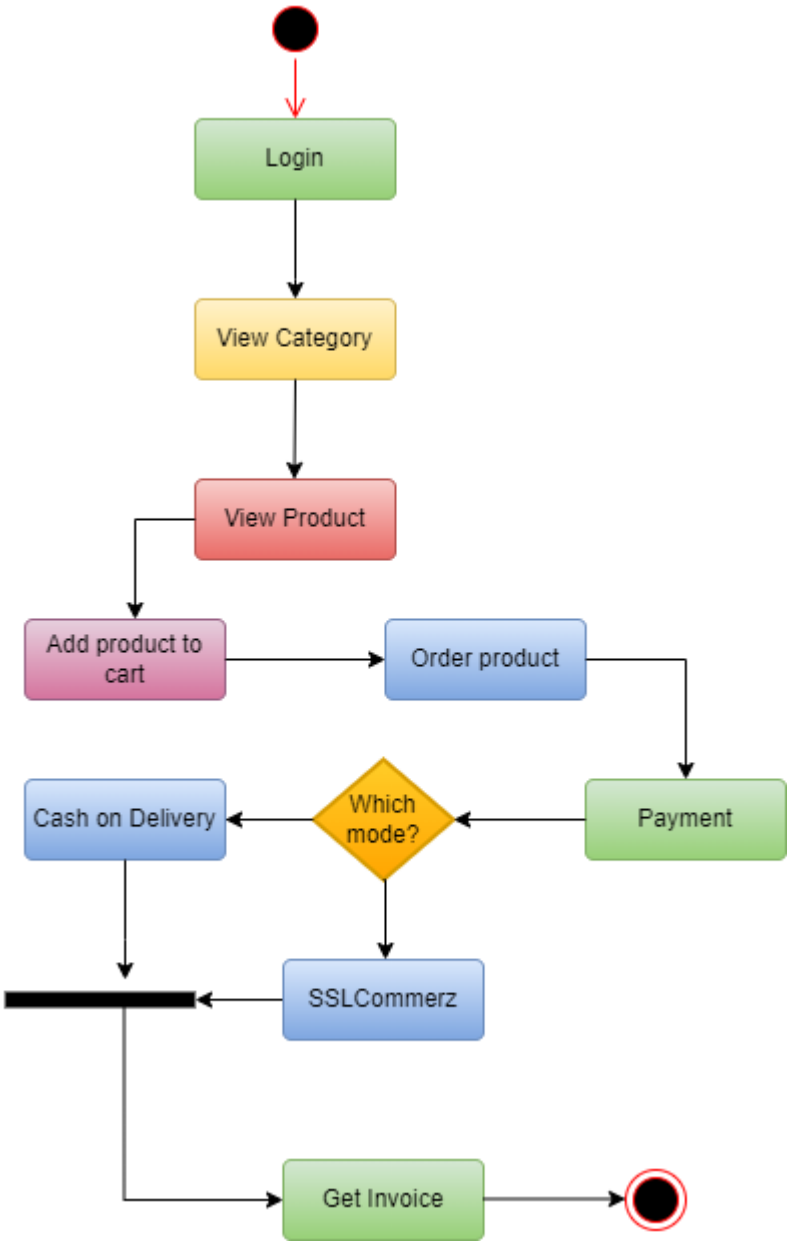


Figure 11: User activity diagram of VOFS

8.3 Full System Use Case Diagram

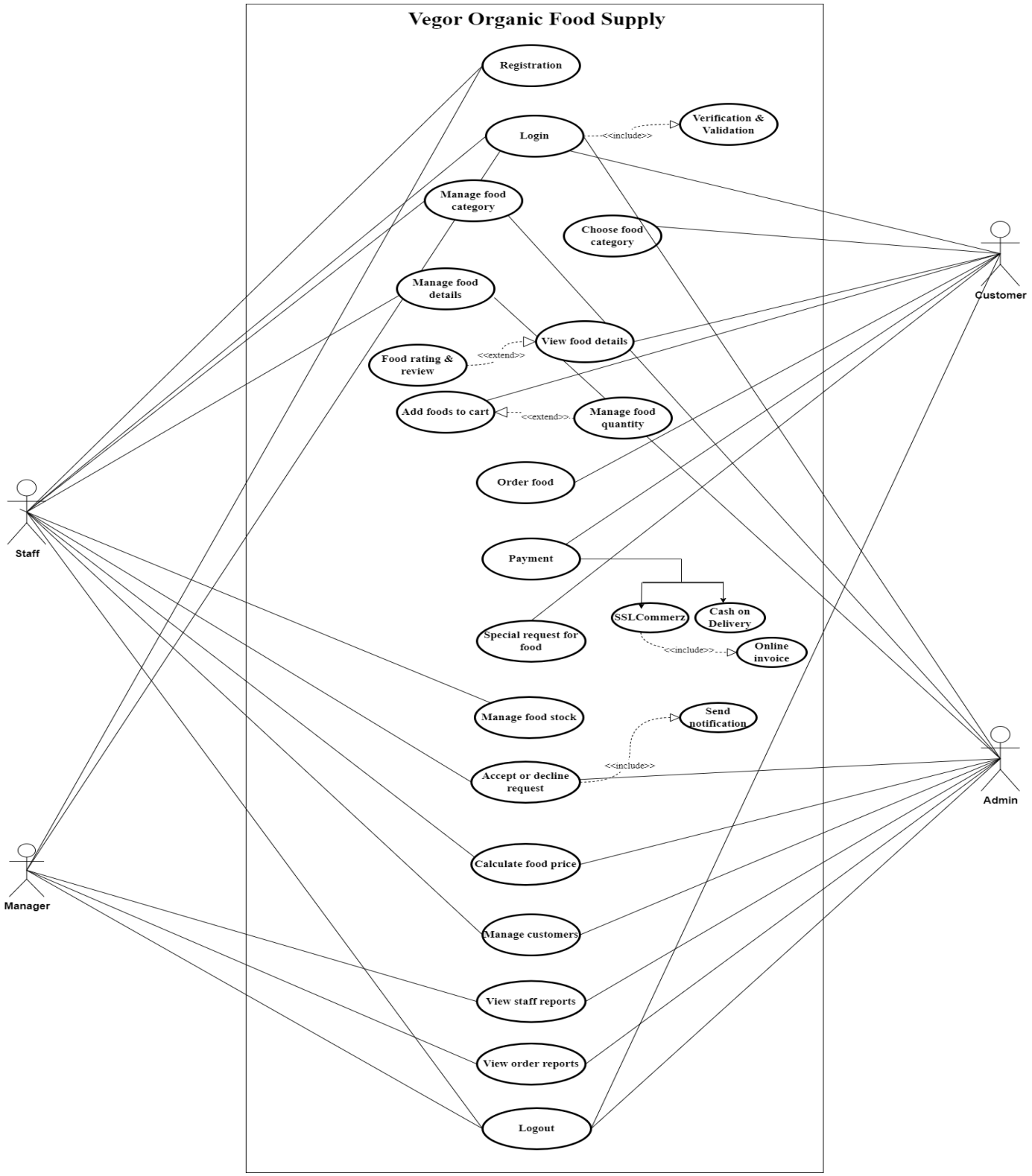


Figure 12: Use case of the purpose of VOFS

8.4 Full System activity diagram

The recommended VOFS project has several types of users with several workflows. I am attaching the activity diagrams for these work flows below:

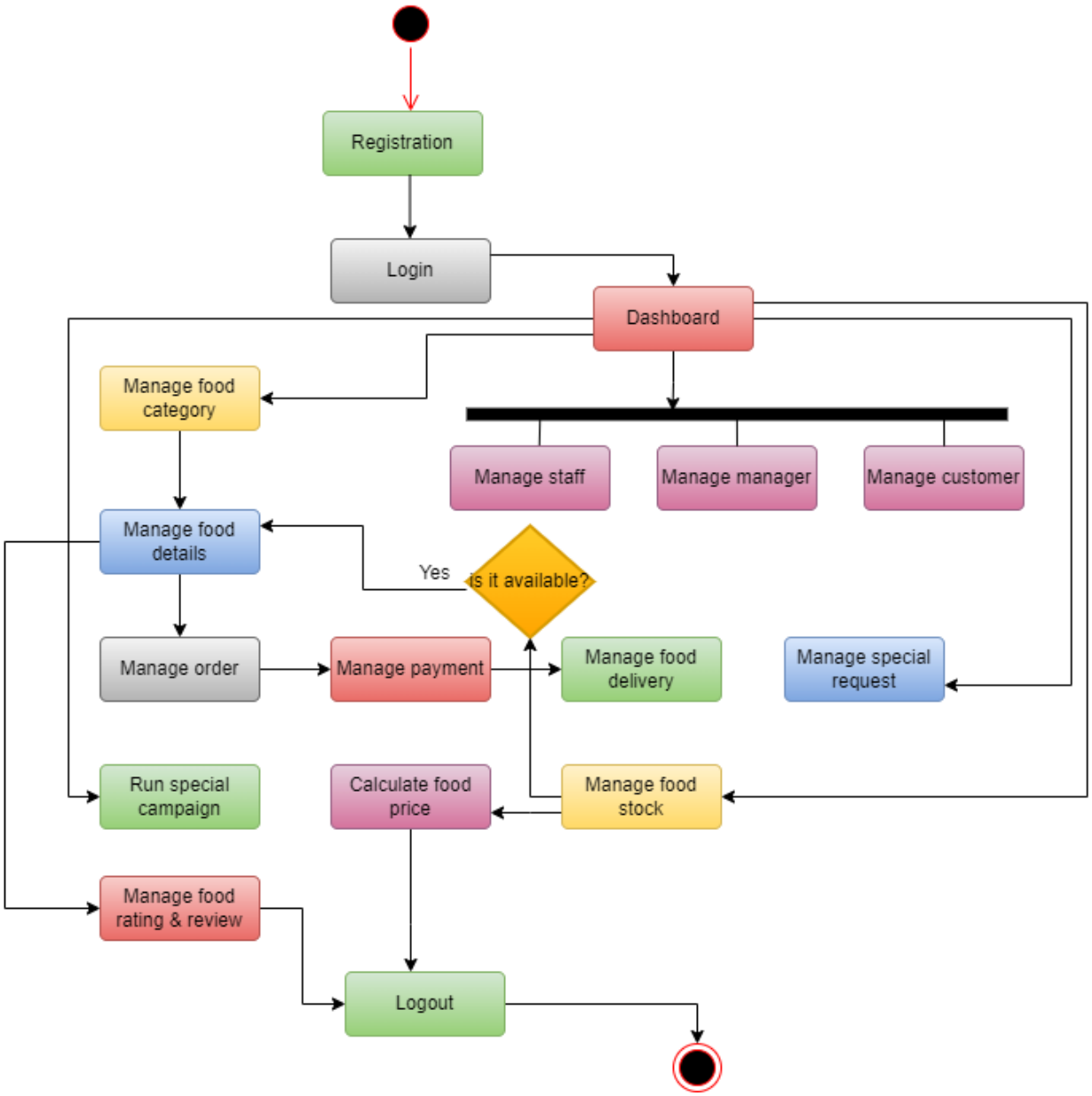


Figure 13: Admin's activity diagram of VOFS

Manager full activity is given below:

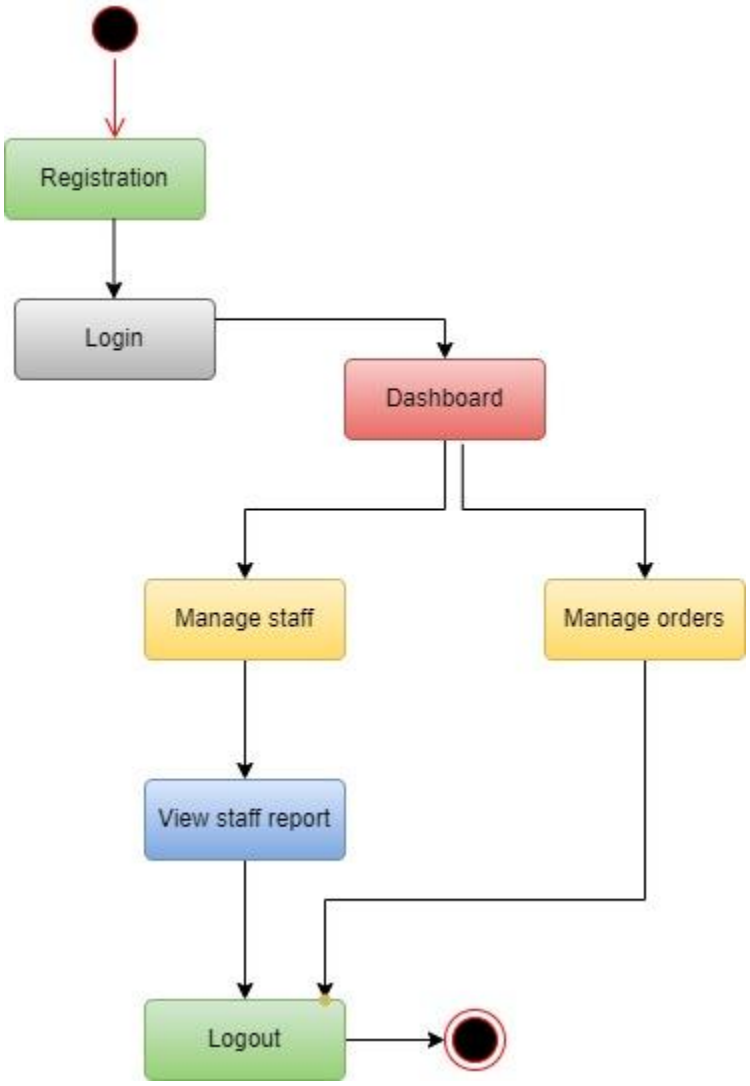


Figure 14: Manager's activity diagram of VOFS

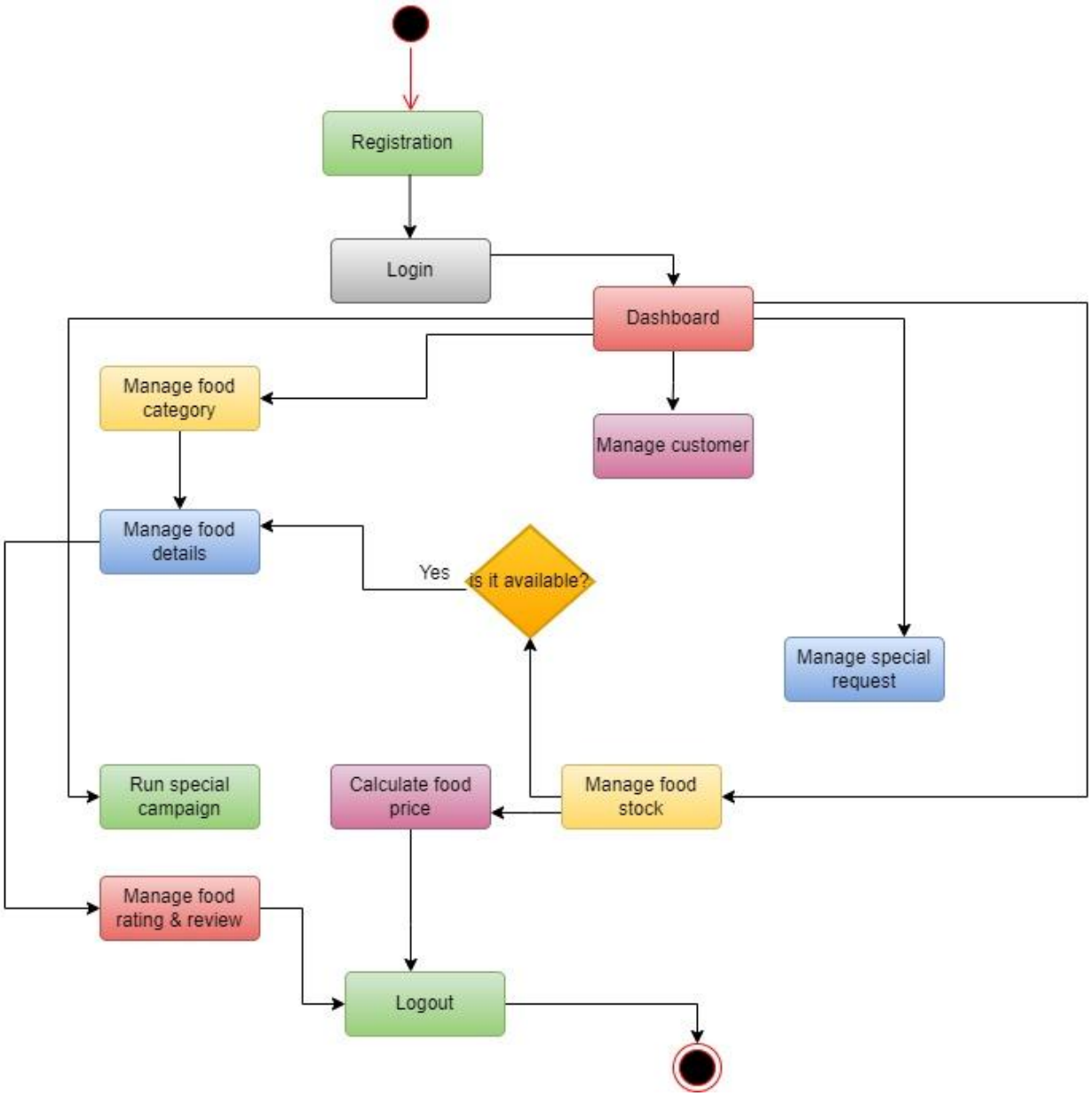


Figure 15: Staff's activity diagram for VOFS

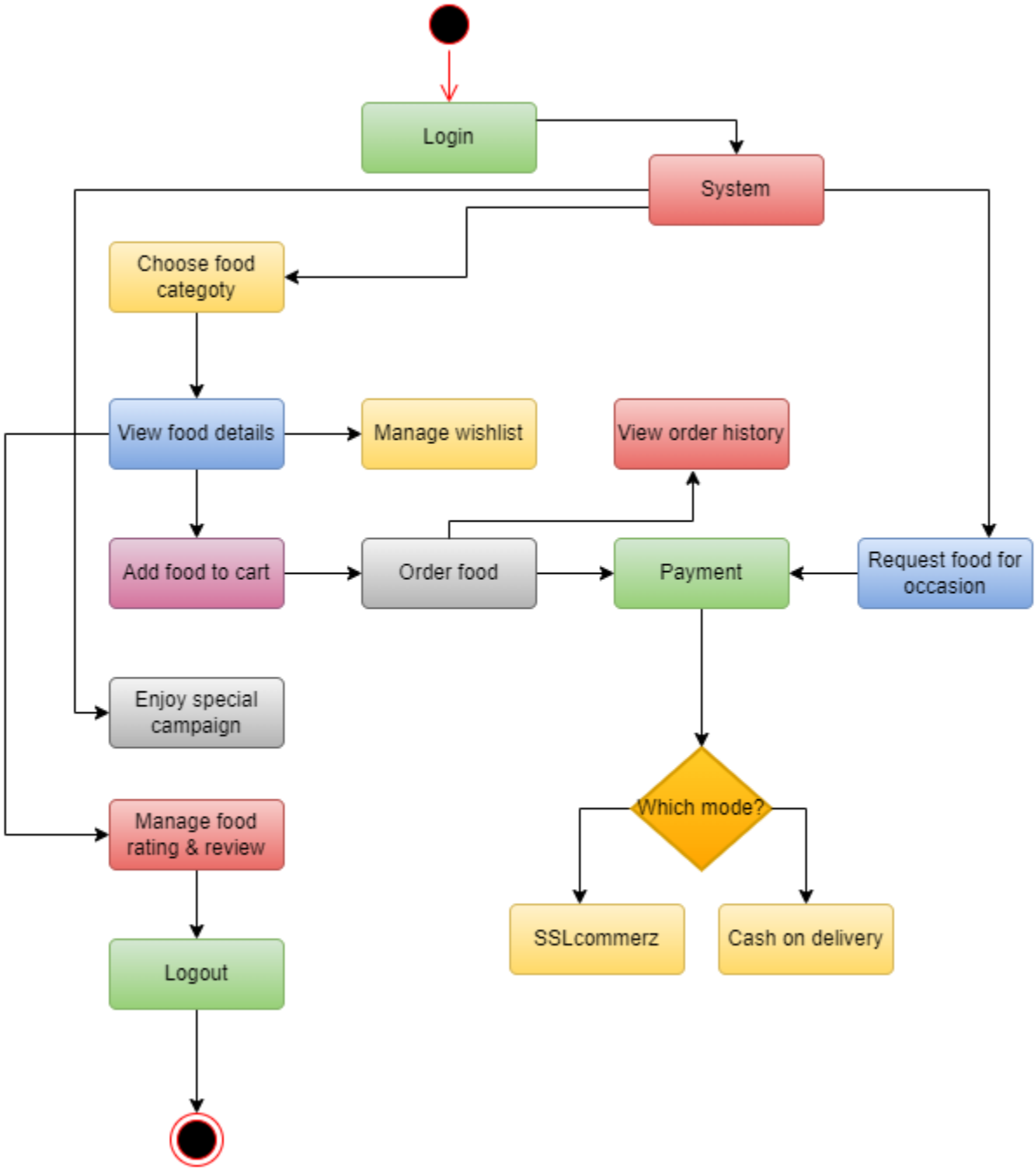


Figure 16: Customer's activity diagram for VOFS

8.5 Requirement catalog

The required catalog is a list of all the identified requirements of a project. Here I am going to add the requirements identified in the required catalog following the standard format.

Authentication Requirement Catalogue

Source	**Sign in**	**Priority**	**Requirements**
Admin	All user	Must	M-001
Functional Requirement			
User Authentication System	All users need to be registered & logged in to the system. Only the elite customers do need to register on their own, they will access the system by login		
Nonfunctional Requirements			
Description	Target value	Acceptance value	comment
Login registration per day	15	10	

Table 16: Requirement catalog for login registration

Choose Category and Products Requirement Catalogue

Source	Sign in	Priority	Requirements
Admin	All user	Must	M-002
Functional Requirement			
Choose category and food	Customers will be able to choose category & food. They will be able to view food details and availability.		
Nonfunctional Requirements			
Description	Target value	Acceptance value	Comment
Choosing category and product per day	15	10	

Table 17: Requirement catalog for choosing categories and products

Pre-ordering Requirement Catalogue

Source	Sign in	Priority	Requirements
Admin	All user	Must	M-003
Functional Requirement			
Pre-ordering system	After ordering food, customers can pay money online or cash on delivery.		
Nonfunctional Requirements			
Description	Target value	Acceptance value	Comment
Pre-order per day	15	10	

Table 18: Requirement catalog for Pre-booking

8.6 Prioritized Requirements List (PRL)

I have the user Moscow priority strategy to create a priority list of identified requirements. A list of priority requirements for the Vegor Organic Food Supply system is given below:

Must-have requirements -

SL	Requirements	Priority
1	User Authentication System	Must have
2	Food Category Manipulation	Must have
3	Food Availability and Stock Management	Must have
4	Pre-Ordering System	Must have
5	Online Payment System	Must have
6	Special Ordering System	Must have
7	Food Price Prediction System	Must have

Should have requirements -

SL	Requirements	Priority
1	Delivery Status System (Notification)	Should have
2	Food Quality Rating and Review System	Should have
3	Report Observing System	Should have
4	Wishlist Manipulation	Should have

Could have requirements -

SL	Requirements	Priority
1	Discounts Management/Special Campaign	Could have

8.7 Prototype of the new system

Vegor Organic Food Supply's some of the prototypes given below:

Admin panel prototype

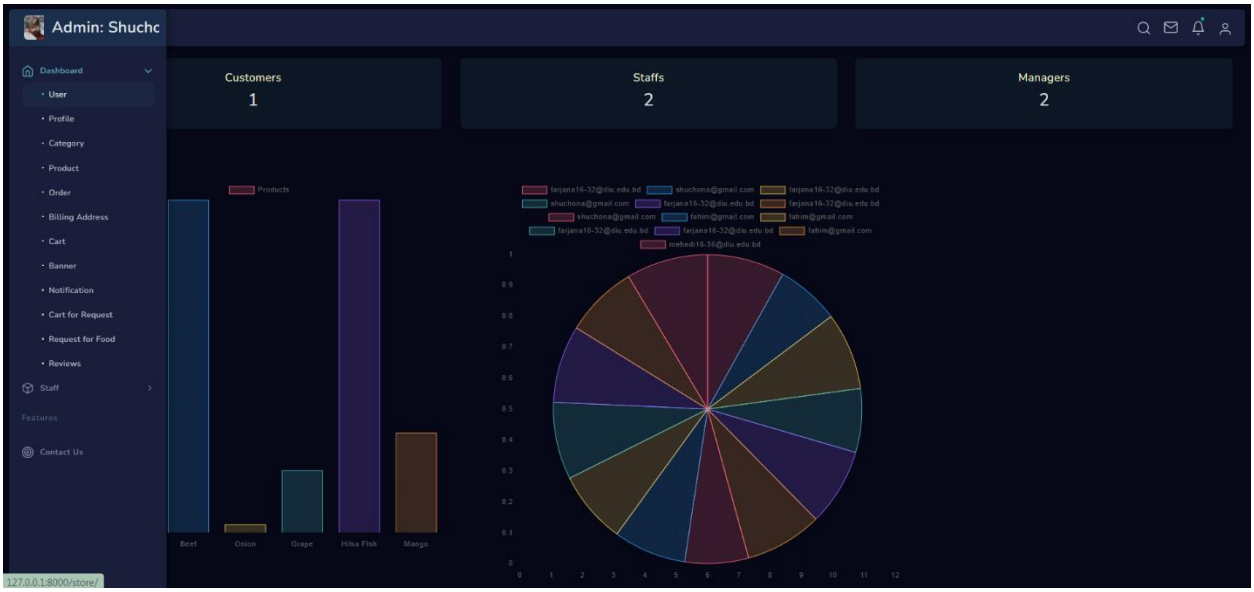


Figure 17: Admin panel prototype

Login Page Prototype

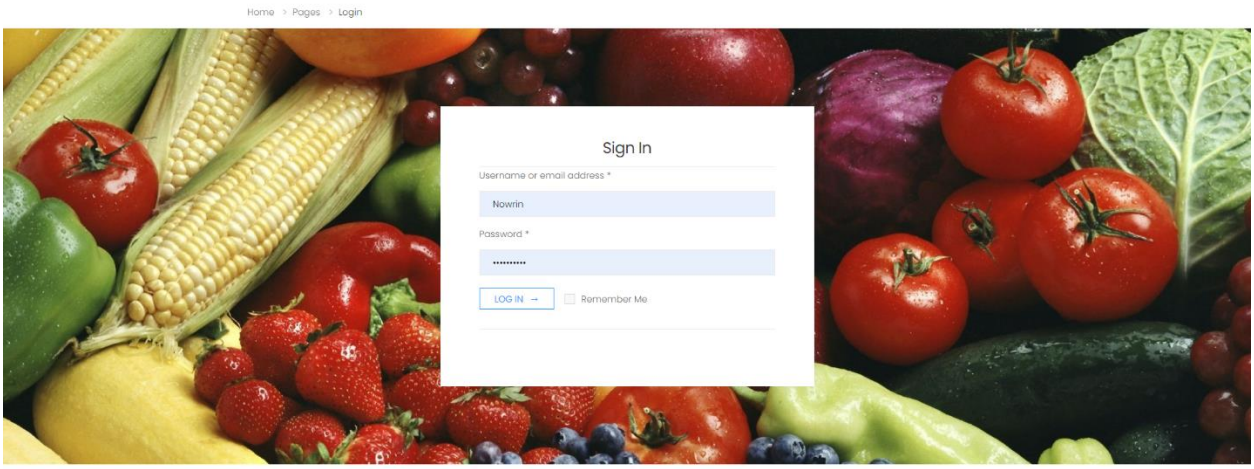


Figure 18: Login Prototype

Food Category Prototype

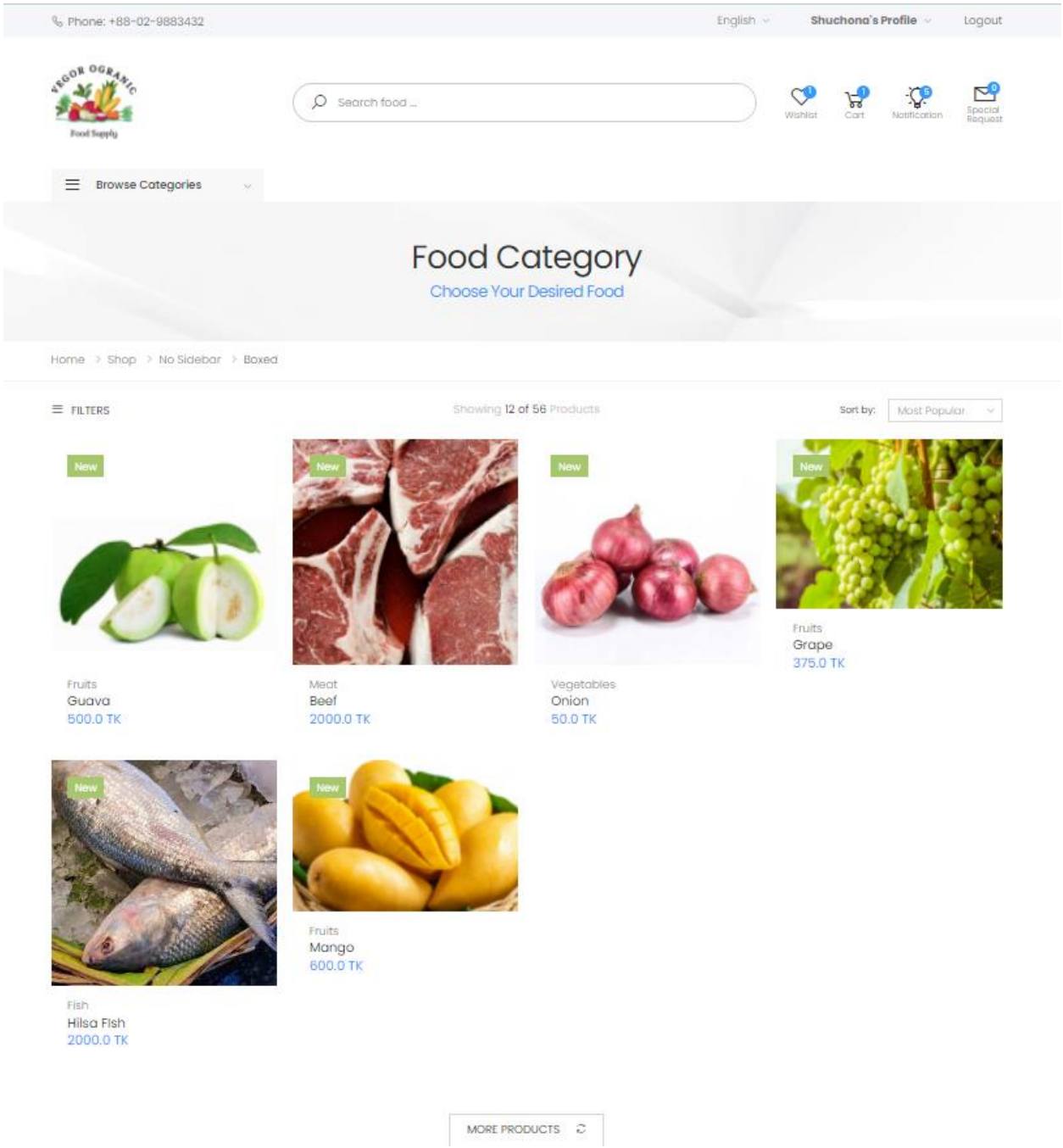


Figure 19: Food Category prototype

Product Page prototype

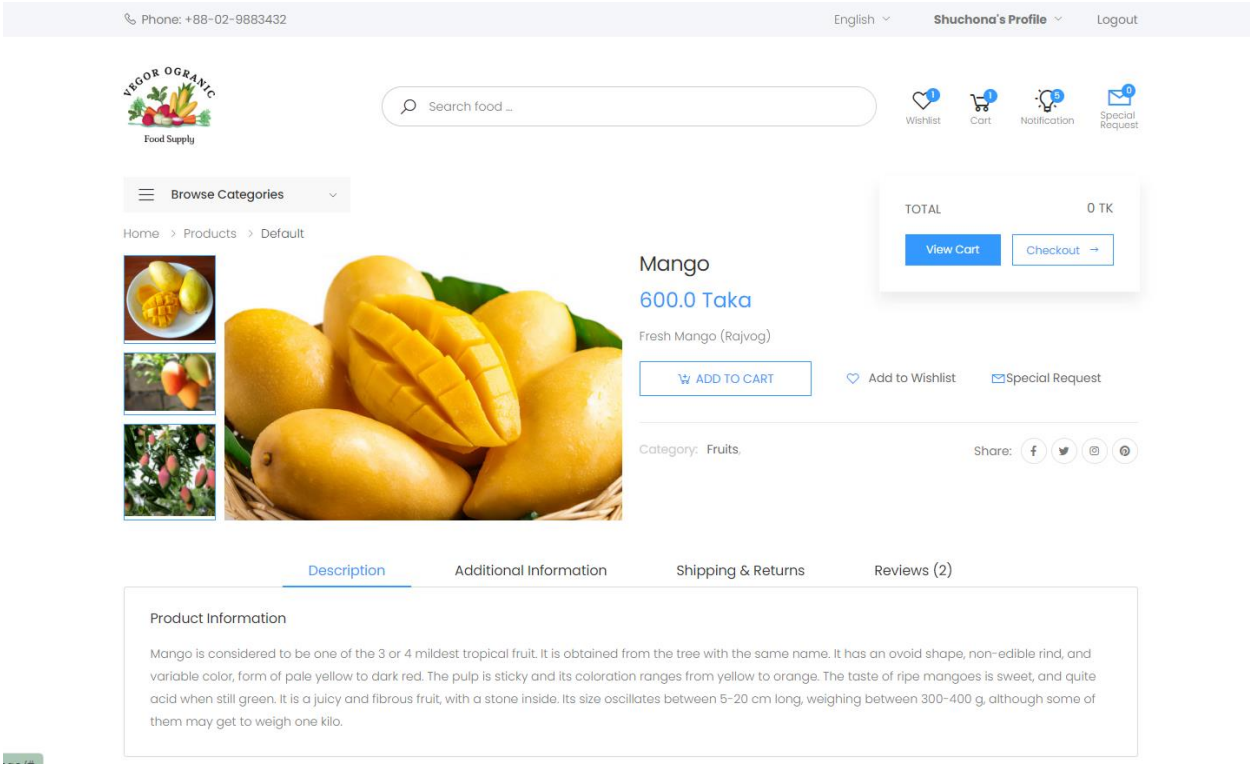


Figure 20: Product page prototype

Cart Prototype

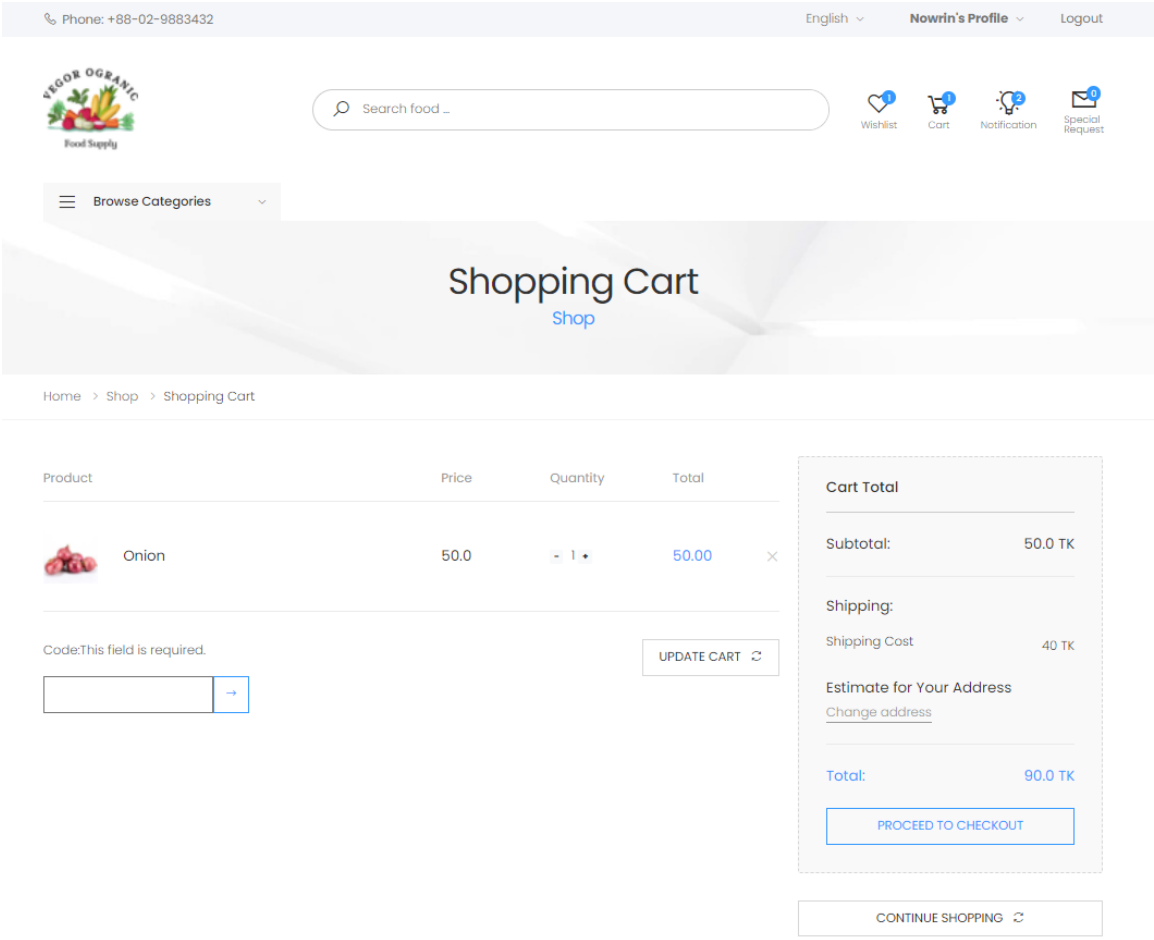


Figure 21: Cart Prototype

SSLCommerz Payment prototype

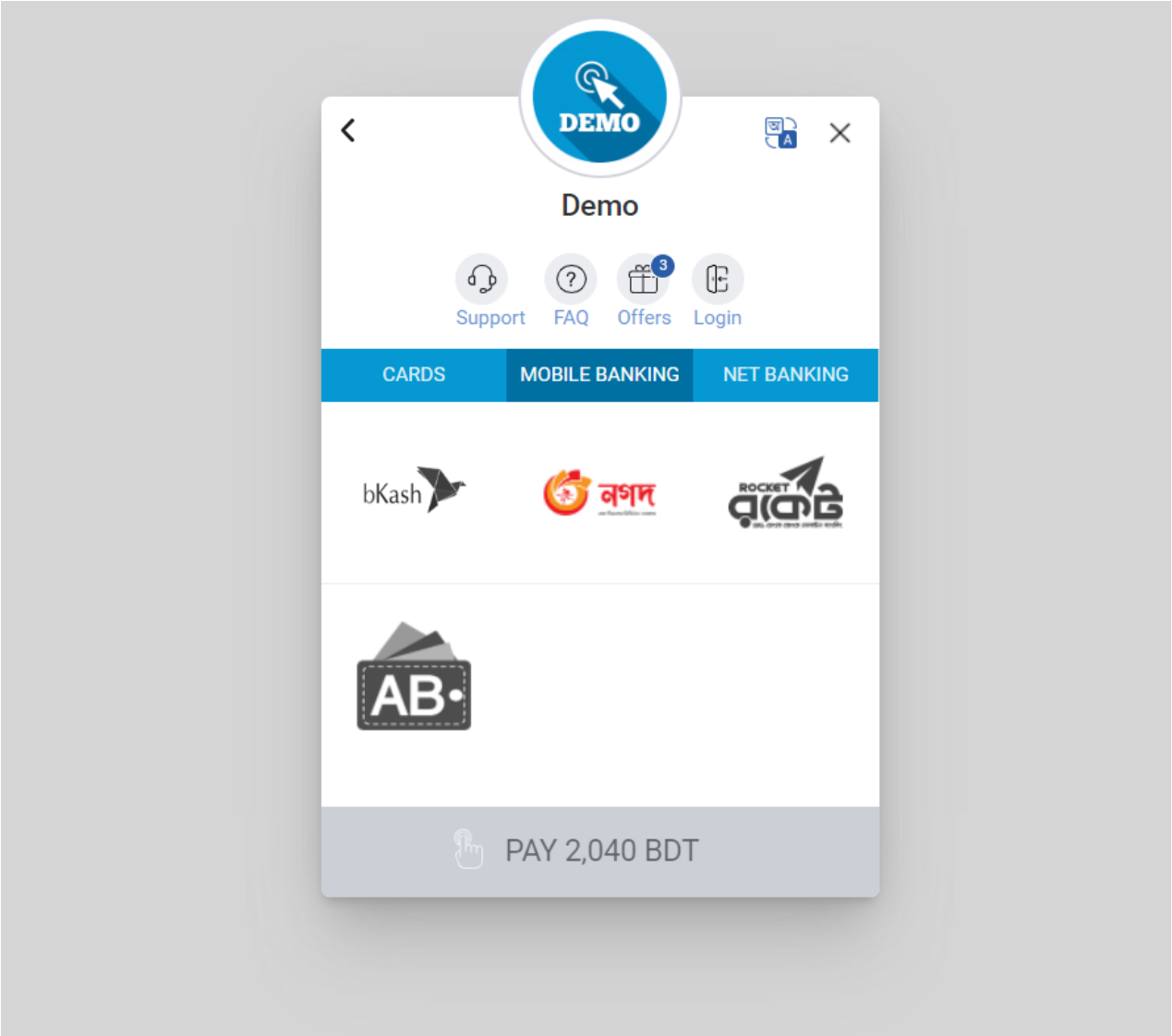


Figure 22: SSLCommerz Payment Prototype

CHAPTER 9

Engineering

9.1 New system modules

There are some modules in the VOFS system. So, I am explaining some of the key modules, they are listed below along with the work process.

Login module

SL	User Action	SL	System action
1	Users need to press the link on the VOFS website	1	The system will show a form for login
2	The user will give a login data	2	The system validates all of the provided information
3	User press the login button	3	If authentication validates it will direct to the index page if the login action is done by customers. On the other hand, if it is done by admin, staff, or manager, it will direct their dashboard page. If any wrong information is provided, it will show an error.

Table 19: Login module

View food category & food module

SL	Customer Action	SL	System action
1	Customers will login to the system	1	It will show the main page where the food category will show
2	Customers will choose the category	2	It will show foods as the category
3	Customers will view products, and enter the food details page	3	It will show the details of the food

Table 20: View food category & food module

Wishlist module

SL	Customer action	SL	System action
1	Customer will add food in their Wishlist	1	The chosen foods will show on Wishlist section

Table 21: Wishlist module

Cart module

SL	Customer action	SL	System action
1	Customers will be able to add different categorized foods to their cart, besides, they will be able to increase and decrease the quantity of chosen foods	1	The food which is chosen to take in the cart will show in a cart section
2	After adding foods to the cart, customer will have to complete their billing address properly	2	The checkout section will show a form of billing address and their total cost for proceed the payment

Table 22: Cart module

Order module

SL	Customer action	SL	System action
1	After coming to the checkout page, they will have to click on the order button to order successfully	1	When the order will be placed successfully, they will get instant mail with their invoice, also, after ordering successfully, they will be able to print their invoice. On the other hand, if the order is not placed successfully, it will show an error page with the message that the order is not placed

Table 23: Order module

Report module

SL	Manager action	SL	System action
1	Manager will login to the system	1	It will come to the dashboard page
2	The manager will view the order reports	2	The system will allow the manager to show the order history
3	The manager will view the price prediction reports	3	The system will allow the manager to show the price prediction activities

Table 24: Report module

Staff wok module

SL	Staff action	SL	System action
1	Staff will login to the system	1	It will come to the dashboard page
2	Upload food details	2	The system will show the food details to the customers
3	Run special campaign	3	The special campaign will show as a special offer on the banner section

Table 25: Staff module

Food price prediction module

SL	Staff action	SL	System action
1	Staff will login to the system	1	It will come to the dashboard page
2	Staff will input all amounts which are costs for growing a food product	2	The system will show a form to save the amounts.
3	Staff will click on the 'Save' button	3	The system will calculate the final price & show all the costs as a report

Table 26: Food price prediction module

Payment module

SL	Customer action	SL	System action
1	The customer will proceed to order	1	The system will provide the option to pay; cash on delivery & SSLCommerz
2	If customers choose SSLCommerz, they will redirect to the SSLCommerz page to pay as their available account	2	The system will show the successful message if the payment is successfully done by allowing the print invoice
3	The customer will download the invoice		The system will send emails to customers mail with the invoice
4	On the other hand, they will be able to order their products in cash on the delivery system		The system will accept their order & save their order in order history

Table 27: Payment module

9.2 Use-case diagram for VOFS

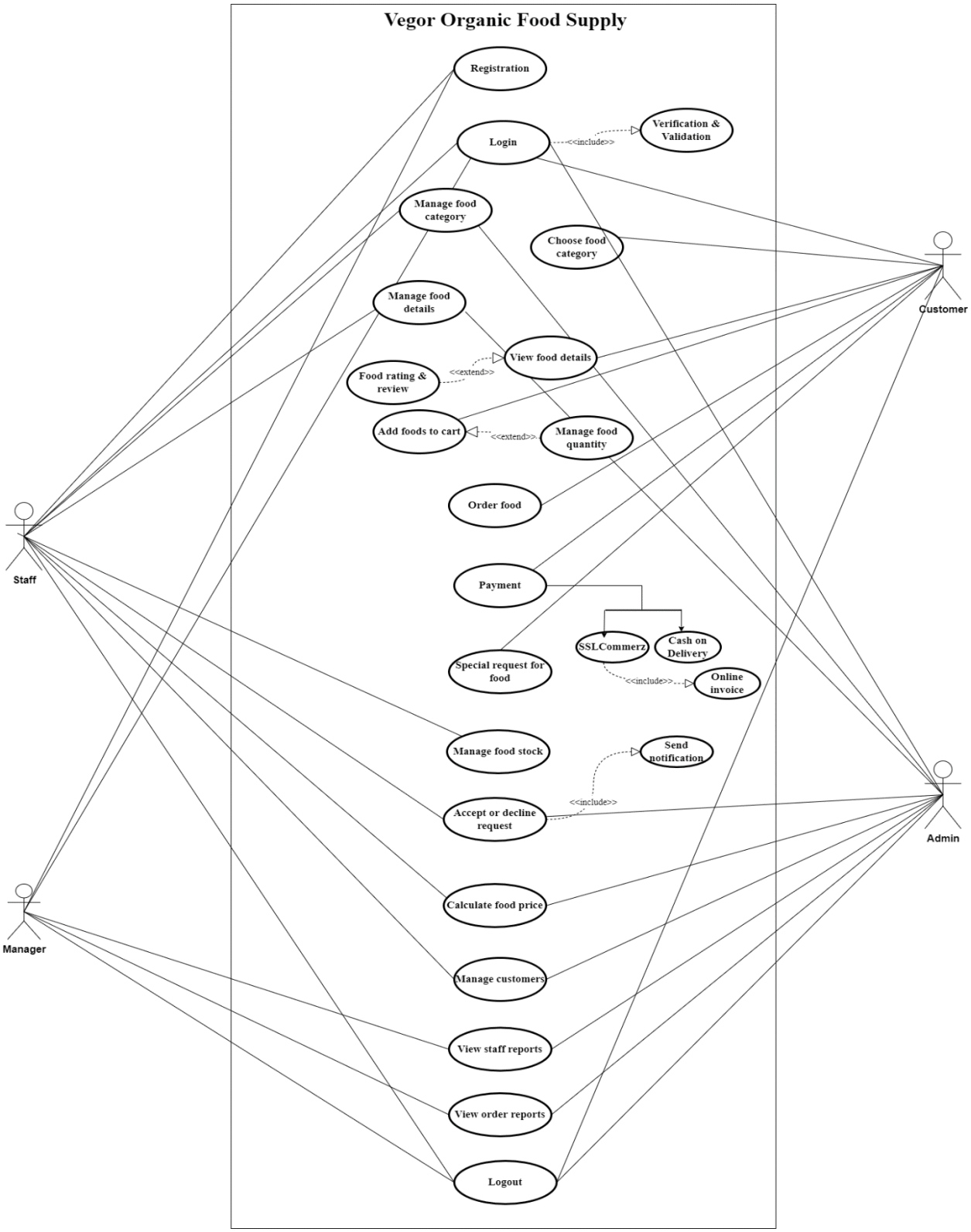


Figure 23: Use case diagram for VOFS

9.3 Class diagram of VOFS

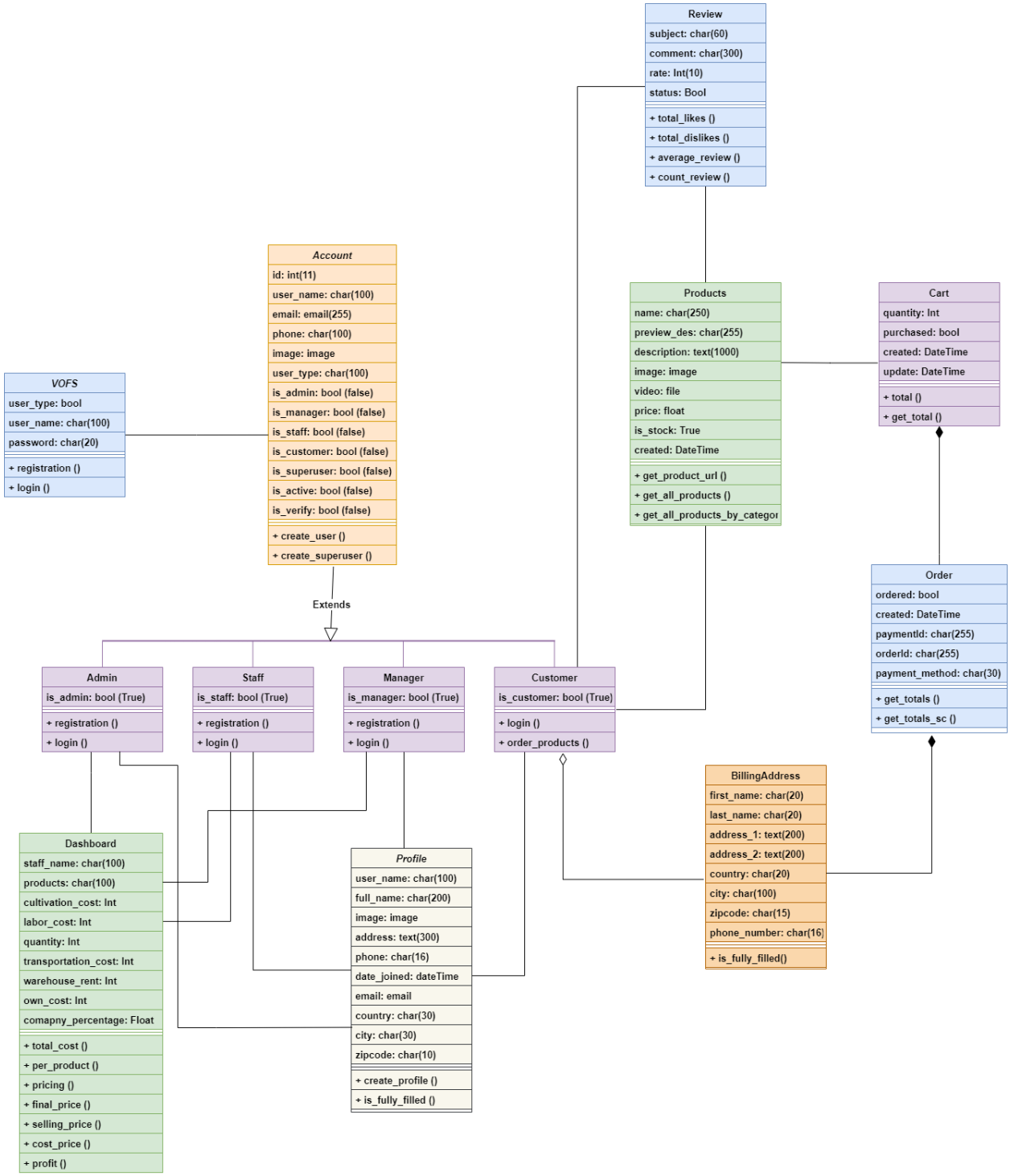


Figure 24: Class diagram of VOFS

9.4 Entity Relationship diagram

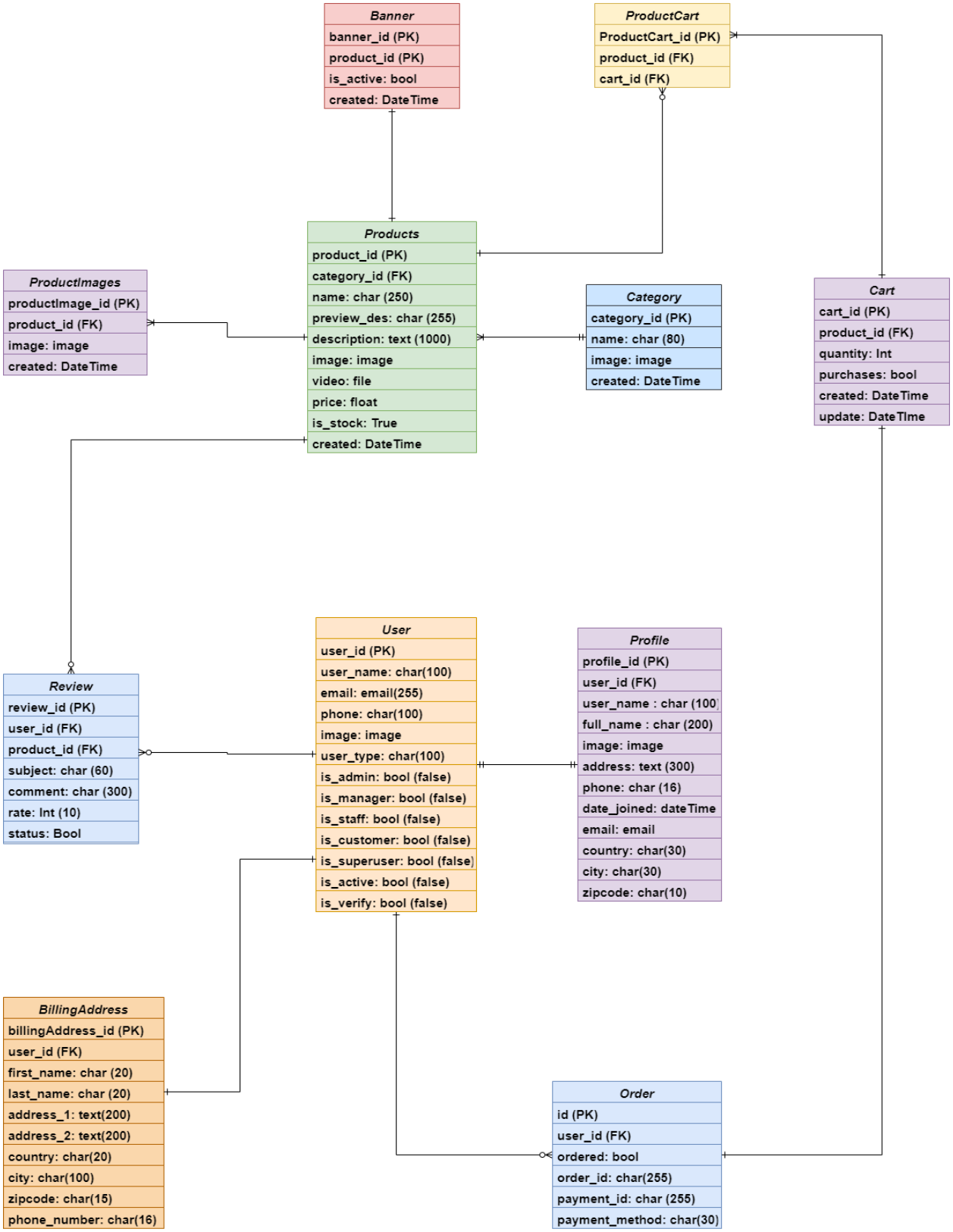


Figure 25: Entity Diagram of VOFS

9.5 Sequence Diagram

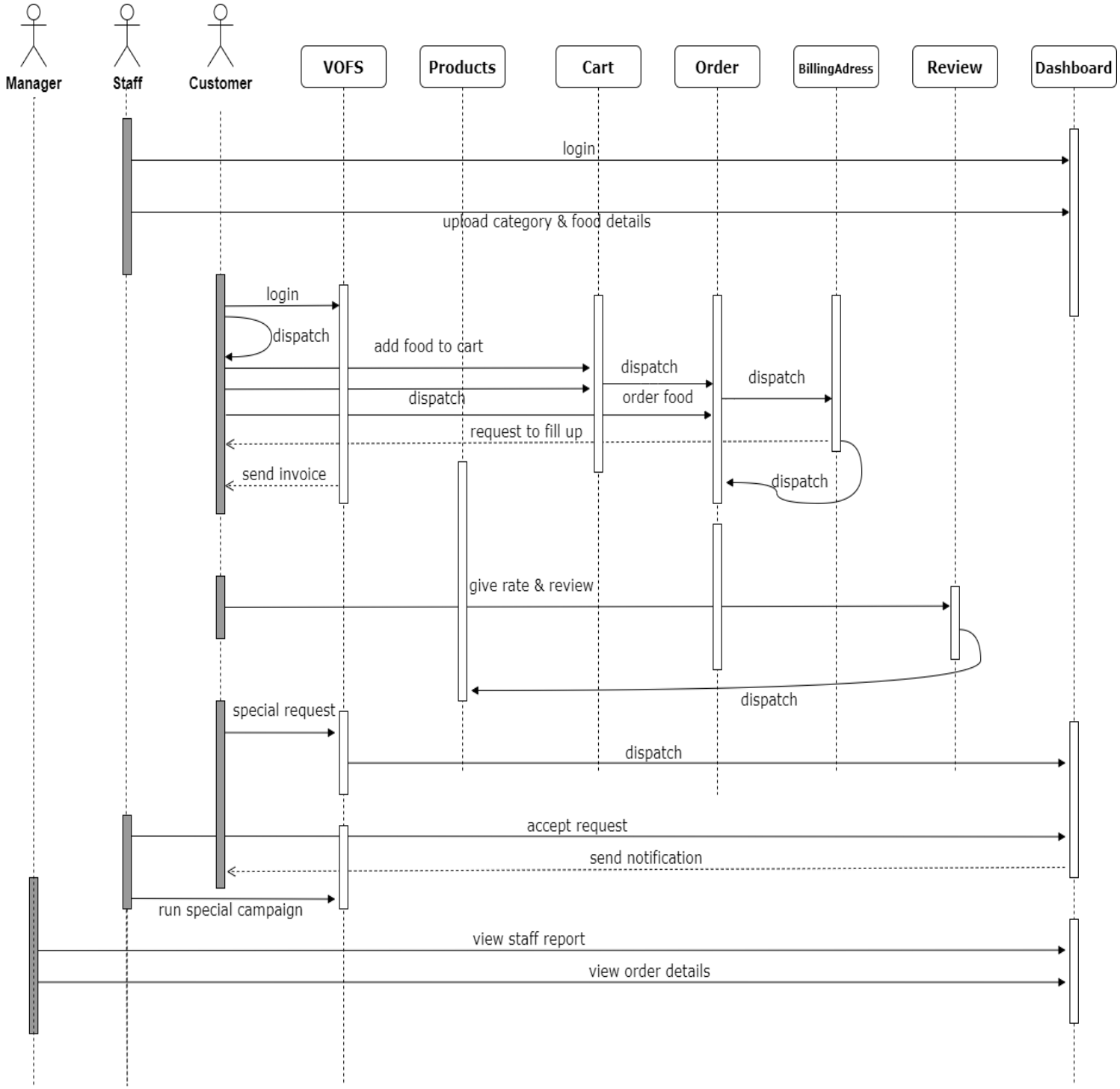


Figure 26: Sequence Diagram of VOFS

9.6 Component diagram of the VOFS

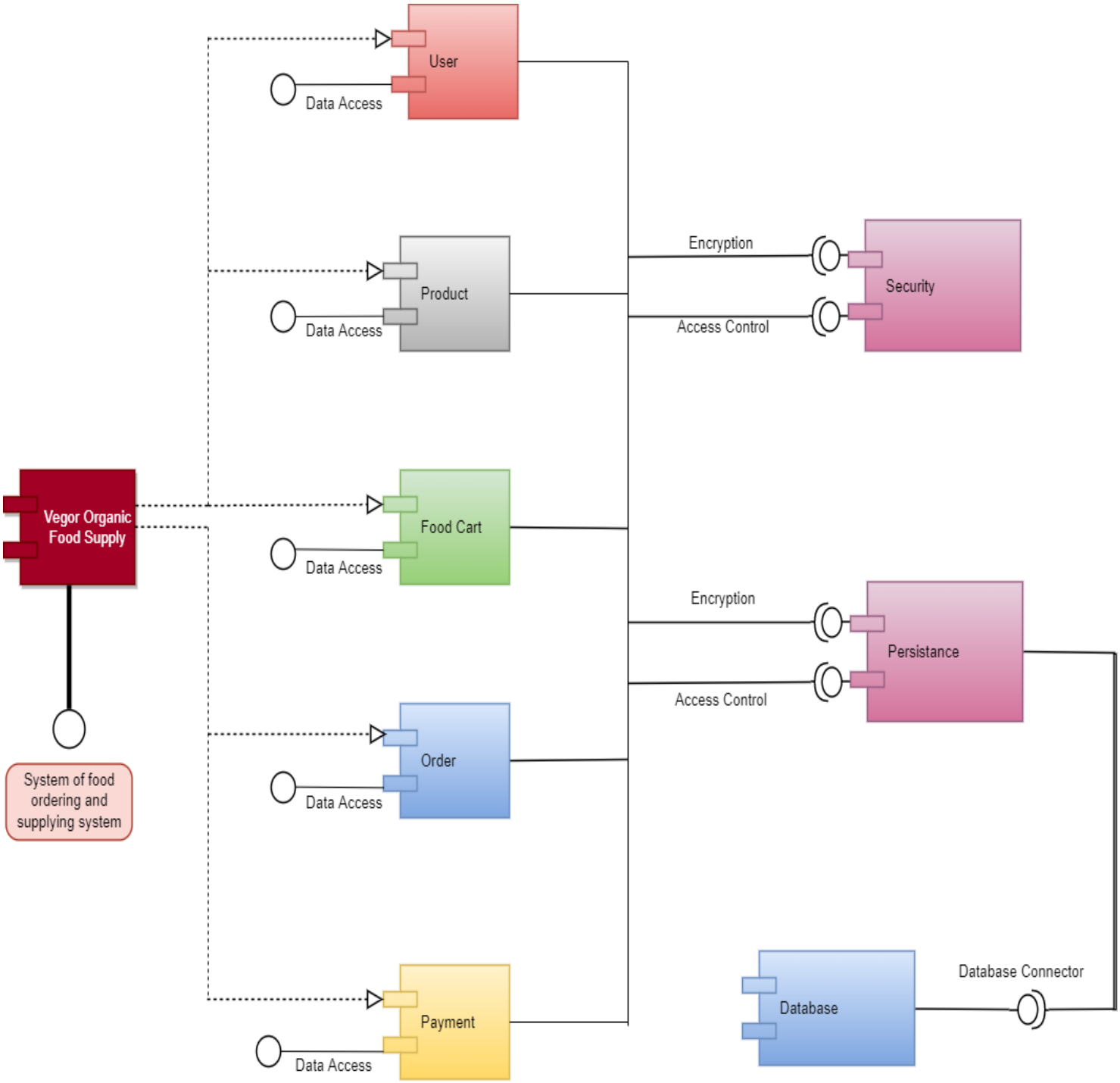


Figure 27: Component diagram of the VOFS

9.7 Deployment Diagram

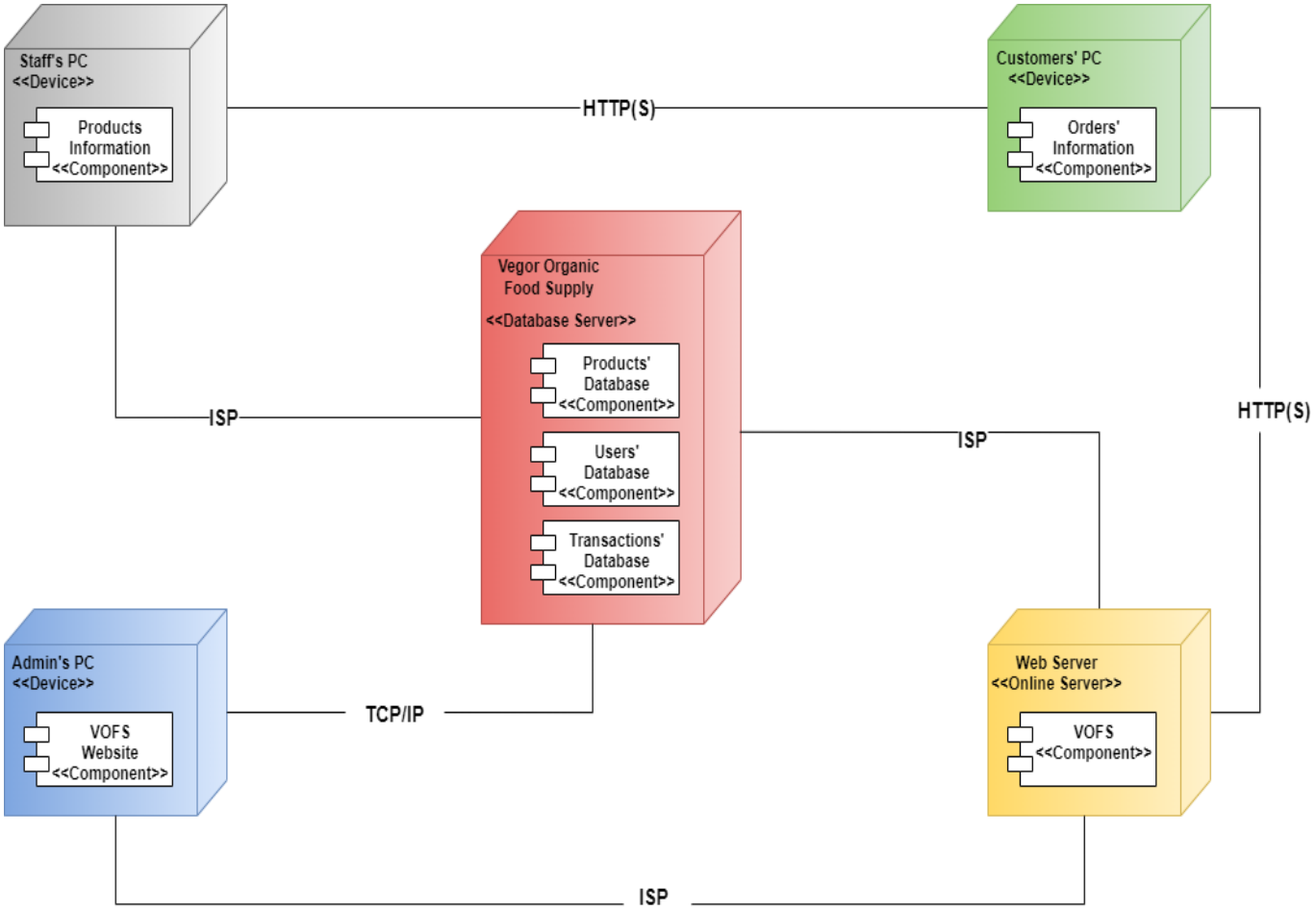


Figure 28: Deployment Diagram of VOFS

9.8 System interface design

Home page interface

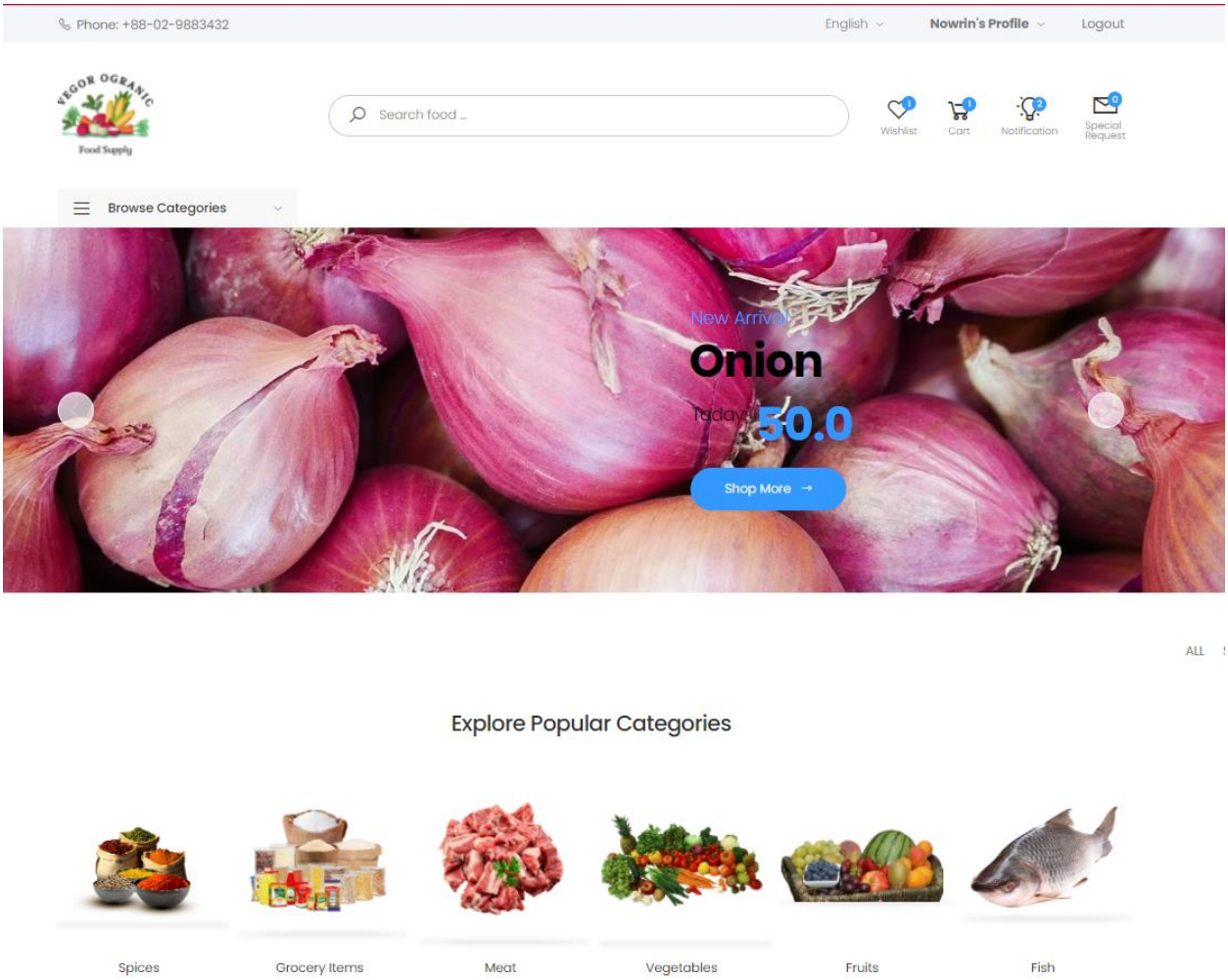


Figure 29: Home page interface

Food Category interface

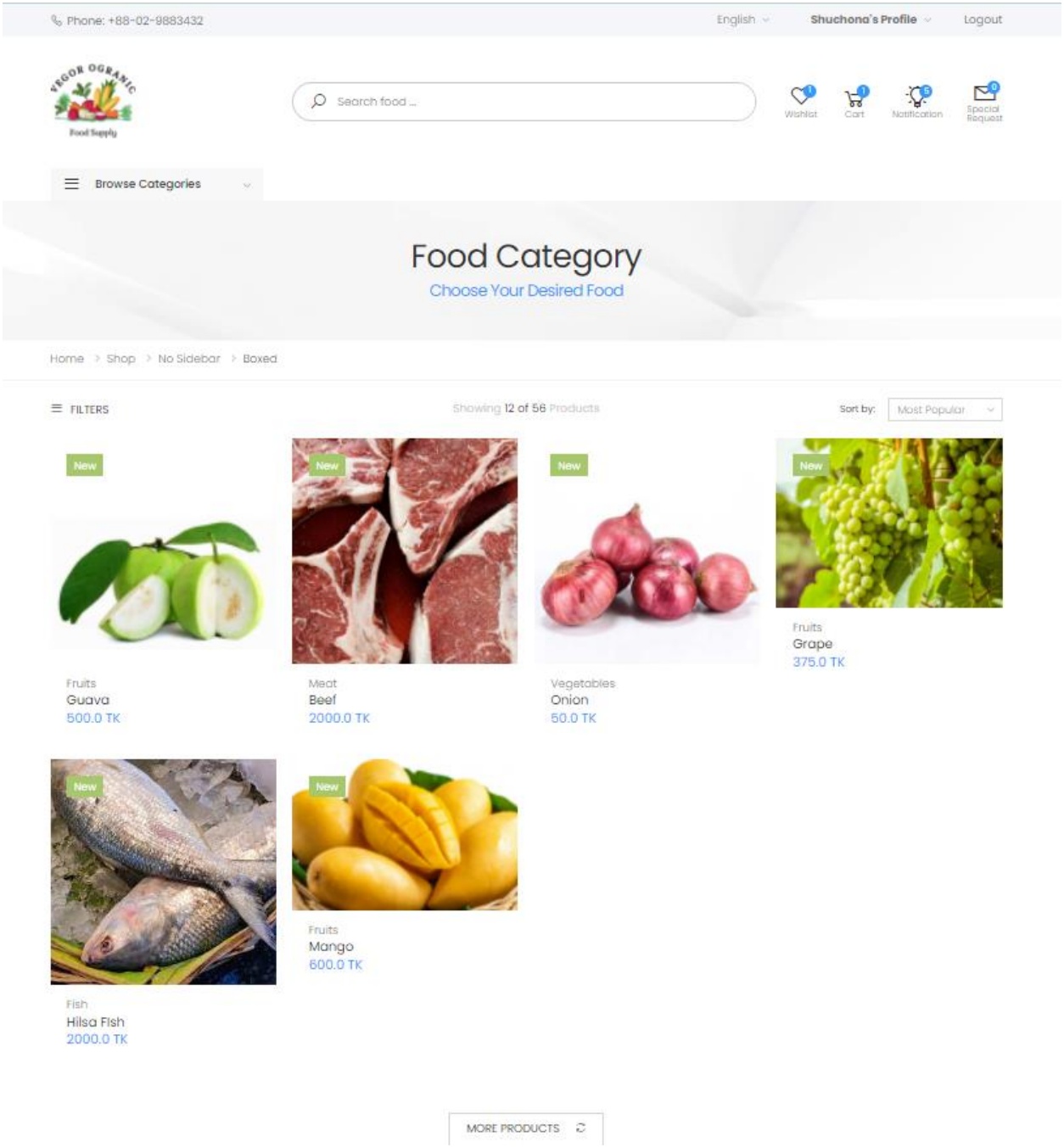


Figure 30: Food Category interface

Admin panel interface

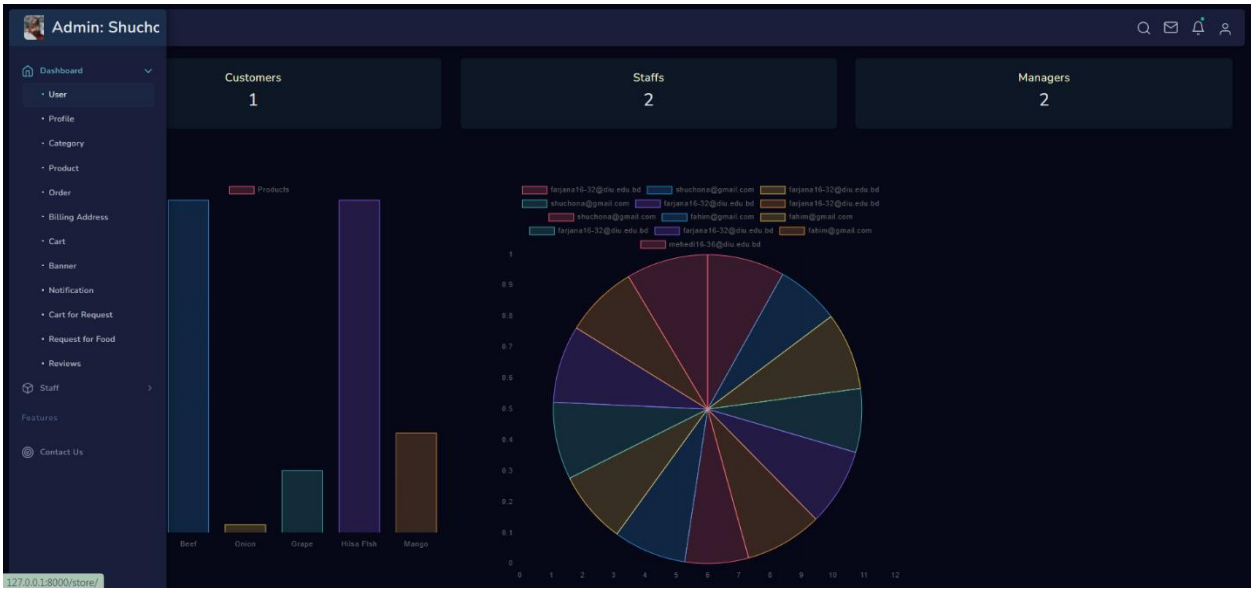


Figure 31: Admin Panel interface

Add products by admin interface

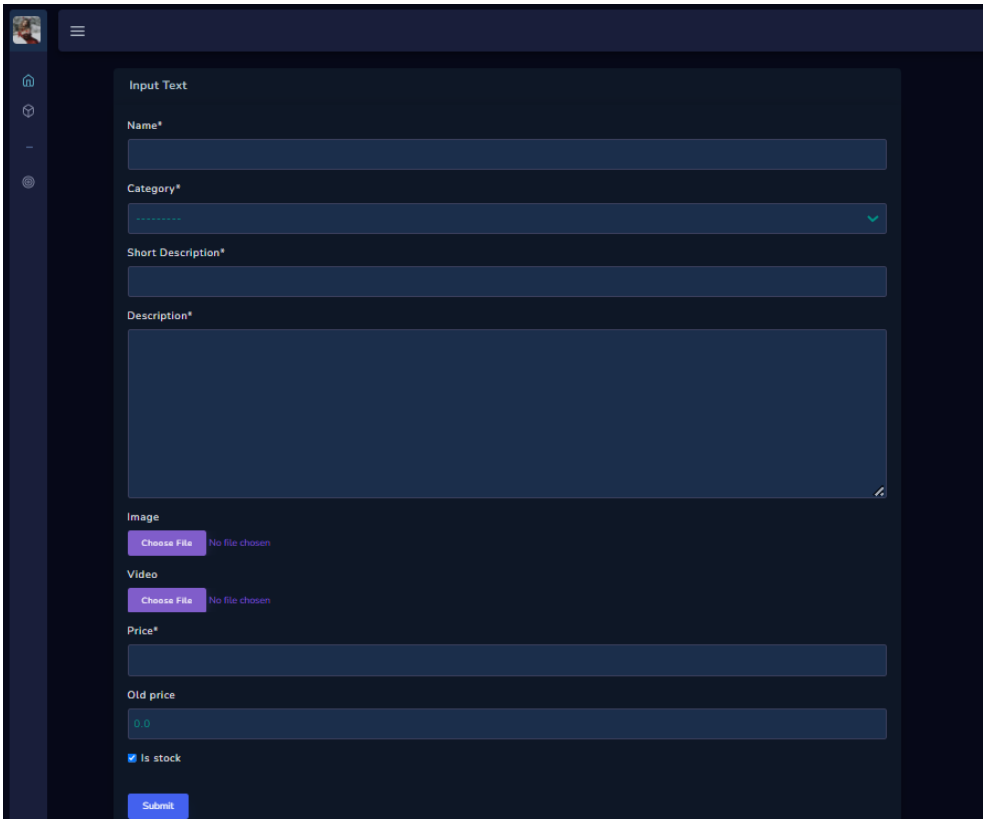


Figure 32: Add products by admin interface

Customer profile interface

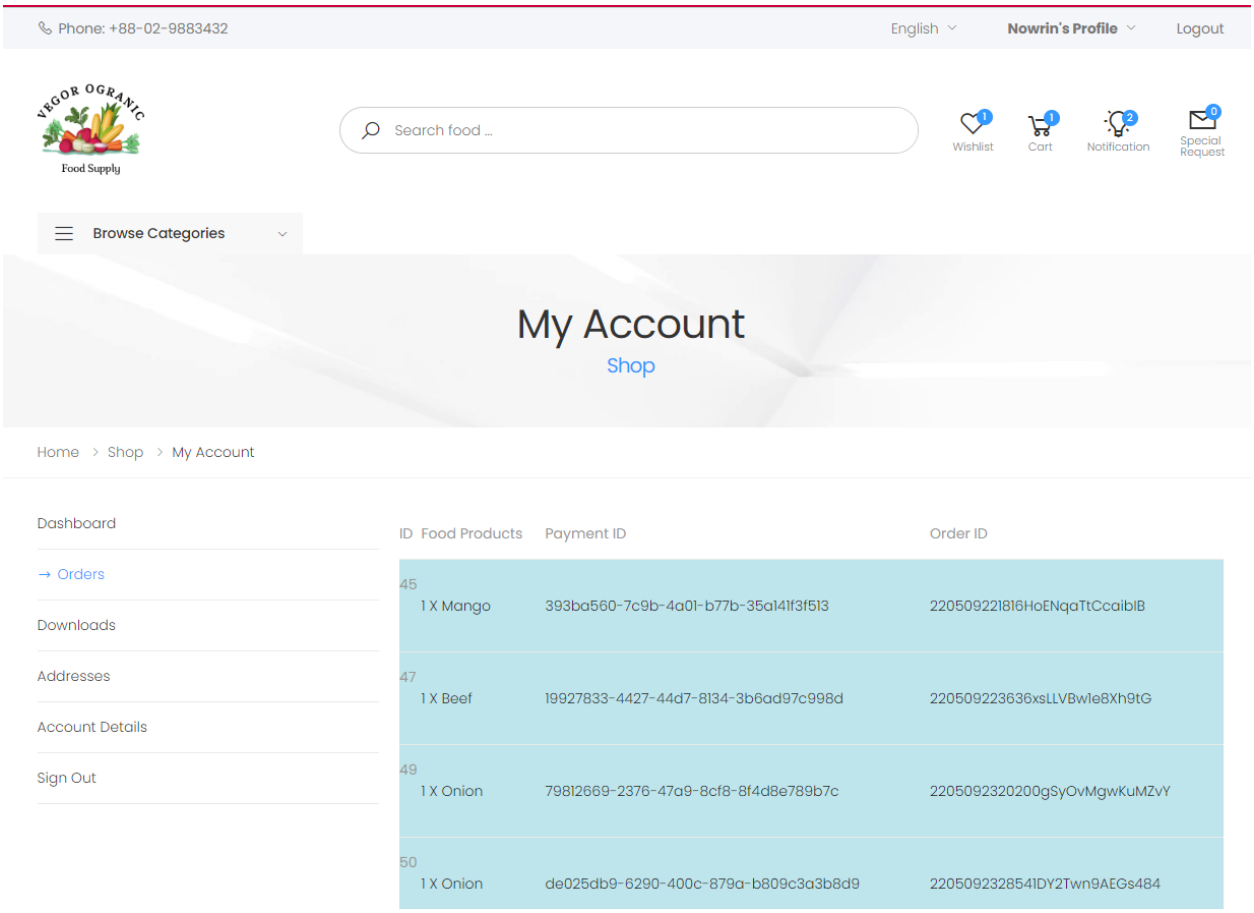


Figure 33: Customer profile interface

CHAPTER 10

Deployment

10.1 Core module coding sample

I have used some language for coding. They are JavaScript, HTML5, Bootstrap, jQuery & CSS for the front-end design of VOFS. For the database management system, I have used SQLite and Python-Django framework for the back-end development. The main portions of coding samples are given below:

Login script for the user:

payment\views.py | product_list.html | dashboard\forms.py | dashboard\views.py | store\forms.py

```python

def adminLogin(request):
    if request.method == 'POST' or request.method == 'post':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(request, username=username, password=password)

        if user is not None and user.is_admin:
            login(request, user)
            messages.success(request, f"Successfully Login as Admin!")
            return redirect('dashboard:dashboard_index')
        elif user is not None and user.is_manager:
            login(request, user)
            messages.success(request, f"Successfully Login as Manager!")
            return redirect('dashboard:dashboard_index')
        elif user is not None and user.is_staff:
            login(request, user)
            messages.success(request, f"Successfully Login as Staff!")
            return redirect('dashboard:dashboard_index')
        elif user is not None and user.is_customer:
            login(request, user)
            return redirect('store:index')
        else:
            return HttpResponse('404')
    return render(request, 'login.html')
```

Figure 34: Log-in page

```python
def userLogout(request):
    logout(request)
    messages.success(request, f"Successfully Logout!")
    return redirect('account:login')
```

Figure 35: Logout sample

Add to cart system

```python
def add_to_cart(request, pk):
    if request.user.is_authenticated:
        item = get_object_or_404(Products, pk=pk)
        order_item = Cart.objects.get_or_create(item=item, user=request.user, purchased=False)
        order_qs = Order.objects.filter(user=request.user, ordered=False)
        if order_qs.exists():
            order = order_qs[0]
            if order.orderitems.filter(item=item).exists():
                quantity = request.POST.get('quantity')
                if quantity:
                    order_item[0].quantity += int(quantity)
                else:
                    order_item[0].quantity += 1
                    order_item[0].save()
                    return redirect('store:index')
            else:

                order.orderitems.add(order_item[0])
                return redirect('store:index')
        else:
            order = Order(user=request.user)
            order.save()
            order.orderitems.add(order_item[0])
            message = f"Product added to your cart."
            SendNotification(request.user, message)

            return redirect('store:index')
    else:
        return redirect('account:login')
```

Figure 36: Add to cart system

payment\views.py × product_list.html × dashboard\forms.py × dashboard\views.py × store\forms.py ×

```python
def cart_view(request):
    if request.user.is_authenticated:
        carts = Cart.objects.filter(user=request.user, purchased=False)
        orders = Order.objects.filter(user=request.user, ordered=False)
        if carts.exists() and orders.exists():
            order = orders[0]
            coupon_form = CouponCodeForm(request.POST)
            if coupon_form.is_valid():
                current_time = timezone.now()
                code = coupon_form.cleaned_data.get('code')
                coupon_obj = Coupon.objects.get(code=code, active=True)
                if coupon_obj.valid_to >= current_time:
                    get_discount = (coupon_obj.discount / 100) * order.get_totals()
                    total_price_after_discount = order.get_totals() - get_discount
                    request.session['discount_total'] = total_price_after_discount
                    request.session['coupon_code'] = code
                    return redirect('order:cart')
            total_price_after_discount = request.session.get('discount_total')
            coupon_code = request.session.get('coupon_code')
            context = {
                'carts': carts,
                'order': order,
                'coupon_form': coupon_form,
                'total_price_after_discount': total_price_after_discount
            }

            return render(request, 'store/cart.html', context)
        return HttpResponseRedirect(request.META["HTTP_REFERER"])
    else:
        return redirect('account:login')
```

Figure 37: Cart View system

payment\views.py | product_list.html | dashboard\forms.py | dashboard\views.py | store\forms.py | accour

```python
    def post(self, request, *args, **kwargs):
        saved_address = BillingAddress.objects.get_or_create(user=request.user or None)
        saved_address = saved_address[0]
        form = BillingAddressForm(instance=saved_address)
        payment_obj = Request_for_food.objects.filter(user=request.user)[0]
        payment_form = PaymentMethodForm(instance=payment_obj)
        if request.method == 'post' or request.method == 'POST':
            form = BillingAddressForm(request.POST, instance=saved_address)
            pay_form = PaymentMethodForm(request.POST, instance=payment_obj)
        if form.is_valid() and pay_form.is_valid():
            form.save()
            pay_method = pay_form.save()
        if not saved_address.is_fully_filled():
            messages.warning(request, f"Please complete your billing address!")
            return redirect('checkout')
        if not request.user.profile.is_fully_filled():
            messages.warning(request, "Please complete your profile details!")
            return redirect('account:profile')
            # cash on delivery payment process
        if pay_method.payment_method == 'Cash on Delivery':
            req_qs = Request_for_food.objects.filter(user=request.user)
            req = req_qs[0]
            req.reqId = req.id
            req.paymentId = pay_method.payment_method
            req.save()
            cart_items = Cart_for_req.objects.filter(user=request.user, purchased=False)
            for item in cart_items:
                item.purchased = True
                item.save()
            message = f"Paid successfully!"
            messages.success(request, f"Paid successfully!")
            SendNotification(request.user, message)
            return redirect('store:index')
```

Figure 38: Payment System (Cash on Delivery)

10.2 Possible Problem Breakdown

The whole project needs to be divided into smaller tasks to compose the project development smooth & more effortless. The potential breakdowns of the planned project are:

- ✓ Database design & analysis
- ✓ Dashboard manipulation development
- ✓ User-panel manipulation

Analysis and database design

- ✓ Analyze and select specifications
- ✓ Collect data to deal with
- ✓ Establish the collected data
- ✓ Run Django Admin panel
- ✓ Create models for Django Admin panel (database)

Dashboard development

- ✓ Registration and login system for different categories of role
- ✓ Design the necessary pages
- ✓ Develop Admin manipulation system
- ✓ Develop Staff manipulation systems
- ✓ Develop Manager manipulation systems

User panel development

- ✓ Frontend design
- ✓ Registration and login system
- ✓ Product management system
- ✓ Add to cart system
- ✓ Ordering system
- ✓ Payment system
- ✓ Wishlist system
- ✓ Special Requesting system

CHAPTER 11

Testing

11.1 Test plan acceptance

Test plan acceptance plays an important role in creating a successful software test. All the tests should be prepared during the analysis section. This arrangement needs to be adopted by the user and the developer. The test plan for the system will assure related test specifications & user acceptance criteria. 2 types of tests are applied here:

- Functional test

The functional tests are separated into 3 types of tests. Those are given below:

- Unit test
  - Validation of form field for the input fields
  - Admin approval filter
  - Approved, pending, and rejected from the admin
- Module testing
  - Submit the form of registration & log in without data
  - Register with the unreasonable data
  - Ordering with the appropriate billing address
- Integration Testing
  - Log in with the valid data
  - Order placement correctly
- Nonfunctional testing

  I am going to execute 4 types of non-functional testing in this system, those are given below:
- Acceptance testing
  - Personal information updating from the different user perspectives
- Security Testing
  - Dashboard and page access role-wise
  - Login attempts with the invalid data

- Accessibility Testing
  - User-friendliness testing
  - Contrast & color trialing by the color-blind user
- Usability Testing
  - Admin-panel trailing
  - Examining with customers

11.2 Test case

The test acceptance plans need to be planned for the exam after finalization. These are test cases of the VOFS system

Unit test –test case:

Test case name	**Unit test**		
Test class			
Test description			
Data source	Test setup	predicted output	original output

Module test –test case:

Test case name	**Unit test**		
Test class			
Test description			
Data source	Test setup	predicted output	original output

Integration test –test case:

Test case name	**Unit test**		
Test class			
Test description			
Data source	Test setup	predicted output	original output

11.3 Unit testing

Unit test-1

Test case

Test case name	**Unit test**
Test class	User registration
Test description	User Email or username verification
Data source	Test setup predicted output original output
User entry	Submit form an error message need validation

Table 28: Unit test case-1

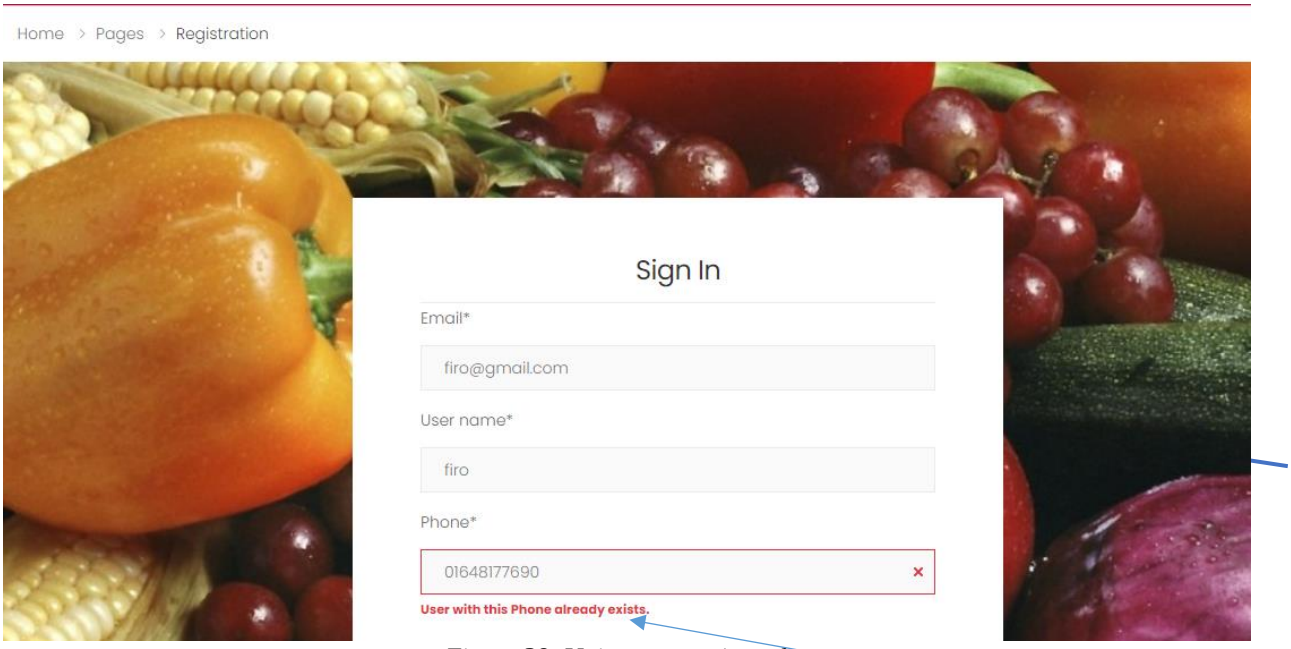


Figure 39: Unit test case-1 results

Unit test-2

Staff registration

Test case name	**Unit test**		
Test class	Staff registration		
Test description	User Email Username and phone number		
Data source	Test setup	predicted result	original result
User entry	Input all data and submit the form	Got an error message	Need valid information which is matched in the database

Table 29: Unit test case-2

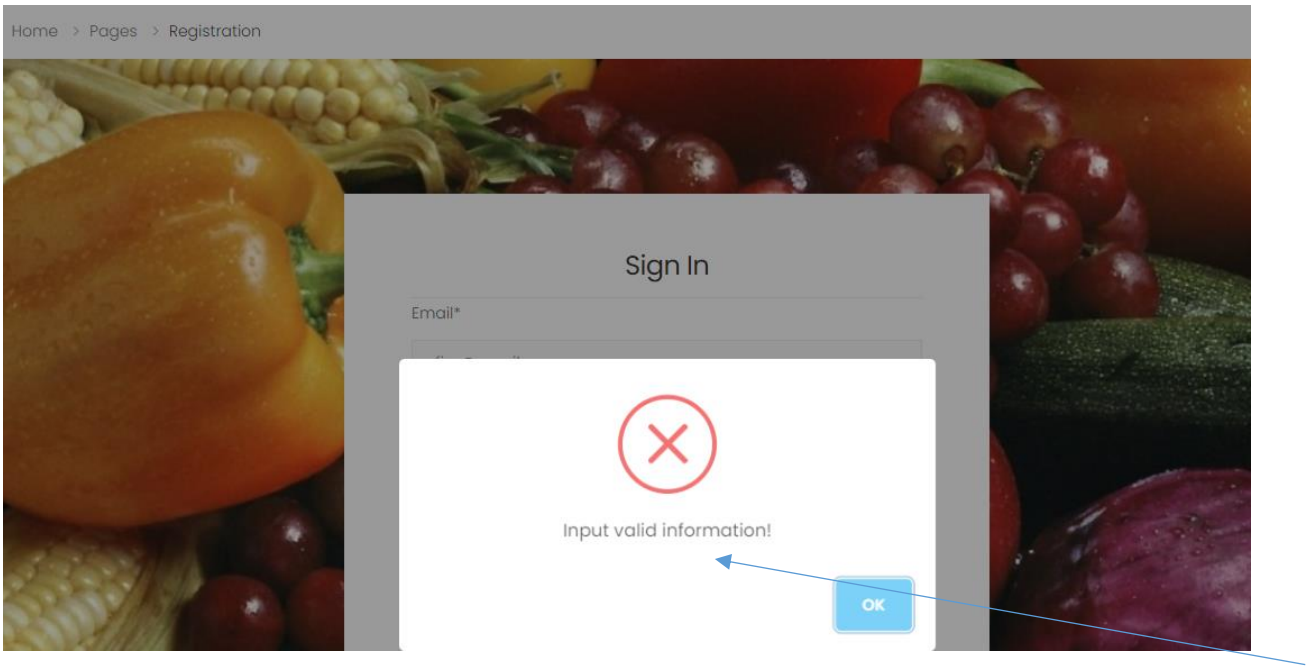


Figure 40: Unit test case-2 results

Use test-3

User login

Test case name	**Unit test**		
Test class	User login		
Test description	User registered email and password		
Data source	Test setup	predicted result	original result
User entry	Wrong registered data and submit the form	Got invalid data message	Need valid registered data

Table 30: Unit test case-3

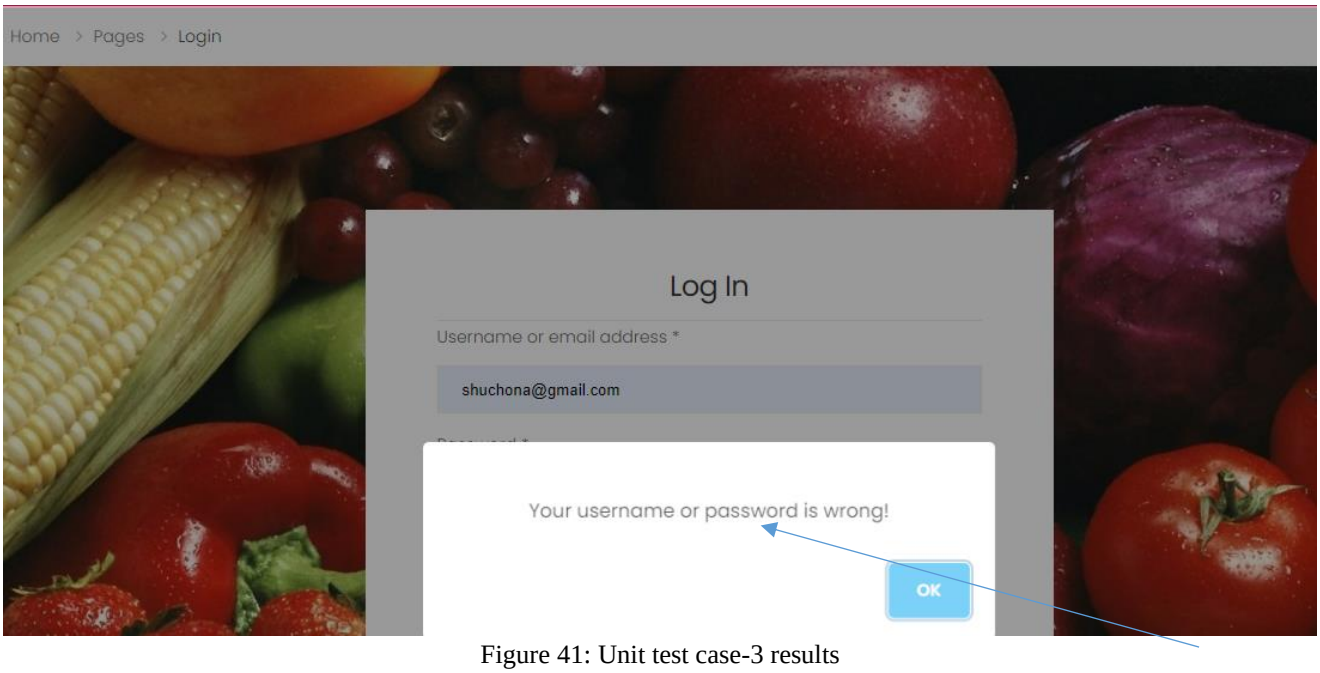


Figure 41: Unit test case-3 results

11.4 Module test

Module test-1

Test case name	Unit test		
Test class	Food Ordering Process		
Test description	Customers need to fill up the billing address form		
Data source	Test setup	Expected result	Actual result
User entry	Any blank field is not acceptable	Got error message	Need proper data

Table 31: Module test-1

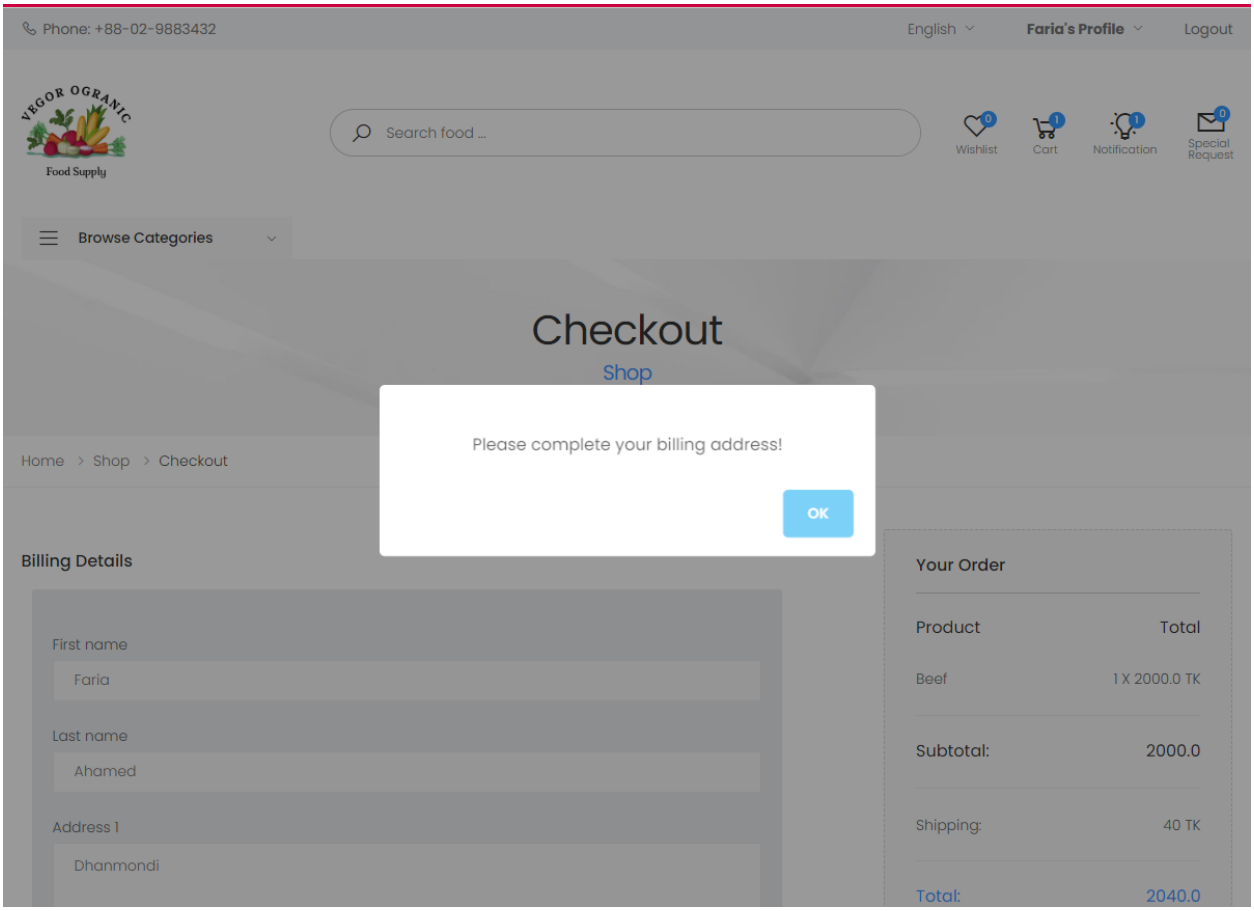


Figure 42: Module test-1 result

Module test-2

Test case name	Unit test		
Test class	Staff		
Test description	Need to provide all data about a product (food)		
Data source	Test setup	Expected result	Actual result
User entry	Blank field is not acceptable	Got invalid message	Need data

Table 32: Module test-2

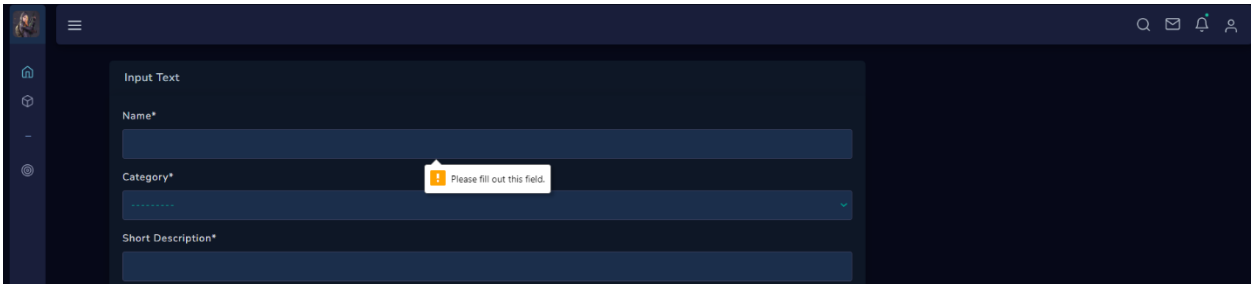


Figure 43: Module test-2 result

11.5 Integration testing

Test case name	**Unit test**		
Test class	User		
Test description	1. Auth or Login Controller 2. Redirect If verified. Prosperous login attempts & redirect to dashboard		
Data source	Test setup	Predicted output	Actual result
User entry	1. Go to the login page 2. Provide valid information 3. Press on the login	The user should be authenticated	It shows that email The address is invalid.

Table 33: Integration testing

User login integrate test

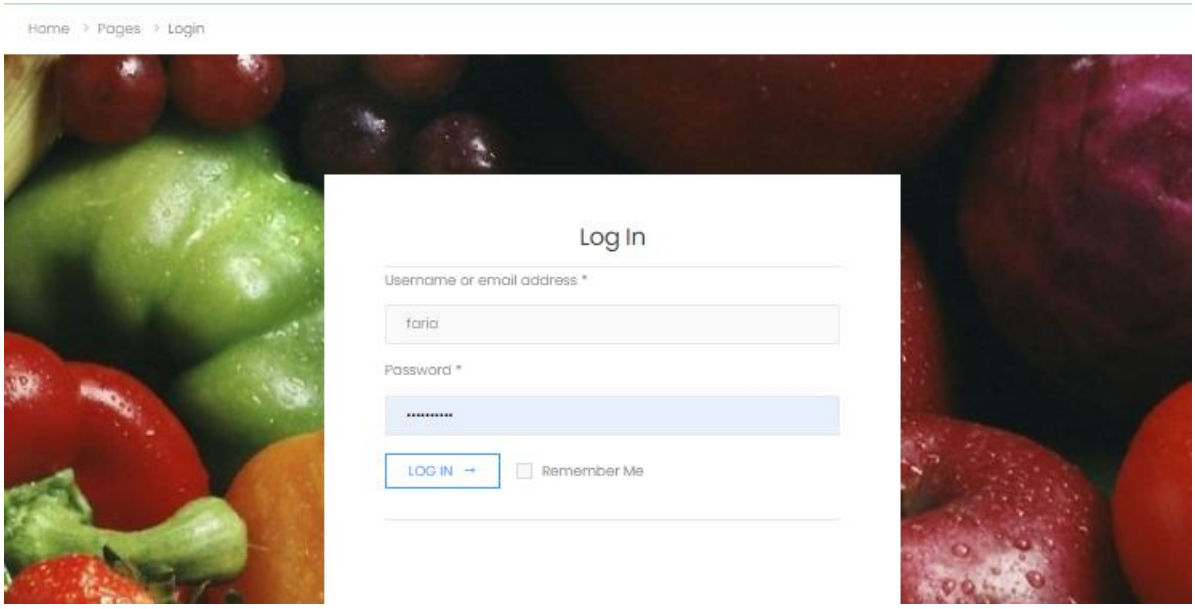


Figure 44: User login integrate test

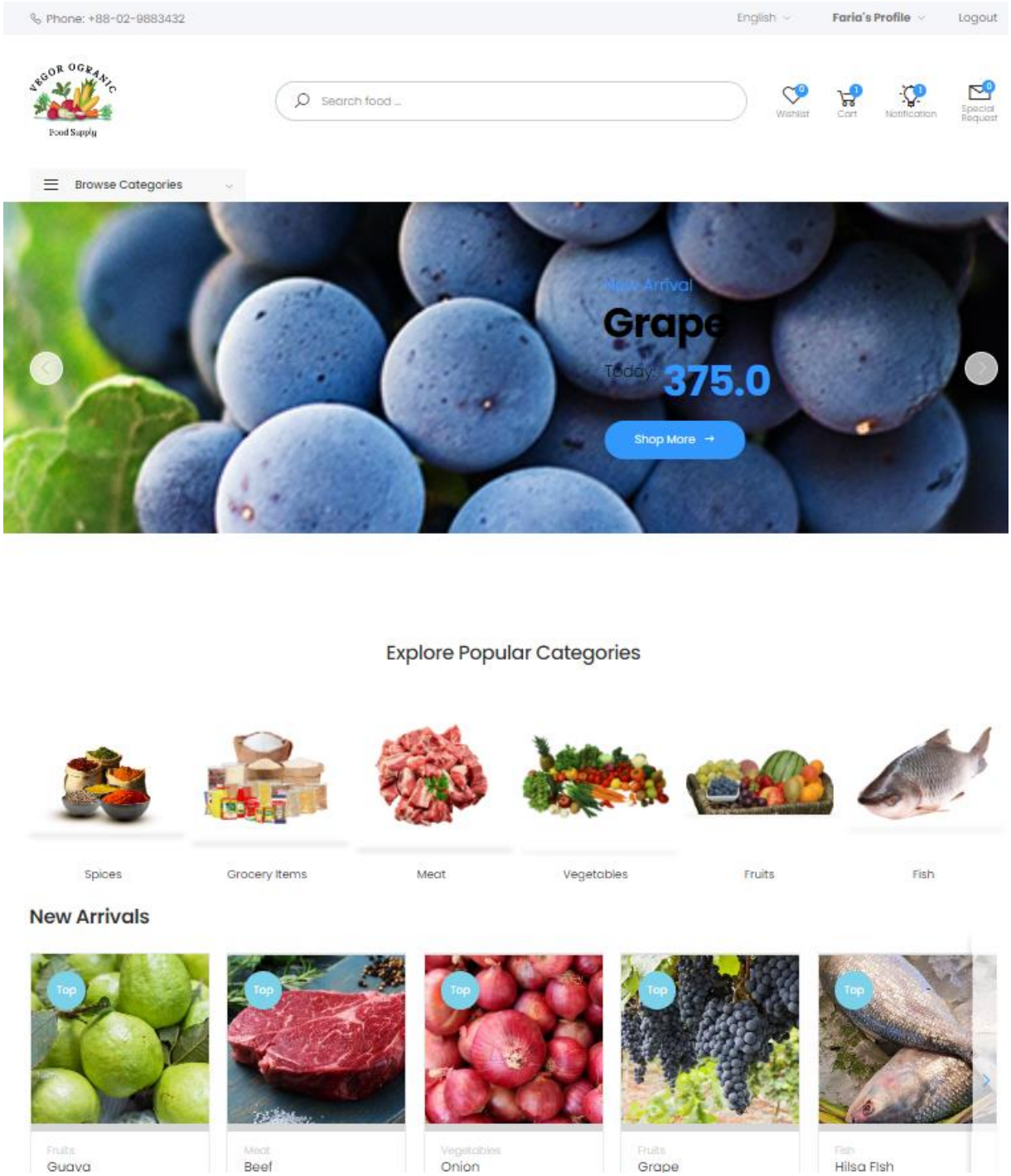


Figure 45: User login integrates test results

11.6 Security testing

Test case name	**Unit test**		
Test class	Customer		
Test description	1. Auth\Login Controller		
Data source	Test setup	Predicted output	Original output
User entry	Invalid auth attempt security testing	Should not be logged in and a message Should show.	The result is as Expected.

Table 34: User security test

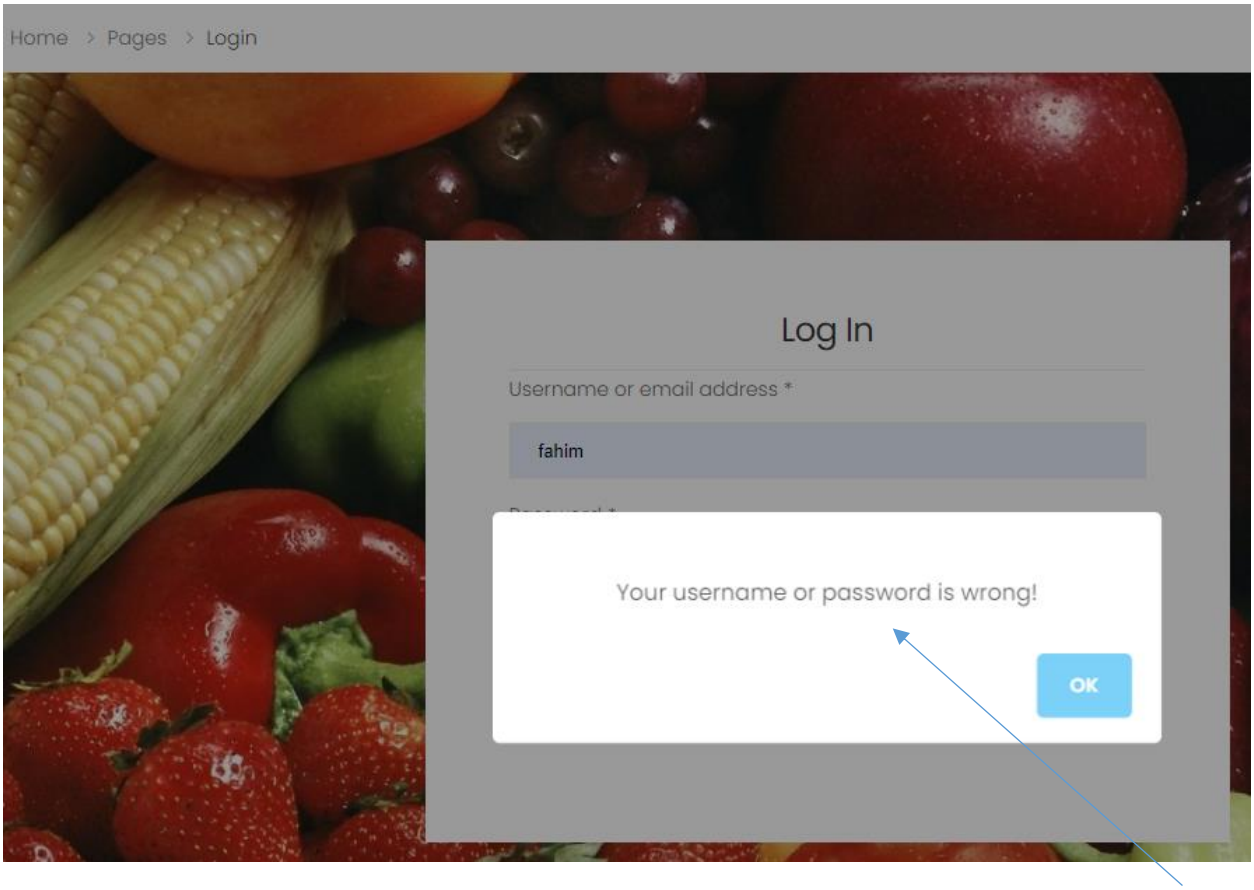


Figure 46: User security test results

11.7 Accessibility testing

Accessibility testing case-1

Test case name	**Unit test**		
Test class	Customer		
Test description	Customer Controller		
Data source	Test setup	Predicted output	Actual result
User entry	Customers use the system	No problem occurs during the use.	The owner did not face any difficulties.

Table 35: Customer controller accessibility test

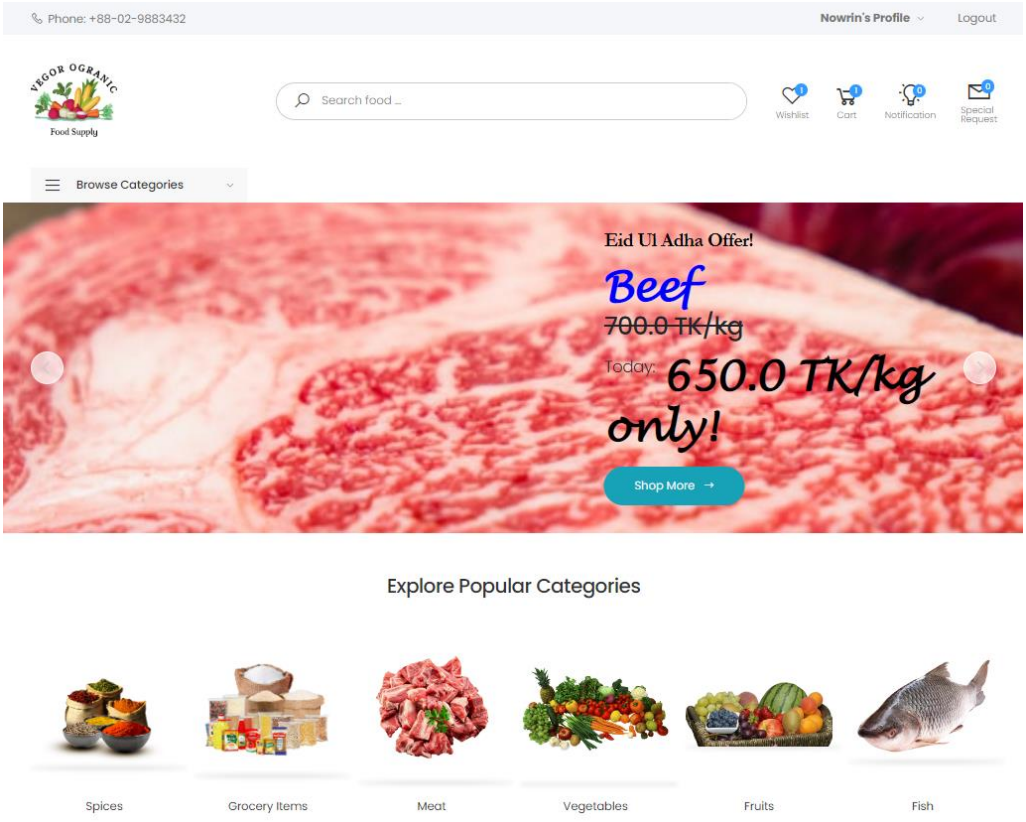


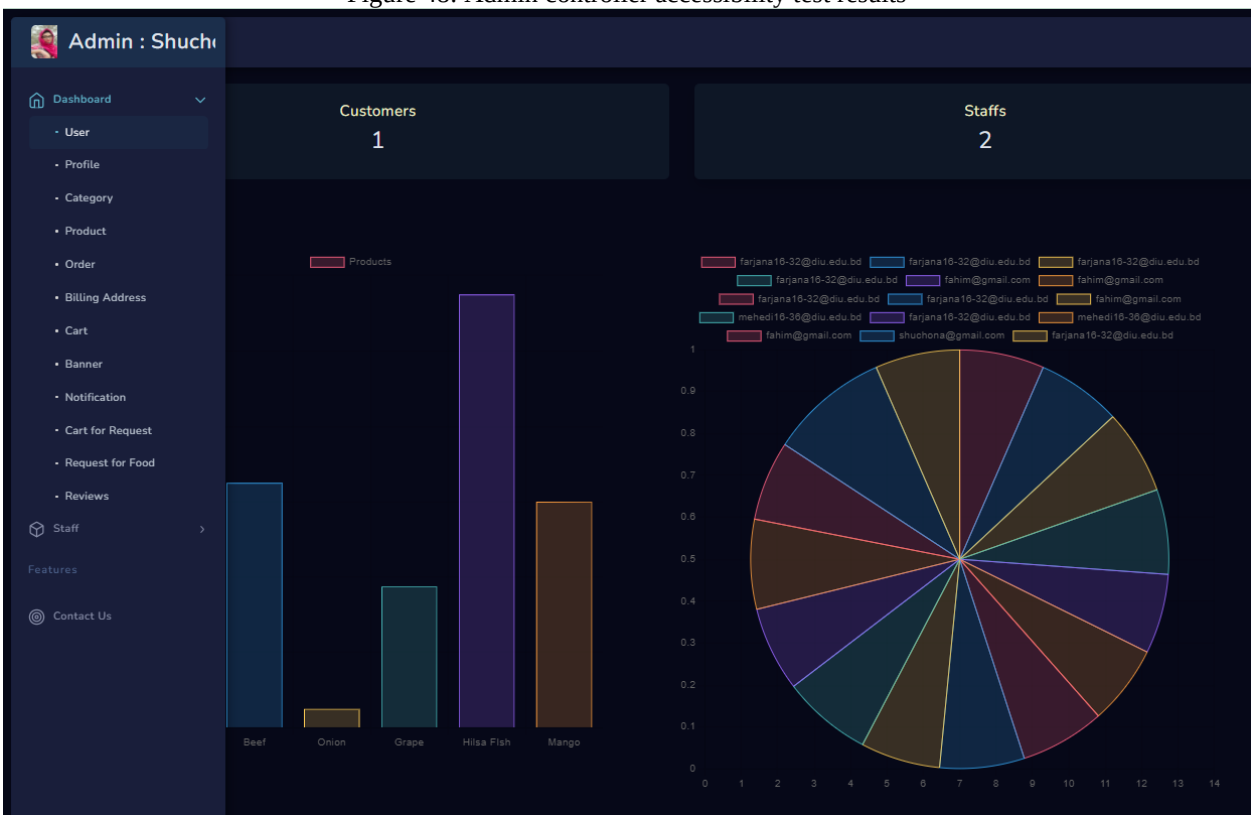
Figure 47: Customer controller accessibility test results

Accessibility testing case-2

Test case name	**Unit test**		
Test class	Admin Controller		
Test description	User-friendly testing		
Data source	Test setup	Expected result	Actual result
User entry	Giving an admin to use the system	The system is user-friendly.	The result is expected.

Table 36: Admin controller accessibility test-2

Figure 48: Admin controller accessibility test results



CHAPTER 12

Implementation

12.1 Training

The training is needed to familiarize users with advanced systems. When this is operational, the team members must assure that users or operators are sufficient to manage the activity throughout the project. A table describing the training method is given below:

SL	User	Training Scope	Time	Comment
1	User	Registration as user	50 Minutes	Users are perceiving the methods accordingly.
2	Admin	Add a database of external user	1.5 Hours	Admin can smoothly figure out the operation.

Table 37: User training

12.2 Implementation Scheme

Big Bang

The big bang shuts down a current system & immediately enables the new project to work. This action is much quicker than any other & uses the new project instantly after the testing is over. Loss of the old system can lead to data exchange and instability as data can be lost and new systems can crash. It has been included in the human resource consideration of a single site.

12.3 Scaling

My plan is will give my project to the responsible people of Palal Group so that they can use it for their own and their particular customers.

12.4 Load Balancing

The load balancing implies the project has been enhanced against the impacts of clients. User hits hint how many users are working on the system at the same time and how long the system lasts, it refers to the load equalizer & load balance. And it breaks the capacity between various servers so that the project can work faster. Afterward, there are 4 types of users and lots of hits per day, a right load balancing action should be created.

CHAPTER 13

Critical Appraisal and evaluation

13.1 Objective could be met

The objectives that were originally announced are given down:

- Authentication system for user
- Food Category Manipulation
- User & admin panel manipulation
- Food pre-ordering system
- Special ordering system
- Payment system
- Food Price Prediction System
- Food Availability and Stock Management
- Report Observing System

Achievement rate and others

The registration and login system are successfully implemented for all types of users (Customer, admin, staff, manager). Different types of users can manage by Admin. Customers can successfully order products by viewing their availability, pictures, videos, rate reviews, and details with payment. Customers can request food as their want. I have implemented the Django Valid Authentication System for authentication on users' systems. My progress rate is 100 percent out of 100 against each goal. The system can be made better by allowing all customers around our country to use it though it will transfer to the e-commerce website.

13.2 Objectives that don’t meet of touched

As I make this system for some elite customers so there is no need for a location tracking system. But if I want to change the specific customers to all people of our country, I will have to make a location tracking system.

The reasons why it could be touch

The location tracking system is important if the customer order a product with cash on delivery system, but I am not going to implement this system now because my system is web-based and the location-tracking system is usable for the mobile app system.

What could have been done

For ordering products, customers are required to fulfill their billing address properly and they have to pay an extra fixed delivery charge for their purchase as there is no location tracking system due to a web-based system.

CHAPTER 14

Lessons Learned

14.1 Pre-project-Review-closing

Vegor Organic Food Supply (VOFS), from the very beginning to create a web-based application I have to pursue a pre-planned format such as system proposal submission, system 1st defense, development, & documentation. So, the main concern of this project is making an online food ordering system for Vegor Agri Complex Ltd. as they don't have any automated online system.

14.2 What I have learned

To create a format, I have prepared many important things. I figured out how to manage a format in different sections like design, layout, & database area. Additionally, I have upgraded my enterprise management skills & testing ability (black & white boxes, units & acceptability, usability & accessibility) which will encourage me in my destiny. Additionally, it enhances my programming skills, e.g., creating an application programmable interface (API) with appropriate authentication & security tests. Furthermore, I learned Python and Django Framework. This initiative allows me to gather a lot of basic information that is needed to improve my quality of life.

14.3 The problem I face

During the life cycle of the project, I encountered many problems and challenges. Since I have followed the clever DSDM method that ensures recurrent development, I have to face recurring problems and overcome them. This means that a problem occurs in the one-time box which is solved in another time box and works. One of the main problems was I couldn't fix the payment error; I couldn't install the SSLCommerz payment system in my project. On the other hand, I was facing a problem while creating a different dashboard for the admin panel, staff, and manager. I solve it by searching for the solution on the internet and cooperation with my friends & my supervisor.

14.4 What solution Occurred

Whenever I faced a problem, I try to find a way to solve my problem. Sometimes, it takes 2-3 days to fix an error as I recently learned the Django framework. I have faced some big and small problems, one of the big problems was the installation of SSLCommerz in my system. I fixed the problem by installing the old version of python as SSLCommerz does not support on latest version of python. Furthermore, I fixed the dashboard problem by adding a different template and making a different app in Django. Sometimes, while creating models, I couldn't modify them when I wanted to. It was always showing migrations problems. But now I properly solve all the problems by the grace of Almighty Allah. To manage the time box, I perform very seriously and followed the time limit rigorously.

CHAPTER 15

Conclusion

15.1 Summary of the project

Vegor Organic Food Supply System is an online platform and a solution for Palal Group as they want to integrate an online system for their elite customers and own. All the information will be kept safely and customers will be ordering food as their wish, they will allow requesting food in a huge quantity. The system is built with the support of HTML, CSS, JavaScript, jQuery, SQLite, and Python Django Framework. All relevant diagrams & charts are given in the document.

15.2 Goal of the project

The goal of the project was to make an online-based solution for the elite customers of Vegor Agri Complex Ltd. The main objective of the project is-

- Way of income source.
- Allow fresh food ordering online.
- All information keeping safely.
- Allow customers to request huge quantity food.
- Allow customers to pay online.

15.3 Success of the project

The development goals of the system are achieved as the targets are met against the targets. Customers can easily order food and request food on any occasion. Moreover, staff can manage customers, pricing, add product information, etc. As it meets its requirements and object successfully, I can say the project is successfully done.

15.4 What I have done in the documentation

I have accepted all the work required for documentation from the beginning of documentation. I have managed time table, countless images, analyses, etc. The document also has a variety of goals & they have been appropriately addressed. The entire information required to finish a project is included in this document.

15.5 Value of the project

The technology will consistently beat over hand-operated labor & methods. The food ordering system will enhance the experience of ordering food and paying money online. This project has an important value in the marketplace for organizations, where they can use it for their elite customers to sell their products; moreover, this system can be turned into an e-commerce website for all people of our country and we know how people are interested in the e-commerce system. On the other hand, this project has been done with Python and Django frameworks which are now top growing language-framework. The experience gained from the development of this system during this session will benefit me in my qualified job.

15.6 My Experience

Finally, I have gained enough experience all over this project. Though I had a lot of complications & I dealt with those obstacles that brought me enough experience. I learned how to maintain a complete project and how to deal with all the criteria of the project in a limited time which awarded me a great maturity.

REFERENCE

[1] *2021 Dell OptiPlex 7090 SFF Desktop - Intel Core i7 10th Gen*. (n.d.). Ubuy. Retrieved April 6, 2022, from https://www.ubuy.com.bd/en/

[2] *Shwapno*. (n.d.). Shawpno. Retrieved May 3, 2022, from https://www.shwapno.com/

[3] Mahmud, N. (2020, February 5). *Dhaka tribune*. Dhaka Tribune. Retrieved May 6, 2022, from https://archive.dhakatribune.com/bangladesh/2020/02/05/the-high-cost-of-consuming-adulterated-food

[4] *Desktop Computer Set Vector In Perspective View Free Vector*. (n.d.). [Illustration]. Pngtree. https://pngtree.com/so/computer

[5] *RAD*. (n.d.). [Graph]. NIX United. https://nix-united.com/blog/the-ultimate-guide-to-rapid-application-development/

[6] Atlas, A. (2019, November). *Foundation of Agile Part I* (No. 18F). Alan Atlas and Alan Brouilette. https://digital.gov/pdf/Foundations-of-Agile-Part-I.pdf

[7] Siau & Wang, K. Y. (2007). Cognitive evaluation of information modeling methods. *Cognitive Evaluation of Information Modeling Methods*, 458. https://doi.org/10.1016/j.infsof.2006.07.001

[8] Indeed Editorial Team. (2021, October 8). *Dynamic Systems Development Method: Advantages and Disadvantages*. Indeed. Retrieved May 10, 2022, from https://www.indeed.com/career-advice/career-development/dynamic-systems-development-method

[9] Martin, M. (2022, July 15). *What is RAD Model? Phases, Advantages and Disadvantages*. Guru99. Retrieved July 22, 2022, from https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html

[10] *Chaldal*. (n.d.). Chaldal. Retrieved July 25, 2022, from https://chaldal.com/

[11] *Walmart*. (n.d.). Walmart. Retrieved July 25, 2022, from https://www.walmart.com/

[12] *Palal Group*. (n.d.). Palal Group. Retrieved February 1, 2022, from https://palal-minerva.com/