1. Synopsis

1.1 Introduction of the System

1.1.1 Project title: Sell Gros

1.1.2 Category: Web Application

1.1.3 Overview:

This project is a web-based application that provides an online platform for the grocery store or possible customers to order their basic, needed products. The main purpose of this simple project is to let customers buy their groceries without going to the shop or store, with a fixed price in every region. A government tender will be called to provide these items. User orders for those items are based on their requirements. Admin will assign the dealers to supply these items to particular regions and also add bill generation. The system will list all products with a fixed price, and customers can save their desired product to their shopping cart and checkout when they are done.

1.2. Background

1.2.1 Introduction of the Company

Not applicable

1.2.2 Brief note on Existing System

- The present scenario for shopping is to visit the shops and market manually and then from the available product list one needs to choose the item he or she wants and then pay for the same item mainly in cash mode.
- Existing system is not much user-friendly as one needs to go to the market physically and then select items only from the available list So mostly it is difficult to get the product as per our desire.
- Till now we have these types of systems without having the fixed price for the items in different regions. Our system will provide the fixed price for the items in every region.

1.3. Objective of the System

- Objective of the system is something that you plan to do or achieve.
- The main objective of the sell gross is to manage the customers grocery orders and assign dealer to particular orders. It manages all the information about Shopping, products, region, dealer.
- It may be useful during the pandemic situations.

1.4. Scope of the System

- Scope is the limitation that a process faces from the beginning to the end.
- People can shop from home whenever they want. No worrying about store closing, holidays etc.
- Online shopping offers great safety in shopping from home, especially in times like the present – when the global pandemic is threatening the health of all those who mingle in public.
- We can also add some extra features like ordering food, electronic gadgets in one website.

1.5. Structure of the System

1.5.1 Login module

The Login Module is a portal module that allows customer/dealers/admin to type their user's name and password to log in.

1.5.2 Registration module

1.5.2.1 Customer registration

This module is used to register customer by entering their necessary details.

1.5.2.2 Dealer registration

This module is used to register dealer by entering their necessary details.

1.5.3 Admin module

1.5.3.1 Product management

This module is used to manage the products and their Information.

1.5.3.2 Dealer authorization

This module manages the authorized dealer.

1.5.3.3 Dealer allotment

This module is used to allot the dealer for the customer's order.

1.5.3.4 Region management

This module is used to manage the region.

1.5.3.5 View Feedback

This module is used to manage the user's feedback and report.

1.5.4 Dealers' module

1.5.4.1 Pending orders

This module shows the active orders.

1.5.4.2 Previous orders

This module shows the previous orders.

1.5.4.3 Bill generation

This module generates the invoice/bill for active orders.

1.5.5 Customers module

1.5.5.1 Cart

This module shows the products which is to be ordered.

1.5.5.1.1 Modify cart

This module is used add or remove quantity of the products in a cart.

1.5.5.1.2 Place order

This module allows the customers to order required items.

1.5.5.2 Previous order

This module shows previous order which is placed by the customer

1.5.5.2.1 View bill

Customer can view the bill based on orders.

1.5.5.2.2 Feedback

Feedback module allows to get a review on how customers feel about dealer and products.

1.5.5.3 Pending order

This module shows the order that is pending.

1.6 System Architecture

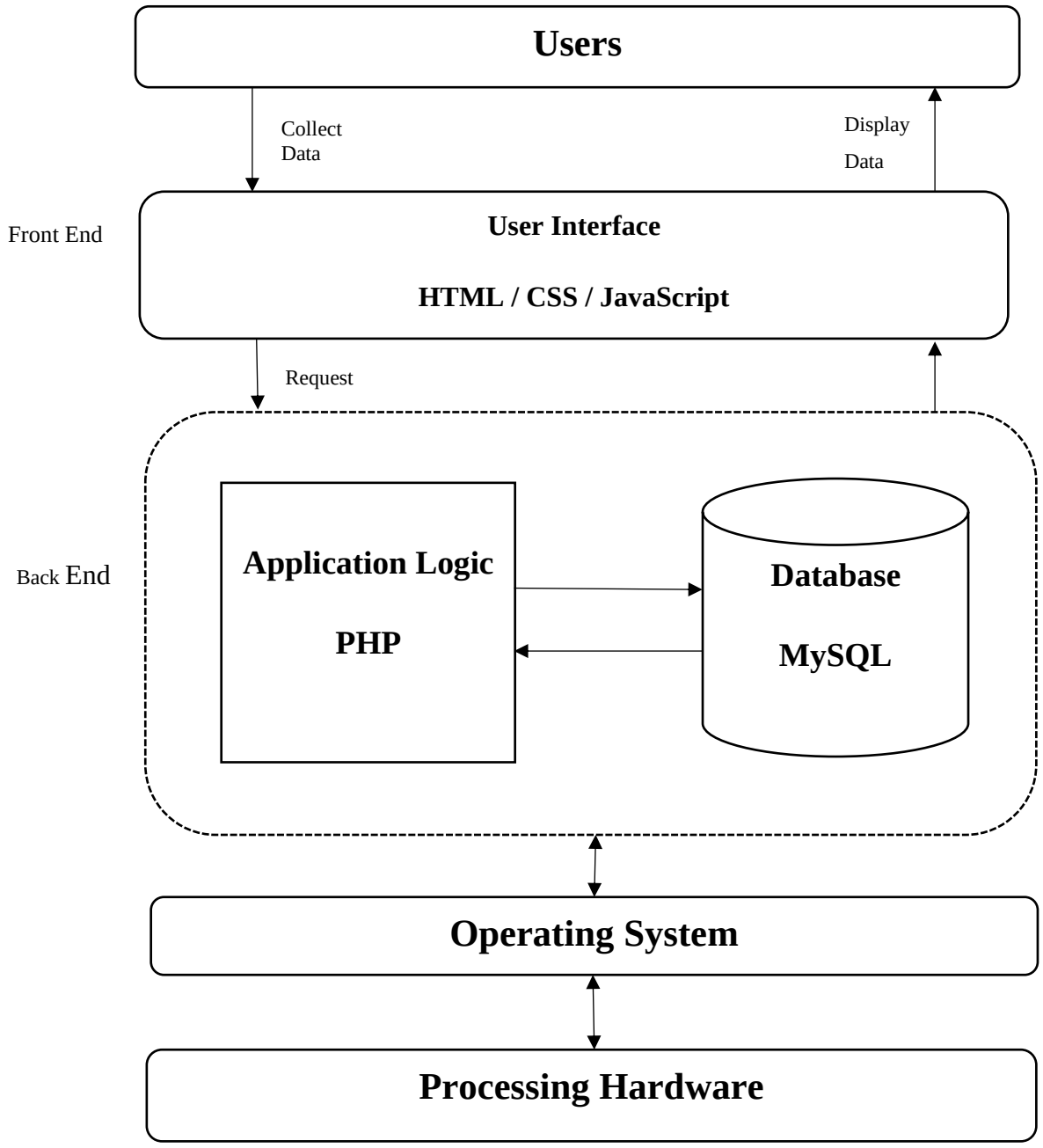


Figure 1

1.7 End User

- All type of people with some knowledge of using computer.

1.8 Software and Hardware needs for the development.

1.8.1 Hardware needs for the develop this product

- **CPU:** Intel or AMD processor with 64-bit support.
- **RAM**: 4 GB or higher.
- **Disk Storage:** 4 GB of free disk space or higher.
- Keyboard, Mouse, Laptop or Computer

1.8.2 Software needs for the develop this product

- **OS:** Windows 7 or higher.
- **Front End**: HTML/CSS
- **Back End**: PHP 5.0, phpMyAdmin.
- **Code Editor**: VS code
- **Web Server**: Apache Tomcat
- **Other Tools**: XAMPP.

1.9 Software and Hardware needs for the implementation.

1.9.1 Hardware needs for implementing this product

- **CPU:** Intel or AMD processor with 64-bit support.
- **RAM**: 4 GB or higher.
- **Disk Storage:** 4 GB of free disk space or higher.
- Keyboard, Mouse, Laptop or Computer

1.9.2 Software needs for implementing this product

- Web browser
- Internet

2. Software Requirement Specification

2.1 Introduction

Software requirement specification describes completely an external behavior of the proposed software. The software requirement specification is a document that completely describes what the proposed software should do without describing how the software will do it. It lays out functional and non-functional requirements. The basic purpose of SRS is to bridge the communication gap between the user and developer. Another important purpose is if developing the SRS is helping the clients to understand their own needs. Boundaries of software products are defined set of requirements. The software developer team designing, implements, tests, and delivers these requirements to client. This software requirement specification document will be the basic for final system. A high-quality software specification is a pre-requisite to high quality software and to reduce development cost.

2.2 Overall Description

This section describes the function of the project and their aim. It also includes the constraints and the requirements of the project.

2.2.1 Product Perspective

2.2.1.1 System Interface

This application runs in the latest version of Chrome or Firefox browser on windows, Linux, and mac.

2.2.1.2 User Interface

- **GUI (Graphical user interface)** is used to interact between user and system through different components.
- Each part of the user interface is designed to be as user-friendly.

2.2.1.3 Hardware Interface

- **Processor:** 1.9 gigahertz (GHz) x86- or x64-bit dual core processor. Recommended- 3.3 gigahertz (GHz) or faster 64-bit dual core processor.
- **RAM:** Minimum 4GB or higher
- **Storage:** Minimum 64GB is needed.

2.2.1.4 Software Interface

- Windows Xp and higher version and any compatible browser like chrome, Edge, Internet explorer.
- Xampp and MySQL

2.2.1.5 Communication Interface

Stable internet connection and browser.

2.2.1.6 Interface with Server

This application allows to interface with SQL Server, Xampp.

2.2.2 Product Function

2.2.2.1 Admin: Admin can add, update the price, item, and dealer.

2.2.2.2 Dealer: Dealer can view the customer details and he can generate bill.

2.2.2.3 Customer: Customer order the products and view the bill.

2.2.3 User Characteristics

2.2.3.1 Admin: He/she should have the knowledge of manipulating web applications.

2.2.3.2 Dealer: Basic knowledge of using computer and smartphone.

2.2.3.3 Customer: He/she should know how to use the browser and computer.

2.2.4 General Constraints

Not applicable

2.2.5 Assumption and Dependencies

- For windows 11 1 gigahertz (GHz) or faster with 2 or more cores on a compatible 64-bit processor or System on a Chip (SoC). 4 gigabytes (GB). 64 GB or larger storage device.
- The system is dependent on the availability of an Apache Tomcat Server to run.

2.3 Special requirements

Not applicable

2.4 Functional requirements

2.4.1 Login module

In this customer/dealer/admin can login to their system using username and password.

Input: Username, password

Process: Check for username, password

Output: If username, password is valid customer/dealer/admin can login to their system.

2.4.2 Registration module

2.4.2.1 Customer registration

Input: Details of the customer

Process: Details are stored in database.

Output: Registration successful message will be displayed.

2.4.2.2 Dealer registration

Input: Details of the Dealer

Process: Details are stored in database.

Output: Registration successful message will be displayed.

2.4.3 Admin module

2.4.3.1 Product management

2.4.3.1.1 Add

Input: Enter the product name, price, image, description, stock, unit.

Process: It validates the product details. If it is valid then it stores into database otherwise shows error message.

Output: product details are stored in database and Successful message will be displayed.

2.4.3.1.2 Update

Input: Enter the product name, price, image, description, stock, unit which is to be updated.

Process: It validates the product details.

Output: Product will be updated, and successful message will be displayed.

2.4.3.1.3 Active/Inactive

Input: Select the toggle button to active or inactive the products

Process: Enables/disables the products which is added.

Output: Display the respective message.

2.4.3.2 Dealer Authorization

Input: Select accept or reject button for authenticating dealer.

Process: operation is used accept/reject dealer.

Output: Dealer status will be updated in database and gets respective message.

2.4.3.3 Dealer Allotment

Input: click on allot button.

Process: Assigning dealer based on order to supply orders.

Output: Dealer Id will be assigned to every order in database.

2.4.3.4 Region management

2.4.3.4.1 Add

Input: Enter Region name, pin code.

Process: It validates the region details. If it is valid then it stores into database otherwise shows error message.

Output: Display the successful message.

2.4.3.4.2 Update

Input: click on update and change region name, pin code

Process: It validates the updated region details. If it is valid then it stores into database otherwise shows error message.

Output: Region details will be updated in database and display the successful message

2.4.3.4.3 Active/Inactive

Input: Select the toggle button to active or inactive the region

Process: Enables/disables the region which is added.

Output: Display the respective message.

2.4.3.5 View Feedback

Input: click on view Feedback

Process: Feedback will be retrieved from database.

Output: Feedback will be displayed.

2.4.4 Dealer

2.4.4.1 Pending orders

Input: Click on pending orders.

Process: Orders will be retrieved from the database.

Output: Active order will be displayed.

2.4.4.2 Previous order

Input: Click on previous orders.

Process: Dealer assigned order will be retrieved from database.

Output: Previous orders will be retrieved from the database.

2.4.4.2 Bill Generation

Input: Click on bill generate.

Process: The bill will be generated stored to database.

Output: Bill will be displayed.

2.4.5 Customer

2.4.5.1 Cart

2.4.5.1.2 Modify cart

Input: Products will be loaded and select increase/decrease button.

Process: Product quantity can be increased or decreased.

Output: Changes are updated in database.

2.4.5.1.2 place order

Input: click on place order

Process: Order will be placed and updated in database.

Output: Display the successful message

2.4.5.2 Previous order

2.4.5.2.1 view bill

Input: Select orders.

Process: Calculation of amount and bill generation.

Output: Bill will be displayed.

2.4.5.2.2 Feedback

Input: click on feedback and report.

Process: Process the id and feedback or report and stores it.

Output: Notification message will be shown regarding feedback or report.

2.4.5.3 Pending orders.

Input: Click on Pending orders.

Process: Orders will be retrieved from the database.

Output: Active order will be displayed.

2.5 Design Constraints

2.5.1 Hardware Constraints

RAM:4 GB or higher

Storage:64 GB or higher

2.5.2 Software Constraints

HTML, CSS, PHP, MySQL, Internet, Browser.

2.5.3 Fault Tolerance

At the time of verification and validation invalid information will be removed.

Only valid data will be stored in the database.

2.5.4 Security

Only authorized user can access the application by using their username and password.

2.5.5 Standard Compliances

Not applicable

2.6 System Attributes

- **Availability**

  Available for 24x7

- **Portability**

  This application is machine independent and can be used in any systems.

- **Reliability**

  This application is reliable and works efficiently. All input data will be verified and validated to avoid system failure.

- **Maintainability**

  The application is maintained in a better way by providing updates. If any requirement needed it should be updated immediately.

- **Scalability**

  The application's functionalities can be changed as per the user's demand. Software will remain stable and works constantly while making changes or updating or upgrading software.

2.7 Other Requirements

Not applicable

3. System Design

3.1 Introduction

- System design is the process of defining the architecture, module interfaces and data for a system to satisfy specified requirements.
- The purpose of the design phase is to plan the solution of the problem specified by the requirement documents.
- This is the first step that moving from problem domain to the solution domain.
- The design of the system is essentially a blueprint or a plan for a solution for the system.

3.2 Assumption and constraints

- For windows 11 1 gigahertz (GHz) or faster with 2 or more cores on a compatible 64-bit processor or System on a Chip (SoC). 4 gigabytes (GB). 64 GB or larger storage device.
- The system is dependent on the availability of an Apache Tomcat Server to run.
- 3 months of time for completion.

3.3 Functional Decomposition

Functional decomposition is the process of taking a complex process and breaking it down into its smaller, simpler parts. Using functional decomposition large or complex functionalities are more easily understood. It is mainly used during project analysis phase, so each phase can be viewed as software. So, this has modular with some sub modules.

3.3.1 System Software Architecture

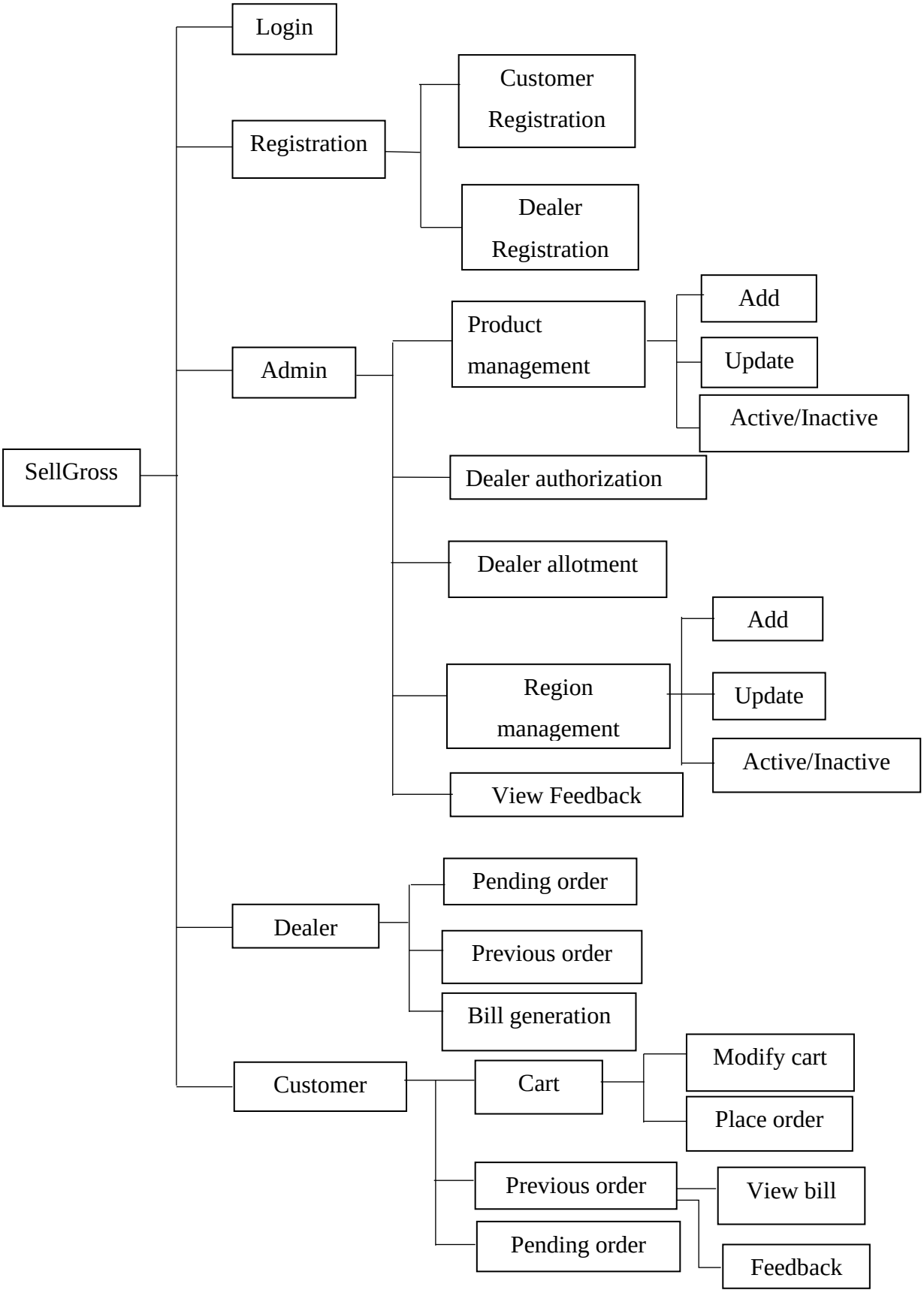
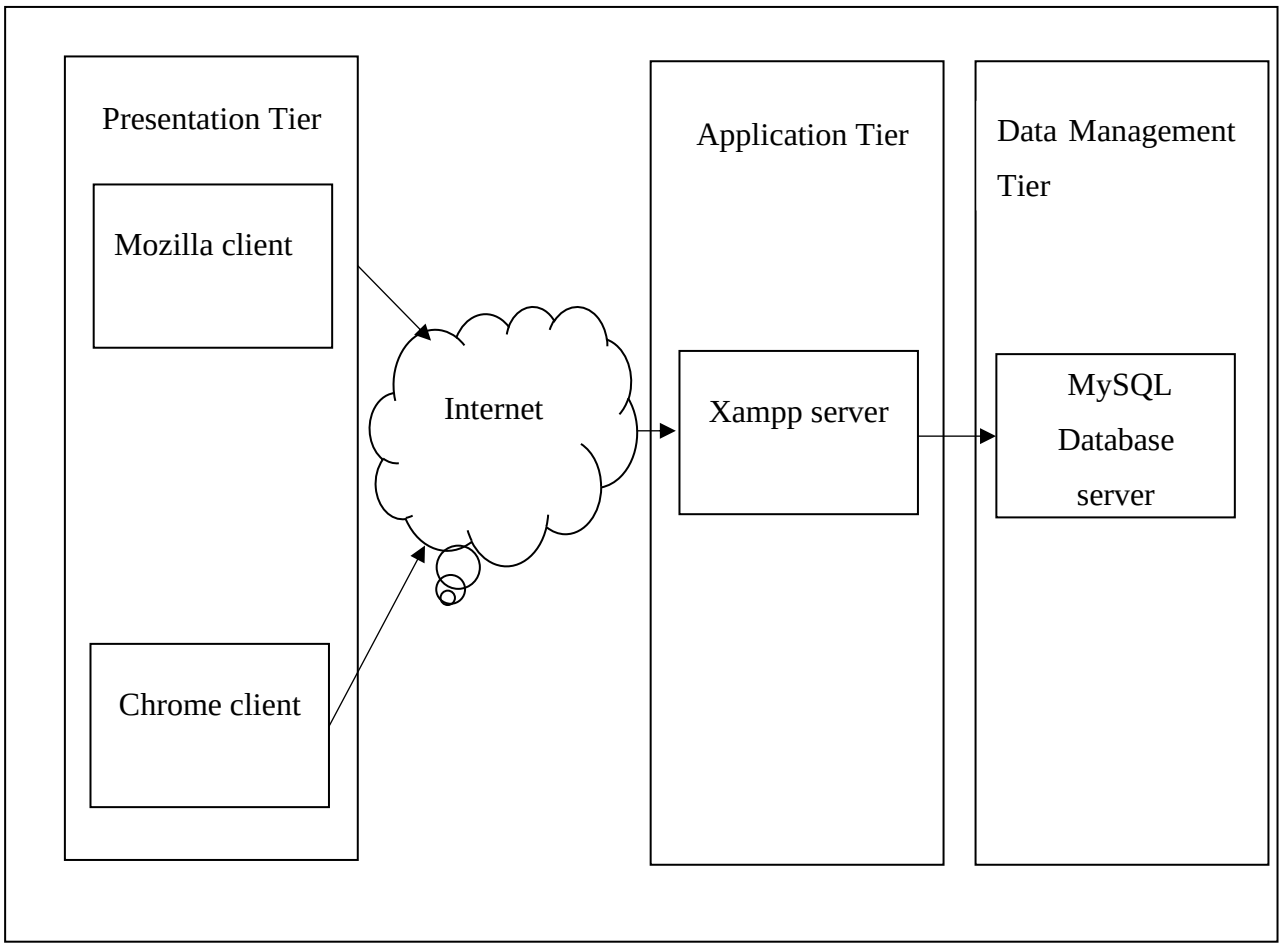


Figure 2

3.3.2 System technical architecture



3.3.3 System hardware architecture

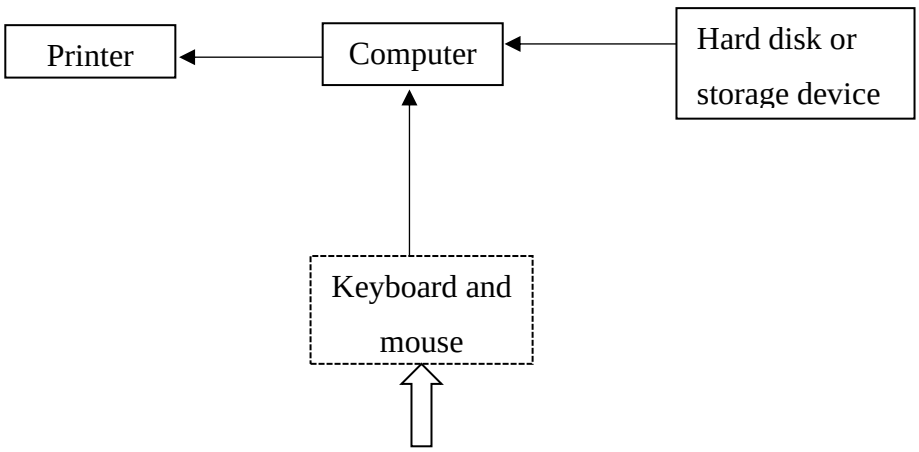


Figure 3

3.3.4 External Interface

Not applicable

3.4 Description of programs

3.4.1 Context Flow Diagram

In CFD entire system is considered as a single process. Context flow diagram shows input and outputs of the system. It shows all the external entities that interact with the system and how the data flows between these external entities and system.

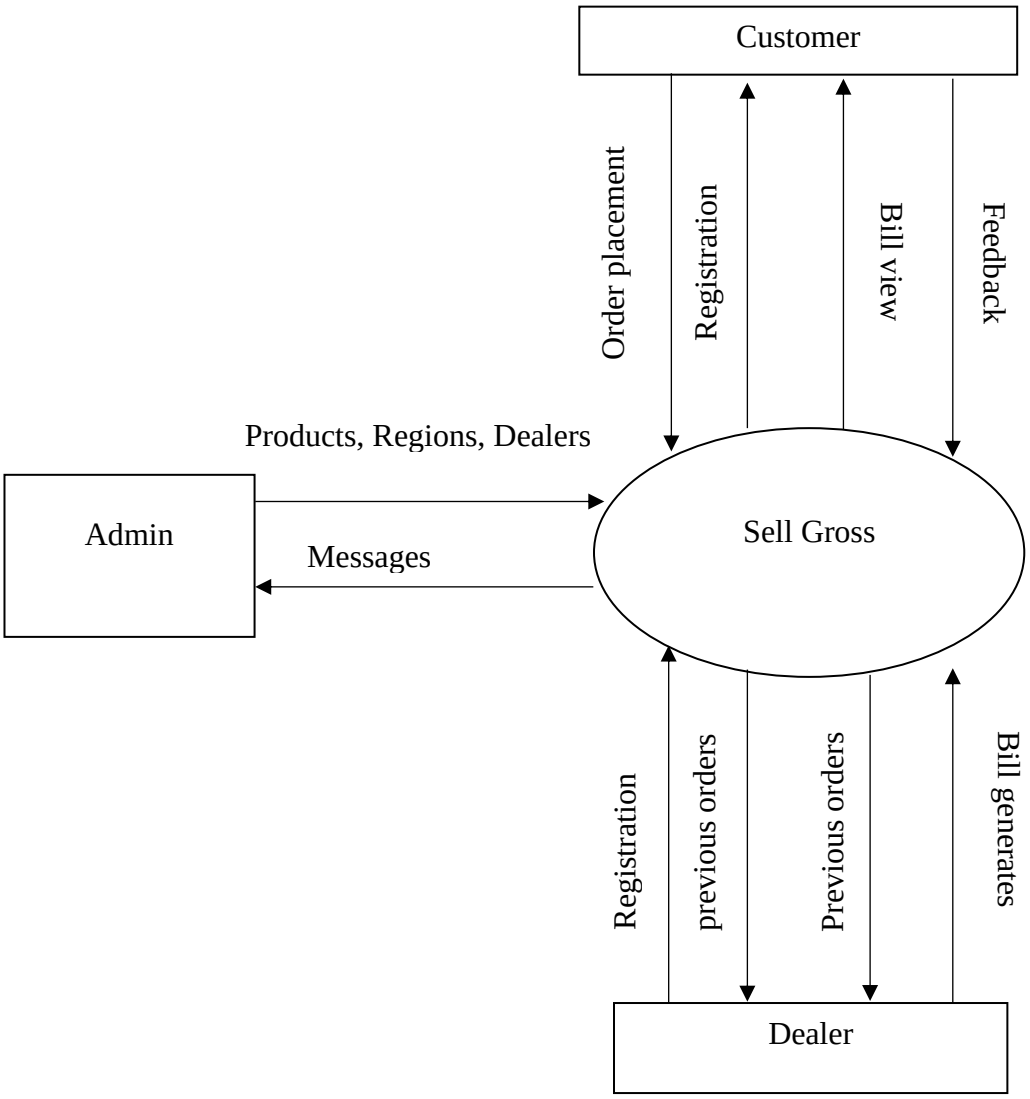
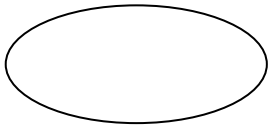


Figure 4

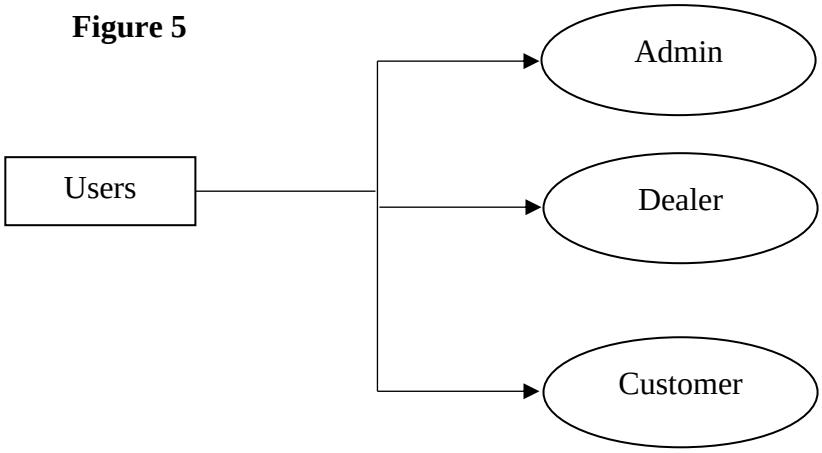
3.3.1 Data Flow Diagram

Data flow diagram shows the flow of data through system. Data flow diagrams also called the data flow graphs. It views a system as a function that transforms the inputs into desired outputs. It aims to capture the transformation that taken place within a system to the input data so that eventually the output data is produced.

Symbol	Name	Description
	Process	It performs transformation of data from one state to another.
	Sink/Source	It represents the external entity that may be either source or sink.
	Flow of data	It represents the flow of data from source to destination.
	Data Source/Data storage	It is the place where data is stored.

Top level DFD

Figure 5



3.5 Description of the components

3.5.1 Login module

3.5.1.1 Input

User name and password

3.5.1.2 Process

Read details and validation

3.5.1.3 Output

Logged in to their respective pages.

3.5.1.4 Interface with another functional components

Independent

3.5.1.5 Resource allocation

Admin/Dealer/Customer table

3.5.1.6 User interface

Textboxes are provided to enter the username and password. Login button is provided to move to next page

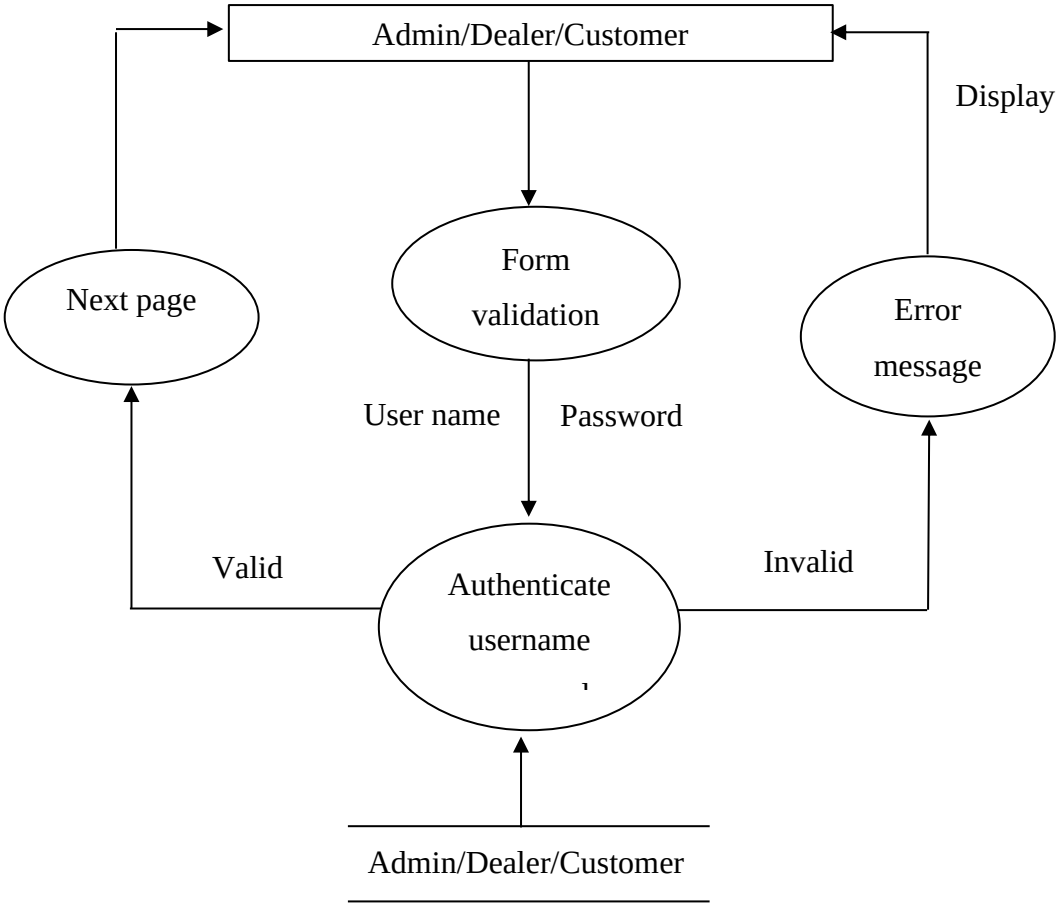
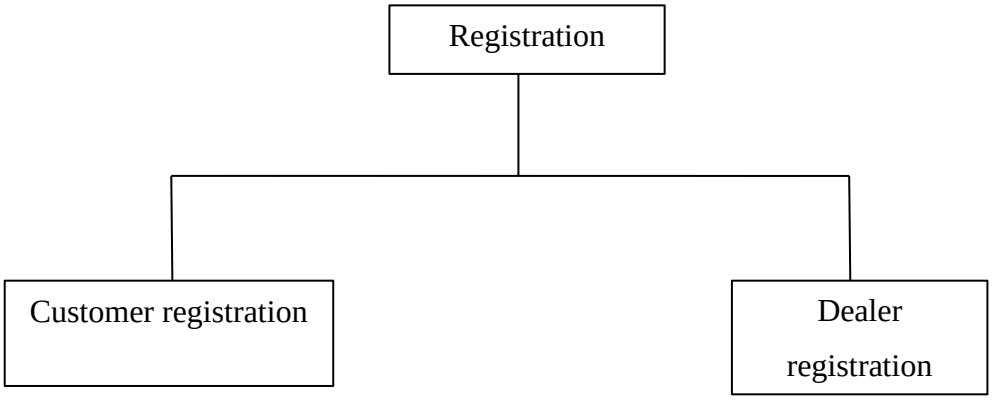


Figure 6

3.5.2 Registration module

Figure 7



3.5.2.1 Customer Registration

3.5.2.1.1 Input

Registration details of the customer.

3.5.2.1.2 Process

Read customer registration details

3.5.2.1.2 Output

Registration successful message view be display and stored in database.

3.5.2.1.3 Interface with another functional components

Independent

3.5.2.1.4 Resource allocation

Customer table

3.5.2.1.5 User interface

Textboxes are provided to enter the details. Register button is provided to store the details.

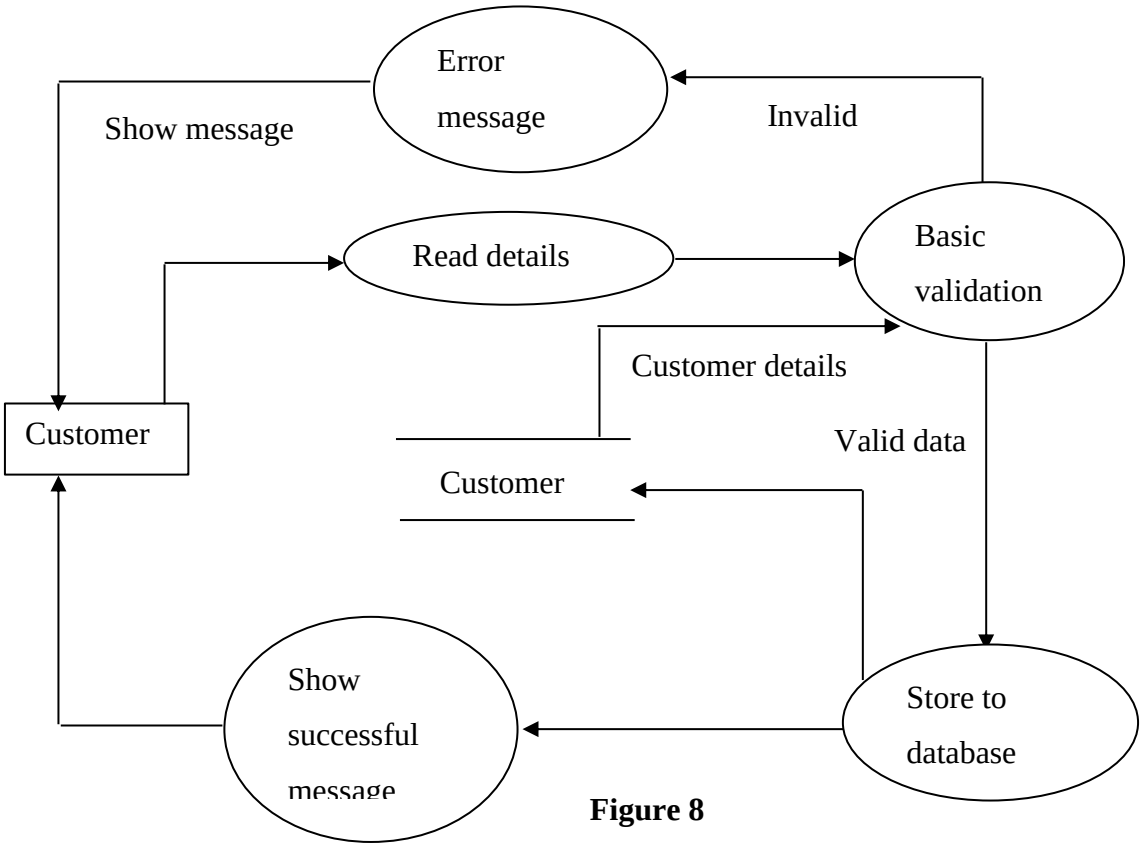


Figure 8

3.5.2.2 Dealer Registration

3.5.2.2.1 Input

Registration details of the Dealer.

3.5.2.1.2 Process

Read Dealer registration details

3.5.2.2.2 Output

Registration successful message view be display and stored in database.

3.5.2.2.3 Interface with another functional components

Independent

3.5.2.2.4 Resource allocation

Dealer table

3.5.2.2.5 User interface

Textboxes are provided to enter the details. Register button is provided to store the details.

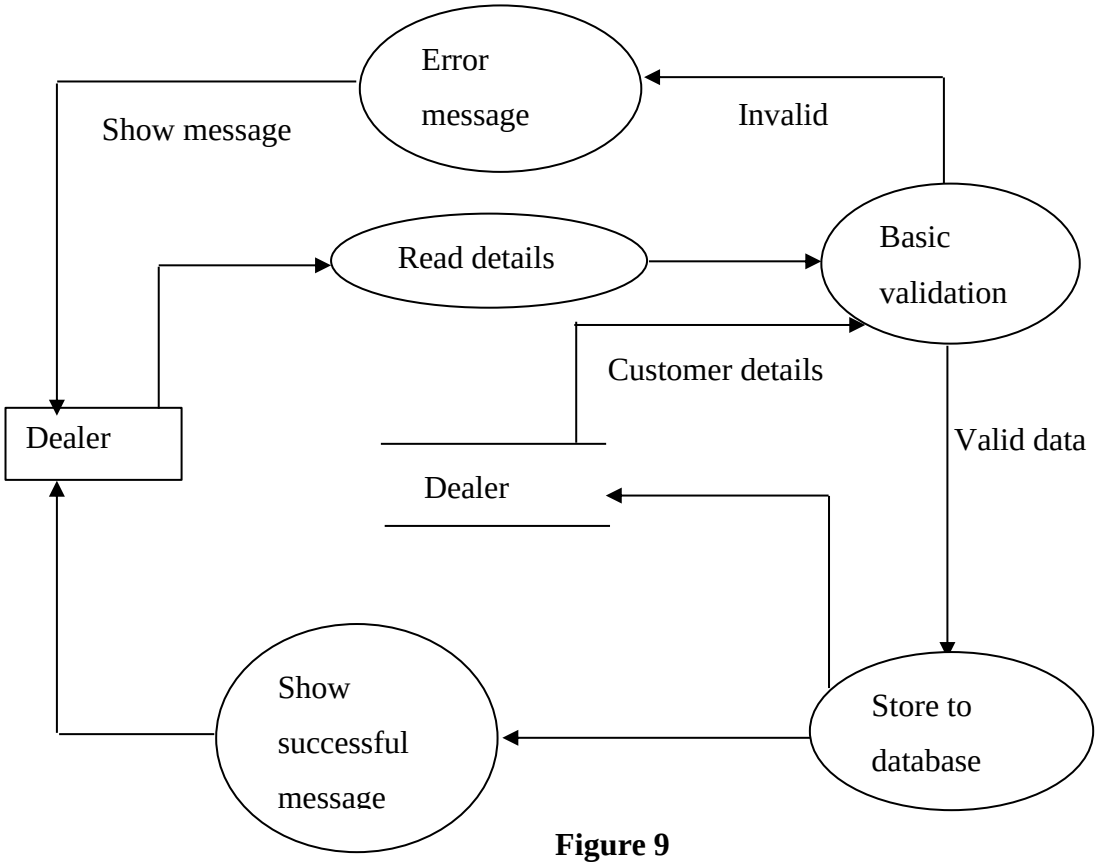


Figure 9

3.5.3 Admin module

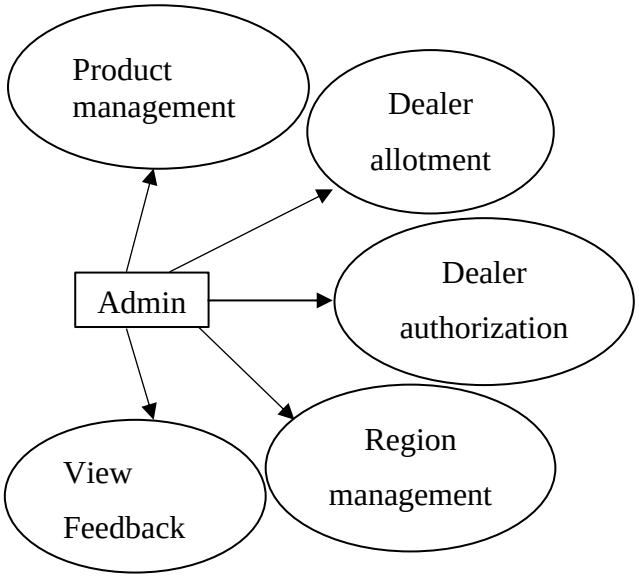
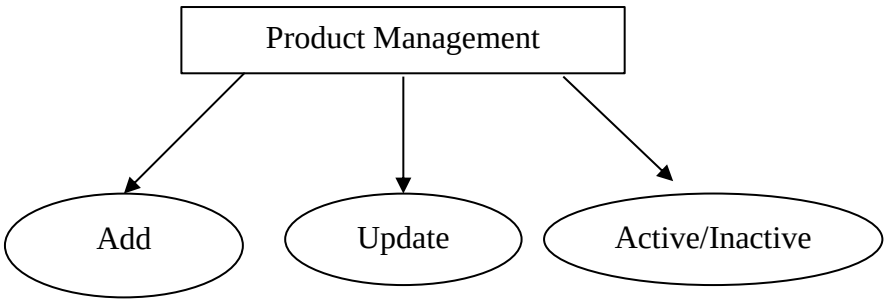


Figure 10

3.5.3.1 Product Management

Figure 11



3.5.3.1.1 Add

3.5.3.1.1.1 Input

Product Details- name, image, price, description, stock, Unit.

3.5.3.1.1.2 Process

Validates the products and Store to database

3.5.3.1.1.3 Output

Show successful message

3.5.3.1.1.4 Interface with another component

Independent

3.5.3.1.1.5 Resource allocation

Product table

3.5.3.1.1.6 User Interface

Textboxes are provided for entering product details and add products button is provided to add product

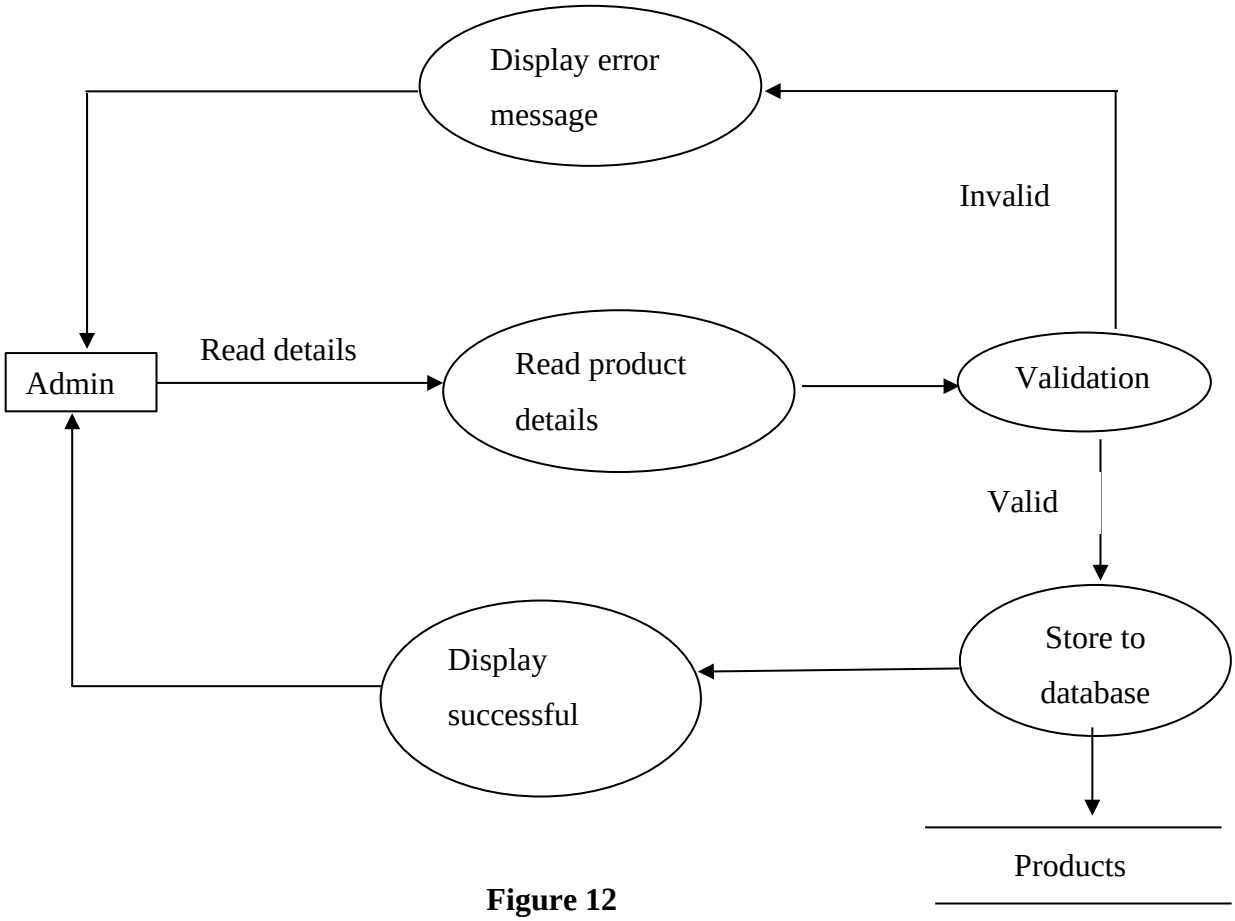


Figure 12

3.5.3.1.2 Update

3.5.3.1.2.1 Input

Enter Product Details- name, image, price, description, stock, Unit to be updated.

3.5.3.1.2.2 Process

Validate input and update database

3.5.3.1.2.3 Output

Changes are updated in database and successful message will be displayed.

3.5.3.1.2.4 Interface with another components

Independent

3.5.3.1.2.5 Resource allocation

Product table

3.5.3.1.2.6 User interface

Update button will be provided for update details. Textboxes are provided for updating details. Update button will help to save details to database.

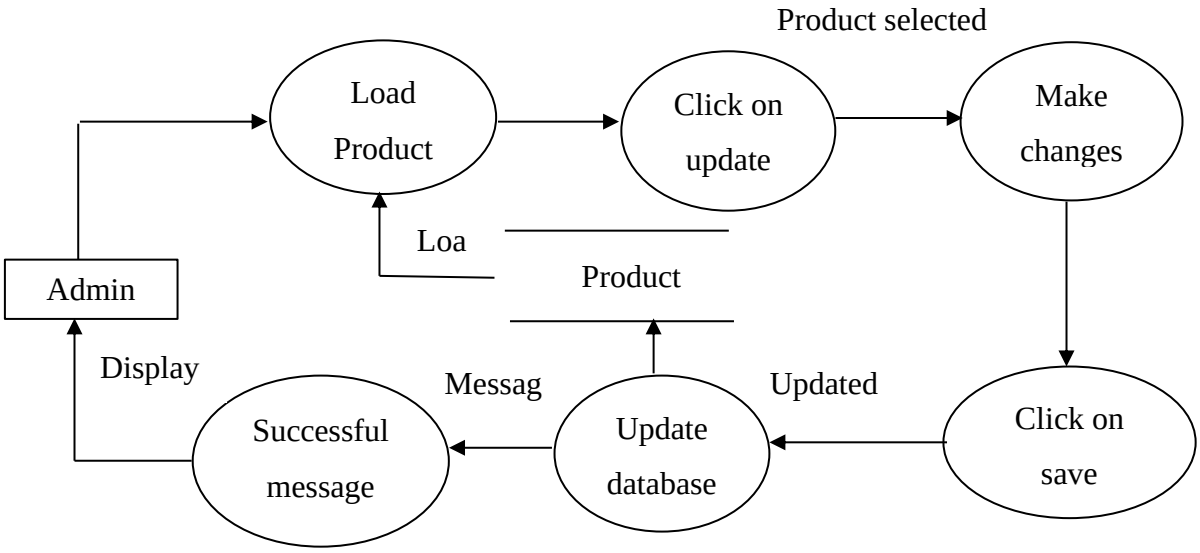


Figure 13

3.5.3.1.3 Active/Inactive

3.5.3.1.3.1 Input

Product ID

3.5.3.1.3.2 Process

Product will not be available to use.

3.5.3.1.3.3 Output

product will be disabled and product status will be updated

3.4.3.1.3.4 Interface with other components

Independent

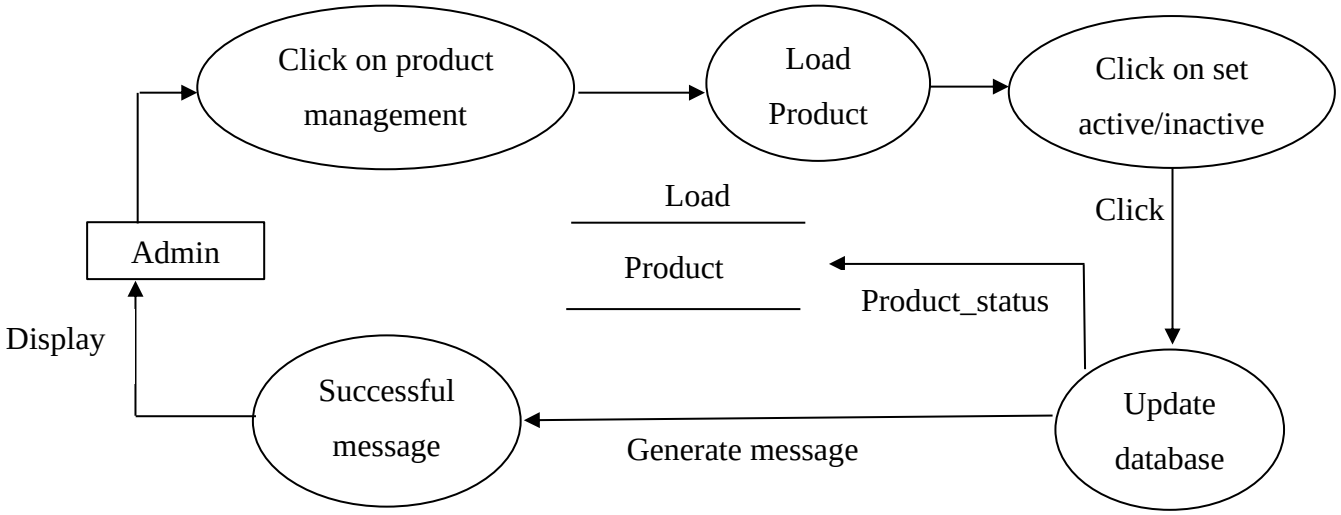
3.4.3.1.3.5 Resource allocation

Product table

3.4.3.1.3.6 User Interface

Item list will be displayed in the form of list. To active/inactive the product, set as active/inactive button is be provided.

Figure 14



3.4.3.2 Dealer Authorization

3.4.3.2.3 Input

Dealer details.

3.4.3.2.4 Process

Selection process and update table

3.4.3.2.5 Output

Successful message.

3.4.3.2.6 Interface with another functional components

Independent

3.4.3.2.7 Resource allocation

Dealer table

3.4.3.2.8 User interface

Approve and reject button is provided to authenticate.

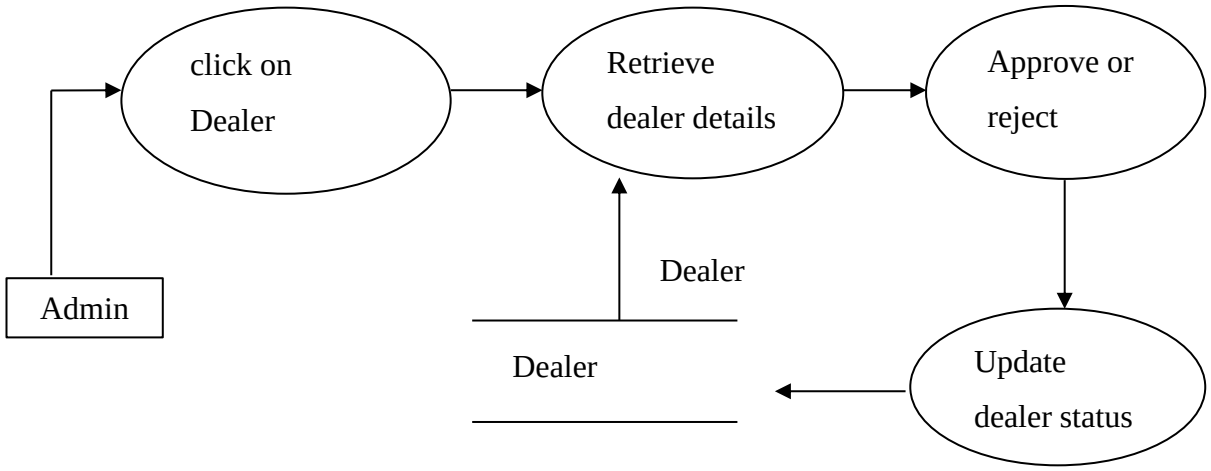


Figure 15

3.4.3.3 Dealer Allotment

3.4.3.3.3 Input

Dealer details and item order details.

3.4.3.3.4 Process

Dealer is allotted to orders and updated in database.

3.4.3.3.5 Output

Successful message

3.4.3.3.6 Interface with another functional components

Independent

3.4.3.3.7 Resource allocation

Dealer table and order table.

3.4.3.3.8 User interface

Dropdown list for selecting dealer and allot button to allot dealer.

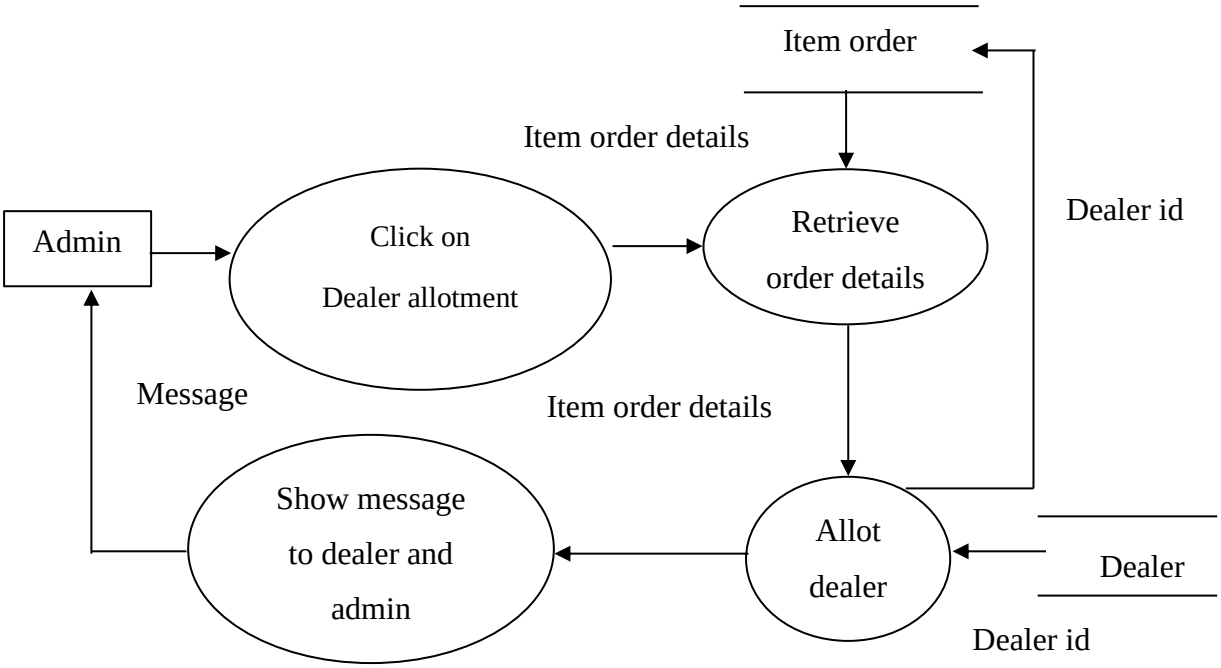


Figure 16

3.4.3.4 Region Management

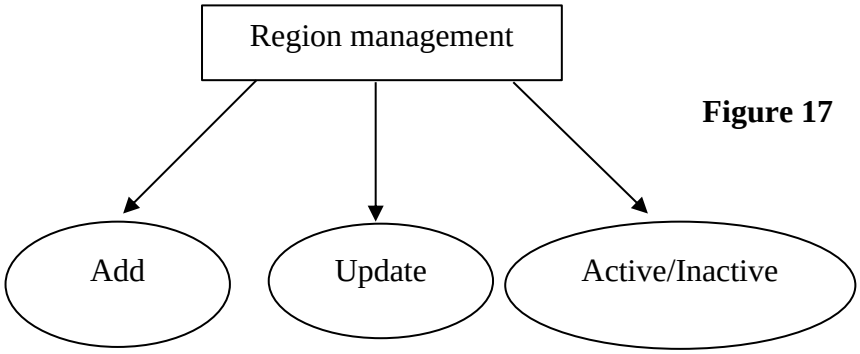


Figure 17

3.4.3.4.3 Add

3.4.3.4.3.5 Input

Region details- name, pin code

3.5.3.4.1.2 Process

Validation for existence

3.5.3.4.1.3 Output

Region will be added to database and success message will be shown.

3.5.3.4.1.4 Interface with other components

Independent

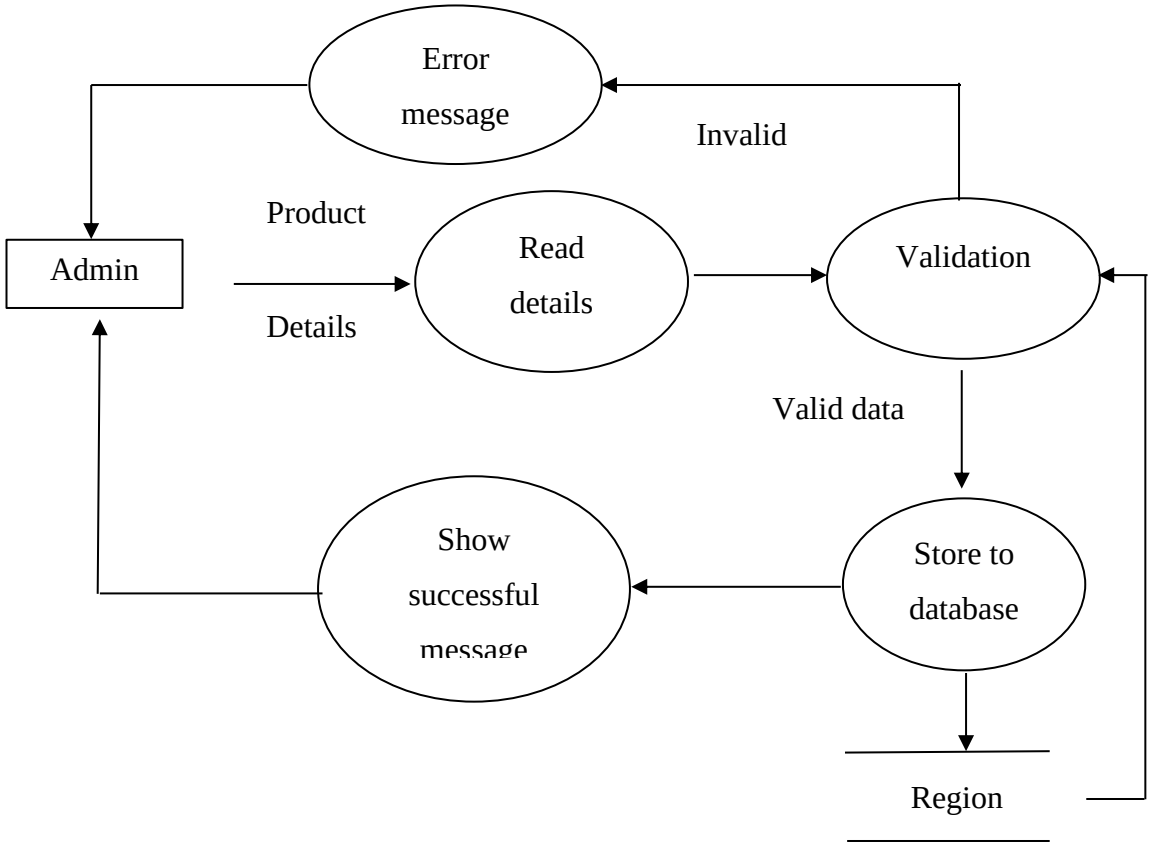
3.5.3.4.1.5 Resource allocation

Region table

3.5.3.4.1.6 User interface

Textboxes are provided for adding region details. By clicking add button region will be added to table.

Figure 18



3.5.3.4.2 Update

3.5.3.4.2.1 Input

Region details- name, pin code

3.5.3.4.2.2 Process

Update operation and store to database

3.5.3.4.2.3 Output

Successful message

3.5.3.4.2.4 Interface with other components

Independent

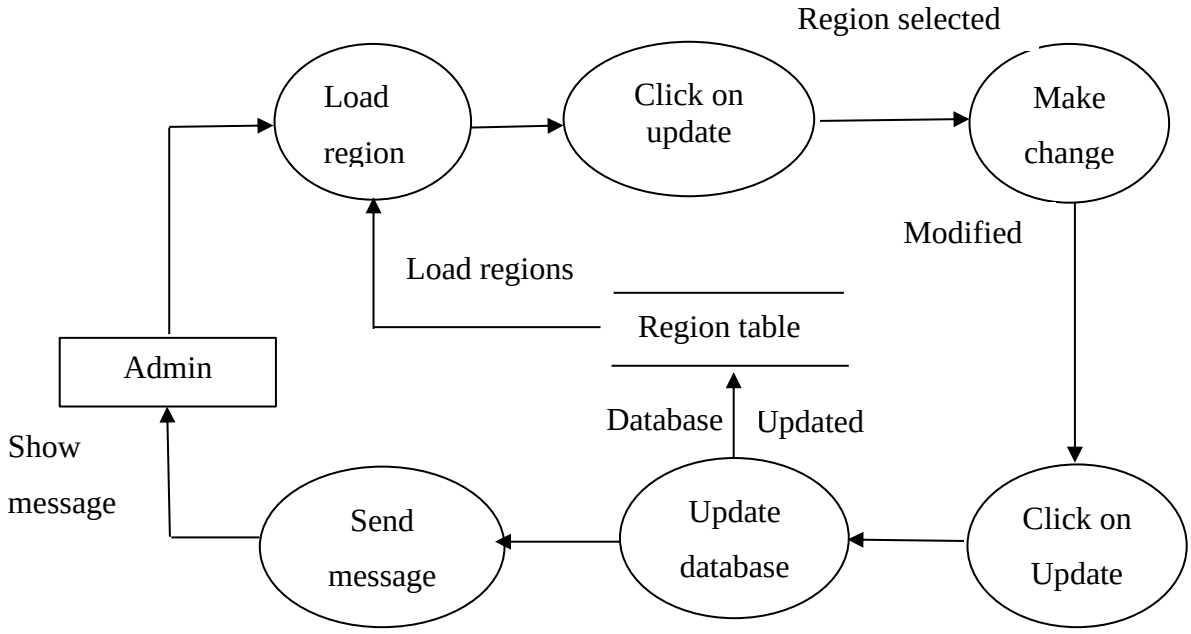
3.5.3.4.2.5 Resource allocation

Region table

3.5.3.4.2.6 User interface

Textboxes are provided for entering details. update button will be provided to update database.

Figure 19



3.5.3.4.3 Active/Inactive

3.5.3.4.3.1 Input

Region id

3.5.3.4.3.2 Process

Region_status update operation and update database

3.5.3.4.3.3 Output

Successful message

3.5.3.4.3.4 Interface with another functional components

Independent

3.5.3.4.3.5 Resource allocation

Region table

3.5.3.4.3.6 User interface

Region details will be displayed on the screen. and set as active/inactive button will be provided for active/inactive operation.

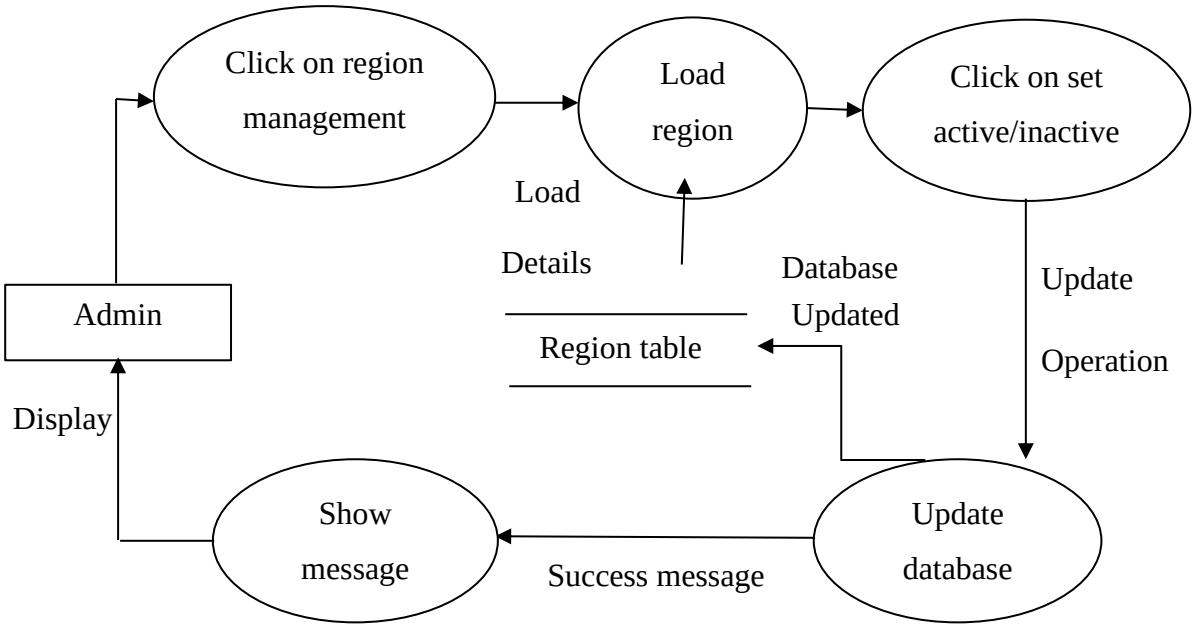


Figure 20

3.4.3.5 View feedback

3.4.3.5.1 Input

Feedback_id

3.4.3.5.2 Process

Feedback of the order will be displayed which is retrieved from feedback table.

3.4.3.5.3 Output

Successful message

3.4.3.5.4 Interface with another functional components

Independent

3.4.2.4.5 Resource allocation

Feedback table

3.4.2.4.6 User interface

Feedback from the user will be display for respective orders

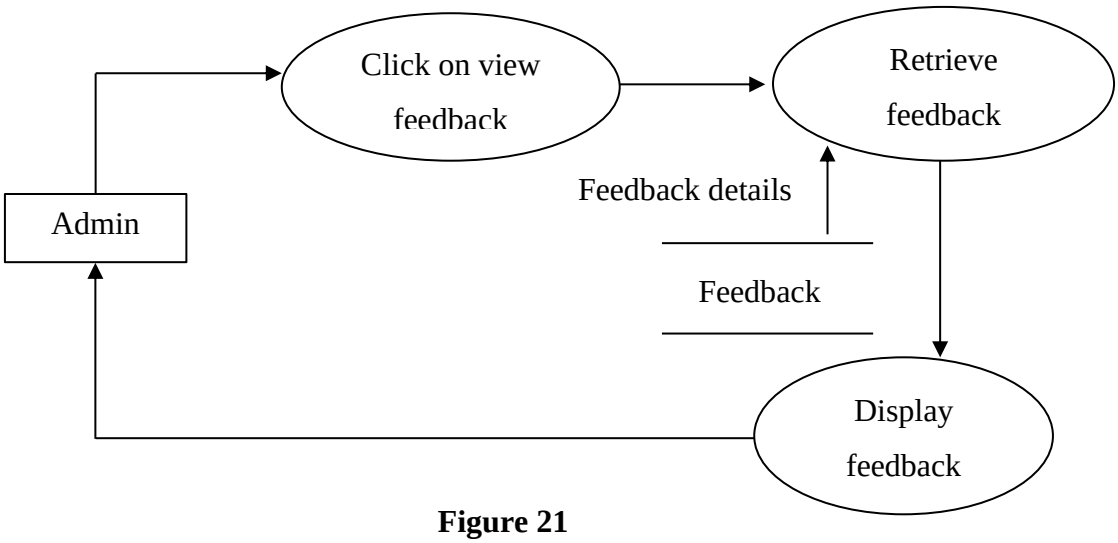


Figure 21

3.4.4 Dealer

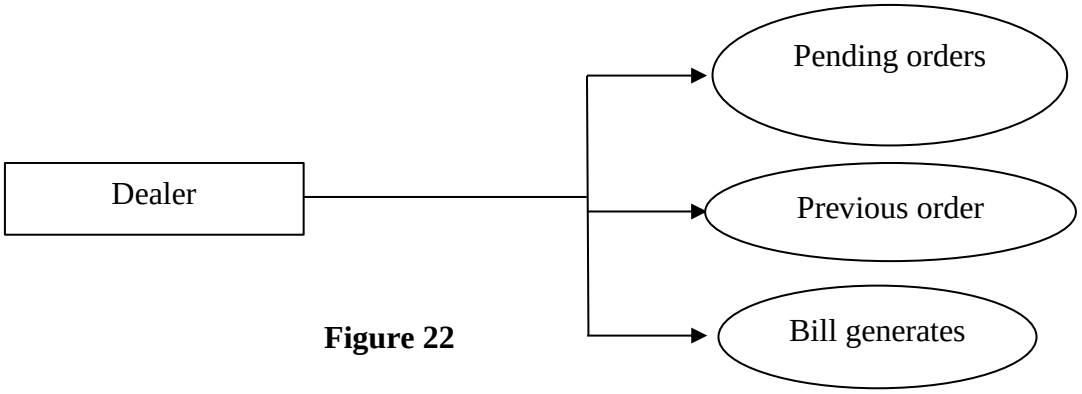


Figure 22

3.5.4.1 Pending orders

3.5.4.1.1 Input
Order id

3.5.4.1.2 Process
Retrieve pending order details

3.5.4.1.3 Output
Pending orders will be displayed

3.5.4.1.4 Interface with another functional components
Independent

3.5.4.1.5 Resource allocation
Item order table

3.5.4.1.6 User interface
Shows all pending order.

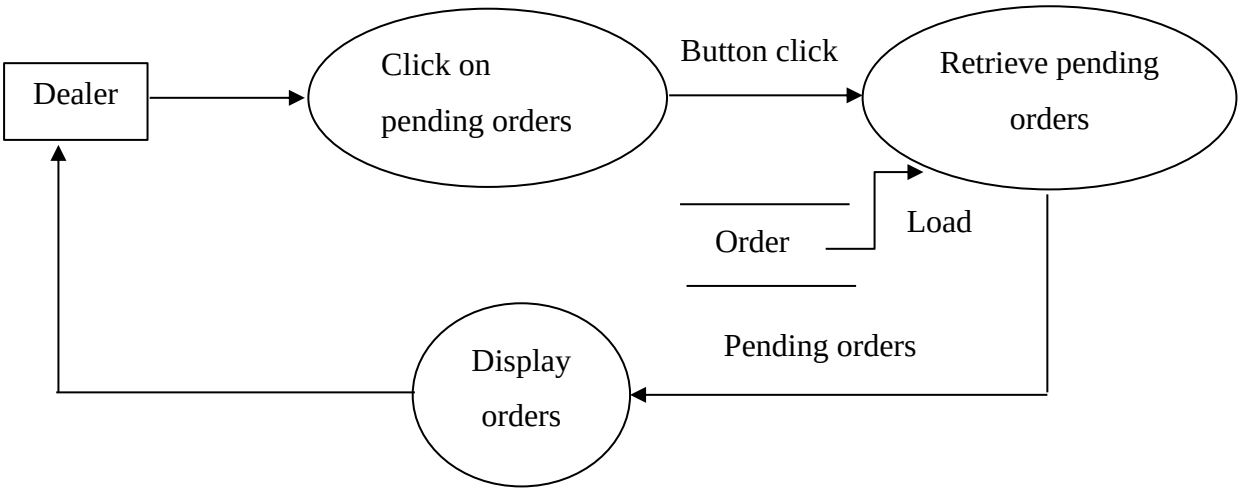


Figure 23

3.5.4.2 Previous orders

3.5.4.2.1 **Input**
click

3.5.4.2.2 **Process**
Retrieving order details

3.5.4.2.3 **Output**
Orders will be displayed

3.5.4.2.4 **Interface with another functional components**
Independent

3.5.4.2.5 **Resource allocation**
Item order table

3.5.4.2.6 **User interface**
Shows all proceeded order.

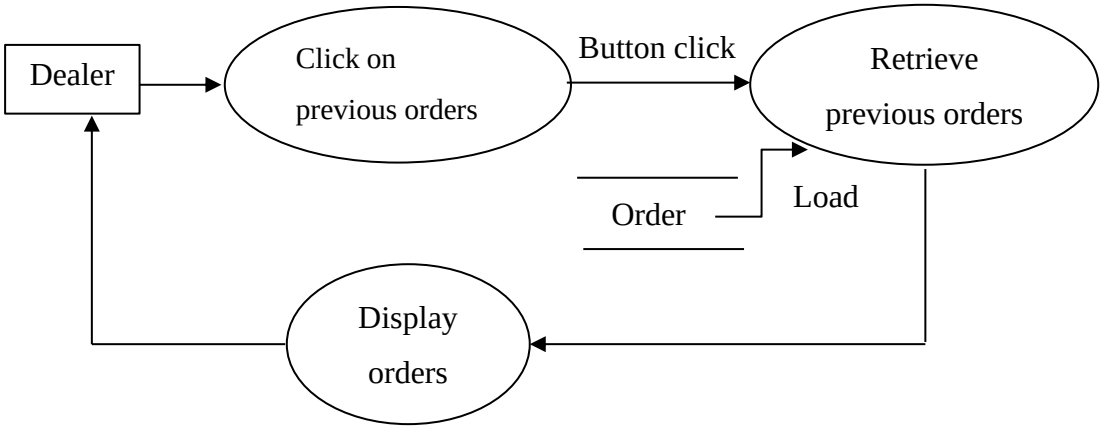


Figure 24

3.5.4.3 Bill generation

3.5.4.3.1 Input

Order details

3.5.4.3.2 Process

Bill generation

3.5.4.3.3 Output

Bill will be displayed

3.5.4.3.4 Interface with another functional components

Independent

3.5.4.3.5 Resource allocation

Cart table, Item order table, Bill table.

3.5.4.3.6 User interface

After clicking on generate bill button the bill will be generated and displayed on the screen.

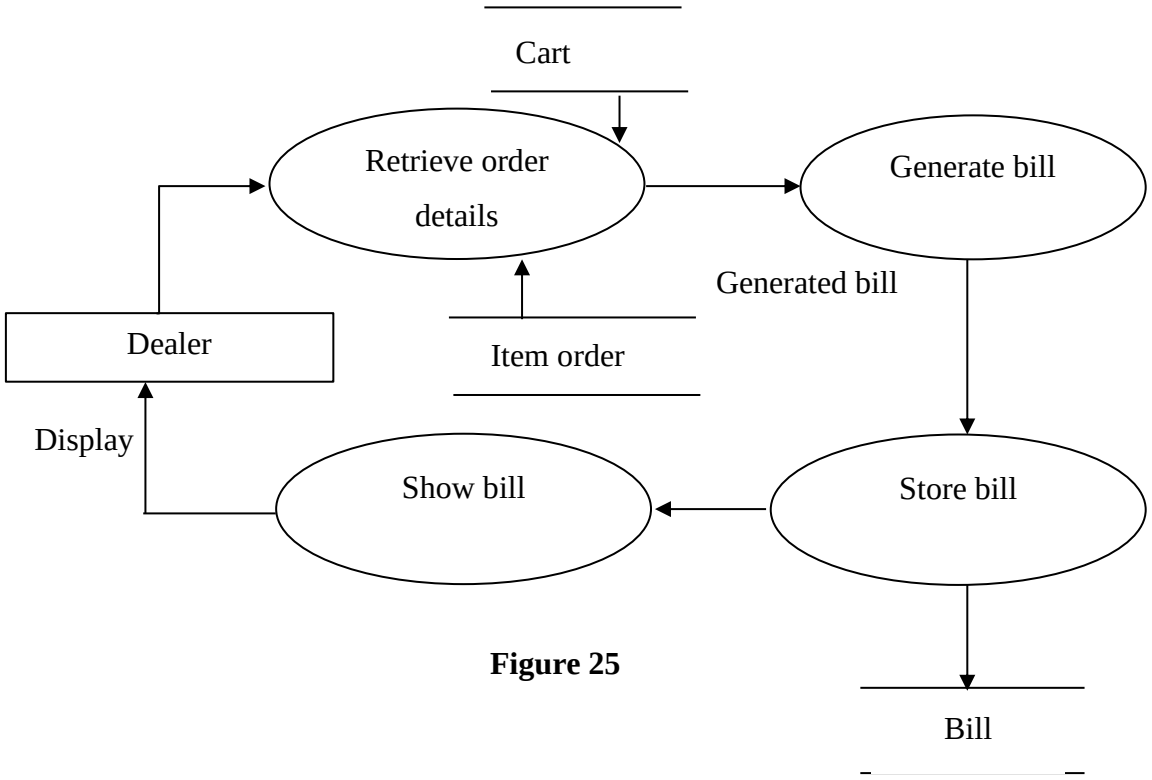


Figure 25

3.5.5 Customer

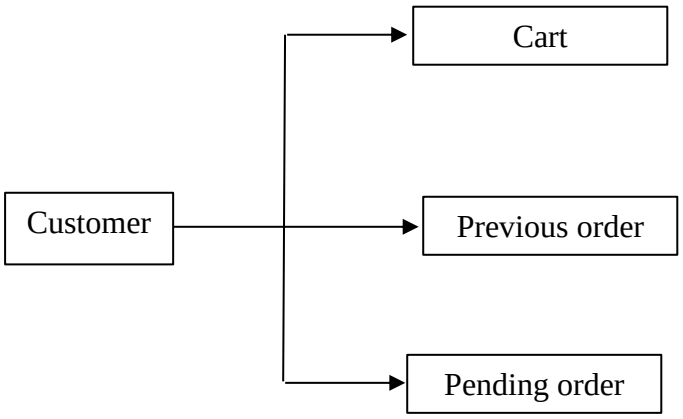


Figure 26

3.5.5.1 Cart

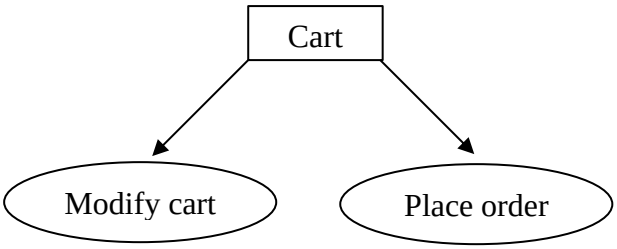


Figure 27

3.5.5.1.1 Modify cart

3.5.5.1.1.1 Input
Button clicks

3.5.5.1.1.2 Process
Store details to cart

3.5.5.1.1.3 Output
Successful message

3.5.5.1.1.4 Interface with another functional component
Independent

3.5.5.1.1.5 User interface
It shows all products which are in cart.

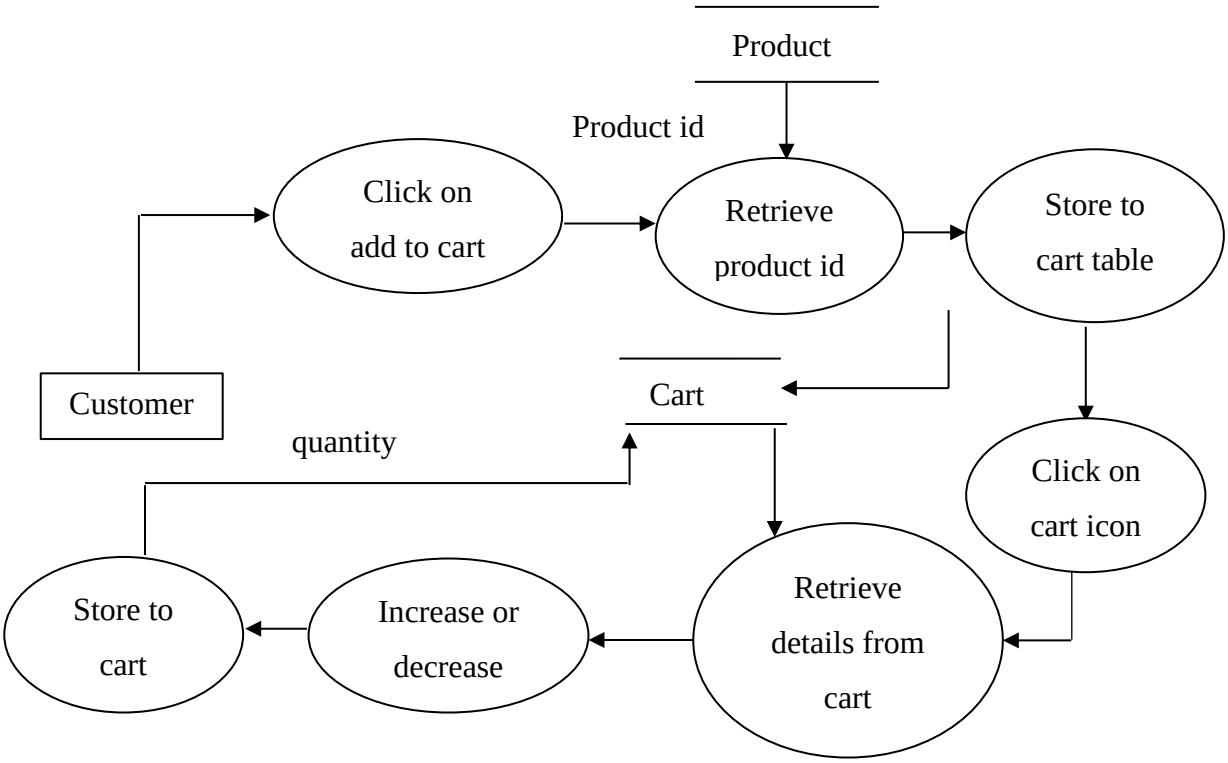


Figure 28

3.5.5.1.2 Place order

3.5.5.1.2.1 Input

Order details

3.5.5.1.2.2 Process

Generate order_id and store it to order table

3.5.5.1.2.3 Output

Successful message

3.5.5.1.2.4 Interface with another functional component

Independent

3.5.5.1.2.5 User interface

Button is used to place order

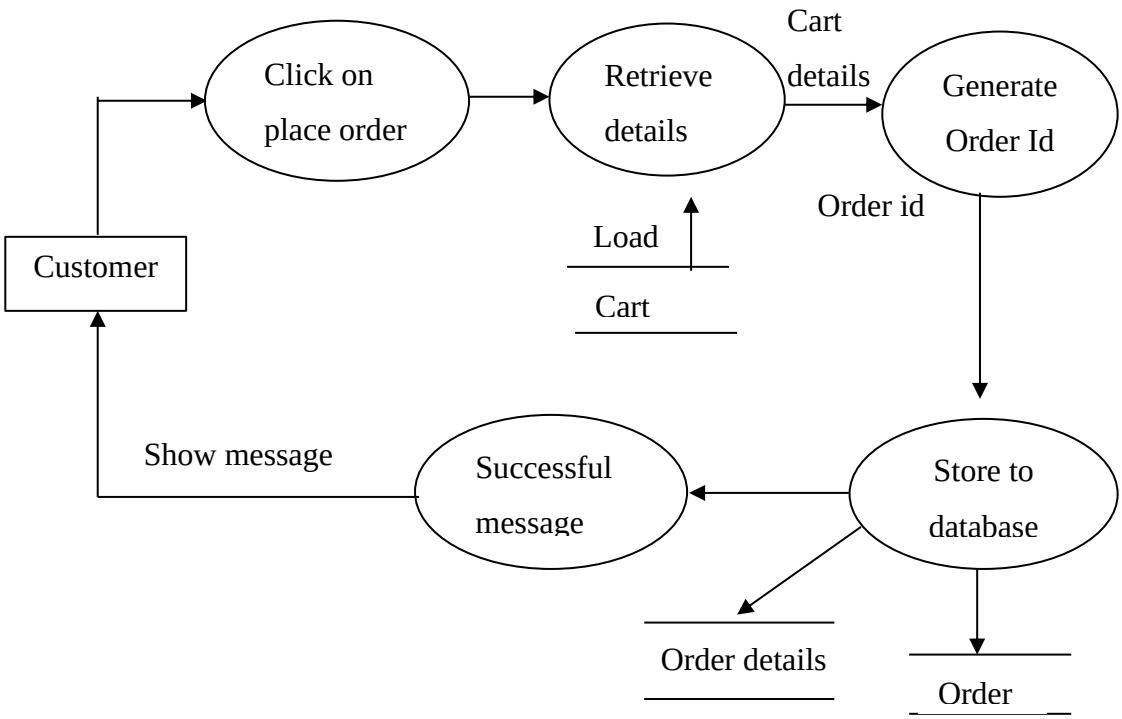


Figure 29

3.5.5.2 Previous order

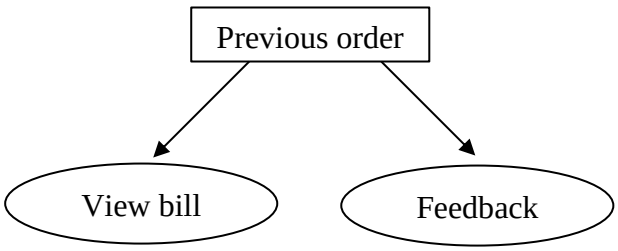


Figure 30

3.5.5.2.1 View bill

3.5.5.2.1.1 Input

Button clicks

3.5.5.2.1.2 Process

Retrieve bill from table

3.5.5.2.1.3 Output

Display bill

3.5.5.2.1.4 Interface with another functional component

Independent

3.5.5.2.1.5 User Interface

View button is given to view bill for previous order.

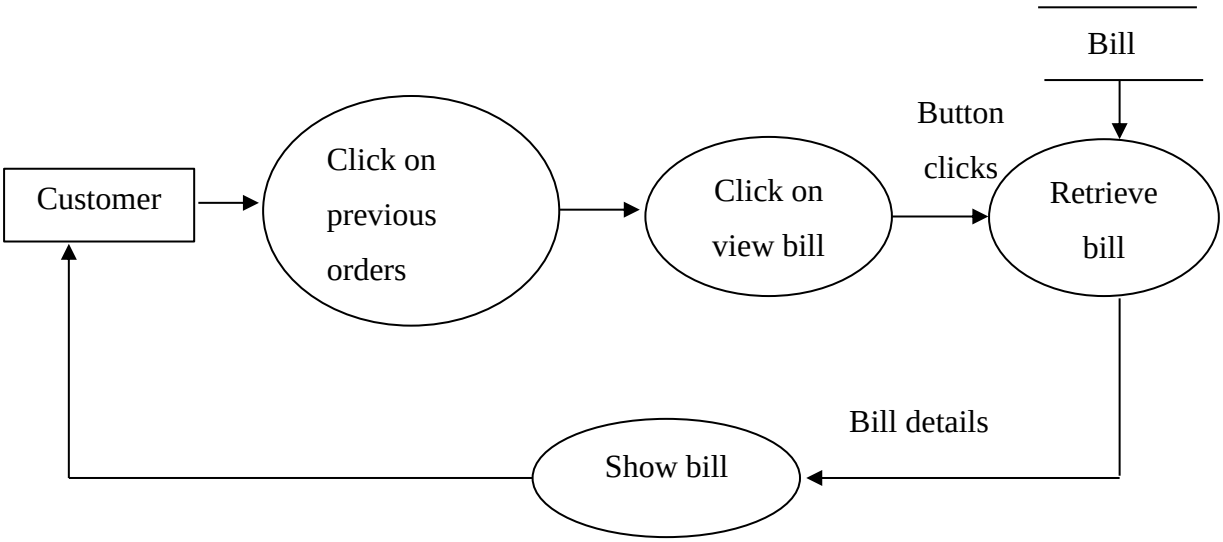


Figure 31

3.5.5.2.2 Feedback

3.5.5.2.2.1 Input

Entered feedback

3.5.5.2.2.2 Process

Store to feedback table

3.5.5.2.2.3 Output

Show successful message

3.5.5.2.2.4 Interface with another functional component

Independent

3.5.5.2.2.5 User Interface

Feedback button is given to write feedback.

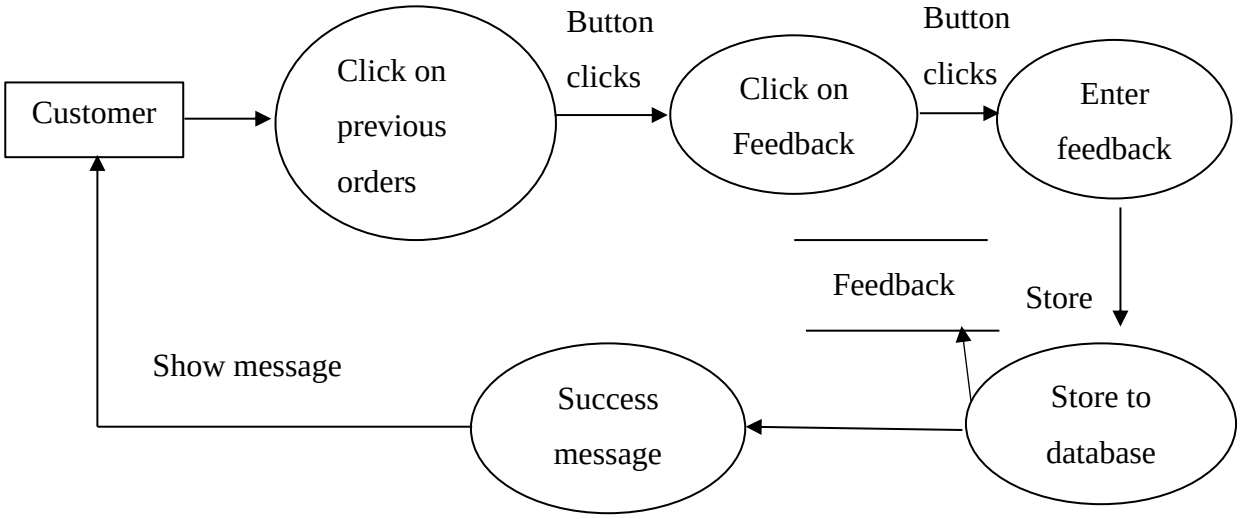


Figure 32

3.5.5.3 Pending order

3.5.5.3.1 Input

Button clicks

3.5.5.3.2 Process

Retrieve previous orders

3.5.5.3.3 Output

Display active orders

3.5.5.3.4 Interface with another functional component

Independent

3.5.5.3.5 User Interface

Button is given to view pending order.

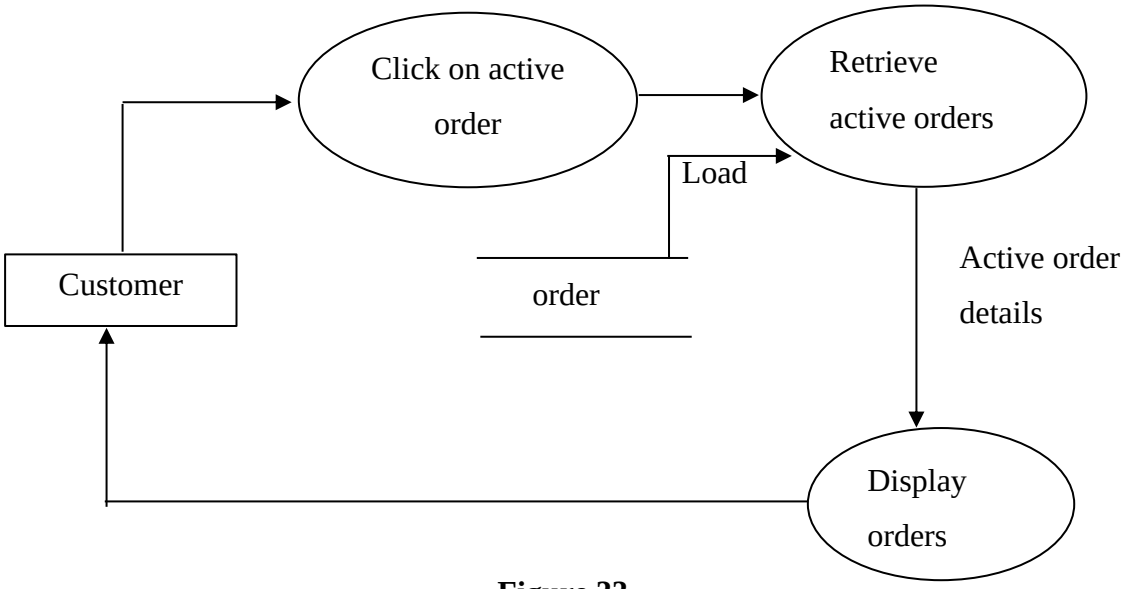


Figure 33

4. Database Design

4.1 Introduction

Database is a collection of related data. Relational database stores data in a table or relations. The data stored in a relation are arranged in records. Each record consists of set of attributes. Fields can be referred to characteristics of records. This document describes the table that is used to design software, its attributes, data types, constraints and relationship among those tables.

The design process consists of the following steps:

- Determine the purpose of your database. Find and organize the information required.
- Divide the information into tables. Turn information items into columns...
- Specify primary keys.
- Set up the table relationships.
- Refine your design.
- Apply the normalization rules.

4.2 Purpose and Scope

Purpose

- Avoid Redundant Data

  The table in the database should be constructed following standards and with utmost dedication. It should have different fields and minimize redundant data. The table should always have a Primary Key that would be a unique id.

- Faultless Information

  The database should follow the standards and conventions and provide meaningful information useful to the organization. (Constraint)

- Data Integrity

  Integrity assists in guaranteeing that the values are valid and faultless. Data Integrity is set to tables, relationships, etc.

- Modify

  The database developed should be worked upon with the conventions and standards, so that it can be easily modified whenever the need arises.

Scope

- Normalization of Database.
- Imposing Integrity Constraint.
- Establishing the Relation between the tables.
- Accessing the data from multiple tables. (Usage of join and sub query….

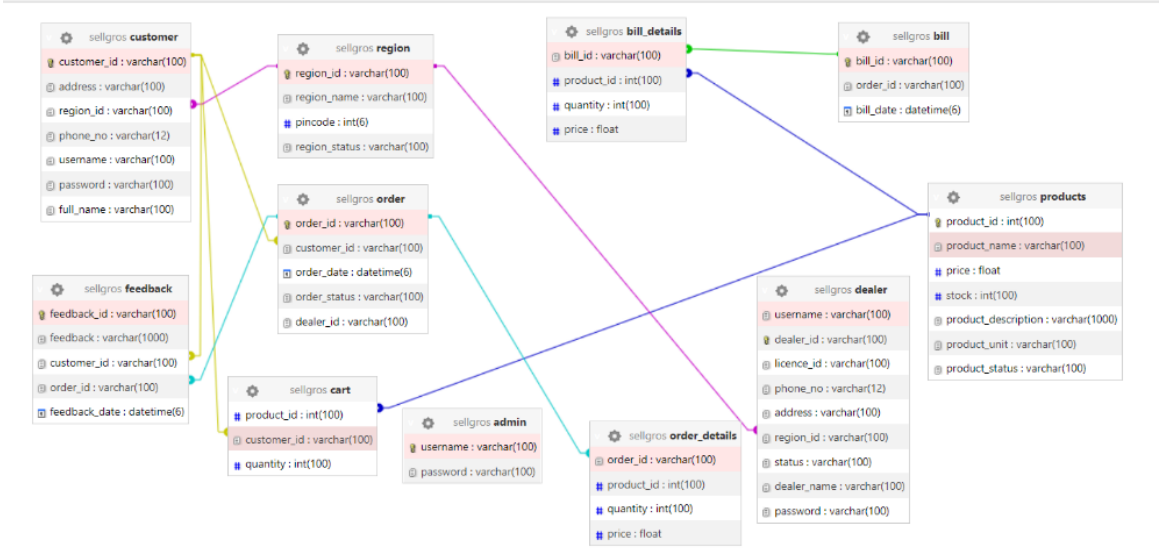
4.3 Database Identification

The identification of database by unique name given to the various database objects. The identifier is the name of database object. The following are the various database objects.

4.4 Schema Information

Database schema its structure described in a formal language supported by the Database Management System (DBMS). The term "schema" refers to the organization of the data as a blueprint of how database is constructed (divide into database tables in the case of relational databases)

In relational database, the schema defines the tables, fields, relationships, views, indexes, packages, procedures, functions, queues, triggers, types, sequences, materialized views.



4.5 Table Definition

4.5.1 Dealer table

Column name	Data type	Size	Constraint	Description
username	varchar	100	Not null	User name of the dealer
dealer_id	varchar	100	primary key	Id of the dealer
licence_id	varchar	100	Not null	License id of dealer
phone_no	varchar	12	Not null	Phone no. of the dealer
address	varchar	100	Not null	Address of the dealer
region_id	varchar	100	foreign key	Region_id of the dealer
status	varchar	100	Not null	Status of the dealer
dealer_name	varchar	100	Not null	Name of the dealer
Password	varchar	100	Not null	Password of the dealer

4.5.2 Customer table

Column name	Data type	Size	Constraint	Description
customer_id	Varchar	100	primary key	Id of the customer
address	Varchar	100	Not null	Address of the customer
region_id	Varchar	100	foreign key	Region_id of the customer
phone_no	Varchar	12	Not null	Phone no. of the customer
username	Varchar	100	Not null	Username of the customer
password	Varchar	100	Not null	Password of the customer
full_name	Varchar	100	Not null	Fullname of the customer

4.5.3 Admin table

Column name	Data type	Size	Constraint	Description
username	Varchar	100	Not null	Username of the admin
password	Varchar	100	Not null	Password of the admin

4.5.4 Region table

Column name	Data type	Size	Constraint	Description
region_id	varchar	100	primary key	Id of the region
region_name	Varchar	100	Not null	Name of the region
pincode	Int	6	Not null	Pin code for the region
Region_status	varchar	100	Not null	Status of the Region

4.5.5 product table

Column name	Data type	Size	Constraint	Description
product_id	Int	100	primary key	Id of the product
product_name	Varchar	100	Not null	Name of the product
Price	Float		Not null	Price of the product
Stock	Int	100	Not null	Stock of the project
product_description	Varchar	100	Not null	Description of the product
product_unit	Varchar	100	Not null	Unit of the product
Product _status	varchar	100	Not null	Status of the product

4.5.6 Order table

Column name	Data type	Size	Constraint	Description
order_id	Varchar	100	primary key	Id of the order
customer_id	Varchar	100	foreign key	Id of the customer
order_date	Datetime	6	Not null	Date of the order
order_status	Varchar	100	Not null	Status of the order
dealer_id	Varchar	100	null	Id of the dealer

4.5.7 Order_details table

Column name	Data type	Size	Constraint	Description
order_id	Varchar	100	foreign key	Id of the order
product_id	Int	100	foreign key	Id of the products
quantity	Int	100	Not null	Quantity of the product
price	float		Not null	Price of the product

4.5.8 Cart table

Column name	Data type	Size	Constraint	Description
product_id	Int	100	foreign key	Id of the products
customer_id	Varchar	100	foreign key	Id of the customer
quantity	Int	100	Not null	Quantity of the product

4.5.9 Bill table

Column name	Data type	Size	Constraint	Description
Bill_id	Varchar	100	primary key	Id of the bill
Order_id	Varchar	100	Not null	Id of the order
Bill_date	Datetime	6	Not null	Date of the bill

4.5.10 Bill_details table

Column name	Data type	Size	Constraint	Description
Bill_id	Varchar	100	foreign key	Id of the bill
Product_id	Varchar	100	foreign key	Id of the products
quantity	Int	100	Not null	Quantity of the products
price	Float		Not null	Price of the products

4.5.11 Feedback table

Column name	Data type	Size	Constraint	Description
feedback_id	Varchar	100	primary key	Id of the feedback
feedback	Varchar	1000	Not null	Feedback about order and dealer
customer_id	Varchar	100	foreign key	Id of the customer
order_id	Varchar	100	foreign key	Id of the order
feedback_date	Datetime	6	Not null	Date of the feedback

4.6 Physical Design

Physical design is where you translate schemas into actual database structures.

- Entity to table
- Tuples to rows
- Attribute to columns
- Primary Key and Alternate Key to Unique Index
- Domain into Constraints

4.7 Data Dictionary

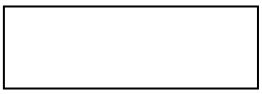
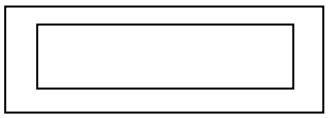
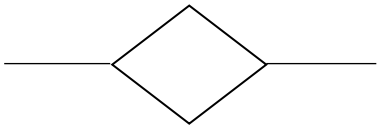
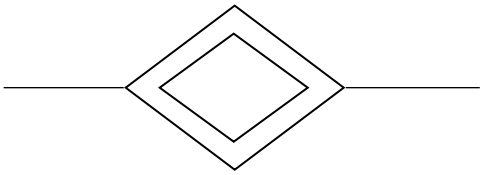
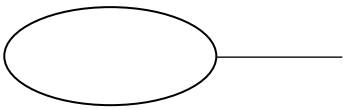
A data dictionary is a file or set of files that include metadata. The data dictionary holds records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

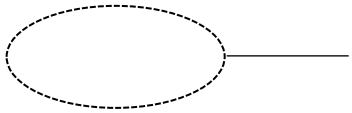
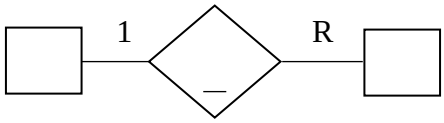
The data dictionary, in general, includes information about the following:

- Name of the data item
- Aliases
- Description/purpose
- Related data items
- Range of values
- Data structure definition

4.8 ER Diagram

ER-modeling is a data modeling method used in software engineering to produce a conceptual data model of an information system. Diagram created using ER-modeling method are called Entry-Relationship diagram or ER-diagram or ERDs.

Symbol	Conversion
	Entity
	Weak entity
	Relationship
	Identity relation
	Attribute

	Derived attribute
	Cardinality ratio 1: N to E1:E2 to R

Components of an ER-diagram

1. **Entity**

   An entity can be a real word object, either animate or inanimate, that can be merely identifiable.

   An entity is denoted as a rectangle in an ER diagram. For example, in a school database, students, teachers, classes, and courses offered can be treated as entities. All these entities have some attributes or properties that give them their identity.

   Entity set

   An entity set is a collection of related types of entities.

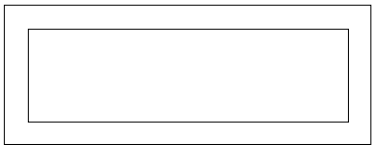
   Strong entity

   An entity with uniquely identified by its attribute.

   Weak Entity

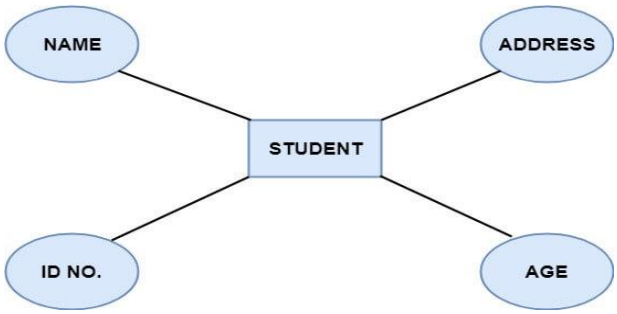
   In a relational database, a week entity is an entity that cannot be uniquely identified by its attributes alone.

2. **Attributes**

   Entities are denoted utilizing their properties, known as attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.

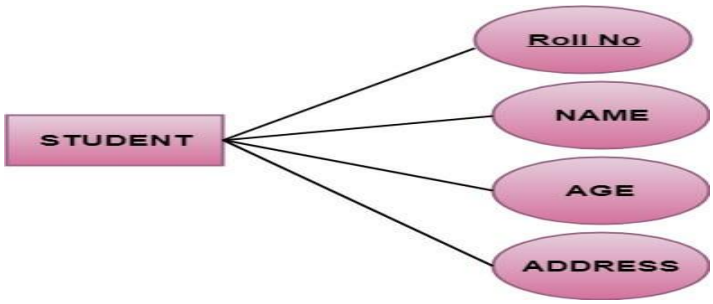
   There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.



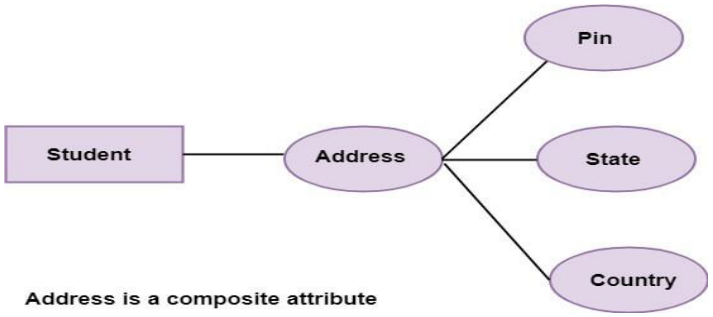
There are four types of Attributes:

1. **Key attribute**
2. **Composite attribute**
3. **Single-valued attribute**
4. **Multi-valued attribute**
5. **Derived attribute**

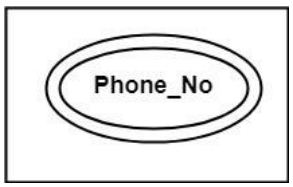
1. **Key attribute**: Key is an attribute or collection of attributes that uniquely identifies an entity among the entity set. For example, the roll number of a student makes him identifiable among students.



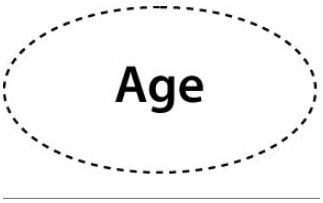
2. **Composite attribute:** An attribute that is a combination of other attributes is called a composite attribute. For example, in student entity, the student address is a composite attribute as an address is composed of other characteristics such as pin code, state, country.

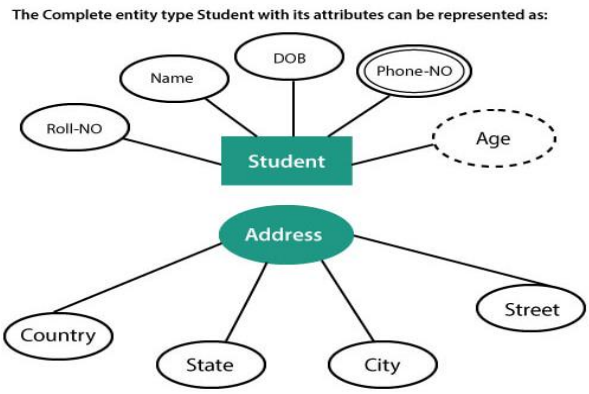


3. **Single-valued attribute**: Single-valued attribute contain a single value. For example, Social_Security_Number.
4. **Multi-valued Attribute**: If an attribute can have more than one value, it is known as a multi-valued attribute. Multi-valued attributes are depicted by the double ellipse. For example, a person can have more than one phone number, email-address, etc.



5. **Derived attribute**: Derived attributes are the attribute that does not exist in the physical database, but their values are derived from other attributes present in the database. For example, age can be derived from date_of_birth. In the ER diagram, Derived attributes are depicted by the dashed ellipse.





3. Relationships

The association among entities is known as relationship. Relationships are represented by the diamond-shaped box. For example, an employee works at a department, a student enrolls in a course. Here, Works at and enrolls are called relationships.

Degree of a relationship set

The number of participating entities in a relationship describes the degree of the relationship. The three most common relationships in E-R models are:

1. Unary (degree1)
2. Binary (degree2)
3. Ternary (degree3)

1. **Unary relationship**: This is also called recursive relationships. It is a relationship between the instances of one entity type. For example, one person is married to only one person.

Fig: Unary Relationship

2. **Binary relationship**: It is a relationship between the instances of two entity types. For example, the Teacher teaches the subject.

Fig: Binary Relationship

3. **Ternary relationship**: It is a relationship amongst instances of three entity types. In fig, the relationships "may have" provide the association of three entities, i.e., TEACHER, STUDENT, and SUBJECT. All three entities are many-to-many participants. There may be one or many participants in a ternary relationship.

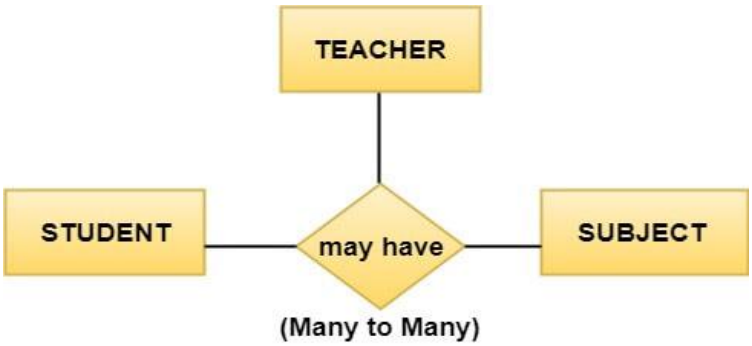


Fig: Ternary Relationship

4. Cardinality Ratio

Cardinality describes the number of entities in one entity set, which can be associated with the number of entities of other sets via relationship set.

Types of Cardinalities

1. **One to One**: One entity from entity set A can be contained with at most one entity of entity set B and vice versa. Let us assume that each student has only one student ID, and each student ID is assigned to only one person. So, the relationship will be one to one.

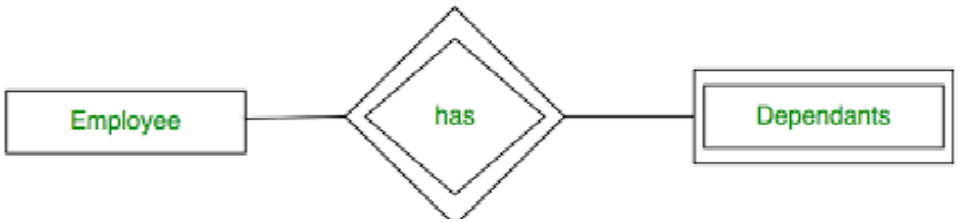
2. **One to many**: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationships. For example, a client can place many orders; a order cannot be placed by many customers.

3. **Many to One**: More than one entity from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A. For example - many students can study in a single college, but a student cannot study in many colleges at the same time.

4. **Many to Many:** One entity from A can be associated with more than one entity from B and vice-versa. For example, the student can be assigned to many projects, and a project can be assigned to many students.

5. **Identifying relationship**

An identiffying relationship is a relationship between two entities in which an instance of a child entity is identified through its association with a parent entity, which means the child entity is dependent on the parent entity for its identity and cannot exits without it.



6. **Participation Contraints**

The participation constraint specifies the number of instance of an entity can participate in a relationship set.

Total participation – Each entity is involved in the relatioship. Total participation is represented by double lines.

Partial participation – Not all entities are involved in the relatioship. Partial participation is represented by single lines.

Example :

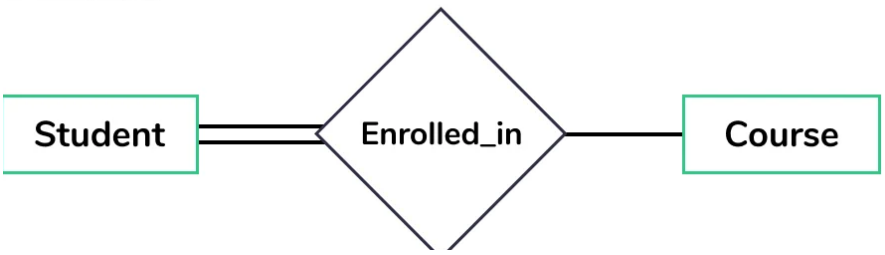
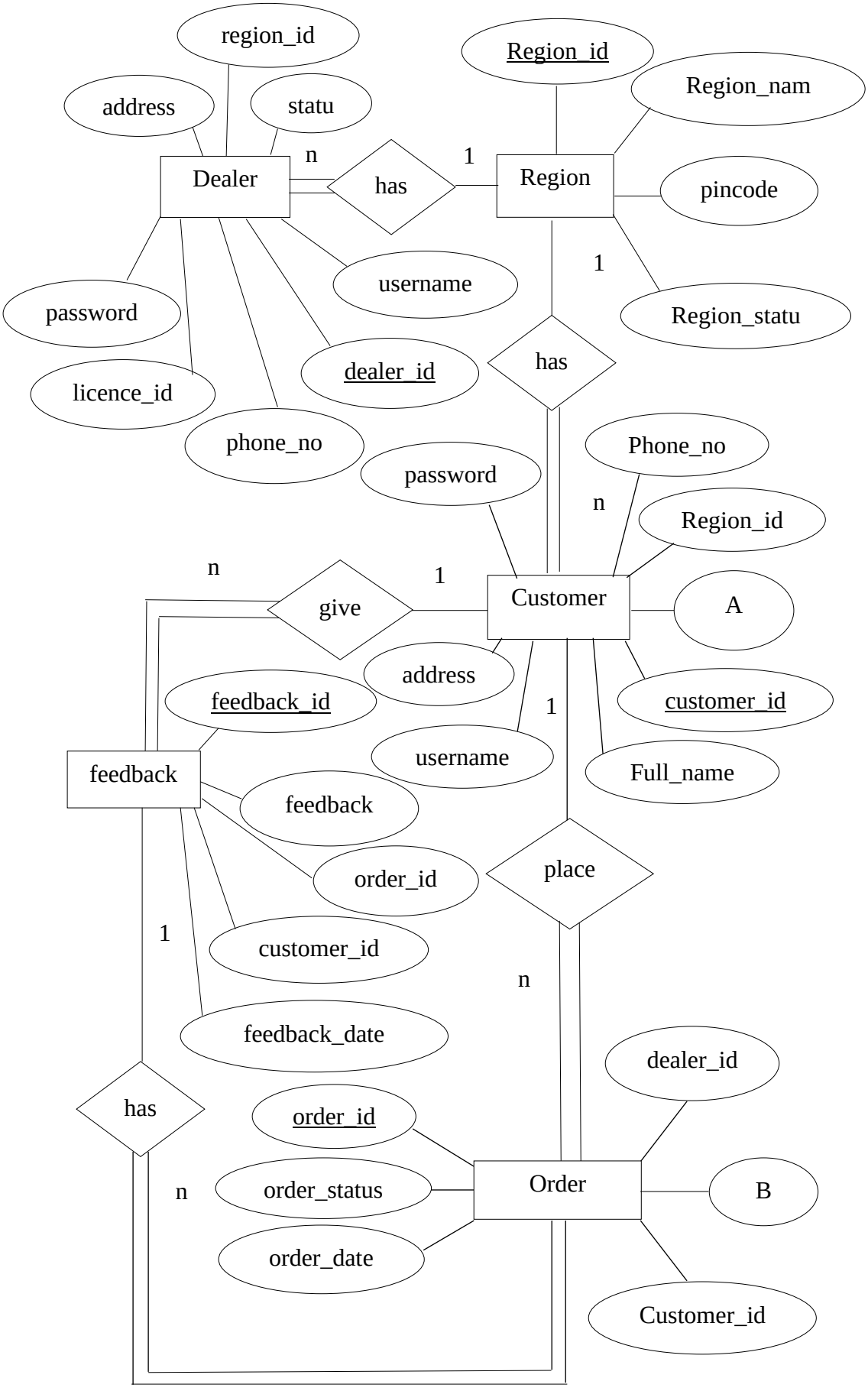
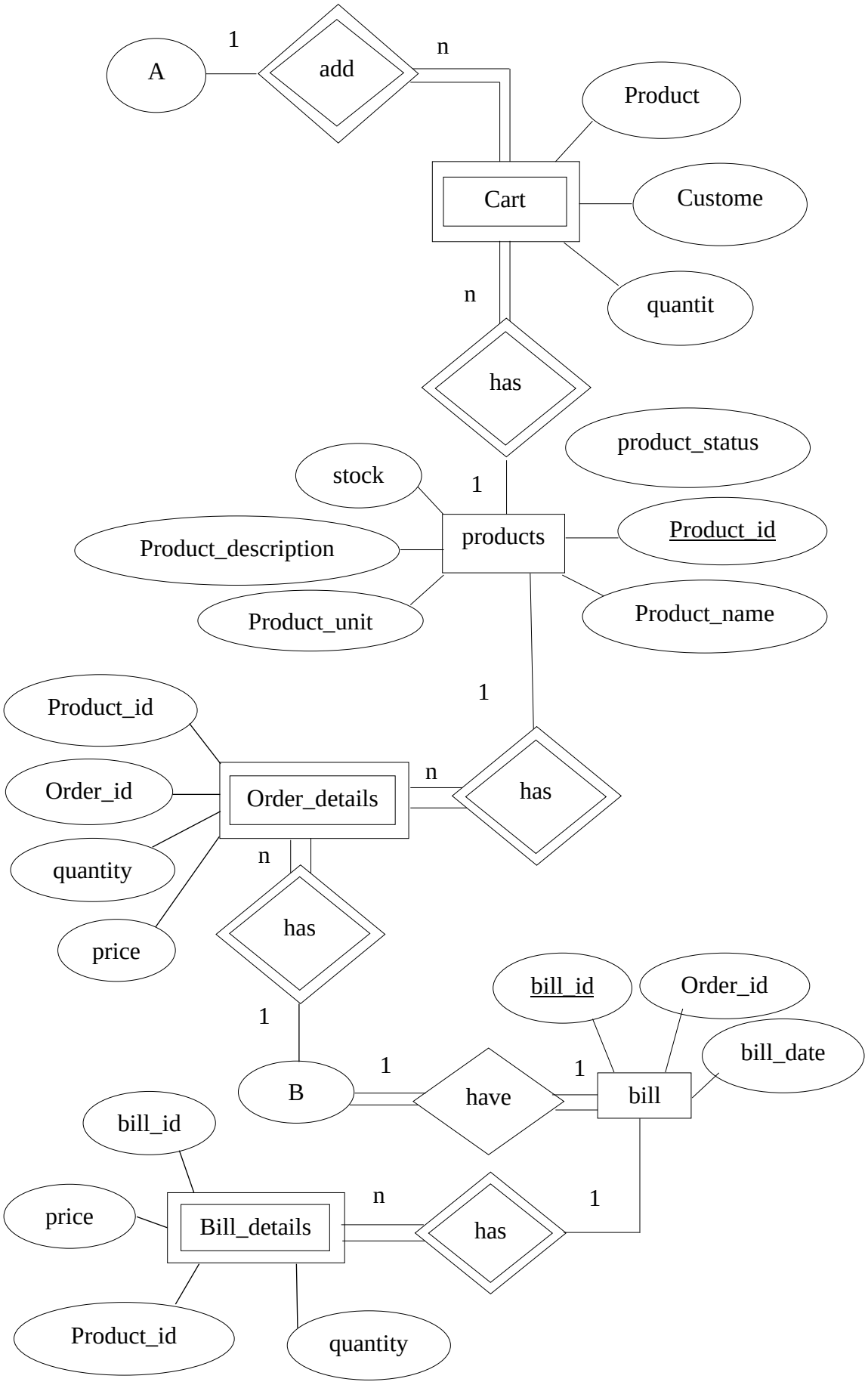


Figure 34


region_id
Region_id
address
statu
Region_nam
n
1
Dealer
has
Region
pincode
username
1
password
Region_statu
dealer_id
has
licence_id
phone_no
password
Phone_no
n
Region_id
n
give
1
Customer
A
address
feedback_id
customer_id
1
username
Full_name
feedback
feedback
order_id
place
1
customer_id
n
feedback_date
dealer_id
has
order_id
n
order_status
Order
B
order_date
Customer_id


A
1
add
n
Product
Cart
Custome
quantit
n
has
product_status
stock
1
products
Product_id
Product_description
Product_unit
Product_name
1
Product_id
Order_id
Order_details
n
has
quantity
n
has
price
bill_id
Order_id
bill_date
1
B
1
have
1
bill
bill_id
price
Bill_details
n
has
1
Product_id
quantity

4.9 Database Administration

4.9.1 DBMS system Information

A database is an organized collection of structured information, or data typically stored electronically in a computer system. A data base is usually controlled by a database management system (DBMS).

In our project we are using MySQL database.

4.9.2 DBMS Configuration

Steps for configure Apache and MySQL in XAMPP

1. In phpMyAdmin, click the Users tab at the top.
2. Find the row that has User root and Host 127.0.0.1.
3. Click Edit Privileges.
4. Click Change password.
5. Enter the password twice (write it down somewhere if you're sure, you can remember it)
6. Click the Go button

4.9.3 Software Support Required

MySQL Required XAMPP

Software Requirements

The following operating systems are officially supported:

- Windows xp or higher version (64-bit, Professional level or higher)
- Mac OS X 10.6.1 or higher
- Ubuntu 9.10 (64bit) or higher version

4.9.4 Hardware (storage) requirements

- Hard Disk:1 TB Required 500GB(Recommended)
- CPU: Intel Core 3GHz (or Dual Core 2GHz) or equal AMD CPU
- Cores: Dual (Quad Core is recommended)
- RAM: 4 GB (6 GB recommended)
- Graphic Accelerators: NVidia or ATI with support of OpenGL 1.5 or higher
- Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

4.9.5 Backup and Recover

Recovery is the process of restoring a database to the correct state in the event of a failure

Database backup is a way to protect and restore a database. It is performed through database replication and can be done for a database or a database server.

Using phpMyAdmin to Back Up or Restore MySQL

If you're running phpMyAdmin backing up and restoring your MySQL database is simple.

The export function is used as a backup, and the import function is used to restore.

Step 1: Create a MySQL Database Backup

1. Open phpMyAdmin. On the directory tree on the left, click the database you want to back up.

This should open the directory structure in the right-hand window. You'll also notice that, in the directory tree on the left, all the assets under the main database are highlighted.

2. Click Export on the menu across the top of the display.

You'll see a section called "Export Method." Use Quick to save a copy of the whole database. Choose Custom to select individual tables or other special options.

Leave the Format field set to SQL, unless you have a good reason to change it.

3. Click Go. If you select Quick, your web browser will download a copy of the database into your specified downloads folder. You can copy that to a safe location.

Step 2: Clear the Old Database Information

It’s important to clear out old data before restoring a backup. If there’s any old data, it isn’t overwritten when you restore. This can create duplicate tables, causing errors and conflicts.

1. Open phpMyAdmin, on the navigation pane on the left, choose the database you want to restore.

2. Click the check all box near the bottom. Then, use the drop-down menu labeled with selected to select Drop.

3. The tool should prompt you to confirm that you want to go forward. Click yes.

This will get rid of all the existing data, clearing the way for your restoration.

Step 3: Restore Your Backed up MySQL Database

In phpMyAdmin, the Import tool is used to restore a database.

1. On the menu across the top, click Import.

2. The first section is labeled File to import. A couple of lines down, there’s a line that starts with “Browse your computer,” with a button labeled Choose File. Click that button.

3. Use the dialog box to navigate to the location where you’ve saved the export file that you want to restore. Leave all the options set to default. (If you created your backup with different options, you can select those here.)

4. Click Go.

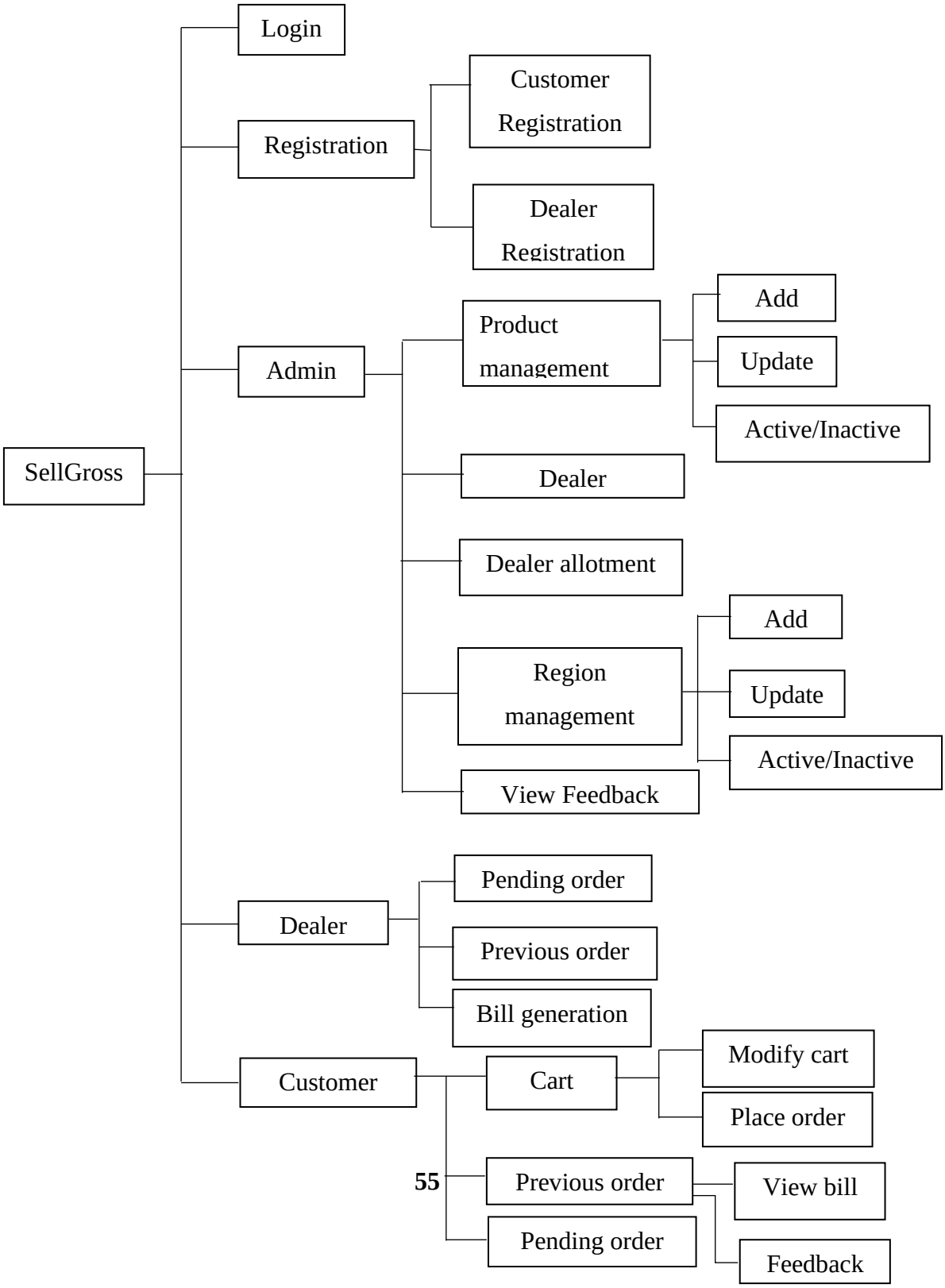
5. Detail Design

5.1 Introduction

During detailed design, the internal logic of each module specified in system design is decided. During this phase further details of the modules are decided. Design of each of the modules usually specified in a high-level description language which is independent of the language in which software eventually be implemented.

5.2 Structure of the system

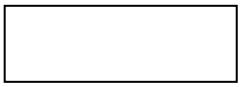
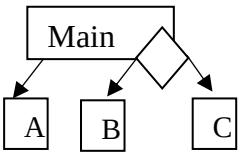
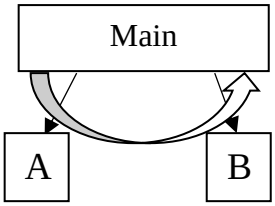
Figure 35



5.3 Module Description

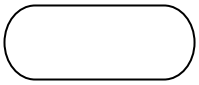
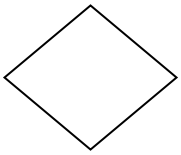
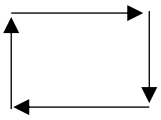
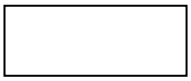
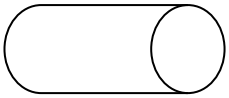
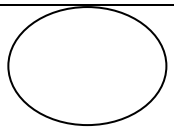
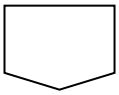
Structure Chart

Structure chart is a top-down modular design, consist of squares representing different models in a systems and lines**.** Structure chart shows how program has been partitioned into manageable modules hierarchy and organization of those modules and communicational interface.

Symbol	Name	Process
	Data Flow	Shows the direction flow of data
	Control Flow	Shows the direction flow of control
	Processing	Shows manipulation, calculation and processing
	Module invocation	It represents subordinate module invoked by super ordinate module
	Conditional invocation	It indicates the invocation of sub-ordinates. Module depends on the evaluation of condition
	Invocation	It represents the repetition

Flow Chart

A flowchart is a graphically representation of the structure of process or system, algorithm or the step-by-step solution of the problem. Flowchart describes the flow of data through an information processing system and the parts of the flows. The flow is a set of the logic operations that meet the certain requirements.

Symbol	Name	Purpose
	Terminator	It indicates the start and end of the process
	Input/Output	Input/Output data
	Decision	It represents the comparison or question that determines an alternate path to be followed
	Flow direction	Shows the direction of data flow
	Processing	It represents manipulation, calculation or information processing
	Direction action storage	File storage
	Preparation (Looping)	An instruction or looping
	In page	Connects different modules in same page
	Off page	Connects different modules in another page
	Delay	The Delay flowchart symbol depicts any waiting period that is part of a process

5.3.1 Login

5.3.1.1 Input

Username and password

5.3.1.2 Procedural Details

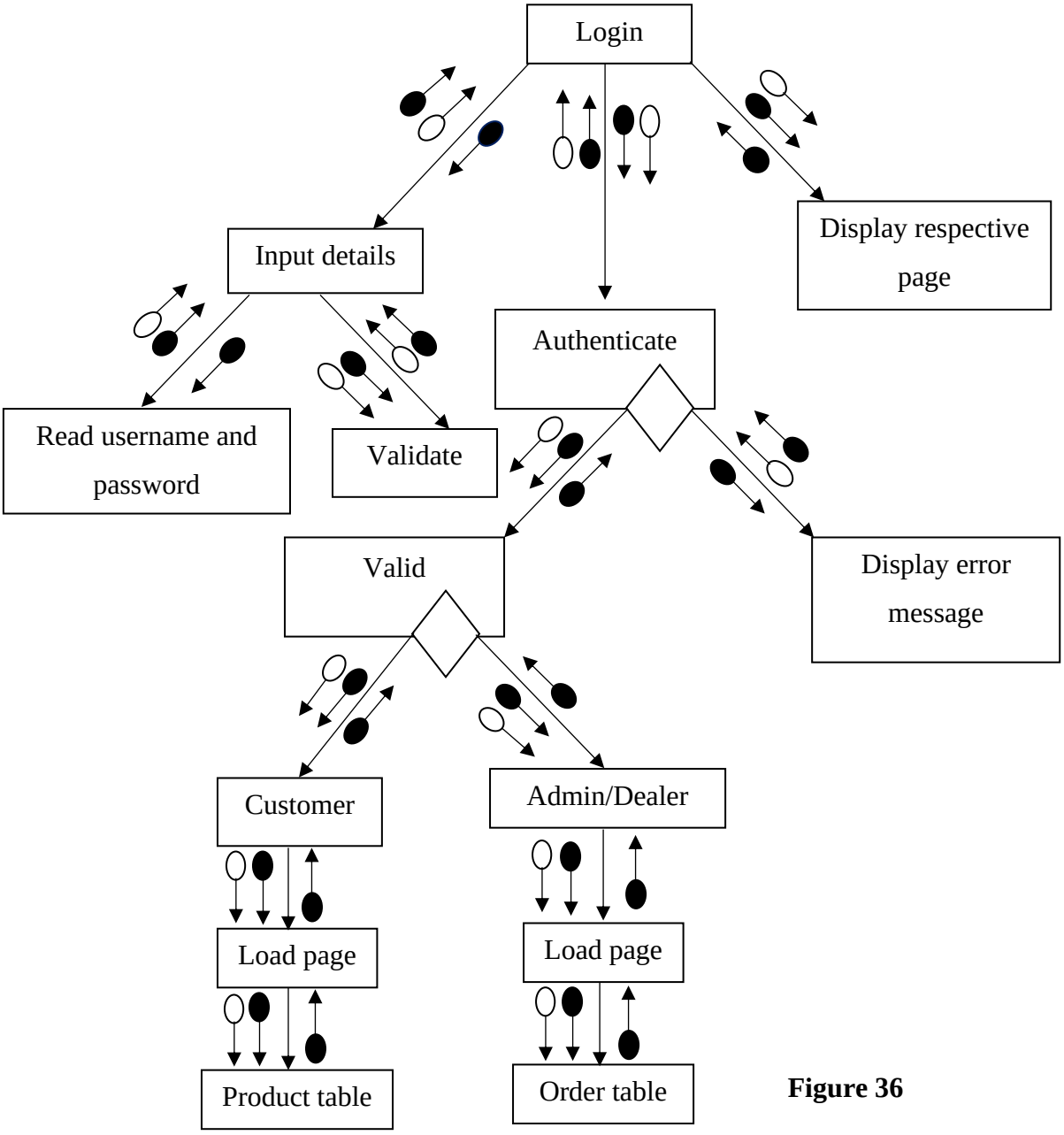


Figure 36

5.3.1.3 File I/O Interfaces

Admin/Dealer/Customer table

5.3.1.4 Output

Login and display respective pages

5.3.1.5 Implementation aspects (if any)

Textboxes, Buttons and Tabs

5.3.2 Registration

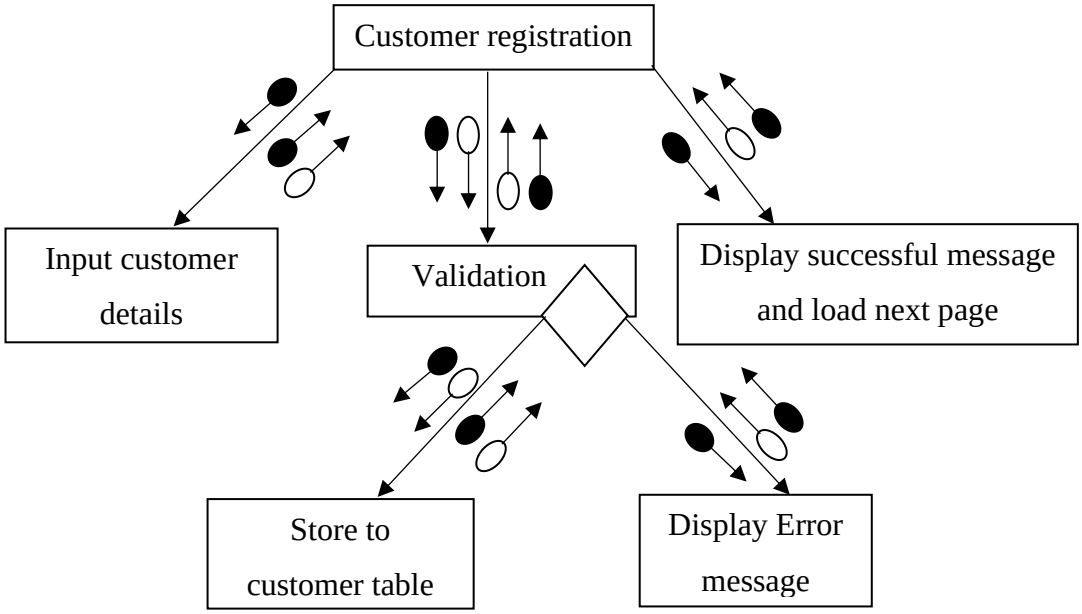
5.3.2.1 Customer

5.3.2.1.1 Input

Customer registration details

5.3.2.1.2 Procedural Details

Figure 37



5.3.2.1.3 File I/O Interfaces

Customer table

5.3.2.1.4 Output

Add details to customer table and show message

5.3.2.1.5 Implementation aspects (if any)

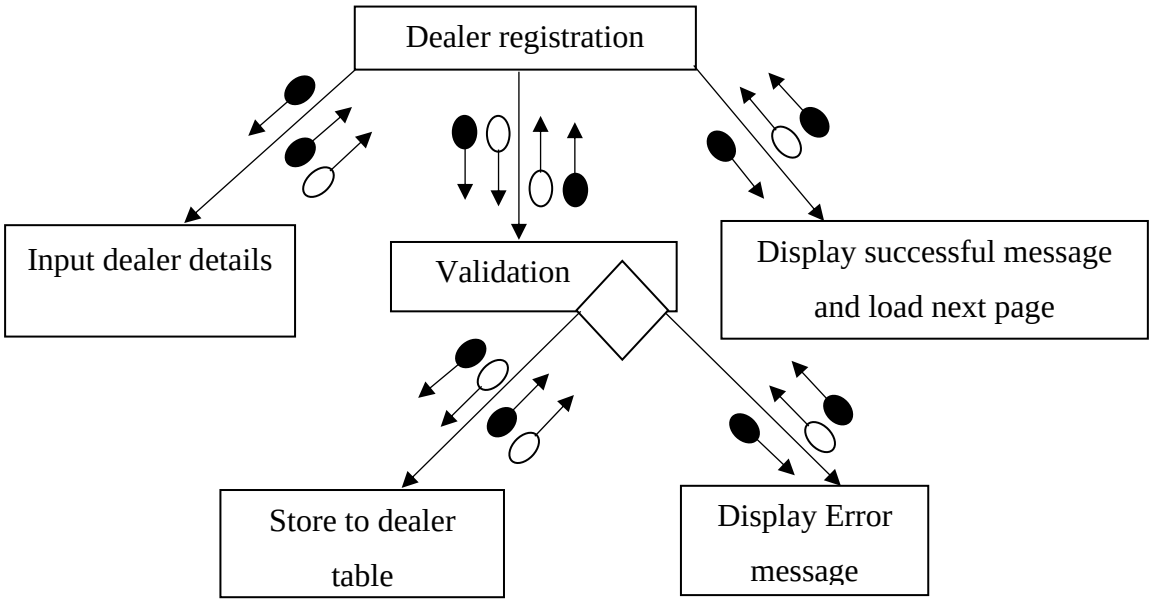
5.3.2.2 Dealer

5.3.2.1.1 Input

Dealer registration details

5.3.2.1.2 Procedural Details

Figure 38



5.3.2.1.3 File I/O Interfaces

Dealer table

5.3.2.1.4 Output

Details will be stored and successful message will be displayed

5.3.2.1.5 Implementation aspects (if any)

Textboxes, Buttons and Dropdown list

5.3.3 Admin

5.3.3.1 Product Management

5.3.3.1.1 Add

5.3.3.1.1.1 Input

Product details

5.3.3.1.1.2 Procedural Details

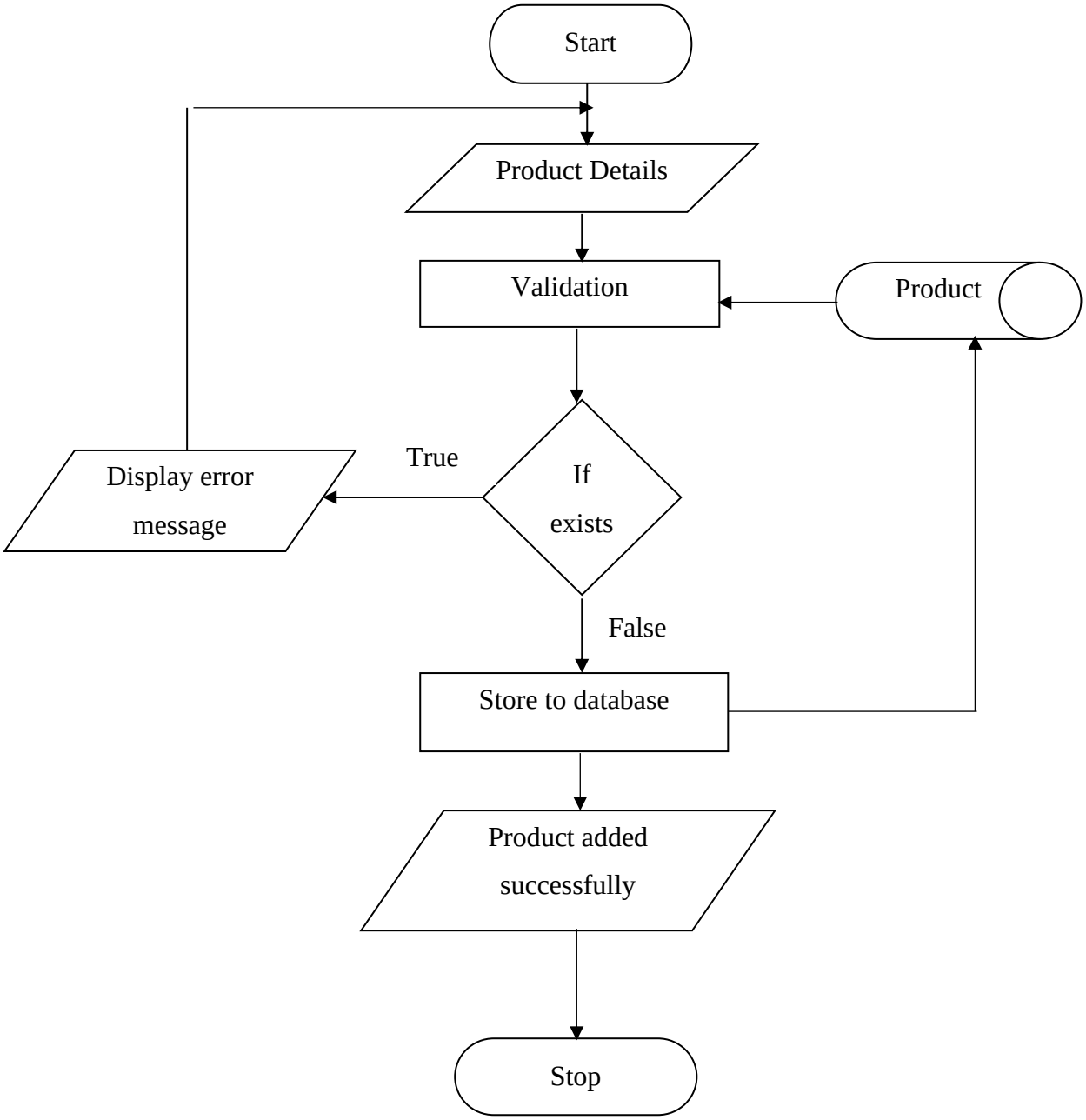


Figure 39

5.3.2.1.1.3 File I/O Interfaces

Product table

5.3.2.1.1.4 Output

Store the product details and show successful message

5.3.2.1.1.5 Implementation aspects (if any)

Textboxes, and Buttons

5.3.3.1.2 Update

5.3.3.1.2.1 Input

Updated product details

5.3.3.1.2.2 Procedural Details

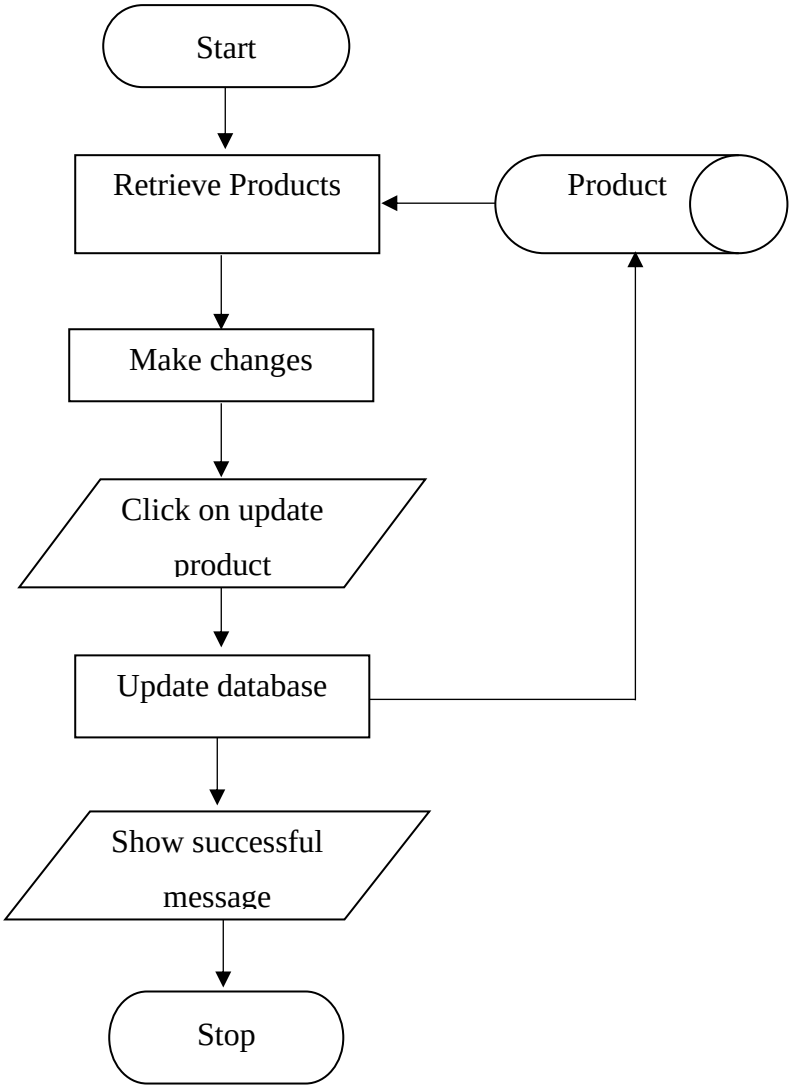


Figure 40

5.3.3.1.2.3 File I/O Interfaces

Product table

5.3.3.1.2.4 Output

Changes will be updated

5.3.3.1.2.5 Implementation aspects (if any)

Textboxes, and Buttons

5.3.3.1.3 Active/Inactive

5.3.3.1.3.1 Input

Active/Inactive text from buttons

5.3.3.1.3.2 Procedural Details

Step:1 Start

Step:2 CLICK on set as active/inactive button

Step3: IF active THEN

UPDATE product table SET status as active

ELSE IF inactive THEN

UPDATE product table SET status as inactive

Step4: DISPLAY message as product status has been changed

Step5: END

5.3.3.1.3.3 File I/O Interfaces

Product table

5.3.3.1.3.4 Output

Successful message

5.3.3.1.3.5 Implementation aspects (if any)

Buttons

5.3.3.2 Dealer Authorization

5.3.3.2.1 Input

Dealer details

5.3.3.2.2 Procedural Details

Step1: Start

Step2: retrieve dealer details where status is pending

Step3: CLICK on accept or reject

Step4: IF accept THEN

SET status as approve

END IF

IF reject THEN

SET status as reject

END IF

Step5: END

5.3.3.2.3 File I/O Interfaces

Dealer table

5.3.3.2.4 Output

Update dealer status and show message

5.3.3.2.5 Implementation aspects (if any)

Buttons

5.3.3.3 Dealer Allotment

5.3.3.3.1 Input

Dealer details and item order details

5.3.3.3.2 Procedural Details

Step1: Start

Step2: CLICK on DEALER ALLOTMENT

Step3: retrieve orders from order table where dealer_id is NULL

Step4: retrieve customer order details from order details table

Step5: retrieve region from dealer table where region_id in dealer table

Step6: SHOW orders and dealer_id

Step7: SELECT dealer_id from dropdown list

Step8: CLICK on allot button

Step9: assign dealer_id to orders and UPDATE dealer_id in order table

Step10: END

5.3.3.3.3 File I/O Interfaces

Dealer table and item order table

5.3.3.3.4 Output

Successful message

5.3.3.3.5 Implementation aspects (if any)

Dropdown list, buttons

5.3.3.4 Region Management

5.3.3.4.1 Add

5.3.3.4.1.1 Input

Region details

5.3.3.4.1.2 Procedural Details

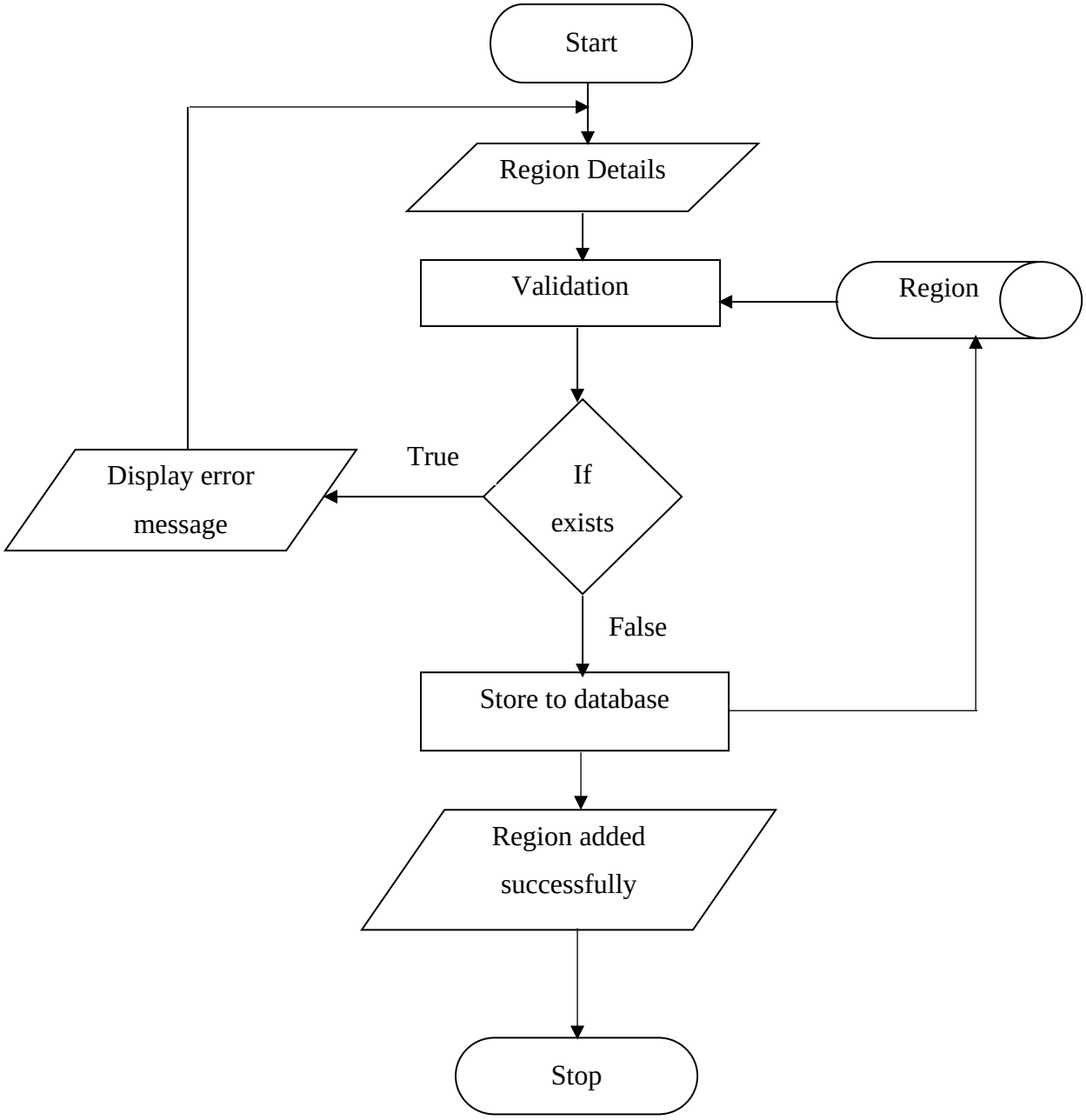


Figure 41

5.3.3.4.1.3 File I/O Interfaces

Region table

5.3.3.4.1.4 Output

Region will be stored and successful message will be displayed

5.3.3.4.1.5 Implementation aspects (if any)

Textboxes and Buttons

5.3.3.4.2 Update

5.3.3.4.2.1 Input

Updated region details

5.3.3.4.2.2 Procedural Details

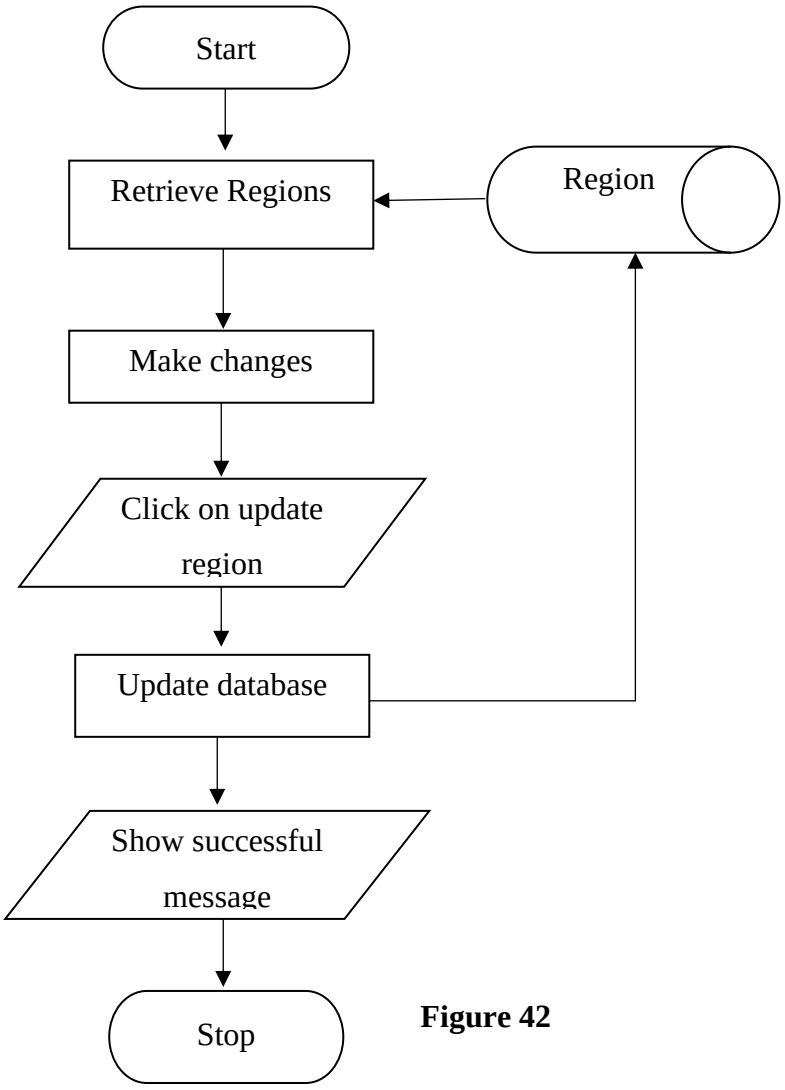


Figure 42

5.3.3.4.2.3 File I/O Interfaces

Region table

5.3.3.4.2.4 Output

Region updated message

5.3.3.4.2.5 Implementation aspects (if any)

Textboxes and Buttons

5.3.3.4.3 Active/Inactive

5.3.3.4.3.1 Input

Region id

5.3.3.4.3.2 Procedural Details

Step1: Start

Step2: CLICK on active/inactive button

Step3: IF active THEN

UPDATE region table SET status as active

ELSE IF inactive THEN

UPDATE region table SET status as inactive

Step4: DISPLAY message as region status has been changed

Step5: END

5.3.3.4.3.3 File I/O Interfaces

Region table

5.3.3.4.3.4 Output

Successful message

5.3.3.4.3.5 Implementation aspects (if any)

Buttons

5.3.3.5 View feedback

5.3.3.5.1 Input

Button clicks

5.3.3.5.2 Procedural Details

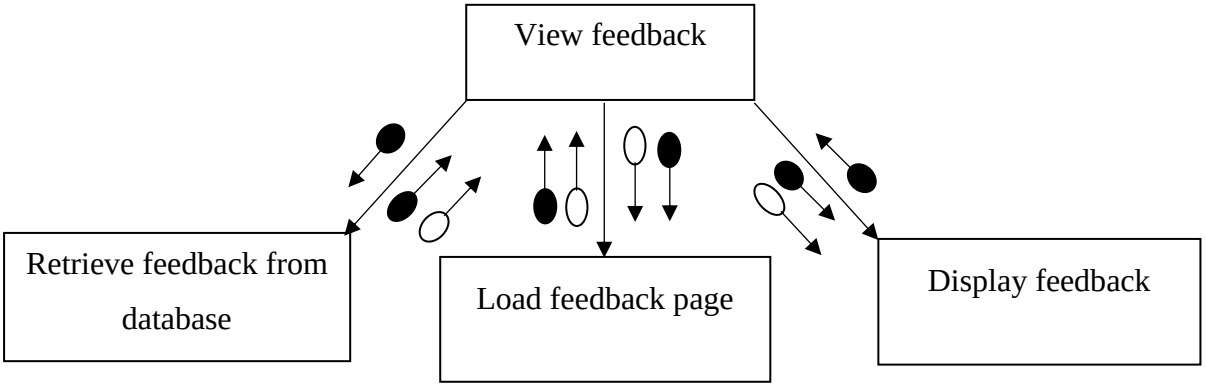


Figure 43

5.3.3.5.3 File I/O Interfaces

Feedback table

5.3.3.5.4 Output

Feedback will be displayed

5.3.3.5.5 Implementation aspects (if any)

Buttons

5.3.4 Dealer

5.3.4.1 Pending orders

5.3.4.1.1 Input

Button clicks

5.3.4.1.2 Procedural Details

Step1: start

Step2: CLICK on pending order

Step3: retrieve orders where dealer_id in order table where order_status is pending.

Step4: DISPLAY orders with generate bill button

5.3.4.1.3 File I/O Interfaces

Order table

5.3.4.1.4 Output

Pending orders will be displayed

5.3.4.1.5 Implementation aspects (if any)

Buttons

5.3.4.2 Previous orders

5.3.4.2.1 Input

Button clicks

5.3.4.2.2 Procedural Details

Step1: Start

Step2: CLICK on previous order

Step3: retrieve order where order_status is processed in order table

Step4: retrieve order_id where order_id in order table

Step5: DISPLAY orders.

Step6: END

5.3.4.2.3 File I/O Interfaces

Order table

5.3.4.2.4 Output

Previous order will be displayed

5.3.4.2.5 Implementation aspects (if any)

Buttons

5.3.4.3 Bill Generation

5.3.4.3.1 Input

Order details

5.3.4.3.2 Procedural Details

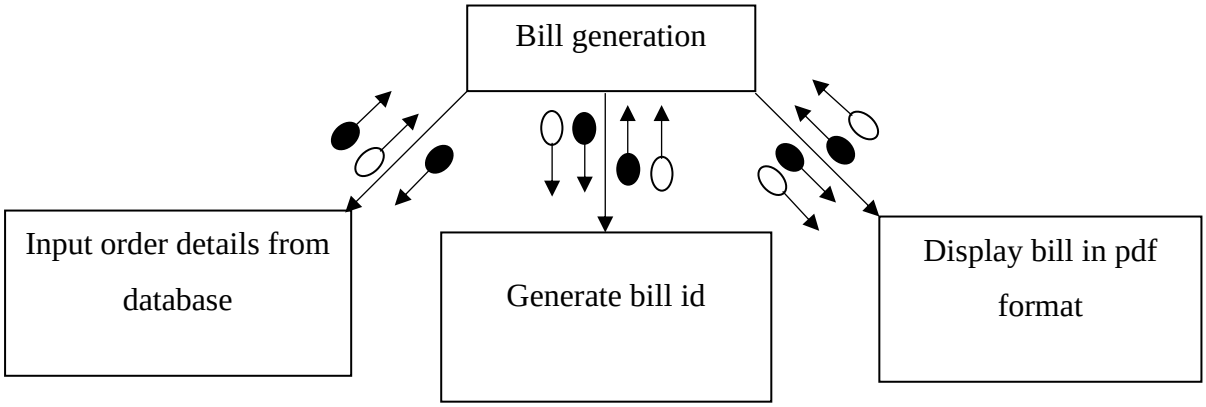


Figure 44

5.3.4.3.3 File I/O Interfaces

Cart table, Order table and Bill table

5.3.4.3.4 Output

Display bill

5.3.4.3.5 Implementation aspects (if any)

Buttons

5.3.5 Customer

5.3.5.1 Cart

5.3.5.1.1 Modify cart

5.3.5.1.1.1 Input

Items in the cart

5.3.5.1.1.2 Procedural Details

Step1: start

Step2: CLICK on add to cart button

Step3: retrieve product_id from product table

Step4: store to cart table

Step5: CLICK on cart

Step6: retrieve details from cart table

Step7: SET quantity by increasing or\ decreasing

Step8: store quantity to cart table

5.3.5.1.1.3 File I/O Interfaces

Cart table

5.3.5.1.1.4 Output

Items will be added to cart with quantity

5.3.5.1.1.5 Implementation aspects (if any)

Buttons

5.3.5.1.2 Place order

5.3.5.1.2.1 Input

Details from cart

5.3.5.1.2.2 Procedural Details

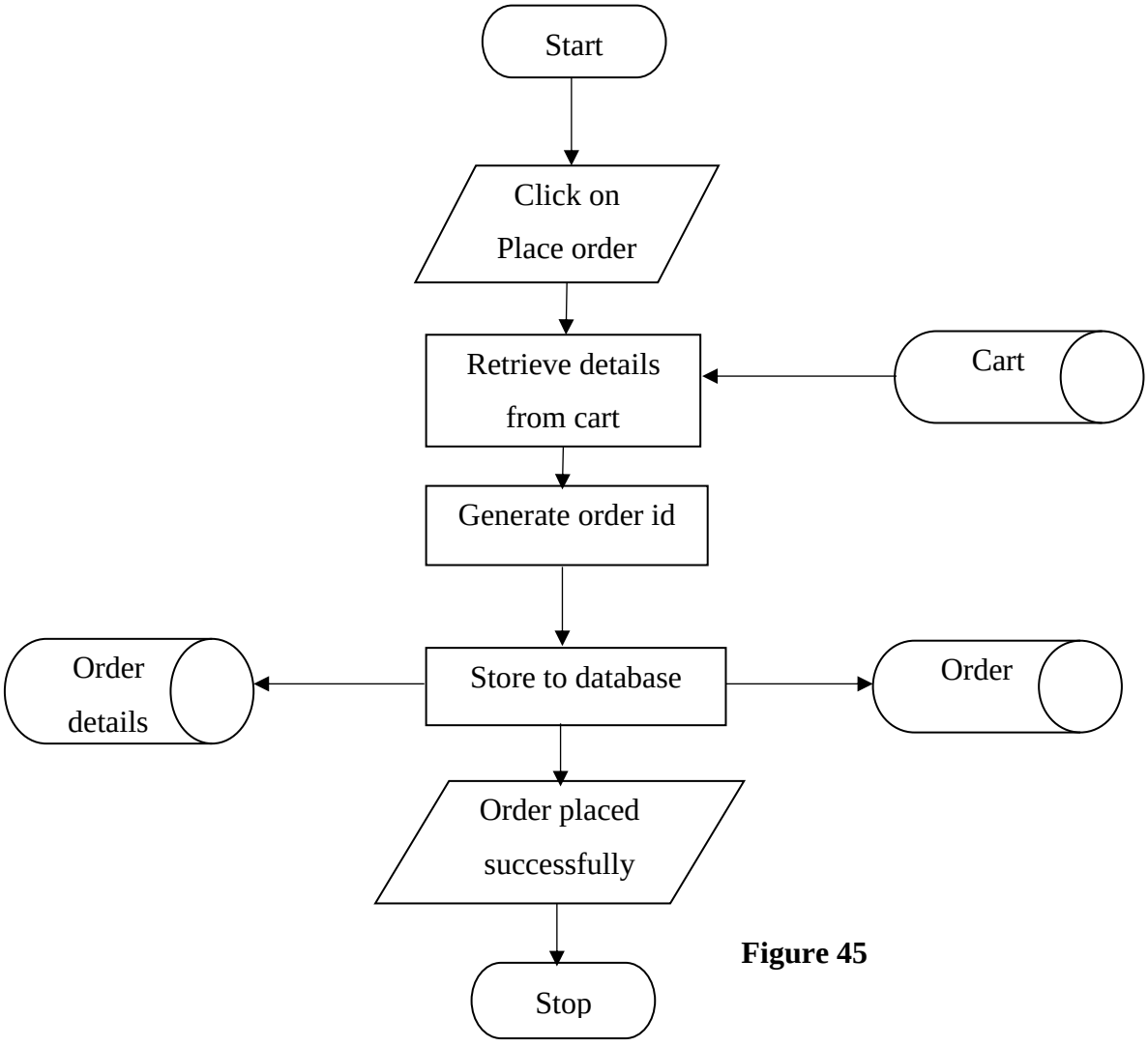


Figure 45

5.3.5.1.2.3 File I/O Interfaces

Order table, Cart table

5.3.5.1.2.4 Output

Order placed message

5.3.5.1.2.5 Implementation aspects (if any)

Buttons

5.3.5.2 Previous order

5.3.5.2.1 View bill

5.3.5.2.1.1 Input

Button clicks

5.3.5.2.1.2 Procedural Details

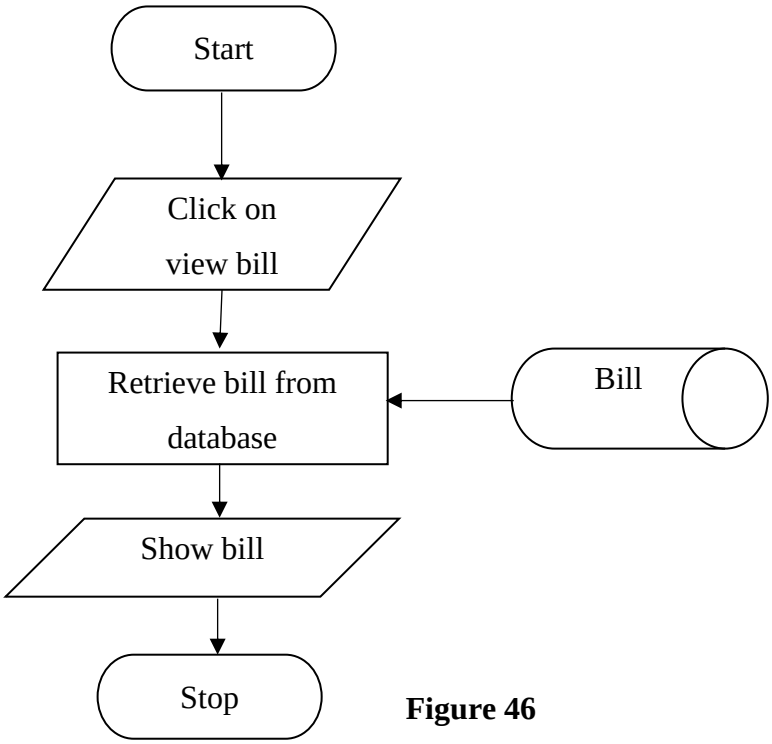


Figure 46

5.3.5.2.1.3 File I/O Interfaces

Bill table

5.3.5.2.1.4 Output

Bill will be displayed

5.3.5.2.1.5 Implementation aspects (if any)

Buttons

5.3.5.2.2 Feedback

5.3.5.2.2.1 Input

Feedback details

5.3.5.2.2.2 Procedural Details

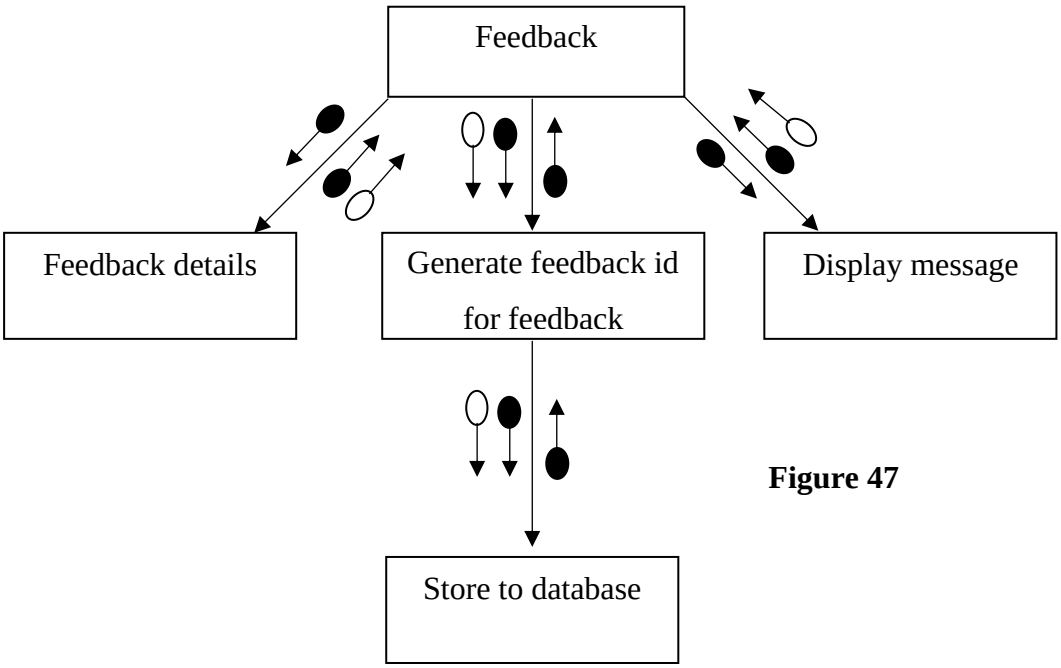


Figure 47

5.3.5.2.2.3 File I/O Interfaces

Feedback table

5.3.5.2.2.4 Output

Successful message

5.3.5.2.2.5 Implementation aspects (if any)

Textboxes, Buttons, Hyperlinks

5.3.5.3 Pending orders

5.3.5.3.1 Input

Click

5.3.5.3.2 Procedural Details

Step1: start

Step 2: CLICK on pending orders

Step 3: Retrieve dealer_id from order table where dealer_id is NULL

Step 4: DISPLAY orders which are pending.

5.3.5.3.3 File I/O Interfaces

Order table

5.3.5.3.4 Output

Pending orders will be displayed

5.3.5.3.5 Implementation aspects (if any)

Buttons

6. Program code listing

```
<?php require "./database/connect.php" ?>
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Sell Gros | Login</title>
   <link rel="stylesheet" href="general/public/css/login.css">
   <script src="general/public/js/sweetalert2.min.js"></script>
   <link rel="stylesheet" href="general/public/css/sweetalert2.min.css" />
   <link type="text/css" rel="stylesheet" href="general/public/css/fonts.css">
</head>
<body>
   <?php
      session_start();
      if(isset($_POST["user-type"])) {
         $user_type = $_POST["user-type"];
         $username = $_POST["username"];
         $password = $_POST["password"];
         $query = "select * from $user_type where username='$username'";
         $result = mysqli_query($con, $query);
         if($result == false || $result->num_rows == 0) {
            $error = "username does not exist!";
         } else {
            $hashed = md5($password);
            if($hashed == mysqli_fetch_array($result)["password"]) {
               $_SESSION["loggedin"] = "$username:$user_type:$hashed";
               if($user_type == "customer") {
                  $_SESSION["flash_message"] = "<script>
                  Swal.fire({
                     toast:true,
                     position:\"top\",
                     timer:2000,
                     \"title\":\"Logged in successfully.\",
                  })
                  </script>";
```

```
                header('Location: index.php');
                exit();
            } else if($user_type == "dealer"){
                $_SESSION["flash_message"] = "<script>
                Swal.fire({
                    toast:true,
                    position:\"top\",
                    timer:2000,
                    \"title\":\"Logged in successfully.\",
                })
                </script>";
                header('Location: dealer/index.php');
                exit();
            } else if($user_type == "admin") {
                $_SESSION["flash_message"] = "<script>
                Swal.fire({
                    toast:true,
                    position:\"top\",
                    timer:2000,
                    \"title\":\"Logged in successfully.\",
                })
                </script>";
                header('Location: admin/index.php');
                exit();
            }
        } else {
            $error = "Password mismatch!";
        }
    }
}
if(isset($_SESSION['flash_message'])) {
    $message = $_SESSION['flash_message'];
    unset($_SESSION['flash_message']);
    session_destroy();
    echo $message;
}
```

```
  ?>
  <div class="login-container">
    <h1>Login to Sell Gros</h1>
    <div class="tabs">
      <div class="tab-controls">
        <div class="tab-control" id="customer">Customer</div>
        <div class="tab-control" id="dealer">Dealer</div>
        <div class="tab-control" id="admin">Admin</div>
      </div>
      <div class="tab-body-container">
        <div class="tab-body" id="customer-tab">
          <form method="POST" action="login.php">
            <?php
              if(isset($error) && isset($user_type) && $user_type == "customer")
      {
                echo "<div class='form-element error-msg'>$error</div>";
              }
            ?>
<div class="form-element"><label>Username:</label><input class="form-field"
required type="text" name="username" /> </div>
<div class="form-element"><label>Password:</label><input class="form-field"  required
type="password" name="password" /> </div>
<div class="form-element register-and-login"><a
href="registration/customer.php">Customer Registration</a><input type="submit"
value="Login" /></div>
<input type="hidden" name="user-type" value="customer" />
</form>
 </div>
 <div class="tab-body" id="dealer-tab">
 <form method="POST" action="login.php">
 <?php
  if(isset($error) && isset($user_type) && $user_type == "dealer") {
   echo "<div class='form-element error-msg'>$error</div>";
 }
?>
```

```
<div class="form-element"><label>Username:</label><input class="form-field"
required type="text" name="username" /> </div>
<div class="form-element"><label>Password:</label> <input class="form-field"
required type="password" name="password" /> </div>
<div class="form-element register-and-login"><a href="registration/dealer.php">Dealer
Registration</a><input type="submit" value="Login" /></div>
                    <input type="hidden" name="user-type" value="dealer" />
                </form>
            </div>
            <div class="tab-body" id="admin-tab">
                <form method="POST" action="login.php">
                    <?php
                        if(isset($error) && isset($user_type) && $user_type == "admin") {
                            echo "<div class='form-element error-msg'>$error</div>";
                        }
                    ?>
                    <div class="form-element"><label>Username:</label> <input
class="form-field"  required type="text" name="username" /> </div>
                    <div class="form-element"><label>Password:</label> <input
class="form-field"  required type="password" name="password" /> </div>
                    <div class="form-element"><input type="submit" value="Login" /></div>
                    <input type="hidden" name="user-type" value="admin" />
                </form>
            </div>
        </div>
    </div>
    <script>
        const divs = [...document.querySelectorAll(".tab-control")];
        function resetActive() {
            const activeEls = [...document.querySelectorAll(".active")];
            for (const act of activeEls) {
                act.classList.remove("active");
            }
        }
```

```
        function changeActiveBody(id) {
            id = id ? id : "customer";
            const tabbody = document.querySelector("#" + id + "-tab");
            tabbody.classList.add("active");
        }
        for (const btn of divs) {
            btn.addEventListener("click", function(e) {
                const id = e.target.id;
                resetActive();
                e.target.classList.add("active");
                changeActiveBody(id);
            });
        }
        <?php
        if (isset($_POST["user-type"])) {
            $activeTab = "'{$_POST['user-type']}'";
        } else {
            $activeTab = "false";
        }
        ?>
        hangeActiveBody(<?php echo $activeTab; ?>);
        if (<?php echo $activeTab; ?>) {
            document.querySelector("#" + <?php echo $activeTab; ?>).classList.add("active");
        } else {
            document.querySelector("#customer").classList.add("active");
        }
    </script>
</body>
</html>

<?php
    session_start();
    unset($_SESSION["loggedin"]);
    $_SESSION["flash_message"] = "<script>
    Swal.fire({
```

```
        toast:true,
        position:\"top\",
        timer:2000,
        \"title\":\"Logged out successfully.\",
    })
    </script>";
    header('Location: index.php');
    exit();
?>
<?php require "database/connect.php"; ?>
<?php require "general/common/external.php"; ?>
<?php require "authentication/check_customer_logged_in.php"; ?>
<body>
    <?php require "general/common/header.php"; ?>
    <?php
    if (isset($_SESSION['flash_message'])) {
        $message = $_SESSION['flash_message'];
        unset($_SESSION['flash_message']);
        echo $message;
    }
    if (!isset($_GET["product-id"])) {
        echo "<h3>Please do not open this page directly.";
    } else {
        $query = "select * from products where product_id={$_GET['product-id']}";
        $result = mysqli_query($con, $query);
        echo mysqli_error($con);
        if ($result == false || mysqli_num_rows($result) == 0) {
            echo "invalid product id!";
        } else {
            $product = mysqli_fetch_array($result);
    ?>
            <div class="product-details-container">
                <a href="index.php" class="back-button"><img
src="general/public/images/back.png" /> Back</a>
                <div class="product-details">
                    <div class="left-section">
```

```
                <div class="product-image">
                    <img src="<?php echo "product-images/{$product["product_id"]}.jpg";
        ?>" />
                </div>
            </div>
            <div class="right-section">
                <div class="product-name">
                    <?php echo $product["product_name"] ?>
                </div>
                <div class="product-price">
                    Rs. <?php echo $product["price"] ?> / <?php echo
$product["product_unit"] ?>
                </div>
                <div class="product-description">
                    <?php echo $product["product_description"] ?>
                </div>
                <span class="product_add_to_cart">
                    <form class="add-to-cart-form" method="POST" action="cart/add-to-
cart.php">
                    <input type="hidden" name="product_id" value="<?php echo
$product['product_id']; ?>" />
                    <input type="hidden" name="quantity" value="1" />
                    <input type="hidden" name="action" value="increment" />
                        <button>Add to Cart</button>
                    </form>
                </span>
            </div>
        </div>
        <div class="divider"></div>
    </div>
  <?php
      }
  }
  ?>
</body>
</html>
```

```
<?php require "database/connect.php"; ?>
<?php require "general/common/external.php"; ?>
<?php require "authentication/check_customer_logged_in.php"; ?>
<body>
   <?php require "general/common/header.php"; ?>
   <?php
   if(isset($_SESSION['flash_message'])) {
      $message = $_SESSION['flash_message'];
      unset($_SESSION['flash_message']);
      echo $message;
   }
   ?>
   <div class="product-listing">
      <?php
      $query = " where product_status='active';";
      if(isset($_GET["search-query"])) {
         $query = " where product_name like '%{$_GET["search-query"]}%' AND
product_status='active' ";
      }
      $query = "select * from products" . $query;
      $result = mysqli_query($con, $query);
      $n = mysqli_num_rows($result);
      if ($n != 0) {
         while ($row = mysqli_fetch_array($result)) {
            $product_id = $row["product_id"];
            $product_name = $row["product_name"];
            $product_price = $row["price"];
            $product_stock = $row["stock"];
      ?>
            <a class="product" href="product.php?product-id=<?php echo
$product_id?>">
               <span class="product_image">
                  <img src="product-images/<?php echo $product_id . ".jpg?tmp=" . time();
?>" />
```

```
                </span>
                    <span class="product_name">
                  <?php echo $product_name; ?>
                </span>
                <div class="bottom-row">
                  <span class="product_price">
                    Rs. <?php echo $product_price; ?>
                  </span>
                  <span class="product_add_to_cart">
                    <form class="add-to-cart-form" action="cart/add-to-cart.php"
method="POST">
                    <input type="hidden" name="product_id" value="<?php echo
$product_id?>" />
                    <input type="hidden" name="quantity" value="1" />
                    <input type="hidden" name="action" value="increment" />
                    <button>Add to Cart</button>
                    </form>
                  </span>
                </div>
              </a>
      <?php
          }
      } else {
          echo "<h3>No products listed!</h3>";
      }
      ?>
      <div>
      <script>
          const add_to_cart_forms = [...document.querySelectorAll(".add-to-cart-form")];
          add_to_cart_forms.forEach(f => {
              f.addEventListener("submit", (e) => {
                  e.preventDefault();
                  e.stopPropagation();
                  e.target.closest("form").submit()
              })
          })
```

```
        </script>
</body>
</html>
<?php
function generateRandomString($length = 10)
{
    $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $charactersLength = strlen($characters);
    $randomString = '';
    for ($i = 0; $i < $length; $i++) {
        $randomString .= $characters[rand(0, $charactersLength - 1)];
    }
    return $randomString;
}
?>
<?php require "../database/connect.php"; ?>
<?php
$redirect = true;
include "../authentication/check_customer_logged_in.php";

if (isset($_GET["redirect_to"])) {
    $redirect_to = $_GET["redirect_to"];
} else {
    $redirect_to = "../index.php";
}
if (isset($_POST["action"])) {
    $action = $_POST["action"];
    $customer_id = $userDetails["customer_id"];
    if ($action == "remove") {
        $product_id = $_POST["product_id"];
        $query = "delete from cart where customer_id='%s' and product_id=%s";
        $query = sprintf($query, $customer_id, $product_id);
        echo $query;
        $deleted = mysqli_query($con, $query);
        if($deleted == true) {
```

```
            header("Location: {$redirect_to}");
            exit(0);
        } else {
            echo mysqli_error($con);
        }
    } else {
        if (!isset($_POST["product_id"]) || !isset($_POST["quantity"])) {
            echo "invalid request";
            exit(0);
        }
        $product_id = $_POST["product_id"];
        $quantity = $_POST["quantity"];
        $query = "select * from cart where customer_id='%s' and product_id=%s";
        $query = sprintf($query, $customer_id, $product_id);
        echo $query;
        $already_added_to_cart = mysqli_query($con, $query);
        var_dump($already_added_to_cart);
        if ($already_added_to_cart != false && mysqli_num_rows($already_added_to_cart)
!= 0) {
            $prev_quantity = mysqli_fetch_array($already_added_to_cart)["quantity"];
            $query = "update cart set quantity='%d' where product_id='%s' and
customer_id='%s'";
            $new_quantity = $prev_quantity;
            if ($action == "increment") {
                $new_quantity += $quantity;
            } else {
                $new_quantity -= $quantity;
                if ($new_quantity == 0) {
                    $new_quantity = 1;
                }
            }
            $query = sprintf($query, $new_quantity, $product_id, $customer_id);
            $added_to_cart = mysqli_query($con, $query);
            if ($added_to_cart == true) {
            $_SESSION["flash_message"] = "<script>
            Swal.fire({
```

```
            toast:true,
            timer:1000,
            position:\"top\",
            \"title\":\"Cart modified..\",
          })
          </script>";
            header("Location: {$redirect_to}");
          }
        } else {
          $query = "insert into cart values(%s,'%s', '%d')";
          $query = sprintf($query, $product_id, $customer_id, $quantity);
          $added_to_cart = mysqli_query($con, $query);
          if ($added_to_cart == true) {
          $_SESSION["flash_message"] = "<script>
          Swal.fire({
            toast:true,
            timer:1000,
            position:\"top\",
            \"title\":\"Added item to cart..\",
          })
          </script>";
            header('Location: ../index.php');
          }
        }
    }
}
?>
<?php require "../database/connect.php"; ?>
<?php require "./common/external.php"; ?>
<?php
$redirect = true;
require "../authentication/check_customer_logged_in.php"; ?>
<body>
    <?php require "./common/header.php"; ?>
    <?php
    if (isset($_SESSION['flash_message'])) {
```

```
    $message = $_SESSION['flash_message'];
    unset($_SESSION['flash_message']);
    echo $message;
  }
  ?>
  <div class="cart-container">
    <?php
    $query = "select * from cart
      inner join products
      on products.product_id=cart.product_id
      where customer_id='{$userDetails["customer_id"]}'";
    $result = mysqli_query($con, $query);
    echo mysqli_error($con);
    if ($result != false && mysqli_num_rows($result) != 0) {
      foreach ($result as $product) {
    ?>
        <div class="cart-item">
          <div class="product-image">
            <img src="../product-images/<?php echo $product["product_id"] . ".jpg" .
"?tmp=" . time(); ?>" />
          </div>
          <div class="name-description-container">
            <div class="product-name"><?php echo $product["product_name"];
?></div>
            <div class="product-description"><?php echo
$product["product_description"]; ?></div>
          </div>
          <div class="modify-cart">
            <form action="add-to-cart.php?redirect_to=cart.php" method="POST">
              <button type="submit" name="action" value="increment">+</button>
              <div class="item-quantity"><?php echo $product["quantity"]; ?></div>
              <button type="submit" name="action" <?php echo $product["quantity"]
== 1 ? "disabled" : "" ?> value="decrement">-</button>
              <input type="hidden" name="product_id" value="<?php echo
$product["product_id"] ?>" />
              <input type="hidden" name="quantity" value="1" />
```

```
                </form>
            </div>
            <div class="price-remove-container">
                <div class="price">
                    Rs. <?php echo $product["price"]; ?> / <?php echo
$product["product_unit"]; ?>
                </div>
                <form action="add-to-cart.php?redirect_to=cart.php" method="POST">
                    <input type="hidden" name="product_id" value="<?php echo
$product["product_id"] ?>" />
                    <button type="submit" value="remove"
name="action">Remove</button>
                </form>
            </div>
        </div>
    <?php
        }
        echo "<form action='place-order.php' method='POST' class='place-order-
container'>
                <button class='place-order-button' type='submit'>Place Order</button>
            </form>";
    } else {
        echo "<div class='status-messages'>";
        echo "<h2><div class='status-message'>No items added to cart!</div></h2>";

        echo "<div>";
    }
    ?>
  </div>
</body>
</html>
<?php require "../database/connect.php"; ?>
<?php require "../utils/utils.php"; ?>
<?php
$redirect = true;
include "../authentication/check_customer_logged_in.php";
```

```
$customer_id = $userDetails["customer_id"];
$query = "select * from cart
inner join products
on cart.product_id=products.product_id
where customer_id='%s'";
$query = sprintf($query, $customer_id);
$items_in_cart = mysqli_query($con, $query);
echo mysqli_error($con);
$num = mysqli_num_rows($items_in_cart);
if ($num == 0) {
    echo "no items in cart.";
} else {
    $tz = 'Asia/Kolkata';
    $timestamp = time();
    $dt = new DateTime("now", new DateTimeZone($tz)); //first argument "must" be a
string
    $dt->setTimestamp($timestamp); //adjust the object to correct timestamp
    $nowFormat = $dt->format('Y-m-d H:i:s');
    $order_id = generateRandomString();
    $customer_id = $customer_id;
    $order_date = $nowFormat;
    $order_status = "pending";
    $dealer_id = "NULL";
    $query = "INSERT INTO `order` (`order_id`, `customer_id`, `order_date`,
`order_status`, `dealer_id`) VALUES ('%s','%s','%s','%s',%s)";
    $query = sprintf(
        $query,
        $order_id,
        $customer_id,
        $order_date,
        $order_status,
        $dealer_id,
    );
    mysqli_query($con, $query);
    echo mysqli_error($con);
    $order_details = array();
```

```
    $query = "insert into `order_details` (`order_id`, `product_id`, `quantity`, `price`)
values ";
    $vals = array();
    foreach ($items_in_cart as $item) {
        $vals[] = sprintf("('%s', %d, %d, %f)", $order_id, $item["product_id"],
$item["quantity"], $item["price"]);
    }
    $query .= implode(",", $vals) . ';';
    echo $query;
    $result = mysqli_query($con, $query);
    echo mysqli_error($con);
    if ($result == true) {
        $query = "delete from cart where customer_id='%s'";
        $query = sprintf($query, $customer_id);
        $result = mysqli_query($con, $query);
        echo var_dump($result);
        if(isset($result) && $result == true) {
            $_SESSION["flash_message"] = "<script>
            Swal.fire({
                toast:true,
                timer:2000,
                position:\"top\",
                \"title\":\"Order placed successfully..\",
            })
            </script>";
            header('Location: cart.php');
        }
    } else {
        echo "Error inserting to orders table";
    }
}
?>

* {
    margin: 0;
    padding: 0;
```

```
}
body {
    background: #F0F0F0;
    font-family: Inter;
}
input, button{
    font-family: inherit;
}
/*
//
// HEADER
//
*/
header {
    background: linear-gradient(to right, #161a66, #685bd5db);
    display: flex;
    justify-content: space-between;
    padding: 20px 20px;
}
header h3 {
    color: white;
    font-size: 25px;
}
header h3 a {
    color: inherit;
    text-decoration: none;
}
header .navigation a {
    text-decoration: none;
    color: white;
    border: 2px #ffffff59 solid;
    padding: 6px 20px;
    border-radius: 10px;
    margin: 0 5px;
    display: inline-block;
    background: #143b4b63;
```

```
}
.header-right {
    display: flex;
    justify-content: center;
    align-items: center;
}
form.search-form {
    margin-right: 2rem;
}
.search-form input[type="text"] {
    padding: 8px 7px;
    background: none;
    border: 2px solid;
    border-radius: 6px;
    border-color: #c3c3c3;
    width: 250px;
    color: white;
}
.search-form input[type="submit"] {
    border: 2px solid;
    padding: 8px 12px;
    background: none;
    color: white;
    border-radius: 6px;
    border-color: #e5e5e580;
}
.search-form input[type="text"]::placeholder {
    color: white;
}
/*
//
// HOME PAGE
//
*/
.product-listing {
    width: 86%;
```

```
    margin: auto;
    margin-top: 2rem;
    background: white;
    min-height: 85vh;
    margin-bottom: 2rem;
    display: grid;
    grid-template-columns: repeat(6, 200px);
    justify-content: center;
}
.product {
    width: 200px;
    height: fit-content;
    display: inline-block;
    box-shadow: 0 0 20px #7676761f;
    border-radius: 1px;
    text-decoration: none;
    position: relative;
    color: black;
}
.product .bottom-row {
    display: flex;
    justify-content: space-between;
    margin: 18px 20px;
    align-items: center;
    margin-top: 0;
}
span.product_image img {
    width: 135px;
    max-height: 140px;
}
span.product_image {
    width: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 165px;
```

```
}
span.product_name {
    display: block;
    font-size: 1.4rem;
    text-align: center;
}
span.product-description {
    position: absolute;

    background: white;
    /* left: -19px; */
    padding: 18px;
    box-shadow: 0 0 40px #00000026;
    top: 0px;
    opacity: 0;
    user-select: none;
    pointer-events: none;
    transition: 0.3s all cubic-bezier(0.42, 0, 0.25, 1.11);
    line-height: 33px;
    border-radius: 11px;
    top: 200px;
    box-sizing: border-box;
    width: 100%;
}
.product:hover span.product-description {
    opacity: 1;
    top: 268px;
}
span.product_add_to_cart button {
    background: green;
    border: none;
    border-radius: 3px;
    width: 91px;
    padding: 9px 0px;
    color: white;
    font-weight: 500;
```

```
}
a.product:hover {
    box-shadow: 0 0 11px #76767669;
}
/*
//
//    PRODUCT DETAILS PAGE
//
*/
.product-details-container {
    width: 86%;
    margin: auto;
    margin-top: 2rem;
    background: white;
    min-height: 85vh;

    display: flex;
    justify-content: flex-start;
    flex-direction: column;
    align-items: center;
    position: relative;
}
.product-details {
    width: fit-content;
    margin-top: 10rem;
    display: flex;
}
.right-section {
    display: flex;
    flex-direction: column;
    /* border: 1px solid; */
    height: fit-content;
    margin-left: 6rem;
}
.product-name {
    font-size: 3rem;
```

```
}
.product-price {
    font-size: 1rem;
    margin: 1rem 0rem;
}
.product-description {
    /* background: red; */
    max-width: 250px;
    overflow: hidden;
    line-height: 1.8rem;
}
span.product_add_to_cart {
    margin: 1rem 0;
}
.product-details .product_add_to_cart button {
    padding: 1rem 2rem;
    width: auto;
    font-size: 1rem;
}
.divider {
    width: 58%;
    height: 2px;
    background: #dddddd3d;
    margin-top: 15rem;
}
a.back-button {
    text-decoration: none;
    display: flex;
    align-items: center;
    justify-content: center;
    position: absolute;
    top: 5rem;
    left: 15rem;
    color: #3699f3;
    font-size: 1.3rem;
}
```

```
.back-button img {
    width: 43px;
    margin-right: 0.6rem;
}
.product-details-container div.product-details>div.left-section>div>img {
    width: 250px;
}
.previous-orders .orders-list {
    margin: 2rem;
}

.previous-orders .order {
    display: flex;
    flex-direction: column;
    background: white;
    border: 1px solid rgba(0, 0, 0, 0.17);
}
.previous-orders .middle-row {
    display: flex;
    margin: 2rem;
    margin-top: 1.5rem;
}
.previous-orders .order-detail {
    display: flex;
    flex-direction: column;
}
.previous-orders .left-column {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-gap: 34px;
    grid-column-gap: 45px;
}
.previous-orders .right-column {
    height: min-content;
    align-self: flex-end;
    margin-left: auto;
```

```
    display: flex;
}
.previous-orders .order_id {
    font-weight: 500;
    font-size: 20px;
    line-height: 23px;
    margin-left: 2rem;
    margin-top: 2rem;
}
.previous-orders span.detail-label {
    font-weight: 500;
    font-size: 18px;
}
.previous-orders span.detail-value {
    font-size: 18px;
}
.previous-orders .right-column select {
    padding: 0.5rem 0.5rem;
    background: white;
}
.previous-orders .right-column button {
    padding: 0.5rem 0.8rem;
    font-weight: 600;
    background: #CBD0FF;
    border: 1px solid #787878;
    color: black;
}
.previous-orders .right-column label {
    margin-bottom: 0.5rem;
    display: block;
    font-weight: 500;
    font-size: 18px;
}
table.bottom-row {
    text-align: center;
    border: 1px solid;
```

```
    margin: 2rem;
    margin-top: 0;
}
.bottom-row tr {
    height: 43px;
    background: #f8f8f8;
}
.bottom-row tr th {
    background: #cad0ff;
}
.order-total {
    text-align: right;
    margin-right: 3rem;
    margin-bottom: 2rem;
}
.order-total span {
    background: #cad0ff;
    color: black;
    font-weight: bold;
    display: inline-block;
    padding: 0.5rem 1rem;
    margin-left: 0.5rem;
    border-radius: 6px;
}
.feedback-form-container {
    display: flex;
    flex-direction: column;
    align-items: stretch;
    width: 100%;
}
.feedback-form-container form {
    display: flex;
    flex-direction: column;
    margin: 2rem;
}
.feedback-form-container form button {
```

```
    width: 100px;
    padding: 0.4rem;
    margin-top: 1rem;
    background: #bdc4f7;
}
.feedback-form-container h3 {
    margin: 0 2rem;
}
.feedback-form-container form textarea {
    height: 100px;
    padding:1rem;
}

<?php require "database/connect.php"; ?>
<?php require "general/common/external.php"; ?>
<?php
$redirect = true;
$redirect_uri = "login.php";
require "authentication/check_customer_logged_in.php"; ?>
<body>
<?php require "general/common/header.php"; ?>
<?php
if(isset($_SESSION['flash_message'])) {
    $message = $_SESSION['flash_message'];
    unset($_SESSION['flash_message']);
    echo $message;
}
?>
<div class="main-content previous-orders">
<div class="orders-list">
<?php
$query = "select * from `order`
where customer_id = '{$userDetails['customer_id']}' and order_status='pending' ORDER
BY order_date DESC";
$orders = mysqli_query($con, $query);
echo mysqli_error($con);
```

```
if(mysqli_num_rows($orders) == 0) {
   echo "<h1>No orders.</h1>";
   exit(0);
}
foreach($orders as $order) {
   echo "<div class='order'>";
   echo "<div class='order_id'>Order #{$order['order_id']}</div>";
   echo "<div class='middle-row'>";
   echo "<div class='left-column'>";
   echo "<div class='order-detail'>
      <span class='detail-label'>Customer ID</span>
      <span class='detail-value'>{$order['customer_id']}</span>
   </div>";

   $query = "select * from order_details where order_id='{$order['order_id']}'";
   $order_items = mysqli_query($con, $query);
   $no_of_items = mysqli_num_rows($order_items);
   echo "<div class='order-detail'>
   <span class='detail-label'>Order Items Count</span>
   <span class='detail-value'>{$no_of_items}</span>
   </div>";
   $date = date('Y/m/d H:i:s',strtotime($order['order_date']));

   echo "<div class='order-detail'>
   <span class='detail-label'>Order Date</span>
   <span class='detail-value'>{$date}</span>
   </div>";
   echo "<div class='order-detail'>
   <span class='detail-label'>Order Status</span>
   <span class='detail-value'>{$order['order_status']}</span>
   </div>";
   echo "</div>";
   echo "</div>";
   echo "<table class='bottom-row'>
   <tr>
      <th>Product Name</th>
```

```
        <th>Product Price</th>
        <th>Product Quantity</th>
        <th>Product Total</th>
    </tr>
    ";
    $query = "select * from order_details
    inner join products
    on products.product_id=order_details.product_id
    where order_id='{$order['order_id']}'";
    $order_items = mysqli_query($con, $query);
    $no_of_items = mysqli_num_rows($order_items);
    $order_total = 0;
    foreach($order_items as $product) {
        $total=$product['price'] * $product['quantity'];
        $order_total += $total;
        echo "<tr>";
        echo "<td>{$product['product_name']}</td>";
        echo "<td>{$product['price']} / {$product['product_unit']}</td>";
        echo "<td>{$product['quantity']}</td>";
        echo "<td>{$total}</td>";
        echo "</tr>";
    }
    echo "</table>";
    echo "<div class='order-total'>Order Total <span>Rs. {$order_total}</span></div>";
    echo "</div>";
}
?>
</div>
</div>
</div>
</body>
</html>
<?php require "database/connect.php"; ?>
<?php require "general/common/external.php"; ?>
<?php
$redirect = true;
```

```
$redirect_uri = "login.php";
require "authentication/check_customer_logged_in.php"; ?>
<body>
<?php require "general/common/header.php"; ?>
<?php
if(isset($_SESSION['flash_message'])) {
   $message = $_SESSION['flash_message'];
   unset($_SESSION['flash_message']);
   echo $message;
}
?>
<div class="main-content previous-orders">
<div class="orders-list">
<?php
$query = "select * from `order`
where customer_id = '{$userDetails['customer_id']}' and order_status='processed' order
by order_date DESC limit 30;";
$orders = mysqli_query($con, $query);
echo mysqli_error($con);
if(mysqli_num_rows($orders) == 0) {
   echo "<h1>No orders.</h1>";
   exit(0);
}
foreach($orders as $order) {
   echo "<div class='order'>";
   echo "<div class='order_id'>Order #{$order['order_id']}</div>";
   echo "<div class='middle-row'>";
   echo "<div class='left-column'>";
   echo "<div class='order-detail'>
      <span class='detail-label'>Customer ID</span>
      <span class='detail-value'>{$order['customer_id']}</span>
   </div>";
   $query = "select * from order_details where order_id='{$order['order_id']}'";
   $order_items = mysqli_query($con, $query);
   $no_of_items = mysqli_num_rows($order_items);
   echo "<div class='order-detail'>
```

```
<span class='detail-label'>Order Items Count</span>
<span class='detail-value'>{$no_of_items}</span>
</div>";
$date = date('Y/m/d H:i:s',strtotime($order['order_date']));
echo "<div class='order-detail'>
<span class='detail-label'>Order Date</span>
<span class='detail-value'>{$date}</span>
</div>";
echo "<div class='order-detail'>
<span class='detail-label'>Order Status</span>
<span class='detail-value'>{$order['order_status']}</span>
</div>";
echo "</div>";
echo "
<div class='right-column'>
<form action='view-bill.php' method='POST' class='view-bill-form'>
";
echo "
<input type='hidden' name='order_id' value='{$order['order_id']}' />
<button type='submit' style='margin-right:10px;' value='view-bill'>View Bill</button>
</form>
<form action='feedback.php' method='GET'>
";
 echo "
 <input type='hidden' name='order_id' value='{$order['order_id']}' />
 <button type='submit' value='feedback'>feedback</button>
 </form>
</div>
";
echo "</div>";
echo "<table class='bottom-row'>
<tr>
    <th>Product Name</th>
    <th>Product Price</th>
    <th>Product Quantity</th>
    <th>Product Total</th>
```

```
    </tr>
    ";
    $query = "select * from order_details
    inner join products
    on products.product_id=order_details.product_id
    where order_id='{$order['order_id']}'";
    $order_items = mysqli_query($con, $query);
    $no_of_items = mysqli_num_rows($order_items);
    $order_total = 0;
    foreach($order_items as $product) {
        $total=$product['price'] * $product['quantity'];
        $order_total += $total;
        echo "<tr>";
        echo "<td>{$product['product_name']}</td>";
        echo "<td>{$product['price']} / {$product['product_unit']}</td>";
        echo "<td>{$product['quantity']}</td>";
        echo "<td>{$total}</td>";
        echo "</tr>";
    }
    echo "</table>";
    echo "<div class='order-total'>Order Total <span>Rs. {$order_total}</span></div>";
    echo "</div>";
}
?>
</div>
</div>
</div>
<script>
    const viewbills = [...document.getElementsByClassName("view-bill-form")];
    function func(e) {
        e.preventDefault();
        let form= new FormData(e.target);
        const order_id = form.get("order_id");
        const temp = document.createElement("iframe");
        temp.setAttribute("src", `temp.php?order_id=${order_id}`);
        temp.style="width:0; height:0; visibility:none";
```

```
        window.document.body.appendChild(temp);
    }
    viewbills.forEach(vb => {
        vb.addEventListener("submit", (e) => func(e));
    })
</script>
</body>
</html>

<?php
    require "./database/connect.php";
    $redirect = true;
    $redirect_uri = "login.php";
    require "authentication/check_customer_logged_in.php"; ?>
<h1 align="center">
    SellGros
</h1>
<style>
    h1 {
        margin:2rem 0;
    }
.order {
    display: flex;
    flex-direction: column;
    background: white;
    border: 1px solid rgba(0, 0, 0, 0.17);
}
.middle-row {
    display: flex;
    margin: 2rem;
    margin-top: 1.5rem;
}
.order-detail {
    display: flex;
    flex-direction: column;
}
```

```
.left-column {
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-gap: 34px;
    grid-column-gap: 45px;
}
.right-column {
    height: min-content;
    align-self: flex-end;
    margin-left: auto;
}
.order_id,.bill_id {
    font-weight: 500;
    font-size: 20px;
    line-height: 23px;

}
.top-details {display: flex;margin: 2rem 2rem;margin-bottom: 0;}
.bill_id {margin-left: 3rem;}
span.detail-label {
    font-weight: 500;
    font-size: 18px;
}
span.detail-value {
    font-size: 18px;
}
.right-column select {
    padding: 0.5rem 0.5rem;
    background: white;
}
.right-column button {
    padding: 0.5rem 0.8rem;
    font-weight: 600;
    background: #CBD0FF;
    border: 1px solid #787878;
    color: black;
```

```
}
.right-column label {
   margin-bottom: 0.5rem;
   display: block;
   font-weight: 500;
   font-size: 18px;
}
table.bottom-row {text-align: center;border: 1px solid;margin: 2rem;margin-top: 0;}
table.bottom-row tr {height: 43px;background: #f8f8f8;}
table.bottom-row tr th {background: #cad0ff;}
.order-total {text-align: right;margin-right: 3rem;margin-bottom: 2rem;}
.order-total span {background: #cad0ff;color: black;font-weight: bold;display: inline-
block;padding: 0.5rem 1rem;margin-left: 0.5rem;border-radius: 6px;}
</style>
<?php

$query = "SELECT *
FROM `bill`
INNER JOIN `order`
ON bill.order_id = order.order_id
where bill.order_id='{$_GET["order_id"]}'";
$order = mysqli_fetch_array(mysqli_query($con, $query));
echo mysqli_error($con);
echo "<div class='order'>";
echo "<div class='top-details'><div class='order_id'>Order
#{$_GET["order_id"]}</div>";
echo "<div class='bill_id'>Bill #{$order["bill_id"]}</div></div>";
echo "<div class='middle-row'>";
echo "<div class='left-column'>";
echo "<div class='order-detail'>
   <span class='detail-label'>Customer ID</span>
   <span class='detail-value'>{$order['customer_id']}</span>
</div>";
$query = "select * from bill_details where bill_id='{$order["bill_id"]}'";
$order_items = mysqli_query($con, $query);
$no_of_items = mysqli_num_rows($order_items);
```

```
echo "<div class='order-detail'>
<span class='detail-label'>Order Items Count</span>
<span class='detail-value'>{$no_of_items}</span>
</div>";
$date = date('Y/m/d H:i:s',strtotime($order['bill_date']));
echo "<div class='order-detail'>
<span class='detail-label'>Bill Date</span>
<span class='detail-value'>{$date}</span>
</div>";
echo "<div class='order-detail'>
<span class='detail-label'>Order Status</span>
<span class='detail-value'>{$order['order_status']}</span>
</div>";
echo "</div>";
echo "</div>";
echo "<table class='bottom-row'>
<tr>
   <th>Product Name</th>
   <th>Product Price</th>
   <th>Product Quantity</th>
   <th>Product Total</th>
</tr>
";
$query = "select * from order_details
inner join products
on products.product_id=order_details.product_id
where order_id='{$_GET["order_id"]}'";
$order_items = mysqli_query($con, $query);
$no_of_items = mysqli_num_rows($order_items);
$order_total = 0;
foreach($order_items as $product) {
   $total=$product['price'] * $product['quantity'];
   $order_total += $total;
   echo "<tr>";
   echo "<td>{$product['product_name']}</td>";
   echo "<td>{$product['price']} / {$product['product_unit']}</td>";
```

```
    echo "<td>{$product['quantity']}</td>";
    echo "<td>{$total}</td>";
    echo "</tr>";
}
echo "</table>";
echo "<div class='order-total'>Order Total <span>Rs. {$order_total}</span></div>";
echo "</div>";
?>
<script>
    print();
    window.close();
</script>
<?php require "../database/connect.php" ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="../general/public/css/registration.css">
    <link type="text/css" rel="stylesheet" href="../general/public/css/fonts.css">
</head>
<body>
    <?php
    if(isset($_POST["username"])) {
        $username = $_POST["username"];
        $password = $_POST["password"];
        $query = "insert into admin values('%s', '%s')";
        $query = sprintf($query, $username, md5($password));
        $ret = mysqli_query($con, $query);
        if($ret == true) {
            echo "<h3>Registered successfully.</h3>";
        }
    }
    ?>
```

```
<header>
   <h3><a href="index.php">Sell Gros</a></h3>
</header>
   <div class="register-container">
   <h1>Admin Registration</h1>
   <form src="admin.php" method="POST">
      <div class="form-element">
            <label>Username: </label> <input type="text"  class="form-field"
name="username" />
      </div>
      <div class="form-element">
            <label>Password: </label> <input type="text"  class="form-field"
name="password" />
      </div>
      <div class="form-element">
            <input type="submit" value="submit" />
      </div>
   </form>
</body>
</html>
<?php require "../database/connect.php" ?>
<?php require "../utils/utils.php"; ?>
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Sell Gros | Login</title>
   <link rel="stylesheet" href="../general/public/css/registration.css">
   <link type="text/css" rel="stylesheet" href="../general/public/css/fonts.css">
</head>
<body>
   <?php
   if (isset($_POST["full_name"])) {
      $full_name = $_POST["full_name"];
      $username = $_POST["username"];
      $password = $_POST["password"];
      $address = $_POST["address"];
```

```
    $region_id = $_POST["region_id"];
    $phone_no = $_POST["phone_no"];
    $query = "INSERT INTO customer values ('%s', '%s', '%s', '%s', '%s', '%s', '%s')";
    $query = sprintf(
        $query,
        generateRandomString(),
        $address,
        $region_id,
        $phone_no,
        $username,
        md5($password),
        $full_name,
    );
    $ret = mysqli_query($con, $query);
    echo mysqli_error($con);
    echo var_dump($ret);
    session_start();
    $_SESSION["flash_message"] = "<script>
        Swal.fire({
            position:\"top\",
            timer:2000,
            \"title\":\"Registered Customer Successfully.\",
        })
        </script>";
    header('Location: ../login.php');
    exit();
}
?>
<header>
    <h3><a href="../index.php">Sell Gros</a></h3>
</header>
<div class="register-container">
    <h1 class="heading">Customer Registration</h1>
    <form method="POST" action="customer.php">
        <?php
        if (isset($error)) {
```

```
            echo "<div class='form-element error-msg'>$error</div>";
        }
        ?>
        <div class="form-element"> <label>Full Name</label><input type="text"
required class="form-field" id="full_name" name="full_name" /> </div>
        <div class="form-element"> <label>Username</label><input type="text"
required class="form-field" name="username" /> </div>
        <div class="form-element-group">
            <div class="form-element"> <label>Password</label><input type="password"
class="form-field" name="password" /> </div>
            <div class="form-element"> <label>Region</label>
                <select name="region_id" class="select-field" id="region_id">
                    <?php
                    $query = "select * from region where region_status='active'";
                    $regions = mysqli_query($con, $query);
                    if ($regions) {
                        while ($row = mysqli_fetch_array($regions)) {
                            echo "<option
value='{$row['region_id']}'>{$row["region_name"]}</option>";
                        }
                    }
                    ?>
                </select>
            </div>
        </div>
        <div class="form-element"> <label>Address</label><input type="text" required
class="form-field" name="address" /> </div>
        <div class="form-element"> <label>Phone Number</label><input type="text"
id="phone_no" minlength="10" required class="form-field" name="phone_no" /> </div>
        <div class="form-element"><input type="submit" value="Register" /></div>
        <input type="hidden" name="user-type" value="customer" />
    </form>
</div>
<script>
    const textOnlyFields = [document.querySelector("#full_name")];
    for (let field of textOnlyFields) {
```

```
            field.oninput = function(e) {
                if (/\d/.test(field.value)) {
                    field.setCustomValidity("Please enter text only!");
                    field.reportValidity();
                } else {
                    field.setCustomValidity("");
                }
                field.checkValidity();
            }
            field.onchange = (e) => {
                field.checkValidity();
            }
        }
        const numberOnlyValidation = [document.querySelector("#phone_no")];
        for (let field of numberOnlyValidation) {
            field.oninput = function(e) {
                if (field.value.match(/[^$,.\d]/)) {
                    field.setCustomValidity("Please enter number only!");
                    field.reportValidity();
                } else {
                    field.setCustomValidity("");
                }
                field.checkValidity();
            }
            field.onchange = (e) => {
                field.checkValidity();
            }
        }
    </script>
</body>
</html>

<?php require "../database/connect.php" ?>
<?php require "../utils/utils.php"; ?>
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <title>Sell Gros | Login</title>
    <link rel="stylesheet" href="../general/public/css/registration.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link type="text/css" rel="stylesheet" href="../general/public/css/fonts.css">
</head>
<body>
    <?php
    if (isset($_POST["username"])) {
        $username = $_POST["username"];
        $licence_id = $_POST["licence_id"];
        $phone_no = $_POST["phone_no"];
        $address = $_POST["address"];
        $region_id = $_POST["region_id"];
        $status = "pending";
        $dealer_name = $_POST["dealer_name"];
        $password = $_POST["password"];

        $query = "INSERT INTO `dealer`(`username`, `dealer_id`, `licence_id`, `phone_no`,
`address`, `region_id`, `status`, `dealer_name`, `password`) VALUES
('%s','%s','%s','%s','%s','%s','%s','%s','%s')";
        $query = sprintf(
            $query,
            $username,
            generateRandomString(),
            $licence_id,
            $phone_no,
            $address,
            $region_id,
            $status,
            $dealer_name,
            md5($password),
        );

        $ret = mysqli_query($con, $query);
```

```
        $err = mysqli_error($con);
        echo var_dump($ret);
        if ($ret == true) {
            session_start();
            $_SESSION["flash_message"] = "<script>
                Swal.fire({
                    position:\"top\",
                    timer:2000,
                    \"title\":\"Registered dealer Successfully.\",
                })
                </script>";
            header('Location: ../login.php');
            exit();
        } else {
            $error = $err;
        }
    }
    ?>
    <header>
        <h3><a href="../index.php">Sell Gros</a></h3>
    </header>
    <div class="register-container">
        <h1 class="heading">Dealer Registration</h1>
        <form method="POST" action="dealer.php">
            <?php
            if (isset($error)) {
                echo "<div class='form-element error-msg'>$error</div>";
            }
            ?>
            <div class="form-element">
                <label>Dealer Name</label>
                <input type="text" required id="dealer_name" name="dealer_name"
class="form-field" />
            </div>
            <div class="form-element">
                <label>User Name</label>
```

```
            <input type="text" required name="username" class="form-field" />
        </div>
        <div class="form-element-group">
            <div class="form-element">
                <label>Password</label>
                <input type="password" required name="password" class="form-field" />
            </div>
            <div class="form-element">
                 <label>Region</label>
                <select name="region_id" class="select-field" id="region">
                    <?php
                    $query = "select * from region where region_status='active'";
                    $regions = mysqli_query($con, $query);
                    if ($regions) {
                        while ($row = mysqli_fetch_array($regions)) {
                            echo "<option
value='{$row['region_id']}'>{$row["region_name"]}</option>";
                        }
                    }
                    ?>
                </select>
            </div>
        </div>

        <div class="form-element">
            <label>License ID</label>
            <input type="text" required name="licence_id" class="form-field" />
        </div>
        <div class="form-element">
            <label>Phone No</label>
            <input type="text" required id="phone_no" name="phone_no" minlength="10"
class="form-field" />
        </div>
        <div class="form-element">
            <label>Address</label>
            <input type="text" required name="address" class="form-field" />
```

```
            </div>
            <div class="form-element">
                <input type="submit" value="Register" /></div>
            <input type="hidden" name="user-type" value="dealer" />
        </form>
    </div>
    <script>
        const textOnlyFields = [document.querySelector("#dealer_name")];
        for(let field of textOnlyFields) {
            field.oninput = function(e) {
                if(/\d/.test(field.value)) {
                    field.setCustomValidity("Please enter text only!");
                    field.reportValidity();
                } else {
                    field.setCustomValidity("");
                }
                field.checkValidity();
            }
        }
        const numberOnlyValidation = [document.querySelector("#phone_no")];
        for(let field of numberOnlyValidation) {
            field.oninput = function(e) {
                if(field.value.match(/[^$,.\d]/)) {
                    field.setCustomValidity("Please enter number only!");
                    field.reportValidity();
                } else {
                    field.setCustomValidity("");
                }
                field.checkValidity();
            }
            field.onchange = (e) => {
                field.checkValidity();
            }
        }
    </script>
</body>
```

```
</html>
<?php
   $logged_in = false;
   $userDetails = NULL;
   session_start();
   if(isset($_SESSION["loggedin"])) {
      $token = $_SESSION["loggedin"];
      $arr = explode(":", $token);
      $username = $arr[0];
      $user_type = "admin";
      $hash = $arr[2];
      $query = "select * from $user_type where username='$username'";
      $result = mysqli_query($con, $query);
      if($result == false || $result->num_rows == 0) {
         unset($_SESSION["loggedin"]);
         header('Location: /sellgros/login.php');
         exit(0);
      } else {
         $row = mysqli_fetch_array($result);
         if($hash == $row["password"]) {
            $logged_in = true;
            $userDetails = $row;
         }
      }
   } else {
      header('Location: ../login.php');
      exit(0);
   }
?>
<?php
$logged_in = false;
$userDetails = NULL;
session_start();
if(isset($_SESSION["loggedin"])) {
      $token = $_SESSION["loggedin"];
      $arr = explode(":", $token);
```

```
        $username = $arr[0];
        $user_type = "customer";
        $hash = $arr[2];
        $query = "select * from $user_type where username='$username'";
        $result = mysqli_query($con, $query);
        if($result == false || $result->num_rows == 0) {
            if(isset($redirect) && $redirect = true) {
                if(isset($redirect_uri)) {
                    header("Location: $redirect_uri");
                } else {
                    header('Location: ../login.php');
                }
            }
        } else {
            $row = mysqli_fetch_array($result);
            if($hash == $row["password"]) {
                $logged_in = true;
                $userDetails = $row;
            }
        }
    } else {
        if(isset($redirect) && $redirect = true) {
            if(isset($redirect_uri)) {
                header("Location: $redirect_uri");
            } else {
                header('Location: ../login.php');
            }
            exit(0);
        }
    }
?>
<?php
    $logged_in = false;
    $userDetails = NULL;
    session_start();
    if(isset($_SESSION["loggedin"])) {
```

```
        $token = $_SESSION["loggedin"];
        $arr = explode(":", $token);
        $username = $arr[0];
        $user_type = "dealer";
        $hash = $arr[2];
        $query = "select * from $user_type where username='$username'";
        $result = mysqli_query($con, $query)
        if($result == false || $result->num_rows == 0) {
            unset($_SESSION["loggedin"]);
            header('Location: /sellgros/login.php');
            exit(0);
        } else {
            $row = mysqli_fetch_array($result);
            if($hash == $row["password"] && $row['status'] == 'approve') {
                $logged_in = true;
                $userDetails = $row;
            } else {
                $logged_in = true;
                $userDetails = $row;
                if($_SERVER["REQUEST_URI"] != '/sellgros/dealer/index.php') {
                    header('Location: ../index.php');
                };
            }
        }
    } else {
        header('Location: ../login.php');
        exit(0);
    }
?>
<?php include "base.php"; ?>
<?php include "../../utils/utils.php"; ?>
<?php
if (isset($_POST["product-name"])) {
    $product_id = mt_rand(1000000, 9999999);
    if (isset($_FILES["product-image"]) && $_FILES["product-image"]["size"] != 0) {
        $target_dir = "../../product-images/";
```

```
        $target_file = $target_dir . $product_id . ".jpg";
        if (move_uploaded_file($_FILES["product-image"]["tmp_name"], $target_file)) {
            echo "The file " . htmlspecialchars(basename($_FILES["product-
image"]["name"])) . " has been uploaded.";
        } else {
            echo "Sorry, there was an error uploading your file.";
        }
    }
    $query = "insert into `products` values('%d','%s', %f, %d, '%s', '%s', 'active')";
    $query = sprintf($query, $product_id, $_POST["product-name"], $_POST["product-
price"], $_POST["product-stock"], $_POST["product-description"], $_POST["product-
unit"]);
    $inserted = mysqli_query($con, $query);
    if ($inserted == false) {
        echo $query . "<br>";
        echo mysqli_error($con);
    } else {
        $_SESSION["flash_message"] = "<script>
        Swal.fire({
            toast:true,
            timer:2000,
            position:\"top\",
            \"title\":\"Created product successfully.\",
        })
        </script>";
         header('Location: product-management.php');
    }
}
?>
<div class="main-content product-add-page">
    <form method="POST" action="add-product.php" enctype="multipart/form-data">
        <input type="hidden" required name="action" value="update" />

        <div class="form-field">
            <input type="file" name="product-image" required id="product-image"
accept=".jpg,.jpeg" />
```

```
        </div>
        <div class="form-field"> Product Name <input type="text" required
name="product-name" /></div>
        <div class="form-field"> Product Price <input type="number" required
name="product-price" /></div>
        <div class="form-field"> Product Stock <input type="number" required
name="product-stock" /></div>

        <div class="form-field"> Product Unit <input type="text" required name="product-
unit" /></div>
        <div class="form-field"> Product description <textarea name="product-description"
id="product-description" cols="30" rows="10"></textarea></div>
        <input type="submit" value="Add Product" />
    </form>
</div>
<?php include "footer.php"; ?>
<?php include "base.php"; ?>
<?php include "../../utils/utils.php"; ?>
<?php
if(isset($_POST["region-name"])) {
    $query = "insert into region values ('%s','%s', %d, 'active')";
    $query = sprintf($query,GenerateRandomString(), $_POST["region-name"],
$_POST["pincode"] );
    $inserted = mysqli_query($con, $query);
    if($inserted == false) {
        echo mysqli_error($con);
    } else {
        $_SESSION["flash_message"] = "<script>
        Swal.fire({
            toast:true,
            timer:2000,
            position:\"top\",
            \"title\":\"Created region successfully.\",
        })
        </script>";
        header('Location: region-management.php');
```

```
    }
}
?>
<div class="main-content add-region-page">
    <form method="POST" action="add-region.php">
            <div class="form-field">Region Name <input type="text" required
name="region-name" /></div>
            <div class="form-field">Pincode <input id="pincode" minlength="6"
maxlength="6" type="text" required name="pincode"/></div>
            <input type="submit" value="Add Region" />
    </form>
</div>
<script>
     const numberOnlyValidation = [document.querySelector("#pincode")];
for (let field of numberOnlyValidation) {
    field.oninput = function(e) {
        if (field.value.match(/[^$,.\d]/)) {
            field.setCustomValidity("Please enter number only!");
            field.reportValidity();
        } else {
            field.setCustomValidity("");
        }
        field.checkValidity();
    }
    field.onchange = (e) => {
        field.checkValidity();
    }
}
    </script>
<?php include "footer.php"; ?>
<?php include "base.php"; ?>
<div class="main-content">
System.out.printf("%d", x);
            <?php
$order_id = $_POST["order_id"];
$dealer_id = $_POST["dealer_id"];
```

```
$query = "update `order`
set dealer_id='%s'
where order_id='%s'
";
$query = sprintf($query, $dealer_id, $order_id);
echo $query;
$result = mysqli_query($con, $query);
echo mysqli_error($con);
if($result == true) {
        $_SESSION["flash_message"] = "<script>
        Swal.fire({
                toast:true,
                position:\"top\",
                timer:2000,
                \"title\":\"Dealer allotted successfully..\",
        })
        </script>";
        header('Location: dealer-allotment.php');
};
?>
</div>
<?php  include "base.php"; ?>
<div class="main-content dealer-allotment">
<div class="orders-list">
   <?php
   $query = "select * from `order`
   inner join customer
   on customer.customer_id=order.customer_id
   where dealer_id IS NULL
   ORDER BY order_date DESC
   ;";
   $orders = mysqli_query($con, $query);
   echo mysqli_error($con);
   if(mysqli_num_rows($orders) == 0) {
      echo "<h1>No orders.</h1>";
      exit(0);
```

```
}
foreach($orders as $order) {
    echo "<div class='order'>";
    echo "<div class='order_id'>Order #{$order['order_id']}</div>";
    echo "<div class='bottom-row'>";
    echo "<div class='left-column'>";
    echo "<div class='order-detail'>
        <span class='detail-label'>Customer ID</span>
        <span class='detail-value'>{$order['customer_id']}</span>
    </div>";
    $query = "select * from order_details where order_id='{$order['order_id']}'";
    $order_items = mysqli_query($con, $query);
    $no_of_items = mysqli_num_rows($order_items);
    echo "<div class='order-detail'>
    <span class='detail-label'>Order Items Count</span>
    <span class='detail-value'>{$no_of_items}</span>
    </div>";
    $date = date('Y/m/d H:i:s',strtotime($order['order_date'])    );

    echo "<div class='order-detail'>
    <span class='detail-label'>Order Date</span>
    <span class='detail-value'>{$date}</span>
    </div>";
    echo "<div class='order-detail'>
    <span class='detail-label'>Order Status</span>
    <span class='detail-value'>{$order['order_status']}</span>
    </div>";
    echo "</div>";
    $region_id = $order["region_id"];
    $dealers_of_region_query="select * from dealer where region_id='{$region_id}'";
    $dealers_of_region=mysqli_query($con, $dealers_of_region_query);
    echo "
    <div class='right-column'>
    <label>Select Dealer</label>
    <form action='allot-dealer.php' method='POST'>
    <select name='dealer_id'>";
```

```
        foreach($dealers_of_region as $dealer) {
            echo "<option
value='{$dealer["dealer_id"]}'>{$dealer["dealer_name"]}</option>";
        }
        echo "</select>
        <input type='hidden' name='order_id' value='{$order['order_id']}' />
        <button type='submit'>Allot Dealer</button>
        </form>
        </div>
        ";

        echo "</div>";
        echo "</div>";
    }
    ?>
</div>
</div>
<?php include "footer.php" ?>
<?php  include "base.php"; ?>
<div class="main-content dealer-allotment">
<div class="orders-list">
    <?php
    $query = "select * from `order`
    inner join customer
    on customer.customer_id=order.customer_id
    where dealer_id IS NULL
    ORDER BY order_date DESC
    ;";

    $orders = mysqli_query($con, $query);
    echo mysqli_error($con);
    if(mysqli_num_rows($orders) == 0) {
        echo "<h1>No orders.</h1>";
        exit(0);
    }
    foreach($orders as $order) {
```

```
echo "<div class='order'>";
echo "<div class='order_id'>Order #{$order['order_id']}</div>";
echo "<div class='bottom-row'>";
echo "<div class='left-column'>";
echo "<div class='order-detail'>
   <span class='detail-label'>Customer ID</span>
   <span class='detail-value'>{$order['customer_id']}</span>
</div>";
$query = "select * from order_details where order_id='{$order['order_id']}'";
$order_items = mysqli_query($con, $query);
$no_of_items = mysqli_num_rows($order_items);
echo "<div class='order-detail'>
<span class='detail-label'>Order Items Count</span>
<span class='detail-value'>{$no_of_items}</span>
</div>";
$date = date('Y/m/d H:i:s',strtotime($order['order_date'])    );
echo "<div class='order-detail'>
<span class='detail-label'>Order Date</span>
<span class='detail-value'>{$date}</span>
</div>";

echo "<div class='order-detail'>
<span class='detail-label'>Order Status</span>
<span class='detail-value'>{$order['order_status']}</span>
</div>";
echo "</div>";
$region_id = $order["region_id"];
$dealers_of_region_query="select * from dealer where region_id='{$region_id}'";
$dealers_of_region=mysqli_query($con, $dealers_of_region_query);
echo "
<div class='right-column'>
<label>Select Dealer</label>
<form action='allot-dealer.php' method='POST'>
<select name='dealer_id'>";
foreach($dealers_of_region as $dealer) {
   echo "<option
```

```
value='{$dealer["dealer_id"]}'>{$dealer["dealer_name"]}</option>";
        }
        echo "</select>
        <input type='hidden' name='order_id' value='{$order['order_id']}' />
        <button type='submit'>Allot Dealer</button>
        </form>
        </div>
        ";
        echo "</div>";
        echo "</div>";
    }
    ?>
</div>
</div>
<?php include "footer.php" ?>
<?php  include "base.php"; ?>
<div class="main-content">
    <?ph
    // form submission code... step 2.
    // when approve or reject button is clicked, below code is invoked
    // if approve is clicked, dealer table is updated with status=approved
    // if reject is clicked, dealer table is updated with status=rejected

    if(isset($_POST['dealer_id'])) {
        $action = $_POST['action'];
        $dealer_id = $_POST['dealer_id'];
        $query = "update dealer set status='{$action}' where dealer_id='{$dealer_id}'";
        $result = mysqli_query($con, $query);
        echo mysqli_error($con);
        $txt = $action === 'approve' ? 'Dealer approved' : 'Dealer rejected';
        $_SESSION["flash_message"] = "<script>
        Swal.fire({
            toast:true,
            timer:2000,
            position:\"top\",
            \"title\":\"{$txt}.\",
```

```
        })
        </script>";
        header('Location: ' . $_SERVER['REQUEST_URI']);
    }
    // display code. step 1.
    // if dealers are there,
    // show dealers list, and show approve / reject buttons
    // else show no dealers to approve
    $query = "select * from dealer where status='pending'";
    $result = mysqli_query($con, $query);
    if($result == false || mysqli_num_rows($result) == 0) {
        echo "<h1>No dealers to approve..<h1>";
    }  else {
        echo "<h1>Dealers waiting for approval:</h1>";
    foreach($result as $row) {
        echo "<div class='dealer'>";
        foreach($row as $field => $val) {
            if($field === "password") continue;
            $field_2 = ucwords(implode(" ",explode("_", $field)));
            echo "<span class='fieldname'>{$field_2}</span><span
class='fieldvalue'>{$val}</span>" ;
        }
        echo "<div>
            <form action='{$_SERVER['REQUEST_URI']}' method='POST'>
            <input type='hidden' name='dealer_id' value='{$row['dealer_id']}' />
            <button type='submit' class='approve-button' name='action'
value='approve'>Approve</button><button type='submit' name='action' class='reject-
button' value='reject'>Reject</button>
            </form>
        </div>";
        echo "</div>";
    }
    }
?>
</div>
<?php include "footer.php" ?>
```

```
<?php include "base.php"; ?>
<div class="main-content">
<div class="feedback-container">
   <?php
$query = "
select dealer.username, customer.full_name, feedback.feedback,feedback.feedback_date
from dealer
left outer join `order`
on order.dealer_id=dealer.dealer_id
inner join `feedback`
on order.order_id=feedback.order_id
inner join `customer`
on feedback.customer_id=customer.customer_id
";
$result = mysqli_query($con, $query);
echo mysqli_error($con);
$dealers = array();
if ($result != false && mysqli_num_rows($result) != 0) {
      while ($row = mysqli_fetch_assoc($result)) {
         $dealers[$row['username']][] = $row;
      }
      foreach ($dealers as $key => $value) {
         echo "<div class='feedback-dealer-section'>";
         echo "<span class='dealer-name'>Dealer: $key</span>";
         foreach ($value as $feedback) {
            $date = date('Y/m/d H:i:s',strtotime($feedback['feedback_date']));
            echo "<div class='feedback-dealer-feedback'>
            <span class='feedback-top-section'>
            <span class='feedback-customer-name' >{$feedback['full_name']}</span>
            <span class='feedback-date' >{$date}</span>
            </span>
            <span class='feedback-body'>{$feedback['feedback']}</span>
            </div>";
         }
         echo "</div>";
      }
```

```
    } else {
        echo "No feedbacks.";
    }
    ?>
</div>
</div>
<?php include "footer.php"; ?>
<header>
    <h3>Sell Gros | Admin</h3>
    <div class="navigation">
        <?php if ($logged_in) { ?>
            <font color='white' size="4">Welcome <?php echo $userDetails["username"];
?></font>
            <a href="/sellgros/logout.php" class="navigation-item">Logout</a>
        <?php } else { ?>
            not logged in should redirect
        <?php } ?>
    </div>
</header>
<?php include "base.php"; ?>
<?php
if (isset($_POST["action"])) {
    $action = $_POST["action"];
    if ($action == "active" || $action == "inactive") {

        $query = "update products
        set product_status='%s'
        where product_id='%s'";
        $query = sprintf($query,$action, $_POST["product-id"]);
        $inserted = mysqli_query($con, $query);
        echo mysqli_error($con);
        $_SESSION["flash_message"] = "<script>
        Swal.fire({
            toast:true,
            timer:2000,
            position:\"top\",
```

```
        \"title\":\"changed product status successfully.\",
      })
      </script>";
      header('Location: product-management.php');
   } else if ($action == "update") {
      if (isset($_POST["product-name"])) {
         if(isset($_FILES["product-image"]) && $_FILES["product-image"]["size"] != 0)
{
            $target_dir = "../../product-images/";
            $target_file = $target_dir . $_POST["product-id"] . ".jpg";
            if (move_uploaded_file($_FILES["product-image"]["tmp_name"],
$target_file)) {
               echo "The file ". htmlspecialchars( basename( $_FILES["product-
image"]["name"])). " has been uploaded.";
            } else {
               echo "Sorry, there was an error uploading your file.";
            }
         }
         $query = "update `products`
         set product_name='%s',
         product_unit='%s',
         price=%f,
         stock=%d,
         product_description='%s'
         where product_id='%s'";
         $query = sprintf($query,$_POST["product-name"], $_POST["product-
unit"],$_POST["product-price"], $_POST["product-stock"], $_POST["product-
description"], $_POST["product-id"] );
         $inserted = mysqli_query($con, $query);
         echo mysqli_error($con);
         $_SESSION["flash_message"] = "<script>
         Swal.fire({
            toast:true,
            timer:2000,
            position:\"top\",
            \"title\":\"updated product successfully.\",
```

```
        })
        </script>";
        header('Location: product-management.php');
    } else {
        $query = "select * from `products` where product_id='{$_POST['product-id']}';";
        $q = mysqli_query($con, $query);
        echo mysqli_error($con);
        $prod = mysqli_fetch_array($q);
        $product_id = $prod["product_id"];
        $product_name = $prod["product_name"];
        $product_price = $prod["price"];
        $product_stock = $prod["stock"];
        $product_unit = $prod["product_unit"];
        $product_description = $prod["product_description"];
?>
        <div class="main-content product-update-page">
            <form method="POST" action="manage-product.php"
enctype="multipart/form-data">
                <input type="hidden" required name="action" value="update" />
                <input type="hidden" required name="product-id" value="<?php echo
$_POST["product-id"]; ?>" />
                <div class="form-field">
                    <img src="<?php echo  $root . "//product-images/" . $product_id .
".jpg?tmp=" . time(); ?>" />
                    <input type="file" name="product-image"  id="product-image"
accept=".jpg,.jpeg" />
                </div>
<div class="form-field">Product Name <input type="text" required name="product-
name" value="<?php echo $product_name; ?>"/></div>
<div class="form-field">Product Price <input type="number" required name="product-
price" value="<?php echo $product_price; ?>" /></div>
<div class="form-field">Product Stock <input type="number" required name="product-
stock" value="<?php echo $product_stock; ?>" /></div>
<div class="form-field">Product unit <input type="text" required name="product-unit"
value="<?php echo $product_unit; ?>" /></div>
<div class="form-field">Product description <textarea name="product-description"
```

```
id="product-description"><?php echo $product_description;?></textarea></div>
                <input type="submit" value="Update Product" />
            </form>
        </div>
    <?php
    }
  }
}
?>
<?php include "footer.php"; ?>
<?php include "base.php"; ?>
<?php
if (isset($_POST["action"])) {
  $action = $_POST["action"];
  if ($action == "active" || $action == "inactive") {
    $query = "update region
    set region_status='%s'
    where region_id='%s'";
    $query = sprintf($query,$action, $_POST["region-id"]);
    $inserted = mysqli_query($con, $query);
    echo mysqli_error($con);
    $_SESSION["flash_message"] = "<script>
    Swal.fire({
      toast:true,
      timer:2000,
      position:\"top\",
      \"title\":\"changed region successfully.\",
    })
    </script>";
    header('Location: region-management.php');
  } else if ($action == "update") {
    if (isset($_POST["region-name"])) {
      $query = "update region
      set region_name='%s',
      pincode=%d
      where region_id='%s'";
```

```
        $query = sprintf($query,$_POST["region-name"], $_POST["pincode"],
$_POST["region-id"] );
        $inserted = mysqli_query($con, $query);
        echo mysqli_error($con);
      $_SESSION["flash_message"] = "<script>
      Swal.fire({
        toast:true,
        timer:2000,
        position:\"top\",
        \"title\":\"updated region successfully.\",
      })
      </script>";
      header('Location: region-management.php');
      } else {
        $query = "select * from region where region_id='{$_POST['region-id']}'";
        $q = mysqli_query($con, $query);
        $reg = mysqli_fetch_array($q);
?>
        <div class="main-content manage-region-page">
          <form method="POST" action="manage-region.php">
            <input type="hidden" required name="action" value="update" />
            <input type="hidden" required name="region-id" value="<?php echo
$_POST["region-id"]; ?>" />
            <div class="form-field">Region Name <input type="text" required
name="region-name" value="<?php echo $reg["region_name"];?>"/></div>
            <div class="form-field">Pincode <input id="pincode" type="text" required
name="pincode" value="<?php echo $reg["pincode"];?>" /></div>
            <input type="submit" value="Update Region" />
          </form>
        </div>
<?php
      }
   }
}
?>
<script>
```

```
    const numberOnlyValidation = [document.querySelector("#pincode")];
for (let field of numberOnlyValidation) {
   field.oninput = function(e) {
      if (field.value.match(/[^$,.\d]/)) {
         field.setCustomValidity("Please enter number only!");
         field.reportValidity();
      } else {
         field.setCustomValidity("");
      }
      field.checkValidity();
   }
   field.onchange = (e) => {
      field.checkValidity();
   }
}
   </script>
<?php include "footer.php"; ?>
<aside>
   <nav>
      <?php
   $tabs = array(

         "dealer-authorization" =>
         array(
               "pages" => array("dealer-authorization.php"),
               "title" => "Dealer Authorization",
               "id" => "dealer-authorization"
            ),
            "dealer-allotment" => array(
               "title" => "Dealer Allotment",
               "pages" => array("dealer-allotment.php"),
               "id" => "dealer-allotment"
            ),
            "product-management" =>
            array(
```

```
                "pages" => array("product-management.php", "manage-product.php","add-
product.php"),
                "title" => "Product Management",
                "id" => "product-management"
            ),
            "region-management" =>
            array(
                "pages" => array("region-management.php", "manage-region.php", "add-
region.php"),
                "title" => "Region Management",
                "id" => "region-management"
            ),
            "feedback" =>
            array(
                "pages" => array("feedback.php"),
                "title" => "View feedback",
                "id" => "feedback"
            )
        );
    $active =  null;
    $temp = basename($_SERVER['REQUEST_URI'], '?' .
$_SERVER['QUERY_STRING']);
    foreach($tabs as $tab) {
        foreach($tab["pages"] as $page) {
            if($page == $temp) {
                $active = $tab;
                break;
            }
        }
    }
    if(!isset($active)) {
        $active = current($tabs);
    }
    foreach($tabs as $key => $value) {
        $tab = $value;
        ?>
```

```
        <div class="nav-item <?php echo $active["id"] == $tab['id'] ? 'active' : ''; ?>"><a
href="<?php echo $value["pages"][0]; ?>"><?php echo $tab["title"];?></a></div>
        <?php
    }
    ?>
    </nav>
</aside>
<?php include "base.php"; ?>
<div class="main-content product-management">
    <div class="product-listing">
        <a href="add-product.php" class="add-product">
            Add Product
        </a>
        <?php
        $query = "select * from products";
        $result = mysqli_query($con, $query);

        $n = mysqli_num_rows($result);
        if ($n != 0) {
            while ($row = mysqli_fetch_array($result)) {
                $product_id = $row["product_id"];
                $product_name = $row["product_name"];
                $product_description = $row["product_description"];
                $product_unit = $row["product_unit"];
                $product_price = $row["price"];
                $product_stock = $row["stock"];
                $change_to = $row["product_status"] == "active" ? "inactive" : "active";
        ?>
                <div class="product">
                    <span class="product_image">
                    <img src="<?php echo  $root . "//product-images/" . $product_id .
".jpg?tmp=" . time(); ?>" />
                    </span>
                    <div class="name_description_container">
                        <span class="product_name">
                            <?php echo $product_name; ?>
```

```
                    </span>
                    <div class="product-description">
                        <?php echo $product_description;?>
                    </div>
                </div>
                <div class="bottom-row">
                    <span class="product_price">
                        Rs. <?php echo $product_price; ?> /  <?php echo $product_unit; ?>
                    </span>
                    <span class="product_modify">
                        <form action='manage-product.php' method='POST'>
<input type='hidden' name='product-id' value='<?php echo $product_id; ?>' />
<button type='submit' class='delete-button' name='action' value='<?php echo $change_to;
?>'>Set as <?php echo $change_to; ?></button>
<button type='submit' class='update-button' name='action'
value='update'>update</button>
                    </form>
                </span>
            </div>
        </div>
        <?php
            }
        } else {
            echo "<h3>No products listed!</h3>";
        }
        ?>
    </div>
</div>
<?php include "footer.php" ?>
<?php include "base.php"; ?>
<div class="main-content">
    <?php
    $query = "select * from region";
    $regions = mysqli_query($con, $query);
    $num = mysqli_num_rows($regions);
    if ($num == 0) {
```

```
    echo "<h3>No regions created.</h3>
    <h3>
    <a href='add-region.php'>Add Region</a>
</h3>
    ";
} else {
?>
    <h3>
        <a href="add-region.php" class='add-region-button'>Add Region</a>
    </h3>
    <div class="regions-table">
        <table class="styled-table">
            <thead>
                <tr>
                    <th>
                        Region Id
                    </th>
                    <th>
                        Region Name
                    </th>
                    <th>
                        Pincode
                    </th>
                    <th>
                        Region Status
                    </th>
                    <th>
                        Actions
                    </th>
                </tr>
            </thead>
            <?php
            foreach ($regions as $region) {
                echo "<tr>";
                foreach ($region as $key => $value) {
                    echo "<td>{$value}</td>";
```

```
                }
                $change_to = $region["region_status"] == "active" ? "inactive" : "active";
                echo "<td>
            <form action='manage-region.php' method='POST'>
            <input type='hidden' name='region-id' value='{$region['region_id']}' />
            <button type='submit' class='delete-button' name='action'
value='{$change_to}'>Set as $change_to</button>
            <button type='submit' class='update-button' name='action'
value='update'>update</button>
            </form>
            </td>";
                echo "</tr>";
            }
            ?>
          </table>
        </div>
    <?php } ?>
</div>
<?php include "footer.php" ?>
```

8 TESTING

8.1 Introduction

Testing is the major quality control measures and during the software development it is used to detect errors that could have occurred during any of the phase like requirement analysis, design, coding. The goal of the testing is to uncover errors in the program.

8.2 Levels of Testing

Testing is done in different levels which includes the following.

- Unit Testing
- Integration Testing
- System testing
- Acceptance testing

  - **Unit Testing**

    In Unit testing each module gets tested during the coding phase itself. The purpose is to exercise the different parts of the module code to detect the coding errors.

  - **Integration Testing**

    After new testing the modules are gradually integrated into sub systems. It is performed to detect design errors by focusing on testing the interconnection between modules.

  - **System Testing**

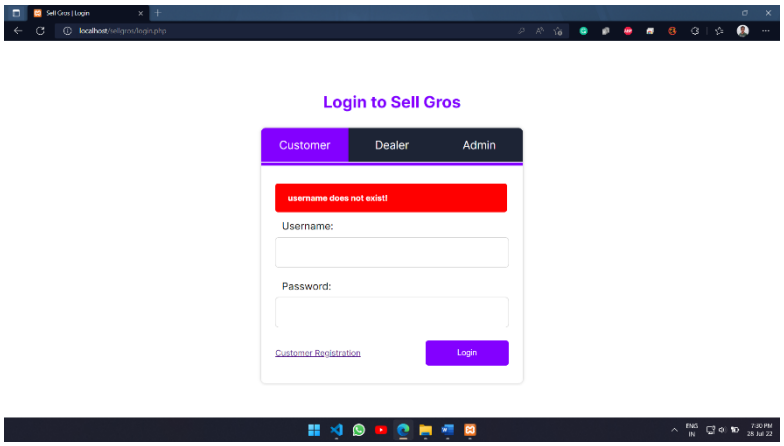
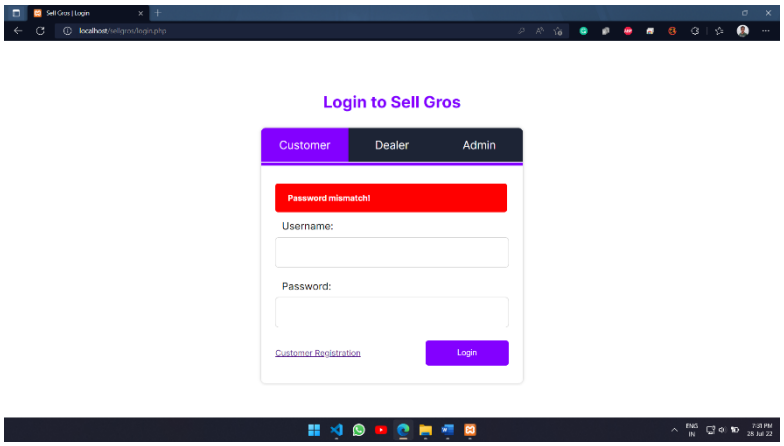
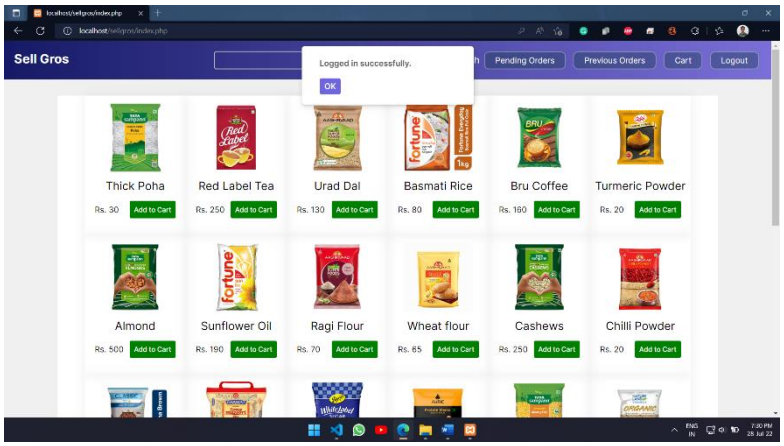
    System is tested against the system requirement if all the requirements are met and if the system performs as specified by the requirement.

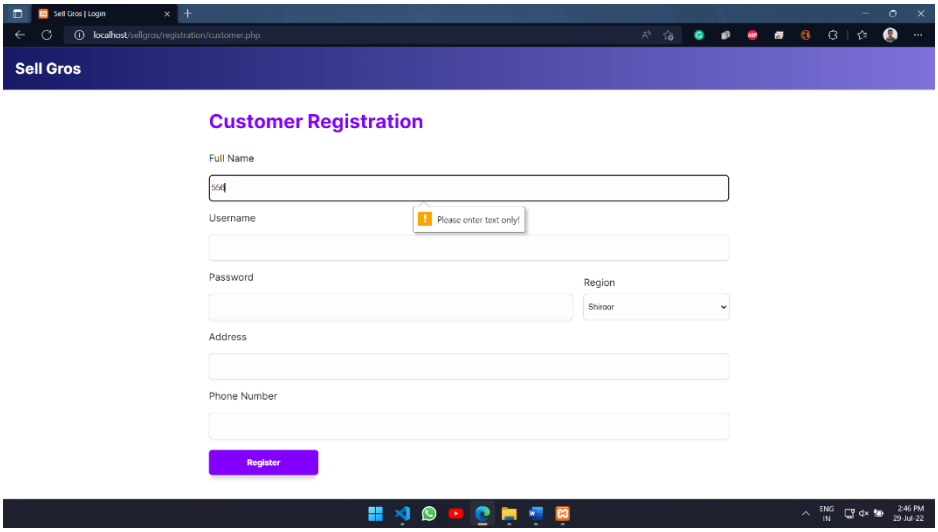
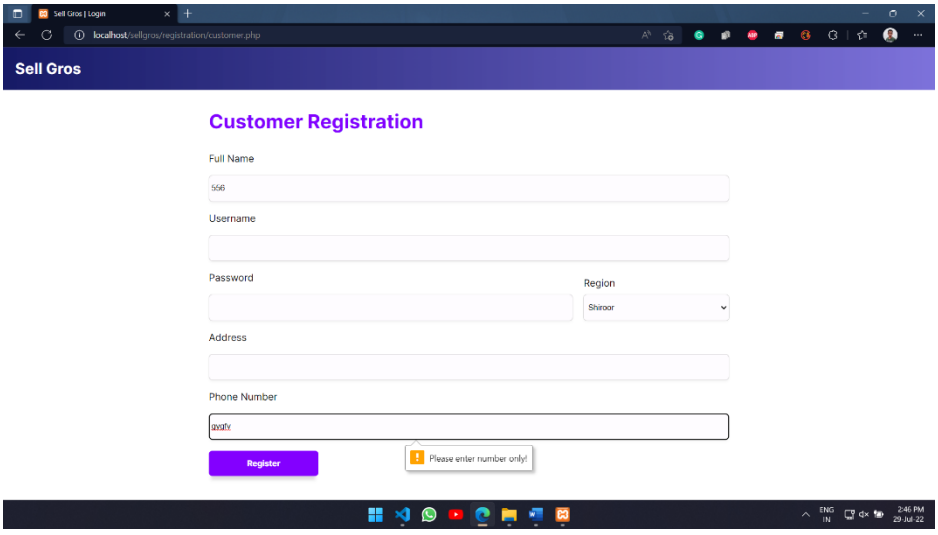
  - **Acceptance Testing**

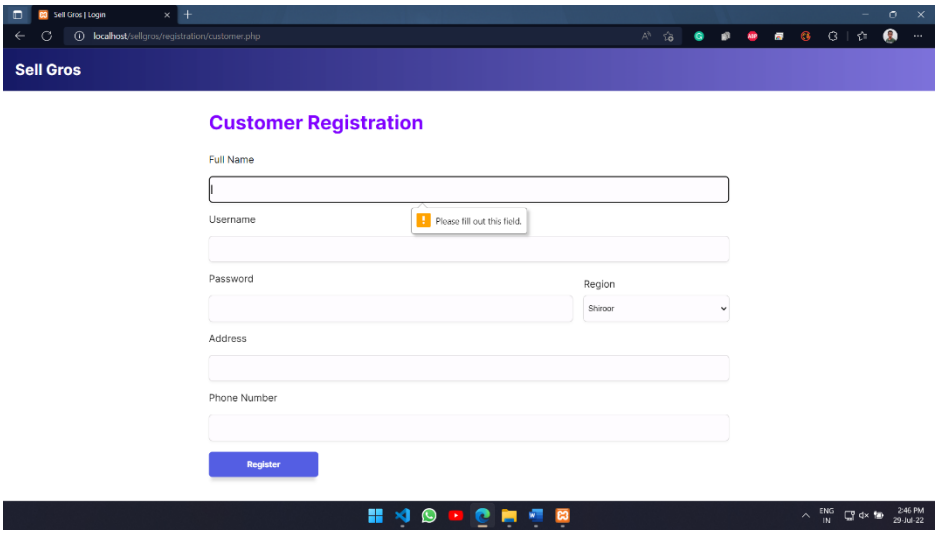
    It is performed to demonstrate to the client on real life data of the client, the operation of the system.

8.3 Test Case

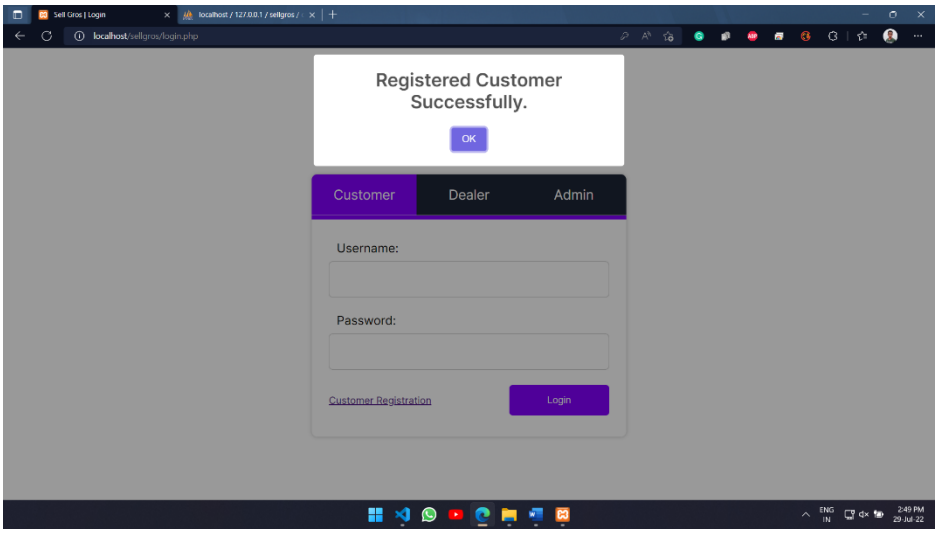
It is the input that tests the genuineness of the program and successful execution of the test case revels. that there are no errors in the program that are under testing. it is a set of conditions or variables under which tester will determine whether an application or software is working currently

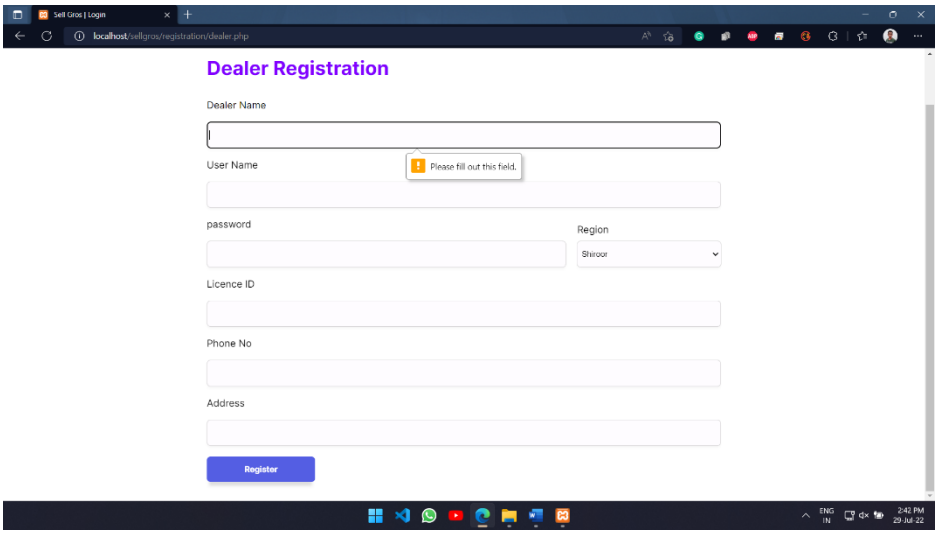
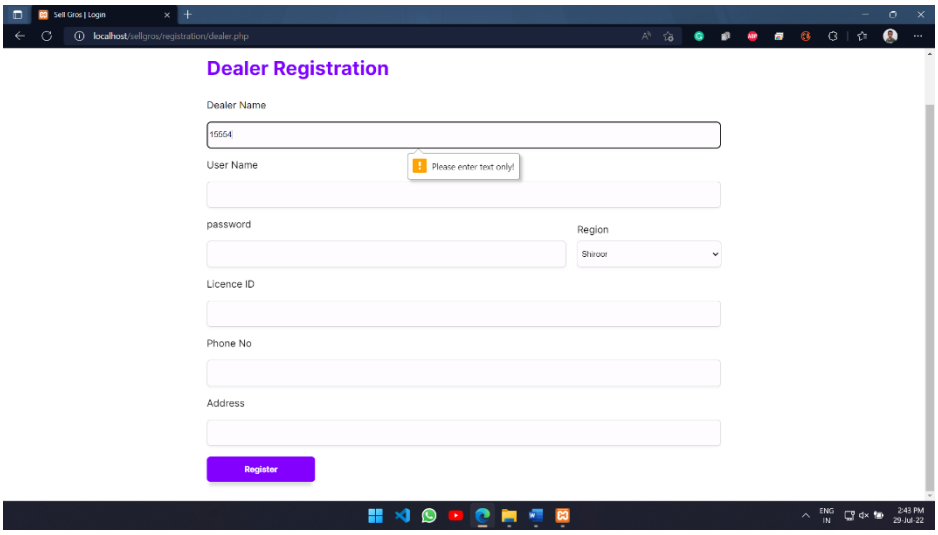
Test case-ID	01
Test Case Title	Login
Purpose of testing	Testing the login
Test data	Username and password
Steps	Step:1 IF username and Password is valid THEN Step:2 DISPLAY Login successful message. Step:3 ELSE DISPLAY Login Unsuccessful message.
Expected Output	Invalid output:   Valid output: 

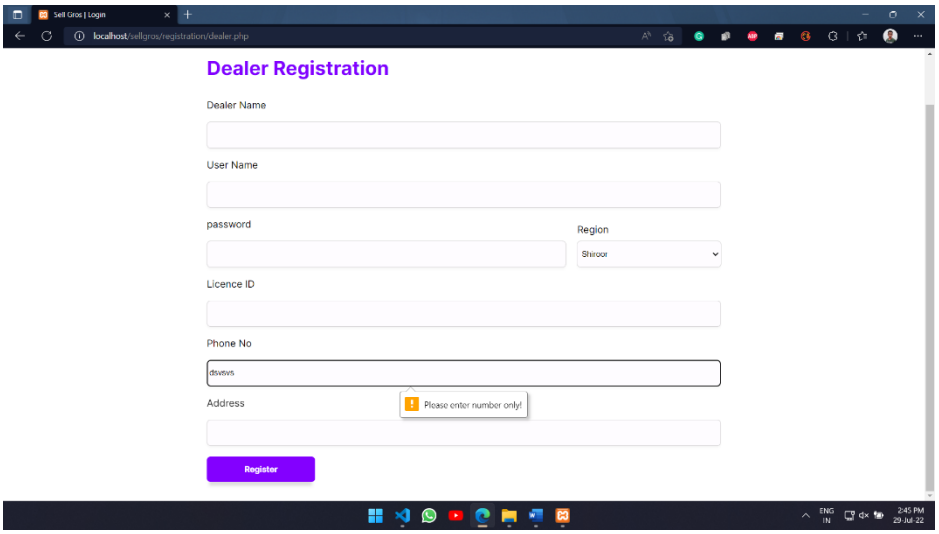
Test case-ID	02
Test Case Title	Registration
Purpose of testing	To register customer for the web-site.
Test data	Full name, Username, Password, Address, Region, Phone number
Steps	1. Click on customer registration. 2. Enter the Full name, Username, Password, Address, Region, Phone number 3. Click on register button 4. If record is valid display registration successful message. 5. Else display warning message with description.
Expected Output	Invalid output:  



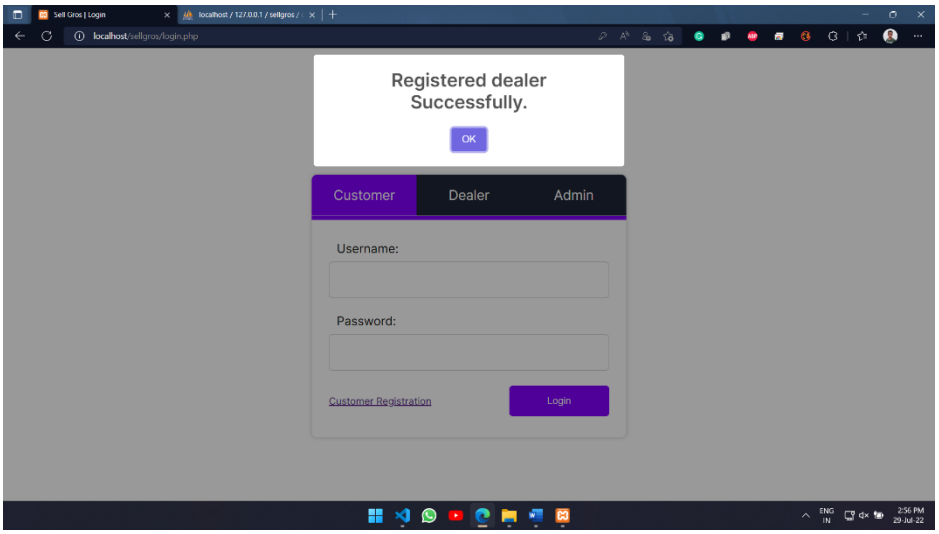
Valid output:

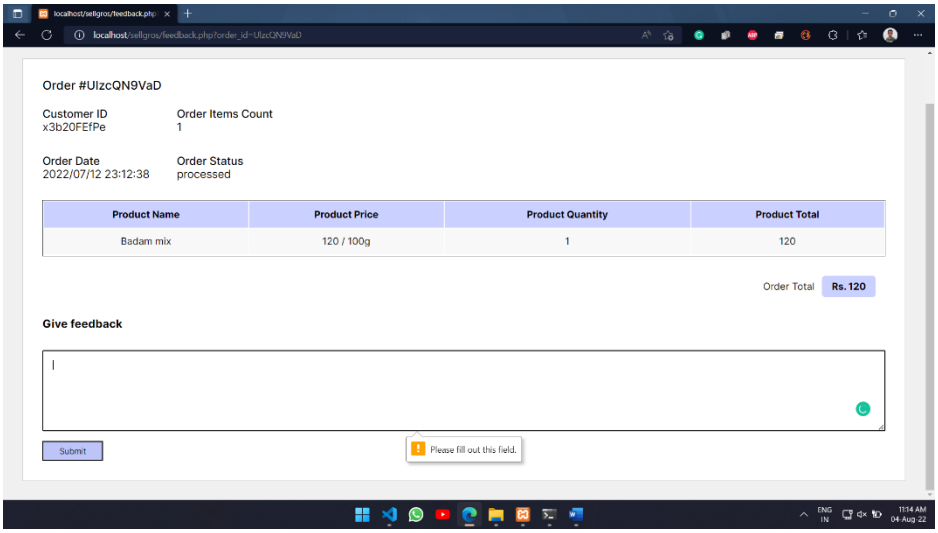
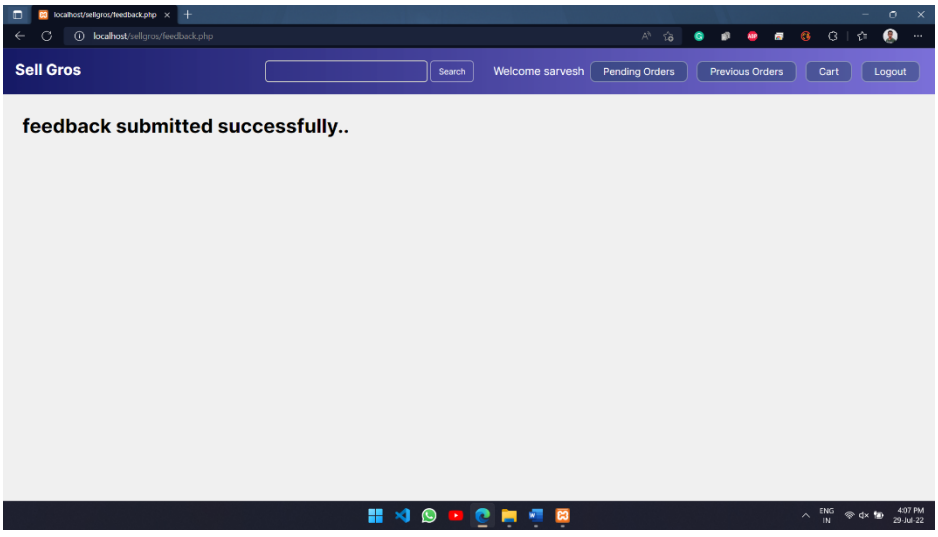


Test case-ID	03
Test Case Title	Registration
Purpose of testing	To register dealer for the web-site.
Test data	Full name, Username, Password, Address, Region, Phone number, License ID
Steps	1. Click on customer registration. 2. Enter the Full name, Username, Password, Address, Region, Phone number, License ID 3. Click on register button 4. If record is valid display registration successful message. 5. Else display warning message with description.
Expected Output	Invalid output:  

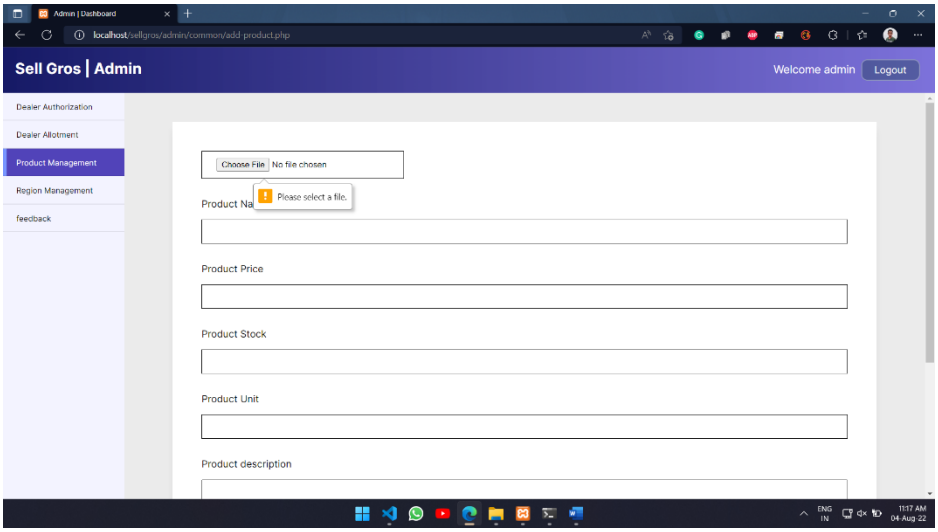


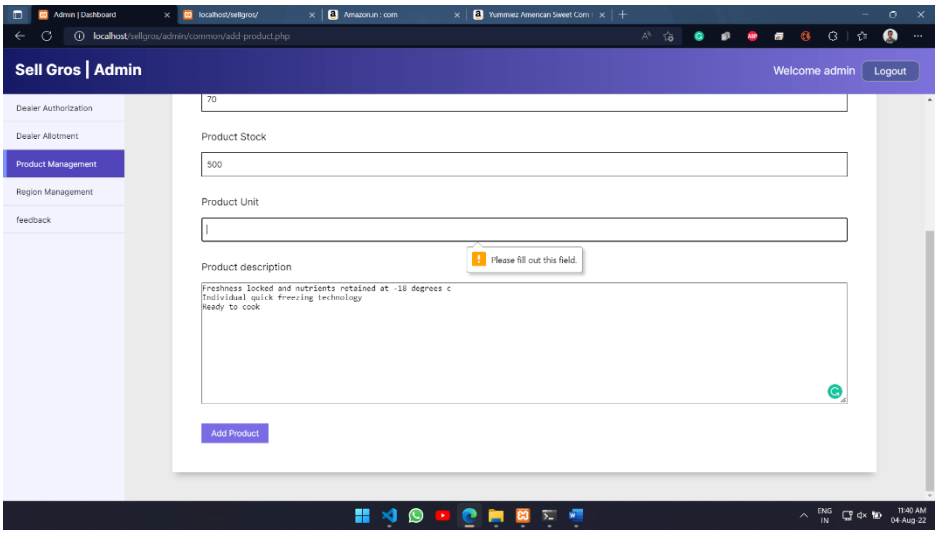
Valid output:



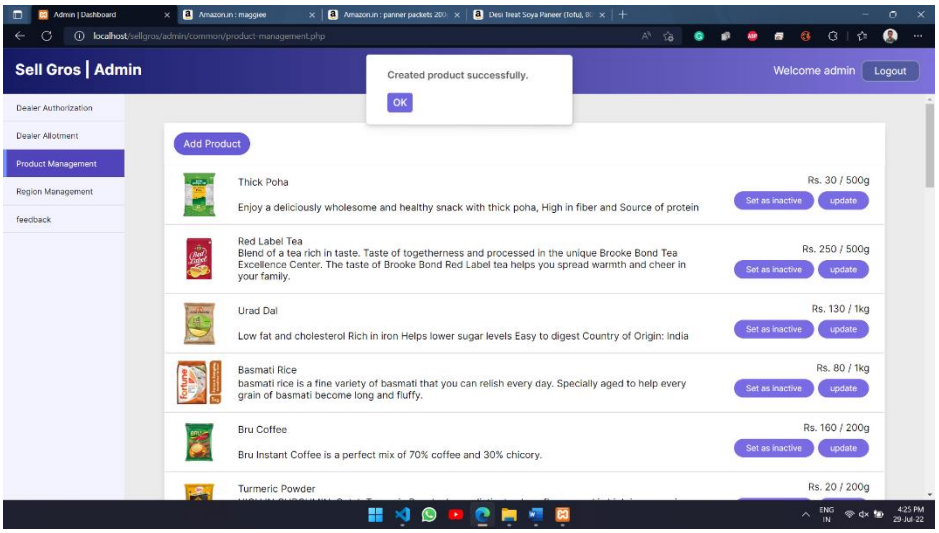
Test case-ID	04
Test Case Title	Feedback
Purpose of testing	To give feedback for the dealer or order.
Test data	Feedback
Steps	1. Click on feedback 2. Enter the feedback 3. Click on submit button 4. If feedback is given display successful message. 5. Else display warning message with description.
Expected Output	Invalid output:  Valid output: 

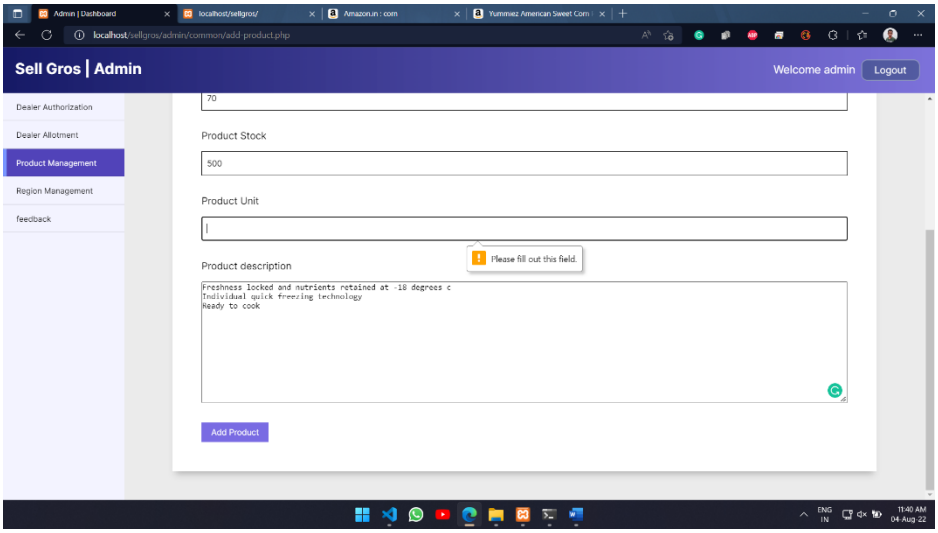
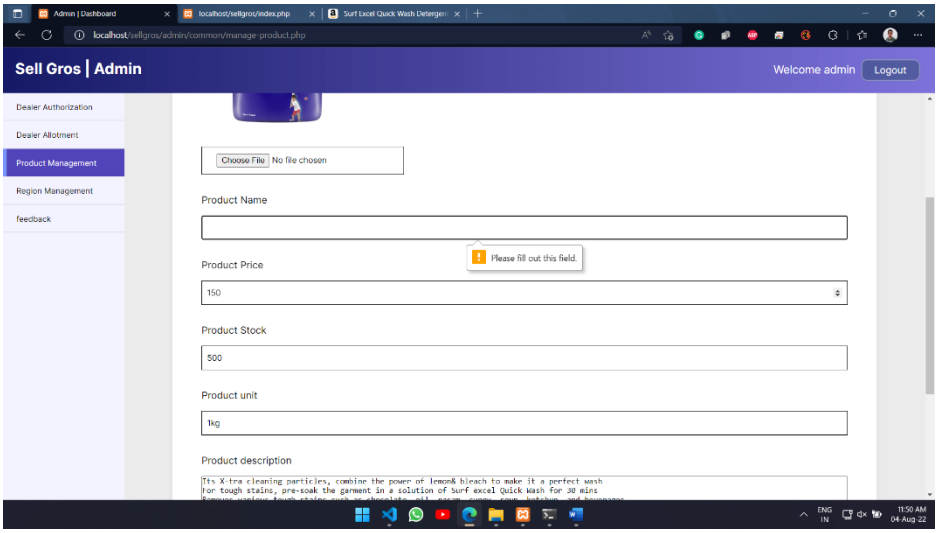
Test case-ID	05
Test Case Title	Add product
Purpose of testing	To add product for website.
Test data	Image, Product name, price, quantity, stock, unit, description.
Steps	1. Click on add product 2. Enter the Image, Product name, price, quantity, stock, unit, description. 3. Click on add product button 4. If product details are added display successful message. 5. Else display warning message with description.
Expected Output	Invalid output:

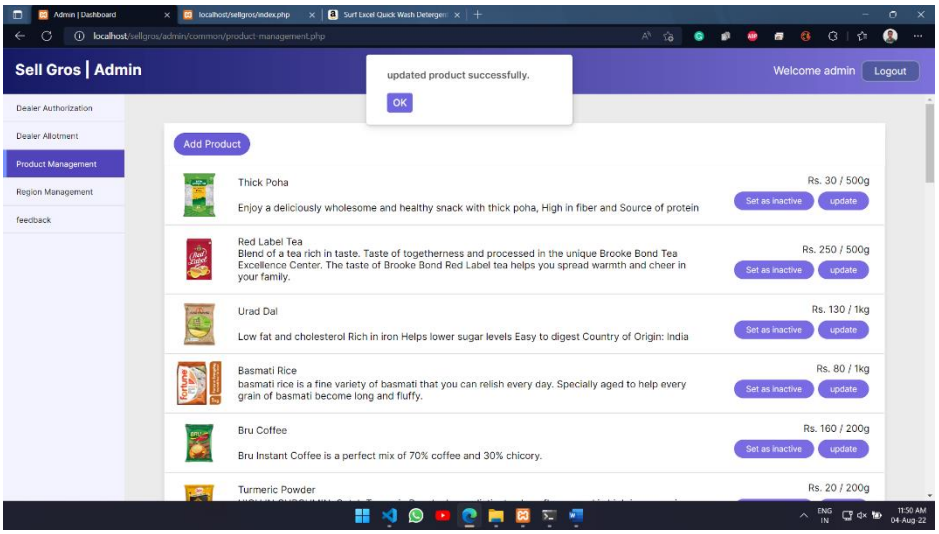


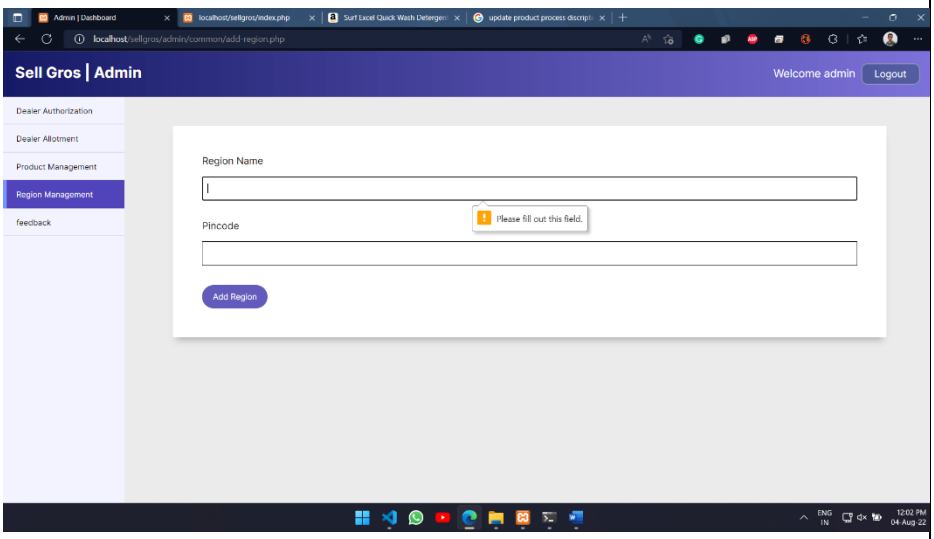
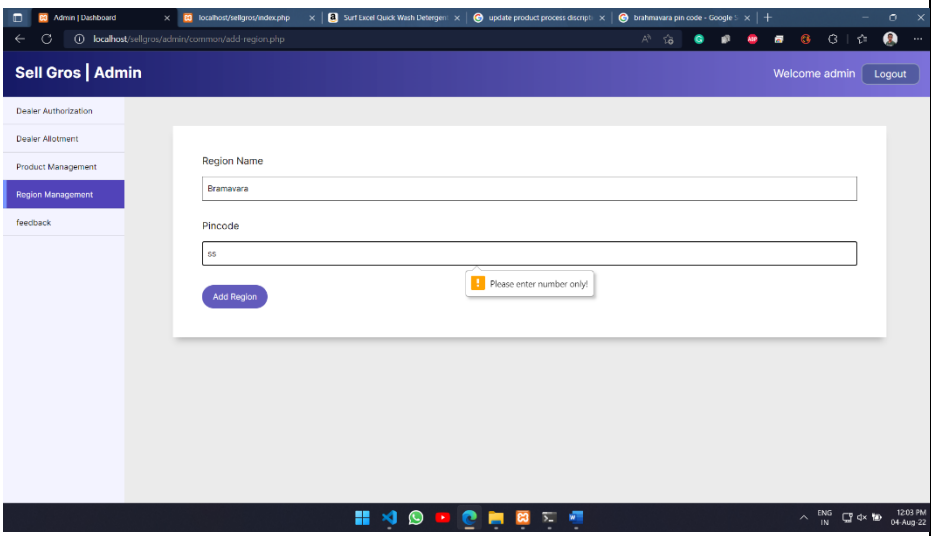


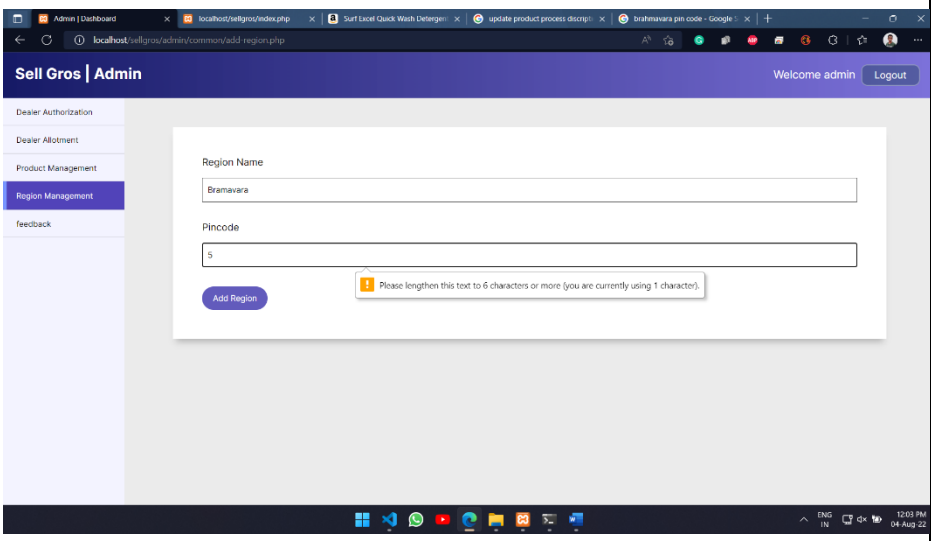
Valid output:



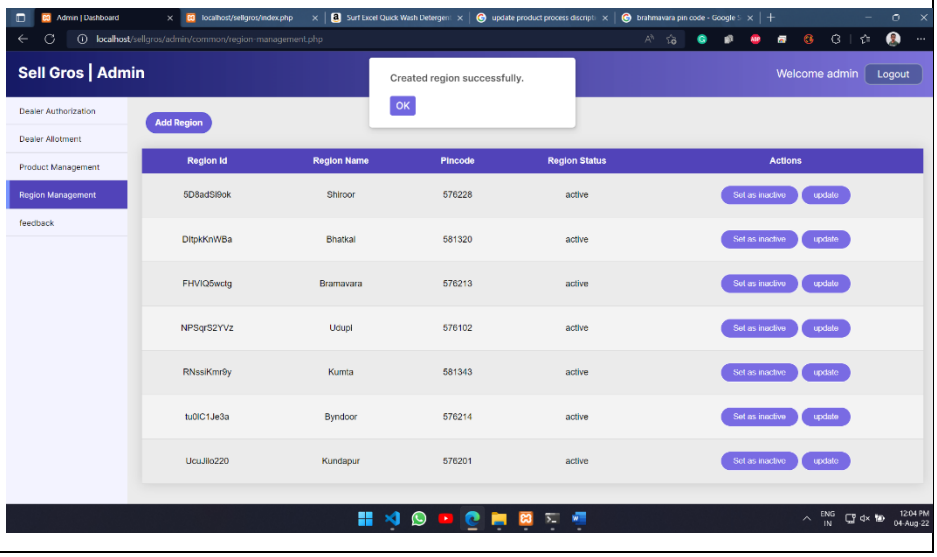
Test case-ID	06
Test Case Title	Update product
Purpose of testing	To change the added product for website.
Test data	Product name/price/quantity/stock/unit/description.
Steps	1. Click on update product 2. Enter the details of the product you want to update. 3. Click on update product button 4. If product details are added display successful message. 5. Else display warning message with description.
Expected Output	Invalid output:  

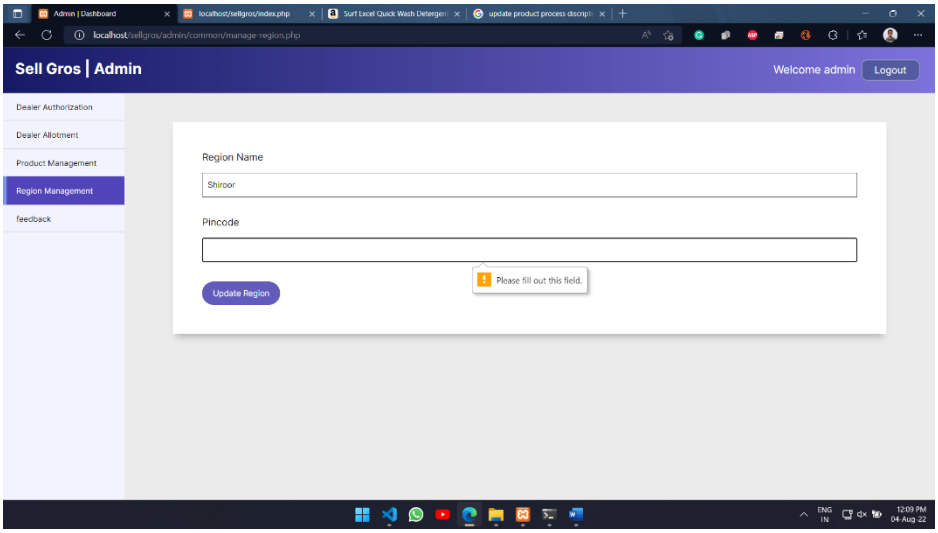
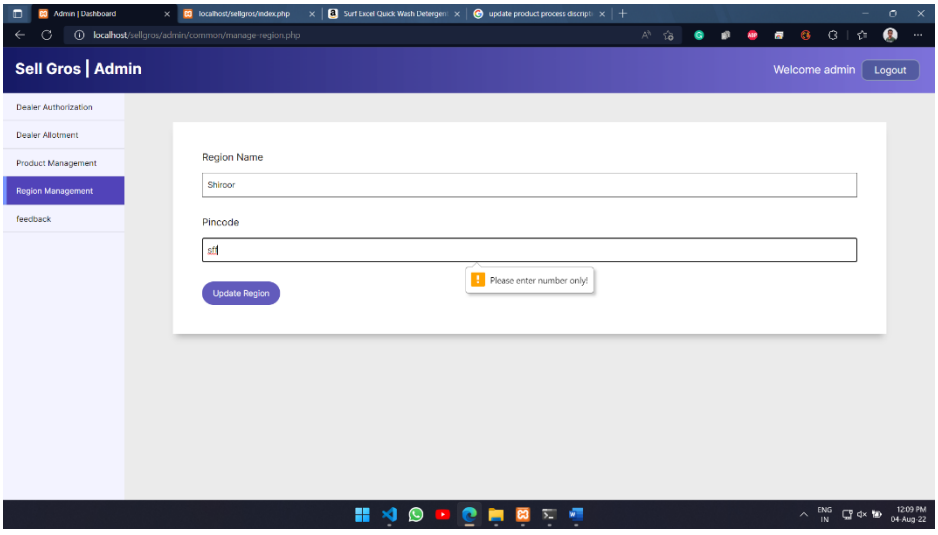
	Valid output: 

Test case-ID	07
Test Case Title	Add region
Purpose of testing	To add region for website.
Test data	Region name and pin code.
Steps	1. Click on add region 2. Enter the region name and pin code. 3. Click on add region button 4. If region details are added display successful message. 5. Else display warning message with description.
Expected Output	Invalid output:  



Valid output:



Test case-ID	08
Test Case Title	Add update region
Purpose of testing	To update region for website.
Test data	Region name/pin code.
Steps	1. Click on update region 2. Enter the region name/pin code. 3. Click on update region button 4. If region details are added display successful message. 5. Else display warning message with description.
Expected Output	Invalid output:  

Valid output: