

# **Store to Door**

**Final Year Project**

**Session 2016-2020**

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of CS & IT

Faculty of Computer Science & Information Technology

The Superior College, Lahore

FALL 2020

Type (Nature of project)	[ <input checked="" type="checkbox"/> ] Development    [ <input type="checkbox"/> ] Research    [ <input type="checkbox"/> ] R&D			
Area of specialization	BS (Computer Science)			
<b>FYP ID</b>	FYP-BSCS-F19-033			
<b>Project Group Members</b>				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F16-275	M BILAWAL NISAR	Bcsm-f16-275@superior.edu.pk	
(ii)	BCSM-F16-294	M USMAN	Bcsm-f16-294@superior.edu.pk	
(iii)	BCSM-F16-449	M JAVID	Bcsm-f16-449@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 M BILAWAL NISAR S/D of NISAR AHMAD, group leader of FYP under registration no BCSM-F16-275 at Software Engineering Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: \_\_\_\_\_ Name of Group Leader: M BILAWAL NISAR      Signature: \_\_\_\_\_

Name of Supervisor: Sir Ahmed Bilal

Designation: Lecturer

Signature: \_\_\_\_\_

Signature: \_\_\_\_\_

HOD: Dr. Irfan

Signature: \_\_\_\_\_

# Project Report

## Store to Door

### Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

\_\_\_\_\_

Name: \_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

## Dedication

*This work is dedicated to our supervisor Sir Ahmed Bilal and co-members who gave us opportunity to work with flow and work in creative way to make this document concise and clean. Our supervisor has guided me to follow standards to achieve our project objectives.*

## **Acknowledgements**

We are really thankful to our supervisor who has guided us to work hard to achieve our objectives as well we know our project was not simple enough but our supervisor has boosted us up to work hard and you will achieve for what you are thinking for. We will never forget our supervisor sir Ahmed Bilal who has great experience in making a standards websites he helps us to select tools and techniques to work on our project and this was very helpful because we were little confused about choosing tools to build our website.

## Executive Summary

Store to Door is an E-commerce website that will have its own store. It will service online shopping process of using daily thing in life and will perform basic functionality of online shopping. It will have some new features like any store if they want to open their shop on this website they can open. They will be strictly announced to have order delivering facilities on their own behalf. So this thing will bring easiness for customer to get their order from desired store by nearby shop within one hour. If customer wants to order a product which belongs to other society then he will wait for one day to receive his/her order. These are some common thing in this website.

## Table of Contents

Store to Door.....	i
Final Year Project .....	i
Session 2016-2020 .....	i
BS in Computer Science .....	i
Dedication .....	v
Acknowledgements.....	vi
Executive Summary .....	vii
Table of Contents.....	viii
List of Figures.....	xi
List of Tables .....	xii
Chapter 1 .....	1
Introduction.....	1
1.1. Background .....	2
1.2. Motivations and Challenges .....	2
1.3. Goals and Objectives.....	3
1.4. Literature Review/Existing Solutions .....	3
1.5. Gap Analysis .....	5
1.6. Proposed Solution .....	5
1.7. Project Plan .....	5
1.7.1. Work Breakdown Structure .....	7
1.7.2. Roles & Responsibility Matrix .....	8
1.7.3. Gantt Chart .....	9
1.8. Report Outline .....	10
Chapter 2.....	12
Software Requirement Specifications.....	12
2.1. Introduction .....	13
2.1.1. Purpose .....	13
2.1.2. Document Conventions .....	13
2.1.3. Intended Audience and Reading Suggestions .....	13
2.1.4. Product Scope.....	13
2.1.5. References .....	14
2.2. Overall Description .....	14
2.2.1. Product Perspective .....	14
2.2.2. Product Functions .....	14
2.2.3. User Classes and Characteristics .....	15
2.2.4. Operating Environment .....	15
2.2.5. Design and Implementation Constraints.....	15
2.2.6. User Documentation.....	16
2.2.7. Assumptions and Dependencies .....	16
2.3. External Interface Requirements.....	16
2.3.1. User Interfaces .....	16
2.3.2. Hardware Interfaces.....	17
2.3.3. Software Interfaces .....	17
2.3.4. Communications Interfaces .....	18

2.4.	System Features.....	18
2.4.1.	System Feature 1 .....	18
2.4.2.	System Feature 2 .....	19
2.4.3.	System Feature 3 (and so on) .....	19
2.5.	Other Nonfunctional Requirements .....	20
2.5.1.	Performance Requirements.....	20
2.5.2.	Safety Requirements.....	20
2.5.3.	Security Requirements.....	20
2.5.4.	Software Quality Attributes.....	20
2.6.	Other Requirements.....	21
Chapter 3	.....	22
Use Case Analysis	.....	22
3.1.	Use Case Model .....	23
3.2.	Use Case Descriptions.....	24
3.3.	Use Case Specification.....	28
Chapter 4	.....	31
System Design	.....	31
4.1.	Architecture Diagram.....	32
4.2.	Domain Model.....	33
4.3.	Entity Relationship Diagram with data dictionary .....	34
4.4.	Class Diagram .....	35
4.5.	Sequence / Collaboration Diagram .....	36
4.6.	Operation contracts .....	36
Operation contracts	.....	36
4.1.	Activity Diagram.....	37
4.2.	State Transition Diagram .....	38
4.3.	Component Diagram .....	39
4.4.	Deployment Diagram .....	40
4.1.	Data Flow diagram .....	41
Chapter 5	.....	42
Implementation	.....	42
5.1.	Important Flow Control/Pseudo codes .....	43
5.1.	Components, Libraries, Web Services and stubs .....	44
5.2.	Deployment Environment .....	44
5.3.	Tools and Techniques.....	45
5.4.	Best Practices / Coding Standards.....	45
5.1.	Version Control .....	46
Chapter 6	.....	47
Testing and Evaluation	.....	47
6.1.	Use Case Testing.....	48
6.2.	Equivalence partitioning .....	53
6.3.	Boundary value analysis.....	55
6.4.	Unit testing.....	56
6.5.	Integration testing.....	56
6.6.	Performance testing.....	57
6.7.	Stress Testing .....	57

Chapter 7 .....	58
Summary, Conclusion and Future Enhancements .....	58
7.1. Project Summary .....	59
7.2. Achievements and Improvements .....	59
7.3. Critical Review.....	59
7.4. Lessons Learnt.....	59
7.5. Future Enhancements/Recommendations .....	59
Appendices.....	60
Appendix A: User Manual .....	61
Reference and Bibliography .....	68

## List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

## List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

# Chapter 1

## **Introduction**

# Chapter 1: Introduction

Store to Door a perfect solution for getting your order's delivered at your doorstep. Our system will make it easy for our customers as it will pick accurate coordinates of the shop, to pick up the order and the address of their place where they have to get it delivered. To make this system work, one must provide the required information, which will include the person's name, his phone number and his address. While placing a delivery a prompt message will be shown if the user wants to deliver on the usual address or another. Moreover, it will provide convenient and an easy to use interface to make it understandable for people of every ages.

## 1.1. Background

As we know there are million plus website that are giving online shopping services to customer where he can place order from anywhere and gets his order on door step some well-known websites like amazon and AliExpress are performing best now a days in this race but there are some issues that a customer is still facing and the big problem is to get product in shortest time with payable shipping fee and payable product price. Our website is not like amazon or AliExpress. Store to door website covers small area that within a society/city. We give service within a society. So I was thinking about these to solve out and then my university gave me an opportunity to bring my idea into reality so I decide to make a website that well sort some of these problems

## 1.2. Motivations and Challenges

There is nothing that is easy in life to do if we will not take it in easy way as we know there are millions website of E-commerce and performing their well and for us to make E-commerce website was more challenging because the question mostly rise was why customer will use your website they have already committed to some of their trusted websites so facing this question was challenging for us but we got a supervisor who boosted us up to start building this project because our mythology was deferent from others we was connecting people by giving them a

online stores available for them by nearby . The most motivation sentence that I was hearing from my supervisor is that a powerful man is a man who builds his dream into reality not keeps the dreams in dreams.

### **1.3. Goals and Objectives**

Our goal is to provide products to customer on door step with suitable price with less shipping cost and to provide a platform to other to open their shop on our website And our objectives are to create a website that will not only provide shopping services online but also will link customer to nearby store or shops. Our other objective is to provide a best thing to customer in time and in budget.

### **1.4. Literature Review/Existing Solutions**

There are millions website performing the best to customers by providing refund services and granted product. They insure customer to buy thing form this platform again because of quality. Most of all E-commerce website are performing same feature but having some deferent features. So according to these websites they are fulfilling customer needs. They are giving a good solution to customers. There are many websites like:

- **AliExpress**

It is an online retail service based in China that is owned by Alibaba Launched in 2010, it is made up of small businesses in China and elsewhere, such as Singapore that offer products to international online buyers. It is the most visited e-commerce website in Russia and was the 10th most popular website in Brazil It facilitates small businesses to sell to customers all over the world, and one can find almost anything for sale. It is more accurately compared to eBay as sellers are independent, using AliExpress as a host to sell to consumers and other businesses.

- **Daraz.pk**

In 2012, Daraz started its operations as an online fashion retailer in Pakistan. It changed

strategy and moved towards becoming a general online marketplace in November 2014. Daraz Group launched operations in Bangladesh and Myanmar in January 2015. In September 2015, Daraz.pk secured \$55 million in the first round of funding in order to continue operations in Pakistan, Bangladesh, and Myanmar. 60% of the investment comes from the former Daraz investor, Asia Pacific Internet Group (APACIG), with the remaining 40% of the investment amount from the CDC Group, a state-owned Development Finance Institution (DFI) in the UK with nearly £3.4 billion under management. Daraz Group was already operating in the South Asian regions of Pakistan, Bangladesh, and Myanmar. In July 2016, Daraz began operations in Sri Lanka and Nepal.

- **Goto.pk**

It is Pakistani online website which provides different kind of products like phone, devices, grocery and home appliances sports and fitness etc. it provides customers to place order within a city and out of city.

**Problems:**

As we have discussed about of these websites are servicing almost the same things both of these website and many more website are time consuming and money as well. At these platform customer needs to wait much longer for receive his/her order. And he/she pays more shipping cost because of distance. So according to these we come to know that Customer is facing time and money problems here.

**Solutions:**

Store to Door is the platform which decreases the delivery time and shipping cost. It is collections of store/shops in each city of the country and world. It makes customer to buy thing with less time and money.

## 1.5. Gap Analysis

All Store To Door website are giving a best solution to customer needs by providing products in door step with refund service and by giving discounts to customers but there is still a problem for customer because he have to wait for one or two weeks to get his order and if he want quick order he have to pay extra for this. According to us was the mean gap between customer and websites so we started to find out the best solution for this problem

## 1.6. Proposed Solution

As discussed above there is a mean problem for customer that is time and money.it saves time and money. Roads free from rush. Easy to use interface. Outstanding value for the injured and disable people. Great for male planning and budgeting

## 1.7. Project Plan

### ✓ Planning

- Requirement gathering
- Finding functional and non-functional requirement
- Choosing tools and techniques
- Choosing architecture
- People
  1. Project manager
  2. Project team
  3. End user
  4. Others

### ✓ Designing

- Architecture
- Front end
- Back end

### ✓ Implementation

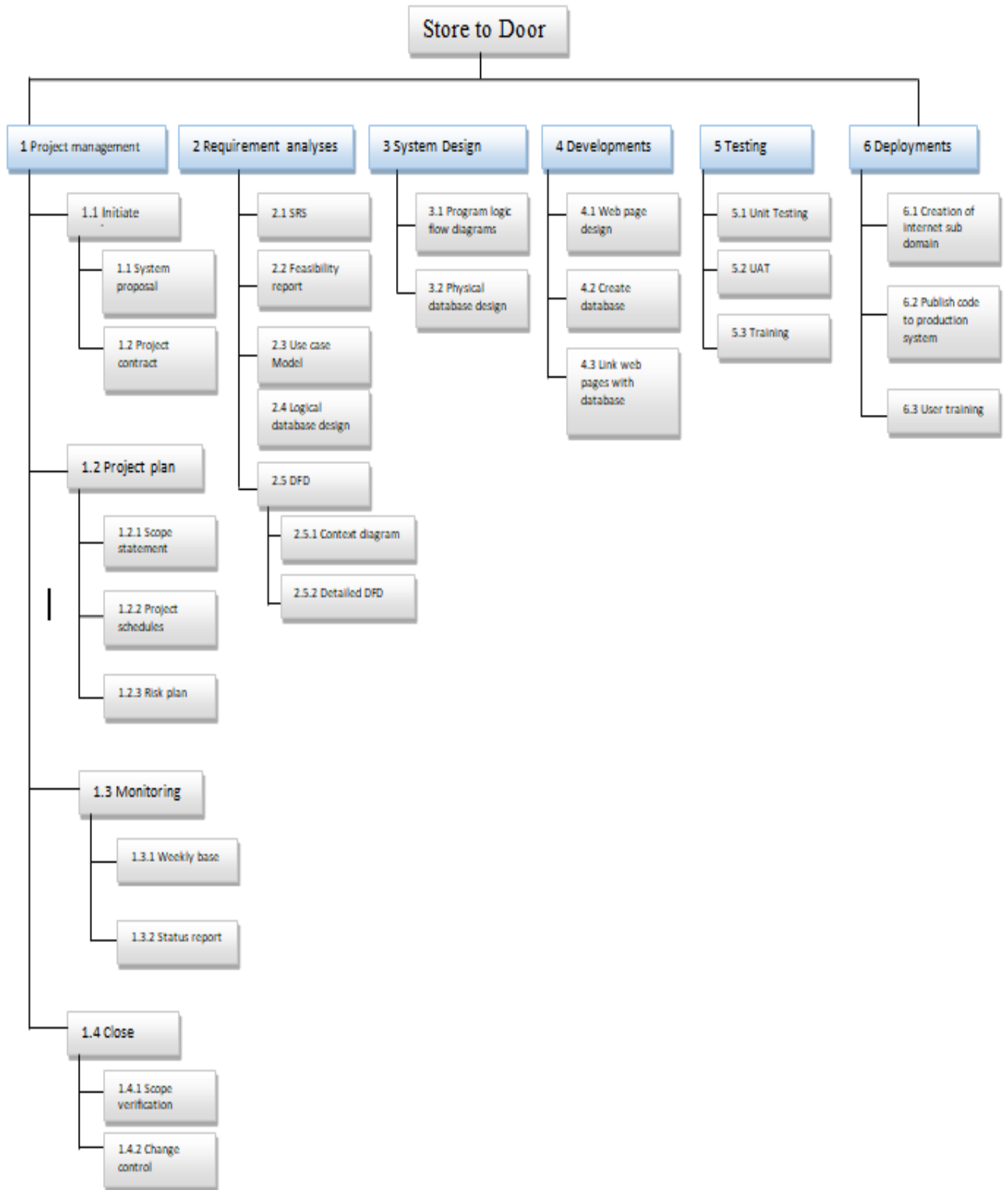
- Coding front end
- Coding back end

✓ Testing

- Black box testing
- White box testing
- Gray box testing
- Website testing

Maintenance

### 1.7.1. Work Breakdown Structure



### 1.7.2. Roles & Responsibility Matrix

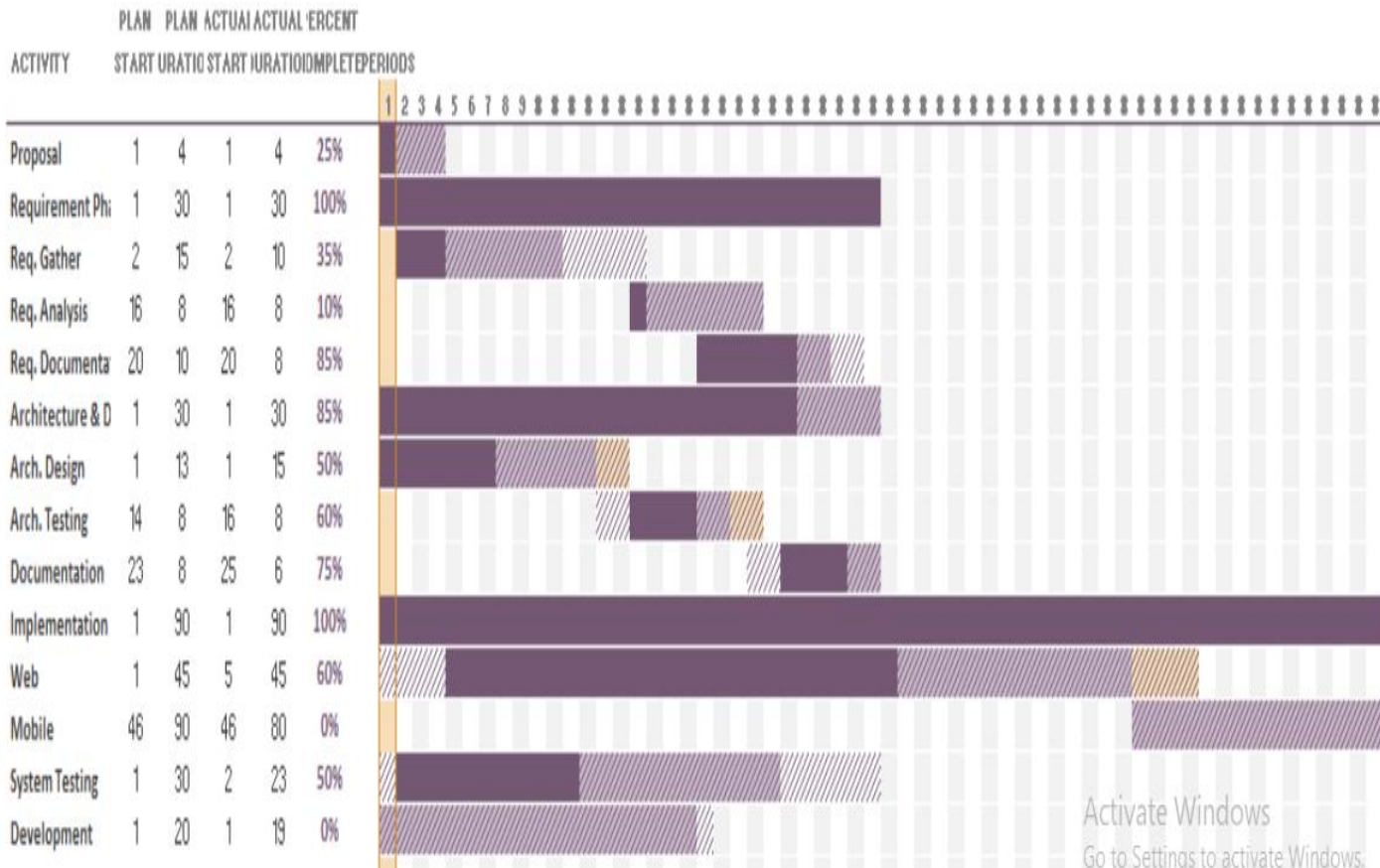
WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	System proposal			1	M Bilawal
2	Project contract	1.1	System proposal	3	M Usman
3	Scope statement	1.2 & 1.1	Project contract, System proposal	3	M Javid
4	Project schedules	1.1	System proposal	3	M Bilawal
5	Risk plan	1.1	Project proposal	6	M Bilawal
6	Monitoring			4	M Usman
7	SRS	1.1	Project proposal	6	M Bilawal
8	Feasibility report	2.1	SRS	3	M Javid
9	Use case model	2.1	SRS	2	M Bilawal
10	Logical database Design	2.1	SRS	5	M Usman
11	DFD	2.1 & 2.3	SRS ,Use case model	3	M Bilawal
12	Program login flow Diagram	2.5	DFD	3	M Usman
13	Physical database Design			3	M Usman
14	Web pages design	3.2	Physical database Design	6	M Bilawal
15	Create database	3.2	Physical database Design	4	M Usman

16	Link web pages with database	4.1 & 3.2	Web page design, Create database	6	M Usman
17	Unit testing	4.3	Linked web pages with database	5	M Bilawal

### 1.7.3. Gantt Chart

## Project Planner

Period High 1
Plan
Actual
% Complete
Actual (beyond plan)
% Complete (beyond plan)



Activate Windows  
Go to Settings to activate Windows.

## 1.8. Report Outline

- Introduction
- Background
- Motivations and challenges
- Goals and Objectives
- Literature Review/Existing Solutions
- Gap Analysis
- Proposed Solution
- Project Plan
  1. Planning
    - Requirement gathering
    - Finding functional and non-functional requirement
    - Choosing tools and techniques
    - Choosing architecture
    - Project sponsor
    - Project manager
    - Project team
    - End user
    - Others
  2. Designing
    - Architecture
    - Front end
    - Back end
  3. Implementation
    - Coding front end
    - Coding back end
  4. Testing
    - Black box testing

- White box testing
- Gray box testing
- Website testing

5. Maintenance

- Work Breakdown Structure
- Roles & Responsibility Matrix
- Gantt chart

# Chapter 2

## Software Requirement Specifications

## Chapter 2: Software Requirement Specifications

### 2.1. Introduction

In this section the documentation of this report is specified. It specifies the purpose, document convention, document scope and also provides a suggestion for the readers of the document.

#### 2.1.1. Purpose

The main goal or purpose of this document is to define the functionality and purpose of our website. This SRS (Software Requirement Specification) include all the specifications like design, interface. There requirements and components. The main purpose of this SRS is to provide a complete understanding that what we are building and achieving in this website.

#### 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 italicized text and includes various diagrams like UML, ERD, and DFD etc.

#### 2.1.3. Intended Audience and Reading Suggestions

Our project document has been done under the guidance of our supervisor and can be useful for any web development team. Students who want to write document related to this project. After reading this project developer and project managers completely understand with SRS of this website. This SRS document ensures the website's adherence of project manager to our original vision.

#### 2.1.4. Product Scope

The product scope with the passage of time technology is advancing to the better way. Over the past century in the field of website there is a big enhancement occurring. Peoples want the technology that is more interactive revolutionary. As we know there are millions website for

online shopping like amazon, AliExpress and daraz.pk. All of them are in trending now a day's that shows how the scope of shopping website is. So "Store to Door" is also a shopping website that will have same scope as others have. This document includes the formal requirements of our website "Store to Door". This document describes the system features and internal and external requirement specification of our website.

### **2.1.5. References**

[https://mycourses.aalto.fi/pluginfile.php/559128/mod\\_resource/content/1/Omnichannel%20Innovation%20Article%20HBR%202012.pdf](https://mycourses.aalto.fi/pluginfile.php/559128/mod_resource/content/1/Omnichannel%20Innovation%20Article%20HBR%202012.pdf)

## **2.2. Overall Description**

Store to Door a perfect solution for getting your order's delivered at your doorstep. Our system will make it easy for our customers as it will pick accurate coordinates of the shop, to pick up the order and the address of their place where they have to get it delivered. To make this system work, one must provide the required information, which will include the person's name, his phone number and his address. While placing a delivery a prompt message will be shown if the user wants to deliver on the usual address or another. Moreover, it will provide convenient and an easy to use interface to make it understandable for people of every ages

### **2.2.1. Product Perspective**

This website replaces the previous big and expensive display. This is E-commerce website which provide online. There are different product categories of branded products the users can select any category and buy any product from the category by using of store to door. They can buy the product and check how product suit on them

### **2.2.2. Product Functions**

- Product categories management
- Product availability management
- Shopping management
- Cart management
- Payment management
- Delivering management

### 2.2.3. User Classes and Characteristics

Admin	Client/User
<ul style="list-style-type: none"> <li>• Login</li> <li>• Edit profile</li> <li>• Send Notification</li> <li>• Manage Order</li> <li>• Manage cart</li> <li>• Add categories</li> <li>• Update</li> <li>• Logout</li> </ul>	<ul style="list-style-type: none"> <li>• Register</li> <li>• Login</li> <li>• Edit profile</li> <li>• View notification</li> <li>• View categories</li> <li>• Place order</li> <li>• Select payment method</li> <li>• Cancel order</li> <li>• Logout</li> </ul>

### 2.2.4. Operating Environment

Server Side	Client/User Side
<p><b>Processor:</b> Intel® Xeon processor 3500 series</p> <p><b>SSD/HDD:</b> minimum 1TB space</p> <p><b>RAM:</b>32 GB</p> <p><b>Operating system:</b> window 7, 8.1, 10, Linux</p> <p><b>Data base:</b> Firebase</p> <p><b>Application:</b> XAAMP, phpMyAdmin, open CV</p>	<p><b>Processor:</b> minimum Intel® dual core</p> <p><b>HDD:</b> minimum 40GB space</p> <p><b>RAM:</b> minimum 4GB</p> <p><b>Operating system:</b> window 7, 8.1, 10, Linux</p> <p><b>Mobil Android/IOS Version:</b> All Latest versions</p> <p><b>RAM:</b> minimum 2 GB</p>

### 2.2.5. Design and Implementation Constraints

- The application will use React js, Node js, jQuery and HTML, CSS as main website technologies.
- HTTP and FTP protocols are used as communication protocols. FTP is used to upload the website application in live domain and the client can access it via HTTP protocol.

- Several types of validations make this web application secured.
- Since this Store to Door E-commerce system is a web-based application, internet connection must be established.
- The Store to door E-commerce system will be used from PCs and mobiles and will function via internet or intranet in any web browser.

### **2.2.6. User Documentation**

System shall provide an online hierarchical system in our web-based application that describes and illustrate all the system functionalities. The first time a new user accesses the system and on user to practice ordering meals using static tutorial menu.

### **2.2.7. Assumptions and Dependencies**

- Each center user has account created and authenticated by admin. This Website can be Accessible within company's internet and other user can see the all details about the data.  
Each user has to first login itself to present him/her after entry in system. This will be done automatically. No user can share their username and password to each other.
- There is no limitation in the operating system in which Store to Door Ecommerce system will work. However, the Store to Door E-commerce system and the database will work on a server that needs to be always online. Users can access the system with any internet browser

## **2.3. External Interface Requirements**

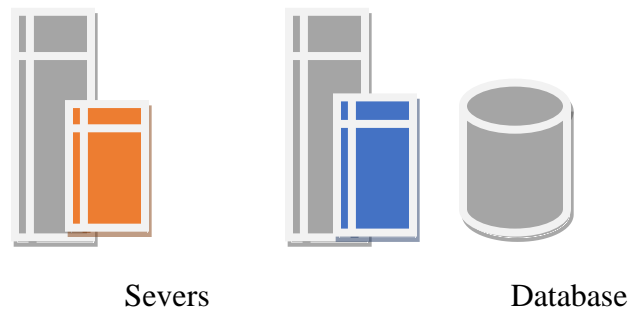
### **2.3.1. User Interfaces**

- All the users will see the same page when they enter in this website. This page asks the users a username and a password.
- After being authenticated by correct username and password, user will be redirect to their corresponding profile where they can do various activities.

- The user interface will be simple and consistent, using terminology commonly understood by intended users of the system. The system will have a simple interface, consistent with standard interfaces, to eliminate the need for user training of infrequent users.

### 2.3.2. Hardware Interfaces

- No extra hardware interfaces are needed.
- The system will use standard hardware and data communication resources.
- This includes, but is not limited to, general network connection at the server/hosting site, network server, and network management tools.
- As we know, websites have servers to store the data to be used in the website, so here will be only server interfaces and hardware used in the servers to store the data.



In supported devices, we tested our website on the latest version of Chrome and on an OPPO A7 Android device to check the responsiveness and working of the website. Here is the demo:

### 2.3.3. Software Interfaces

“Store To Door” has been developed using a series of website development tools:

- Notepad ++
- Sublime
- W3 schools
- Bootstrap
- XAMPP server

### 2.3.4. Communications Interfaces

As we know this online shopping website so internet is basic need of this website but it also need a strong sever to speed up its availability. Website is an online platform so there is needed to make it secure to communicate so we will use HTTP that is good platforms which prevent other to interact with sever and the website. Communication interface can be tablets, android mobiles LCDs and laptops etc.

## 2.4. System Features

These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. It specifies the application functionality that the developers must build into the product to enable users to accomplish their tasks.

### 2.4.1. System Feature 1

#### Login/logout

REQ-SF1-1	The system should allow admin to login to the system using their username and password.
REQ-SF1-2	System should allow user to login to the system using their username and password.
REQ-SF1-3	The system shall allow the manager to create new user account
REQ-SF1-4	The system shall allow admin to change account password.
REQ-SF1-5	The system shall allow user to change account password.
REQ-SF1-6	The system shall allow staff to logout.

---

REQ-SF1- The system shall allow admin and user to logout.

7

---

## 2.4.2. System Feature 2

### Order and payment

---

REQ-SF2-1 The system must allow the client to register for order

REQ-SF2-2 The system shall allow the customer to view detail description of particular Order.

REQ-SF2-3 The system must notify on selection of unavailable product while Order.

REQ-SF2-4 The system must allow the client to select specific product using different search category while Order.

REQ-SF2-5 The system must view list of available item/ product during Order.

REQ-SF2-6 The system shall allow the admin to view Order made by client.

REQ-SF2-7 The system shall allow the client/user to cancel Order.

REQ-SF2-8 The system shall allow the client/user to pay Order fee which method user like.

---

## 2.4.3. System Feature 3 (and so on)

### Store to Door MODELS

---

REQ-SF3-1 The system must allow the client to view Store to Door model

REQ-SF3-2 The system shall allow the admin to manage and update Store to Door models

---

## 2.5. Other Nonfunctional Requirements

### 2.5.1. Performance Requirements

- There should be reasonable, if any, latency or lack during website usage.
- The system response time for every instruction conducted by the user must not exceed more than a minimum of 5 seconds.
- The system should have adequate performance when executing user is input and should be able to provide response within a short time span.

### 2.5.2. Safety Requirements

- Our website will never harm the device but user must be careful while using the website.
- It is better to keep on checking for the latest updates on the websites.

### 2.5.3. Security Requirements

- The website will prompt the user for upgrading and downloading the new features updated by the developer.
- The system provides username and password to prevent the system from unauthorized access.
- The subsystem should provide a high level of security and integrity of the data held by the system.
- Only authorized person (admin) can gain access website secured page on the system; and only users with valid password and username can login to view user's page.

### 2.5.4. Software Quality Attributes

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible.

At

the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

## **2.6. Other Requirements**

### **Availability**

- The website will be available at all time so that user can facilitate 24/7.
- Also, in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that business process is not severely affected.

### **Reliability**

- We will want to maintain the state of the user in case of unexpected crashes to avoid data loss.
- Right now, we are not making this product to be self-healing from the unexpected crashes; however, we will log the crashes for future improvement.

### **Usability**

- The website interface should be easily understandable.
- Guides through the work flow.
- The learn ability of the user for the system will be better.
- The interface will be comfortable to use in terms of colors, icons and images.

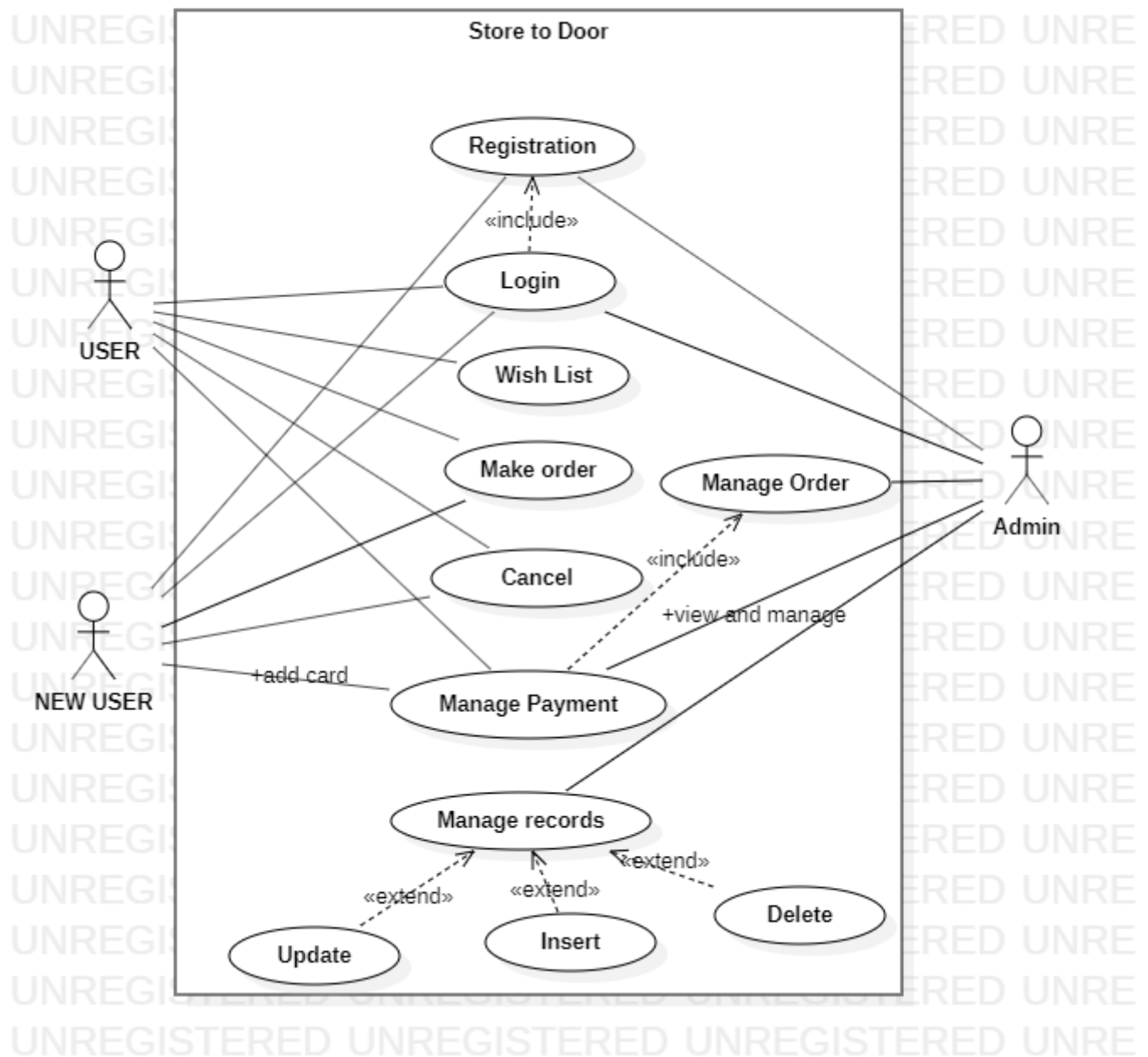
# Chapter 3

## Use Case Analysis

## Chapter 3: System Analysis

In a part we are discussing about how our system will work and perform, following are the use case diagram that will help to understand the whole system

### 3.1. Use Case Model



## 3.2. Use Case Descriptions

### 1. Use Case: Registration

**Primary Actors:** Customer and Admin

**Goal in context:** create an account successfully

**Precondition:**

- Click on website link
- Click on join us button at navigation bar

**Triggers:** Customer and Admin needs to create an account on website

**Scenario:**

- Open website link
- Click on join us button placed at navigation bar
- Fill complete sign up form
- Click on sign up

**Exception:** Customer or admin can enter wrong data in form fields

### 2. Use Case: login

**Primary Actors:** Customer and Admin

**Goal in context:** To go on home page of the website and perform their actions

**Precondition:**

- Customer or Admin should have an account

**Triggers:** Customer and Admin needs to perform their actions

**Scenario:**

- Open website link
- Click on login button placed at navigation bar
- Enter authentic user name and password
- Click on login

**Exception:** User name or password can be wrong or forgotten

### 3. Use Case: Manage Order

**Primary Actors:** shop keeper (Admin)

**Goal in context:** to register his shop on the website and sell his products

**Precondition:**

- Logged in
- Have shop registration number

**Triggers:** Customer and Admin needs to perform their actions

**Scenario:**

- Click on Manage order button place at navigation bar
- Fill the needed requirements in form
- Accept term and conditions of the website
- Click on submit

**Exception:** shop keeper can enter wrong information and can miss any fields to fill

### 4. Use Case: Make Order

**Primary Actors:** Customer

**Goal in context:** Make order to buy the desired products

**Precondition:**

- Customer shop selects the product
- Select color
- Select size
- Add quantity
- click on buy button

**Triggers:** Customer needs to buy the products

**Scenario:**

- Open website link
- Click on shop now button place on the product displayed
- Select size

- Select color
- Select quantity
- Click on buy button
- Fill the form
- Click on submit

**Exception:** Invalid address or information filled in shipping form

**5. Use Case:** cancel order

**Primary Actors:** Customer

**Goal in context:** Customer wants to cancel his order that he was placed before

**Precondition:**

- Order was placed already

**Triggers:** Customer needs to cancel the placed order that he has place before

**Scenario:**

- Click on my order
- Select order id
- Click on cancel

**Exception:** Customer can select wrong id of order to cancel

**6. Use Case:** Manage Order

**Primary Actors:** Customer and Admin

**Goal in context:** customer wants to buy many products and he want to add select Products to cart

**Precondition:**

- Logged in
- Choses product
- Click on shop now button
- Click on Cart button
- Select size color and quantity

**Triggers:** Customer adds products to cart successfully

**Scenario:**

- Chose product and click on shop now
- Select size color and quantity
- Click on cart button

**Exception:** Customer can select wrong size and color or quantity

**7. Use Case:** manage records

**Primary Actors:** Admin

**Goal in context:** To manage records (update, delete, insert) and manage transaction of Record

**Precondition:**

- Logged in
- Designation admin
- Click on manage records
- Select specific records
- Make changes in it

**Triggers:** Admin is need to manage records and can make changes

**Scenario:**

- Open website link
- Click on login button placed at navigation bar
- Enter authentic user name and password
- Click on login
- Click on manage records
- Select specific records
- Make changes
- Click on save

**Exception:** Admin can delete or update wrong records by mistake or can delete all Records by mistakes

### 3.3. Use Case Specification

#### Registration

**ID: 1**

#### Description

Customer and Admin needs to create an account on website

**Primary actors:** Customer and Admin

#### Pre-condition:

- Click on join us button

#### Main flow:

- Open website link
- Click on join us button placed at navigation bar
- Fill complete sign up form
- Click on sign up

#### Post-condition:

- User should go on Homepage
- Use should perform allowed functions of the website

#### Login

**ID: 2**

#### Description

Customer must require to login to place order and admin can give rights to access website

**Primary Actor:** Customer and Admin

#### Pre-condition:

- Account must be created

#### Main flow:

- Open website link
- Click on login button placed at navigation bar
- Enter authentic user name and password
- Click on login

**Post-condition:**

- User should go on Homepage
- Use should perform allowed functions of the website

**Make Order**

**ID:3**

**Description:** to register his shop on the website and sell his products

**Primary Actors:** member and customer

**Pre-condition**

- Logged in
- Have shop registration number

**Main Flow**

- Click on open shop button place at navigation bar
- Fill the needed requirements in form
- Accept term and conditions of the website
- Click on submit

**Post – Conditions:**

- Display pop up message “Your request for open a shop has been sent to admin”
- Redirect to homepage

**Place order**

**ID:4**

**Description:** place order to buy the desired products

**Primary actors:** Customer

**Preconditions:**

- Click on shop now button
- Select size ,color and quantity
- Click on buy button
- Fill form

- Click done

**Main flow:**

- Open website link
- Click on shop now button place on the product displayed
- Select size
- Select color
- Select quantity
- Click on buy button
- Fill the form
- Click on submit

**Post – Conditions:**

- pop up message “your order is submitted successfully”
- Redirect to homepage
- Order info should be goes in my orders list

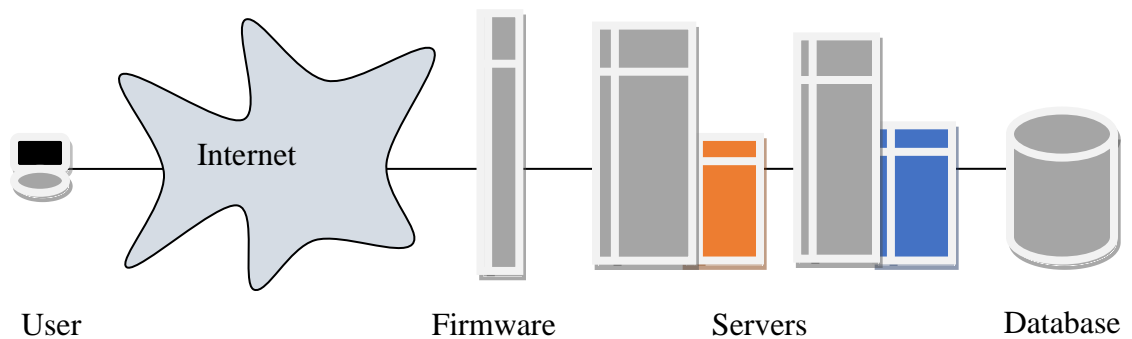
# Chapter 4

## System Design

## Chapter 4: System Design

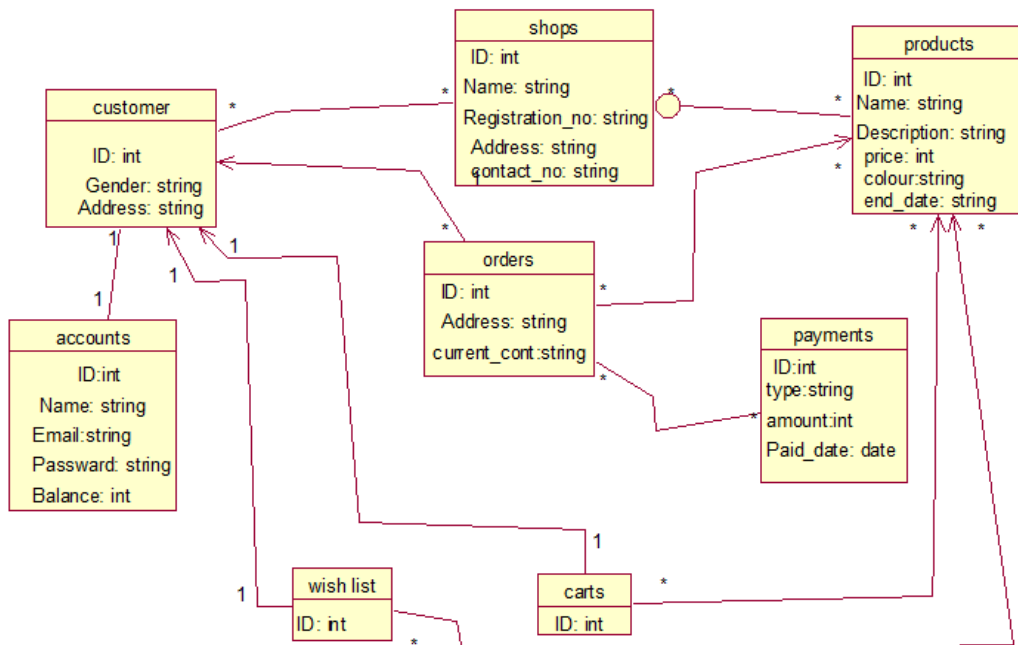
This chapter describes about the system design of our website it will include the different types of diagrams. These are eleven different types of diagrams of software engineering which we are using for detailed design of our website project “Store to Door”

### 4.1. Architecture Diagram

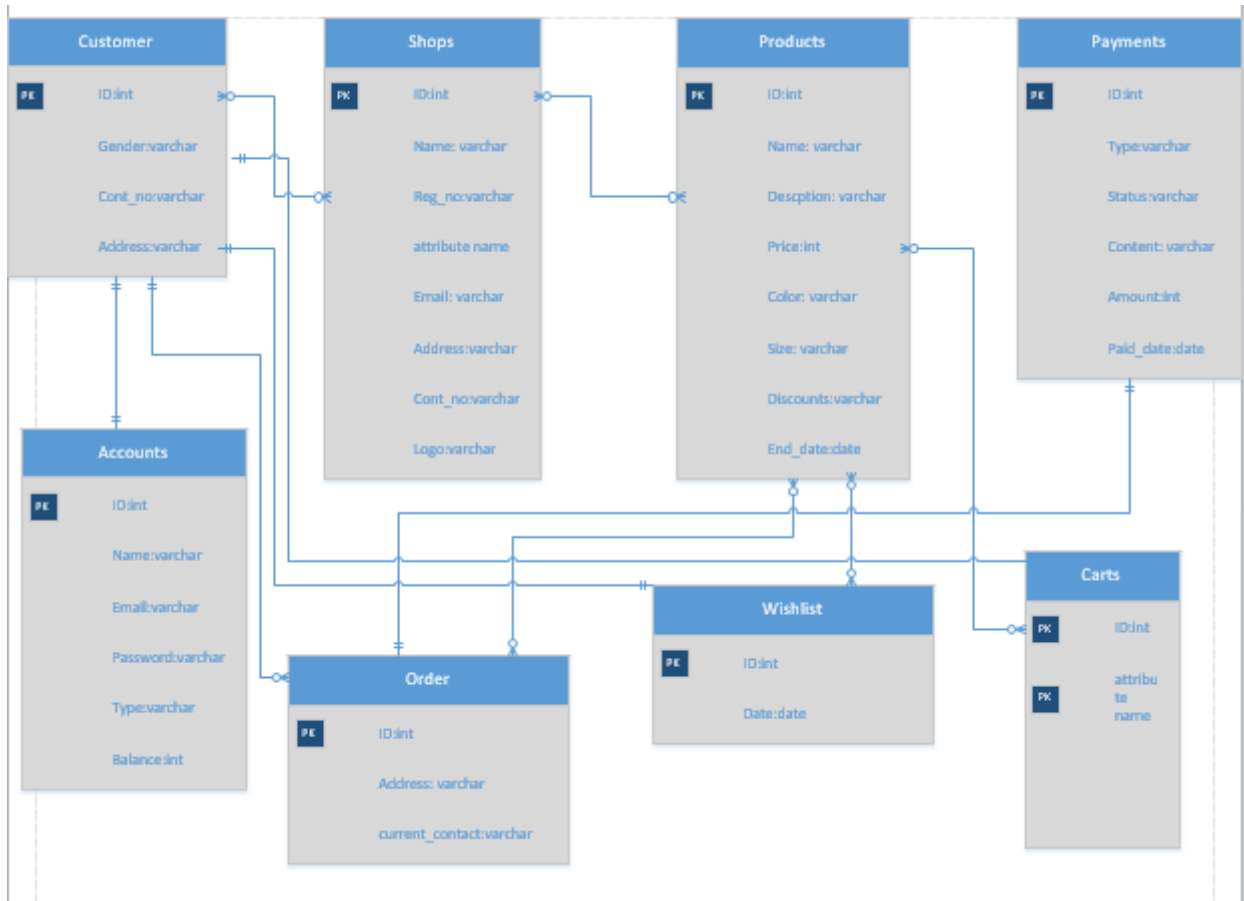


## 4.2. Domain Model

It is now clear now how our system will work. If a user wants to buy a product he will order an item, delivery men will deliver an order and payment process will be on cash and through credit card

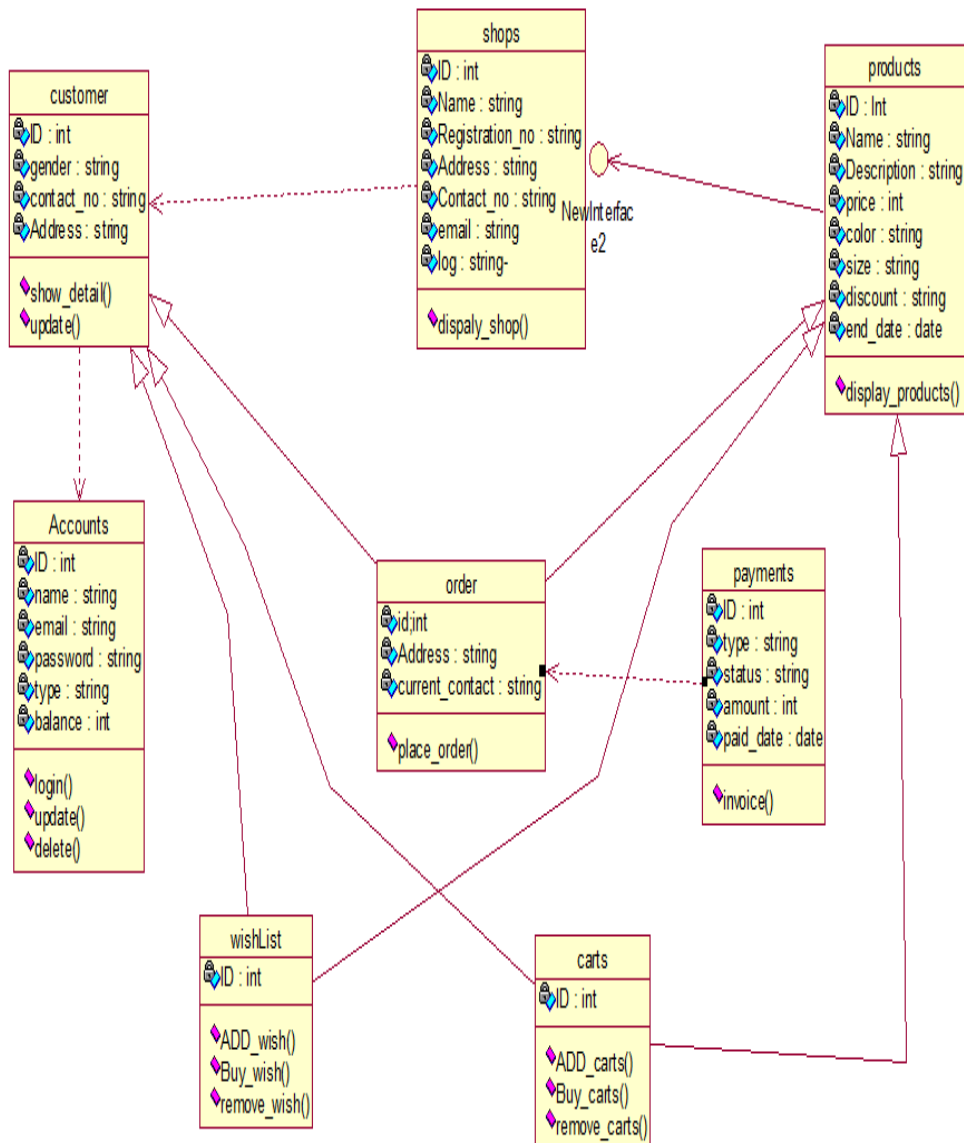


### 4.3. Entity Relationship Diagram with data dictionary

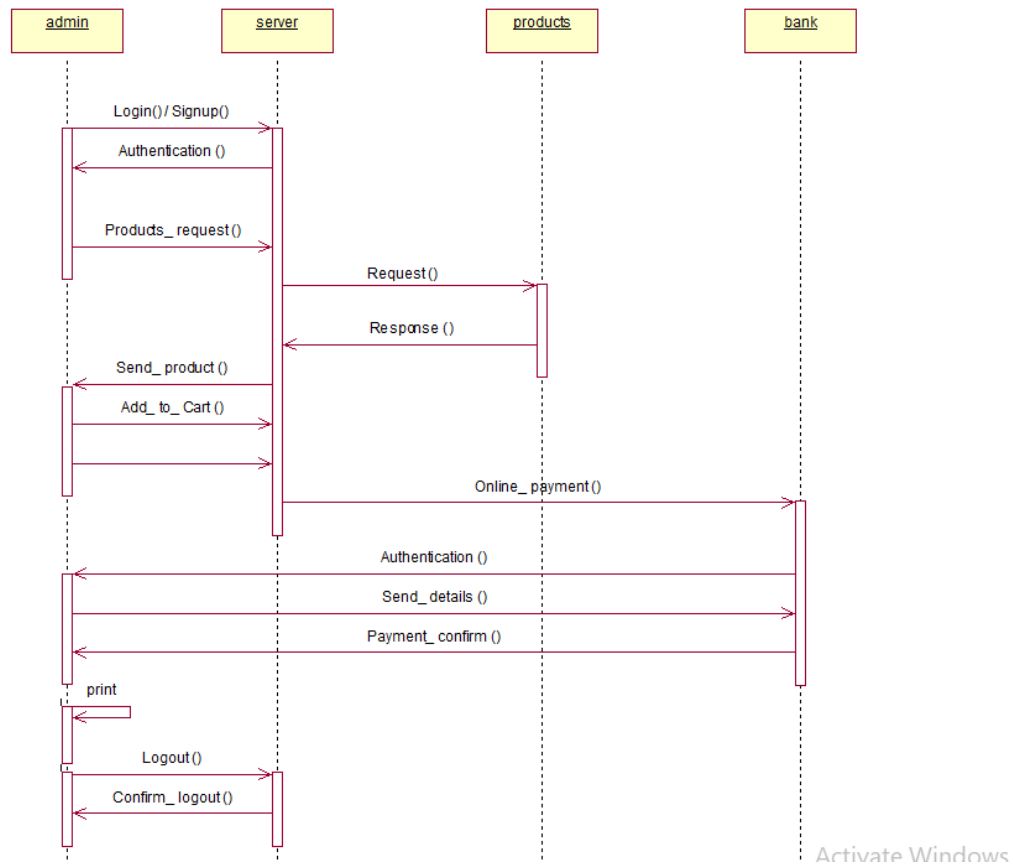


## 4.4. Class Diagram

If a user wants to buy a product user will order an item. User id password is must require for login in their accounts. Users can view items and view status after order an item.



## 4.5. Sequence / Collaboration Diagram



## 4.6. Operation contracts

### Operation contracts

**Operation:** login (username, Password)

**Cross Reference:** Use case: login

**Pre – Conditions:**

- Account must be created

**Post – Conditions:**

- User should go on Homepage
- Use should perform allowed functions of the website

**Operation:** Registration (email, name, password)

**Cross Reference:** Use case: signup

**Pre – Conditions:**

- Click on join us button

**Post – Conditions:**

- User should go on Homepage
- Use should perform allowed functions of the website

**Operation:** Open\_ shop (email, name, password, contact, registration\_ no, logo)

**Cross Reference:** Use case: Open\_ shop

**Pre – Conditions:**

- User should be logged in
- Click on open shop button

**Post – Conditions:**

- Display pop up message “Your request for open a shop has been sent to admin”
- Redirect to homepage

**Operation:** Manage\_ order (email, name, country, city, Street\_ address, Contact, p\_ size, P\_ color, quantity, Image)

**Cross Reference:** Use case: Place\_ order

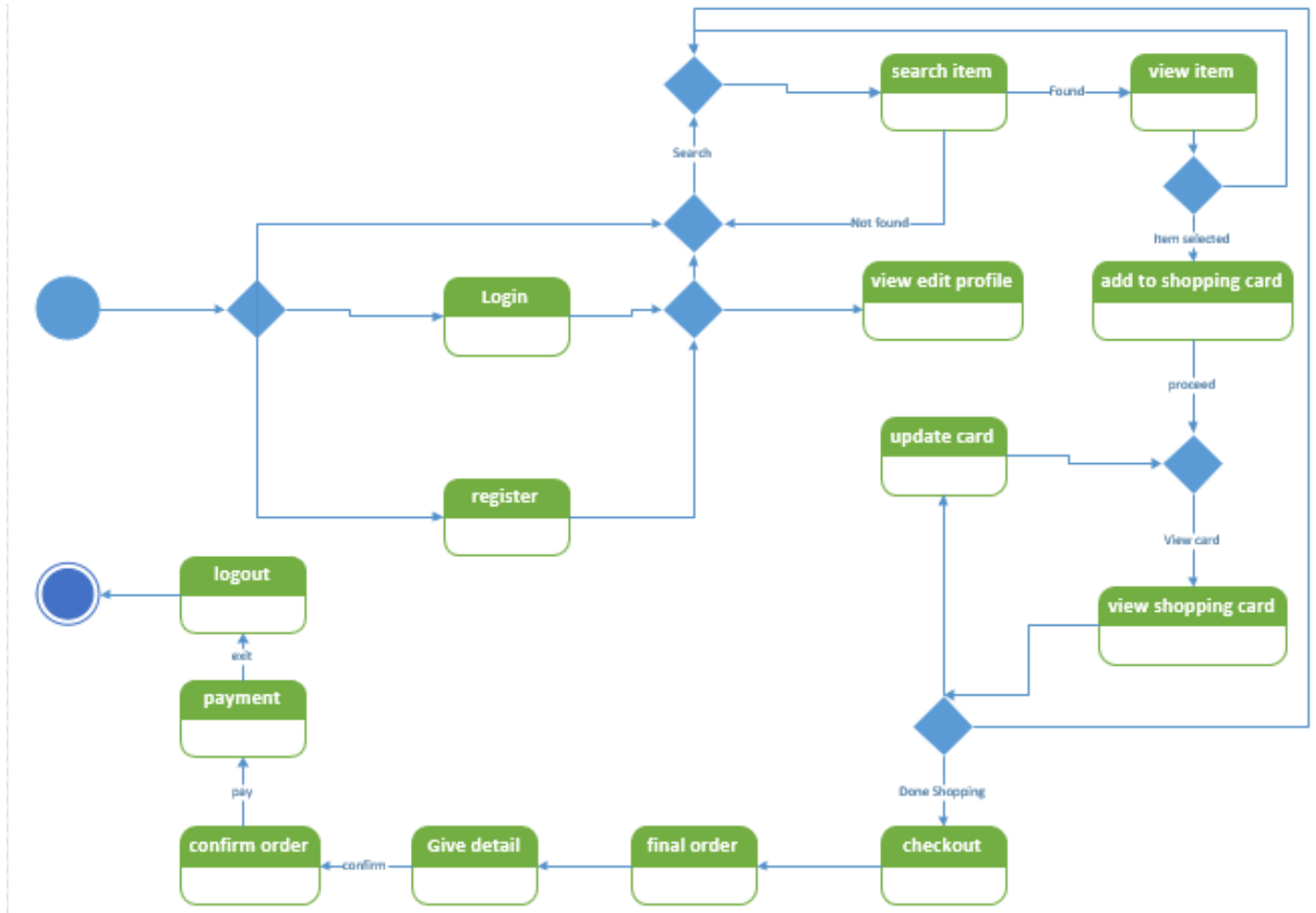
**Pre – Conditions:**

- Click on shop now button
- Select size ,color and quantity
- Click on buy button
- Fill form
- Click done

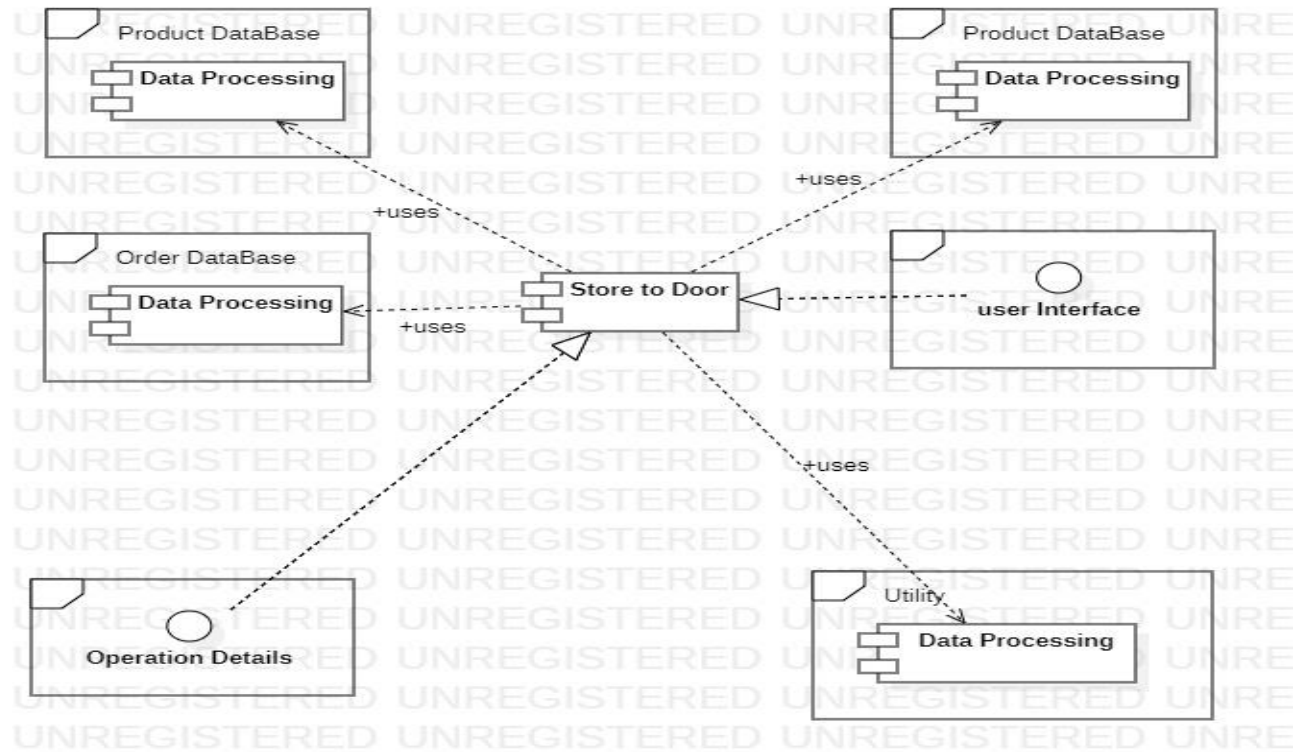
**Post – Conditions:**

- pop up message “your order is submitted successfully”
- Redirect to homepage
- Order info should be goes in my orders list

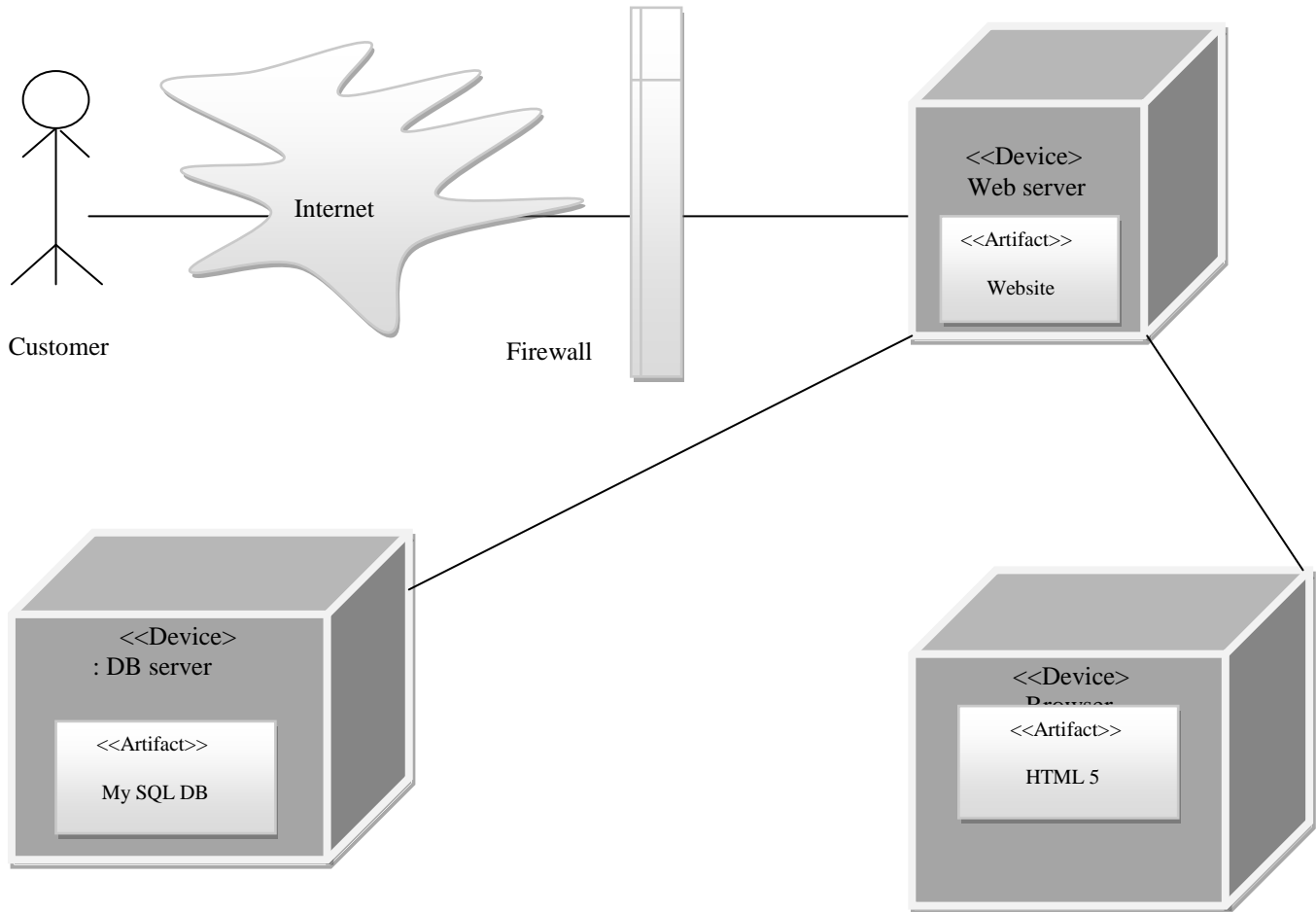
## 4.2. State Transition Diagram



### 4.3. Component Diagram



### 4.4. Deployment Diagram



## 4.1. Data Flow diagram

We are not using structural approach so we don't need to draw DFD

# Chapter 5

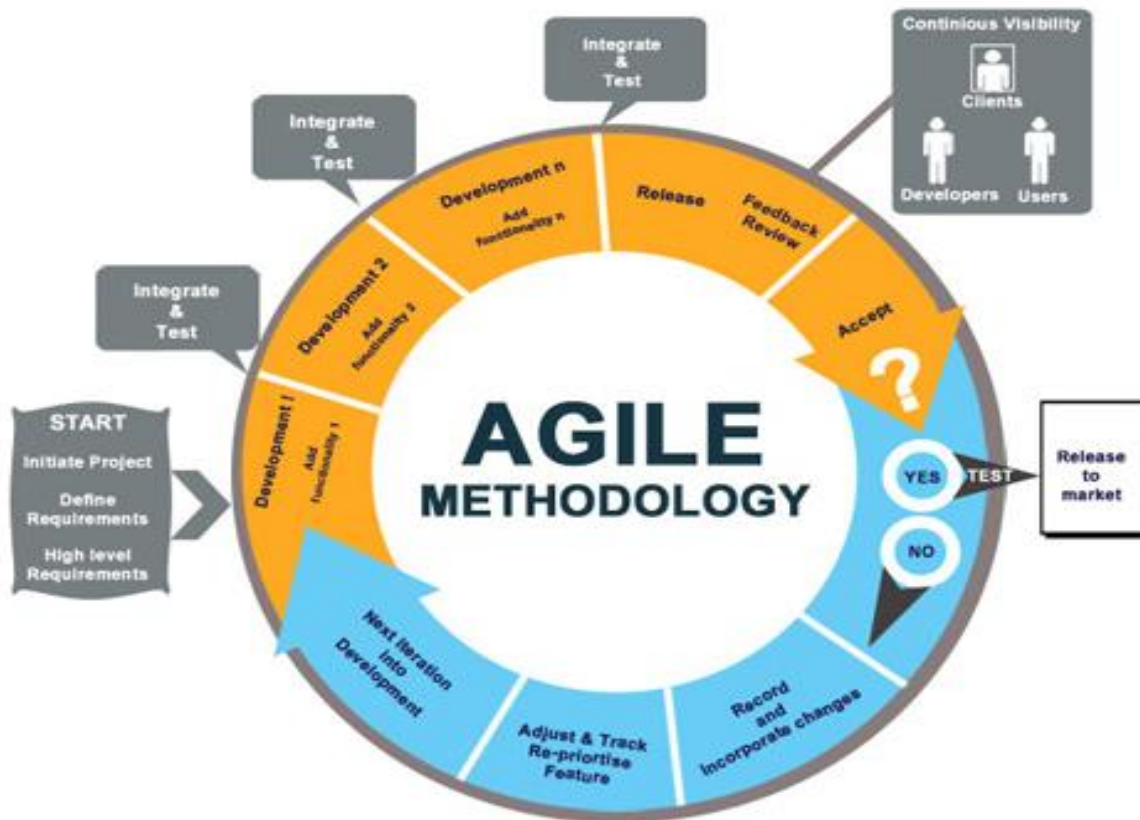
## Implementation

## Chapter 5: Implementation

This chapter covers the implementation of our project features and implementation flow control of our website this chapter also describes the components, libraries and web services used in our website project. The deployment environment, tools and techniques and coding standards are explained in this chapter that we are using in our website project.

### 5.1. Important Flow Control/Pseudo codes

- We use agile method software engineering technique for developing our project



## 5.1. Components, Libraries, Web Services and stubs

We have used basic components of notepad ++ Like

- Debugging
- Creating new files
- Macro to record video while coding
- Searching specific word in the code (Ctrl +F, F3, Shift+ F3)
- And more components that helped us to make coding easy

We did not use the libraries of notepad ++ but we have included some bootstrap libraries in the code that helped us to make our website responsive like

- <https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css>
- <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css>
- <https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js>
- And our own created files like CSS files and JS files

We did not use any web services directly but yes we have added some libraries that work when internet is connected like font-awesome some classes that create icon like menu icon and many more icons. Basic is the server if we include it in web service that is being used in our website and this is basic service that makes our website to function otherwise website is just offline. Web servers have been used only to make website only. HTTP links are also used to secure our website.

## 5.2. Deployment Environment

Web Application will be deployed on a web server supporting SQL server, application services and interface.

## 5.3. Tools and Techniques

### FRONT END:

HTML, CSS, JS and Bootstrap, React library

### BACK END:

PHP, React JS, Node JS, Java Script, Firebase

## 5.4. Best Practices / Coding Standards

### 5. Naming Conventions

6. The naming conventions in unity we used:
7. **Variables:** begins with the lowercase letters. The variables are used to store the website information about any aspects of any state of website
8. **Classes:** Classes are declared with the uppercase letter. The classes are the collection of functions.
9. **Functions:** Functions are starts with uppercase letters. Functions are collection of code and they are written once and they can be used as they need in program.
10. **Layouts:** The code by selecting File->Save As and give page a name
11. **Standards**



12. W3C standards are used to create the website.

## 5.1. Version Control

Notepad++ and XAMPP server provides us services to build our layouts and database as a team in a different ways but rather than using this we individually complete our tasks in our separate machines and the finally joined all this and build website in a one single machine in which our complete project is up and running. For further any update we fix this issue in the same way we did before.

# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

In this chapter both static and dynamics testing of online shopping system. The static testing is including reviewing the basic of the websites whereas the dynamic testing is based on actual code executions. Testing helps to reduce overall system development project.

**Features we tested were as follow:**

- To ensure that the website itself ran (dynamic)
- Login worked efficiently and consistently (dynamic)
- Accessing the web from multiple platforms to ensure cross compatibility (dynamic)
- Checking page load time (dynamic)

### 6.1. Use Case Testing

#### REGISTER USE CASE

<b>Suite ID</b>	RTS000
<b>Case ID</b>	RTC000
<b>Actor</b>	Users
<b>Case Summary</b>	Verify that the Registration form contains Username, First Name, Last Name, Password, Confirm Password, Email Id, Phone number, Date of birth, Gender, Location, Terms of use, submit (if you have account)
<b>Related Requirement</b>	-----
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> </ul>
<b>Procedure</b>	<ul style="list-style-type: none"> <li>• Go to site <a href="http://www.thestoretodoor.com/">http://www.thestoretodoor.com/</a></li> <li>• User should enter all personal information required in registration page.</li> </ul>

<b>Data</b>	Username, First Name, Last Name, Password, Confirm Password, Email Id, Phone number, Date of birth, Gender, Location
<b>Expected Result</b>	After entering all information system bring users to main home page.
<b>Status</b>	Pass

### Register Use Case Testing 2

<b>Suite ID</b>	RTS002
<b>Case ID</b>	RTC002
<b>Actor</b>	Users
<b>Case Summary</b>	Verify that case sensitivity of Username. And username already exist.
<b>Related Requirement</b>	-----
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> </ul>
<b>Procedure</b>	<ul style="list-style-type: none"> <li>• Go to site <a href="http://www.thestoretodoor.com/">http://www.thestoretodoor.com/</a></li> <li>• User should enter all personal information required in registration page.</li> </ul>
<b>Test Data</b>	Username
<b>Expected Result</b>	If username fulfil all term and conditions the system must register users and if the username already taken by other user and the sensitivity case not fulfill the term and condition notify user change user name and system also recommend username.
<b>Status</b>	Pass

### Login Use case

<b>Suite ID</b>	LTS101
<b>Case ID</b>	LTC101
<b>Actor</b>	Users
<b>Test Case Summary</b>	Check results on entering valid User Id & Password
<b>Related Requirement</b>	RTS000
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> <li>• User should be register and login to system Database</li> </ul>
<b>Procedure</b>	<ul style="list-style-type: none"> <li>• Go to site <a href="http://www.thestoretodoor.com/">http://www.thestoretodoor.com/</a></li> <li>• Enter valid Email</li> <li>• Enter valid Password</li> <li>• Click Submit</li> </ul>
<b>Data</b>	User name and password
<b>Expected Result</b>	User should login into system if user authorized.
<b>Status</b>	Pass

**Order use case**

<b>Suite ID</b>	OTS201
<b>Case ID</b>	OTC201
<b>Actor</b>	Authorized User
<b>Case Summary</b>	Check results if main Authorized user click on order an item.
<b>Related Requirement</b>	LTS101
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> <li>• User should be login into system database</li> </ul>

<b>Procedure</b>	<ul style="list-style-type: none"> <li>• Go to site <a href="http://www.thestoretodoor.com/">http://www.thestoretodoor.com/</a></li> <li>• Enter valid Email</li> <li>• Enter valid Password</li> <li>• Click Submit</li> <li>• Order item</li> </ul>
<b>Test Data</b>	Available Item for order
<b>Expected Result</b>	System must allow authorized user to order an item.
<b>Status</b>	Pass

#### Use Case for Searching an item/product

<b>Suite ID</b>	STS301
<b>Case ID</b>	STC301
<b>Actor</b>	User
<b>Case Summary</b>	Check results if user search proper and good an item.
<b>Related Requirement</b>	LTS101
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> <li>• User should be register and login to system Database</li> </ul>
<b>Procedure</b>	<ul style="list-style-type: none"> <li>• Go to site <a href="http://www.thestoretodoor.com/">http://www.thestoretodoor.com/</a></li> <li>• Enter valid Email</li> <li>• Enter valid Password</li> <li>• Click Submit</li> <li>• Navigate to "Search for a product" page then enter product name and hit enter</li> </ul>
<b>Data</b>	Searching bar
<b>Expected Result</b>	System should show user a Search product.

<b>Status</b>	Pass
---------------	------

#### Use Case for add item to cart

<b>Suite ID</b>	ACTS501
<b>Case ID</b>	ACTC501
<b>Actor</b>	User
<b>Case Summary</b>	Check results if user should select desired item and add item to cart.
<b>Related Requirement</b>	LTS101
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> <li>• User should be register and login to system Database</li> </ul>
<b>Procedure</b>	<ul style="list-style-type: none"> <li>• User should be login into system.</li> <li>• Select desired item.</li> <li>• Click on add to cart button.</li> </ul>
<b>Data</b>	Product feeds
<b>Expected Result</b>	Item gets added to cart successfully amount and count are incremented accordingly
<b>Status</b>	Pass

#### Use Case for Payment

<b>Suite ID</b>	PST601
<b>Case ID</b>	PCT601
<b>Actor</b>	Users
<b>Case Summary</b>	Check what happens if payment process fails
<b>Related Requirement</b>	OTS201
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Access on website</li> <li>• User should be register and login to</li> </ul>

	system Database
<b>Procedure</b>	<ul style="list-style-type: none"> <li>• Go to site <a href="http://www.thestoretodoor.com/">http://www.thestoretodoor.com/</a></li> <li>• Login to system.</li> <li>• Users Order a product</li> </ul>
<b>Test Data</b>	Pay fee of order item/product.
<b>Expected Result</b>	If payment process fail system must notify users.
<b>Status</b>	Pass

## 6.2. Equivalence partitioning

### Registration EP-01

<b>Name Equivalence partitioning</b>		
<b>Invalid</b>	<b>invalid</b>	<b>Valid</b>
Special character	Numerical	a-z, A-Z

<b>username Equivalence partitioning</b>		
<b>invalid</b>	<b>invalid</b>	<b>Valid</b>
Special character	Numerical	a-z, A-Z, 0-9

**Email Equivalence partitioning**

invalid	invalid	Valid
Special character	A-Z, >=10	a-z, 0-9

**Password Equivalence partitioning**

invalid	invalid	Valid
Special character <=40	>=7	a-z, A-Z, 0-9, special characters

**username Equivalence partitioning**

invalid	invalid	Valid
Special character	Numerical, >=10	a-z, A-Z, 0-9


**Password Equivalence partitioning**

invalid	invalid	Valid
Special character <=40	>=7	a-z, A-Z, 0-9, special characters

**Add Items Equivalence partitioning**

invalid	invalid	Valid
.tiff, JPEG	Zip file	PNG, JPG

**6.3. Boundary value analysis**

  
Member Registration

Name

Username

Email Address

Password

**REGISTER**

Already have account? [Sign In](#)

**Login**

Email Address

Password

**SIGN IN**

## 6.4. Unit testing

```

File Edit Selection Find View Goto Tools Project Preferences Help
OPEN FILES
x index.php
5
6 ?>
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <title></title>
11 <link rel="stylesheet" type="text/css" href="style.css">
12 <link href='http://fonts.googleapis.com/
css?family=Montserrat:400,700%7CPT+Serif:400,700,400italic' rel='stylesheet'
13 >
14 <link href="https://fonts.googleapis.com/css?family=Montserrat|Open+Sans"
rel="stylesheet">
15 <style>
16 * {
17 margin: 0;
18 padding: 0;
19 }
20 body {
21 background-color: grey;
22 font: normal 13px/1.5;
23 font-family: 'Roboto Condensed', sans-serif;
24 color: #333;
25 }
26 wrapper {

```

## 6.5. Integration testing

Summary Report

Name:

Comments:

Write results to file / Read from file

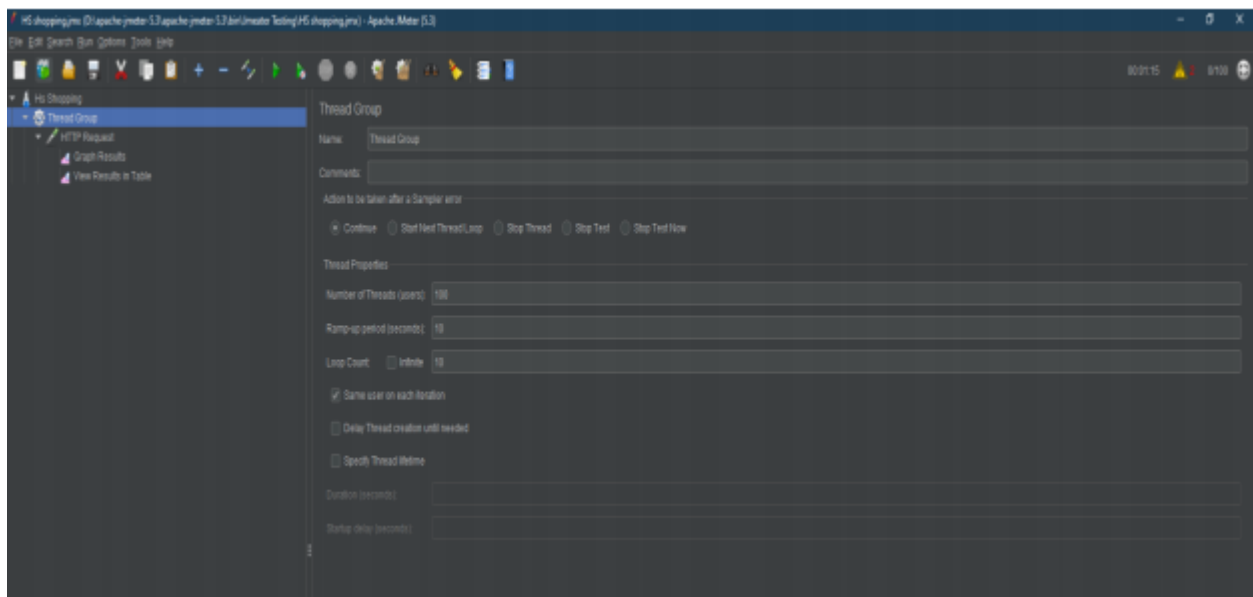
Filename:    Log/Display Only  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received Kbytes	Sent Kbytes	Avg. Bytes
HTTP Request	2000	3316	575	35535	2946.29	53.15%	5.9/sec	254.36	1.36	44416.0
TOTAL	2000	3316	575	35535	2946.29	53.15%	5.9/sec	254.36	1.36	44416.0

## 6.6. Performance testing



## 6.7. Stress Testing



# Chapter 7

## Summary, Conclusion and Future Enhancements

## Chapter 7: Summary, Conclusion & Future Enhancements

### 7.1. Project Summary

Store to Door is an E-commerce website that will have its own store. It will service online shopping process of using daily thing in life and will perform basic functionality of online shopping. It will have some new features like any store if they want to open their shop on this website they can open. They will be strictly announced to have order delivering facilities on their own behalf. So this thing will bring easiness for customer to get their order from desired store by nearby shop within one hour. If customer wants to order a product which belongs to other society then he will wait for one day to receive his/her order. These are some common thing in this website.

### 7.2. Achievements and Improvements

The biggest achievements here are that we were able to compile all we learned in four years of studying software engineering and apply it to this project. We learned software architecture design techniques, UML modeling, project management, software quality assurance, testing and much more, and were able to apply it all in this project. We learnt during this project, new languages, frameworks, libraries and data handling techniques, all that will be useful for us in our futures.

### 7.3. Critical Review

### 7.4. Lessons Learnt

- Using a multi-platform language for developing your application works well
- A project should be well managed and tasks completed within their timeframe, else they will affect the upcoming task

### 7.5. Future Enhancements/Recommendations

There is always room for further improvement, enhancements will keep coming

# Appendices

## Appendix A: User Manual

### 1-Home

Home view for user



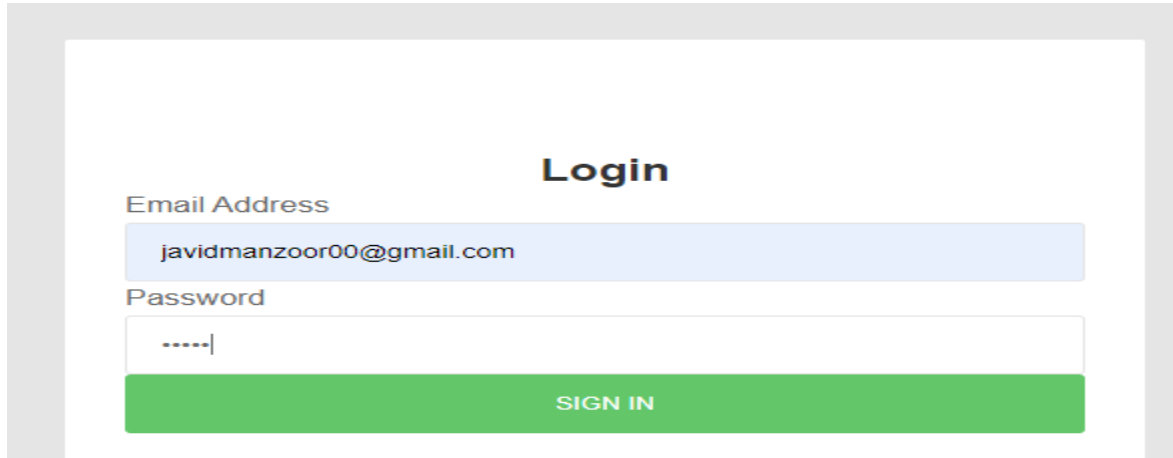
### 2-sighup

Sign up for new user

A screenshot of the 'Member Registration' form. At the top center is a blue icon of a person with a plus sign and the text 'Member Registration'. Below this are five input fields: 'Name' with the value 'javed', 'Username' with 'manzoor', 'Email Address' with 'javidmanzoor00@gmail.com', and 'Password' with masked characters '----|'. A green 'REGISTER' button is positioned below the password field. At the bottom, there is a link: 'Already have account? [Sign In](#)'.

### 3-Login

Log in for old user

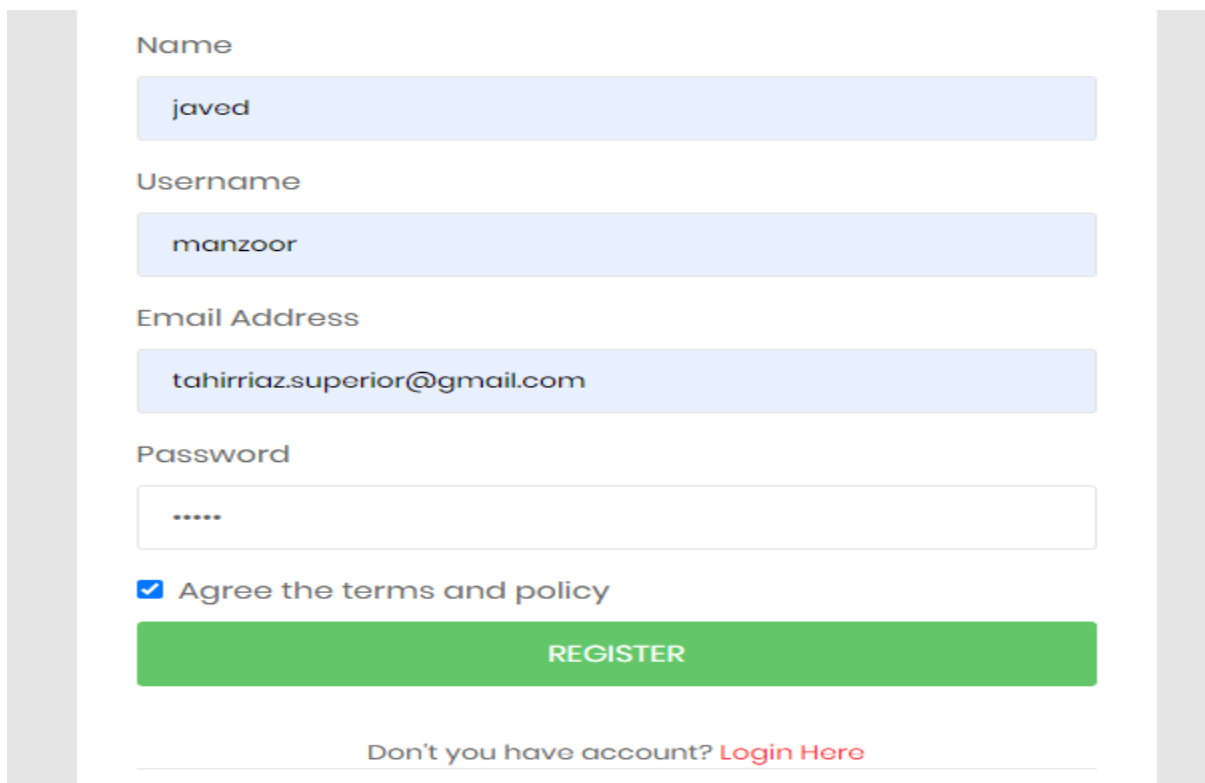


The screenshot shows a login form with the following fields and elements:

- Title:** Login
- Email Address:** A text input field containing the email address "javidmanzoor00@gmail.com".
- Password:** A text input field containing masked characters ".....|".
- Button:** A green button labeled "SIGN IN".

### 4-Admin sign up

Sign up for authorized admin

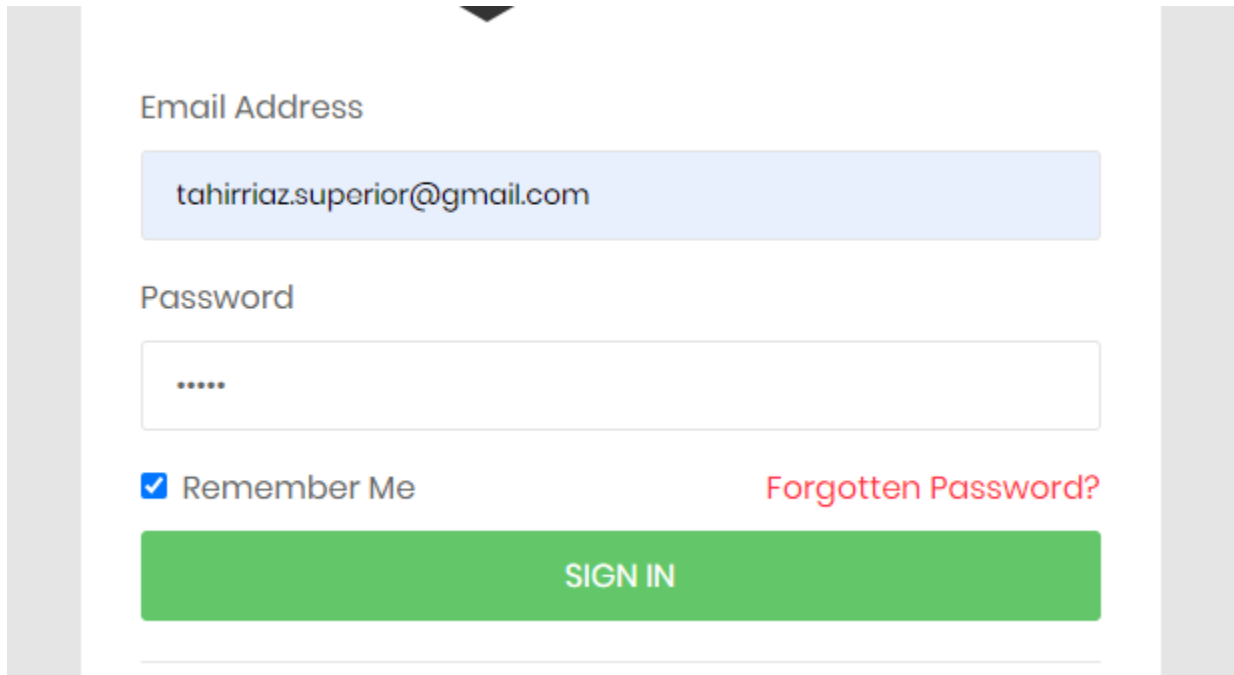


The screenshot shows an admin sign-up form with the following fields and elements:

- Name:** A text input field containing the name "javed".
- Username:** A text input field containing the username "manzoor".
- Email Address:** A text input field containing the email address "tahirriaz.superior@gmail.com".
- Password:** A text input field containing masked characters ".....".
- Checkbox:** A checked checkbox labeled "Agree the terms and policy".
- Button:** A green button labeled "REGISTER".
- Link:** A link labeled "Don't you have account? [Login Here](#)".

## 5- Admin log

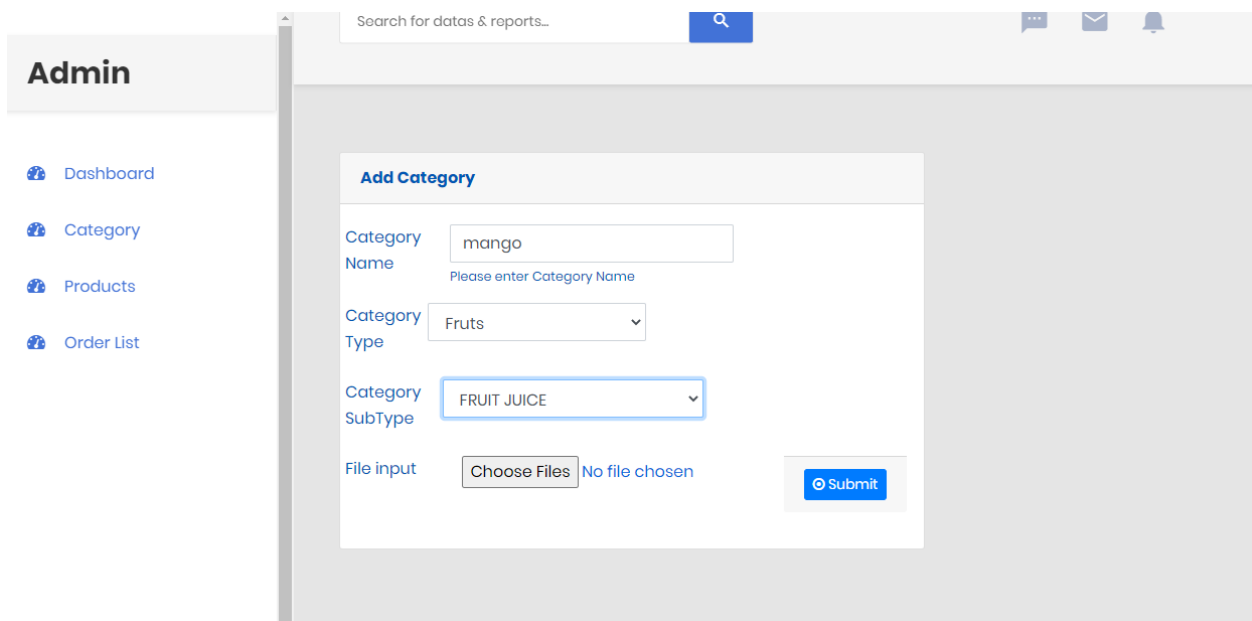
Log in for authorized admin



The image shows a login form for an authorized admin. It features two input fields: 'Email Address' containing 'tahirriaz.superior@gmail.com' and 'Password' with masked characters. Below the password field is a checked 'Remember Me' checkbox and a red link for 'Forgotten Password?'. A large green 'SIGN IN' button is positioned at the bottom of the form.

## 6- add category

Category add by admin



The image displays an admin dashboard with a sidebar menu and a main content area. The sidebar includes links for 'Dashboard', 'Category', 'Products', and 'Order List'. The main content area shows a search bar and a notification icon. The 'Add Category' form is the central focus, containing the following fields:

- Category Name:** A text input field with 'mango' entered and a placeholder 'Please enter Category Name'.
- Category Type:** A dropdown menu with 'Fruts' selected.
- Category SubType:** A dropdown menu with 'FRUIT JUICE' selected.
- File input:** A 'Choose Files' button and the text 'No file chosen'.
- Submit:** A blue 'Submit' button.



## 6.1- view category

### View category by admin

**Admin**

Search for datas & reports...

- Dashboard
- Category
- Products
- Order List

ID	Name	Type	Sub Type	Logo	Edit	Delete
2	vages	Vages	BELL PEPPER		Edit	Delete
4	ghee	Grocery	Oil and Ghee		Edit	Delete

## 7-add product

### add product by admin

**Admin**

Search for datas & reports...

- Dashboard
- Category
- Products
- Order List
- Admin

**Add Products**

Product Name:   
Please enter Product Name

Price:

Quantity:

Product Type:  Product SubType:






File input:  No file chosen

## 7.1 view product

### View product by admin

Admin		Search for datas & reports...						
Dashboard								
Category								
Products								
Order List								

ID	Name	Price	Quantity	Type	Sub Type	Image	Edit	De
2	bell pepper	100	2	Vages	BELL PEPPER		Edit	De
4	ghee	200	2	Grocery	Oil and Ghee		Edit	De
5	coke	100	45	Grocery	FRUIT JUICE		Edit	De
6	apple	100	30	Fruts	STRAWBERRY		Edit	De
7	nescafe	350	45	Grocery	STRAWBERRY		Edit	De
8	lipton	100	20	Grocery	0		Edit	De

## 8. Order List

### Order list view by admin




Admin		Search for datas & reports...		
Dashboard				
Category				
Products				
Order List				

ID	Name	Phone	Address	User ID
1	javed	03067404402	shairshah clony B- block lahore	12

## 9. shop



User can buy anything

 <p>ghee 200</p> <input type="text" value="1"/> <input type="button" value="Add to Cart"/>	 <p>coke 100</p> <input type="text" value="1"/> <input type="button" value="Add to Cart"/>	 <p>apple 100</p> <input type="text" value="1"/> <input type="button" value="Add to Cart"/>
--	--	---

## 10. Add to cart

Add to cart by user

300

 <p>ghee 200</p> <input type="text" value="1"/> <input type="button" value="Remove"/>	 <p>coke 100</p> <input type="text" value="1"/> <input type="button" value="Remove"/>
--	---

## 11. check out

Check out by user finish shopping and given there address

Billing Address	Payment
<p>Full Name</p> <input type="text" value="javed"/>	<input checked="" type="checkbox"/> jazz cash <input type="checkbox"/> cash on delivery
<p>phone</p> <input type="text" value="03074536234"/>	
<p>Address</p> <input type="text" value="shairshah clony B-block lahore"/>	
<input type="button" value="check-out"/>	

**Domain name of our website.**

**www.thestoretodoor.com**

# Reference and Bibliography

## Reference and Bibliography

- [1] UML 2 for Dummies by Michael Jesse Chonoles and James A. Schardt ISBN:0764526146  
Hungry Minds © 2003
- [2] Unified Modeling Language User Guide, The (2 ed.). Addison-Wesley. 2005. p. 496. ISBN  
0321267974. , See the sample content, look for history
- [3] "ISO/IEC 19501:2005 - Information technology - Open Distributed Processing – Unified  
Modeling Language (UML) Version 1.4.2". Iso.org. 1 April 2005. Retrieved 7 May 2015
- [4] Hubert Tardieu, Arnold Rochfeld and René Colletti La methode MERISE: Principes et outils  
(Paperback -1983)
- [5] "Will UML 2.0 Be Agile or Awkward?" (PDF). Retrieved 22 September 2011.
- [6] Andreas Zender (1997) Advanced Concepts, Life Cycle Models and Tools for Object-  
Oriented Software Development. p.122
- [8] "UML". Omg.org. Retrieved 10 April 2014.
- [9] Hubert Tardieu, Arnold Rochfeld and René Colletti La methode MERISE: Principes et outils  
(Paperback - 1983)

<https://Daraz.pk/>

<https://notepad-plus-plus.org/download/v7.6.2.html>

[https://filehippo.com/download\\_xampp/](https://filehippo.com/download_xampp/)

<http://stackoverflow.com>

<https://codepen.io>

<https://youtube.com>

<https://www.wikipedia.org/>

<https://www.w3schools.com/js/>

<https://www.amazon.com/>

[https://www.w3schools.com/bootstrap/bootstrap\\_list\\_groups.asp](https://www.w3schools.com/bootstrap/bootstrap_list_groups.asp)