

ES-Cart

Final Year Project

Session 2019-2023

A project submitted in partial fulfillment of the degree of

BS in Information Technology



Department of Information Technology

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	[<input checked="" type="checkbox"/>] Development [<input type="checkbox"/>] Research [<input type="checkbox"/>] R&D			
Area of specialization				
FYP ID	FYP-BITM-F22-008			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	Bitm-f19-024	Muhammad Arsalan	bitm-f19-024@superior.edu.pk	
(ii)	Bitm-f19-026	Salman Ahmad	bitm-f19-026@superior.edu.pk	
(iii)	Bitm-f19-048	Saif ullah Sharif	bitm-f19-048@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I _____ S/D of _____, group leader of FYP under registration no _____ at Information Technology Department, The Superior University Lahore. I declare that my FYP report is checked by my supervisor.

Date: _____ Name of Group Leader: M.Arsalan

Signature: _____

Name of Supervisor: Javaid Iqbal

Co-Supervisor: Mr.Gohar Mumtaz

Designation: Lecturer

Designation: Senior Lecturer

Signature: _____

Signature: _____

HoD: Dr. Asad Ali Naqvi

Signature: _____

Project Report

ES-Cart

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

This project is whole heartedly dedicated to our parents, who have been our source of inspirations and give us strength when we thought of giving up, who continually provide their moral, spiritual, emotional and financial support.

To our friends, mentor and classmates who shared their words of advice and encouragement to finish this project. And especially we dedicated to this project to Allah Almighty. Thank you for guidance, power of mind, protection and skill and for giving us a healthy life. All of these we offer to you.

Acknowledgements

We are three students of Information Technology in Superior University Lahore, preparing a FYP report on “ES-Cart reporting and Web App”. Primarily we would thank God for being able to complete this report with success. Then we would like to thank our supervisor “Mr. Gohar Mumtaz” whose valuable has been the ones that helped us patch this report and make it full proof success his suggestions and his instruction has served as the major contributor towards the completion of the report.

Then we like to thank our friends who have helped us with their valuable suggestions and guidance has been helpful in various phases of completion of the report.

Executive Summary

We'll begin with motivations to make a multi-seller online store. Multi-dealer stores offer an entire space of possible results single trader stores can't match. With a multi-seller store, you can make a site like ES-Cart where anybody can sell anything or a specialty multimer chant site zeroed in on a particular item class like workmanship or gadgets. The possibilities are practically wearisome. By welcoming various traders, stores can offer things they can't offer themselves. Various sellers can likewise help increment traffic to the store and decrease store the executive's time. Moreover, as the head chief, you don't have to keep a stock. You can go about as a mediator for providers rather being a provider yourself. You can likewise bring in cash from charging merchants an expense for selling on your site. This alternative opens up another wellspring of income for your business. There is additionally an abundance of advantages for dealers. Merchants don't need to experience the pressure of opening another store. They can likewise exploit existing site traffic to improve perceivability. Selling items on a multi-seller webpage implies the majority of the advantages of having an online store without the overhead problem. We turned out a portion of the explanations behind building a multi-seller store. Presently we'll get into the stuff to assemble an effective one. On the off chance that you have experience dealing with a solitary merchant store, you may think the change to a multi-seller store will be simple. Try not to think little of the distinction between these two sorts of stores, however. There are a few essential contrasts to running a store with numerous sellers.

Table of Contents

Plagiarism Free Certificate	ii
Dedication	v
Acknowledgements.....	vi
Executive Summary.....	vii
Table of Contents.....	viii
List of Figures	xi
List of Tables	xii
Chapter 1.....	13
Introduction	13
1.1. Background.....	14
1.2. Motivations and Challenges.....	14
1.3. Goals and Objectives.....	15
1.4. Literature Review/Existing Solutions	15
1.5. Gap Analysis	15
1.6. Proposed Solution	16
1.7. Project Plan	16
1.7.1. Work Breakdown Structure.....	17
1.7.2. Roles & Responsibility Matrix.....	17
1.7.3. Gantt Chart	18
1.8. Report Outline.....	19
Chapter 2.....	20
Software Requirement Specifications	20
2.1. Introduction.....	21
2.1.1. Purpose.....	21
2.1.2. Document Conventions	21
2.1.3. Intended Audience and Reading Suggestions	21
User of project can read this document to get overall concept of this project.	22
2.1.4. Product Scope.....	23
2.1.5. References	23
2.2. Overall Description.....	23
2.2.1. Product Perspective.....	23
2.2.2. User Classes and Characteristics	24
2.2.3. Operating Environment	24
2.2.4. Design and Implementation Constraints.....	24
2.2.5. Assumptions and Dependencies	24
2.3. External Interface Requirements	25
2.3.1. User Interfaces.....	25
User Perspective	25
2.3.2. Hardware Interfaces	25
2.3.3. Software Interfaces	25
2.3.4. Communications Interfaces.....	26
2.4. System Features	26

2.4.1.	Register page	27
2.4.1.1.	Description and Priority	27
2.4.1.2.	Stimulus/Response Sequences	27
2.4.1.3.	Functional Requirements.....	27
2.4.2.	Login page.....	27
2.4.2.1.	Description and Priority	27
2.4.2.2.	Stimulus/Response Sequences	27
2.4.2.3.	Functional Requirements.....	27
2.4.3.1.	Description and Priority	28
2.4.3.2.	Stimulus/Response Sequences	28
2.4.3.3.	Functional Requirements.....	28
2.4.4.	Wishlist Page.....	28
2.4.4.1.	Description and Priority	28
2.4.4.2.	Stimulus/Response Sequences	28
2.4.4.3.	Functional Requirements.....	28
2.4.5.	Cart Page.....	29
2.4.5.1.	Description and Priority	29
2.4.5.2.	Stimulus/Response Sequences	29
2.4.5.3.	Functional Requirements.....	29
2.5.	Nonfunctional Requirements.....	29
2.5.1.	Performance Requirements	29
2.5.2.	Safety Requirements	30
2.5.3.	Security Requirements	30
2.5.4.	Availability Requirements.....	30
2.5.5.	Reliability Requirements.....	30
2.5.6.	Maintainability/Supportability Requirements	31
2.5.7.	Portability Requirements.....	31
2.5.8.	Efficiency Requirements.....	31
2.6.	Domain Requirements	31
Chapter 3.....		32
Use Case Analysis.....		32
3.1.	Use Case Model.....	33
3.2.	Use Cases Description	34
Chapter 4.....		35
System Design.....		35
4.1.	Architecture Diagram	36
4.2.	Domain Model.....	37
4.3.	Entity Relationship Diagram with data dictionary	38
4.4.	Class Diagram	39
4.5.	Sequence/ Collaboration Diagram	40
4.6.	Operation contracts	40
4.7.	Activity Diagram	41
4.8.	State Transition Diagram.....	42
4.9.	Component Diagram	43

4.10.	Deployment Diagram.....	44
4.11.	Data Flow diagram.....	45
Chapter 5.....		47
Implementation		47
5.1.	Important Flow Control/Pseudo codes.....	48
5.2.	Components, Libraries, Web Services and stubs	49
5.3.	Deployment Environment.....	49
5.4.	Tools and Techniques.....	49
5.5.	Best Practices / Coding Standards.....	50
5.6.	Version Control	50
Chapter 6.....		51
Testing and Evaluation.....		51
6.1.	Use Case Testing.....	52
6.2.	Equivalence partitioning	53
6.3.	Boundary value analysis.....	55
6.4.	Data flow testing	56
6.5.	Unit testing.....	58
6.6.	Performance testing.....	60
6.7.	Stress Testing	60
Chapter 7.....		61
Summary, Conclusion and Future Enhancements.....		61
7.1.	Project Summary	62
7.2.	Achievements and Improvements	63
7.3.	Critical Review	2
7.3.	Lessons Learnt.....	2
7.4.	Future Enhancements/Recommendations	3
Reference and Bibliography.....		4

List of Figures

1.1	Work breakdown structure	15
1.2	Gantt chart	17
3.1	Use Case Model	29
4.1	Architecture Diagram	32
4.2	Domain Model	33
4.3	Entity Relationship Diagram with data dictionary	33
4.4	Class Diagram	34
4.5	Sequence/ Collaboration Diagram	34
4.6	Activity Diagram	36
4.7	State Transition Diagram	37
4.8	Component Diagram	37
4.9	Deployment Diagram	38
4.10	Data flow diagram	38

List of Tables

1.1	Role and responsibility matrix	16
2.1	Intended Audience and Reading Suggestions	19

Chapter 1

Introduction

Chapter 1: Introduction

ES-Cart is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a general-purpose e-commerce store where product like clothes can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online shopping for clothes and etc. A web-based application will be the end product of our project in which user will have the option to purchase multiple products on cash and on installments by just clicking on his/her mobile phone. An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e-mail notification is sent to the customer as soon as the order is placed.

1.1. Background

There is a lot of competition among multiple e-commerce sites. When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. Also, users are not sure about the brands or the actual products they want to purchase. They have a very broad idea about what they want to buy. Many customers nowadays search for their products on Google rather than visiting specific e-commerce sites. They believe that Google will take them to the e-commerce sites that have their product.

1.2. Motivations and Challenges

One of the key ecommerce challenges that businesses face is the integration of their ecommerce arms into the company's overall hierarchical structure. For that to happen,

businesses must adopt solutions that focus primarily on data. This is necessary for managing KPIs and tracking average order values across multiple channels, with ecommerce being one of them. A setup is required for capturing the data and then applying the findings to get appropriate results. This can help businesses make more informed long-term decisions and avoid other issues in ecommerce management.

1.3. Goals and Objectives

Our purpose is to design and develop a web-based application will be the end product of our project in which user will have the option to purchase multiple products on cash and on installments by just clicking on his/her mobile phone. The web-based application will let the user decide whether buy a product on cash and on installment according to his/her budget by calculating his installment plan and his price range it will automatically suggest products in his/her price range. This will not only help the user but also useful seller to predict whether which product is most selling product and which brand is top selling brand also they can decide hot products and categories buy watching their sale and purchased products.

1.4. Literature Review/Existing Solutions

The systems are currently running with multiple features but all these systems did not deal with installment purchases. But in our E-commerce system we deal in both cash on delivery online purchases and also customer can buy products on installment by submitting installment request form. Also, there is another feature that our customer can be our partner by sending us a merchant request and register his/her business with us.

1.5. Gap Analysis

There are some common problems that every ecommerce system has to face. When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. Also, users are not sure about the brands or the actual products they want to purchase. They have a very broad idea about what they want to buy. Many customers nowadays search for their

products on Google rather than visiting specific e-commerce sites. They believe that Google will take them to the e-commerce sites that have their product.

The purpose of any e-commerce website is to help customers narrow down their broad ideas and enable them to finalize the products they want to purchase.

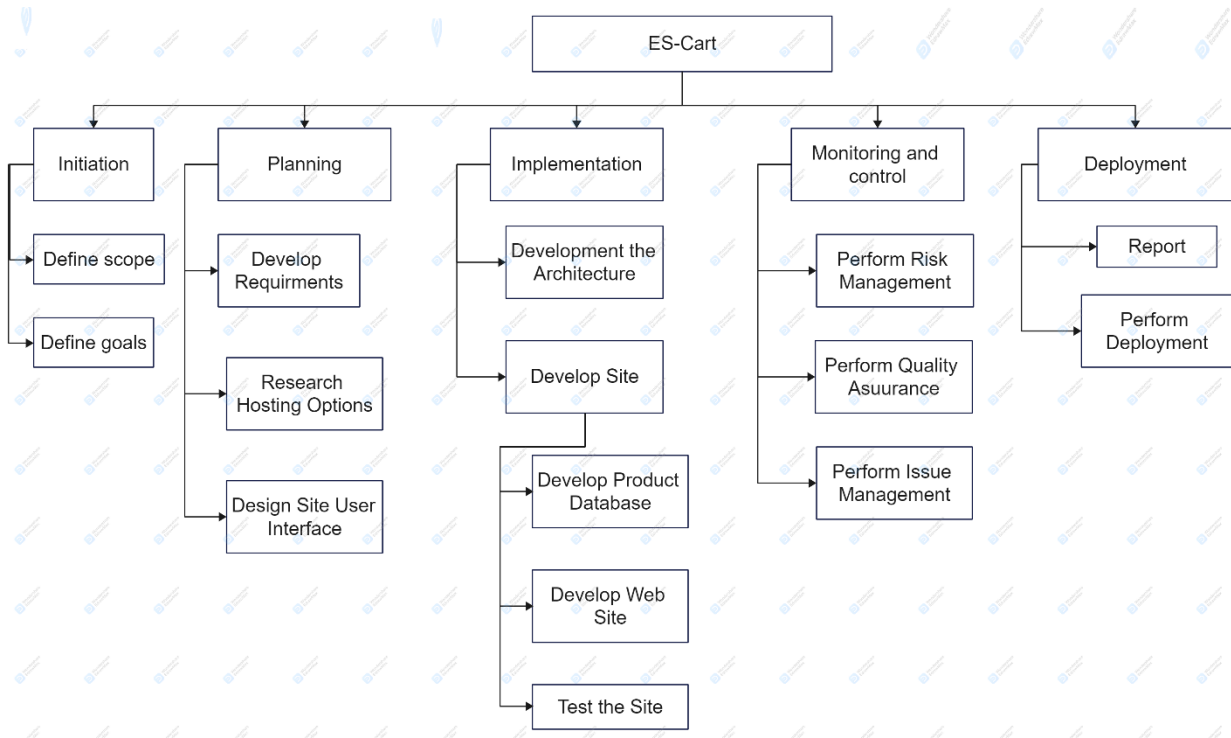
1.6. Proposed Solution

A web-based application will be the end product of our project in which user will have the option to purchase multiple products on cash and on installments by just clicking on his/her mobile phone. The web-based application will let the user decide whether buy a product on cash and on installment according to his/her budget by calculating his installment plan and his price range it will automatically suggest products in his/her price range. This will not only help the user but also useful seller to predict whether which product is most selling product and which brand is top selling brand also they can decide hot products and categories buy watching their sale and purchased products.

1.7. Project Plan

We explained the plan of our project through the Work Breakdown Structure as well with the Gantt chart. In work breakdown structure we have divided all the major task into smaller sub tasks to maintain the efficiency of our project. Further that Work Breakdown Structure has been converted into Gantt chart and in Gantt chart the milestone of the project is labeled and in how much time we will complete the deliverable.

1.7.1. Work Breakdown Structure



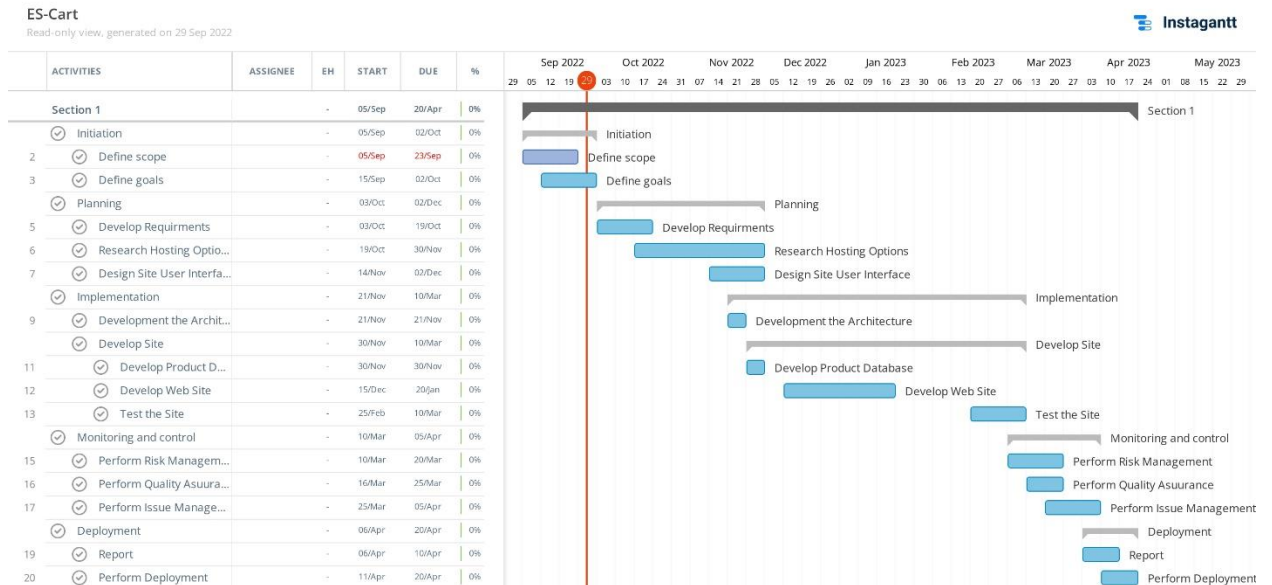
1.7.2. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1.	Initiation	1	Submit Proposal	3 day	M.Arslan
1.1	Define scope	3	Research	1 day	Salman Ahmad
1.2	Define goals	4	Research and proposal	2 days	Salman Ahmad
2.	Planning	5	Surveys and meetings	10 days	Arslan and Salman
2.1	Develop Requirments	6	Research and surveys	2 days	M.Arslan
2.2	Research Hosting Options	7	Testing	5 days	Saif and Salman Ahmad
2.3	Design Site User Interface	8	Making report	3 days	M.Arslan
3.	Implementation	9	Coding, testing and	1 month	Whole team

			execution		
3.1	Development the Architecture	10	Coding and execute	15 days	Whole Team
3.2	Develop Site	11	Coding and execute	15 days	Whole team
3.2.1	Develop Product Database	14	Analysis and testing	5 days	Salman Ahmad
3.2.2	Develop Web Site				Whole team
3.2.3	Test the Site				Whole team
4.	Monitoring and control	16	Perform tracking	20 days	Salman Ahmad , Saif
4.1	Perform Risk Management	17	Testing and analyzing	3 days	M.Arslan
4.2	Perform Quality Assurance				M.Arslan
4.3	Perform Issue Management				Salman Ahmad , Saif
5.	Deployment	19	Testing and delievery	1 month	Whole team
5.1	Project report	20	reporting	5 days	M.Arslan
5.2	Perform Deployment	21	Testing and delievery	3 days	Whole team

1.7.3. Gantt Chart



1.8. Report Outline

Chapter 1 is complete and define introduction of our Easy Shopping Cart reporting web app.

- Introduction.
- Background.
- Motivation and Challenges.
- Goals and Objectives.
- Literature Review/Existing Solution.
- Gap Analysis.
- Proposed Solution.
- Project Plan.
- Work Breakdown Structure WBS.
- Roles and Responsibilities Matrix.
- Gantt chart.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

ES-Cart is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a general-purpose e-commerce store where product like clothes can be bought from the comfort of home through the Internet.

2.1.1. Purpose

A web-based application will be the end product of our project in which user will have the option to purchase multiple products on cash and on installments by just clicking on his/her mobile phone. The web-based application will let the user decide whether buy a product on cash and on installment according to his/her budget by calculating his installment plan and his price range it will automatically suggest products in his/her price range. This will not only help the user but also useful seller to predict whether which product is most selling product and which brand is top selling brand also they can decide hot products and categories buy watching their sale and purchased products.

2.1.2. Document Conventions

The whole document is written in MLA format. Topics introduced in section and subsection headings in a bold text. Highlighting is to point out words in the glossary and Calibri text and includes various diagrams like UML, ERD, and Architecture Diagram.

2.1.3. Intended Audience and Reading Suggestions

This document is intended for anyone who wants to know about the project, its purpose, its development & operating method or who wants to relate and develop similar projects. It is

intended for all the stakeholders that has direct or indirect relation with this project. Following are some examples of how different readers can use this document.

#	Audience	Document usage
1	User	User of project can read this document to get overall concept of this project.
		They can understand its purpose and problem that it solves.
		Users can read this document in order to learn how to use this document.
2	Project Manager	Project manager can use this document as a reference to verify development process.
		Project manager can use this document to assess the responsible person and properly manage development process.
3	Developer	Developers can use this document to understand the flow of data and relations between components.
		They can use this to develop system properly.
4	Testers	Testers can use this document as a reference to develop test plan.
		They can use this to understand flow of data and conventions.
		They can use this document in order to determine the test data.
5	Business / Marketing personal	They can use this document in order to understand project's working, the problem it solve, the opportunity and technology it provides.
6	Investors	Investors can read this document in order to get the idea of market potential and success percentage.
7	Document writers	Document writes can use this document as a reference to develop similar documentation.
8	Other readers	Anyone can read this document to get an idea about project. What is this all about, how project works, how to use it and what features it provides.

2.1.4. Product Scope

Product of our project in which user will have the option to purchase multiple products on cash and on instalments by just clicking on his/her mobile phone. The web-based application will let the user decide whether buy a product on cash and on instalment according to his/her budget by calculating his instalment plan and his price range it will automatically suggest products in his/her price range.

2.1.5. References

This ES-CART documents uses the following references.

- See the project scope document for detail information regarding to the project scope.
- A use case diagram and fully dressed use case are present in the chapter 3 to accompany section: 2.2 Overall product description and section: 2.3 Product Feature.
- To understand about the system and its problem chapter no. 1 is added in this report.

2.2. Overall Description

This section will provide the background information about the specific requirements of “Easy shopping cart” system development environment and services required to develop the system in details.

2.2.1. Product Perspective

Following the rapid development of the web Applications, shopping applications are widely used today for various goals. An essential point in the prosperity of these Applications is their level of usability. Accordingly, measuring this usability is indispensable for these Applications to check whether they are moving in the right path. In addition to assessing usability scores, the design parameters of these Applications have been classified and reviewed in a detailed manner in order to determine the design priorities of these Applications as one of the main results of this study.

2.2.2. User Classes and Characteristics

Our design scheme is to design of our application user friendly to our customer. Therefore, the experience with tracking app will not be a factor that uses application like that. The user of specific age like 18 to on word will be able to operate this application.

2.2.3. Operating Environment

In this case, we selected a computer or laptop or mobile through which a user will access the web application url / path and then perform different action to find out different things according with his/her need.

2.2.4. Design and Implementation Constraints

Following are the design and implementation constraints of our system.

Design Constraints:

- User interference design must be simple and attractive.
- System design must be optimized and efficient.
- User interface design should be easy to learn.
- User interface design should be easy to use.

Implementation Constraints:

- Code must be clean and readable.
- Product must me developed within time and budget.

2.2.5. Assumptions and Dependencies

We have assumed that the user can properly handle this system through mobile and computer system, and have Wi-Fi or internet to access and must have some knowledge to operate the application and perform its basic features.

2.3. External Interface Requirements

2.3.1. User Interfaces

User Perspective

- Registration page
- Login page
- Product listing page
- Product detail page
- Cart page
- Wishlist page
- Brands page
- Category page
- Vendors page
- Installment request form
- Merchant request form
- Checkout form

2.3.2. Hardware Interfaces

This document will describe the environment in which the software will operate, including the hardware platform, operating system. The system will operate with the following hardware and software components.

- mobile
- Pc

2.3.3. Software Interfaces

The app will be made with the help of these tools: Visual studio

2.3.4. Communications Interfaces

This document will describe the environment in which the software will operate, including the communication interface. The system will operate with the following communication interfaces

- Whatsapp
- Email
- Instagram
- Twitter
- Facebook

2.4. System Features

These are the features of the system that are really important to write down there in such order if developer start its work from first requirement to the last it will not face any problem of like this requirement should in first now, I have to change all the code to fulfil that requirement such problems will cause a great time lose.

Functional requirements are the requirements which is important for completion of the small print given by the client. In this some requirements are putted by programmer which is important for correct functionality. For example

- Minimum steps to make a purchase.
- Mobile-friendliness.
- Unique, recognizable design.
- Relevant, useful content.
- Email tools.
- Live chat

2.4.1. Register page

2.4.1.1. Description and Priority

User must be register to the system by providing the required information. It is the high priority feature to use the system. So, User must have to register.

2.4.1.2. Stimulus/Response Sequences

Home page:

After successfully login the home window will show the following options.

2.4.1.3. Functional Requirements

REQ-SF1-1: To test the Login/Authentication interface for the Admin

REQ-SF1-2: To test the Login/Authentication interface for the users

REQ-SF1-3: To test, users can view the items they add in the shopping cart

2.4.2. Login page

2.4.2.1. Description and Priority

The admin and Users login to the system by entering valid user email and password.

2.4.2.2. Stimulus/Response Sequences

After login the home interface show and display the all items and etc.

2.4.2.3. Functional Requirements

REQ-SF2-1: The Admin shall be able to view all the users registered in the system.

REQ-SF2-2: The Admin shall be able to upload new/revised items as well as to add/modify the categories.

REQ-SF2-3: The Easy Shopping-cart web application shall be accessible to all the users to browse all the categories and the items

2.4.3. Search Bar

2.4.3.1. Description and Priority

Users should use search bar to search their desire product. Write their what user wants to search.

2.4.3.2. Stimulus/Response Sequences

Searched results will be shown in a list.

2.4.3.3. Functional Requirements

REQ-SF3-1: The Users is searching the product or items is add to the Wishlist and shopping cart page.

2.4.4. Wishlist Page

2.4.4.1. Description and Priority

The Users will add to favorite product through this page. User can add one or more products in Wishlist page.

2.4.4.2. Stimulus/Response Sequences

The product is added in the Wishlist and will display in Wishlist page.

2.4.4.3. Functional Requirements

REQ-SF4-1: The product user wants to but will fits go to cart. He/She will see detail and then put that product into cart and then place order.

2.4.5. Cart Page

2.4.5.1. Description and Priority

The Users should add item in cart when click add to cart Click item.

2.4.5.2. Stimulus/Response Sequences

The product is added to the Cart.

2.4.5.3. Functional Requirements

REQ-SF5-1: The user is buying the item.

2.5. Nonfunctional Requirements

- System should be able to handle multiple users.
- Login by username, password should be incorporated wherever necessary
- Should be user friendly and display easy to understand error messages

2.5.1. Performance Requirements

- Its performance should be accurate
- It should be accessible all the time to users/admin
- Its server's link should never be break down
- To get best/better performance or speed, should need best internets to servers
- It should bear the load of as many users at a time, can login to site system shouldn't be crashed
- System shouldn't be slow on having load on site Safety Requirements

2.5.2. Safety Requirements

- Servers should be placed in low temperature or in cold place
- System should have a backup server for data backup
- System should have a backup database in case of any server damage or loss of data
- Systems should be well updated

2.5.3. Security Requirements

- Security should be of fool proof
- It should have logins
- It should have a case sensitive password
- It should have pin option
- Account should be integrated
- Personal information of a client should not be leaked out
- Password should not be shown on the URL

2.5.4. Availability Requirements

System should be available all the 24/7 so that it could not be have any difficulty in getting any order form the customer or updating its database.

2.5.5. Reliability Requirements

The usability of a product is indispensable to ensuring that it works properly based on the determined targets and does not create any trouble for customers during the consumption period. In fact, the main goal of these evaluations is to examine the bond between the customers and the product to understand whether customers are able to learn and apply it for reaching their goals.

2.5.6. Maintainability/Supportability Requirements

Reliability can be determined by noticing the number of failures of a system in a specific period. Our system must be most reliable which we can achieve by testing it again and again after development by the testing team.

2.5.7. Portability Requirements

The application is developed in ASP.NET. It would be portable to another operating system provided .NET Framework is available for the OS. As the database is made in SQL Server, porting the database to another database server would require some development effort.

2.5.8. Efficiency Requirements

Efficiency requirements address the user concern for how fast the system functions, how efficiently the system takes in inputs and processes outputs, and how much can be processed at a time.

2.6. Domain Requirements

Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain must necessarily exhibit come under this category.

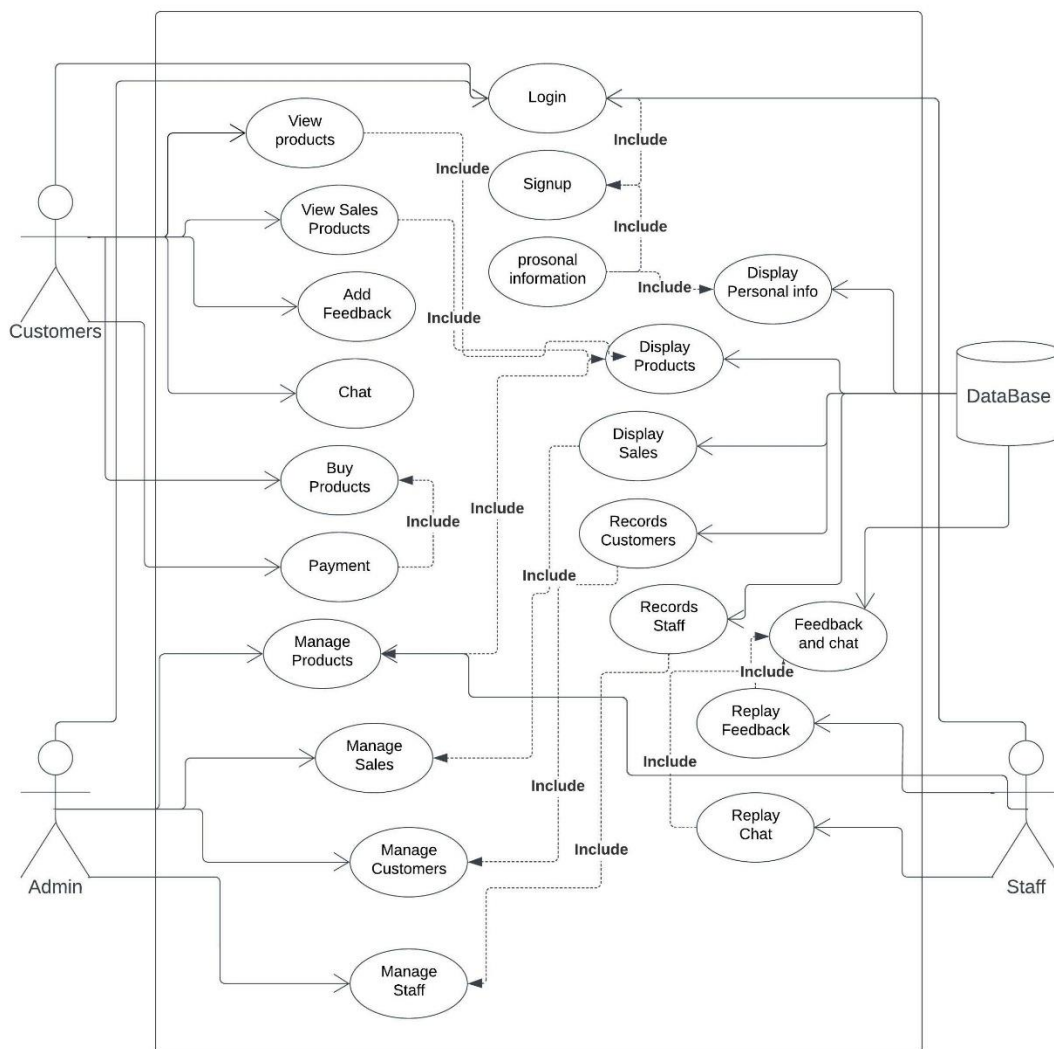
Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

In this chapter we discuss all about use case. Use case analysis is a technique used to identify the requirements of a system and the information used to both define processes used and classes which will be used both in the use case diagram and the overall use case in the development or redesign of a software system or program.

3.1. Use Case Model



3.2. Use Cases Description

A utilization case is an accumulation of related achievement and disappointment situations that depict on-screen characters utilizing the framework to help an objective. A situation is a particular succession of activities and associations amongst performing artists and the framework under discourse

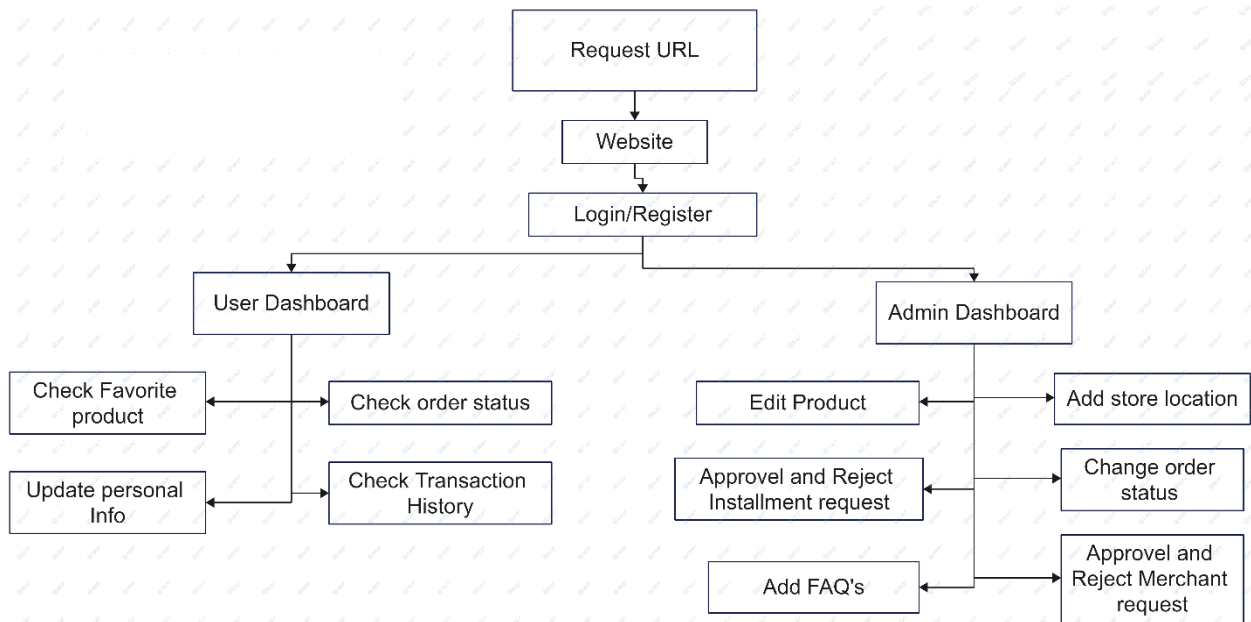
Chapter 4

System Design

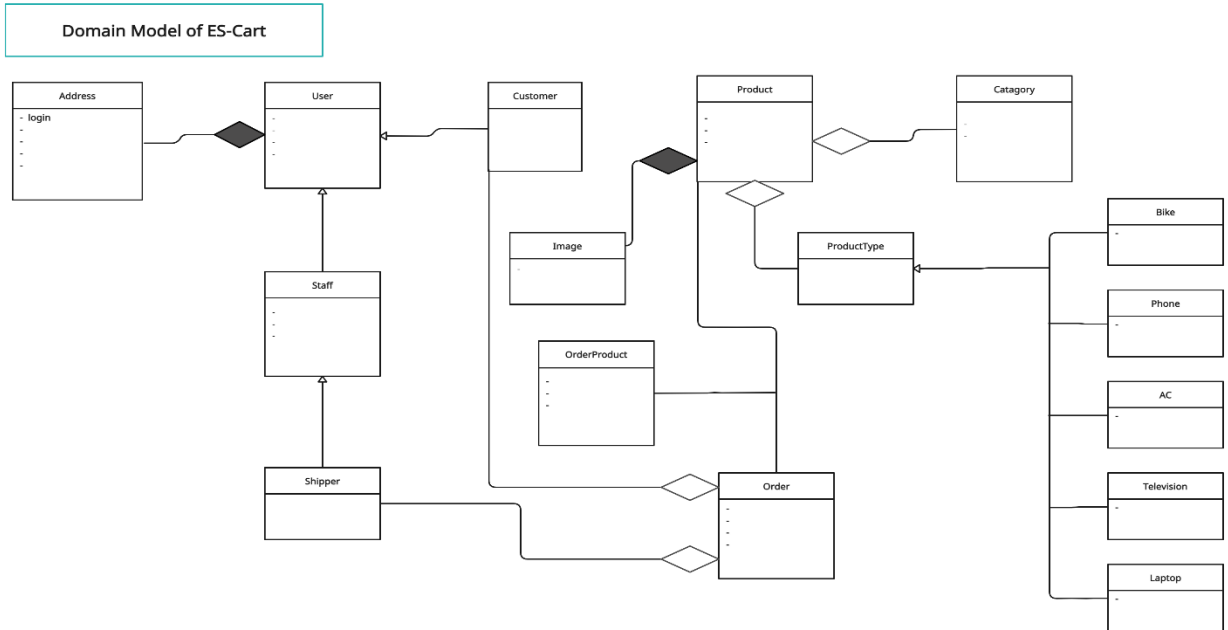
Chapter 4: System Design

This chapter describe about system use cases and how the different actors are interacting with the system. In use case main functionalities of the system are defined that are divided in different modules e.g. major use case in which all functionalities of the system are defined is divide into customers and admin module and each actor interacting with module and performing his module functionalities.

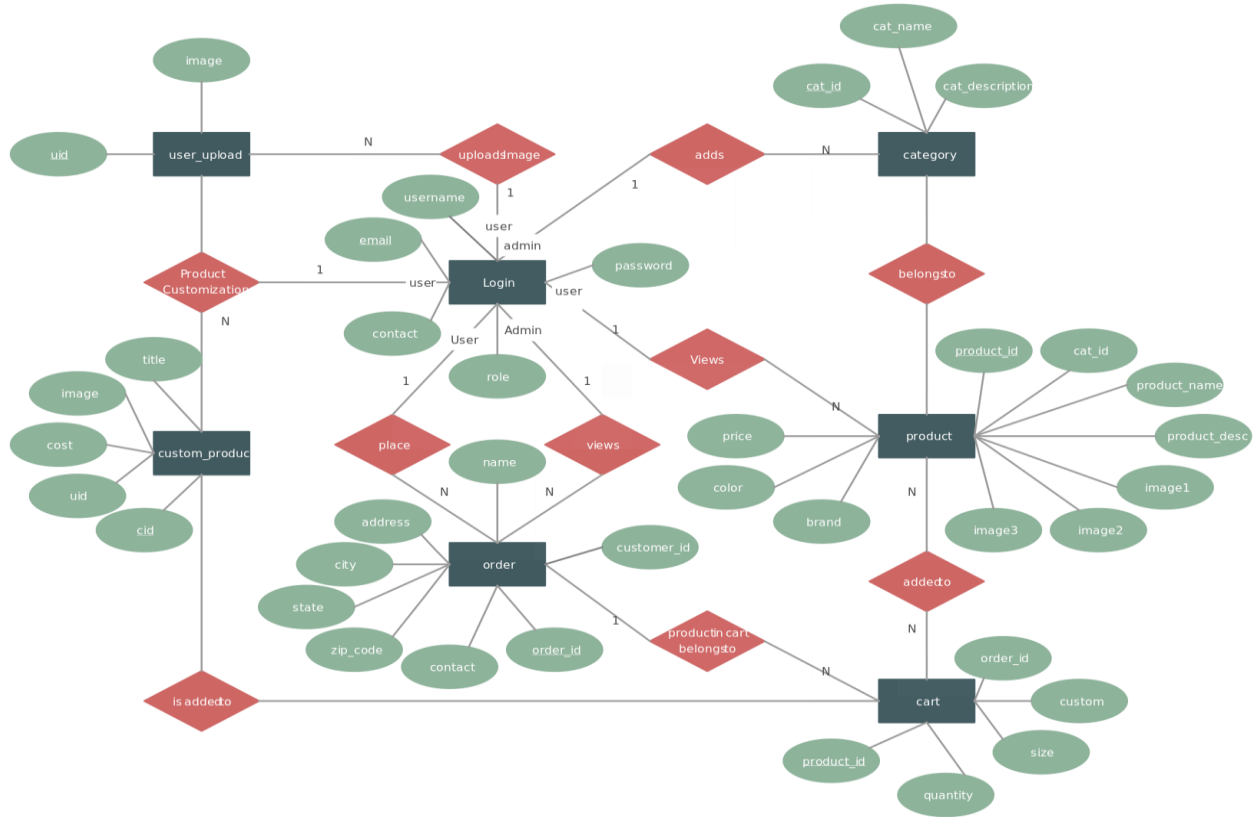
4.1. Architecture Diagram



4.2. Domain Model

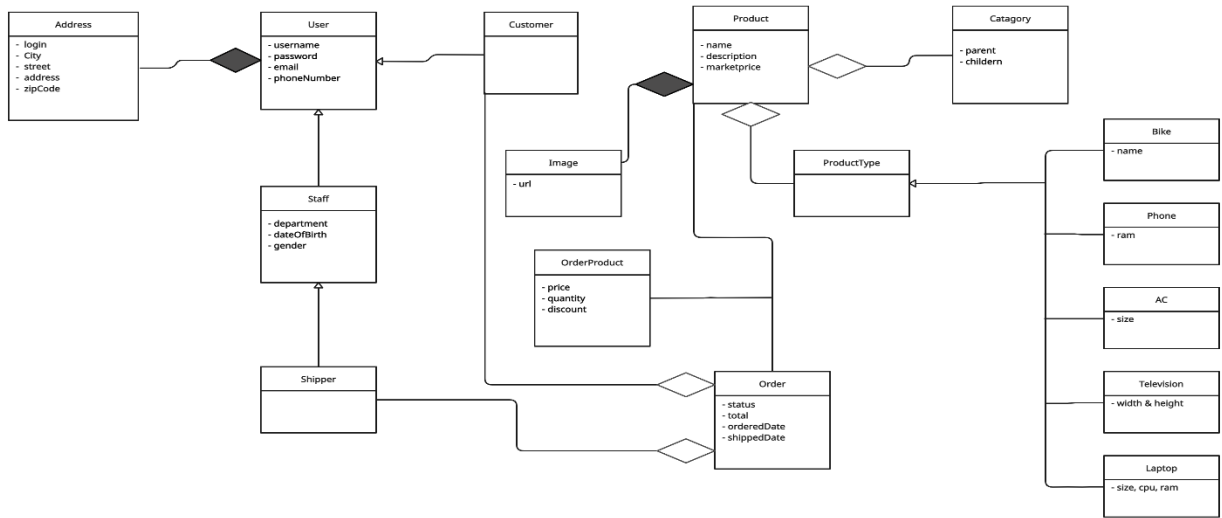


4.3. Entity Relationship Diagram with data dictionary

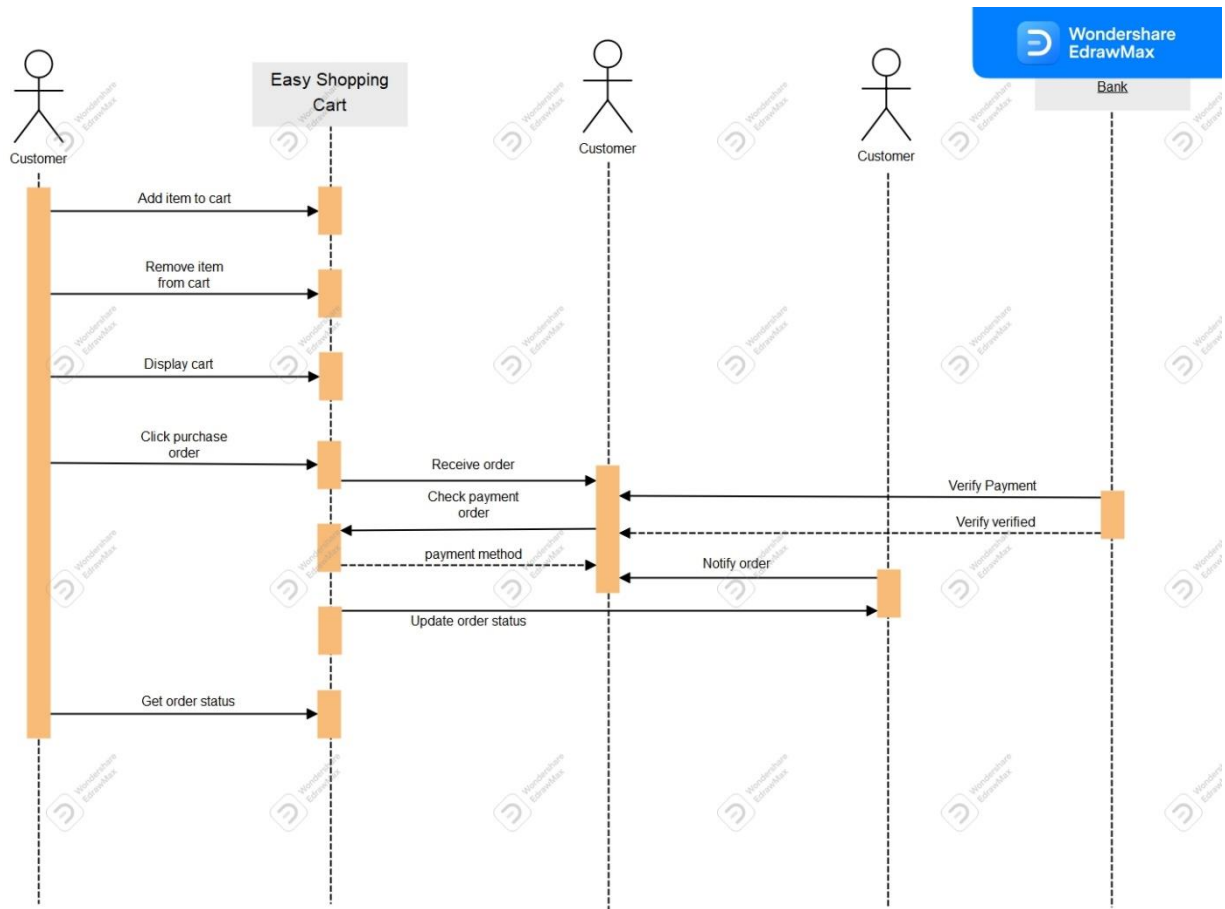


4.4. Class Diagram

Class diagram of ES-Cart



4.5. Sequence/ Collaboration Diagram

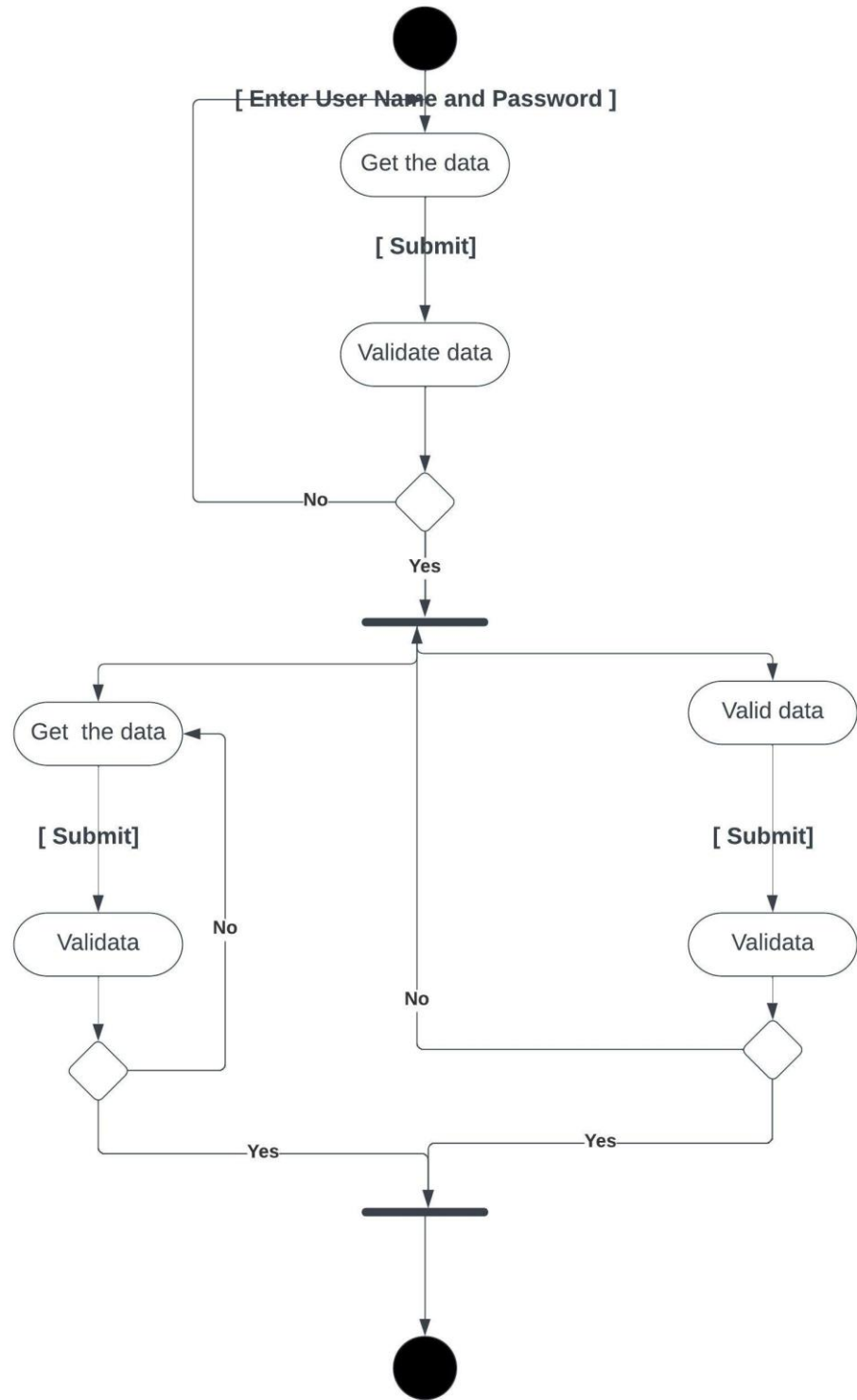


4.6. Operation contracts

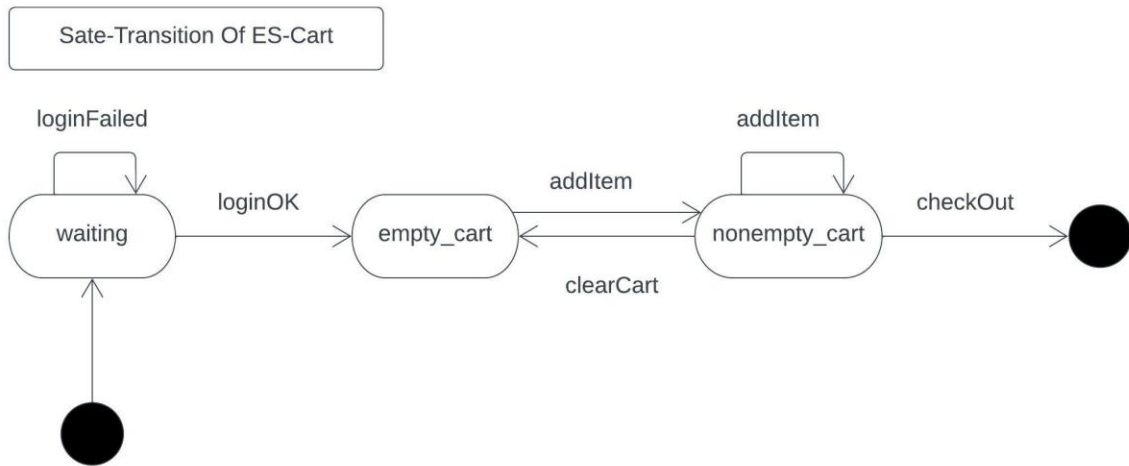
Use Cases often fully describe the behavior of a system | But they may not be enough | Operation Contracts describe how the internal state of the concepts in the Domain Model may change | Operation Contracts are described in terms of preconditions and post conditions.

4.7. Activity Diagram

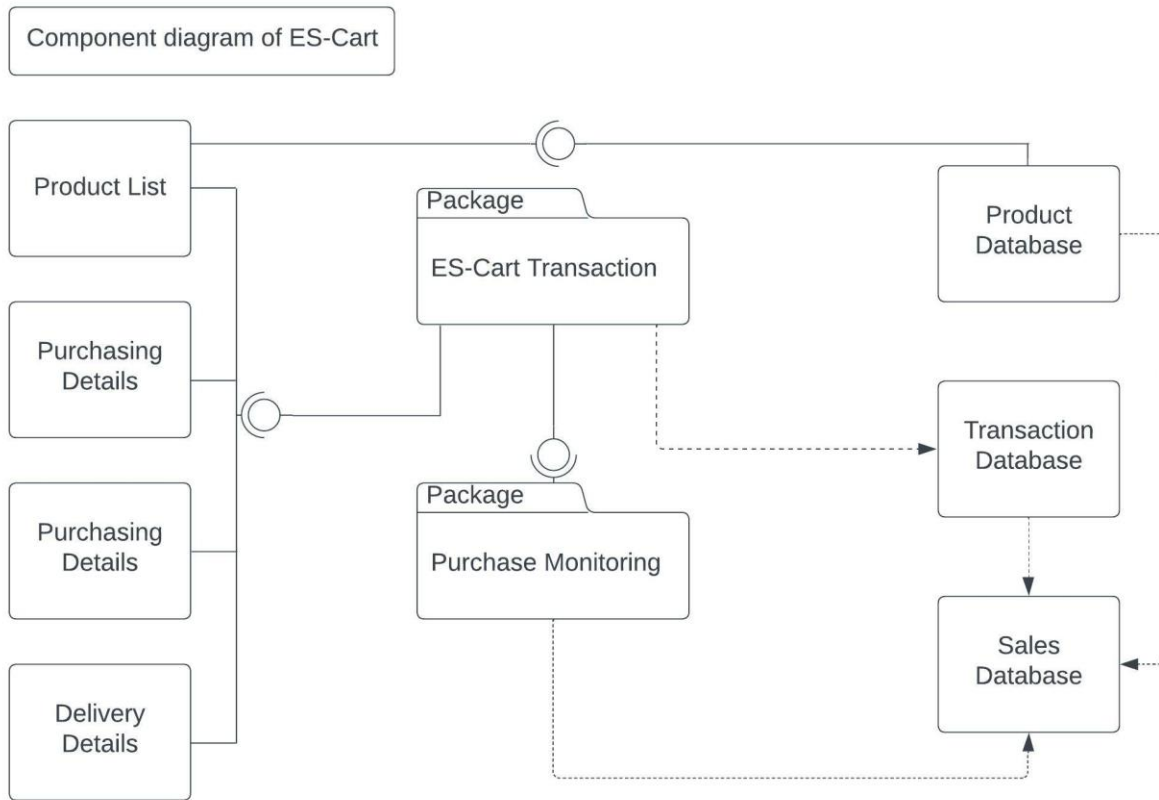
ES-Cart Activity Diagram



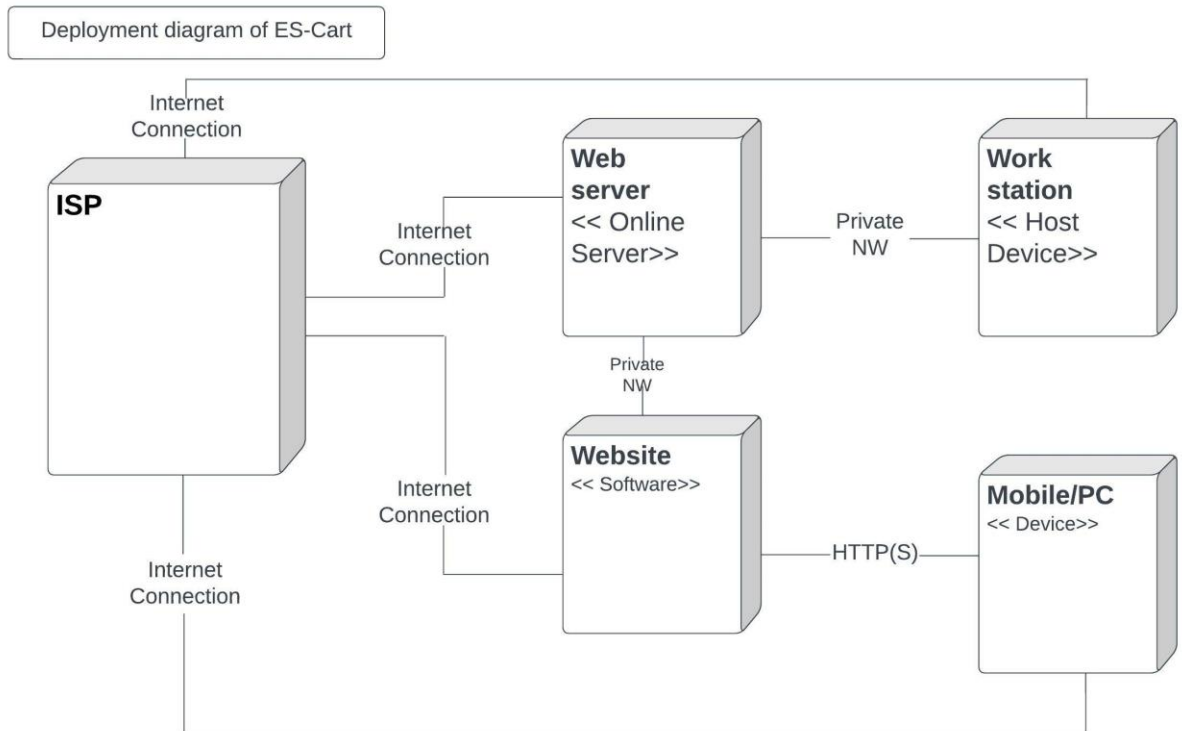
4.8. State Transition Diagram



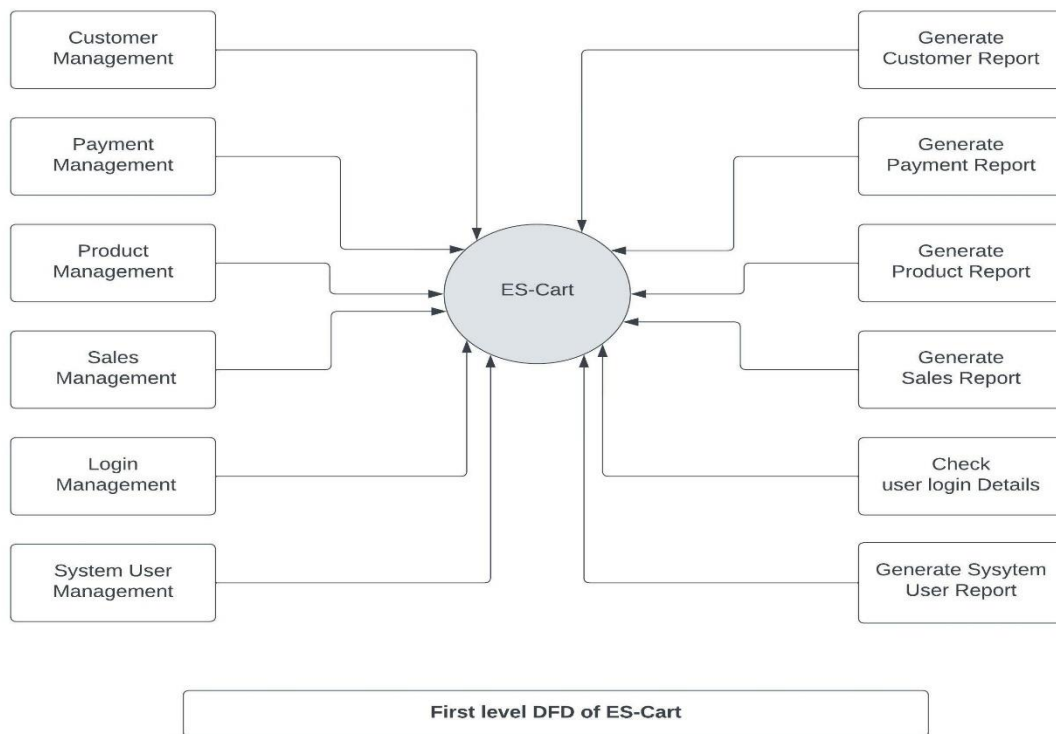
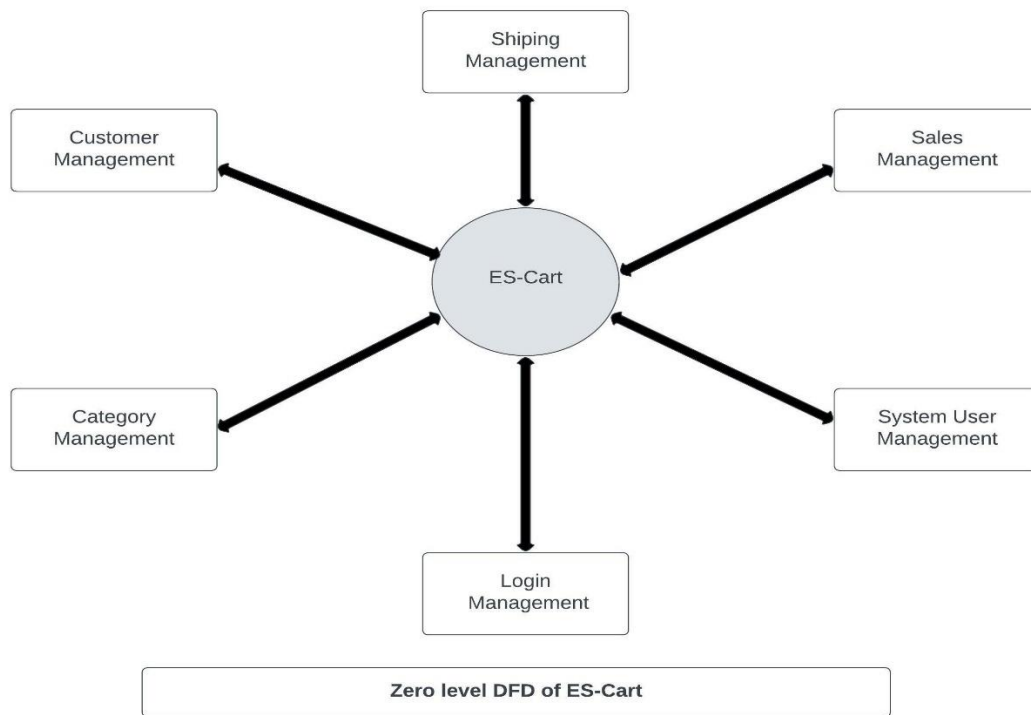
4.9. Component Diagram

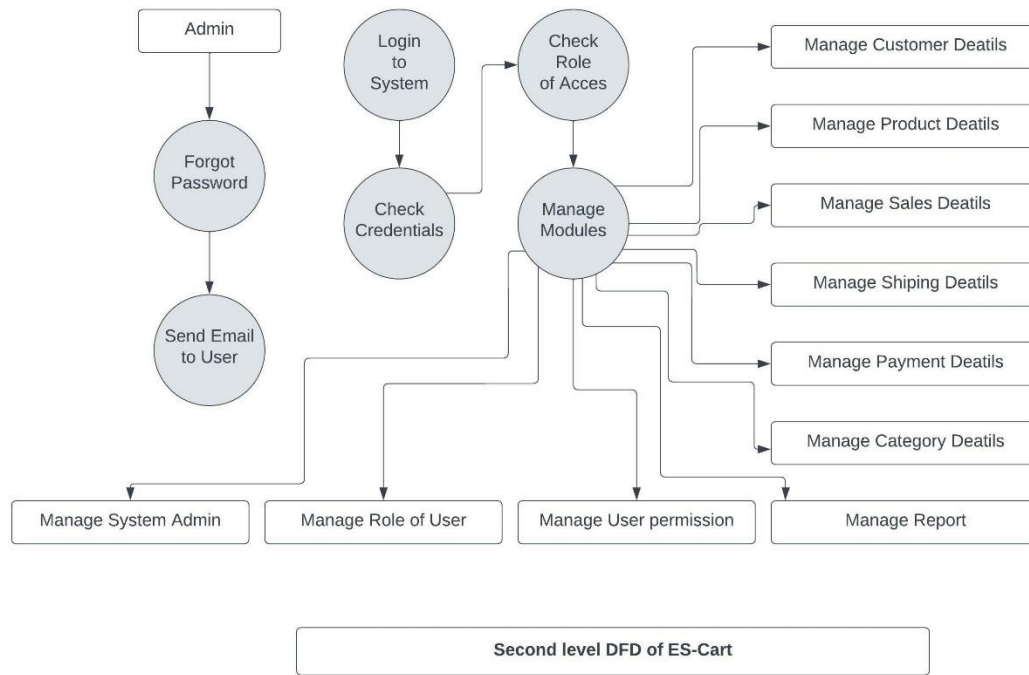


4.10. Deployment Diagram



4.11. Data Flow diagram





Chapter 5

Implementation

Implementation

In this chapter we will discuss about the web application and hardware implementation which is connected to our system and its development cycles including tools and techniques to easy shopping cart system. This web application will be user-friendly and easy to control.

5.1. Important Flow Control/Pseudo codes

Admin Panel:

- Register
- Manage Products
- Manage Sales
- Manage Customer
- Manage Staff

Staff Panel:

- Register
- Reply feedback
- Reply chat

Customer Panel:

- Logs/Register
- View Products
- View sales Products
- Chat
- Buy products
- Payment

5.2. Components, Libraries, Web Services and stubs

This is a Window Application Based Project as per requirement on Visual Studio and Sql Data base is fully dedicated to our kind of projects as it has list of Libraries to perform such operation i.e:

- Bootstrap
- JavaScript/ jQuery

5.3. Deployment Environment

This is a desktop application for windows which can also work or run operation on desktop PC. The Minimum specification to run this project is following:

- Intel I3 Processor 4th generation
- 4 GB RAM
- 500 GB Hard Disk
- Cache 4MB
- Mobile Graphics Chipset of 1982 MB

5.4. Tools and Techniques

1 Microsoft Window

Microsoft Window Rational rose is an object-oriented unified modeling language software design tool intended for visual modeling and component construction of enterprise-level software. We use this tool for system Architect diagram in our project.

2 Visual Studio

Visual studio is a developing tool that is used to developed web applications. The purpose is to use this tool for developed web application.

3 Xampp

Xampp as a local server.

5.5. Best Practices / Coding Standards

For good software best practice is that you use best coding standards.

- Commenting
- Readable and efficient code
- Use your IDE drop down menu
- Assign name to all variables
- The shorter and clear your name or commands are the better off you will be

5.6. Version Control

Version control is a method of tracking changes to documents and files so that you always know which version is the current iteration. In version control we change feature of app in demand of different users this is our first version of app, after launching in market or Google play store and after getting user feedback we will create changes day by day on user demand.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Test & Evaluation (T&E) is the process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to assess the progress of design, performance, supportability, etc. Developmental test and evaluation (DT&E) is an engineering tool used to reduce risk throughout the acquisition cycle. Operational test and evaluation (OT&E) are the actual or simulated employment, by typical users, of a system under realistic operational conditions.

6.1. Use Case Testing

Use case testing is a type of functional testing that focuses on testing the user's interaction with the system. In the context of an ES-Cart website, use case testing involves testing the website's functionality by simulating user scenarios. Here are use case testing scenarios for an eCommerce website:

6.1.1 Registering a new account:

Test the registration process for new users, including validation of input fields, email confirmation, and account creation.

6.1.2 Searching for products:

Test the search functionality of the website by performing searches for various products using different keywords and filters.

6.1.3 Adding products to cart:

Test the cart functionality by adding products to the cart, verifying the quantity, and removing products from the cart.

6.1.4 Checking out:

Test the checkout process by selecting a shipping method, providing payment information, and confirming the order.

6.1.5 Cancelling an order:

Test the ability to cancel an order before it is shipped by selecting the appropriate option on the website.

6.1.6 Updating account information:

Test the ability to update user account information, such as shipping address, payment information, and contact details.

6.1.7 Providing feedback:

Test the feedback functionality by providing feedback on products, the website, or the customer service experience.

Use case testing for an ES-Cart website should cover a wide range of scenarios to ensure that the website is functioning as intended and providing a positive user experience. The testing should also be repeated regularly to ensure that the website remains functional and up-to-date with the latest user needs and expectations.

6.2. Equivalence partitioning

Equivalence partitioning is a technique used in software testing to divide the input data into equivalence classes or partitions, such that the behavior of the system is expected to be the same for any input within the same partition. This technique helps in reducing the number of test cases while still ensuring that different variations of inputs are covered. Here's how equivalence partitioning can be applied to the ES-Cart website:

6.2.1. User Registration:

- Partition 1: Valid user registration data (e.g., unique email, valid password, correct format).

- Partition 2: Invalid user registration data (e.g., duplicate email, invalid password format, missing required fields).
- Partition 3: Edge cases and boundary values (e.g., minimum/maximum length of fields, special characters in the password).

6.2.2. Product Search:

- Partition 1: Valid search query (e.g., product name, category, brand).
- Partition 2: Invalid search query (e.g., empty search query, invalid category).
- Partition 3: Edge cases and boundary values (e.g., search query with special characters, extremely long search query).

6.2.3. Add to Cart:

- Partition 1: Successful addition of a product to the cart.
- Partition 2: Failure to add a product to the cart (e.g., out of stock, invalid product identifier).
- Partition 3: Edge cases and boundary values (e.g., adding the maximum number of products to the cart, adding a negative quantity).

6.2.4. User Reviews:

- Partition 1: Valid user review submission.
- Partition 2: Invalid user review submission (e.g., empty review text, rating out of range).
- Partition 3: Edge cases and boundary values (e.g., extremely long review text, maximum/minimum rating).

By applying equivalence partitioning to the above functionalities of the ES-Cart website, you can identify representative test cases for each partition. This ensures that you cover a variety of input scenarios while minimizing the number of test cases needed. It's important to consider both valid and invalid inputs as well as edge cases to validate the behavior and functionality of the website under different conditions.

6.3. Boundary value analysis

Boundary value analysis is a software testing technique that focuses on testing the boundaries and limits of input values to determine if the system behaves correctly and handles these boundaries appropriately. Here's how boundary value analysis can be applied to the ES-Cart website:

6.3.1. User Registration:

- Test with the minimum and maximum length of input fields for registration data, such as the username, email, and password.
- Verify the behavior of the system when entering one character less and one character more than the minimum and maximum limits.
- Test with boundary values for other fields, such as the date of birth or phone number, to ensure they are handled correctly.

6.3.2. Product Pricing:

- Test the behavior of the website when a product is priced at the minimum and maximum values allowed. Verify that the system handles these extreme values correctly.
- Test with prices just below and just above the minimum and maximum limits to ensure that the system behaves as expected in these boundary conditions.
- Verify the behavior of the system when entering a discounted price that is the same as the regular price or exceeds it.

6.3.3. Quantity of Products:

- Test with the minimum and maximum quantity of products that can be added to the cart or ordered. Verify that the system allows the minimum and maximum limits and displays appropriate messages when exceeded.

- Test with quantities just below and just above the minimum and maximum limits to ensure that the system handles these boundary conditions correctly.

6.3.4 Payment Amount:

- Test with the minimum and maximum payment amounts allowed. Verify that the system processes payments correctly for these extreme values.
- Test with payment amounts just below and just above the minimum and maximum limits to ensure proper handling of these boundary conditions.
- Verify the behavior of the system when entering a payment amount that is equal to the minimum or maximum allowed.

6.3.5. User Reviews:

- Test with the minimum and maximum length of review text allowed. Verify that the system handles these boundary conditions correctly.
- Test with review text just below and just above the minimum and maximum limits to ensure proper behavior.
- Verify the behavior of the system when submitting a review with the minimum or maximum rating allowed.

By performing boundary value analysis on the ES-Cart website, you can ensure that the system handles edge cases and boundary conditions appropriately. This technique helps identify any issues related to input validation, error handling, and boundary checks. It is important to consider both the lower and upper boundaries as well as the values immediately outside these boundaries to validate the robustness and correctness of the website.

6.4. Data flow testing

Data flow testing for the ES-Cart website, or any similar eCommerce website, involves analyzing the flow of data throughout the system and testing various scenarios to ensure data integrity, accuracy, and security. Here are some key considerations for data flow testing:

6.4.1. Data Inputs:

Identify the various sources of data inputs in the ES-Cart website. This can include user inputs such as registration details, product search queries, payment information, or customer reviews. Test the website's ability to handle different types of input data, including valid data, invalid data, edge cases, and malicious data.

6.4.2. Data Storage:

Understand how data is stored and managed within the ES-Cart website. This includes user account information, product details, transaction records, and other relevant data. Verify that data is correctly stored, updated, and retrieved from databases or other data storage systems. Test scenarios such as creating new accounts, updating user information, or processing orders to ensure data consistency and accuracy.

6.4.3. Data Flows Between Components:

Analyze how data flows between different components and modules within the ES-Cart website. This includes interactions between frontend and backend systems, APIs, payment gateways, inventory management systems, and other integrated services. Test the data flow to ensure seamless communication, accurate data exchange, and proper handling of errors or exceptions.

6.4.4. Data Security and Privacy:

Pay attention to data security measures within the ES-Cart website. This includes encryption of sensitive data, protection against data breaches or unauthorized access, adherence to privacy regulations, and secure transmission of data during transactions. Test the website's security features and perform vulnerability assessments to ensure data protection.

6.4.5. Data Backup and Recovery:

Ensure that appropriate data backup and recovery mechanisms are in place for the ES-Cart website. Test backup and restore procedures to validate the ability to recover data in the event

of system failures, data corruption, or disasters. Regularly perform backup tests to verify the integrity of backed-up data.

By performing comprehensive data flow testing, the ES-Cart website can ensure the reliability, accuracy, and security of the data within the system. It helps identify and rectify any issues related to data handling, processing, and storage, ensuring a smooth and trustworthy user experience.

6.5. Unit testing

Unit testing of the ES-Cart website involves testing individual components or units of code in isolation to ensure their functionality and correctness. Here are some key considerations for unit testing the ES-Cart website:

6.5.1. Test Framework:

Select a suitable unit testing framework, such as PHP Unit for PHP-based applications, to write and execute unit tests. The framework provides a structure for organizing tests, running them, and generating test reports.

6.5.2. Testable Units:

Identify the units of code that can be tested in isolation. This can include individual functions, methods, classes, or modules within the ES-Cart website's codebase. Each unit should have a clearly defined responsibility and input-output behavior.

6.5.3. Test Coverage:

Aim to achieve high test coverage by ensuring that all critical and complex units are tested. This helps identify potential bugs, improve code quality, and enhance overall system reliability.

6.5.4. Test Cases:

Develop test cases that cover different scenarios and edge cases for each unit. Consider inputs, outputs, boundary values, error conditions, and exceptions. Test cases should verify the expected behavior of the unit and ensure that it produces the correct results.

6.5.5. Mocking and Stubbing:

Use mocking and stubbing techniques to isolate the unit under test from its dependencies, such as external APIs, databases, or services. Mock objects or stubs can simulate the behavior of these dependencies, allowing for focused testing of the unit itself.

6.5.6. Test Data:

Prepare test data specific to each unit test case. This can include providing valid inputs, invalid inputs, and edge case data to cover a wide range of scenarios.

6.5.7. Test Execution and Automation:

Run unit tests regularly during the development process to identify issues early on. Automate the execution of unit tests so that they can be easily run and integrated into the continuous integration and deployment (CI/CD) pipeline.

6.5.8. Code Review and Refactoring:

Use unit testing as a means to review and improve the quality of the code. Analyze test results and refactor the code if necessary to improve its readability, maintainability, and testability.

6.5.9. Continuous Integration and Test Suites:

Integrate unit tests into the CI/CD pipeline to ensure that new code changes do not introduce regressions or break existing functionality. Maintain a suite of unit tests that can be run collectively to provide comprehensive coverage and quickly identify any issues.

Unit testing plays a crucial role in identifying defects and ensuring the stability and correctness of individual code units within the ES-Cart website. By conducting thorough unit testing,

developers can catch bugs early, improve code quality, and build a robust and reliable ecommerce platform.

6.6. Performance testing

Performance testing is the process of determining the speed, responsiveness, and stability of an application, software program, or device under a workload. Performance testing can involve quantitative tests done in a lab. The performance of the software is determined by three factors:

- Speed
- Scalability
- Stability

ES-Cart website application has started working and its performance is according to the expectations, all the modules are working properly. ES-Cart website application is properly controlled. All the validations and verifications are working smoothly. We perform the load testing in our application and it works properly.

6.7. Stress Testing

Stress testing pushes the website beyond its normal capacity to evaluate its performance under extreme conditions. It involves applying heavy loads, increasing user traffic, or stressing specific functionalities to assess how the website handles peak loads, spikes in user activity, or unexpected traffic surges. The goal is to identify the breaking point or measure the degradation in performance when the system is pushed to its limits. ES-Cart website application is working properly under stress condition and working correctly.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

This document is the report about our FYP project, it will give us a detailed view of the data flow and the roles and responsibilities of the actors and the system. There are seven Chapters in this project and the content in them is mentioned below Table.

#	Chapters	Document Usage
1.	Introduction	This section covers the <ul style="list-style-type: none"> • Summary and background of this project. • Project scope, goals, and objectives are also mentioned • Project and work breakdown structure
2.	Software Requirement Specifications	This section covers the <ul style="list-style-type: none"> • Description of the product • Functional and non-functional requirements • Product interfaces and business rules
3.	Use Case analysis	This section covers <ul style="list-style-type: none"> • Use cases that are intended for developers, testers, and other technical readers <p><i>“Use case is a list of actions or event steps typically defining the interactions between a role and a system to achieve a goal”</i></p>
4.	System Design	<ul style="list-style-type: none"> • This Section is solely intended for the developers and the project managers • Detailed diagrams are the part of this section

		describing the system architect and the data flow
5.	Implementation	<p>This section describes</p> <ul style="list-style-type: none"> • The tools and techniques used for the development • Development environment components and best practices are also mentioned.
6.	Testing and Evaluation	<p>This section describes</p> <ul style="list-style-type: none"> • This section describes the types of testing through which the System has passed
7.	Summary, Conclusion, and Future Enhancements	<p>This section describes</p> <ul style="list-style-type: none"> • The overall summary of the chapters • The conclusions made from this Project • The future enhancements that are to be made in it
8.	Appendix	This section includes any additional or supplementary information on the book's topic

7.2. Achievements and Improvements

By working on this project, we got hands-on practical experience on coding knowledge that we gained during our 4 years of degree this project cleared our basic to a high level of concepts or any confusion that we had. By doing this project we not only technologically but as a team learned many things. We learned to do teamwork and how to manage and divide work and save time by doing so. Teamwork is the collaborative effort of a group to achieve a common goal or to complete a task most effectively and efficiently. This concept is seen within the greater framework of a team, which is a group of interdependent individuals who work

together towards a common goal. Contribute to Open-Source Projects, Teach Others What You Know, Challenge Yourself with a New Skill, Schedule Regular Downtime.

Our Improvement Involves increase in technical abilities these skills include working with modern-day technologies like.

- Frontend Skill
- Responsive design skill
- Testing and debugging skill
- Backend skill

Amplifying the need for good communication skills is the fact that coding is something that is very hard to grasp for non-developers. You must be able to make other people understand technical problems, which is something that can be tough. As a way of improving your communication skills, a great idea is to have regular talks with people you know outside of the office and try to make them understand what you are working on.

7.3. Critical Review

A critical review of an ES-Cart website involves an assessment of its strengths, weaknesses, and areas for improvement. Like as user experience is good or not and similar to etc modules are in ES-Cart website.

7.3. Lessons Learnt

This section describes the lesson learned in the process of making this PHP framework of Laravel working which is mentioned below

7.3.1. Applying Knowledge:

The project provided us the opportunity to apply the knowledge that we learned during our 4 years of the degree, and to self-explore and learn new technologies that would prepare us for over future.

7.3.2. Collaboration:

It's incredibly important to have close collaboration with the full project team at the very early stages of the project before the budget is set. Do not make key decisions without your project team already on board.

Collaborating early in the project will save you both time and heartache.

7.3.3. Be very patient

Having trust in each other and on Allah being patient is one of the most important things that we learned from this project do not give up, keep working hard you will achieve your goal.

7.4. Future Enhancements/Recommendations

Here are some future enhancements and recommendations for an ES-Cart website:

- Online payment method
- Integrate Mobile App
- Customer supports
- Social media marketing
- Order Tracking and Notifications
- Voice Search Integration

Reference and Bibliography

Reference and Bibliography

https://www.irjmets.com/uploadedfiles/paper/volume3/issue_3_march_2021/6623/1628083280.pdf

<https://5.imimg.com/data5/SELLER/Doc/2020/9/DA/DF/GO/68259795/multi-vendor-ecommerce-website.pdf>

<https://opus.govst.edu/cgi/viewcontent.cgi?article=1079&context=capstones>