# ONLINE SHOPPING SYSTEM

**SOFTWARE ENGINEERING PROJECT REPORT**

**[Submitted in Partial Fulfilment]**
**As a part of the curriculum of**

## B.Sc. (H) COMPUTER SCIENCE


**Submitted by:**
**DIYA GANDOTRA (21075570029)**
**GYANA KARN (21075570034)**
**MANSI ROHILLA (21075570064)**
**NIDHI GUPTA (21075570072)**
**SHUBHANGI CHAND (21075570114)**

***B.Sc. (H) COMPUTER SCIENCE***
**14, Shyama Prasad Mukherjee College for Women University of Delhi 57, North Ave, West Punjabi Bagh, Punjabi Bagh, Delhi, 110026**

# ACKNOWLEDGEMENT

It gives me immense pleasure to present you to this Online Shopping System, WRAPSHOP. We were lucky to get enormous support from extremely talented people, who deserve our great gratitude.

Firstly, we would like to thank our teacher and guide, Mr. Lavkush Gupta who gave his valuable suggestions and ideas whenever we needed them. Also, he encouraged us to work on this project tirelessly by giving us numerous consultations.

We are also grateful to our parents for their constant support, guidance and providing us the necessary resources for the project.

Lastly, we would like to thank our classmates for their valuable suggestions for the betterment of this project and everybody who has helped us directly or indirectly in completion of this project.

We are immensely grateful to everyone involved in this project a without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

With sincere thanks,

DIYA GANDOTRA
GYANA KARN
MANSI ROHILLA
NIDHI GUPTA
SHUBHANGI CHAND

# CERTIFICATE

This is to certify that the project entitled, "**ONLINE SHOPPING SYSTEM - WRAPSHOP**", has been submitted by Diya Gandotra, Gyana Karn, Mansi Rohilla, Nidhi Gupta and Shubhangi Chand in partial fulfilment of the requirements of Bachelor of Computer Science(Honours.) embodies the work done by them during, semester IV of their course under the supervision of **Mr. Lavkush Gupta, Assistant Professor,** Department of Computer Science, Shyama Prasad Mukherji College for Women, University of Delhi.

Mr. Lavkush Gupta
(PROJECT GUIDE)

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1.PROBLEM STATEMENT

This document is intended to outline the features of an online shopping system so that it can act as a guide for programmers and a software validation document for potential customers. The goal of the web application for the online shopping system is to offer a comprehensive solution to both merchants and customers.

It will make it possible for Operators to put up online stores that customers can explore and buy from without having to physically visit the stores. Operators can manage products from the admin panel.

A system administrator will be able to maintain a variety of shop categories and request new shops using the administrator module.
Customers residing in various parts of the world need to be able to purchase various product categories from the online shopping system. All products will be shown on the website in a classified manner. Customers can browse through any product to learn more about it, including its pricing, and place an order. Orders must include shipping and billing information. The order amount can be paid for by customers using a credit card, debit card, net banking, or cash (cash on delivery).

The system's primary goal is to make it easy for people to explore and purchase goods from anywhere in the world, expanding the potential for business.

General description

Customers can browse through the shops using the online shopping system program, vendors can build up their own online stores, and system administrators can accept or deny requests to add new stores and keep lists of store categories. Additionally, the developer is creating an online store to handle the merchandise in the stores and assist clients in making purchases without physically visiting the stores. Clients can use our website to sell their

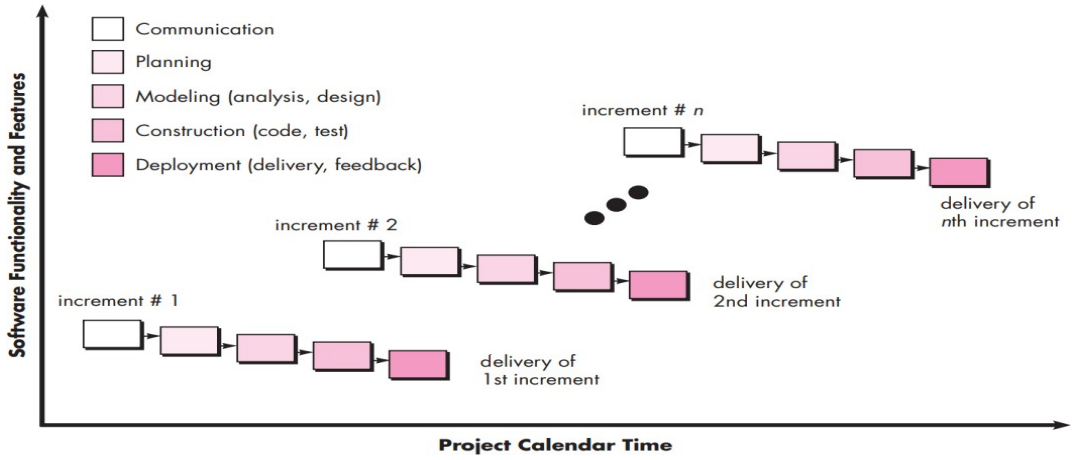
goods.

# 2.PROCESS MODEL

Model best suited for this project is Incremental Process model.
We have used the incremental model as it combines elements of linear and parallel process flows. It generates working software quickly and early during the software lifecycle. This model is more flexible and less costly to change scope and requirements. It is easier to test and debug during a smaller iteration. In this model, the customers can respond to each built. Also, functionality can be refined and expanded in the later stages in the later software releases. The user can visualise the software before the completion of the entire project in order to evaluate and provide feedback. We are using this model as requirements are completely understood, however, small changes can be incorporated.

Advantages of Incremental Model
o Errors are easy to be recognised.
o Easier to test and debug
o More flexible.
o Simple to manage risk because it handled during its iteration.
o The Client gets important functionality early.

Fig : 2.1

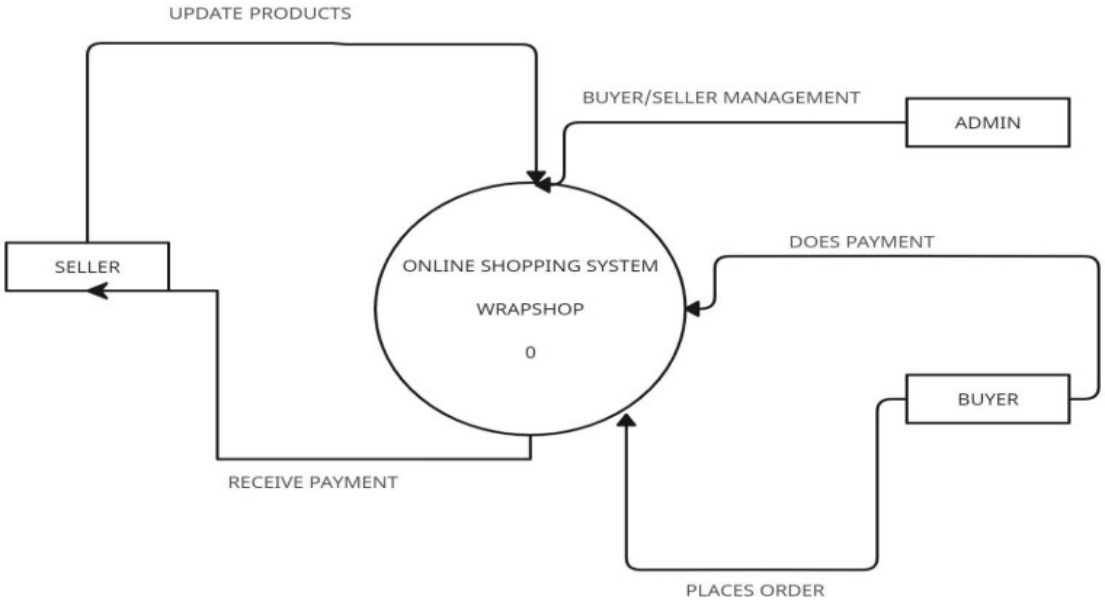


# 3. REQUIREMENT ANALYSIS

## 3.1 DATA FLOW DIAGRAMS

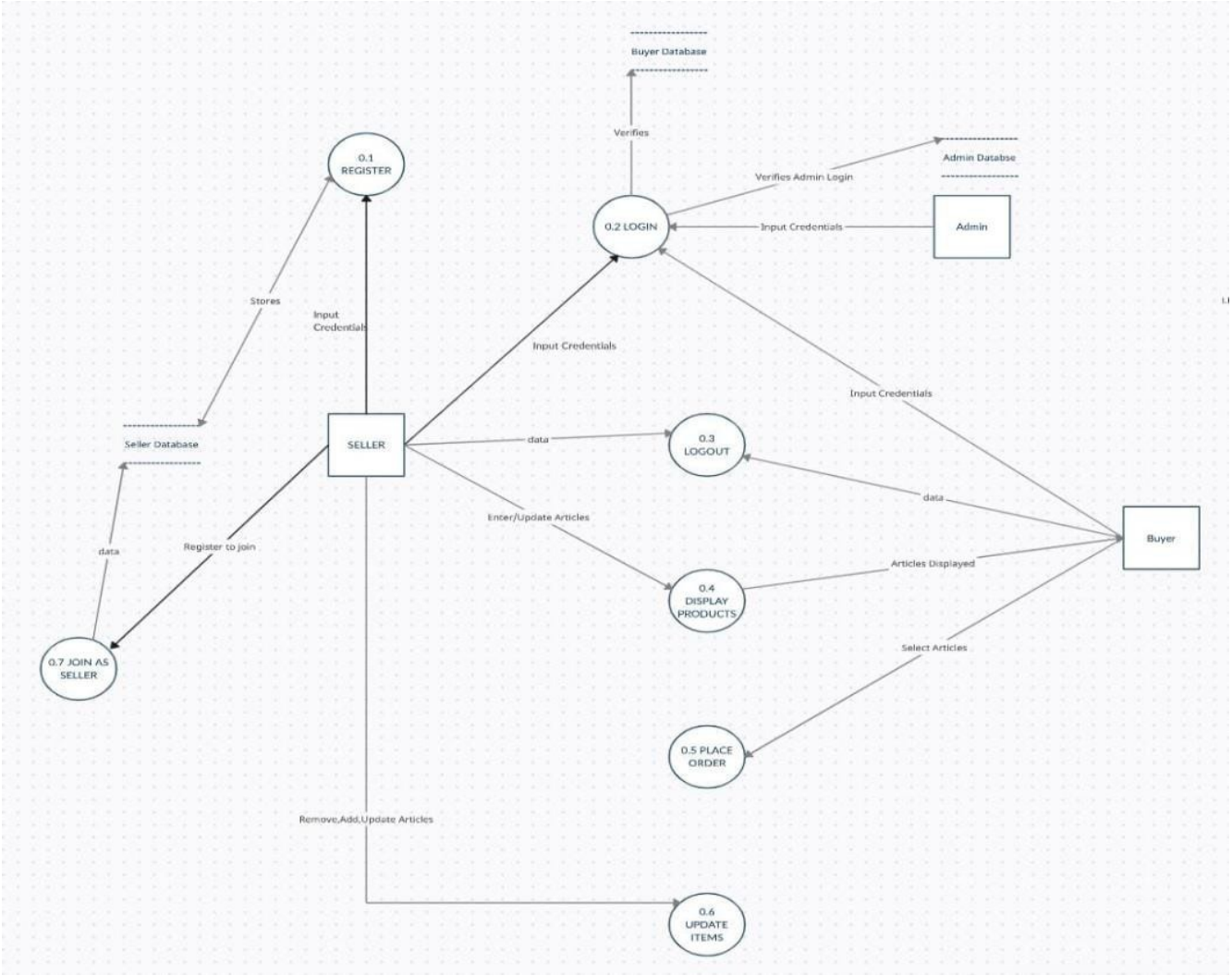
### 3.1.1 CONTEXT LEVEL DIAGRAM

Fig 3.1 Level 0 DFD



# 3.1.2 LEVEL 1 DFD

Fig 3.2 Level-1 DFD



# 3.1.3 LEVEL 2 DFD

Fig 3.3 level 2 dfd REGISTER

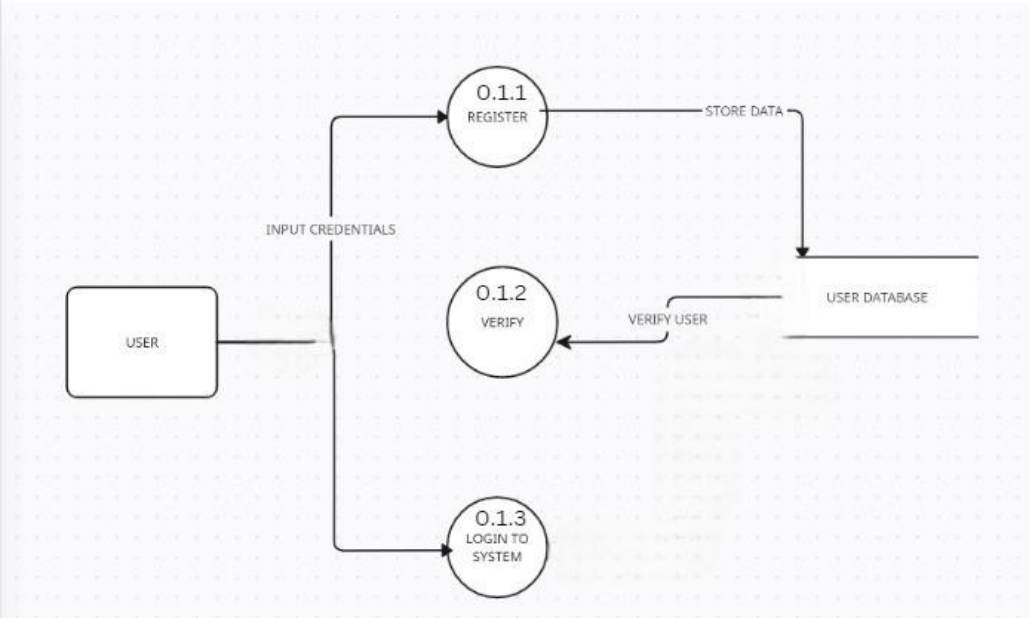



Fig 3.4 level 2 dfd LOGIN

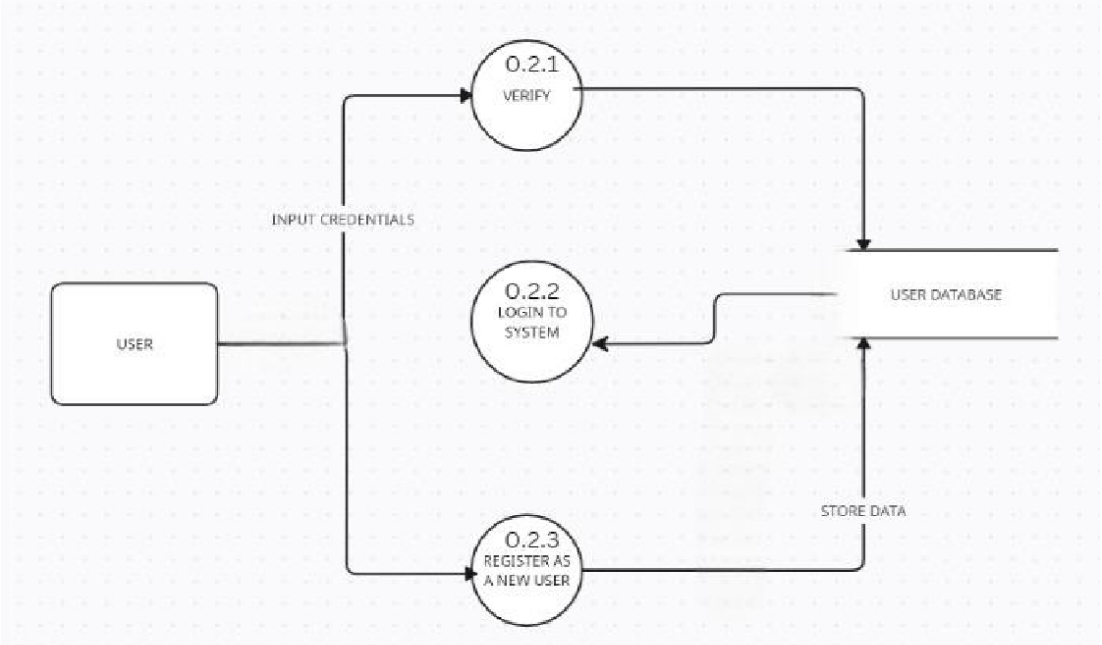


Fig 3.5 level 2 dfd LOGOUT

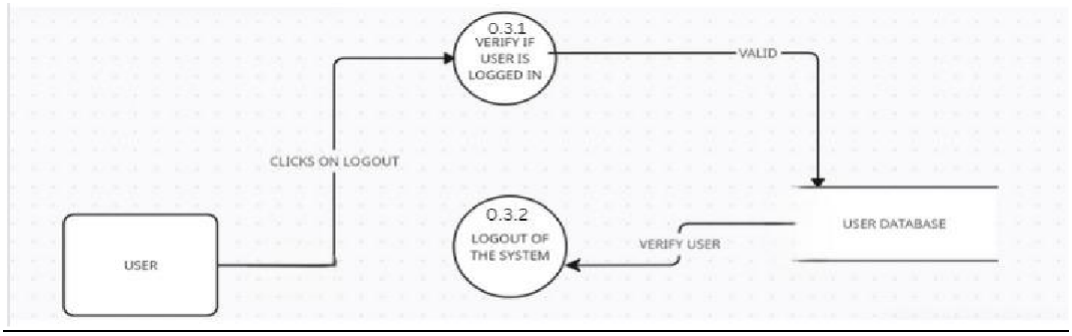



Fig 3.6 level 2 dfd DISPLAY YOUR PRODUCT

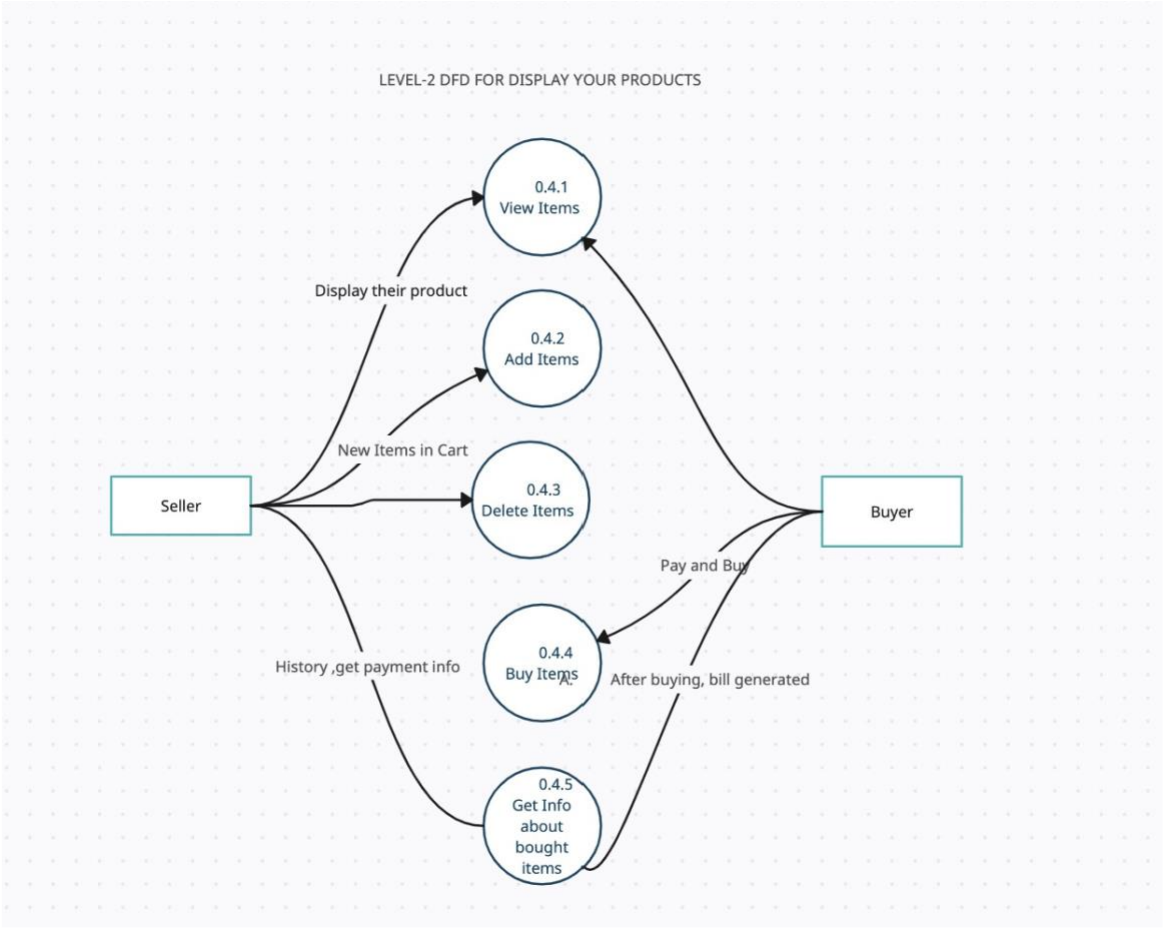


Fig 3.7 level 2 dfd PLACE YOUR ORDER

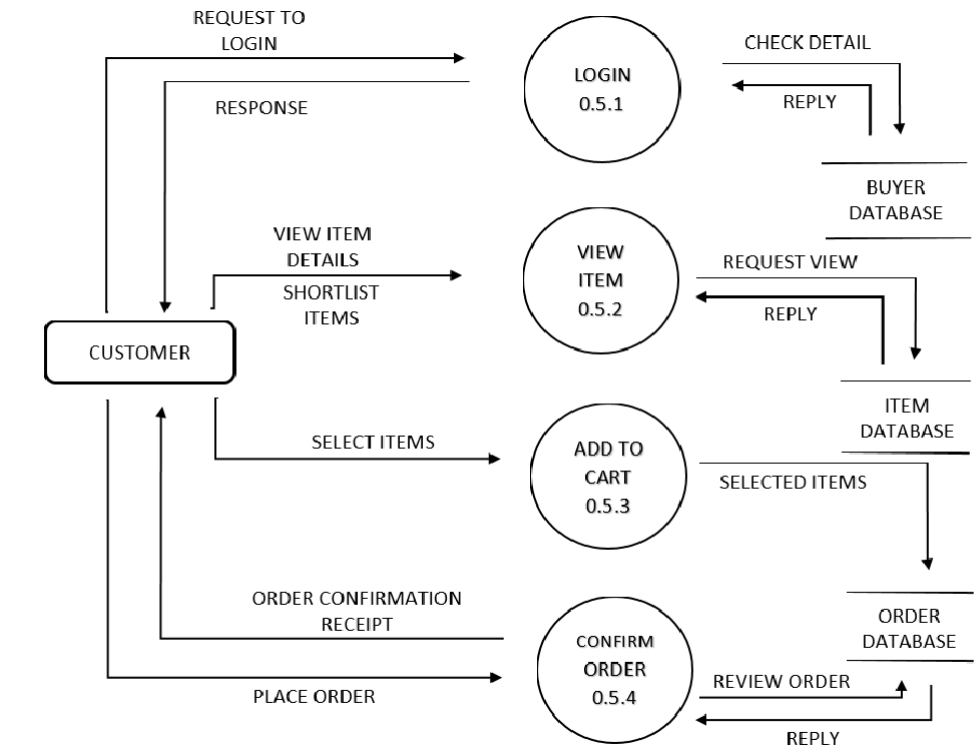



Fig 3.8 level 2 dfd UPDATE ITEMS

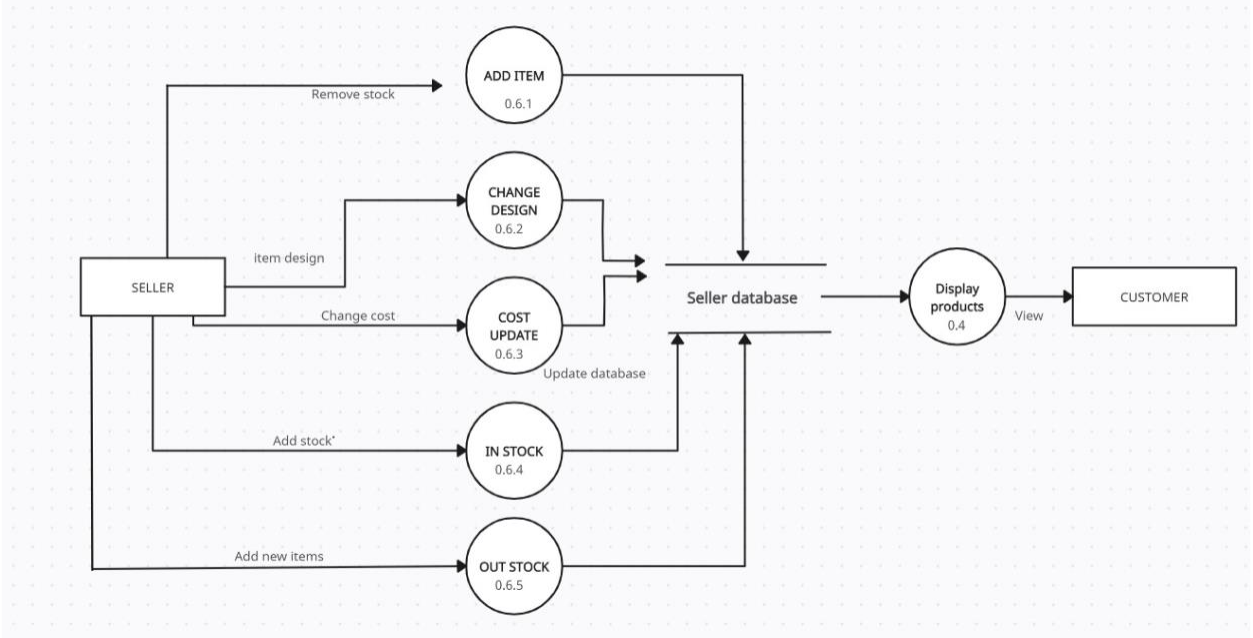
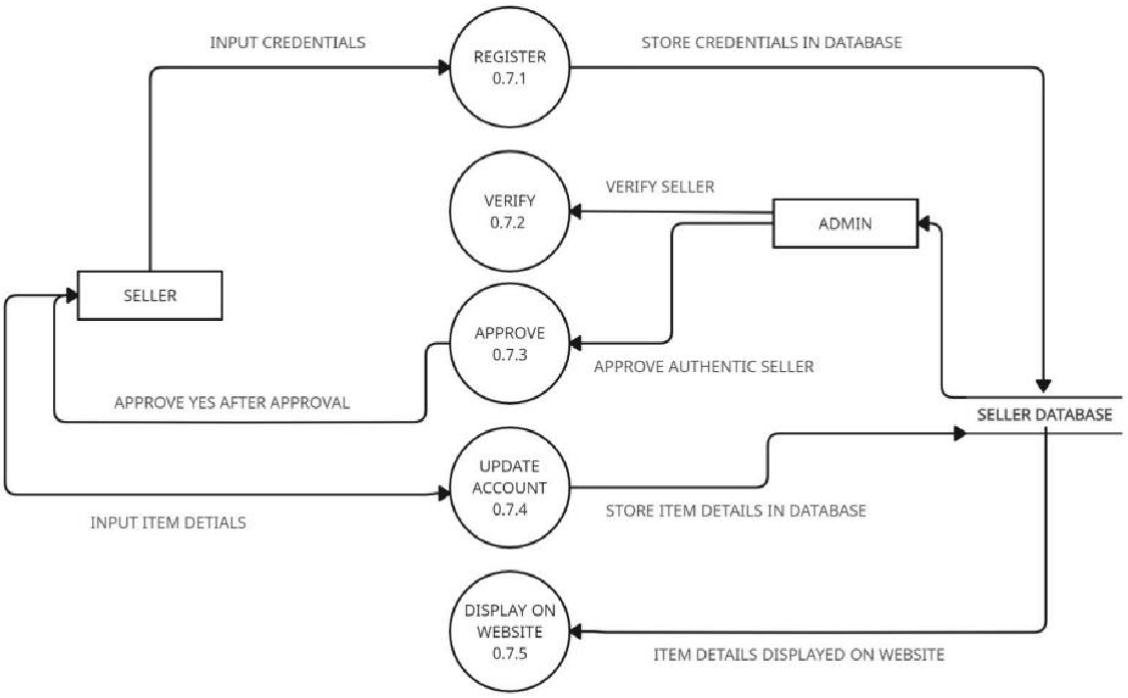


Fig 3.9 level 2 dfd JOIN AS  SELLER



# 3.2 DATA DICTIONARY

Legal character: [a-z|A-Z]  Digit: [0-9]  Special character: [@|$|#|+|-|.]

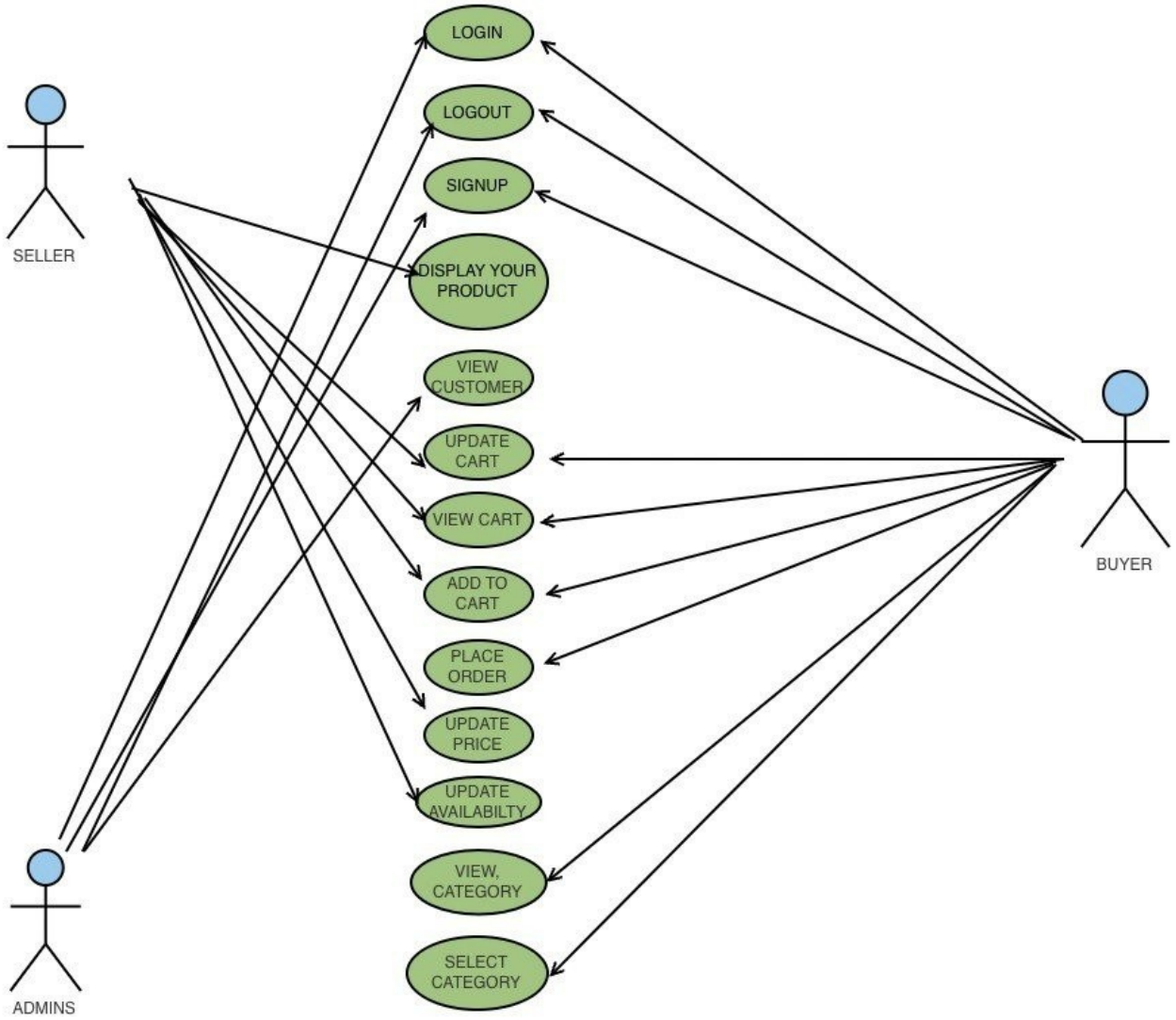
| | |
|---|---|
| **NAME** | **{Legal character} *** |
| **EMAIL** | **{Legal character + Digit + Special Character} *** |
| **PASSWORD** | **{Legal character + Digit + Special Character} *** |
| **ID** | **{Legal character + Digit }*** |
| **PHONE** | **{ Digit + Digit + Digit + Digit + Digit + Digit + Digit + Digit + Digit + Digit}** |
| **PRICE** | **{ Digit }*** |
| **STOCK** | **{ Digit }*** |
| **IMAGE** | **{ Digit }*** |
| **TIME** | **{ Digit + Special character }*** |
| **DATE** | **{ Digit + Special character }*** |
| **ADDRESS** | **{ Legal character + Digit + Special character }*** |

Table 3.1

# 3.3 USE CASES

## 3.3.1 USE CASE DIAGRAM

Fig 3.10



## 3.3.2 USE CASE DESCRIPTION

### LOGIN: -

| | |
|---|---|
| **Brief description** | This use case allows admin, sellers, buyers to login into the system<br>to get access to various the functionalities according to their roles. |
| **Actors** | Admin, sellers, buyers |
| **Basic flow** | User enters email id,<br>password, role System<br>verifies the credentials<br>User is directed to a landing page according to the role |
| **Alternative flow** | If login credentials do not match, user is prompted to attempt again<br>with the correct details |
| **Pre-condition** | User must have registered before login |
| **Post-condition** | User is directed to the screen according to their roles |

### LOGOUT: -

| | |
|---|---|
| **Brief description** | This use case allows admin, sellers, buyers to login into the system<br>and directs them to the landing page |
| **Actors** | Admin, sellers, buyers |
| **Basic flow** | Select log out button<br>Redirected to the<br>landing page Can log<br>in again (optional) |
| **Alternative flow** | - |
| **Pre-condition** | User must have logged in |
| **Post-condition** | User is directed to the landing page |

## DISPLAY YOUR PRODUCT: -

| | |
|---|---|
| **Brief description** | Allows the user to put up products for sale |
| **Actors** | Seller |
| **Basic flow** | Seller updates the products that needs to be added Product information is updated in the database<br>Changes are reflected to the users accordingly |
| **Alternative flow** | If the product was out of stock, database will be updated accordingly<br>If a new product id added, database will be updated accordingly |
| **Pre-condition** | If the product exists, appropriate message will be shown<br>Database will be updated accordingly |
| **Post-condition** | Changes are reflected to all the users |

## SIGN UP: -

| | |
|---|---|
| **Brief description** | Allows user to register with a unique email id and password |
| **Actors** | Sellers, buyers |
| **Basic flow** | User enters email id and creates a password Re-enters the password to confirm it<br>Selects sign up to get registered |
| **Alternative flow** | If the re-entered password does not match, user is prompted to enter the password again |
| **Pre-condition** | - |
| **Post-condition** | Directs the user to login into the system |

## UPDATE, VIEW, ADD CART: -

| | |
|---|---|
| **Brief description** | Allows user to add items to the card, remove, update and view it |
| **Actors** | Buyer |
| **Basic flow** | User selects items from the displayAdds items to the card<br>Can increase or decrease the quantity of the items |
| **Alternative flow** | Empty cart will have nothing to view |
| **Pre-condition** | User must have logged in |
| **Post-condition** | User will be directed to the 'place your order' page |

## VIEW CUSTOMERS: -

| | |
|---|---|
| **Brief description** | Allows the user to view the list of all the customers who have registered |
| **Actors** | Admin |
| **Basic flow** | View the complete list of all the customers<br>Selects a customer to view history of purchases from the database |
| **Alternative flow** | Cannot view details of a user who has deleted the account |
| **Pre-condition** | Buyer must be registered |
| **Post-condition** | Details of all registered buyers is displayed |

## PLACE ORDER: -

| | |
|---|---|
| **Brief description** | Allows users to place order for the chosen items |
| **Actors** | Buyer |
| **Basic flow** | Select items of choice to add<br>them in cartSelect quantity<br>Enter detail for delivery and payment method<br>Number of items from the database will be updated accordingly<br>after the checkout |
| **Alternative flow** | If the cart is empty, user is prompted to fill in the cart with articles |
| **Pre-condition** | Cart must not be empty |
| **Post-condition** | Redirects user to the payment page to confirm their order |

## VIEW, SELECT CATEGORIES: -

| | |
|---|---|
| **Brief description** | Allows user to filter articles according to their choice for<br>customization of product |
| **Actors** | Buyer |
| **Basic flow** | User selects categories to filter the product of choice<br>Filter can be made according to occasion, price, popularity, rating |
| **Alternative flow** | If no categories are selected, then randomly products are displayed<br>from the landing page |
| **Pre-condition** | Categories shall be selected from the available list that is from the<br>options that exists |
| **Post-condition** | Filtered products will be displayed according to the availability |

## AVAILABILITY OF PRODUCTS: -

| | |
|---|---|
| **Brief description** | User can update availability of the products |
| **Actors** | Seller |
| **Basic flow** | Select the<br>productView<br>availability<br>Update/re-stock/remove if needed<br>Database is updated accordingly |
| **Alternative flow** | - |
| **Pre-condition** | Product must exist in the database in order to update the<br>availability or view the status |
| **Post-condition** | If changes are made, it is reflected in the database and display<br>pages accordingly |

## UPDATE PRICE OF ITMES: -

| | |
|---|---|
| **Brief description** | User can update the price of existing products |
| **Actors** | Seller |
| **Basic flow** | Select the<br>productView<br>current price<br>Increase/decrease it<br>accordingly Database is<br>updated accordingly |
| **Alternative flow** | - |
| **Pre-condition** | Product must exist in the database in order to heighten/lower its<br>price |
| **Post-condition** | If changes are made, it is reflected in the database and display<br>pages accordingly |

# 3.4 SEQUENCE DIAGRAMS

Fig 3.11

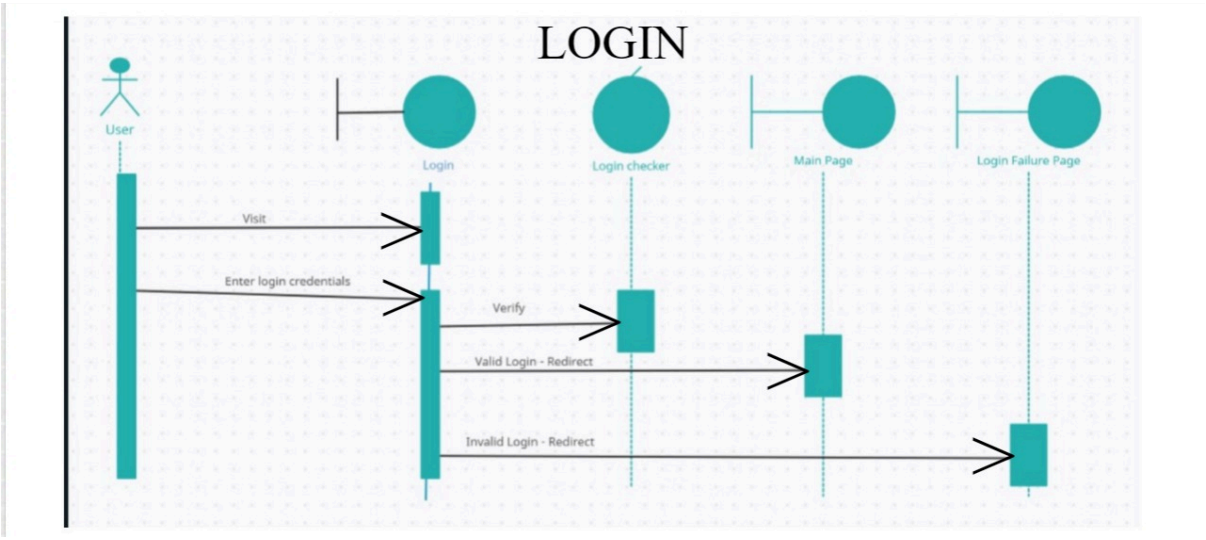



Fig 3.12

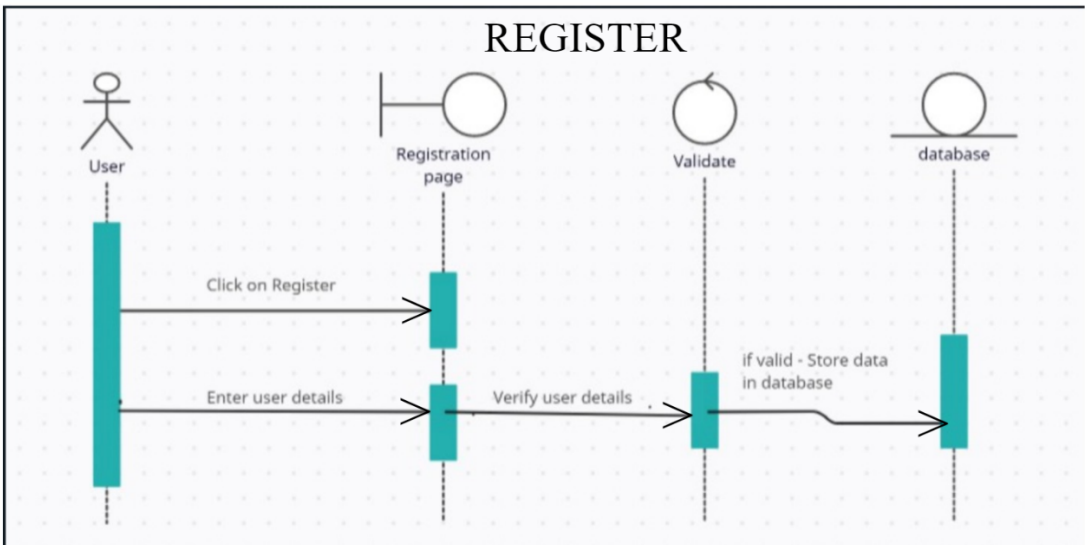


Fig 3.13

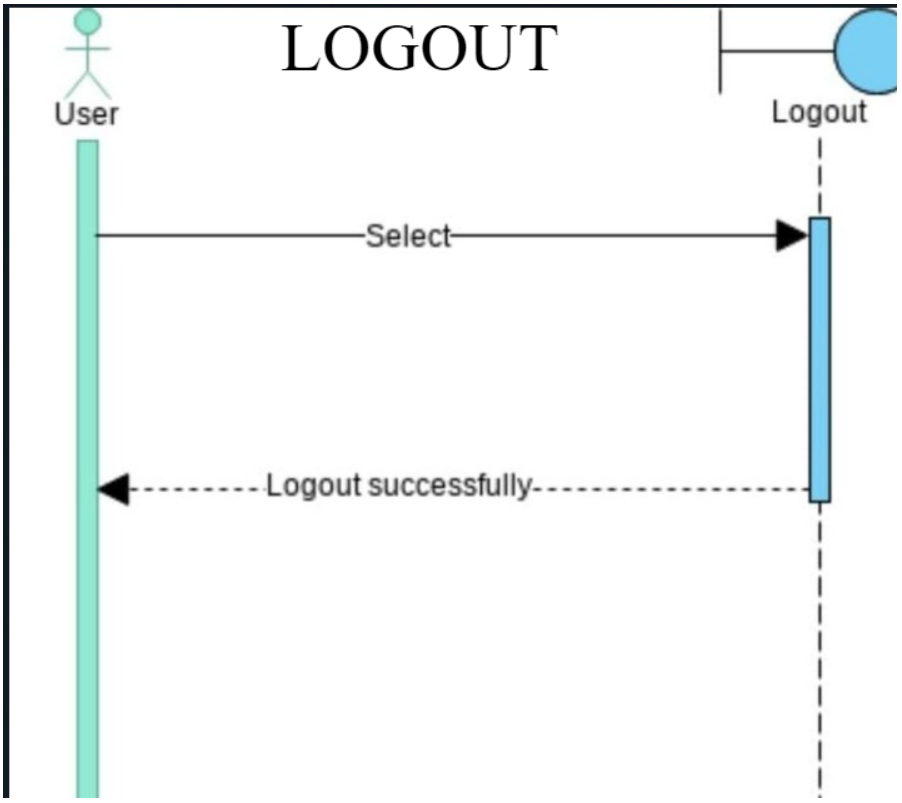



Fig 3.14

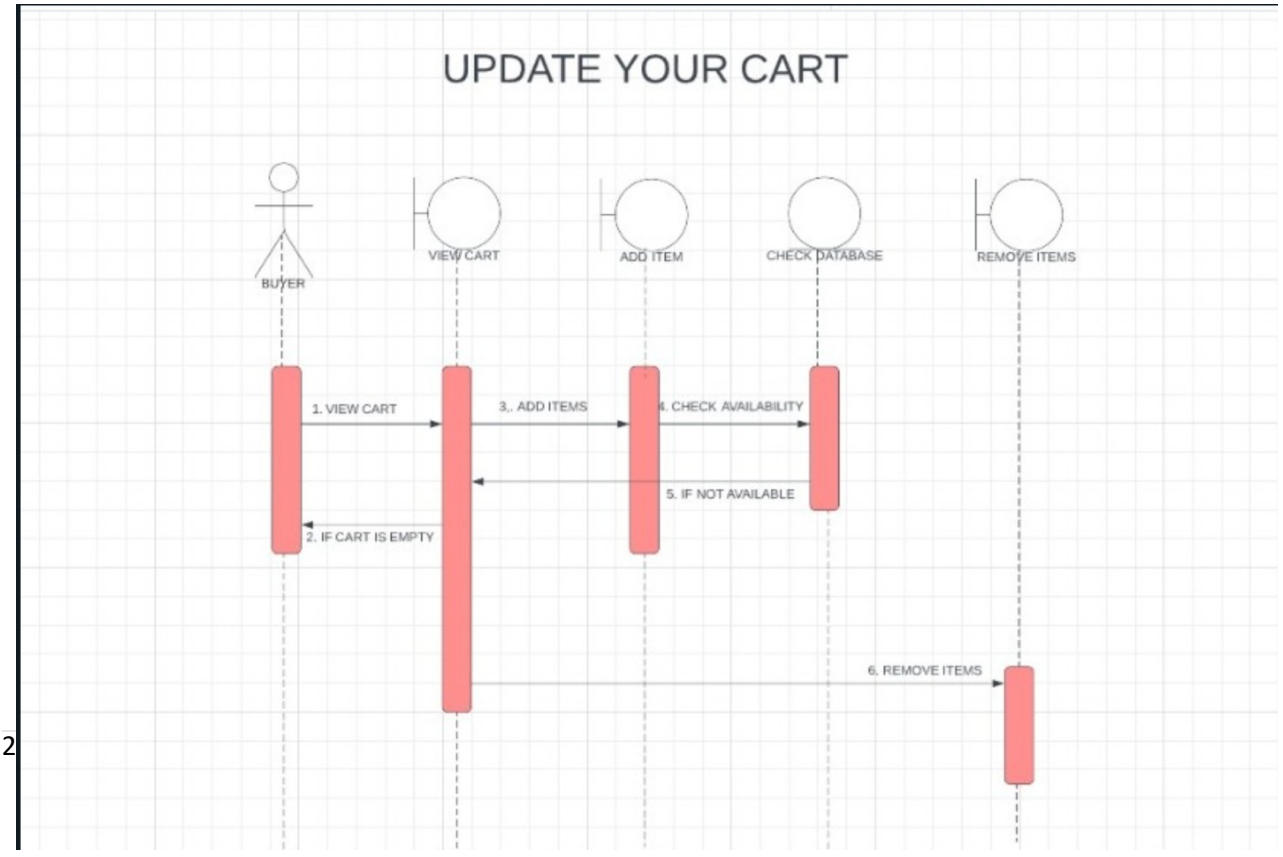


Fig 3.15

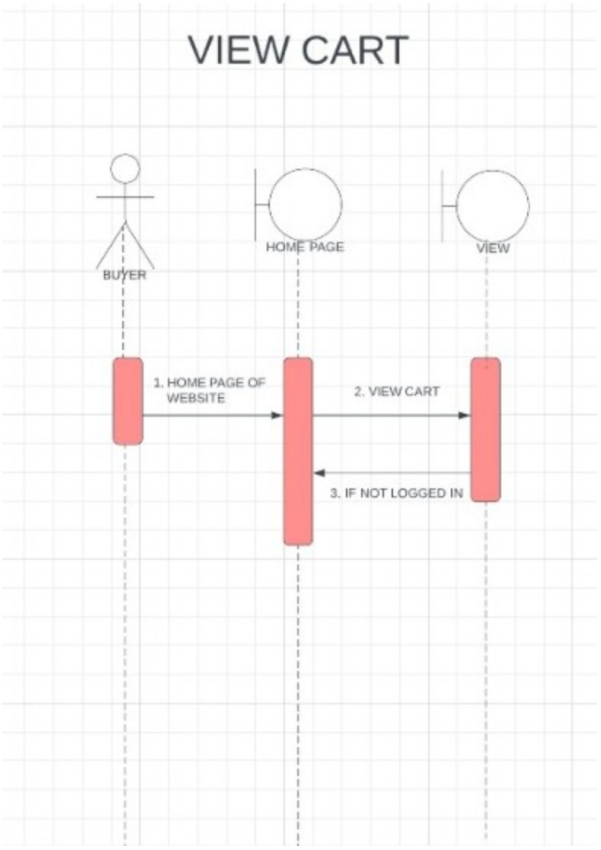



Fig 3.16

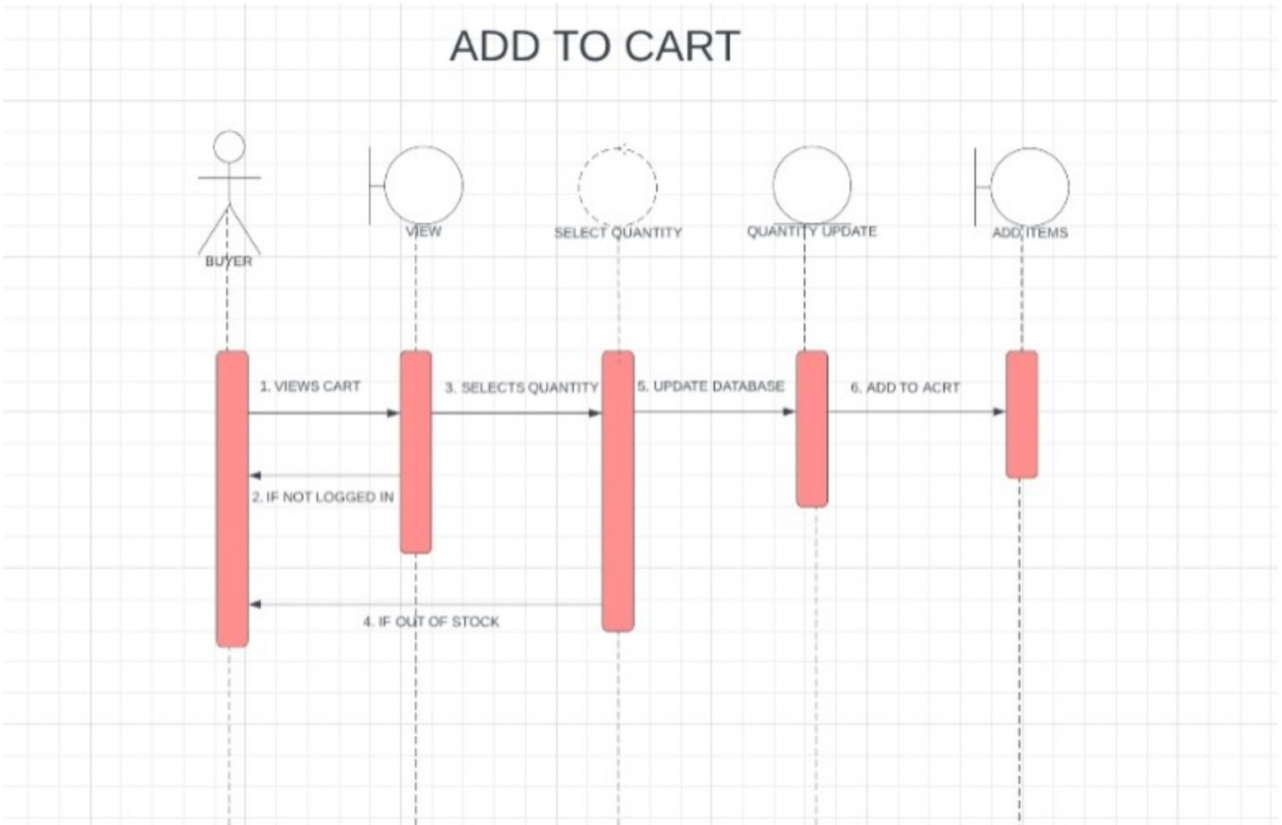


Fig 3.17

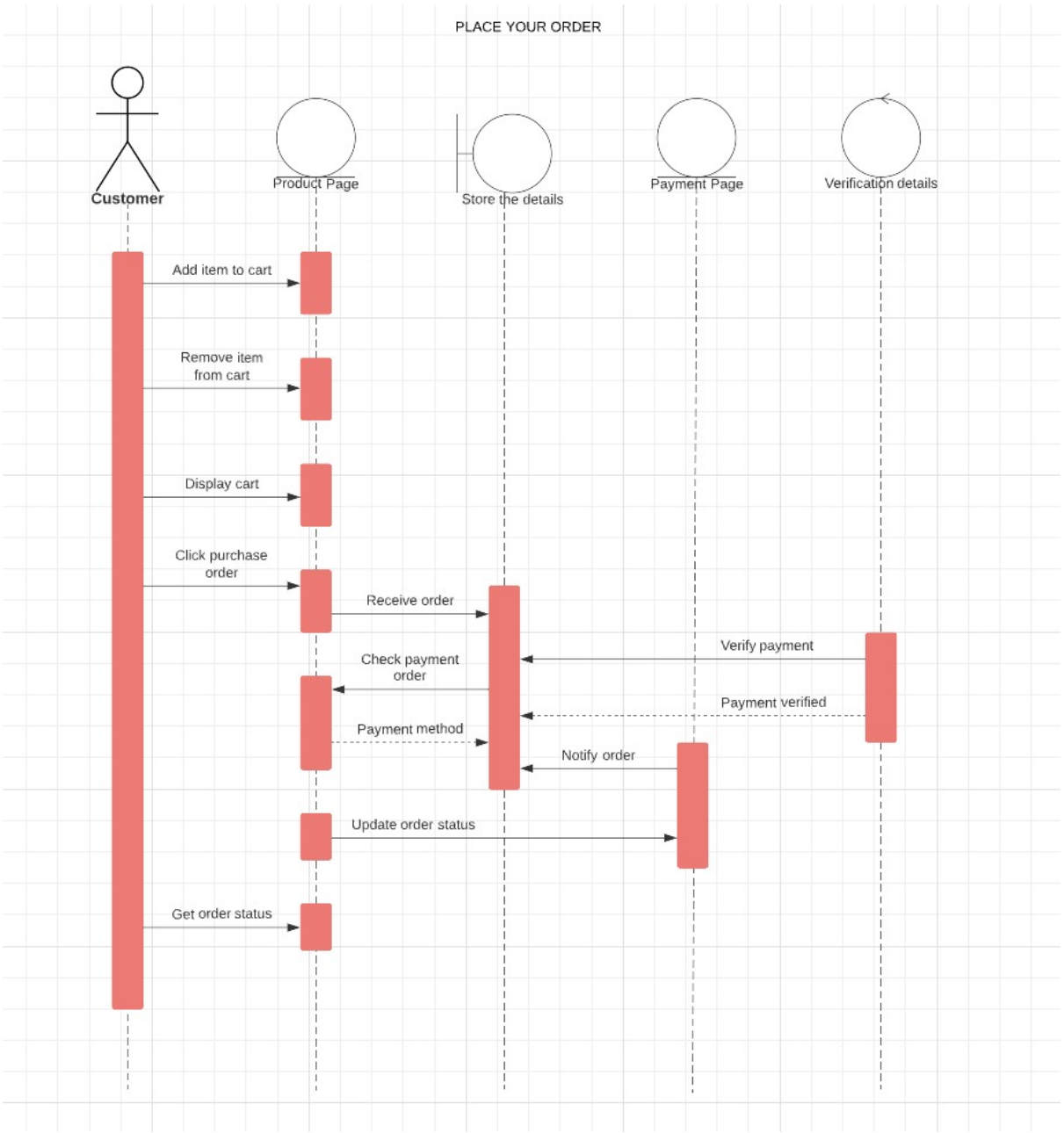

PLACE YOUR ORDER
Customer
Product Page
Store the details
Payment Page
Verification details
Add item to cart
Remove item from cart
Display cart
Click purchase order
Receive order
Verify payment
Check payment order
Payment verified
Payment method
Notify order
Update order status
Get order status

Fig 3.18

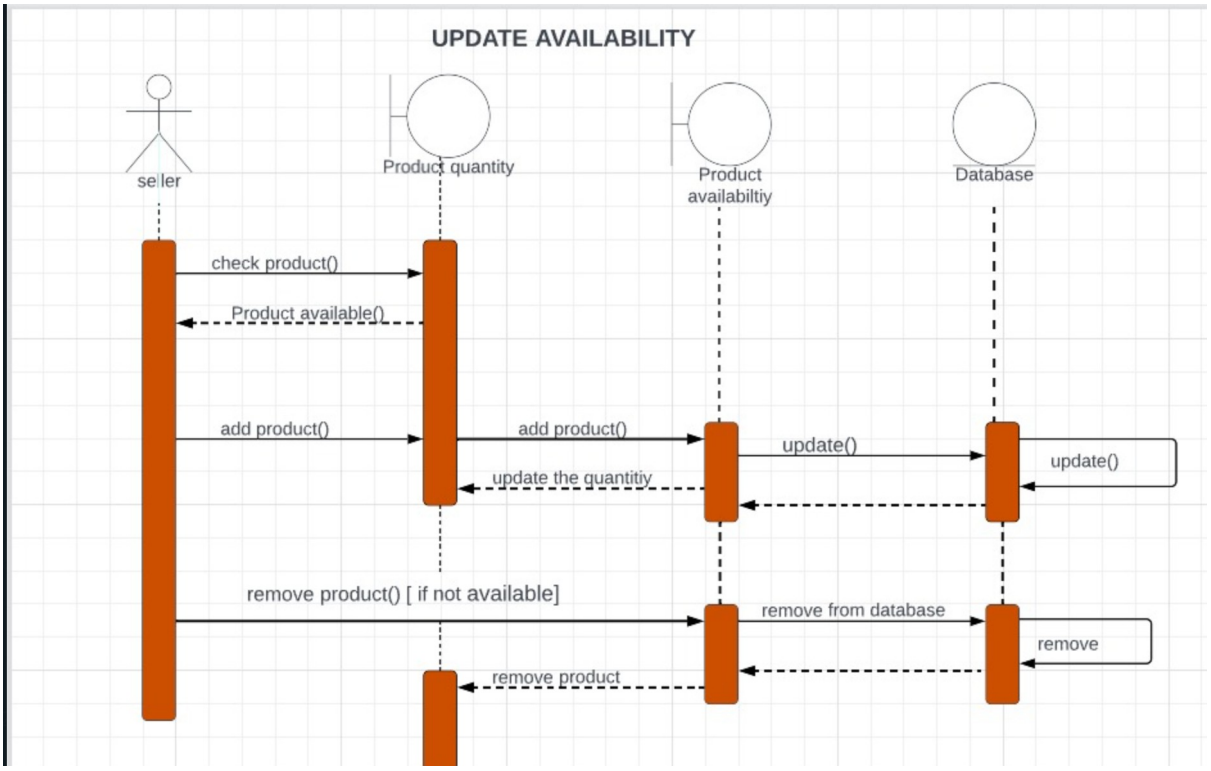



Fig 3.19

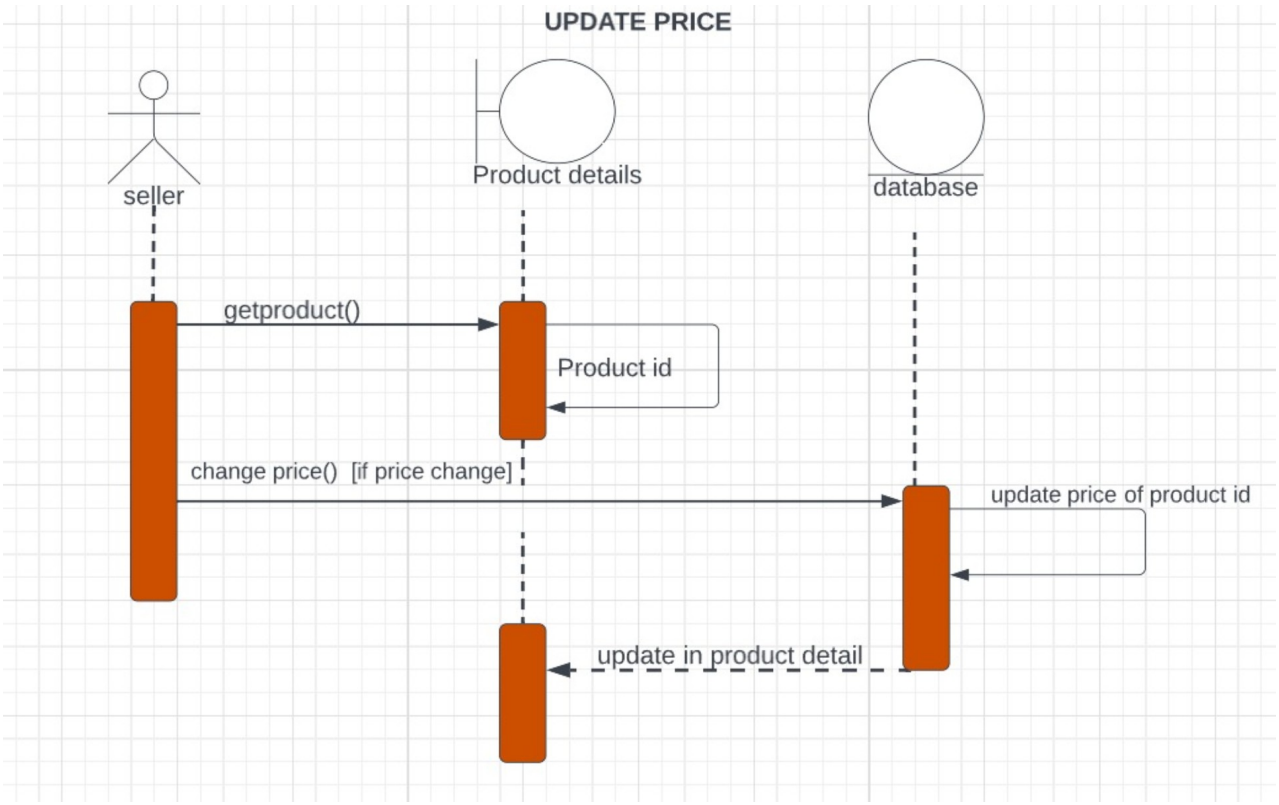


Fig 3.20

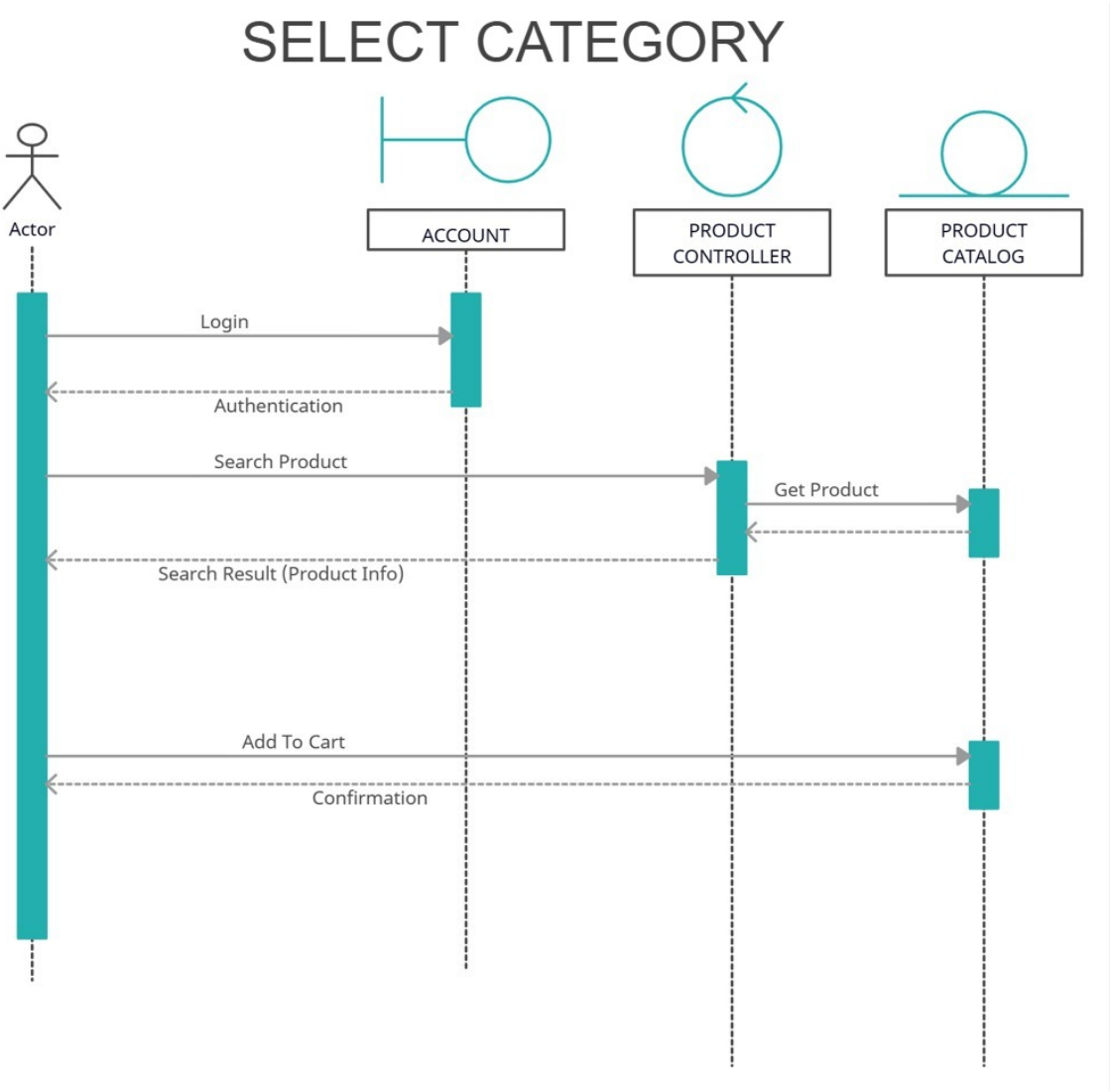
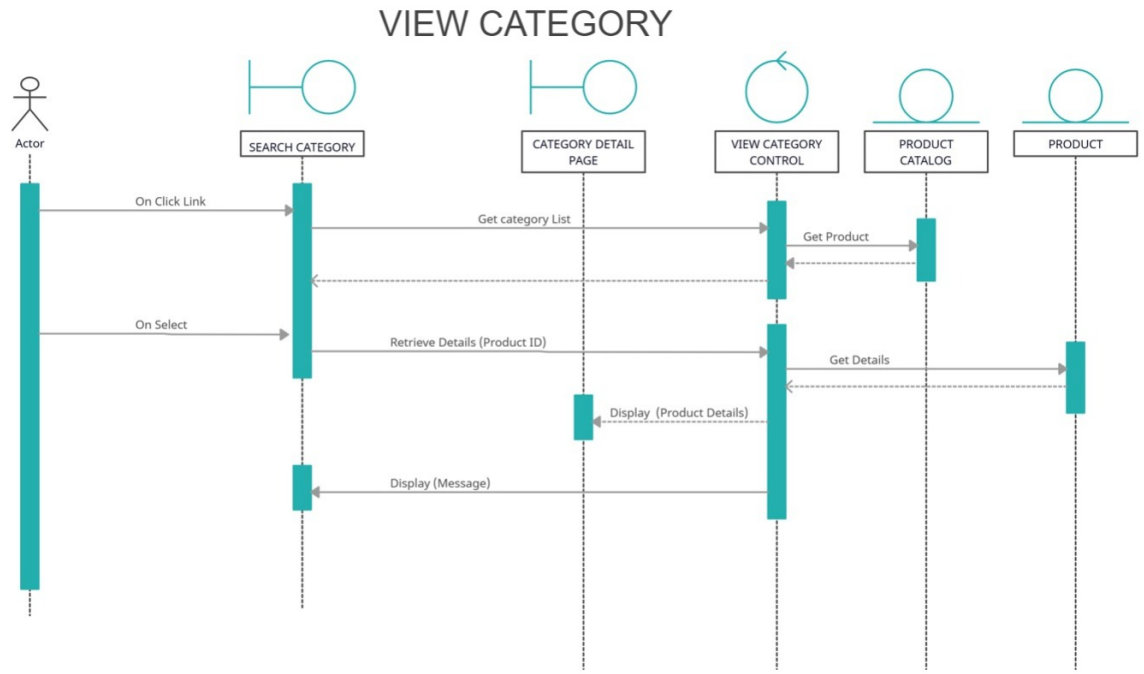

SELECT CATEGORY
Actor
ACCOUNT
PRODUCT CONTROLLER
PRODUCT CATALOG
Login
Authentication
Search Product
Get Product
Search Result (Product Info)
Add To Cart
Confirmation

Fig 3.21


VIEW CATEGORY
Actor
SEARCH CATEGORY
CATEGORY DETAIL PAGE
VIEW CATEGORY CONTROL
PRODUCT CATALOG
PRODUCT
On Click Link
Get category List
Get Product
On Select
Retrieve Details (Product ID)
Get Details
Display (Product Details)
Display (Message)

# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

## 4.1. INTRODUCTION

This section gives a scope description and overview of everything included in document. Also, the purpose of this document is described.

### 4.1.1 PURPOSE

The purpose of this document is to provide a detailed description of the requirements of "WRAP SHOP" software. The primary aim of the project is to develop a system that provides a platform where users can operate as a buyer or seller. the seller could sell their products to customer(buyer) and operate their offline shop on web. This document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

### 4.1.2 SCOPE

The "WRAP SHOP" software is an online gift shopping system which provides the users an ease in finding the best choices and gift ideas just by seating at home and surprise their loved ones. This application can be used by any small or big shops to boost up their market level. The goal of the web application for the online shopping system is to offer a comprehensive solution to both merchants and customers.

It will make it possible for Operators(seller) to put up online stores that customers can explore and buy from without having to physically visit the stores. Operators can manage products from the admin panel. Customers(buyer) residing in various parts of the world need to be able to purchase various product categories from the online shopping system. All products will be shown on the website in a classified manner and customers can also see the trending ideas and new suggestions for their required purchase. Customers can browse through any product to learn more about it, including its pricing, and place an order. Orders must include shipping and billing information. The order amount can be paid for by customers using a credit card, debit card, net banking, or cash (cash on delivery).

It will make easy for both customer(buyer) and operator(seller) to explore in the field of marketing and to explore and purchase/sell the best quality items. Not only it will benefit buyer but will also help generate more income for the operators(seller).

### 4.1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

- OTP- One Time Password
- HTML- Hyper-Text Mark-up Language
- HTTP – Hyper-Text Transfer Protocol
- TCP/IP – Transmission control Protocol/ Internet Protocol

### 4.1.4 REFERENCES

https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

### 4.1.5 OVERVIEW

The rest of the SRS document describes various system requirements, interfaces, features and functionalities in detail.

## 4.2. THE OVERALL DESCRIPTION

Customers can browse through the shops using the online shopping system program, moreover vendors(operators) can build up their own online stores over this system, and system administrators can accept or deny requests to add new stores and keep lists of store categories. User and sign up as a customer or as a seller after that the interface will be displayed according to the type of user, for seller add items, delete items, admin interfaces will be shown.Additionally, the developer is creating an online store to handle the merchandise in the stores and assist clients in making purchases without physically visiting the stores. Clients can use our website to sell their goods. The online shopping system will use the internet as the sole method for selling goods to its consumers.

### 4.2.1 PRODUCT PERSPECTIVE

Wrap shop aims towards bridging the gap between sellers and potential buyers who wish to save their time and energy by not going to the shop instead

ordering products with the ease of their homes.
Some products that are not available in a physical store; the proposed system is a solution to carry out buying/selling products online.

#### 4.2.1.1 USER INTERFACE

User interface may be implemented using any tool like java applets, MS Front Page, Figma, Adobe XD etc.

#### 4.2.1.2 HARDWARE INTERFACE

Since the website run over the internet, all the systems using this website may need to connect with internet and this will be a hardware interface, for example Modem, WAN, LAN, Ethernet, and Cross-Cable.

#### 4.2.1.3 SOFTWARE INTERFACE

The WEBSITE requires the support of scripting languages like PHP, HTML, CSS. Also to handle a large number of users and user's related databases the software requires the help of database systems like MySQL, XAMPP.

#### 4.2.1.4 COMMUNICATION INTERFACE

The system shall use the HTTP protocol for communication over the internet and TCP/IP protocol suite shall be used for intranet communication.

#### 4.2.1.6 MEMORY CONSTRAINTS

The minimum hardware requirement for the system is 128 MB of Ram and a 32-MB hard-disc drive.

#### 4.2.1.7 Reliability and Fault Tolerance

· All the users must be backed up.
·The database must be updated regularly and must be properly maintained.
·Users are authenticated.
· In case of Forgot password immediately a mail must be sent with a link to

reset his password.
·The access to all the databases is available only to admin.
· A log of all activities must be stored.

### 4.2.2 PRODUCT FUNCTION

The online shopping application would have the following basic functions:
· Display all the categories available for shopping on the system's main page.
· Display all the items linked to each category listed on the main page.
· Allow the administrator to add new items to the existing list of available
· Allow administrator to remove items.
· Allow the administrator to modify the price of each item. Allow the administrator to update the description about each item.
· Allow the administrator to view and edit information about each user that checkouts the items from the system.

### 4.2.3 USER CHARACTERISTICS

· The users should be able to perform the following functions using this system:
· View, browse, and select a category on the home page.
· View, add, and update items in the cart.
· Sign-in/login using a username and password. Place the order by completing the order form.

### 4.2.4 CONSTRAINTS

· User must be trained for basic computer functionality.
· User must have basic knowledge of English.
· Accessibility: Initially, the software should be available as a desktop application for a small set of users to test.

### 4.2.5 ASSUMPTION AND DEPENDENCIES

• To launch our wrap-shop, we will first need contacts with people who are up for selling their products online through our website.
• The data will be stored and dependent on a good functional database management system, since stock of the products need to be maintained.
• It is assumed that the user is friendly in using the internet as they might need

to browse along the website for good results.

- Since the application is a web based application there is a need for the internet browser. It will be assumed that the users will possess decent internet connectivity.
- It is assumed that seller who sells with us will also have a good knowledge about internet and they have basic decent internet connectivity.
- The seller should be updating regularly for their stock items, its amount because the website and regular customers might be dependent according to their needs.
- The administrator will also be dependent on the availability of the products since he/she will updating its availability on the website daily.
- It is assumed that the website interface should be user friendly and should be easy to use and understand.
- The seller should have proper hardware requirements as well as ground level items for registering and selling through us.

### 4.2.6 APPORTIONING OF REQUIREMENTS

- Our website may need various machine learning to inquire and suggest products according to the need of user.
- It will also be needing a an established stable connection between the administrators and multiple sellers as the future growth spurts.
- Socials will be linked gradually if the seller market growth is significant.
- Increments in the prices and algorithms will be coded if the product is bestseller and will be recommended to the new users.
- Usage of better front end coding languages such as C#, ReactJS will be used for better functionality.

## 4.3 SPECIFIC REQUIREMENTS

### 4.3.1 EXTERNAL INTERFACES

Various interfaces for the product could be

1). Login Page

2). Registration Form

3). There will be a screen displaying information about products.

4). If the customers select the buy button then another screen of shopping cart will be displayed.

5). Payment interface will be displayed to order the product.

6) After ordering for the product, the system will send one copy of the bill to the customer’s Email address

Software Interface:

It is compatible with Windows, Linux and Mac operating systems.
Our software is web based so it needs a web browser and an internet connection.

Hardware Interface:

Hardware requirements for insurance on internet will be same for both parties which are as follows: Processor: Dual Core
RAM: 2 GB
Hard Disk: 320 GB
NIC: For each party

Communication Interfaces:

The two parties should be connected by LAN or WAN for the communication purpose.

### 4.3.2 FUNCTIONAL REQUIREMENTS

The various functional requirements of the system are as follows:

#### 3.2.1 Registration

If customer wants to buy the product then he/she must be registered, unregistered user can't buy any product unless registered.

#### 3.2.2 Login

Customers can only login to the system by entering valid user id and password.

#### 3.2.3 Updating the Cart

Customers after login or registration can place or cancel order of the product from the cart.

### 3.2.4 Payment

In this system we are dealing the mode of payment by Cash, Credit, Debit, UPI etc.

### 3.2.5 Logout

Costumer must be logged in to buy or surf the website.

### 3.2.6 Report Generation

After ordering for the product, the system will send one copy of the bill to the customer's Email-address and another one for the system data base.

### 4.3.3 PERFROMANCE REQUIREMENTS

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time.
Also the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

### 4.3.4 LOGICAL DATABASE REQUIREMENTS

In this software, the main purpose of a database is to store information for retrieving the product details, customer information, track transactions, and further, maintain the cart. One of the biggest benefits of using a database for WRAP SHOP is structuring vast amounts of shop data.

### 4.3.5 DESIGN CONSTRAINTS

• Software languages
The languages that shall be used for coding the Wrap Shop are
HTML, CSS, JAVASCRIPT, PHP

• In our Database design, we used MYSQL.

#### 4.3.5.1 STANDARD COMPLIANCE

Report format: All the reports produced for this project are in compliance with the standard templates in accordance with the standard guidelines and policy. Naming Conventions: All the documents are named using the standard naming conventions.

### 4.3.6 SOFTWARE SYSTEM ATTRIBUTES (NON-FUNCTIONAL)

System attributes requirements such as those for performance, security, modifiability, reliability and usability have a significant influence on the software architecture of a system. There are a number of quality attributes of software that can serve as requirements.

#### 4.3.6.1 RELIABILTY

The web will be able to handle more than one orders placed at a time. After confirmation of order, database will be updated immediately and costumers won't face any issues.

#### 4.3.6.2 AVAILABILTY

The web will be available all around the world. Costumer can access the website 24x7.

#### 4.3.6.3 SECURITY

The website is password protected and also any update of new product entries and order processing is done by only privileged users.

#### 4.3.6.4 MAINTANABILITY

The website is to be designed so that it is easily maintained. Also it should allow incorporating new requirements in any module of system.

#### 4.3.6.5 PORTABILITY

The website will be easily portable on any system.

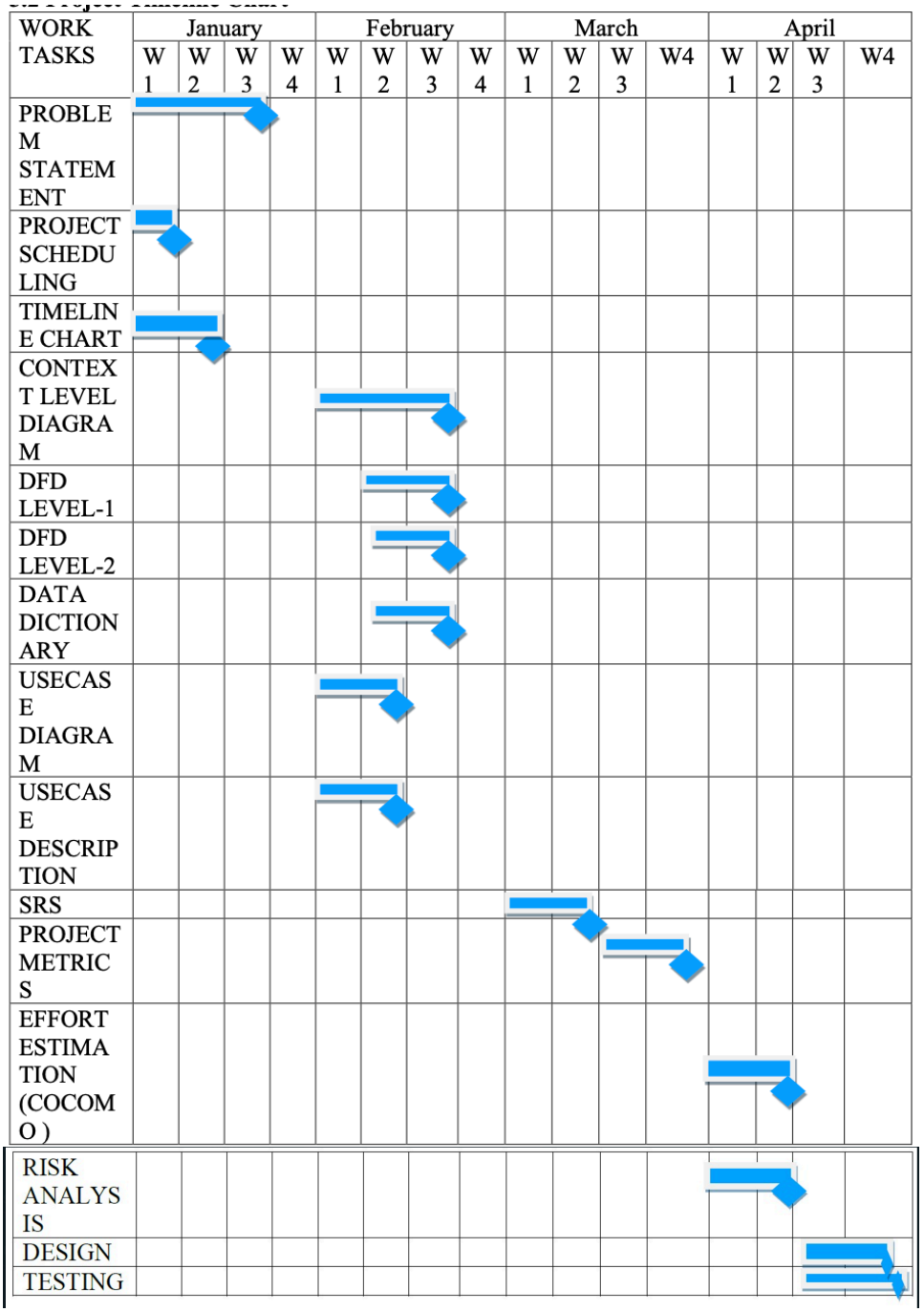
# 5 Project Planning

## 5.1 Project Scheduling

| WORK TASKS | PLANNED START | ACTUAL START | PLANNED COMPLETE | ACTUAL COMPLETE | ASSIGNED PERSON(s) | EFFORT ALLOCATED |
|---|---|---|---|---|---|---|
| PROBLEM STATEMENT | January, w1 | January, w1 | January, w2 | January w3 | Gyana Karn | 1 person per week |
| PROJECT SCHEDULING | January, w1 | January, w1 | January, w1 | January, w1 | Nidhi Gupta | 1 person per week |
| TIMELINE CHART | January, w2 | January, w2 | January, w2 | January, w2 | Nidhi Gupta | 1 person per week |
| CONTEXT LEVEL DIAGRAM | February, w1 | Februay, w2 | February, w2 | February, w3 | Shubhangi | 1 person per week |
| DFD LEVEL-1 | February, w2 | Februay, w2 | February, w2 | February, w3 | Gyana, Mansi, Nidhi, Shubhangi, Diya | 1 person per day |
| DFD LEVEL-2 | February, w2 | Februay, w2 | February, w2 | February, w3 | Gyana, Mansi, Nidhi, Shubhangi, Diya | 1 person per day |
| DATA DICTIONARY | February, w2 | Februay, w2 | February, w3 | February, w3 | Gyana, Mansi, Nidhi, Shubhangi, Diya | 1 person per day |

| | | | | | | |
|---|---|---|---|---|---|---|
| USECASE DIAGRAM | February, w1 | Februar y, w1 | February, w2 | February, w2 | Gyana Karn | 1 person per week |
| USECASE DESCRIPT ION | February, w1 | Februar y, w1 | February, w2 | February, w2 | Shubhan gi | 1 person per week |
| SRS | March, w1 | March, w1 | March, w2 | March, w2 | Gyana, Mansi, Nidhi, Shubhan gi, Diya | 1 person per day |
| PROJECT METRICS | March, w3 | March, w3 | March, w3 | March, w4 | Mansi | 1 person per week |
| EFFORT ESTIMATI ON (COCOMO ) | April, w1 | April, w1 | April, w2 | April, w2 | Gyana | 1 person per week |
| RISK ANALYSIS | April, w1 | April, w1 | April, w2 | April, w2 | Gyana | 1 person per week |
| DESIGN | April, w3 | April, w3 | April, w4 | April, w4 | Mansi | 1 person per week |
| TESTING | April, w3 | April, w3 | April, w4 | April, w4 | Mansi | 1 person per week |

## 5.2 Project Timeline Chart




## 5.3 EFFORT ESTIMATION & FP- BASED COMPUTING

Function Point Metric is an example of Product metrics for Analysis Model.

- It is used as a means for measuring the functionality delivered by a system and also examines requirement/ analysis model for predicting size of resultant system.
- Using historical data, Function Point metric can be used to:- i. Estimate the effort or cost required to design, code or test the software. ii. To predict number

of errors that will be encountered during testing. iii. Forecast number of components or number of projected source links in implemented system. Function points are derived using empirical relationship based on countable (direct) measures of software's information domain and quantitative assessment of software complexity.

● Software Information Domain Values consists of number of:-

i. External Inputs (EI)
ii. External Outputs (EO)
iii. External Inquiries (EI)
iv. Internal Logical Files (ILF)
v. External Interface Files (EIF)

To compute function points (FP), the following relationship is used:
FP= Count total × [0.65 + 0.05 × ∑ (Fi)],
where Count total= Sum of all Function Point entries Calculation of Value Adjustment Factors (VAF) is based on the responses of the following questions:

| | | |
|---|---|---|
| 1 | Does the system require reliable backup and recovery? | 3 |
| 2 | Are specialized data communications required to transfer the information to and from the application? | 3 |
| 3 | Are there distributed processing functions? | 3 |
| 4 | Is performance critical? | 3 |
| 5 | Will the system work in an existing heavily utilized operational environment? | 3 |
| 6 | Does the system require online data entry? | |
| 7 | Does the online data entry require input transaction to be built over multiple screens or operations? | 3 |
| 8 | Are Internal Logical Files updated online? | 3 |
| 9 | Are input-output queries or files complex? | 3 |
| 10 | Is the internal processing complex? | 3 |
| 11 | Is the code designed to be reusable? | |
| 12 | Are conversion and installation included in design? | 3 |
| 13 | Is the system designed for multiple installations in multiple organizations? | 3 |
| 14 | Is the application designed to facilitate changes and ease of use by the user? | 3 |
| **COUNT TOTAL (Σ $F_i$)** | | **42** |

**Table 5.3: Value Adjustment Factors (VAF)**

The count total is the sum of all FP entries obtained from the following tale:

| PROTOTYPES | External input | External output: | External inquiries | Internal logical files | External interface files |
|---|---|---|---|---|---|
| HOME PAGE | 1 | 0 | 0 | 1 | 0 |
| SIGN UP | 3 | 1 | 0 | 1 | 0 |
| LOG IN | 2 | 1 | 0 | 1 | 0 |
| DISPLAY PRODUCT | 1 | 1 | 0 | 1 | 0 |
| SELL WITH US | 3 | 0 | 0 | 1 | 0 |
| SHOP BY THEME | 1 | 0 | 0 | 1 | 0 |
| CART | 0 | 0 | 0 | 1 | 0 |
| PRODUCT DETAILS | 1 | 0 | 1 | 1 | 0 |
| DELIVERYM DETAILS | 5 | 1 | 0 | 1 | 0 |
| PAYMENT | 3 | 1 | 1 | 1 | 1 |
| THANKYOU PAGE | 0 | 0 | 0 | 0 | 0 |

Calculate Functional point of system
Scale -3

| INFORMATION DOMAIN VALUES | COST | SIMPLE | AVERAGE | COMPLEX | COUNT*WEIGHING FACTOR(SIMPLE) |
|---|---|---|---|---|---|
| External input | 20 | 3 | 4 | 6 | 60 |

| | | | | | |
|---|---|---|---|---|---|
| External output: | 5 | 4 | 5 | 7 | 20 |
| External inquiries | 2 | 3 | 4 | 6 | 6 |
| Internal logical files | 9 | 7 | 10 | 15 | 63 |
| External interface files | 1 | 5 | 7 | 10 | 5 |
| Total count | | | | | 154 |

**Formula**

Function point = total count*(0.65 + (0.01* sum(fi))

= 154*(0.65 +(0.01* 14*3)

= 154*0.42 = 64.68PM

## 5.4 COST ESTIMATION: COCOMO II MODEL

Barry Boehm gave a hierarchy of software estimation models called COCOMO i.e constructive cost model. The original COCOMO model was widely used in the industry and was later evolved into a comprehensive model. Estimation model is called the COCOMO II model. COCOMO II is a hierarchy of estimation models that consists of :
Application composition model - It is used during the prototyping of user
• interfaces, assessment of process during system and software interaction
• and evaluation of technology maturity
. • Early design stage model - It is used once. The requirement has been
• stabilized and basic software architecture is established.
• Post-architectural stage needed - This model is used during the
• construction of software.

## APPLICATION OF COMPOSITION MODEL

These model use sizing information for which 3 options are available which are object points, function points and lines of source code. Object-point is an indirect software measure computed using counts of number of screens on the user interface, number of reports generated and number of reusable components and 3 GL Components required to build the application. Each object instance is classified into one of the three complexity levels: Simple, Medium or Difficult.

Complexity is a function of number and source of client and server data tables required to generate screen or report and number of views or sections presented as part of the screen or report.

**In our project:**

No of screens = 11
No of reports = 3
No of 3GL components used = 0

There are simple screens and reports in our project

Thus the object point is = 11*1+3*2 = 11+6 = 17 E

The percentage of reuse is 0 as we are not reusing any component.

NOP=17*[(100-0)/100]=17
PROD = NOP/Person-month = 4

Estimated Effort = NOP/PROD = 17/4 = 4.25 person-month( 4 person-month)

## 5.5 RISK ANALYSIS

SOFTWARE RISKS

It involves 2 aspects: uncertainty and loss. Uncertainty means risk may or may not happen. If risk becomes reality, then unwanted consequences or loss will occur.

- PHASES INVOLVED IN RISK ANALYSIS AND MANAGEMENT

1. Risk Identification
2. Risk Analysis
3. Risk ranking and assessment
4. Creating risk plan or RMMM plan

- TYPES OF RISKS

According to general categorisation there are 3 types of risks:
1. Known Risk
2. Predictable Risk
3. Unpredictable Risk Another category of risk type:
1. Project Risk

2. Technical Risk
3. Business Risk Project Risk: Identify potential problems that might occur in budget, schedule and staffing. It also includes project complexity, project size and degree of structural uncertainty. Technical Risk: Identify potential design problem, implementation problem, interface problems, verification problem and maintenance problem. They threaten the quality of the software produced. Business Risk: Threatens the viability of the software to be built and often jeopardise the project or the product. There are 5 types of business risks:
a) Market risk 42
b) Strategic risk
c) Sales risk
d) Management risk
e) Budget risk

## ASSESSING OVERALL PROJECT RISK

1. Have top software and customer mangers formally committed to support the project? **NO**
2. Are end users enthusiastically committed to the project and the system / product to be built? **YES**
3. Are requirements fully understood by the software engineering team and its customers? **YES**
4. Have customers been involved fully in the definition of requirements? **YES**
5. Do end users have realistic expectations? **YES**
6. Is the project scope stable? **YES**
7. Does the software engineering team have the right mix of skills? **YES**
8. Are project requirements stable? **YES**
9. Does the project team have experience with the technology to be implemented?**YES**
10. Is the number of people on the project team adequate to do the job? **YES**
11.Do all customer / user constituencies agree on the importance of the project and on the requirements for the system / product to be built? **YES**

### 5.4 Risk Management Table

| S.NO | RISKS | CATEGORY | PROBABILITY | IMPACT | RMMM PLAN |
|---|---|---|---|---|---|
| 1 | Customer will change the requirements | Project Size | 80% | Critical | Conduct multiple reviews so that the requests are well understood |

| S.NO | RISKS | CATEGORY | PROBABILITY | IMPACT | RMMM PLAN |
|---|---|---|---|---|---|
| 2 | Delivery deadline will be tightened | Business | 50% | Critical | Continuously trace the timeline chart. Hire Experts to meet the deadline e. |
| 3 | Lack of experience in fixing bugs in code | Development | 30% | Critical | The whole website might crash and experts would be needed. |
| 4 | Stock Might not refill for later purchases | Business | 50% | Critical | Some customers might purchase products altogether that might be available on the time of ordering but might run out while shipping. |
| 5 | Size Estimation may be significantly low | Business | 10% | Mild | Storage issue, might run out of space |

| S.NO | RISKS | CATEGORY | PROBABILITY | IMPACT | RMMM PLAN |
|---|---|---|---|---|---|
| 6 | Funding will be lost | Business | 40% | Critical | Might result in loss of assets, can cause huge losses |
| 7 | End user may resist the system | Business | 20% | Mild | The UI might not be friendly enough to be interactive with the user |
| 8 | Less Reuse than planned | Personal | 25% | Mild | The options added that were supposed to be portable and accessible to all might not be used at all |
| 9 | Large No of user than planned | Personal | 20% | Mild | Stock might be less for the customers demanding |

| S.NO | RISKS | CATEGORY | PROBABILITY | IMPACT | RMMM PLAN |
|---|---|---|---|---|---|
| 10 | Self in-experienced | Business | 50% | Critical | Business Knowledge is not properly available to the admin to progress in stocks |
| 11 | Technology will not requirements | Personal | 40% | Critical | Usage of higher technology languages might be needed in future. |

# 6. DESIGN

## SELLER REGISTER TABLE

| FIELD NAME | TYPE | SPECIFICATIONS | CONSTRAINT | UNIQUE | DESCRIPTION |
|---|---|---|---|---|---|
| NAME | VARCHAR | 30 CHARACTERS | NOT NULL | NO | NAME OF SELLER |
| EMAIL | VARCHAR | 20 CHARACTERS | NOT NULL, PRIMARY | YES | VALID EMAIL ID OF SELLER |
| PASSWORD | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | NO | PASSWORD SET BY SELLER |
| UID | VARCHAR | 16 VARCHAR CHARACTERS | NOT NULL | YES | UNIQUE ID GENERATED BY THE SYSTEM |
| PHONE | BIGINT | 10 DIGITS | NOT NULL | NO | CONTACT NUMBER OF THE SELLER |

**TABLE 6.1 SELLER REGISTER TABLE**

## BUYER REGISTER TABLE

| FIELD NAME | TYPE | SPECIFICATIONS | CONSTRAINT | UNIQUE | DESCRIPTION |
|---|---|---|---|---|---|
| NAME | VARCHAR | 30 CHARACTERS | NOT NULL | NO | NAME OF BUYER |
| EMAIL | VARCHAR | 20 CHARACTERS | NOT NULL, PRIMARY | YES | VALID EMAIL ID OF BUYER |
| PASSWORD | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | NO | PASSWORD SET BY BUYER |
| UID | VARCHAR | 16 VARCHAR CHARACTERS | NOT NULL | YES | UNIQUE ID GENERATED BY THE SYSTEM |
| PHONE | BIGINT | 10 DIGITS | NOT NULL | NO | CONTACT NUMBER OF THE BUYER |

**TABLE 6.2 BUYER REGISTER TABLE**

## ADMIN LOGIN TABLE

| FIELD NAME | TYPE | SPECIFICATIONS | CONSTRAINT | UNIQUE | DESCRIPTION |
|---|---|---|---|---|---|
| NAME | VARCHAR | 30 CHARACTERS | NOT NULL | NO | NAME OF SELLER |
| EMAIL | VARCHAR | 20 CHARACTERS | NOT NULL, PRIMARY | YES | VALID EMAIL ID OF SELLER |
| PASSWORD | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | NO | PASSWORD SET BY SELLER |
| UID | VARCHAR | 16 VARCHAR CHARACTERS | NOT NULL | YES | UNIQUE ID GENERATED BY THE SYSTEM |
| PHN | BIGINT | 10 DIGITS | NOT NULL | NO | CONTACT NUMBER OF THE SELLER |

**TABLE 6.3 ADMIN LOGIN TABLE**

## PRODUCT CATEGORIES TABLE

| FIELD NAME | TYPE | SPECIFICATIONS | CONSTRAINT | UNIQUE | DESCRIPTION |
|---|---|---|---|---|---|
| NAME | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL, PRIMARY | YES | NAME OF THE PRODUCT |
| PRICE | DECIMAL | 8 DIGIT | NOT NULL | NO | PRICE OF THE PRODUCT |
| SELLER ID | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | NO | ID OF THE SELLER WHO SELLS THE PRODUCT |
| STOCK | INTEGER | 8 VARCHAR CHARACTERS | NOT NULL | NO | THE AMOUNT OF PRODUCTS AVAILABLE |
| PRODUCT ID | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL, PRIMARY | YES | AN UNIQUE PRODUCT ID DESIGNATED TO EACH PRODUCT |
| IMAGE | VAR BINARY (MAX) | 15 BINARY NUMERIC CHARACTERS | NOT NULL | YES | PICTURE OF THE PRODUCT, HOW IT LOOKS LIKE |
| DESCRIPTION | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | YES | WHAT THE PRODUCT IS ABOUT, HOW IS IT USEFUL |

**TABLE 6.4 PRODUCT CATEGORIES TABLE**

## ORDER DETAILS TABLE

| FIELD NAME | TYPE | SPECIFICATIONS | CONSTRAINT | UNIQUE | DESCRIPTION |
|---|---|---|---|---|---|
| ORDER ID | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | YES | ORDER ID OF THE CUSTOMER |
| BUYER ID | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL, PRIMARY | YES | BUYER ID OF THE CUSTOMER |
| TOTAL AMOUNT | DECIMAL | 8 DIGIT | NOT NULL | NO | TOTAL AMOUNT OF THE ORDER PLACED |
| TIME OF ORDER | TIME | HOURS-MIN-SEC | NOT NULL | NO | THIS PROVIDES THE TIME WHEN THE ORDER WAS PLACED |
| DATE OF ORDER | DATE | YY-MM-DD | NOT NULL | NO | THIS PROVIDES THE DATE WHEN THE ORDER WAS PLACED |

**TABLE 6.5 ORDER DETAILS TABLE**

## DELIVERY TABLE

| FIELD NAME | TYPE | SPECIFICATIONS | CONSTRAINT | UNIQUE | DESCRIPTION |
|---|---|---|---|---|---|
| PRODUCT ID | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL | YES | AN UNIQUE PRODUCT ID DESIGNATED TO EACH PRODUCT |
| BUYER ID | VARCHAR | 8 VARCHAR CHARACTERS | NOT NULL, PRIMARY | YES | BUYER ID OF THE CUSTOMER |
| AMOUNT | INTEGER | 8 DIGIT | NOT NULL | NO | AMOUNT OF THE ORDER PLACED |
| ADDRESS | VARCHAR | 50 VARCHAR CHARACTERS | NOT NULL | NO | ADDRESS WHERE THE PRODUCTS WILL BE DELIVERED |
| PAYMENT METHOD | VARCHAR | 10 CHARACTERS | NOT NULL | NO | PAYMENT CAN BE MADE THROUGH :- BANK, UPI, COD ETC. |

**TABLE 6.6 DELIVERY TABLE**

# 6.2 COMPONENT LEVEL DESIGNING

## 6.2.1 PSEUDOCODE FOR LOGIN MODULE

1. GET LoginID
2. GET Password
3. IF (LoginID == EnteredUsername && Password == EnteredPassword) THEN
4. Login Successful
5. ELSE
6. Login Failed.
7. ENDIF

### 6.2.2 CODE FOR LOGIN MODULE

```
<!DOCTYPE html>
<html>
    <head>
        <title>
            Log In
        </title>
        <link rel="stylesheet" href="registration.css">
    </head>
    <body>
        <form action="validation.php" method="POST">
            <div class="main">
                <div class="register">
                    <h2>Log In Here</h2>
                    <form action="" id="register" method="post">
```

```
                        <label>Email : </label>
                        <br>
                        <input type="text" name="email" id="name"
placeholder="Enter your valid email">
                        <br><br>
                        <label>Password : </label>
                        <br>
                        <input type="text" name="password" id="name"
placeholder="Enter password">
                        <br><br>

                        <input type="submit" value="Login"
name="submit" id="submit">
                    </form>
                </div>
            </div> <!--end main-->
        </form>
    </body>
</html>
```

```
*{
    margin: 0;
    padding: 0;
}
body,html
{
    min-height: 100%;
}
body{
    background: url("images/register_form_bg.jpg");
    background-size: cover;
    background-repeat: no-repeat;
    background-attachment: fixed;
    object-fit: scale-down;
    height: 100vh;
    display: flex;
    justify-content: center;
    background-position: center;
    margin-left: auto;
    margin-bottom: auto;
    margin-right: auto;
    margin-top: 100px;
    /* background-position: -400px 0px; */
    /* background-position: unset; */

}
div.main{
    width: 400px;
    margin: 100px auto 0px auto;
```

```
}
h2{
    text-align: center;
    padding: 20px;
    font-family: sans-serif;
}
div.register{
    background-color: rgba(0,0,0,0.5);
    width: 100%;
    font-size: 18px;
    border-radius: 10px;
    border: 1px soid rgba(255,255,255,0.3);
    box-shadow: 2px 2px 15px rgba(0,0,0,0.3);
    color:#fff;
}
form#register{
    margin: 40px;
}
label{
    font-family: sans-serif;
    font-size: 18px;
    font-style: italic;
}
input#name{
    width: 300px;
    border:1px solid #ddd;
    border-radius: 3px;
    outline: 0;
    padding: 7px;
    background-color: #fff;
    box-shadow: inset 1px 1px 5px rgba(0,0,0,0.3);
}
input#submit{
    width: 200px;
    padding: 7px;
    font-size: 16px;
    font-family: sans-serif;
    font-weight: 600;
    border-radius: 3px;
    background-color: rgba(79, 49, 30, 0.8);
    color: #fff;
    cursor: pointer;
    border: 1px solid rgba(255,255,255,0.3);
    box-shadow: 1px 1px 5px rgba(0,0,0,0.3);
    margin-bottom: 20px;
}
label,span,h2{
    text-shadow: 1px 1px 5px rgba(0,0,0,0.3)
}
```

## PHP CONNECTION

```
<?php

    $conn = mysqli_connect("127.0.0.1", "root", "",
"registration_database");

        // Check connection
        if($conn === false){
            die("ERROR: Could not connect. "
                .mysqli_connect_error());
        }

        // Taking all values from the form data(input)
        if(isset($_POST['submit']))
        {

        $email = $_POST['email'];
        $password = $_POST['psw'];
        }


        // Performing insert query execution
        // here our table name is registered_user
        $sql = "INSERT INTO registered_users (email,psw)VALUES
        ('$email','$password')";

        if(mysqli_query($conn, $sql))
        {


        } else{
            echo "ERROR: Sorry $sql. "
                . mysqli_error($conn);
        }

        // Close connection
```

```
        mysqli_close($conn);
?>
```

7.TESTING

7.1) We are performing White Box Testing for Place your order module

1. select item to add to cart
2. if(customer is not a valid user)
3. error
4. direct to signup
5. else
6. add to cart
7. calculate orders
8. payment gateway
9. while(payment details wrong)
10. error
11. redirect to cart
12. process payment
13. confirm order
14. generate fee slip
15. return order information

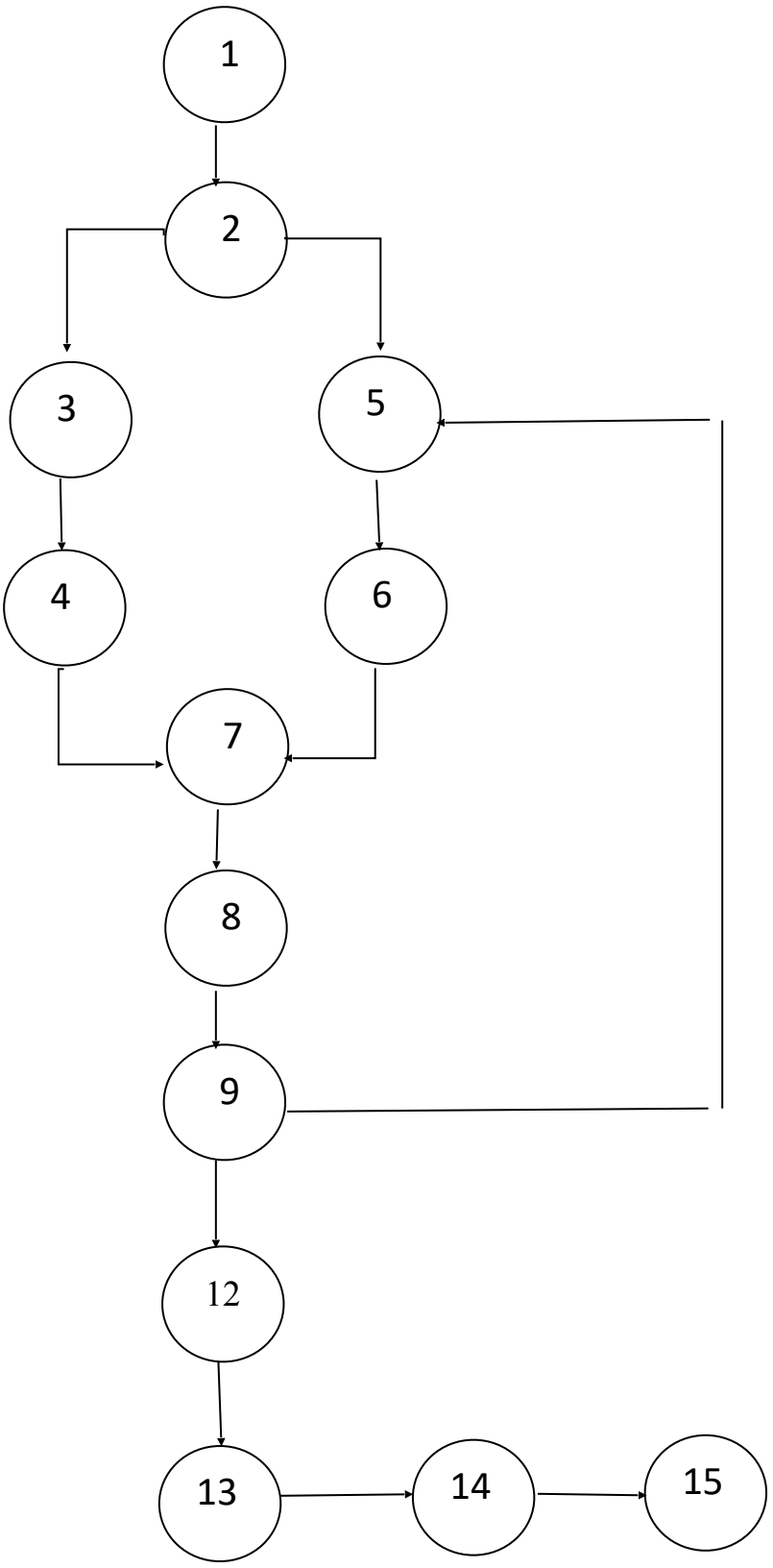



Fig 7.1 Control flow diagram

## CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH

• CC=Total Number of regions
=3
• CC=Number of edges-Number of nodes+2p
=9-8+2(1)
=3
• CC=Number of predicate nodes+1
=2+1
=3

## INDEPENDENT BASIC PATHS FROM THE GRAPH

**Path 1:** 1-> 2-> 3->4->7->8->9->12->13->14->15

**Path 2:** 1-> 2->3->4->7->8->9->5->6->7->8->9->12->13->14->15

**Path 3:** 1-> 2-> 5->6->7->8->9->12->13->14->15

**Path 4:** 1-> 2->5->6->7->8->9->5->6->7->8->9->12->13->14->15

## GENERATE TEST CASES FROM BASIS PATH SET

1) If customer is not a valid user, they will sign up and do payment with correct details.
2) If customer is not a valid user, they will sign up and do payment but payment details are incorrect.
3) Customer is a valid user and do payment with correct details.
4) Customer is a valid user and do payment but payment details are incorrect.

7.2) We are performing White Box Testing for

sign up

1. user interface click on sign up as
2. get the user input from sign up form
3. user.username = form.username.value
4. user.email = form.email.value
5. user.password = form.password.value
6. if(any fiel left blank or details not meeting conditions)
7. alert
8. return
9. .fetch(
10. send a post request to the server to create a new user account)
11. .then(handle the server response
12. if(response is ok)
13. redirect the user to the log in page or show success message
14. else
15. error
16. redirect to sign up)

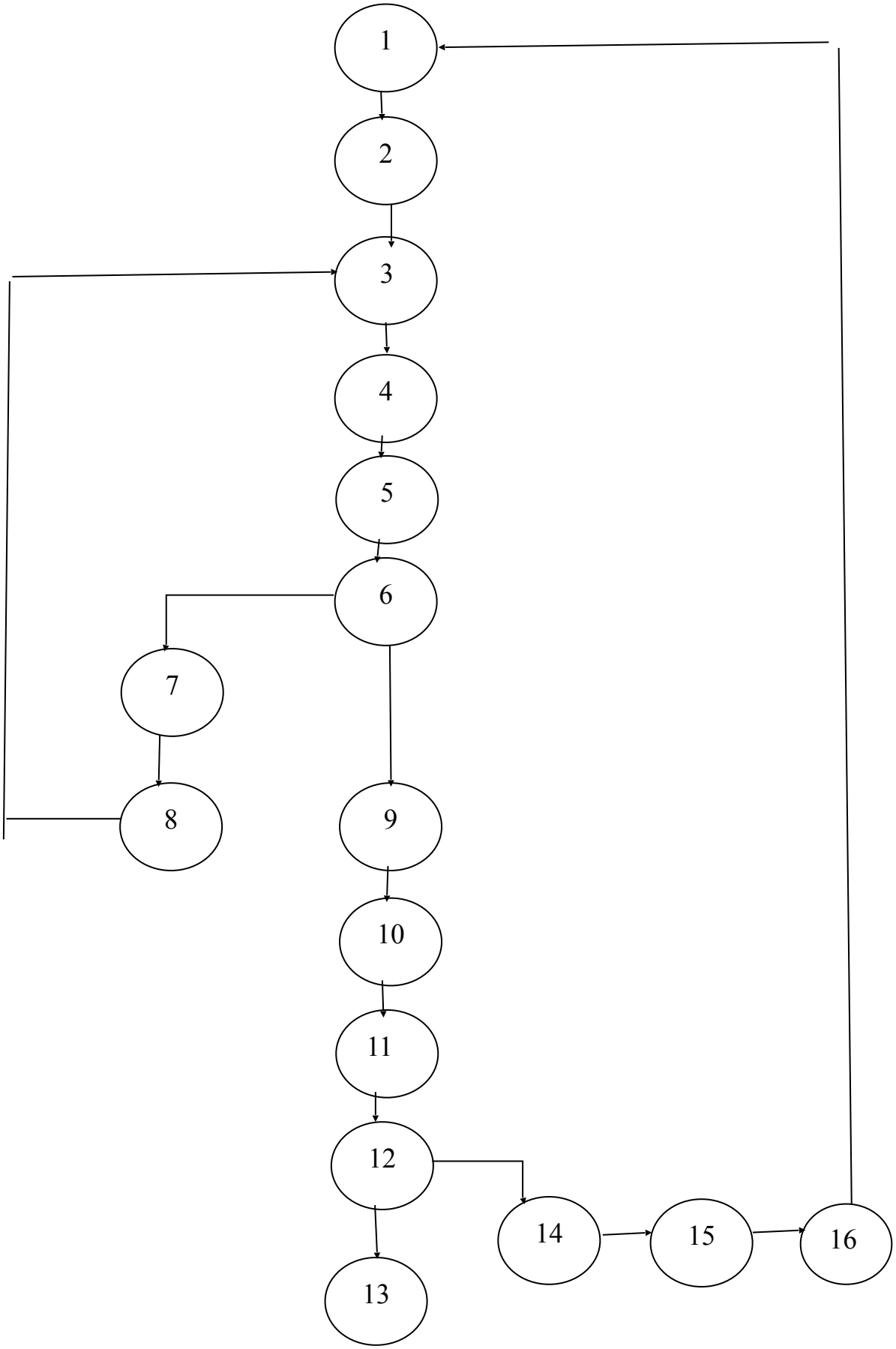



Fig 7.2 Control flow diagram

## CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH

• CC=Total Number of regions
=3
• CC=Number of edges-Number of nodes+2p
=9-8+2(1)
=3
• CC=Number of predicate nodes+1
=2+1
=3

## INDEPENDENT BASIC PATHS FROM THE GRAPH

**Path 1:** 1->2->3->4->5->6->7->8->3->4->5->6->->9->10->11->12->13

**Path 2:** 1->2->3->4->5->6->7->8->3->4->5->6->->9->10->11->12->14->15->16->1->2->3->4->5->6->7->8->3->4->5->6->->9->10->11->12->13

**Path 3:** 1->2->3->4->5->6->9->10->11->12->14->15->16->1->2->3->4->5->6->9->10->11->12->13

**Path 4:** 1->2->3->4->5->6->9->10->11->12->13

## GENERATE TEST CASES FROM BASIS PATH SET

1) User sign' s up with incorrect rule.
2) User sign' s up with incorrect rule and fill it again an error occurs.
3) User sign's up with correct rule but an error occurs
4) User sign's up with correct rule with any error

7.3) We are performing White Box Testing for

log out

1. click on logout
2. fetch(
3. send a post request to the server)
4. .then( response =>
5. if(response is ok)
6. redirect the user to the login page
7. endif
8. else
9. through error
10. endif)
11. return

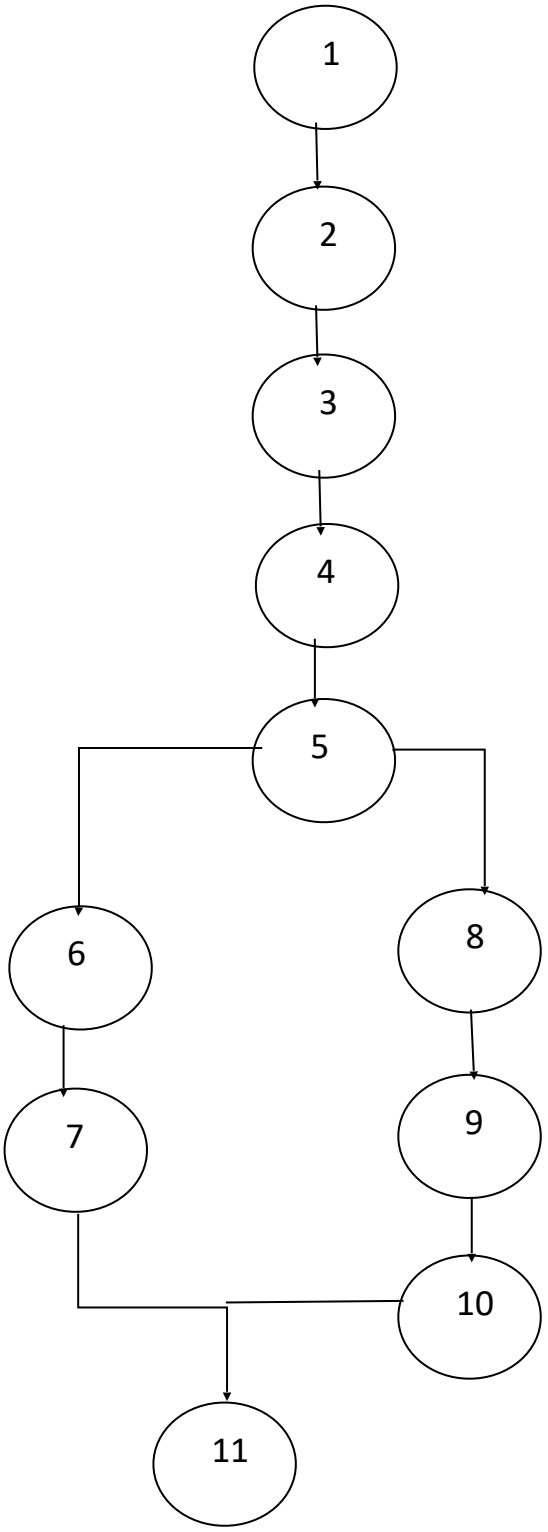



Fig 7.3 Control flow diagram

## CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH

- CC=Total Number of regions
  =2
- CC=Number of edges-Number of nodes+2p
  =6-6+2(1)
  =2
- CC=Number of predicate nodes+1
  =1+1
  =2

## INDEPENDENT BASIC PATHS FROM THE GRAPH

**Path 1:** 1->2->3->4->5->6->7->11

**Path 2:** 1->2->3->4->5->8->9->10->11

## GENERATE TEST CASES FROM BASIS PATH SET

1) User clicks on logout and get correct response
2) User clicks on logout and get no response

# 8.REFERENCES

1.Pressman, R. S., & Maxim, B. R., “Software Engineering: A Practitioner’s Approach”, 8th edition, (2015). McGraw-Hill.

2.Aggarwal, K. K., & Singh, Y., “Software Engineering”, 3rd edition, (2007), New
Age International Publishers.

3.https://www.engpaper.com/cse/artificial-intelligence-result-management.html

4.https://www.smartics.eu/confluence/display/PDAC1/How+to+document+a+Software+Development+Project

# 9.ANNEXURES

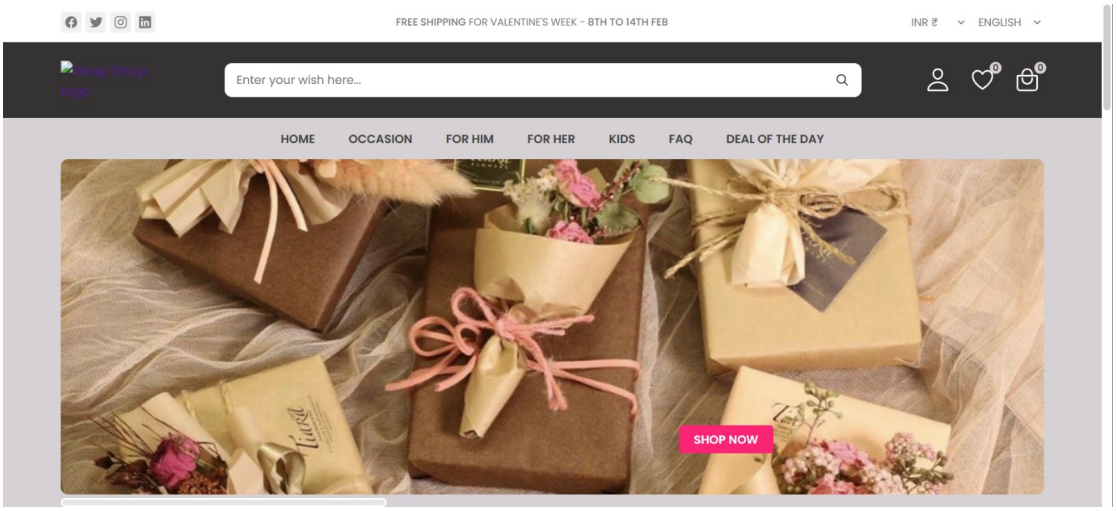



**Fig 9.1 HOME PAGE**

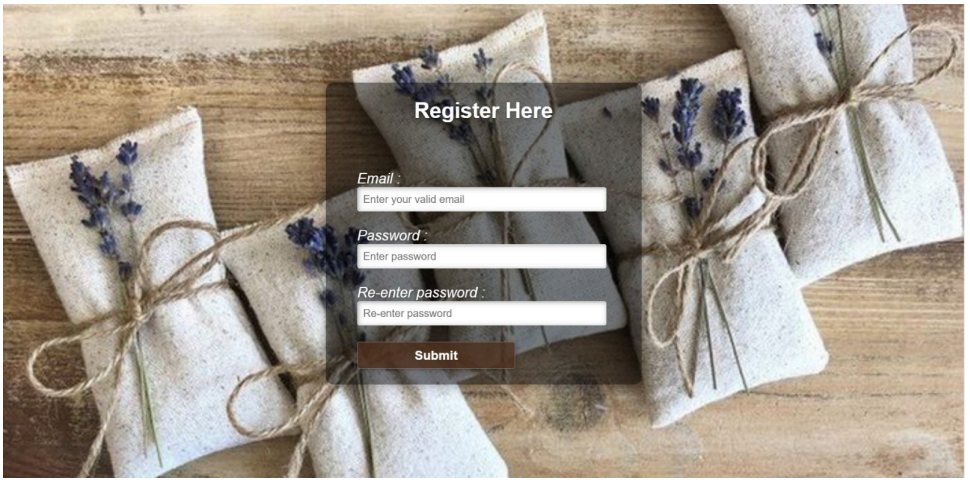



**Fig 9.2 SIGN UP**

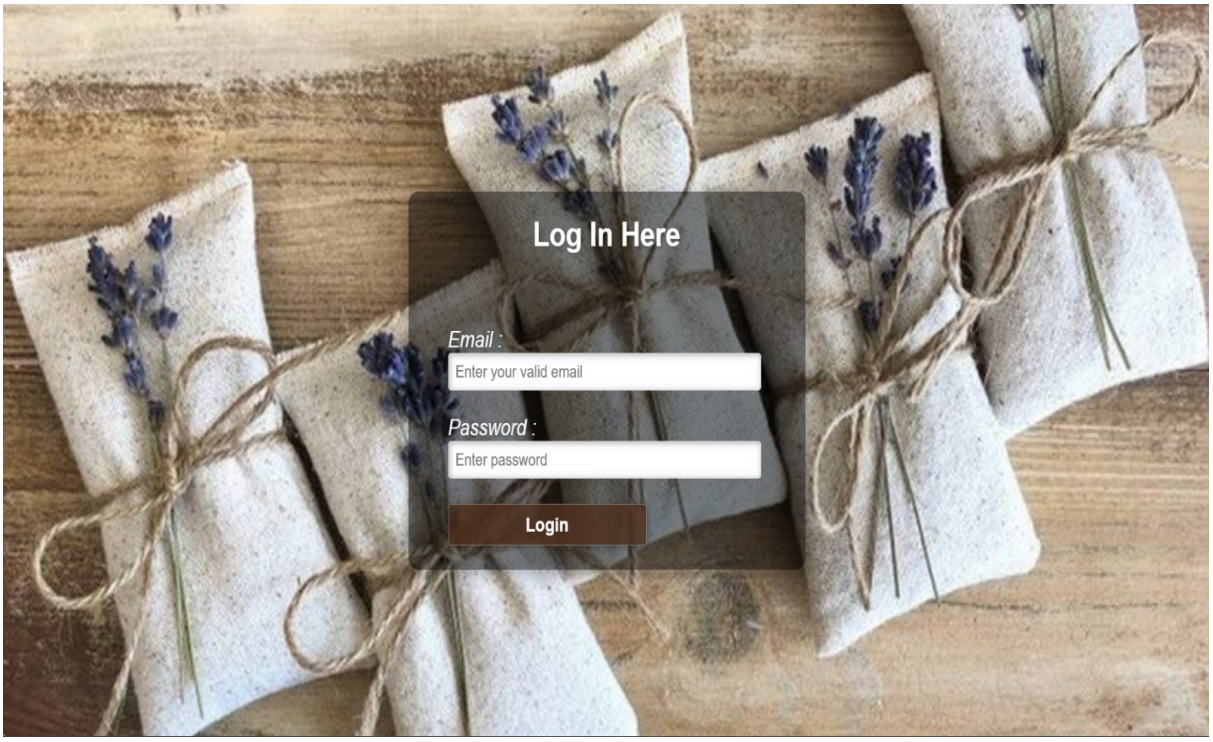



**Fig 9.3 LOG IN**

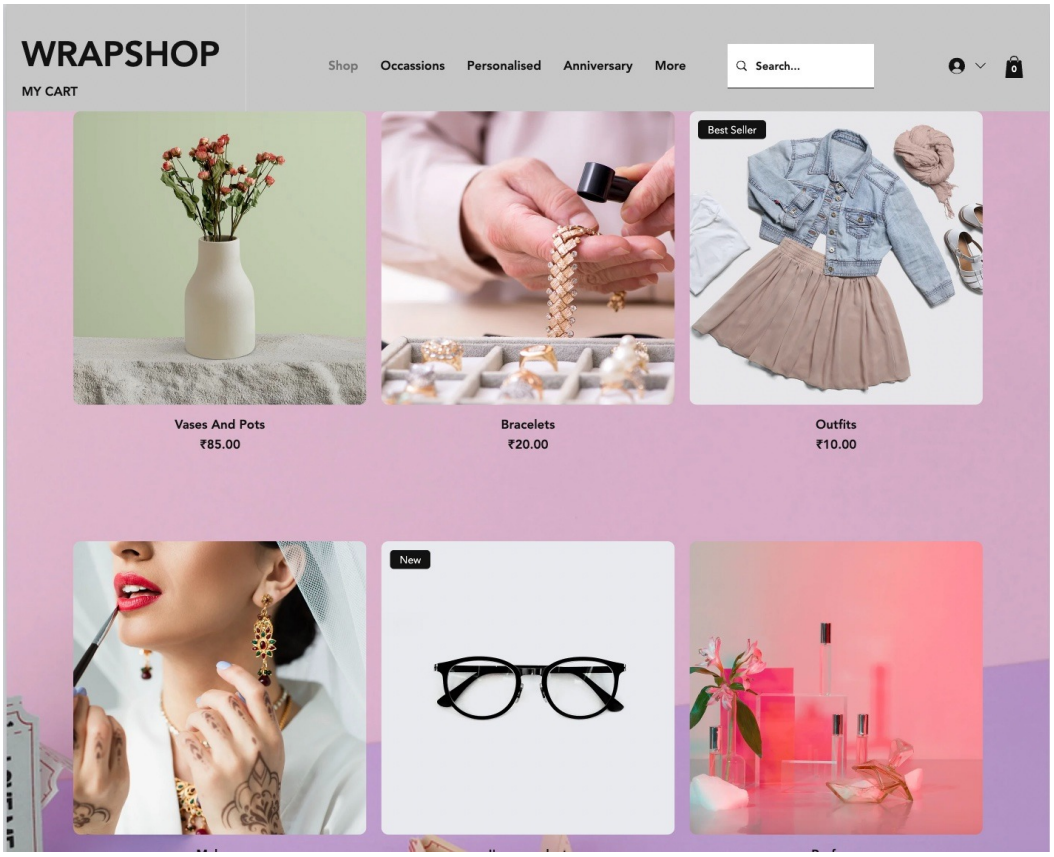



**Fig 9.4 DISPLAY PRODUCT**

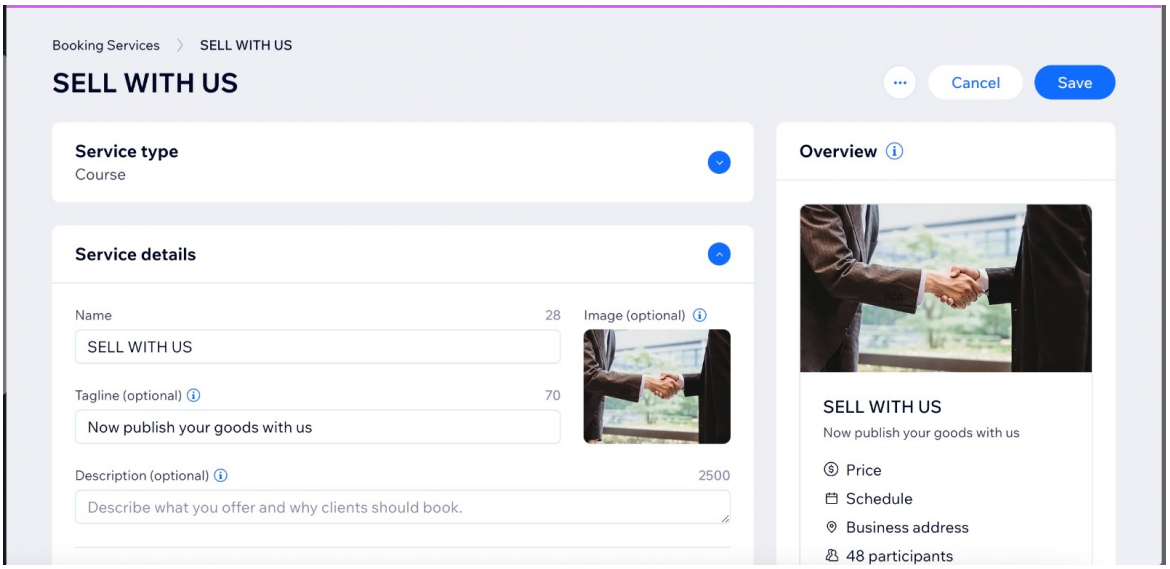



**fig 9.5 SELL WITH US**


**Fig 9.6 SHOP BY THEME**

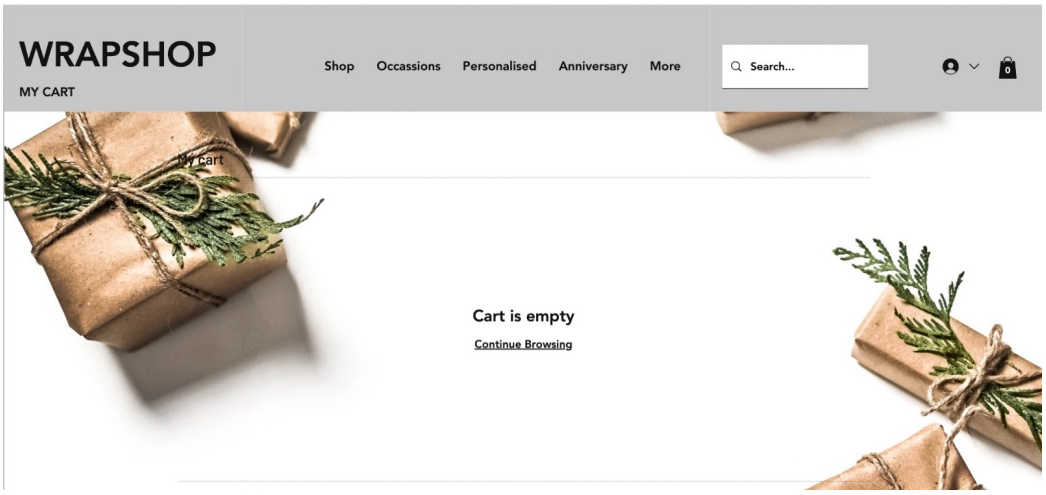



**Fig 9.7 CART**

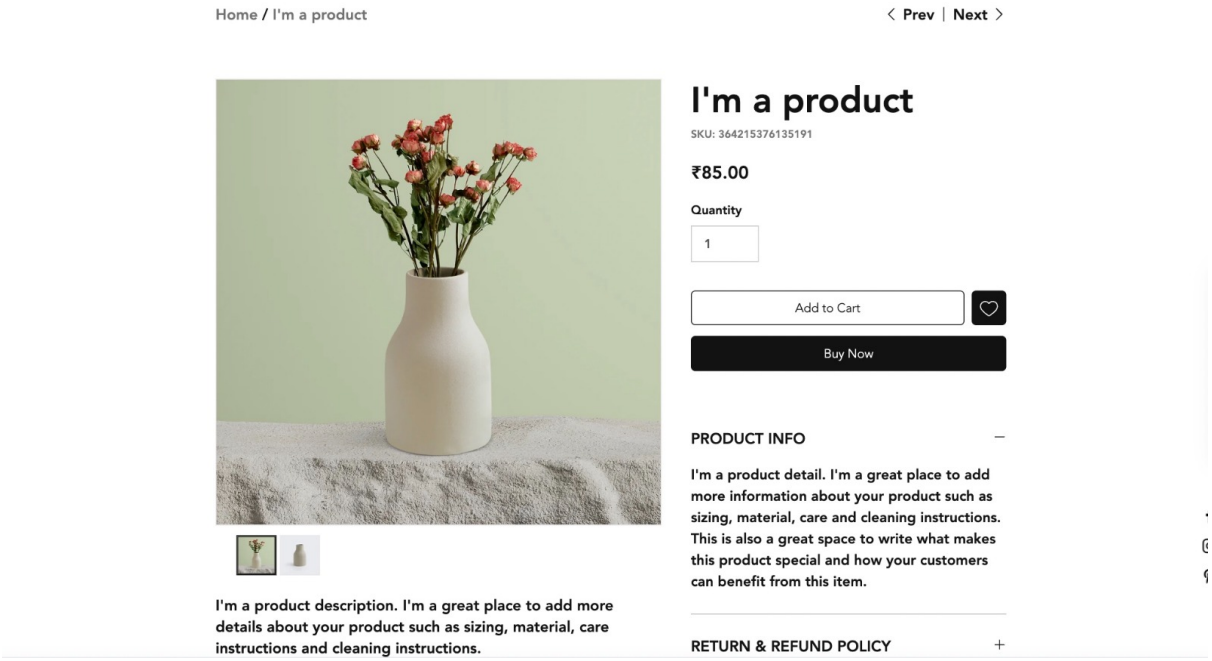



**Fig 9.8 PRODUCT DETAILS**

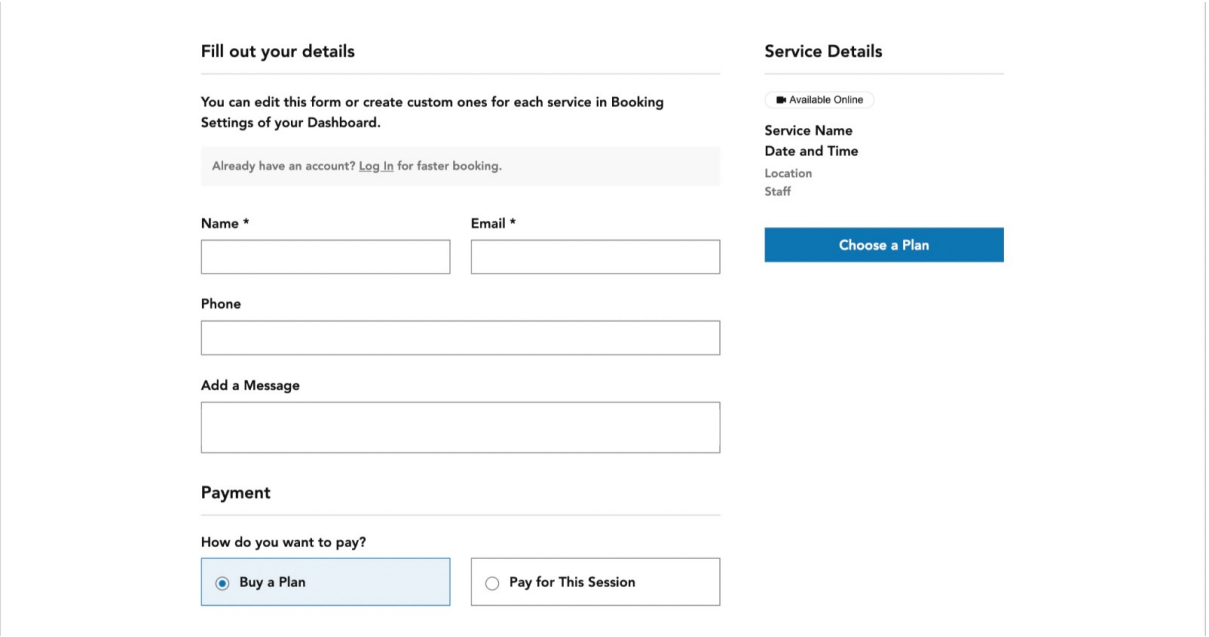



**Fig 9.9 DELIVERY DETAILS**

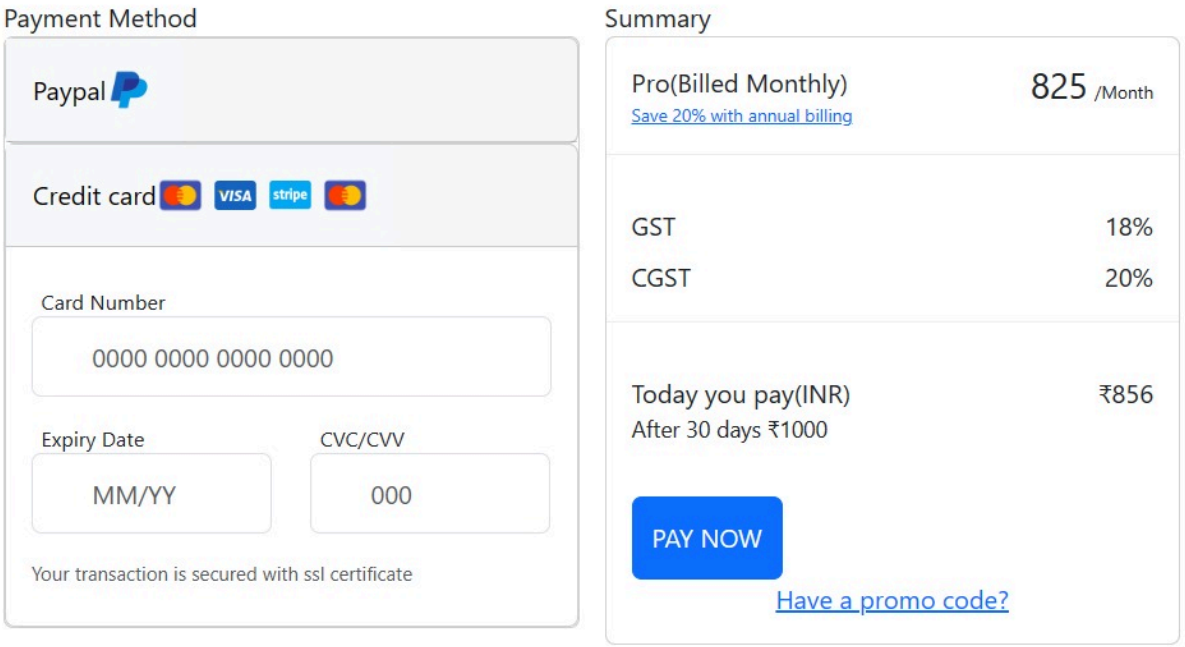



**Fig 9.10 PAYMENT**

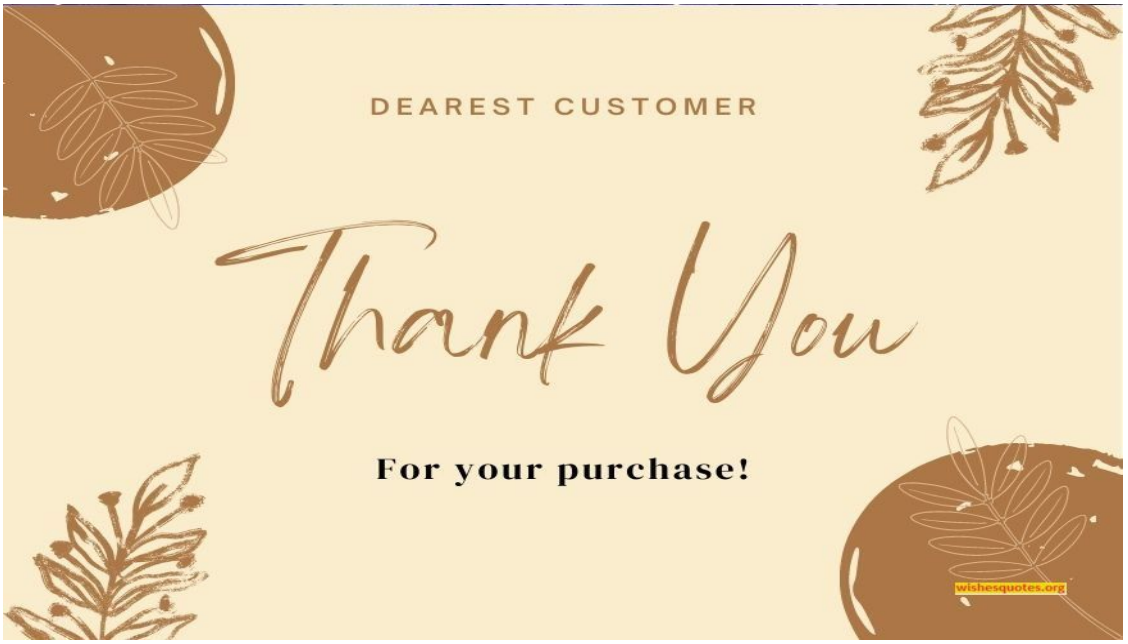



**fig 9.11 THANK YOU PAGE**