A Project Report On,

“Galaxy Fashion online shopping”

Submitted to partial fulfilment of the requirements for the degree of

Bachelor of Computer Application (BCA)

MANGALORE UNIVERSITY

BY

GREESHMA P

(Reg No: 202331522166)

ARPITHA V

(Reg No: 202331522154)

Under the Guidance of

Internal Guide
Mrs Sowmya
Dept. of Computer Science
St.Philomena College, Darbe, Puttur

External Guide
Mr Keerthan Shetty
M/S Sanjeeva Shetty Textiles
Puttur

Dept. of Computer Science
St.Philomena College
Philonagar, Darbe, Puttur D.K574202

2022-2023

ST PHILOMENA COLLEGE, PUTTUR

PHILONAGAR, DARBE, PUTTUR, 574202

DEPT OF COMPUTER SCIENCE

CERTIFICATE

Certified that this is a bona fide record of the Project Work entitled "Galaxy Fashion online Shopping "carried out by Greeshma p(201331522166), Arpitha V (201331522154) of IIIBCA during the academic year 2022-23 for the partial fulfilment of the requirement to award a Bachelor of Computer Applications(BCA) degree by Mangalore University.

Sowmya M
Internal Guide

Vinayachandra
Head of the Dept.

Forwarded to the principal for approval

APPROVED

PRINCIPAL

Submitted to the University Examination on_____a St Philomena College, Puttur Examination Centre.

EXAMINERS

INTERNAL

EXTERNAL

CERTIFICATE

This is to certify that the Project Work entitled **"Galaxy Fashion-Online Shopping"** Submitted by Greeshma P (201331522166) Arpitha V (2013315222154**)** has been done under my guidance and supervision during the period 2022-2023 in partial fulfillment of the requirement for the award of a Bachelor of Computer application degree from Mangalore University.

To the best of my knowledge, the work and analysis mentioned in this Project Dissertation have been undertaken by the candidates themselves, and necessary references have been recognized and acknowledged in the text of the report.

This Dissertation is the authentic record of work carried out by them and is not submitted in the past to any Institute or University

Sowmya
Internal Guide
Department of Computer Science
St Philomena College, Puttur

DECLARATION

We hereby declare that the Project Work entitled "**Galaxy fashion –online Shopping"** an original work done by us under the guidance of **Mrs. Sowmya** and **Mr. Keerthan Shetty** during the academic year 2022-23. This dissertation is submitted to the University in partial fulfillment of the requirements for the award of the Bachelor of Computer Applications degree.

We hereby declare that this Dissertation is the authentic record of work carried out by us and is not submitted in the past to any Institute or University

Greeshma p
(Reg no 201331522166)

Arpitha v
(Reg no 2013315222154)

To Whom It May Concern

This is to certify the project entitled **"Galaxy Fashion-Online Fashion"** has been carried out successfully.

Greeshma p
(Reg.no 201331522166)

Arpitha v
(Reg.no 201331522154)

Final year BCA of **St Philomena college puttur** as partial fulfillment of the requirements for the Bachelor of Computer Application (BCA) Degree of the Mangalore University, During the academic year 2022-23.

Date :
Place:puttur

Mr. Keerthan Shetty
Sanjeeva shetty Texttiles
puttur

ACKNOWLEDGEMENT

We would like to sincerely thank all the people who have placed their roles directly or indirectly in this project. Without their help the completion of this project would have been impossible.

We express our hearty gratitude to **Rev. Dr Antony Prakash Montero** Principal, St Philomena College, and Puttur for extending all possible help to complete this project.

A special gratitude to **Mr.Vinaychandra**, HOD of Computer science department for his valuable guidance and also to all other lectures of the Dept. for their support. We also expressour thanks to our College staff for providing the necessary information.

We thank **Mrs. Sowmya M**, Dept. Computer Science who has guided us and provided support and has given genuine replies to our queries. She has not only guided us in the project but also has given suggestions without which we couldn't improve our work. Her suggestions and interest in helping us to complete our project has led us to the successful result of our project.

The project has been successfully completed under the guidance of **Mr Keerthan Shetty, Proprietor of Sanjeeva Shetty Textiles, Puttur** who let us work on a project of thismagnitude and providing us support, help and encouragement in completing this project.

Our special thanks to our dear friends who provided their help, support and genuinefeedbacks that helped us improve at our best.

Finally, we thank our parents, who financially supported throughout the project and provided the support we needed throughout our life.

Date: 13-08-2022 **By,**

Place: Puttur **Greeshma P**

Arpitha V

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 Introduction of the System

"Galaxy Fashion-Online Ready Wears Shop" is a web-based application that provides various kinds of clothing material. In our fast-facing world, everything has been made available right at their doorsteps so this website will be a delivery website, where the User can order online. On our website, multiple shops are login and sell their product online. Users can buy choose shop and order the product. On the home page, the User can view all the services, like the type of clothes available. The User can register and select the product they want to buy. Then, the shop gives this order to the shipping and the status of this order gets updated. The shop can keep on updating the status as and when delivery is done. The User can give feedback for their product.

1.1.1 Project Title

“Galaxy Fashion online shopping”

1.1.2 Category of the System

Client-based application Type:

- Web-based RDMS System
- Deliverables of the project
- Input: Login credentials
- Error message: Unauthorized access.

1.1.3 Overview of the System

Galaxy Fashion Shop Management System is used to manage all the activities related to Galaxy Fashion. Shop owners can sell his/her products to/from the shop. The software will handle the procedure needed to store, sell, and profit loss calculation, store records of sales, and store products. The system is developed to manage small shops. Our product is a comprehensive software solution that simplifies day-to-day operations and increases performance efficiency, designed to increase performance efficiency for its users.

1.2 Background

1.2.1 Background of the company

Galaxy Fashion is used to manage all the activities related to online shopping. It is providing product for 5 years. These systems are available 24 hours a day to users. The software will handle required product details, and add details about products, stores the details about Users and orders.

A shop owner can store and sell his/her products to/from the shop. The main aim of this project providing the best quality products and provide smart security at any time to users. This provides good and efficient information and thus makes the productsmarter**.** The software will handle the procedure needed to store, sell, and profit loss calculation, store records of sales, and store products. The system is developed to manage small shops. Our product is a comprehensive software solution that simplifies day-to-day operations and increases performance efficiency, designed to increase performance efficiency for its users. It delivers a fully integrated approach to shop management with all its business operations built in a single and comprehensive solution. This software is not particularly built for a kind of shop and it is developed to be general.

Client: Keerthan Setty Owner of Sanjeeva Shetty, Puttur D.K-574201.

Address: Main road, near Venkataramana Temple, Puttur D.K-574201.

E-mail: mssanjeevashettyofficial@gmail.com.

Mobile No: 917022264444

1.2.2 Brief Note on Existing System

Traditionally when a user wishes to buy products they have to walk into a particular clothes shop to look at the product before making any decision in a busy world very often Users will go around different shops and compare them which is time-consuming so this website helps the user buy a product in online.

A shop owner can store and sell his/her products to/from the shop. The main aim of this project providing the best quality products and provide smart security at any time to users. This provides good and efficient information and thus makes the productsmarter**.**

1.3 Objectives of the System

- To buy clothes without stepping out.
- To manage the details of stock.
- To manage the details of payment, sales, discounts, and products.
- To assess the staff in capturing the efforts spent on their respective working areas.
- To utilize human resources.
- To generate several types of reports that can be then used for various purposes.
- To make the process more transparent and accurate
- To reduce the use of paper
- To help in reducing the manpower.
- To develop a systematic method.

1.4 Scope of the Project

The main aim of the project is to provide an online facility for the User to order various clothes materials online. Today is the era of computers. This software project slows all the problems experienced in the present system. This project contains three modules. That is Admin, shop, and user.

The main objectives of the project on the shop management system are to manage the details of sales, stocks, company, and inventory; it manages all the information about the store. The project is built at the administrative end and thus only the administrator is guaranteed access.

Traditionally when a user wishes to buy products they have to walk into a particular clothes shop to look at the product before making any decision in a busy world very often Users will go around different shops and compare them which is time-consuming so this website helps the user buy a product in online.

1.5 Structure of the System

Structure is the type of connection between the elements of a whole. It has its internal dialectic. Wholeness must be composed in a certain way; its parts are always related to the whole. The structure of the project online clothes shopping has three main modules. They are Admin, shop, and User.

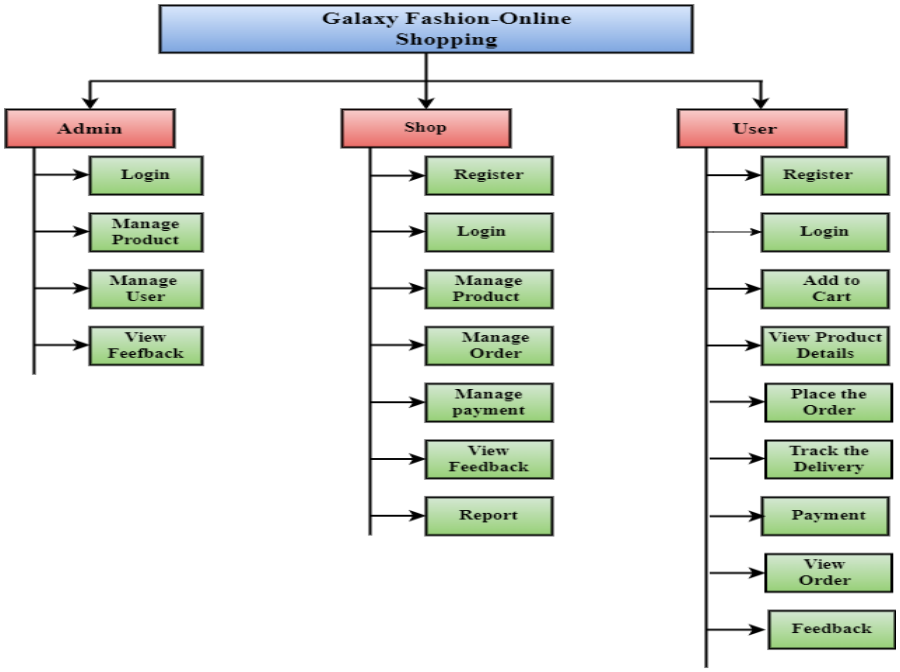

Figure 1.1 Structure of the system

The admin module is responsible for manage in all activities in the project. The shop can manage products, orders, and payments. Users view the products and add the products to the cart so that they can finally purchase the product.

1.6 System Architecture

1.6.1 System Software Architecture

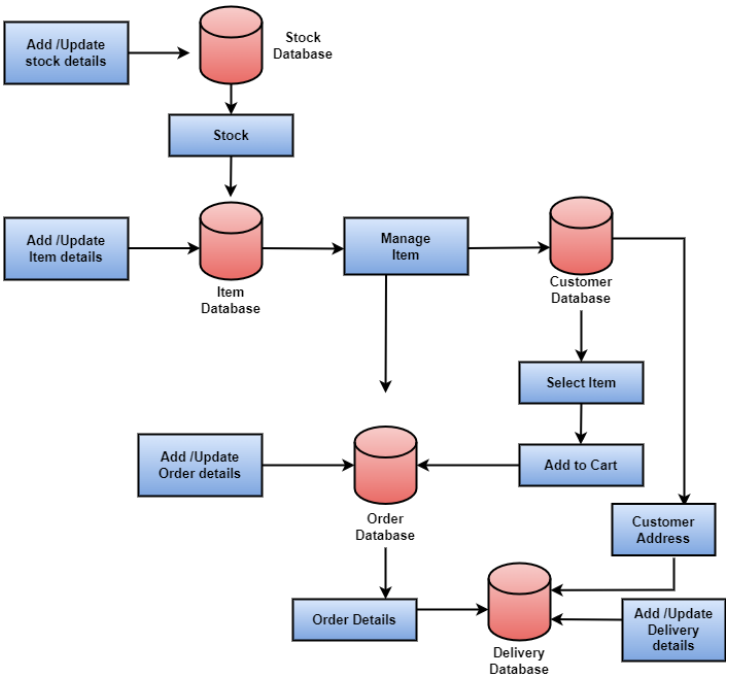

Figure 1.2 System Software Architecture

Software architecture refers to the fundamental structure of a software system and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations.

1.6.2 System Technical Architecture

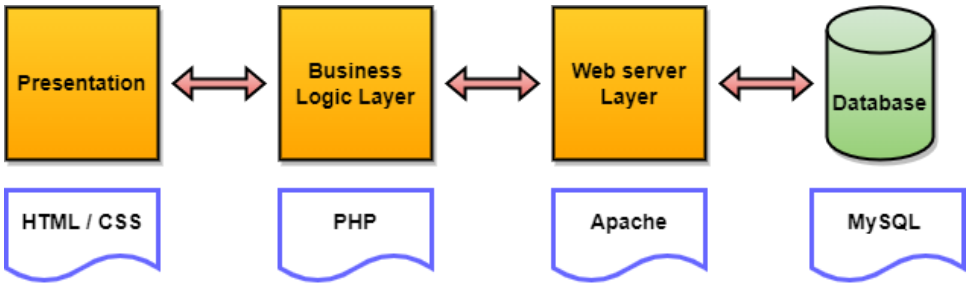

Figure 1.3 System Technical Architecture

Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint concerning the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

The system is made up of four layers:

- **The presentation layer:** includes user interfaces built using the HTML5 and CSS tools. The interface is designed using GUI components which are user-friendly.
- **Business Logic layer:** is used for the retrieval of data obtained from user interfaces or acquired from databases. Activities for calculating, managing, making decisions, and communicating with the front-end application and back-end database are managed by business logic written using PHP language appearing here.
- **Web Server layer:** is to serve websites on the internet. It acts as a middleman between the server and client machines. It pulls content from the server on each user request and delivers it to the web. Apache Tomcat Web Server is used as a Web Server in the system.
- **Database layer:** is used to store data forwarded by the business layer and retrieve data on demand to the business layer. MySQL database software is used as a back-end database of the system.

1.6.3 System Hardware Architecture:

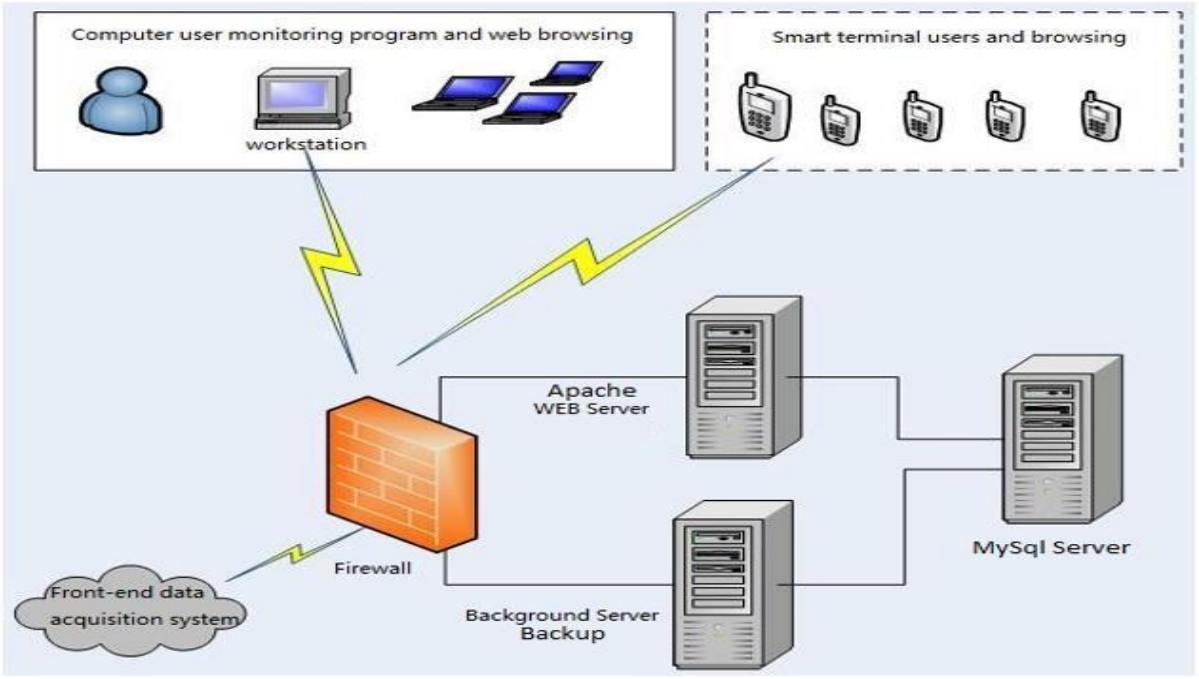

Figure 1.4 System Technical Architecture

The hardware architecture is primarily concerned with the internal electrical interface among the system's components or subsystems and the interface between the external environments, especially the devices operated by the electronic display viewed by a user.

1.7 End User

- Admin
- Shop
- User

1.8 Software / Hardware used for the development

1.8.1 Software Requirements

- Web Technology: PHP 5.4
- Web Components: HTML5/CSS/JavaScript
- Software: XAMPP, Sublime Text3 editor
- Database (Backend): MYSQL server
- Web Server: Apache

1.8.3 Hardware Requirements

- Processor: Intel core i3 or above
- Processor Speed: Minimum 2 GHz
- RAM: 4GB of RAM or Above
- Hard Disk: Minimum 40GB of free space
- Input device: Mouse, Keyboard

1.9 Software / Hardware required for the implementation

1.9.1 Software Requirements

- Front End: HTML, CSS, JAVASCRIPT, PHP.
- Back End: Microsoft SQL Server.
- Database: MySQLi5.0
- Languages: PHP, HTML, CSS
- Operating System: Windows 8 or above.
- Server: APACHE

1.9.2 Hardware Requirements

- RAM: 1GB of RAM or above
- Hard Disk: 10GB or above
- Processor: 2 GHz or above

****

CHAPTER-2

SYSTEM REQUIREMENT SPECIFICATION

2.1 Introduction

Requirements Specification is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analysing the business needs of their clients and stakeholders to help identify business problems and propose solutions.

System specification describes the operational and performance requirements of a system, such as a computer. It is considered a high-level document that dictates global functions. System specifications help to define the operational and performance guidelines for a system. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations.

This Software Requirement Specification document provides a complete description of all the functionalities and the specifications of the "Galaxy Fashion" system. The following section provides an overview of the derived Software Requirements Specification (SRS) for the application. To begin with, the purpose of the document is presented, and its intended audience is outlined. Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

2.2. Overall Description

This section of the SRS describes all general factors of the product and its requirements.

2.2.1 Product Perspective

The project titled "Galaxy Fashion (Online Shopping)" is the software designed for Users to order clothes online. The main aim of the project is to provide productsmart security to the Users. This system provides good and efficient information and thus making the productsmarter. It is an interface for Users to browse the catalogue and order the clothes online. Using the power of internet multiple interest parties can order with complete security and control.

2.2.2 Product Functions

- **Authentication:** User, admin and shop can securely login to the system using their unique credentials.
- **Product Management:** Shop can manage products by updating product information, and categorizing them. Shops can add and update new releases to product and manages orders related to their products.
- **Order Management:** Admin can manage orders placed by users including viewing order details, managing order requests, and updating order status. Shop can manage their product orders and updating order status.
- **Payment Management:** Shop can view payment details made by users, while users can make secure can make secure payments for their orders.
- **Feedback Management:** Shop and admin can view feedback given by users, which can be used to improve the system and user experience.
- **User Management:** Admin can manage user by approving users. User can view book information, filter books, add books to cart, place orders, make payments, and send feedback.

2.2.2 User Classes and Characteristics

Table: 2.1 User Classes and Characteristic

User	Description
Admin	Admin will login to the system. He has separate username and password. He has a dashboard to trace all the activities of the project. The main function of the admin to view all the entities of the project.
User	Users should register with their basic details. They can login with username and password. Once they are logged in to the system then User view the categories, then he/she can make order, then track the delivery, pay the money and buy the products.
Shop	Shop should register with their basic details. They can login with username and password. Once the login process is done the Shop can modify his profile. His main function is to supply the clothes.

2.2.4 General Constraints

Any user can use this software. The general constraints include the following:

- Database is password protected
- Should use less RAM and processing power.
- Every User has individual ID and password.
- Only administrator can access the whole system.

2.2.5Assumptions

- Each user has public account.
- There is only one Administrator.
- Proper browser must be installed.
- Data should be properly connected to the browser.

2.3 Special Requirements (Software/ hardware)

2.3.1. Software Requirements

- Operating system: Windows.
- Text editor: Sublime Text 3
- Language: PHP
- Server: Apache
- User interface: HTML, CSS, JavaScript
- Database: MySQL
- Browser: Chrome, Mozilla Firefox, or any other browsing application.

2.3.2 Hardware Requirements

- Processor: Intel Pentium dual-core or above
- Processor Speed: 2GHz
- RAM – 1GB
- Hard Disk – Minimum 40 GB

2.3.3 User Interface

The user interface is designed in a user-friendly manner. The user will operate interface with mouse and keyboard. The user interface shall be web based, allowing users to remotely access the system via applications. Users will able to use the software through applications such as Microsoft internet Explorer, Mozilla, and Google Chrome, Brave etc. Each part of the user interface intends to be as user friendly as possible.

- This system provides good graphical interfaces.
- The fonts and buttons used will be intended to be very fast and easy to load on web pages.
- The pages will be kept light in space so that it won't take a long time for the page to load.
- This system provides well positioned search bars, user friendly navigation, and smooth check out.
- System provides attractive and well defined interface to users.
- Appropriate error messages are generated when a user performs an operation which is invalid.
- The user needs to enter valid username and password, if both are valid then the login is successful and users are allowed to enter into the next page provided.
- Text boxes are used to enter the input from the User. Software components for which user interface needed as follows:
  - Admin
  - User
  - Shop

Admin Login

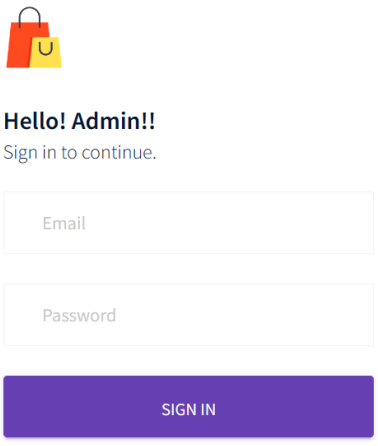

Figure: 2.1 Admin login

Home page

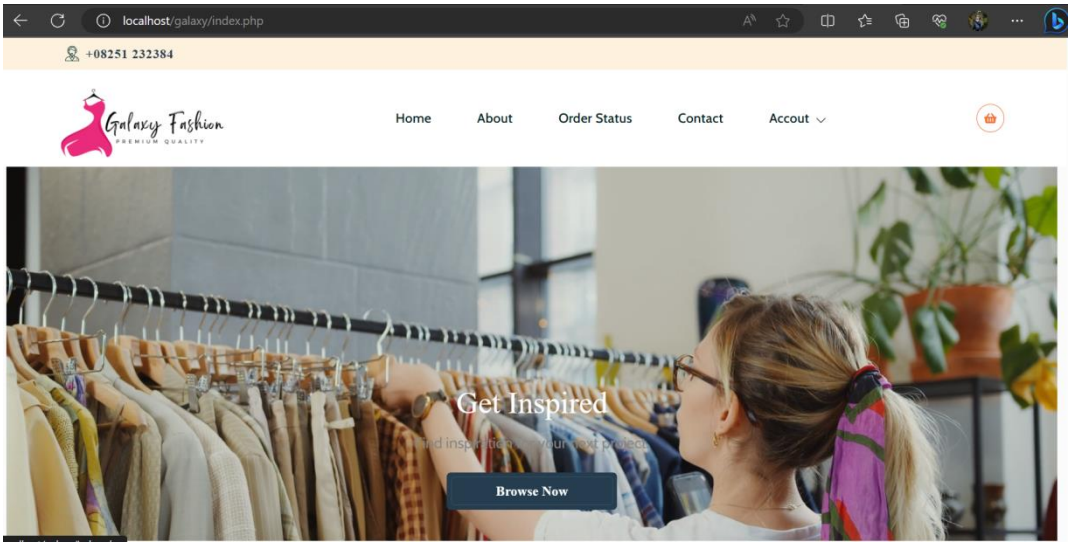

Figure: 2.2 Home pages

Admin Dash board

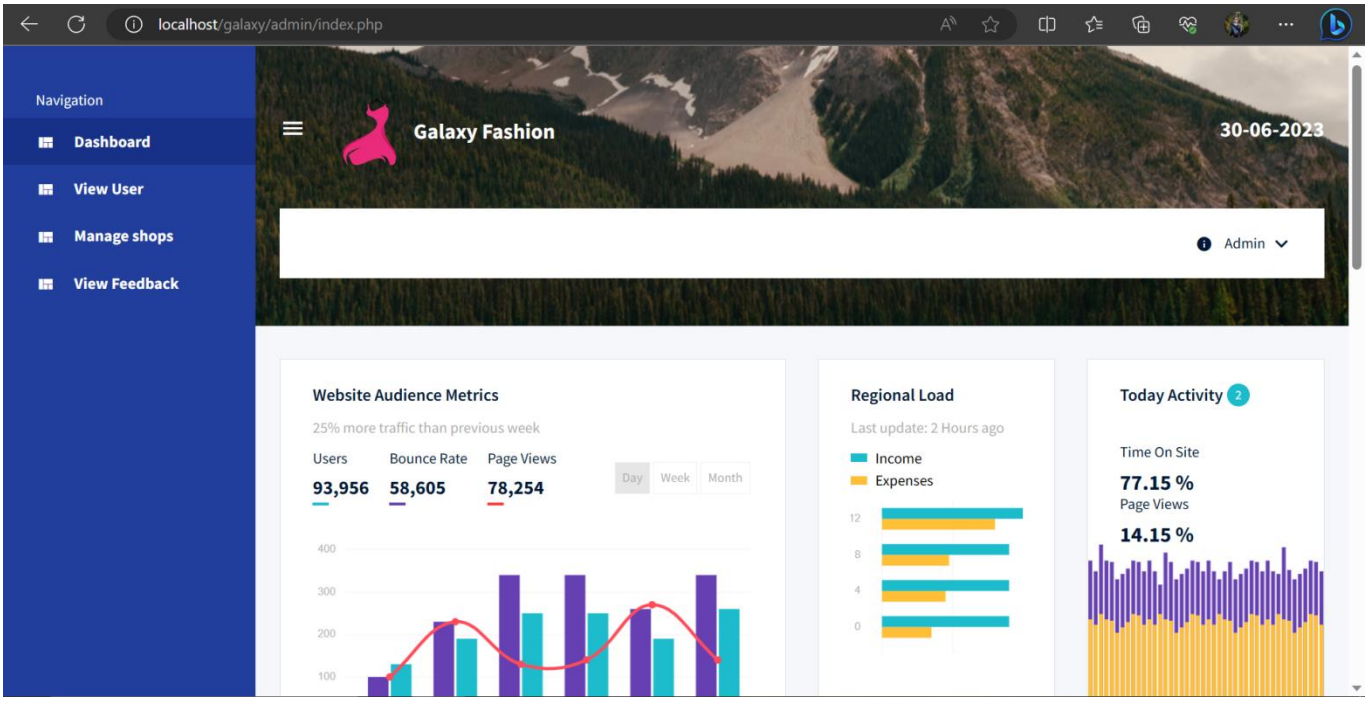

Figure: 2.3 Admin Dash board

User Register

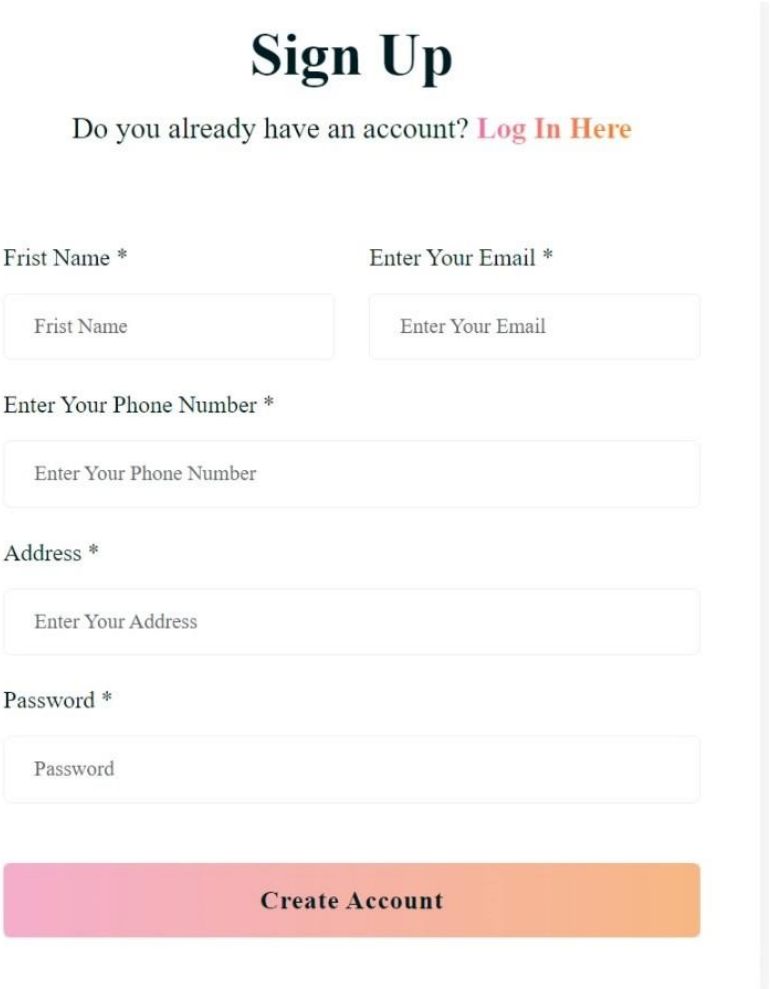

Figure: 2.4 User Registrations

Shop login

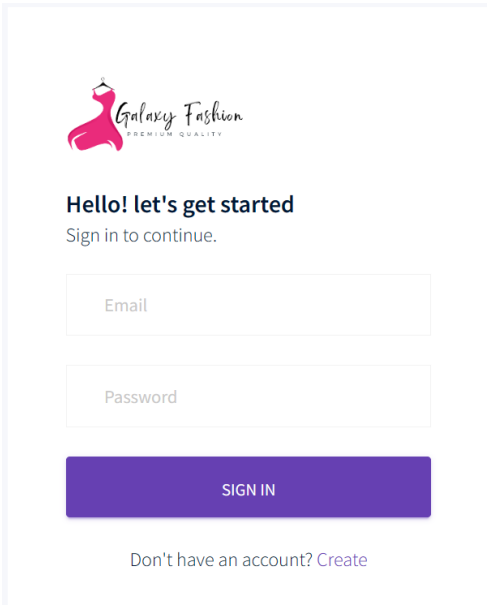

Figure 2.5: Shop login

2.4 Functional Requirements

2.4.1 Admin

- **Login**

  Function: To get the whole database of the user, shop details and manage them.

  Input: Admin need to have Username and password.

  Output: It displays Admin homepage.

- **Manage User**

  Function: In this module admin can manage the user details that are registered to the system.

- **View Feedback:**

  Function: In this module admin can view the feedbacks from user.

- **Manage Shop**

  Function: In this module admin manages the shop login and shop details Admin can be able to accept or reject the shop.

2.4.2 Shop

- **Registration**

  Function: shop has to register by providing primary details before add shop.

  Input: shop required to have owner name, shop name, address, email, password and logo or photo of shop.

  Output: shop login page will be displayed.

- **Login**

  Function: Shop have to login to add shop in website

  Input: shop required to have email and password.

  Output: shop homepage will be displayed.

- **Manage product**

  Function: shop can manages product like add product, add categories, view product etc

- **Manage Payment**

  Function: shop owner can view the user payment details and payment method.

- **Manage order**

Function: shop owner can accepted the product order and updated the shipping details of order.

View Feedback: In this module, the shop can view the feedback from the user.

2.4.3 User

- **Login**

  Function: User have to login to place an order.

  Input: User required having Username and password.

  Output: Users homepage will be displayed.

- **View product details**

  Function: In this module, User can view the products available and the price details.

- **Add to cart**

  Function: User can add the selected items to cart.

- **Place Order**

- Function: User can order the items added to their cart.

- **Track the delivery**

- Function: Once the order is made the User can track the delivery.

- **Payment**

  Function: User can pay for the orders through an online or offline method.

- **View order**

  Function: Here User can view his order details.

- **Feedback:**

  Function: Here User can give feedback.

2.5 Design Constraints

- All the inputs should be checked for validation and messages should be given for the improper data. The invalid data are to be ignored and error messages should be given.
- While adding the product details to the system, mandatory fields must be checked for validation whether the admin has filled appropriate data in these mandatory fields. If not, proper error message should be displayed or else the data is to be stored in database for later retrieval.
- All mandatory fields should be filled by admin, while adding the user detail into the database.

2.6 Software Attributes

The Quality of the website is maintained in such a way so that it can be very user friendly to all the users of the website.

- **Reliability:** Good validation of user inputs will be done to avoid entering incorrect username and password.
- **Availability:** The system shall be available all the time.
- **Security:** Each time there is a security violation; System restricts the user from accessing that function.
- **Maintainability:** The ability to maintain, modify information and update fix problems of the system.
- **Portability:** This system can be run in any operating system and browser.
- **Accessibility:** Administrator and many other users can access the system but the access 25 level is controlled for each user according to their work scope.

2.7 System Features

2.7.1 Manage Product

The Manage in product can login to the system using. He can add the details of the flowers.

2.7.1.1 Description and Priority

The Manage in product can login to the system using. Admin can add the details of the product.

- **Priority Type** : High
- **Priority Level** : 1

2.7.1.2 Stimulus / Response sequence

- Stimulus: Shop requests for Login Page.
- Response: Login page is displayed.
- Stimulus: Shop enters Username and Password and Clicks on Login Button.
- Response: Shop dashboard is displayed if Username and Password is correct or displays an error message if they are wrong.
- Stimulus: Shop clicks on the add product option.
- Response: clothes details and category are displayed.
- Stimulus: Admin clicks on the add item option.
- Response: System provides a form to enter the item details.

- Stimulus: shop clicks on submit button.
- Response: The item added successfully.
- Stimulus: shop clicks view product details.
- Response: product details are displayed.
- Stimulus: shop clicks to Logout button.
- Response: Database connection is terminated.

2.7.1.3 Functional Requirements

Table 2.2: Functional Requirements of shop

Feature Id	Feature Name	Description
FR-F1	Login	The page allows the shop to login to the system.
FR-F2	Manage Product	This allows owner to add a new products and category of product.
FR-F3	Manage Payment	This module is used to manage payment of ordered product.
FR-F4	Manage Order	This module is used to manage the Order.
FR-F5	View Feedback	This module is used to View the Feedback.

2.7.2 Manage User

In this module admin can manage the user details that are registered to the system.

2.7.2.1 Description and Priority

User can create an account for them, login through it. user will make an order for a product through a website.

- Priority Type: high
- Priority level: 4

2.7.2.2 Stimulus/Response Sequences

- Stimulus**:** User request to create an account.
- Response**:** System will display a form to enter registration details.
- Stimulus: User request for login.
- Response: System will display login page.
- Stimulus: User views the product and added to the cart.

- Response: System will display the cart details.
- Stimulus: User request to view product details.
- Response: System will display product details.
- Stimulus: User request to view order details.
- Response: System will display order details.
- Stimulus**:** Admin enters username and Password and Clicks on Login Button**.**
- Response**:** System connects to the database.
- Stimulus**:** User will request to order for product.
- Response: System will provide product details.
- Stimulus**:** User clicks to logout button.
- Response**:** Database connection is terminated.

2.7.2.3 Functional Requirements Table

Table 2.3: Functional Requirements of user

Feature Id	Feature Name	Description
FR-F1	Register	The page allows the User to register the system
FR-F2	Login	The page allows the User to login to the system.
FR-F3	Add to cart	This page allows the Users to product add to cart.
FR-F4	View product details	This page allows the Users to view product's details.
FR-F5	View orders	This page allows the Users to manage order details
FR-F6	Payment Process	This will allow User to do payment for the product

2.7.3 Add to cart

User can add the selected items to cart.

2.7.3.1 Description and Priority

User view the product and then they wishes to buy the product they can added to the cart.

- Priority Type: high
- Priority level: 2

2.7.3.2 Stimulus/Response Sequences

- Stimulus: User can view the product.
- Response: User can select and added to the product.
- Stimulus: User can view the product in cart.
- Response: User can add and remove the product.

2.7.3.3 Functional Requirements Table

Table 2.4: Functional Requirements of add to cart

Feature Id	Feature Name	Description
FR-F1	Add(s)	User will add to the cart.
FR-F2	Delete(s)	The application shall allow the user to delete the product in the cart.
FR-F3	View(s)	The user can view the cart and product.

2.7.4 Order

Shop can view the orders made by the user. User can order the product.

2.7.4.1 Description and Priority

User's makes an order to get a product.

- Priority Type: high
- Priority level: 3

2.7.4.2 Stimulus/Response Sequences

- Stimulus: User makes an order for a product.
- Response: User will be getting the product.
- Stimulus: User request to cancel the order.
- Response: User order is removed from the system.
- Stimulus: User request to view the order.
- Response: System will display order details.
- Stimulus: User request to cancel the order.
- Response: User order is removed from the system

2.7.4.3 Functional Requirements Table

Table 2.5: Functional Requirements of order

Feature Id	Feature Name	Description
FR-F1	Add(s)	user will make an order.
FR-F2	Delete(s)	The application shall allow the user to delete the order.
FR-F3	View(s)	The user can view their order details.

2.7.5 Payment

The user provides payment information to complete the transaction during checkout process

2.7.5.1. Description and Priority

User's makes a payment for an ordering product.

- Priority Type: high
- Priority level: 2

2.7.5.2 Stimulus/Response Sequences

- Stimulus: User makes a payment for a product.
- Response: System will update the transaction, if payment successful it displays a payment details, otherwise it display transaction will unsuccessful.

- Stimulus: User request to view the payment details.
- Response: System will display payment details.
- Stimulus: User request to cancel order and refund the payment.
- Response: System will display cancelled payment and refunded.

2.7.5.3 Functional Requirements Table

Table 2.6: Functional Requirements of payment

Feature Id	Feature Name	Description
FR-F1	Add(s)	User will make the payment
FR-F2	View(s)	The User can view their payment details

2.7.6 Feedback

Admin and shop can view the feedbacks from user. User and shop review the product and the system.

2.7.6.1. Description and Priority

User gives the feedback to the admin and shop.

- Priority Type: high
- Priority level: 4

2.7.6.2. Stimulus/Response Sequences

- Stimulus: User submits the feedback.
- Response: This feedback sent to the admin.
- Stimulus: Shop submits the feedback.
- Response: This feedback sent to the admin
- Stimulus: User deletes the feedback.
- Response: feedback is deleted.
- Stimulus: Shop deletes the feedback.
- Response: feedback is deleted.
- Stimulus: Admin will request to see the feedback.
- Response: feedback will display.

2.7.6.3. Functional Requirements Table

Table 2.7: Functional Requirements of feedback

Feature Id	Feature Name	Description
FR-F1	Add(s)	User will make a feedback.
FR-F2	Delete(s)	The application shall allow the user to delete the feedback.

2.8 Other Requirements

2.8.1 Performance Requirements

- **Response time:** The system will give responses within 1 second after checking the User information and other information.
- **Capacity:** The system must support 100 people at a time.
- **User interface:** User interface screen will response within 5 seconds.
- **Conformity:** They must conform to Microsoft accessibility.

2.8.2 Safety Requirements

- The system shall log every state and state change of every surface computer, tablet and display to provision recovery from system failure.
- If the caps lock is on it must be notified.
- The system shall be capable of restoring itself to its previous state in the event of failure (e.g. a system crash or power loss).
- The system shall be able to display a menu at all times to facilitate manual order taking should the need arise.
- Authorization: Checking for the entity and provide features for them.

2.8.3 Security Requirements

- System is wirelessly networked with an encryption.
- Database is password protected.
- Each user should have individual username and password.

- Only administrator can access the whole system.

2.8.4 Software Quality Requirements:

The Quality of the website is maintained in such a way so that it can be very user friendly to all the users of the website.

- **Reliability:** Good validation of user inputs will be done to avoid entering incorrect username and password.
- **Availability:** The system shall be available all the time.
- **Security:** Each time there is a security violation; System restricts the user from accessing that function.
- **Maintainability:** The ability to maintain, modify information and update fix problems of the system.
- **Portability:** This system can be run in any operating system and browser.
- **Accessibility:** Administrator and many other users can access the system but the access 25 level is controlled for each user according to their work scope.

****

CHAPTER-3

SYSTEM DESIGN AND DESCRIPTION

3.1 Introduction:

System design is the process of defining the architecture, modules, interface, and data for a system to satisfy specified requirements. Systems designs could be seen as the application of system theory to product development. System design implies a systematic approach to the design of a system. It may take a bottom up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created from the architecture to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system.

The goal of the design process is to produce is to produce a module representation of a system, which can be used later to build a system. System design is the process of defining the architecture, modules, interface, and data for a system to satisfy specified requirements. Systems design could be seen as the application of system theory to product development. It is a process by which the software requirements are translated into a representation of software component, interfaces, and data necessary for the implementation phase.

3.2 Assumption and Constraints:

3.2.1 Assumption:

- Each User must have a Username and password.
- There is only one Administrator.
- Server must always run under windows system.
- Proper browser should be installed.
- Data should be property connected to the browser.

3.2.2 Constraints:

- Database is password protected.
- Should use less RAM and processing power.
- Each user should have individual Id and password.
- Only administrator can access the whole system.

3.3 Functional Decomposition-SYSTEM DESCRIPTION:

The functional decomposition is a term which is used to describe a set of steps in which they break is breakdown the overall function of a device, system or process into its smaller parts. Functional decomposition refers broadly to the process of resolving a functional relationship into its constituent parts in such a way that the original can be reconstructed from those parts by function components. In general, this process of decomposition is undertaken either for the purpose of gaining insight into the identity of the constituent component (which may reflect individual physical process of interest) or for the purpose of obtaining a compressed representation of the global function, a task which is feasible only when constituent processes a process a certain level of modularly (i.e., independence or non-interaction).

3.3.1 System Software Architecture:

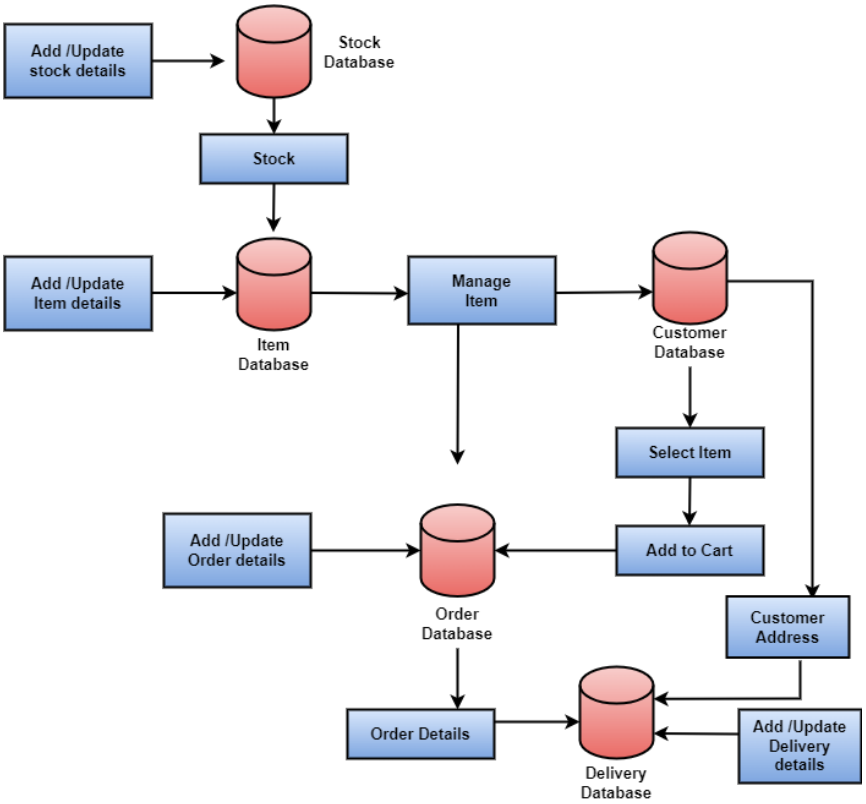

Figure 3.1: System software architecture

The software architecture refers to the fundamental structure of a software system and discipline of creating such structure and systems. Each structure comprises software elements, relations among them and properties of both elements and relations.

3.3.2 System Technical Architecture:

Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

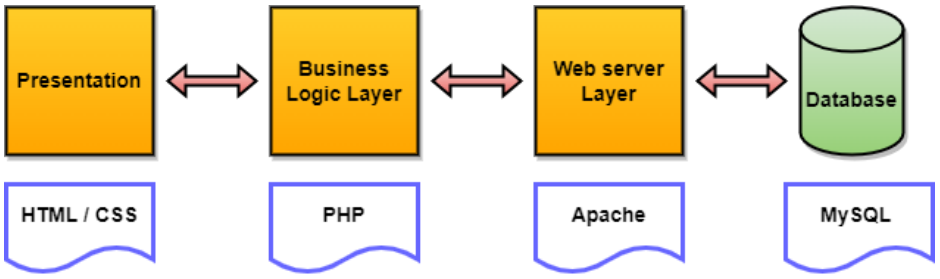

Figure 3.2: System Technical Architecture

The system is made up of four layers:

- **The presentation layer:** includes user interfaces built using the HTML5 and CSS tools. The interface is designed using GUI components which are user-friendly.
- **Business Logic layer:** is used for the retrieval of data obtained from user interfaces or acquired form databases. Activities for calculating, managing and making decisions and communicating with the front-end application and back-end database are managed by business logic written using PHP language are appearing here.
- **Web Server layer:** is to serve websites on the internet. It acts as a middleman between the server and client machines. It pulls content from the server on each user request and delivers it to the web. Apache Tomcat Web Server is used as a Web Server in the system.
- **Database layer:** is used to store data forwarded by the business layer and retrieve data on demand to the business layer. MySQL database software is used as a back-end database of the system.

3.3.3 System Hardware Architecture:

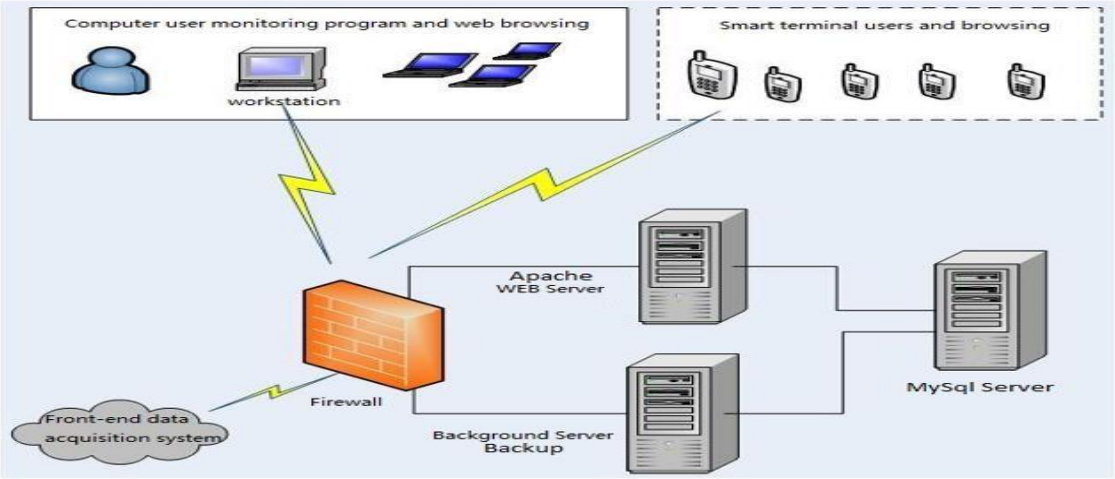

Figure 3.3: System Hardware Architecture

Hardware architecture is primarily concerned with the internal electrical interface among the system's components or subsystems and the interface between the external environments, especially the devices operated by the electronic display viewed by a use.

3.3.4 External interface:

- **Name of application:** Galaxy fashion (online ready wear shop).
- **Owner of the application:** Keerthan Setty
- **Details of interface:** Admin Panel, User Panel, Shop Panel, product panel.
- **Type of interface:** Graphical User Interface, Menu driven, Form based Interface.

3.4. Description of program:

Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flow, process, sources and sink and stores all the description using easily understandable symbols. DFD is one of the most important modelling tools. It is used to model the system, components that interact with the system, uses the data and information flows in the system. DFD shows the information moves through the and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output. DFD is also known as bubble chart or Data Flows Graphs. DFD may be used to represent the system at any level of abstraction.

3.4.1 Structure Chart:

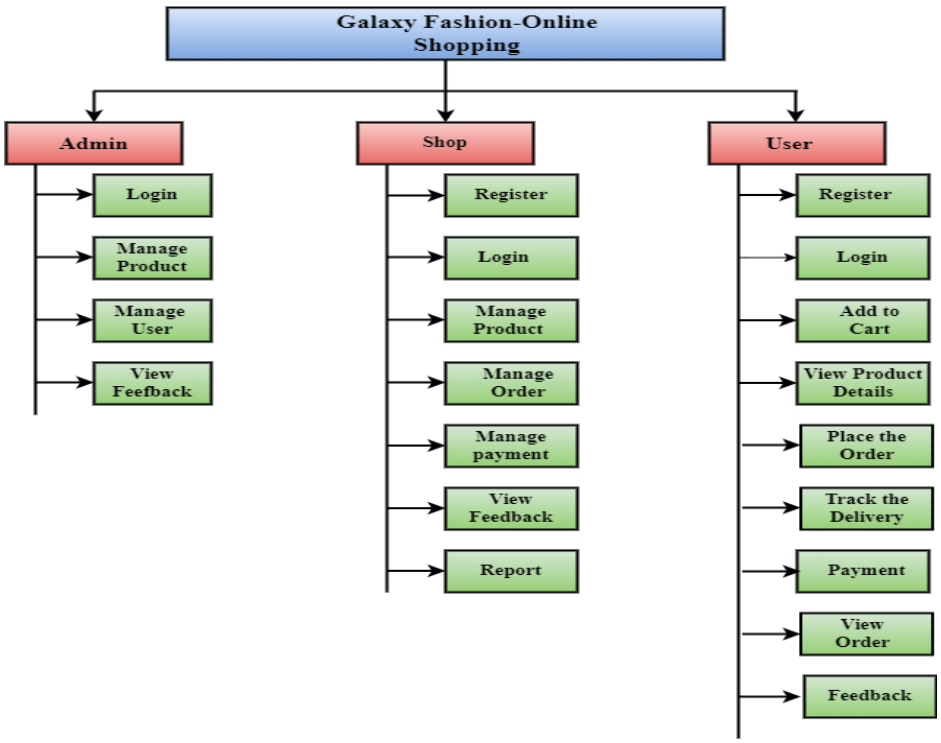

Figure 3.4: Structure of the System

Structure of the project online clothes shopping has two main modules. They are Admin, User. Admin module is responsible for manage in all activities in the project.

3.4.2 Context Flow Diagram:

Context flow diagram is a top level (also known as level 0) data flow diagram. It contains only one process node that generalizes the functions of the entire system in relationship to external entities. In context diagram the entire system is treated as single process and all its inputs, outputs, sinks and sources are identified and shown below.

Figure 3.5: context flow diagram

The above diagram is context flow diagram. It consists of 3 attributes they are: User, Shop and Admin. User Send work request and receive productfrom Shop. The admin manages all the entities of the Project.

3.4.3 Data Flow Diagram:

Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flow, process, sources and sink and stores all the description using easily understandable symbols. DFD is one of the most important modelling tools. It is used to model the system, components that interact with the system, uses the data and information flows in the system. DFD shows the information moves through the and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output. DFD is also known as bubble chart or Data Flows Graphs. DFD may be used to represent the system at any level of abstraction.

Rules Regarding DFD Construction:

- A process cannot have only outputs.
- A process cannot have only inputs.
- The inputs to a process must be sufficient to produce the outputs from the process.
- All data stores must be connected to at least one process.
- All data stores must be connected to a source or sink.

Top level DFD:

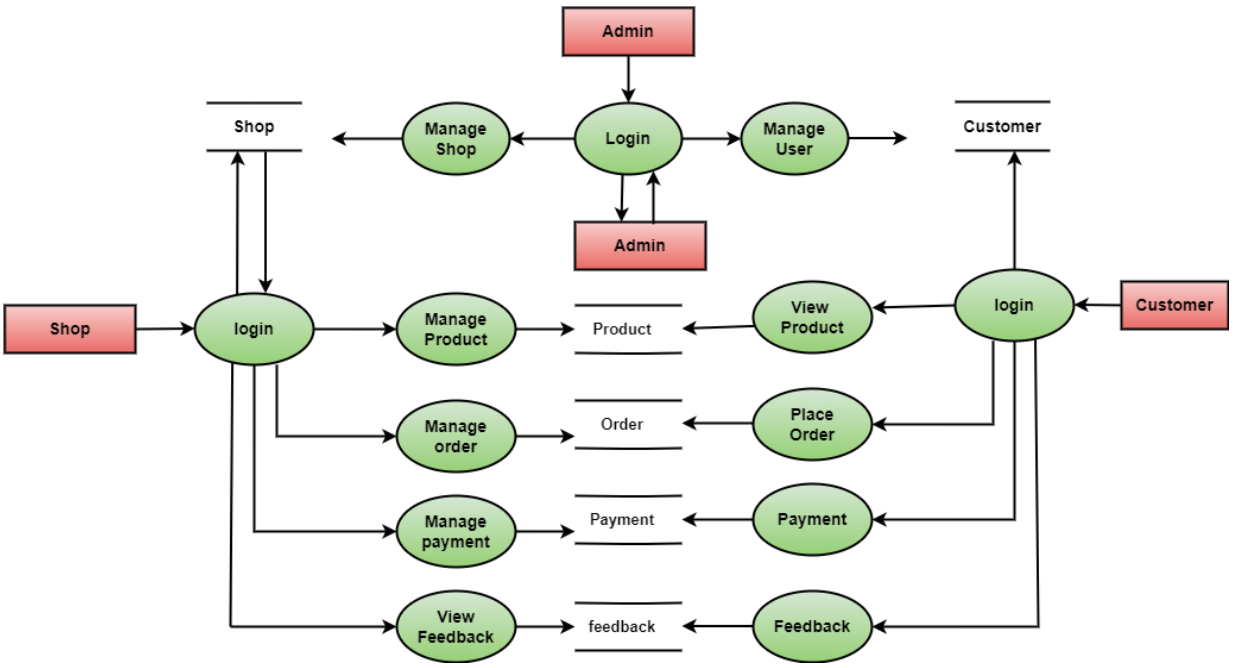

Figure 3.6: Top level data flow diagram

Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flow, process, sources and sink and stores all the description using easily understandable symbols. DFD is one of the most important modelling tools.

DFD Level 1: Admin

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes.

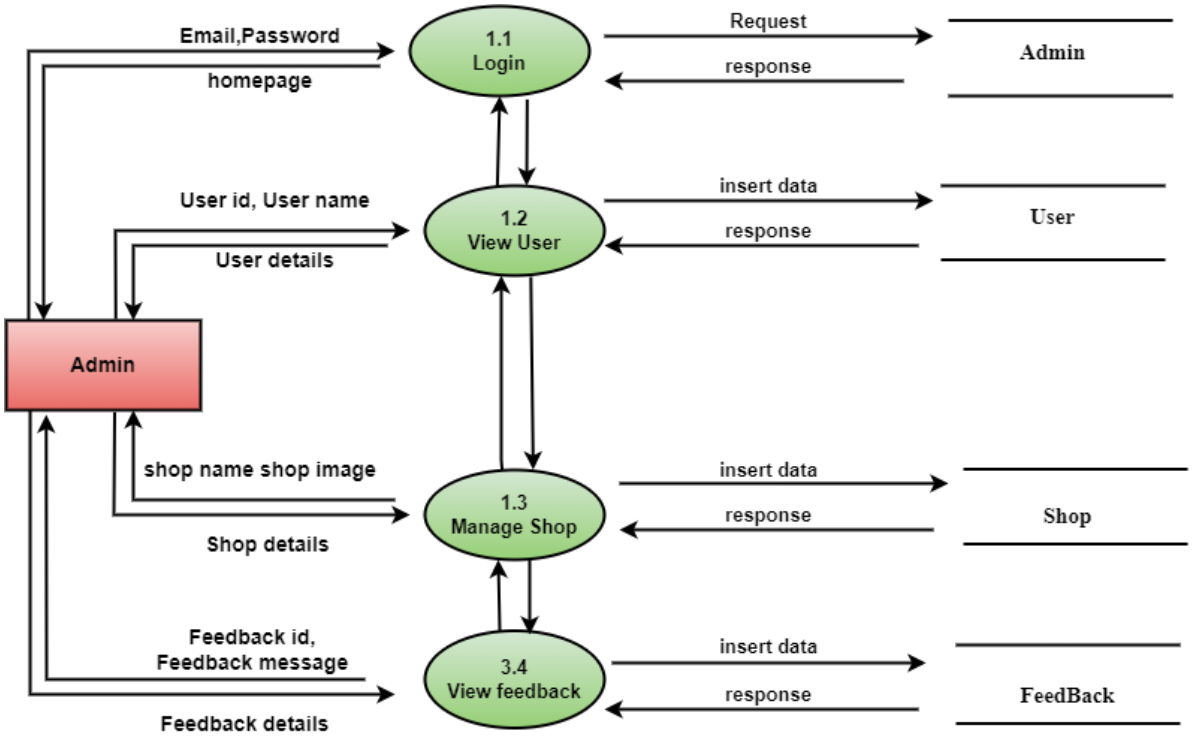

Figure 3.7: level 1data flow diagram admin

In the project Galaxy Fashion shop management system admin is the main module. Admin is module visible all the activities in the project. Such are every data in the database can be accessed by admin. In the above dataflow diagram the module admin have the process named user, manage shop, View feedback etc.

Level of DFD (Shop):

The Shop Management component is divided in a detailed way. There are 6 processes namely Login, manage product, category details, Payment and View order and feedback. There are 6 database tables mentioned in the diagram shop, user, product, delivery, order, payment and feedback. We find the inter process communication between process.

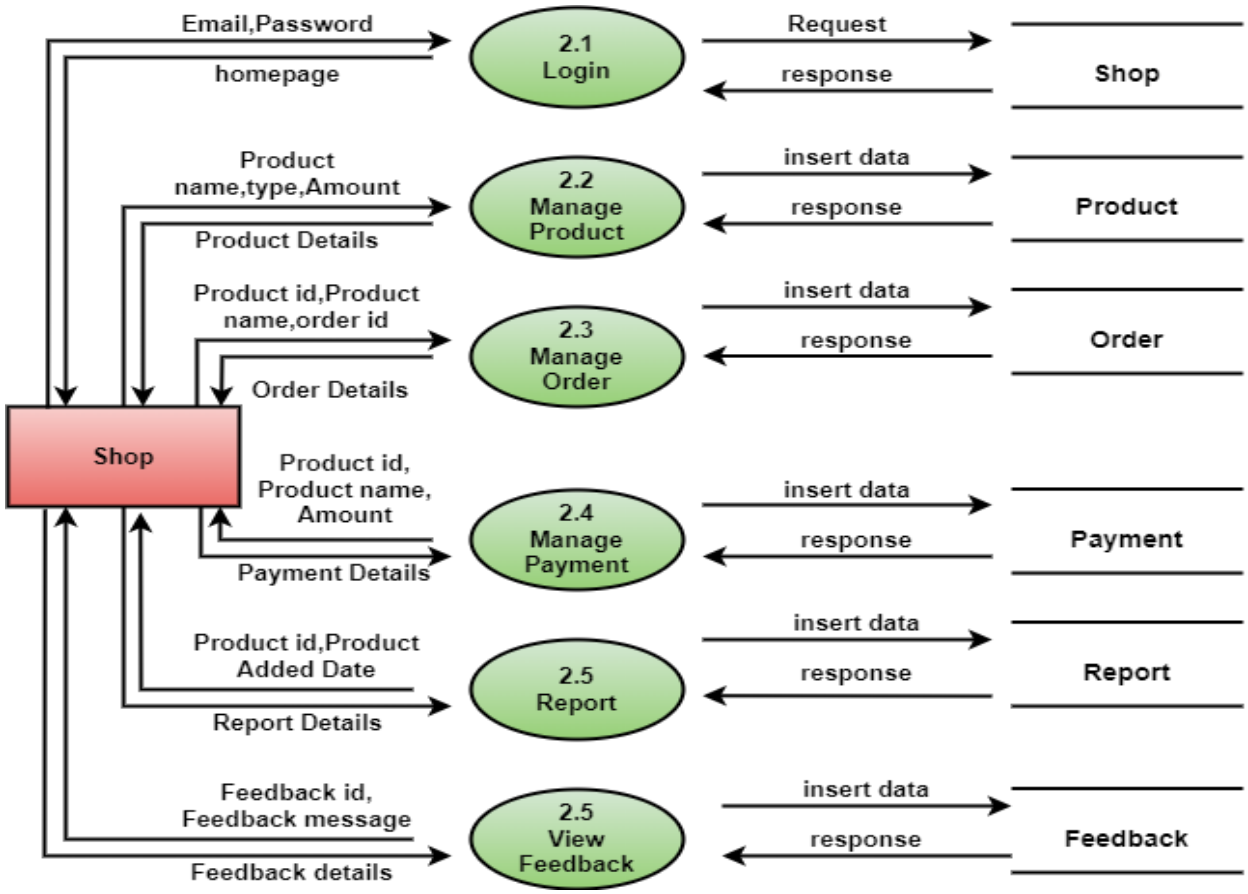

Figure 3.9: level 1data flow diagram admin

DFD Level 1: User

The User Management component is divided in a detailed way. There are 7 database tables mentioned in the diagram admin, user, product, delivery, cart, order, payment and feedback. We find the inter process communication between process.

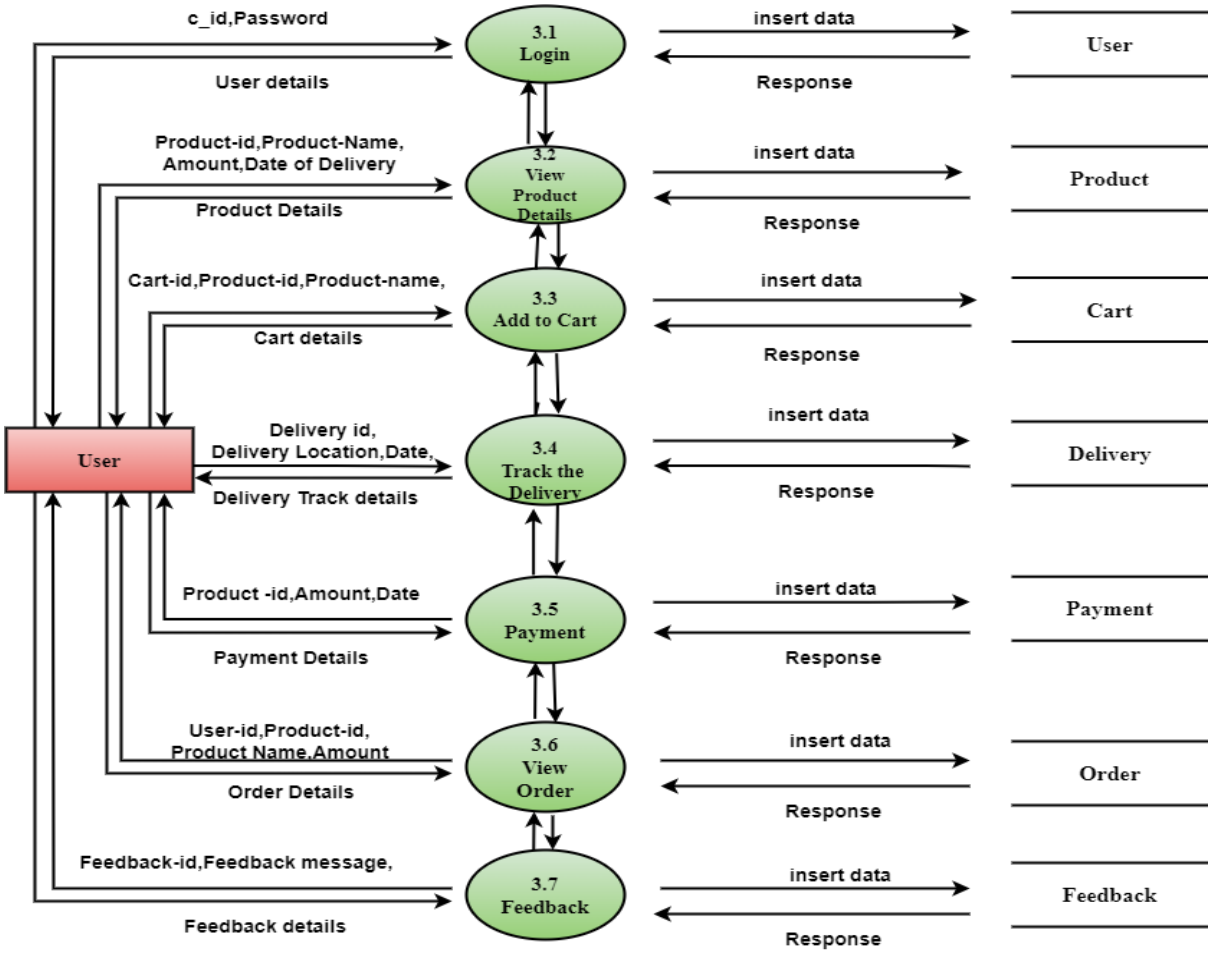

Figure 3.8: level 1data flow diagram user

There are 7 processes namely Login, View product details, Add to cart, Track Delivery, Payment and View order and feedback.

3.5 Description of Components:

The following the entire component used in our application how the process for the inputand output is specified as below

3.5.1 Login

- **Input**: username, password.
- **Process**: The system checks for the correctness of the username and password.
- **Output**: Admin will be taken to the home page on successful login.

3.5.2 Manage user:

- **Input:** User uploads his complete details. (u_id,u_name)
- **Process:** Details will be stored in database.

- **Output:** Details are viewed by Admin.

3.5.3 Manage shop:

- **Input:** Shop uploads his complete details. (s_id,pr_id, pr_name)
- **Process:** Details will be stored in database.
- **Output:** Details are viewed by Admin.

3.5.4 Add to cart

- **Input**: pr_id, pr_name, amt.
- **Process**: The system checks the product added to cart.
- **Output:** System shows the list of product in the cart.

3.5.5 View order

- **Input:** product is ordered by Users. (o_id, p_name, u_id, amount)
- **Process**: ordered products are stored in database
- **Output:** orders Status is delivered, cancel, pending or rejected displayed.

3.5.6 Track the Delivery

- **Input:** product is ordered by Users. (o_id, p_name, u_id, amount, date of delivery)
- **Process**: ordered products are stored in database
- **Output:** orders Status is tracked, delivered, cancel, pending or rejected displayed.

3.5.7 Feedback

- **Input:** User will send the Feedback.(u_id,f_message).
- **Process:** Feedback sent to the admin.
- **Output:** User view or deleted the message.

3.5.8 Manage order

- **Input:** Products are ordered by users and shop will accepted order (o_id, p_name, u_id, amount)
- **Process**: ordered products are stored in database
- **Output:** orders Status is delivered, cancel, pending or rejected displayed.

3.5.8 Manage payment:

- **Input:** User pays for the products. (u_id,u_name)
- **Process:** Details will be stored in database.
- **Output:** Details are viewed by shop.

****

CHAPTER-4

DATABASE DESIGN AND DESCRIPTION

4.1 Introduction

Database is collection of related data, which can be of any size and complexity. By using the concept of Database, we can easily store and retrieve the data. The major purpose of a database is to provide the information, which utilizes it with the information's that the system needs according to its own requirements. Database Design is the process of producing a detailed data model of a database. Database Design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements are interrelated. Database can be extremely important tools for managing large amount of data. A database is collection of related data, which can be of any size and complexity and an also provide important security. Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. The data base system must provide the safety of the information solved despite system crashes or due to the attempts at unauthorized access.

4.2 Purpose and Scope

4.2.1 Purpose

This database requirement specification describes the function and performance requirements of the catering and event hub database. A good database is one that is simple to understand and well planned. The database doesn't have redundant tables. One can use ER Diagram in order to make a good database. This database is used to understand the project caters hub. This database design description describes the function and performance requirements by the Dynamic website for Online ready wear shop System:

This database stores information about:

- Admin

- Shop
- Shop activities.

4.2.2. Scope

- A secure database design can save you from catastrophe in case someone tries to hack into your database.
- Efficient support for complex and interrelated business processes
- Consistent availability of data to support business operations and decision making
- Reduction in redundant data storage
- Increased productivity at work
- Avoidance of inconsistent data.

4.3 Database Identification

Database table name and column names are defined without leaving space.

- Lowercase used to create database tables and columns.
- Primary key and foreign key defined with same name

4.3.1 Naming Conventions

- Identifier should not have quotes.
- Identifier names should not have whitespace.
- Identifiers should be written entirely in lower case. 36
- Database object names, particularly column names, should be a noun describing thefield or object.

4.3.2 Database identification

- Admin
- user
- shop
- Add to cart
- Category

- Main category
- product
- Orders
- Payment
- Feedback
- Shipping

4.3.3 System using database

- Update
- View
- Delete
- Login
- logout
- Register
- Order
- Payment
- Delivery

4.4 Schema Information

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created. In a schema diagram, all database tables are designed with unique columnsand special features, e.g, primary/foreign keys or not null etc. Formats and symbols for expression are university understood, eliminating the possibility of confusion. The table relationships also are expressed via a parent tables primary key lines when joined with the child tables corresponding foreign keys. Schema diagrams have an important function because they force database developers to transpose ideas to paper. This provides an overview of the entire database, while facilitating future database administrator work.

In a schema diagram, all database tables are designed with unique columns and special features, e.g., primary/foreign keys or not null, etc. Formats and symbols for expression are universally understood, eliminating the possibility of the confusion.

4.5 Database Table Structure

Table 4.5.1 Admin

Description: This table is used to store admin details of the website.

Table 4.1: Structure of table Admin

Column	Types	Null	Default	Comments
a_id(primary)	INT (20)	No	-	ID, Required. Primary Key. Autogenerated.
a_name	VARCHAR(20)	No	-	Username, Required. Stored adminusername.
a_password	VARCHAR(8)	No	-	Password, Required. Stored adminpassword
a_email	VARCHAR(25)	No	-	This is used to store admin user name

Table 4.5.2 User

Description: This table is used to store the login details of the Shop.

Table 4.2: Structure of table user

Column	Types	Null	Default	Comments
u_id(primary)	INT (20)	No	-	ID, Required. Primary Key. Auto generated
username	VARCHAR (50)	No	-	Username, Required. Stored shop Username.
password	VARCHAR (50)	No	-	Password, Required. Stored shoppassword
u_name	VARCHAR (30)	No	-	Name, Required. Stores name of the Shop.
email	VARCHAR (30)	No	-	Email, Required. Stored shop email id.

address	VARCHAR (100)	No	-	Address, Required, Stored address of theshop.
contact	VARCHAR (10)	No	-	Contact No, Required. Stores Phone No of the shop.
status	VARCHAR (30)	No	-	Status, Required. Stores the Status

Table 4.5.3 shop

Description: This table is used to store the login details of the shop.

Table 4.3: Structure of table shop

Column	Types	Null	Default	Comments
s_id (primary)	INT (35)	No	-	ID, Required. Primary Key. Auto generated
s_username	VARCHAR (60)	No	-	Username, Required. Stores username of the shop.
s_password	VARCHAR (20)	No	-	Password, Required. Stored shop password
s_name	VARCHAR (40)	No	-	Name, Required. Stores name of the Shop.
s_email	VARCHAR (50)	No	-	Email, Required. Stored shop email id.
s_address	NUMBER (100)	No	-	Address, Required. Stores address of the Shop.
s_contact	VARCHAR (50)	No	-	Contact No, Required, Stored contact of the shop.
s_image	VARCHAR (100)	No	-	Image, Required. Stored shop image.
s_status	VARCHAR (100)	No	-	Status, Required. Stores the status of the shop

Table 4.5.4 Product

Description: This table is used to store the product details.

Table 4.4: Structure of the table product

Column	Types	Null	Default	Comments
pr_id(primary)	INT (50)	No	-	ID, Required. Primary Key. Auto generated
pr_name	VARCHAR (50)	No	-	Name, Required. Stores name of the Product.
pr_image	VARCHAR (100)	No	-	Image, Required. Stored product image.
pr_price	VARCHAR (100)	No	-	Price, Required. Stores the price Product.
pr_description	VARCHAR (200)	No	-	Description, Required. Stores the Description of product.

Table 4.5.4 Order

Description: This table is used to store the ordered details of the product.

Table 4.5: Structure of the table order

Column	Types	Null	Default	Comments
o_id	INT (25)	No	-	ID, Required. Primary Key. Auto generated
pr_id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
pr_amount	VARCHAR (25)	No	-	Amount, Required. Stores the price of the product.
u_id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
date	DATE	No	-	Date, Required. Stores the Date to apply.
u_name	VARCHAR (30)	No	-	Name, Required. Stores name of the user.

Pr_name	VARCHAR (30)	No	-	Name, Required. Stores name of the product.
c_address	VARCHAR (100)	No	-	Address, Required, Stored address of theuser.
c_contact	VARCHAR(30)	No	-	Contact No, Required. Stores phone no of the user
c_pincode	VARCHAR(6)	No	-	Pin code, Required. Stores pin code of the user

Table 4.5.5 Payment

Description: This table is used to store the ordered details of the product.

Table 4.6: Structure of table Payment

Column	Types	Null	Default	Comments
pa_id	INT (15)	No	-	ID, Required. Primary Key. Auto generated
pa_amount	VARCHAR (30)	No	-	Amount, Required. Stores the amount Package.
o_id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
pr_id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
u_ id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
transaction_id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
pa_date	DATE	No	-	Date, Required. Stores the Date to apply

Table 4.5.7 Feedback

Description: This table is used to store the feedback details given by the user.

Table 4.7: Structure of table Feedback

Column	Types	Null	Default	Comments
f_id	INT (25)	No	-	ID, Required. Primary Key. Auto generated
c_id	INT (25)	No	-	ID, Required. Foreign Key. Auto generated
c_name	TEXT	No	-	Name, Required. Stores name of the User.
C email	DATE	No	-	Email, Required. Stored user email id.
subject	VARCHAR (50)	No	-	Subject, Required. Stores subject of thefeedback
message	VARCHAR (300)	No	-	Message, Required. Stores feedback message
date	DATE	No	-	Date, Required. Stores the Date to apply

Table 4.5.8 Cart

Description: This table is used to store the cart details given by the user.

Table 4.8: Structure of table cart

Column	Types	Null	Default	Comments
ca_id(primary)	INT (25)	No	-	ID, Required. Primary Key. Auto generated

Pr_id	INT(25)	No	-	Product id ,required,
pr_name	VARCHAR (50)	No	-	Name, Required. Stores name of the Product.
pr_image	VARCHAR (100)	No	-	Image, Required. Stored product image.
Pr_amt	INT(50)	No	-	Product amount required. Product price required.
date	DATE	No	-	Product

4 .5.9 Table Shipping

Description: This table is used to store the shipping details of the product.

Table 4.9: Structure of table Shipping

Column	Types	Null	Default	Comments
sh_id	INT (25)	No	-	ID, Required. Primary Key. Auto generated
Sh_city	VARCHAR (50)	No	-	Sh_city required, it stored the shipping location or city address.
Sh_address	VARCHAR (50)	No	-	Sh_address Required, It stored the address of shipping location of the product.
O_id	INT(25)	No	-	ID, Required, it stores the order id
C_id	INT(25)	No	-	User id required.it stores the user id.
Pr_id	INT(25)	No	-	Product id required. It stores the product id.
Pr_name	VARCHAR (30)	No	-	Product id required. it stores the product name.

Sh_email	VARHAR(50)	No	-	Sh_email required. It stores the shipping notification.
Sh_date	DATE/TIME	No	-	Sh_date required. It stores the shipping date.
Sh_contact	VARCHAR (50)	No	-	Sh_contact required, it stores the contact details of the user.
sh_name	VARCHAR (50)	No	-	Sh_name required, it stores the name shipping
Sh_zip	VARCHAR (55)	No	-	Sh_zip required, it stores the

4.3.2 Physical Design

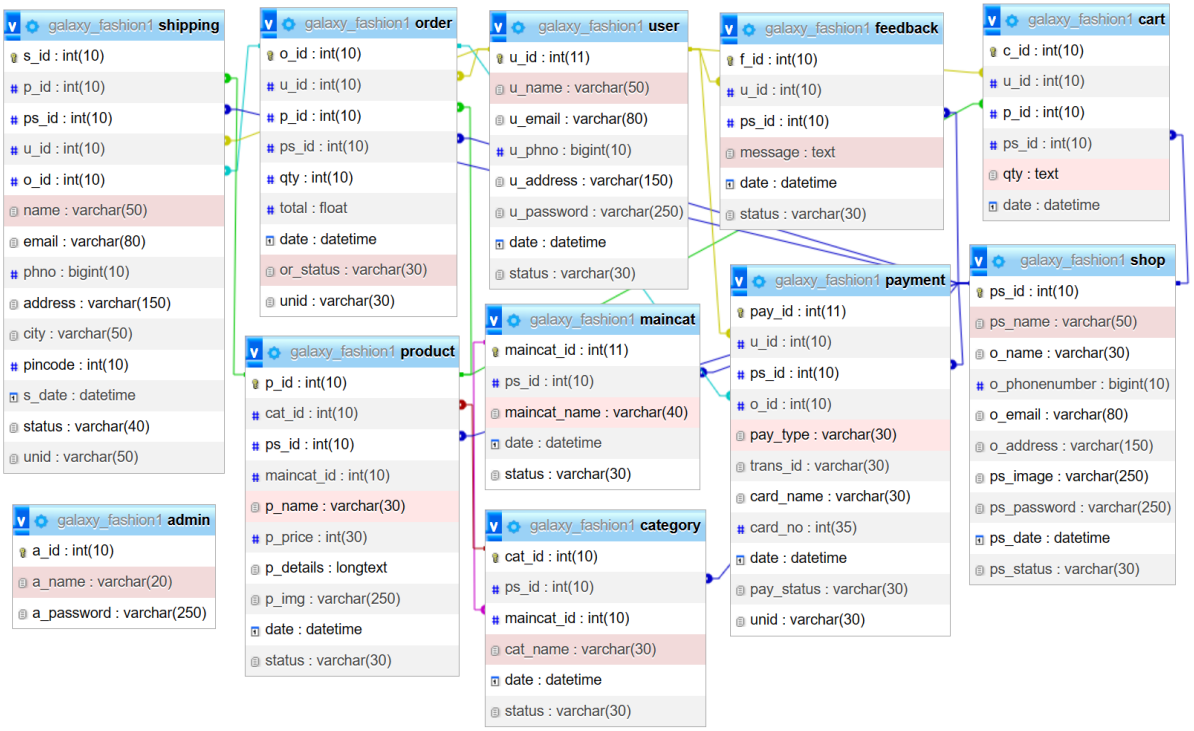

Figure 4.1: physical design of database

The physical design of your database optimizes performance while ensuring data integrity by avoiding unnecessary data redundancies. The task of building physical design is a job that truly never ends. You need to continually monitor the performance and data integrity as time passes. Many factors necessitate periodic refinement to the physical design.

4.4 Physical Structure of Database System

The following diagram represents the physical structure of the database. Here users can delete and modify the records from the table. Browse option will display the records fromthe table. Structure will display the table column and its data type.

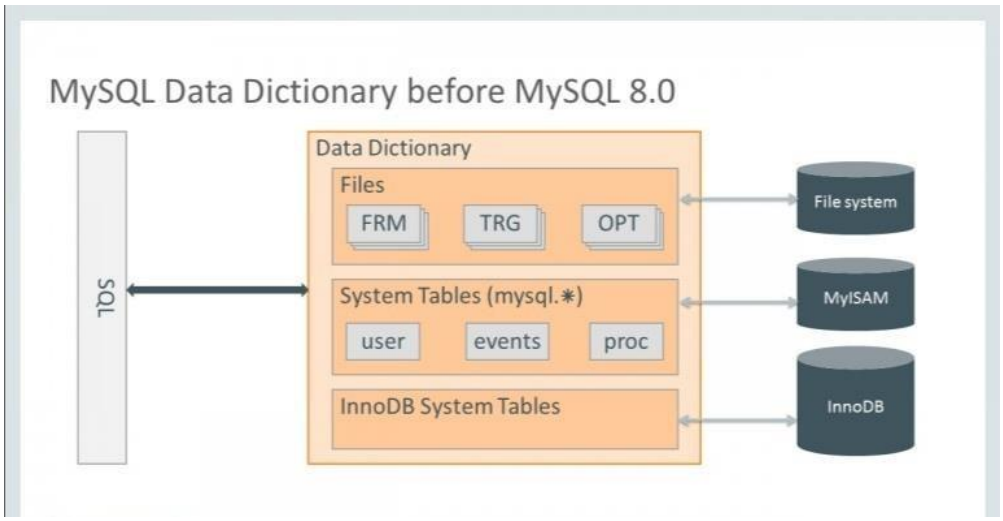

Figure 4.2: physical structure of database

In the search option users can search the records. Insert option is to insert new records in the database. Empty option will clear all the data from the database table. Drop option allows the user to delete database tables.

4.5 Data Dictionary

A data dictionary can be seen as a repository for information about a database. There are no industry standards that go into a data dictionary. It may be as simple as a list of tables with basic descriptions. Alternatively, it can be an extensive list of properties outlining precisely how data is structured, maintained, and used.

4.5.1 Table: Admin

Table 4.10: Data Dictionary of table admin

Column	Type	Null	Default	Example
a_id (primary)	int (25)	No	-	201
a_name	varchar (50)	No	-	avinaa
password	varchar (15)	No	-	avina@9480
email	Varchar(50)	No	-	Avina9480@gamil.com

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	a_id	0	A	No

4.5.2 Table Cart

Column	Type	Null	Default	Example
cart_id (primary)	int (25)	No	-	2023
pr_id	int (25)	No	-	302
c_id	Int(25)	No	-	20326
Cart_date	date	No	-	07-12-23
Cart_status	Varchar(25)	No	-	added

Indexes

Key name	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	Cart_id	0	A	No
C_id	BTREE	No	No	C_id	0	A	No
Pr_id	BTREE	No	No	Pr_id	0	A	No

4.5.3 Table Shop

Table 4.12: Data Dictionary of table cart

Column	Type	Null	Default	Example
s_id *(Primary)*	Int(30)	No	-	2001
s_name	Varchar(60)	No	-	harish
email	Varchar(50)	No	-	harish@gmail.com
password	Varchar(15)	No	-	Super12$
S_contact	varchar(20)	No	-	9867543210
S_address	Varchar(100)	No	-	puttur
S_date	date	No	-	23-10-2023
S_status	Varchar(100)	No	-	deliverd

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	c_id	0	A	No

4.5.3 Table User

Table 4.13: Data Dictionary of table user

Column	Type	Null	Default	Example
c_id *(Primary)*	Int(25)	No	-	302
C_name	Varchar(25)	No	-	arpitha
email	Varchar(50)	No	-	Arpitha13@gmail.com
password	Varchar(50)	No	-	Super14$
contact	Int(30)	No	-	9876543210
address	Varchar(100)	No	-	puttur
date	date	No	-	12-09-2023

status	Varchar(100)	No	-	orderd

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	c_id	0	A	No

4.5.4 Table Feedback

Table 4.14: Data Dictionary of table feedback

Column	Type	Null	Default	Example
f_id (primary)	int (25)	No	-	203
f_message	Varchar(100)	No	-	good
c_id	Int(25)	No	-	302
f_date	date	No	-	12-09-23
f_status	Varchar(20)	No	-	sent

Indexes

Key name	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	f_id	0	A	No
Feedback ibfk_1	BTREE	Yes	No	C_id	0	A	No

4.5.5 Table Order

Table 4.15: Data Dictionary of table order

Column	Type	Null	Default	Example
o_id (primary)	int (25)	No	-	20236
Pr_id	Int(25)	No	-	302
pr_qty	Int(25)	No	-	3
C_id	Int(25)	No	-	20236
O_date	date	No	-	12-09-23
O_status	Varchar(50)	No	-	packed

Indexes

Key name	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	f_id	0	A	No
C_id	BTREE	Yes	No	C_id	0	A	No
Pr_id	BTREE	Yes	No	Pr_id	0	A	No

4.5.6 Table Payment

Table 4.16: Data Dictionary of table cart

Column	Type	Null	Default	Example
pay_id (primary)	int (25)	No	-	201
C_id	Int(25)	No	-	302
Pay_type	Int(25)	No	-	offline
Pr_id	Int(30)	No	-	20236
Pay_price	date	No	-	1200rs
pay_status	Varchar(50)	No	-	paid
Pay_date	date	No	-	16-03-23

Indexes

Key name	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	pay_id	0	A	No
C_id	BTREE	Yes	No	C_id	0	A	No
Pr_id	BTREE	Yes	No	Pr_id	0	A	No

4.5.7 Table Product

Table 4.17: Data Dictionary of table product

Column	Type	Null	Default	Example
pr_id (primary)	int (25)	No	-	20235
Pr_name	Varchar(25)	No	-	kurthis
Pr_qty	Varchar(25)	No	-	3
Pr_amt	Int(50)	No	-	1200rs
Pr_desc	Varchar(300)	No	-	Golden royal blue and red
Pr_image	Varchar(300)	No	-	Image1.jpg
Pr_date	date	No	-	23-06-23
Pr_status	Varchar(300)	No	-	shipping

Indexes

Key name	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	pr_id	0	A	No

4.5.8 Table Shipping

Table 4.18: Data Dictionary of table cart

Column	Type	Null	Default	Example
sh_id (primary)	int (25)	No	-	2045
C_id	Int(25)	No	-	302
Pr_id	Int(25)	No	-	20236
Sh_name	Varchar(50)	No	-	clothes
Sh_email	Varchar(50)	No	-	clothes@gmail.com
Sh_city	Varchar(50)	No	-	puttur
Sh_address	Varchar(100)	No	-	kadaba
Sh_zip	Varchar(100)	No	-	507201
Sh_date	date	No	-	23-09-23
Sh_status	Varchar(100)	No	-	deliverd
Sh_contact	Int(50)	No	-	9876543210
O_id	Int(50)	No	-	Abo1

Indexes

Key name	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	sh_id	0	A	No
C_id	BTREE	Yes	No	C_id	0	A	No
Pr_id	BTREE	Yes	No	Pr_id	0	A	No
O_id	BTREE	Yes	No	O_id	0	A	No

4.8 ER Diagram

Entity relationship diagram is used in modern database software engineering to illustrate logical structure of database. It is a relational schema database modelling method used to model a system and approach. This approach is commonly used in database design. The diagram created using this method is called the E-R diagram.

Entity

Entity is represented by a box within the ER Diagram. Entities are abstract concepts, each representing one or more instances of the concept in question. An entity might be considered a container that holds all of the instances of a particular thing in a system. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity.

Relationship:

Relationships are represented by Diamonds. A relationship is a named collection or association between entities or used to relate to two or more entities with some common attributes or meaningful interaction between the objects.

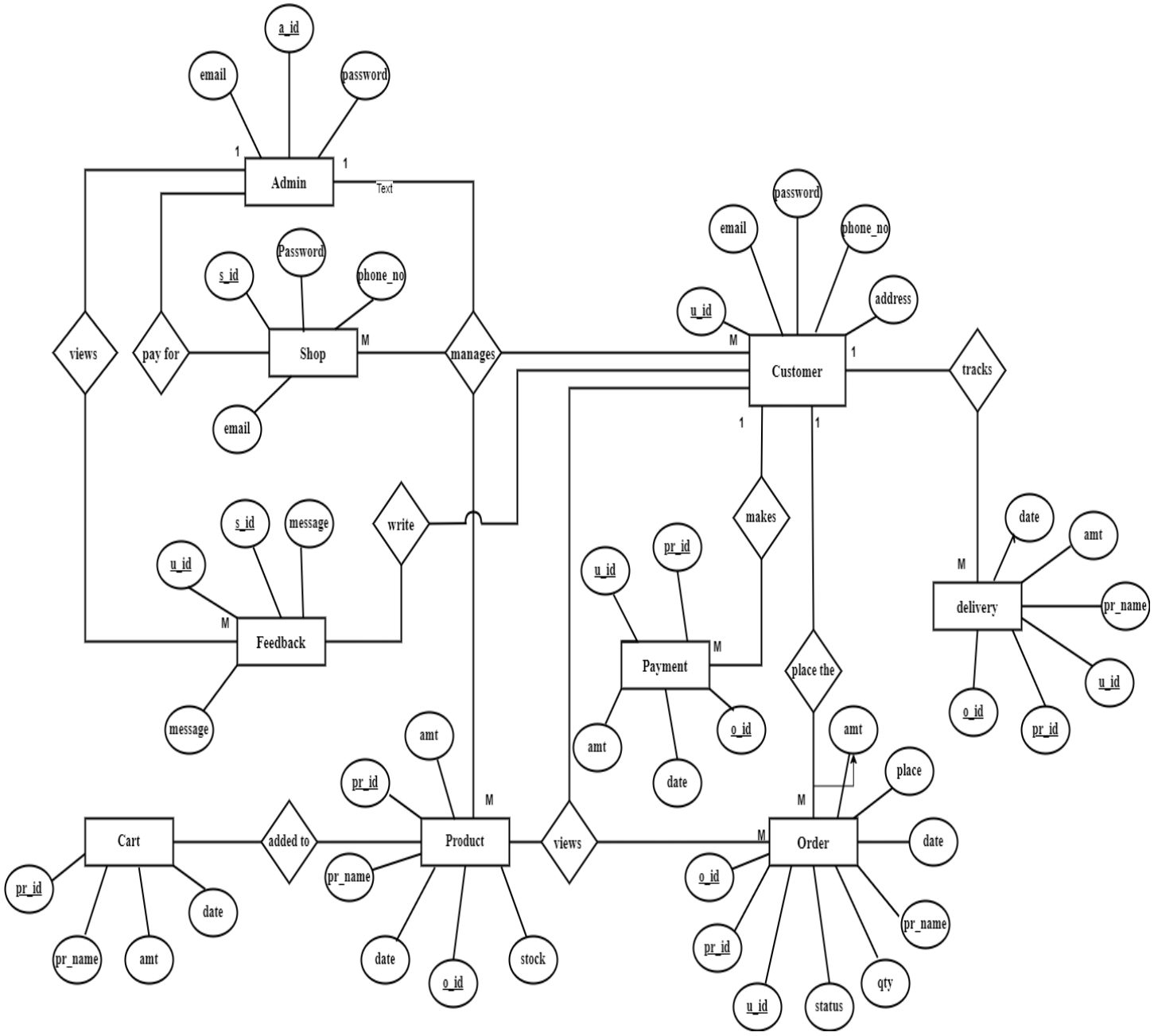

Figure 4.3: Entity Relation Diagram

An entity-relationship (ER) diagram is a specialised graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities.

4.9 Database Administrative Information

4.9.1. System Information

- Server: 127.0.0.1 via TCP/IP
- Server Type: MySQLi
- Server Version: 5.7.14 – MySQLi Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

4.9.2. Database Management System (DBMS)

- Configuration Version: MySQLi 5.7.26
- Protocol Version: 10
- Supported Operating System: Windows 7, 8, 10
- MySQL client Version: mysqlnd 5.0.8-dev-20102224-$
- Revision: 310735$

4.9.3 Support Software Available for Maintaining Database

The system installed a MySQL server while installing WAMP software. The entire backup content is stored in the MySQLi data folder.

4.9.4 Storage requirements

The storage engine represents the heart of a MySQL Server.

The storage engine has a number of duties including:

- The storage engine represents the heart of a MySQL Server.
- The storage engine has a number of duties including:
- Recovering the database from system failure.

- Management of files and database pages used to store data.
- Manage data buffers and system IO to the physical data pages.
- Manage locking and Concurrency issue.

4.9.5 Backup & Recovery

Database recovery is the process of restoring the databases to a correct state following a failure. The failure may be the result of a system crash due to hardware or software errors, a media failure, such as a head crash, or a software error in the application, such as a logical error in the program that is accessing the database. It may also be the result of unintentional or intentional corruption or destruction of data. Whatever the underlying cause of the failure, the DBMS must be able to recover from the failure and restore the database to a consistent state. It is the responsibility of DBMS to ensure that the database is reliable and remains in a consistent state in the presence of failure. In general, backup and recovery refer to the various strategies and procedures involved in protecting the database against data loss and reconstructing the data such that no data is lost after failure.

****

CHAPTER-5

DETAILED DESIGN

5.1 Introduction

Detailed Design the process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to begin implementation. Detailed design starts after the system design phase and system has been certified through the review. The goal of this phase is to develop the internal logic of eachof the modules identified during system design.

In the system design, the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for the modules. Detailed design is a process of developing a fully defined product design from a clear set of requirements while creating deliverables and documentation appropriate for product manufacturing.

The design of the system is perhaps a most critical factor affecting the quality of the software; it has a major impact on the later phases; particularly testing and maintenance. Another key component of the functional specification is a detailed design. For the detailed design you apply real-world technology constraints to the conceptual model, design document.

5.2 Structure of the software package

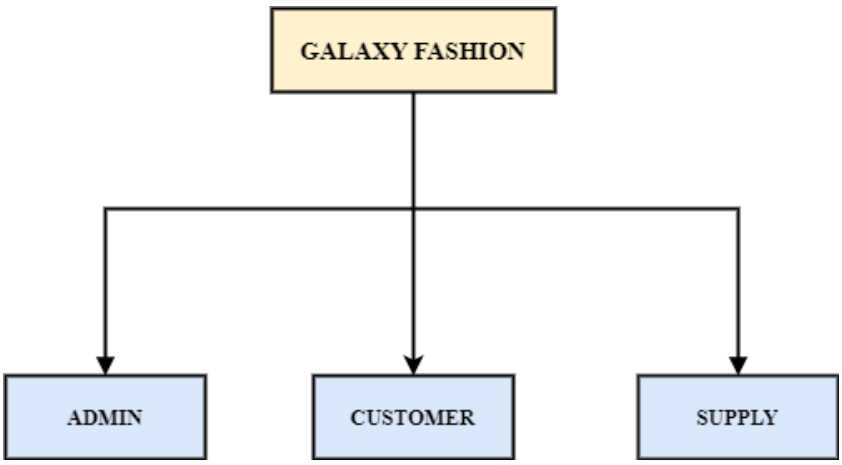

Figure 5.1Structure of software package

The functional components identified during the system design are listed here.

- Functional Component 1: Admin
- Functional Component 2: User
- Functional Component 3: Shop

5.3 Modular decomposition of the System

5.3.1 Admin

5.3.1.1 Identification of modules

- Manage User
- Manage shop
- Manage Feedback

5.3.1.2 Structure chart showing the hierarchy of modules

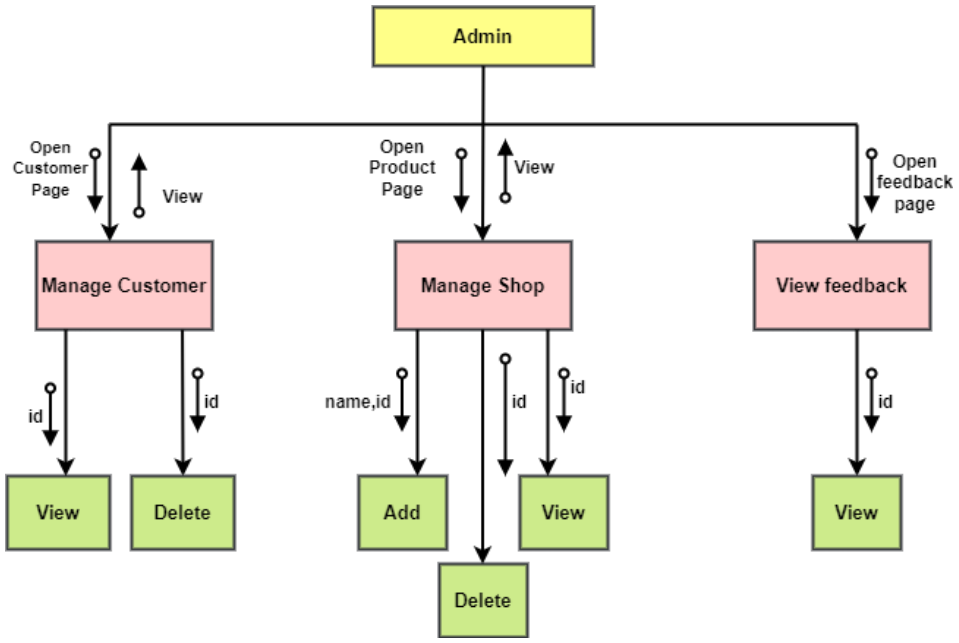

Figure 5.2 Structure chart of Admin

This module is designed in such a way that it allows the administrator to use any of the options easily so that he can add, modify or update any details. Admin module enables theadmin to have control over the user.

5.3.1.3Admin Login

- **Inputs:** Username, Password.
- **Procedural Details** (Flow Chart):

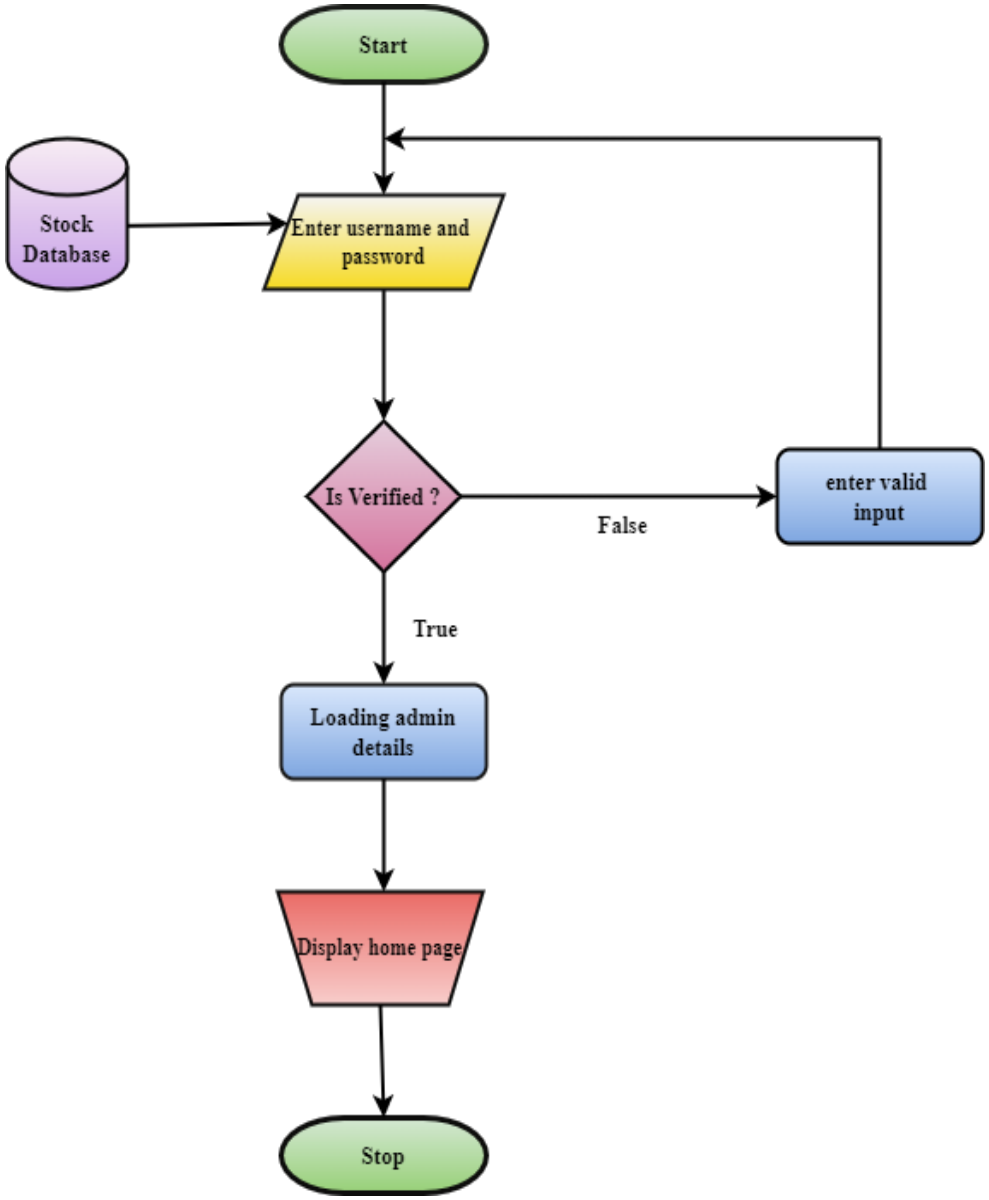

Figure 5.3 modular description of admin login

- **Database I/O interfaces:** System has interface to receive admin login.
- **Outputs:** Entered Username and password will be checked for validity if it is valid,Admin will be redirected to Homepage

5.3.1.4 Admin add shop

- **Inputs:** Username, Password, s_id, s_name
- **Procedural Details** (Flow Chart):

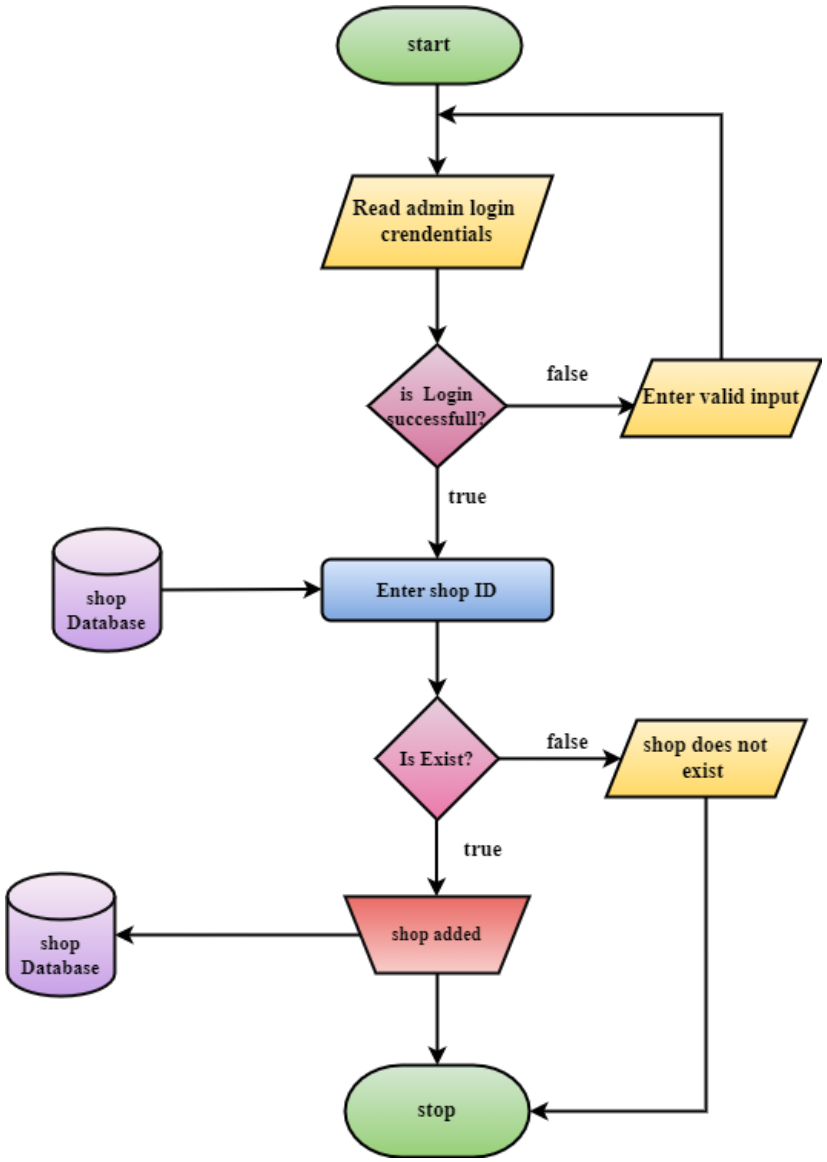

Figure 5.4 modular description of Add shop

Database I/O interfaces: System has interface to receive add product.

Outputs: shop added details are displayed

5.3.1.5 Admin view shop

- **Inputs:** Email Id, password, product ID.
- **Procedural Details** (Flow Chart):

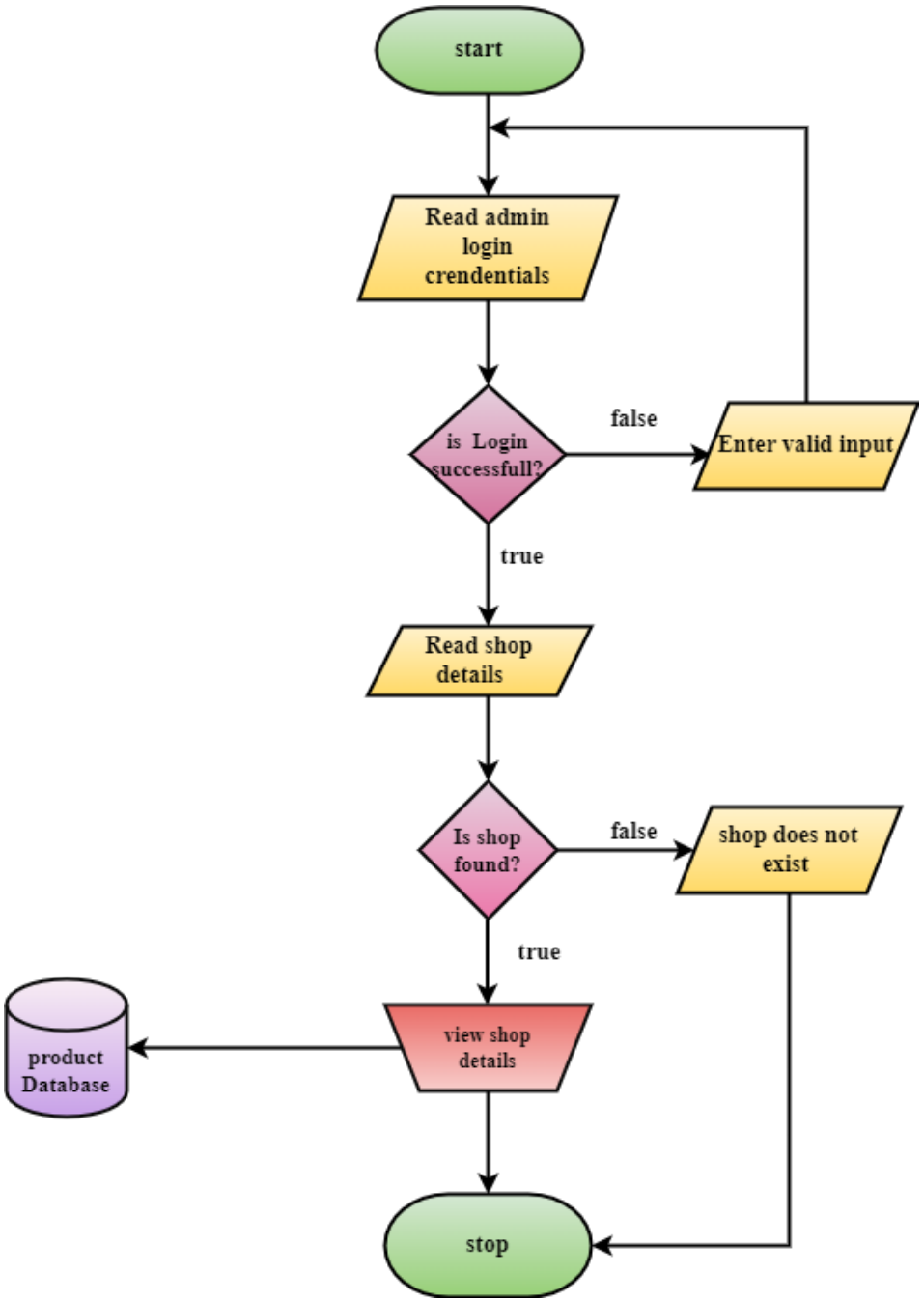

Figure 5.5 Detailed Design of View shop

- **Database I/O Interfaces:** System has interface to receive shop details.
- **Output:** shop details are displayed.

5.3.1.6 Admin delete shop

- **Inputs:** Email ID, password, product ID
- **Procedural Details** (Flow Chart):

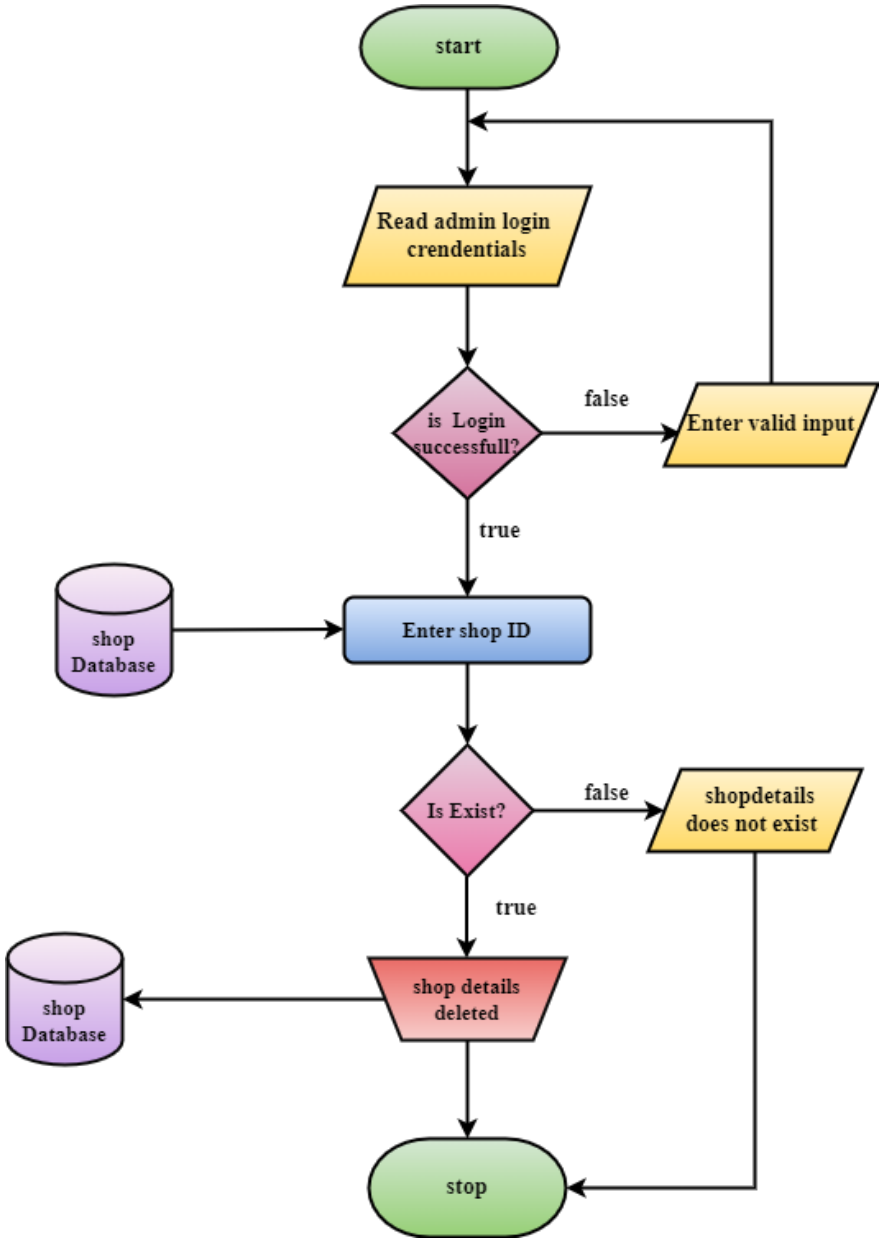

Figure 5.6 modular description of View user

- **Database I/O Interfaces:** System has interface to admin delete the shop.
- **Output:** admin can view the delete shop details.

5.3.1.7 Admin view User

- **Inputs:** User details
- **Procedural Details** (Flow Chart):

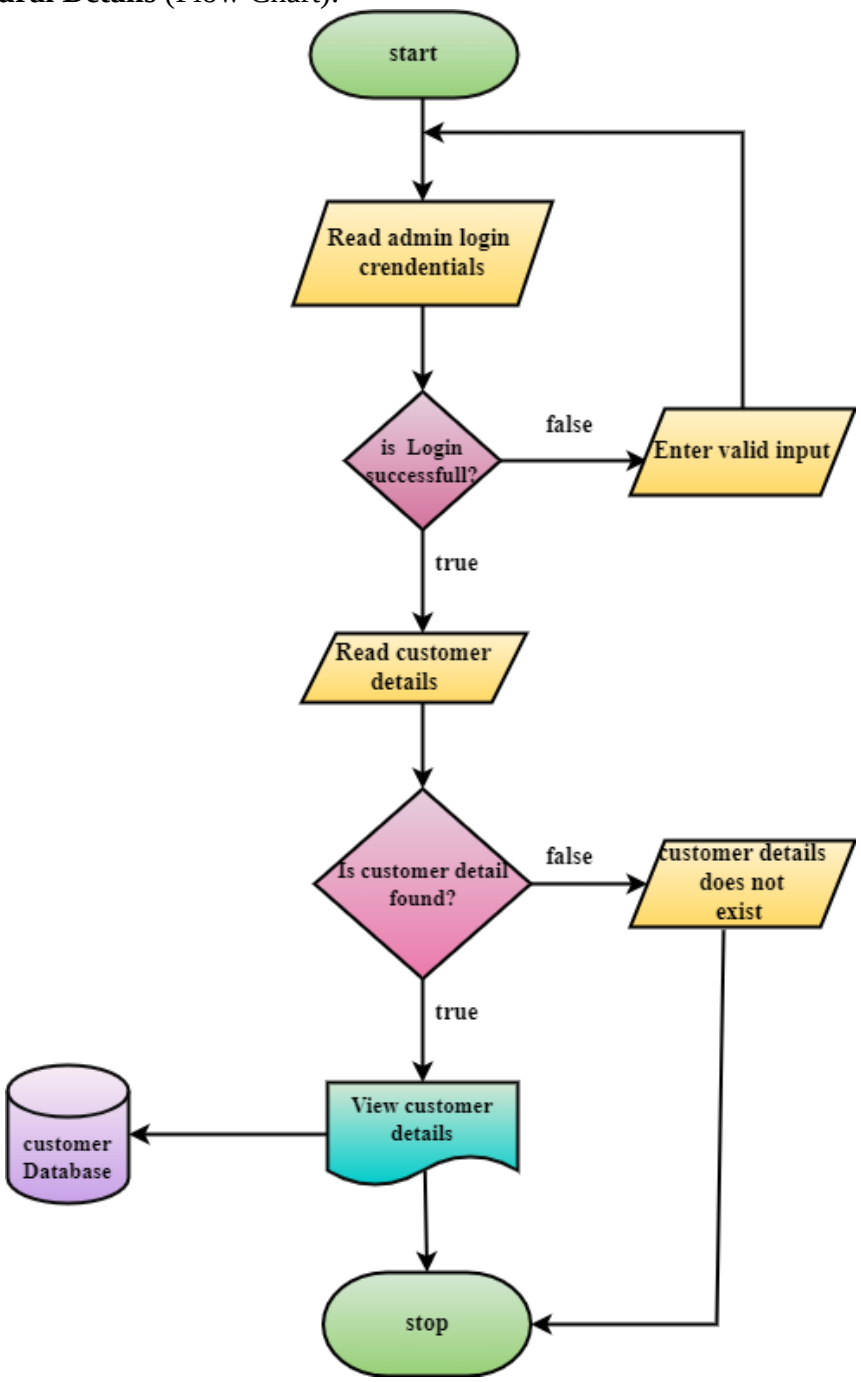

Figure 5.7 Detailed Design of View user

- **Database I/O Interfaces:** System has interface to view user.
- **Output:** admin can view the user details.

5.3.1.8: Admin view feedback

- **Inputs:** Feedback details
- **Procedural Details** (Flow Chart):

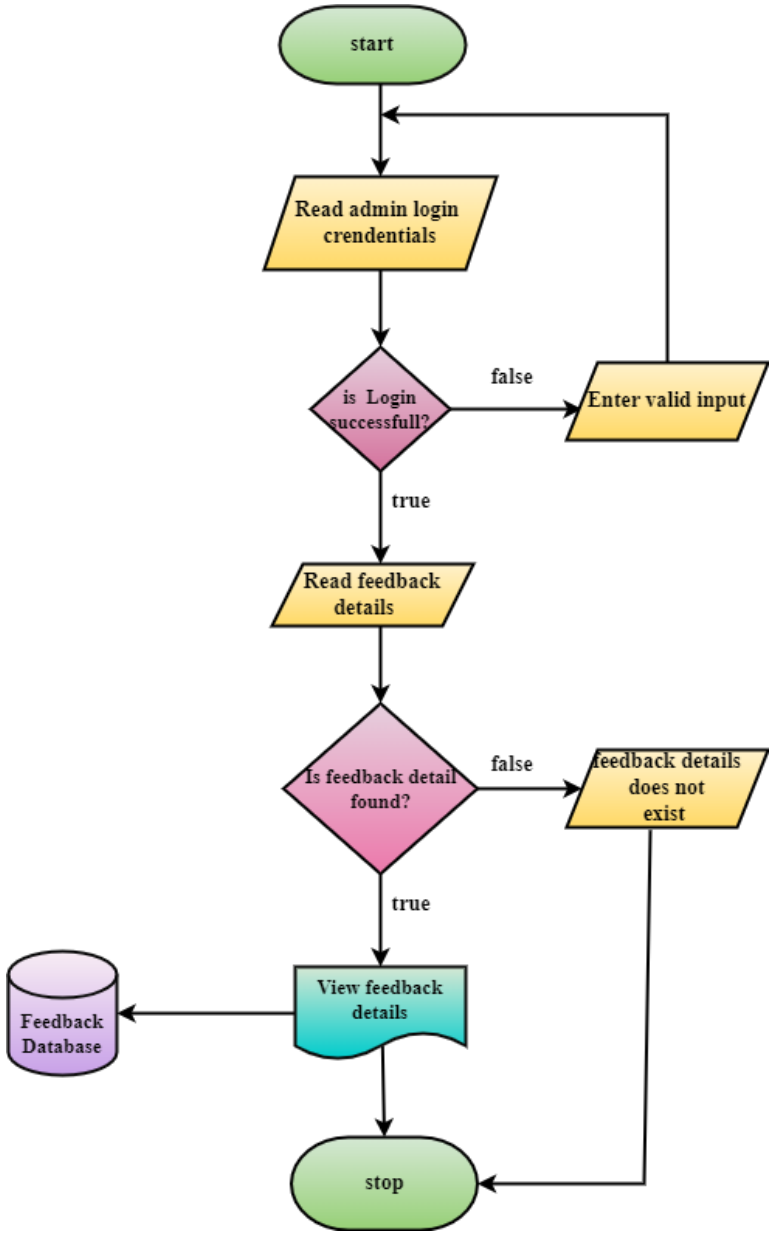

Figure 5.8 Detailed Design of View Feedback

- **Database I/O Interfaces:** System has well defined interface display feedback.
- **Output:** Feedback stored in database.

5.3.2 User

5.3.2.1 Identification of modules

- View Cart
- View products
- Write Feedback
- View report
- View Order
- View Payment

5.3.2.2 Structure chart showing the hierarchy of modules

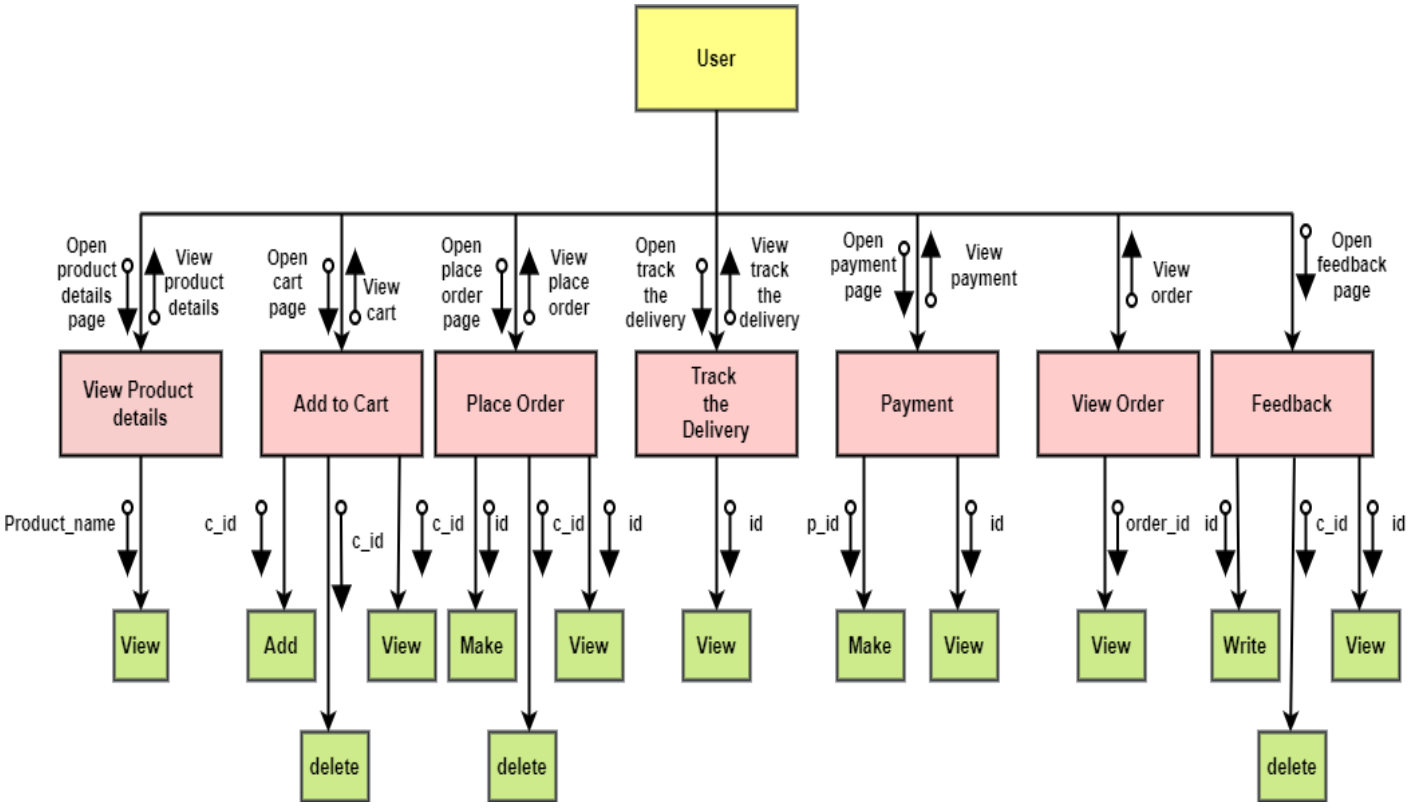

Figure 5.9 Structure chart of User

This module is designed in such a way that it allows the administrator to use any of the options easily so that he can add, modify or update any details. Admin module enables the admin to have control over the user.

5.3.2.3 User Register

- **Inputs:** Username, Password
- **Procedural Details** (Flow Chart):

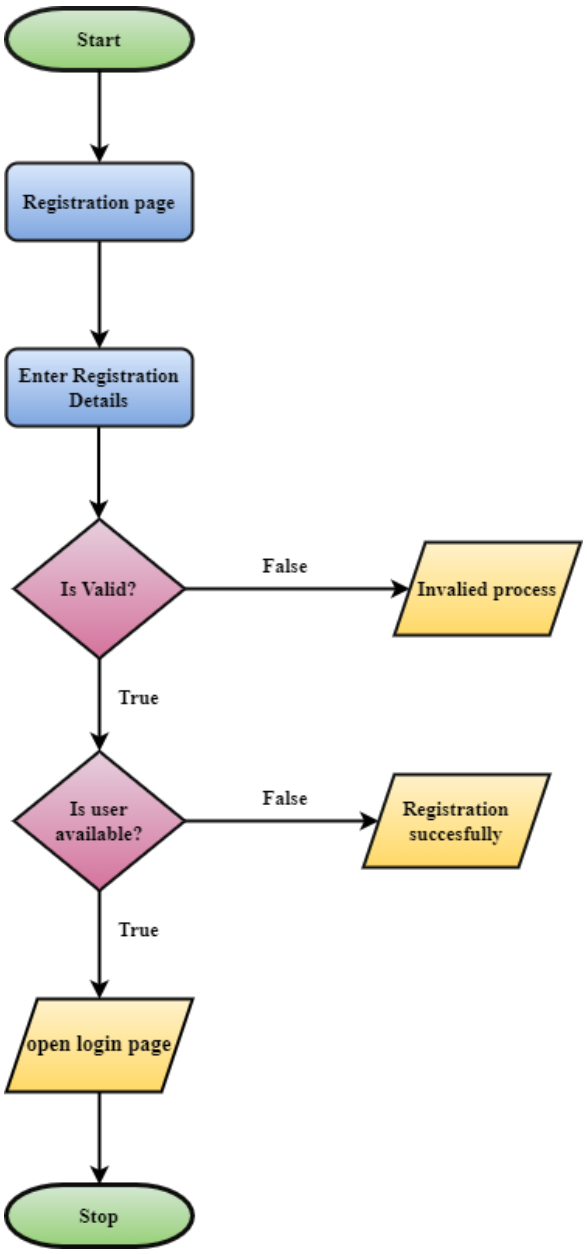

Figure 5.10 modular of user registration

- **Database I/O Interfaces:** System has well defined interface display user registration.
- **Output:** user registration stored in database.

5.3.2.4: User login

- **Inputs:** Username, Password
- **Procedural Details** (Flow Chart):

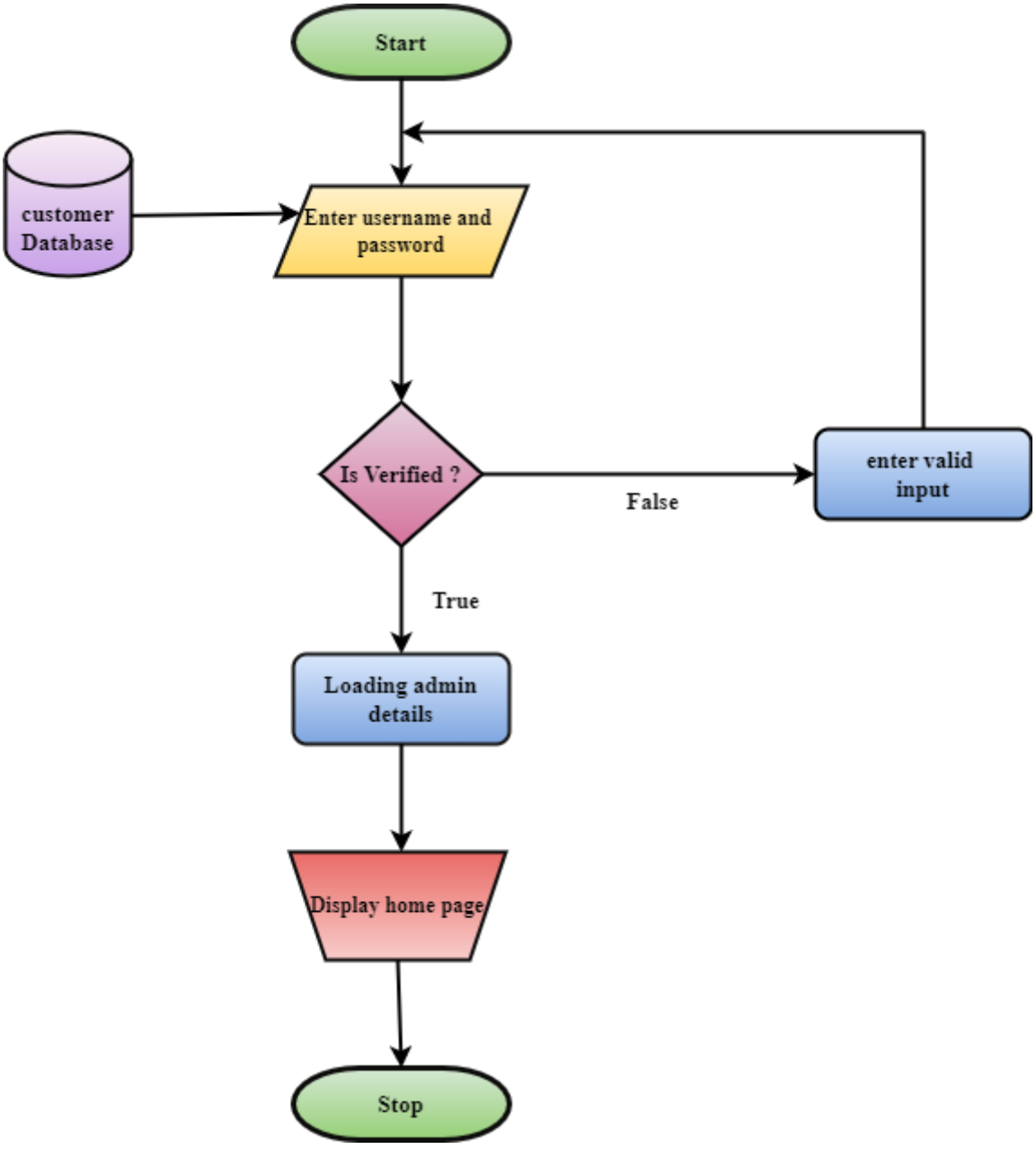

Figure 5.11: modular description of user login

- **Database I/O Interfaces:** System has well defined interface display user login.
- **Output:** user login stored in database.

5.4.2.5: User view product

- **Inputs:** pr_id , pr_name.
- **Procedural Details** (Flow Chart):

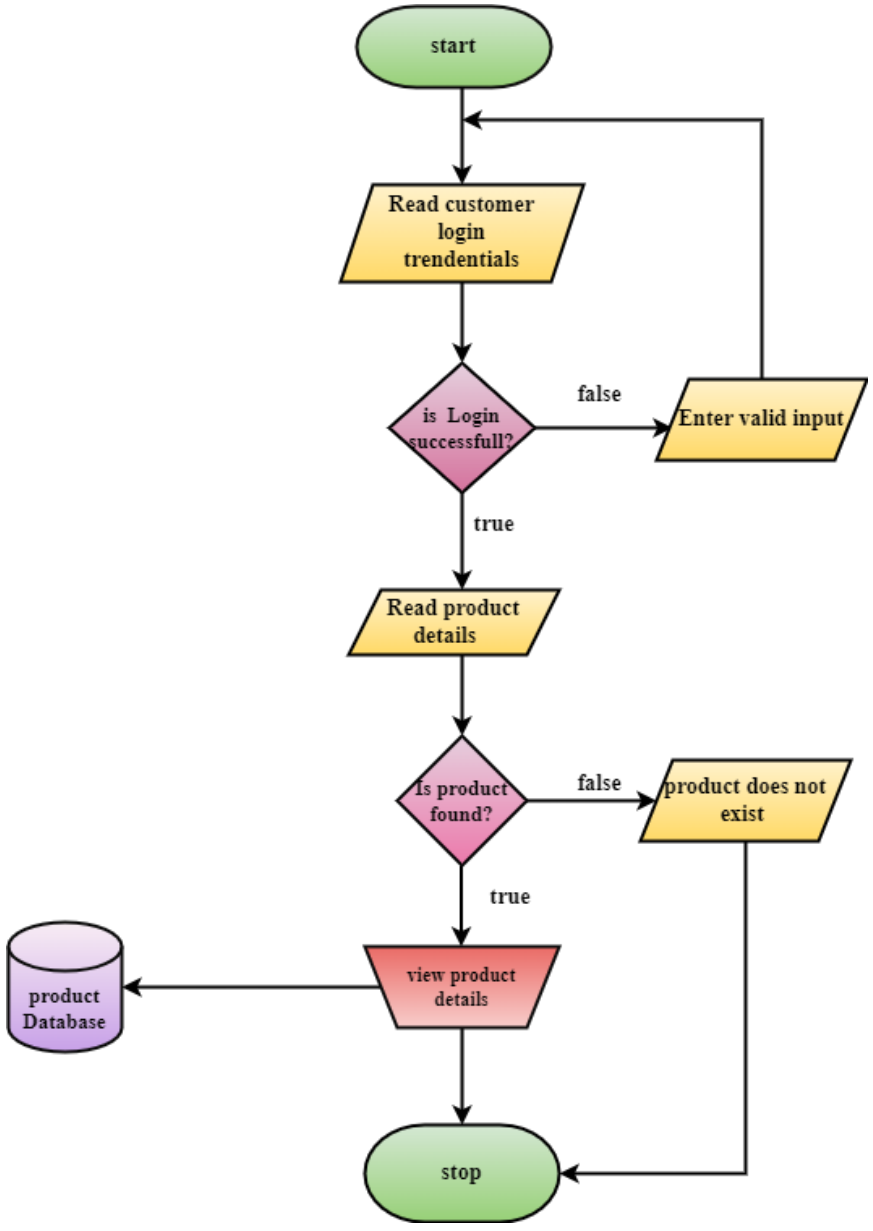

Figure 5.12: Detailed Design of user view product

- **Database I/O Interfaces:** System has well defined interface display user view product.
- **Output:** user view product stored in database

5.4.2.6: User add to cart

- **Inputs:** product details
- **Procedural Details** (Flow Chart):

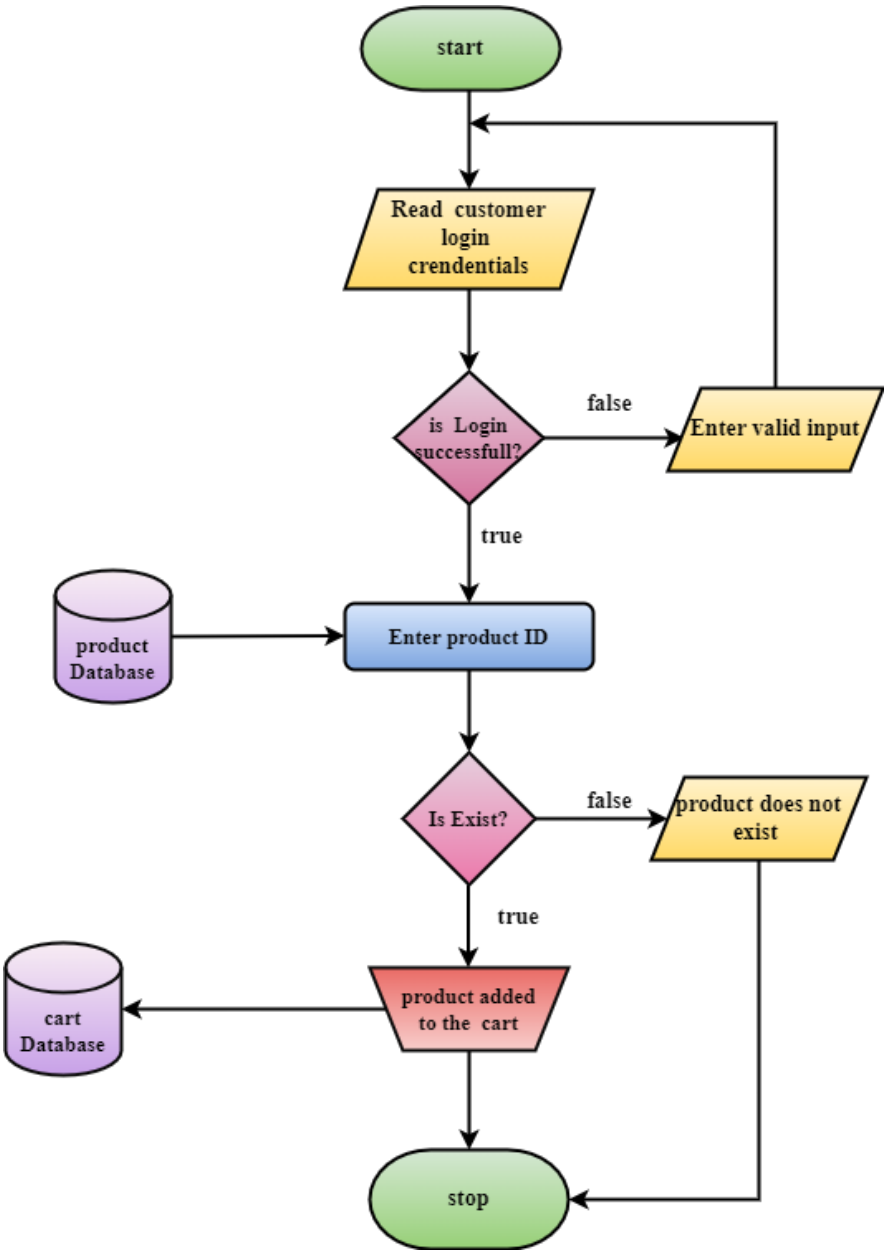

Figure 5.13 modular description of user add to cart

- **Database I/O Interfaces:** System has well defined interface display add to cart.
- **Output:** product added to the cart stored in database

5.4.2.7: User delete to cart

- **Inputs:** cart details
- **Procedural Details** (Flow Chart):

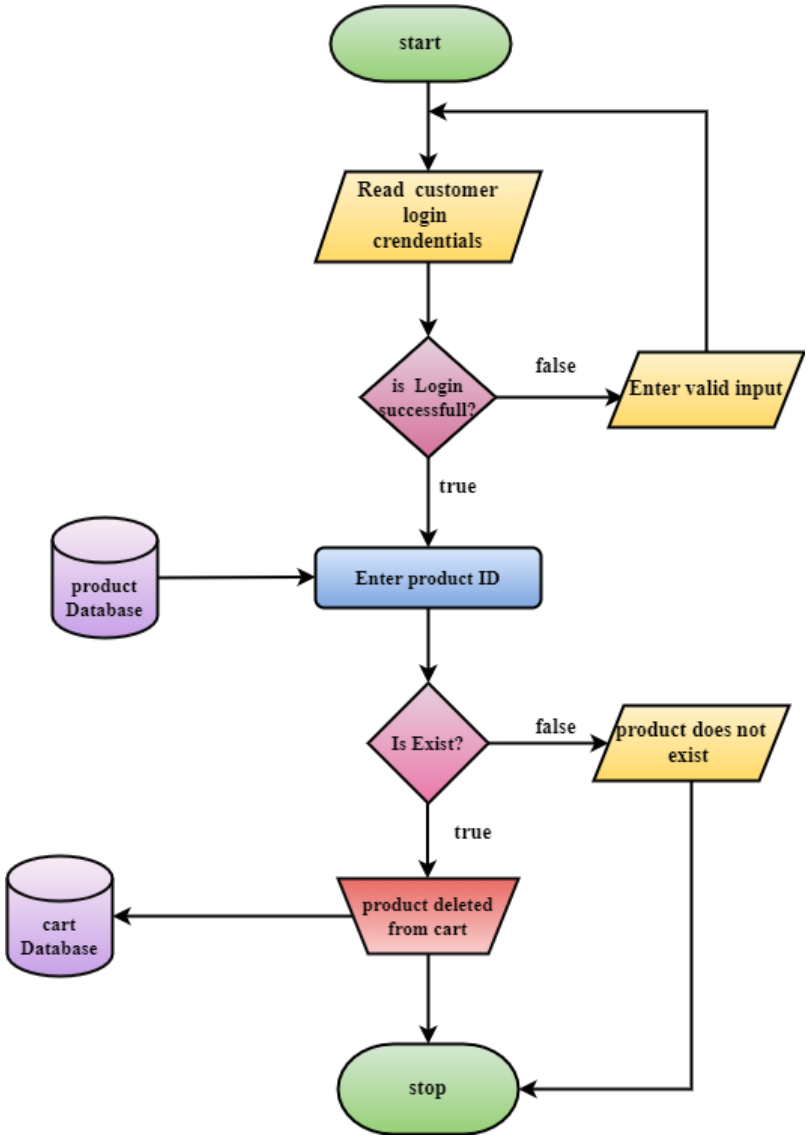

Figure 5.14 modular description of user delete from cart

- **Database I/O Interfaces:** System has well defined interface display delete from cart.
- **Output:** product deleted to the stored in database.

5.4.2.8: User view cart

- **Inputs:** cart details
- **Procedural Details** (Flow Chart):

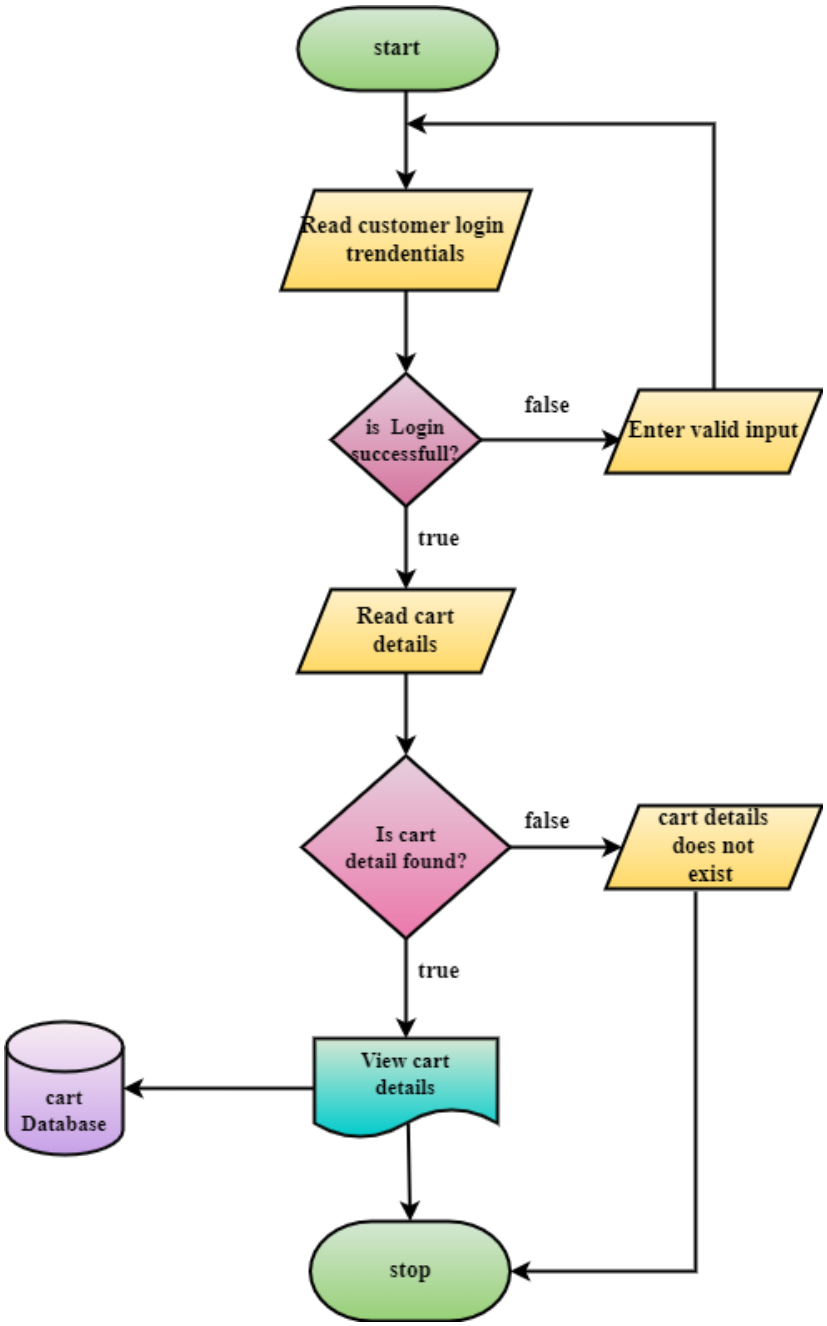

Figure 5.15 modular description of user view cart

- **Database I/O Interfaces:** System has well defined interface display view cart.
- **Output:** product view the stored in database

5.4.2.9: User add order

- **Inputs:** order details
- **Procedural Details** (Flow Chart):

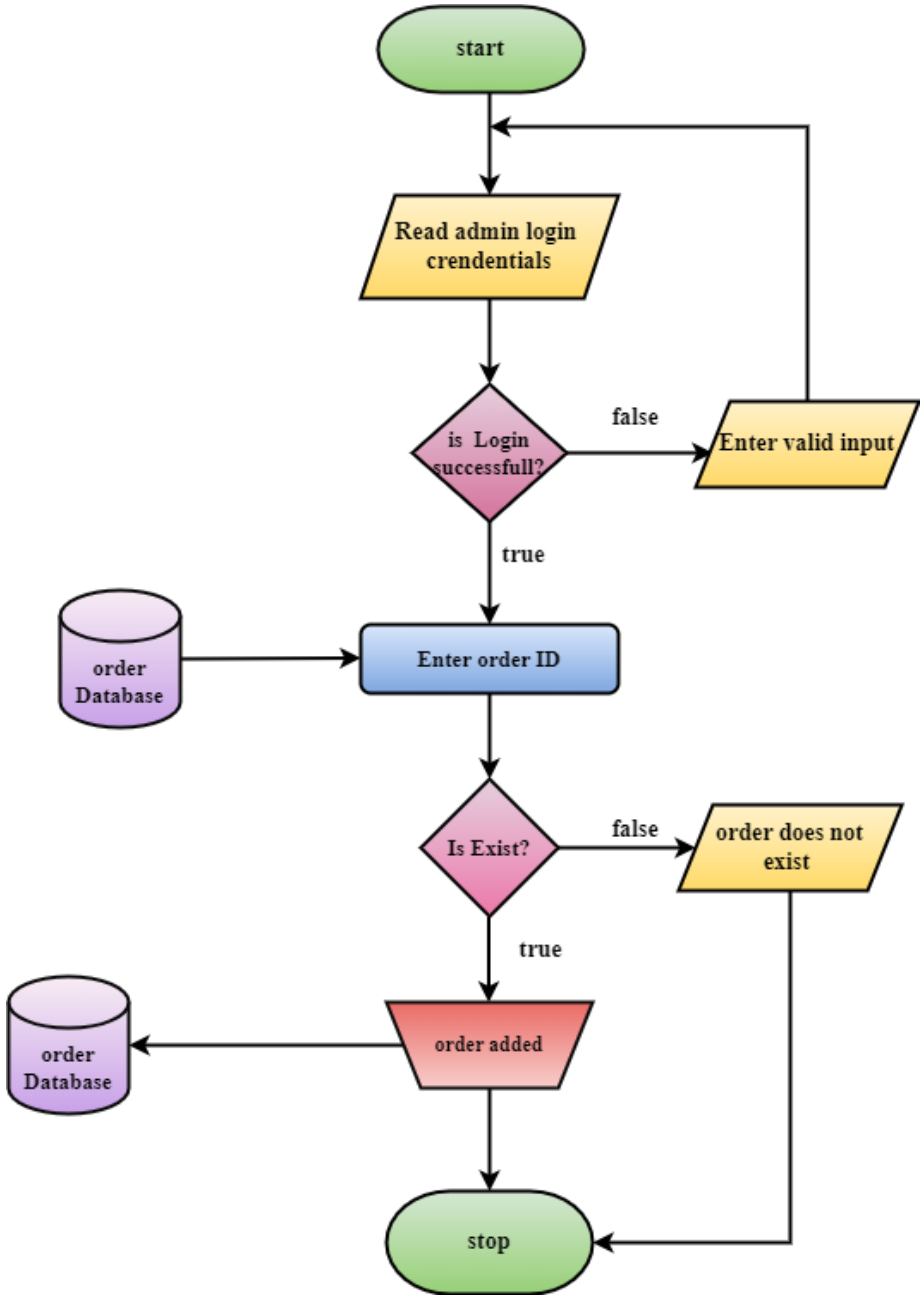

Figure 5.16 modular description of user add order

- **Database I/O Interfaces:** System has well defined interface display add order.
- **Output:** product order added to the stored in database

5.4.2.10: User view order

- **Inputs:** order details
- **Procedural Details** (Flow Chart):

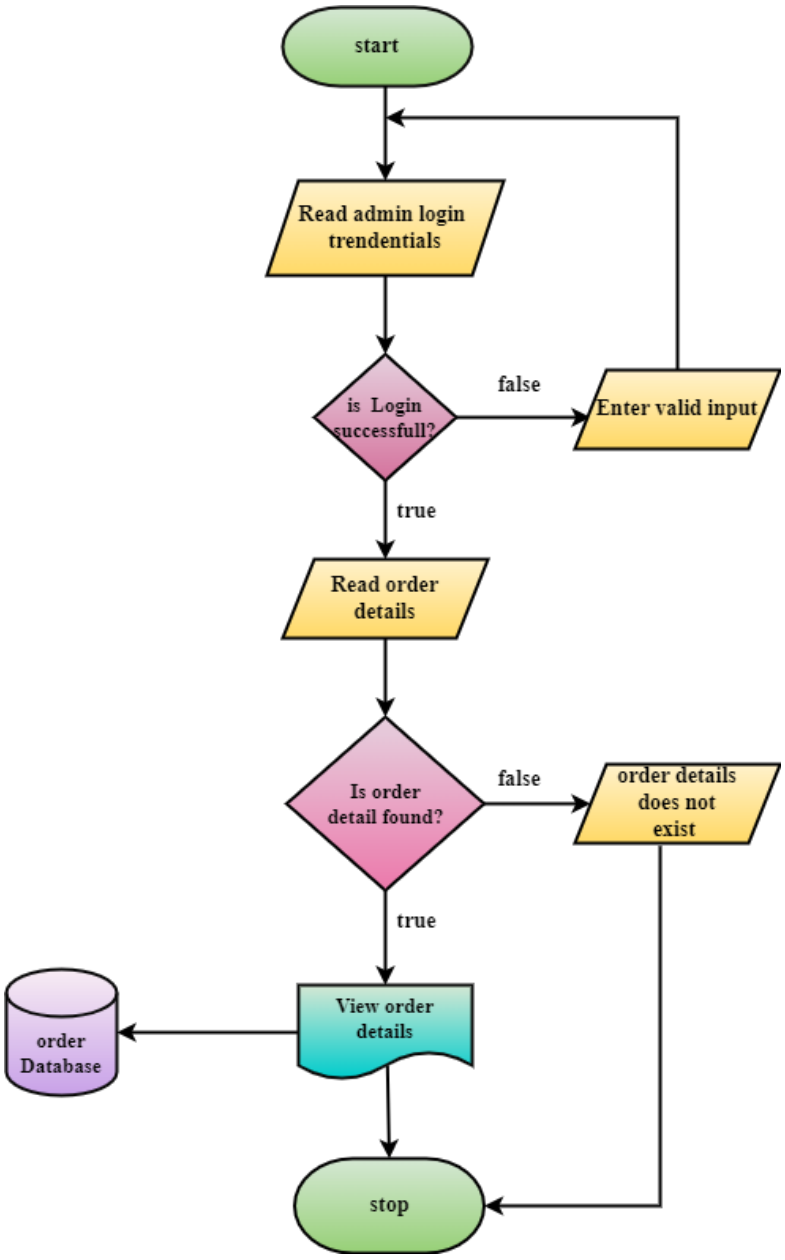

Figure 5.17 modular description of user view order

- **Database I/O Interfaces:** System has well defined interface display view order.
- **Output:** order view added to the stored in database

5.4.2.11: User Add Payment

- **Inputs**: payment details
- **Procedural Details** (Flow Chart):

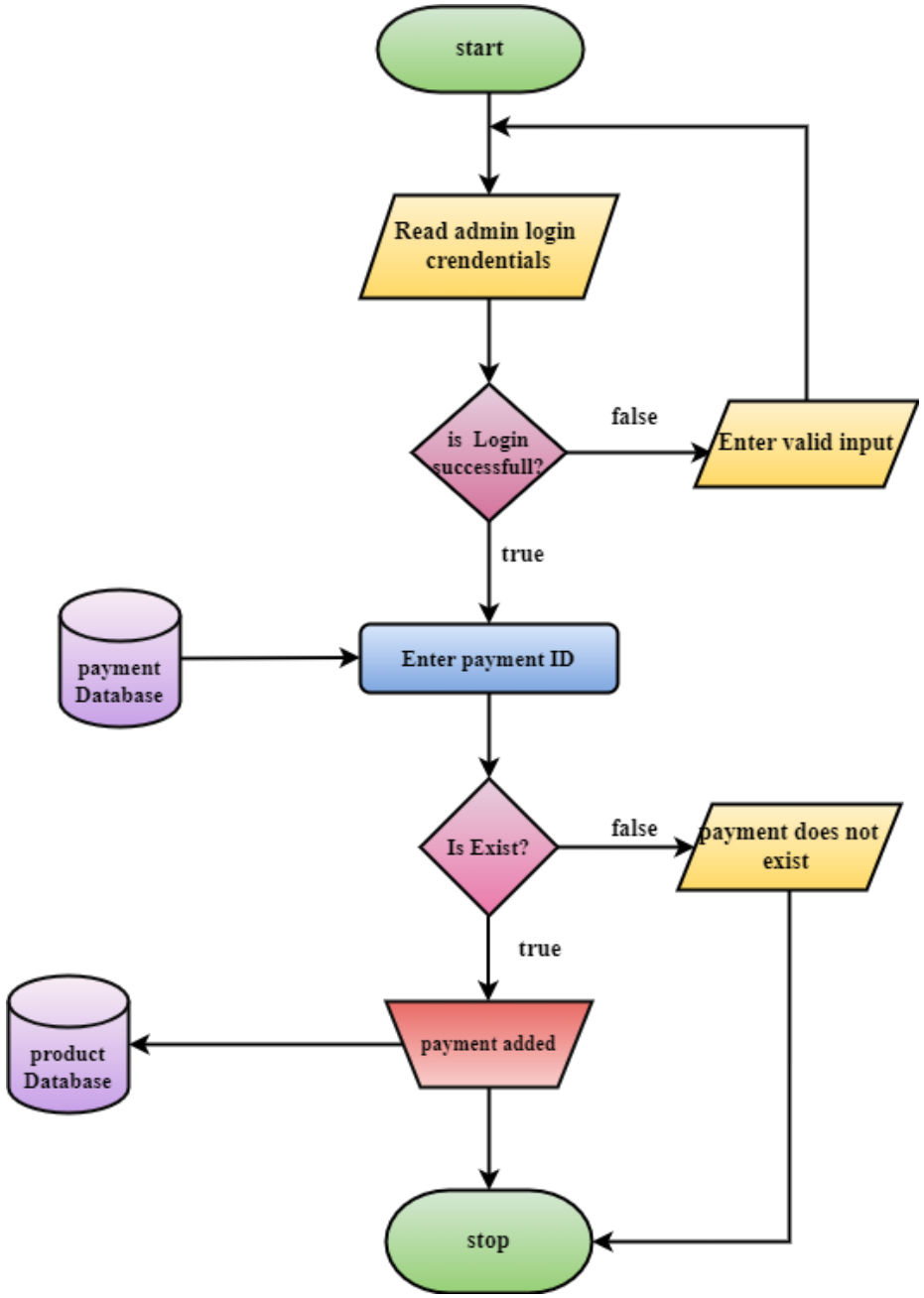

Figure 5.18 modular description of User Add Payment

- **Database I/O Interfaces:** System has well defined interface display add payment.
- **Output:** payment added to the stored in database

5.4.2.12: User View Payment

- **Inputs: payment details**
- **Procedural Details** (Flow Chart):

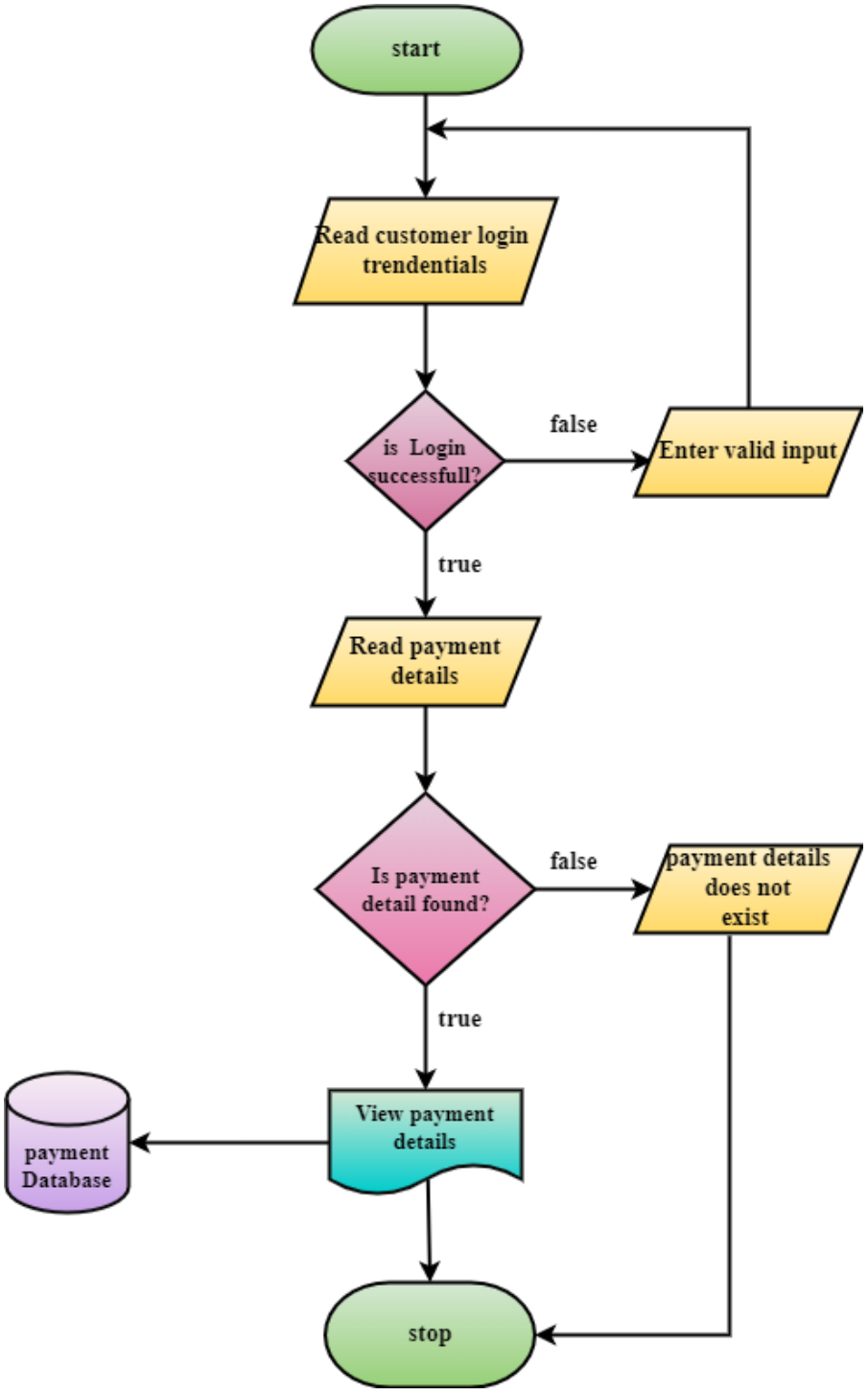

Figure 5.19 modular description of User View Payment

- **Database I/O Interfaces:** System has well defined interface display view payment.
- **Output:** payment added stored to the database

5.4.2.13: User track the delivery

- **Inputs:** order id, shipping_id, pr_name
- **Procedural Details** (Flow Chart):

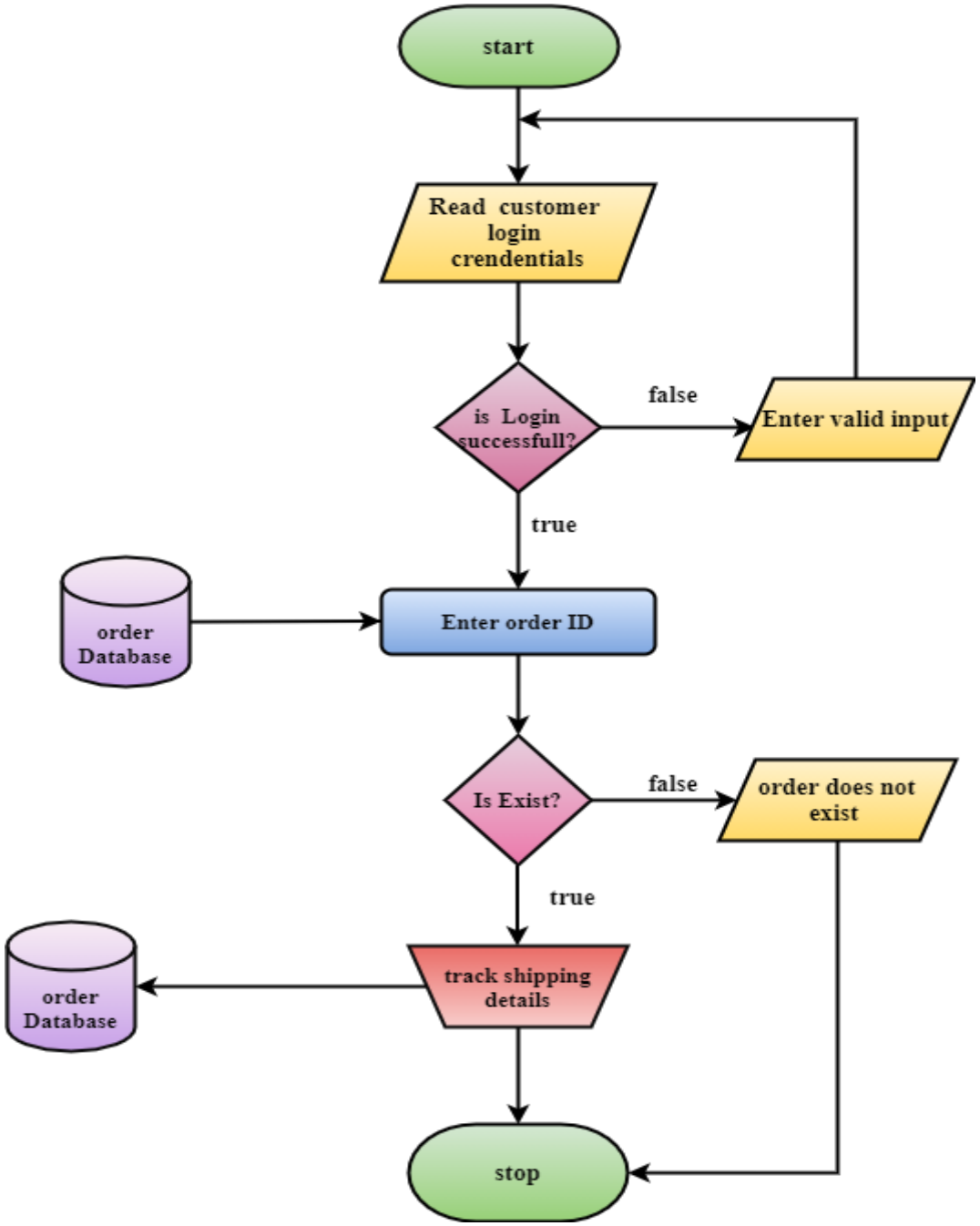

Figure 5.20 modular description of User track the Delivery

- **Database I/O Interfaces:** System has well defined interface display view cart.
- **Output:** product added stored to the database

5.4.2.14: User Delete Order

- **Inputs:** order details
- **Procedural Details (Flow Chart):**

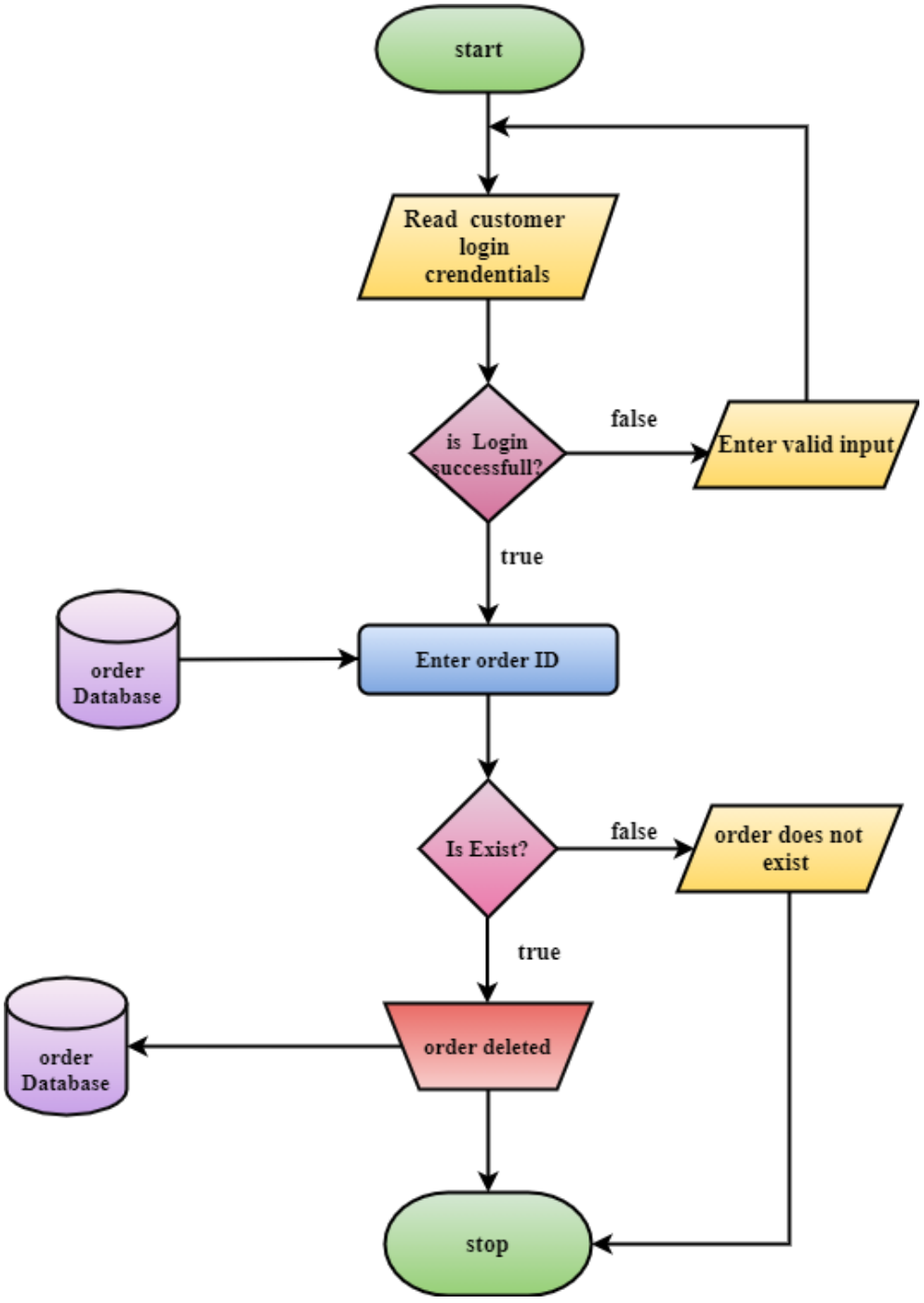

Figure 5.21 modular description of User Delete Order

- **Database I/O Interfaces:** System has well defined interface display view delete order.
- **Output:** order deleted from the stored database

5.4.2.15: User Write Feedback

- Inputs: feedback details, user_id
- Procedural Details (Flow Chart):

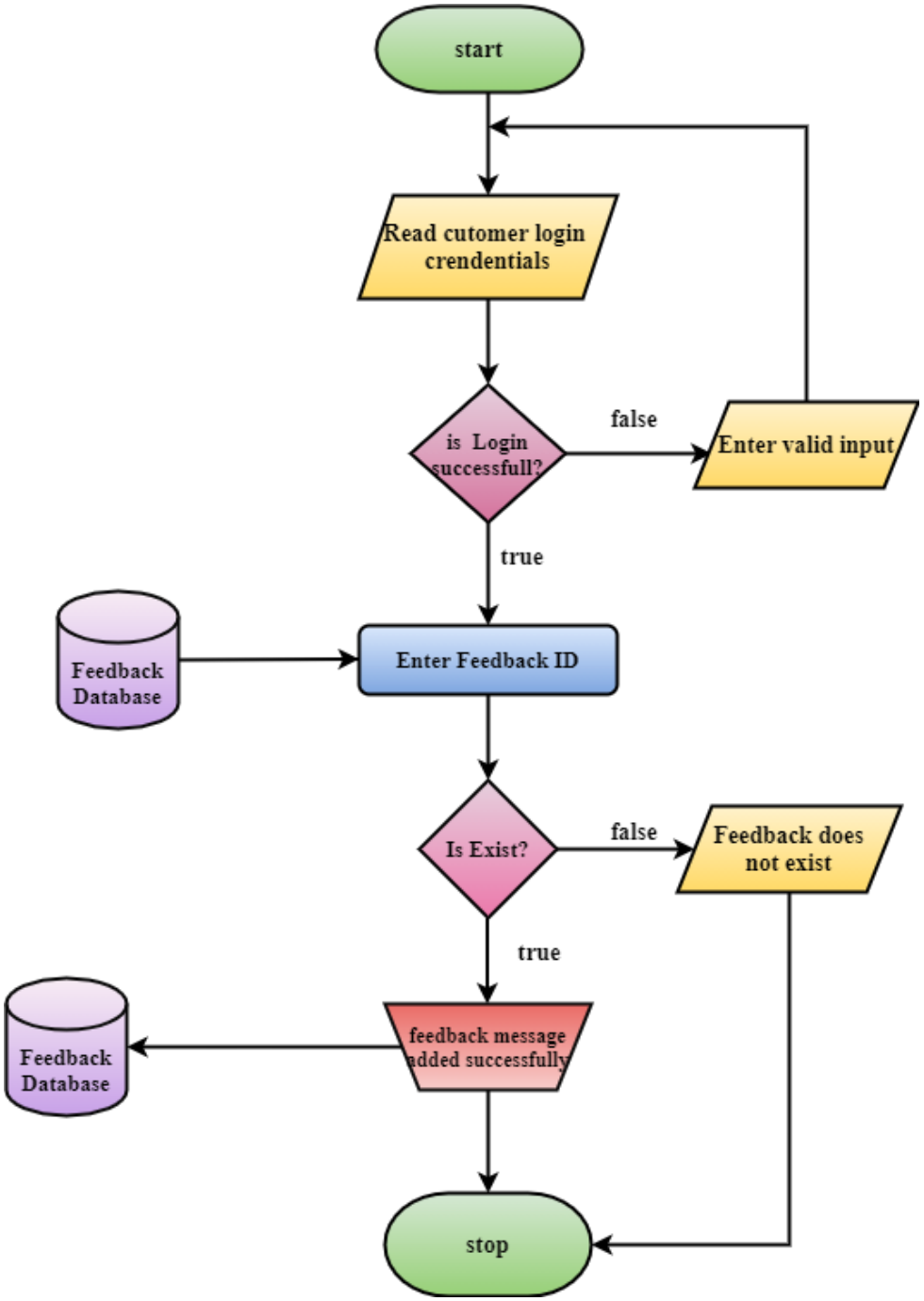

Figure 5.22 modular description of User Write Feedback

- **Database I/O Interfaces:** System has well defined interface display feedback message.
- **Output:** feedback added to the stored in database

5.4.2.16: User View the Feedback

- **Inputs:** c_id, feedback message
- **Procedural Details** (Flow Charts):

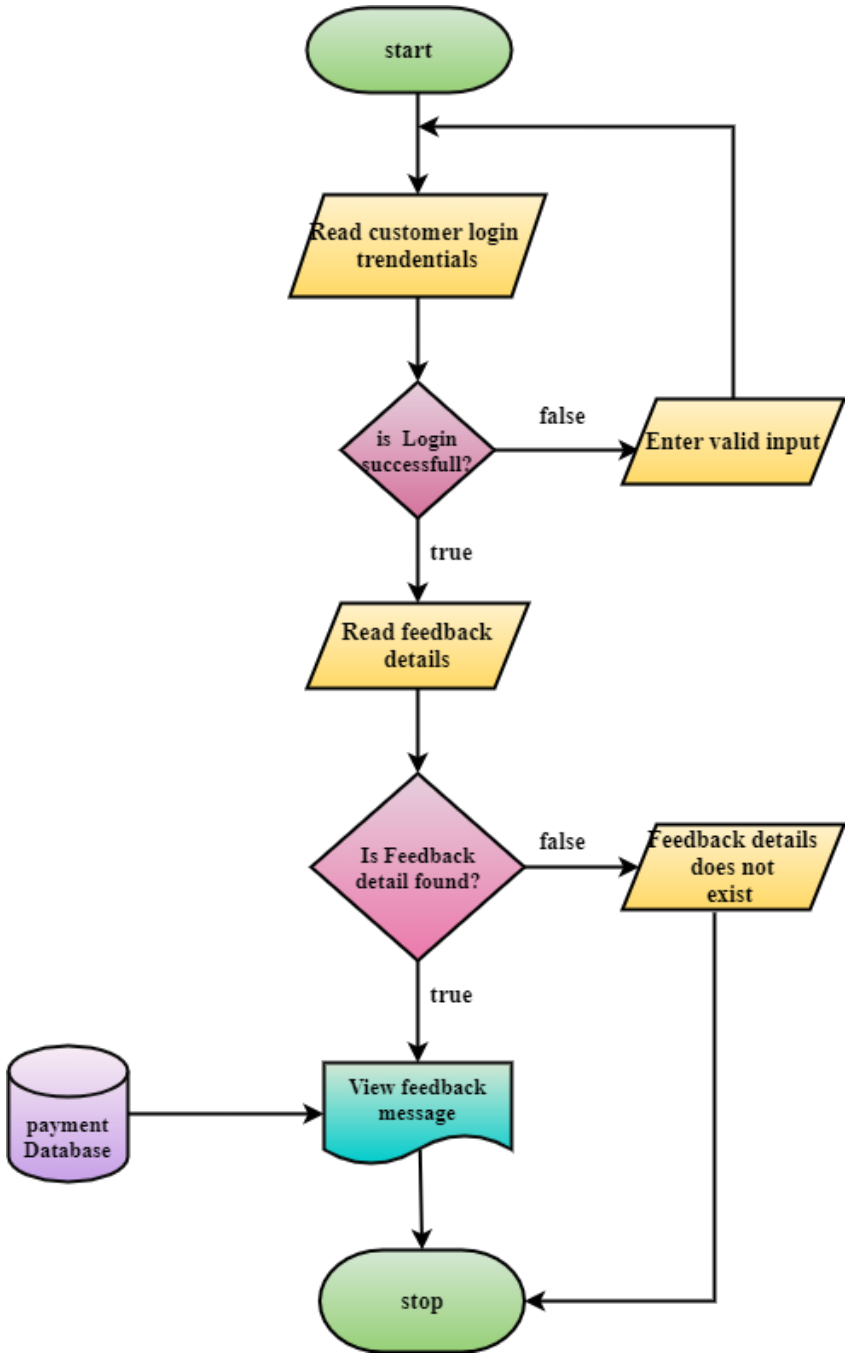

Figure 5.23 modular description of User View the Feedback

- **Database I/O Interfaces:** System has well defined interface display view feedback.
- **Output:** feedback details added to the stored in database

5.4.2.17: User deletes the Feedback

- **Inputs:** c_id, f_id, f_message
- **Procedural Details** (Flow Chart):

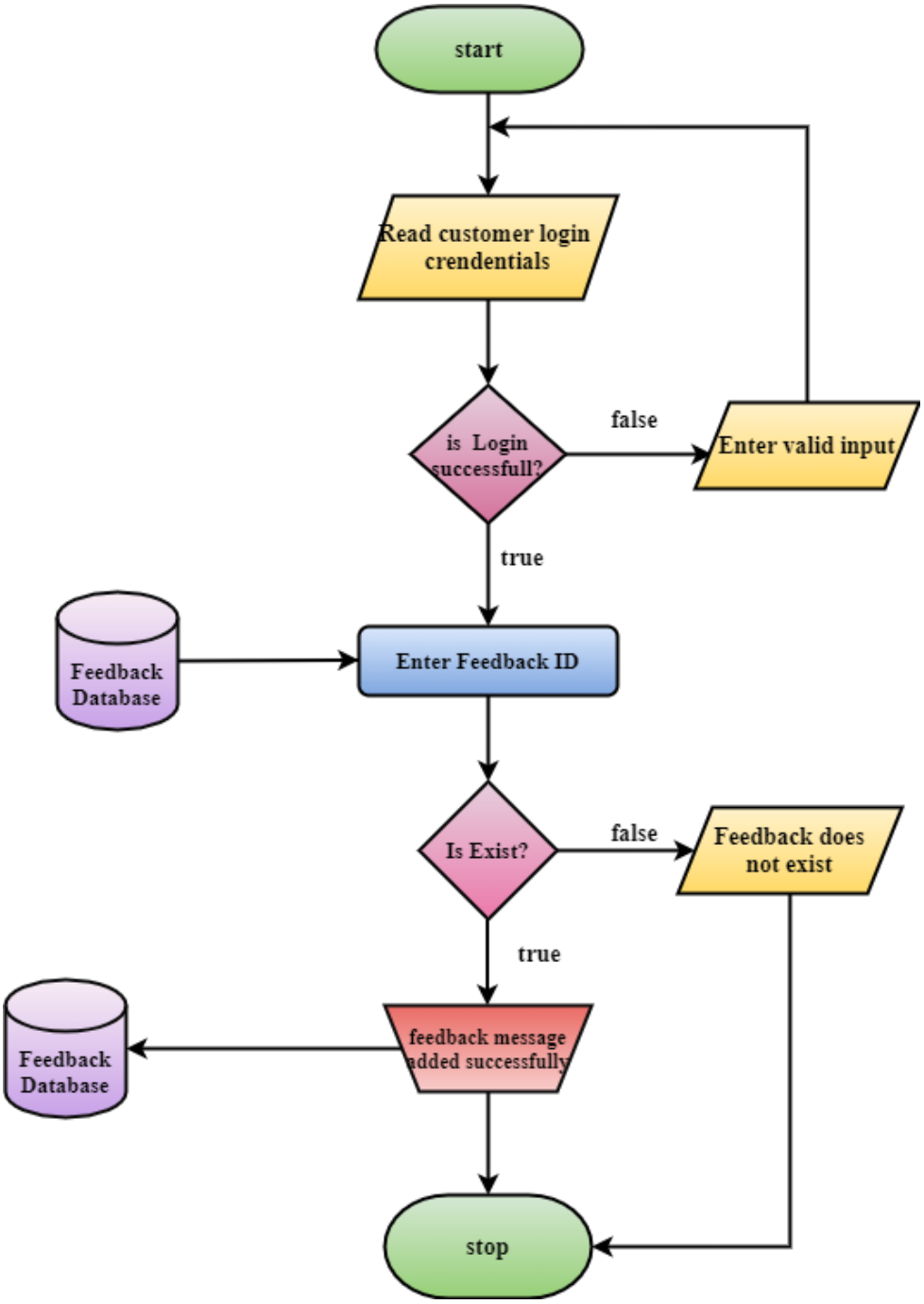

Figure 5.24 modular description of User deletes the Feedback

- **Database I/O Interfaces:** System has well defined interface display delete feedback.
- **Output:** feedback deleted from the stored in database.

5.3.3 Shop

5.3.3.1 Design assumptions

This module is designed in such a way that it allows the administrator to use any of the options easily so that he can add, modify or update any details. Admin module enables the admin to have control over the shop.

5.3.3.2 Identification of modules

- Manage Plants
- View Plants
- View Notification
- View Order
- View Payment

5.3.3.3 Structure chart showing the hierarchy of modules

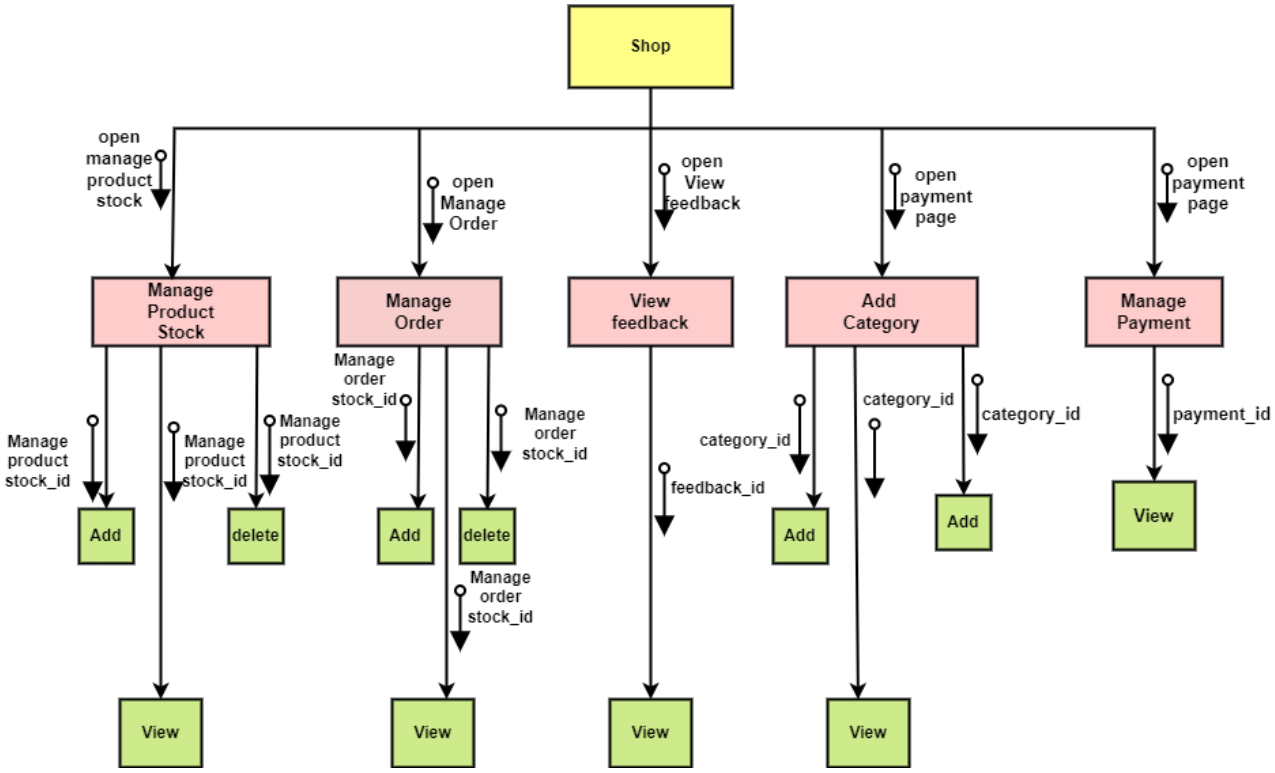

Figure 5.25 Structure chart of Shop

5.3.3.1 Shop Register

- **Inputs:** Username, Password
- **Procedural Details** (Flow Chart):

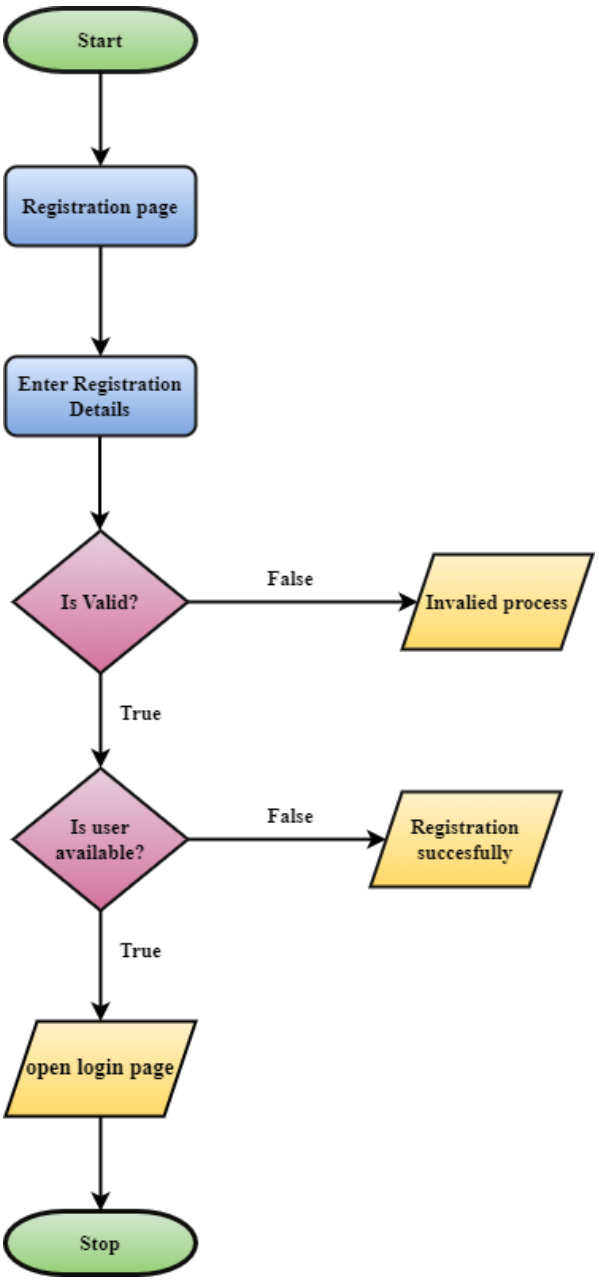

Figure 5.26 modular description of shop registration

- **Database I/O Interfaces:** System has well defined interface display shop registration.
- **Output:** user registration stored in database.

5.3.3.2 Shop Login

- **Inputs:** Username, Password.
- **Procedural Details** (Flow Chart):

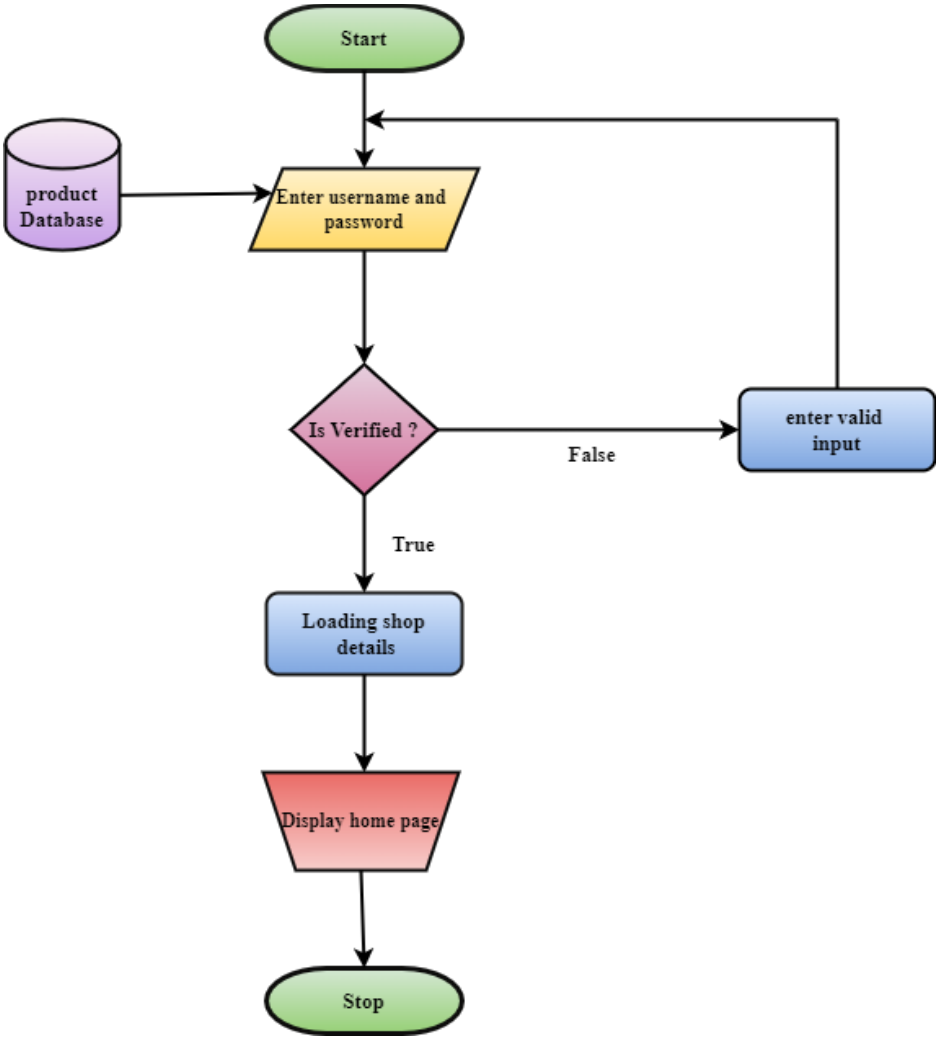

Figure 5.27 modular description of shop login

- **Database I/O interfaces:** System has interface to receive admin login.
- **Outputs:** Entered Username and password will be checked for validity if it is valid,Admin will be redirected to Homepage

5.3.3.3: Shop add product

- **Inputs:** Username, Password, pr_id.
- **Procedural Details** (Flow Chart):

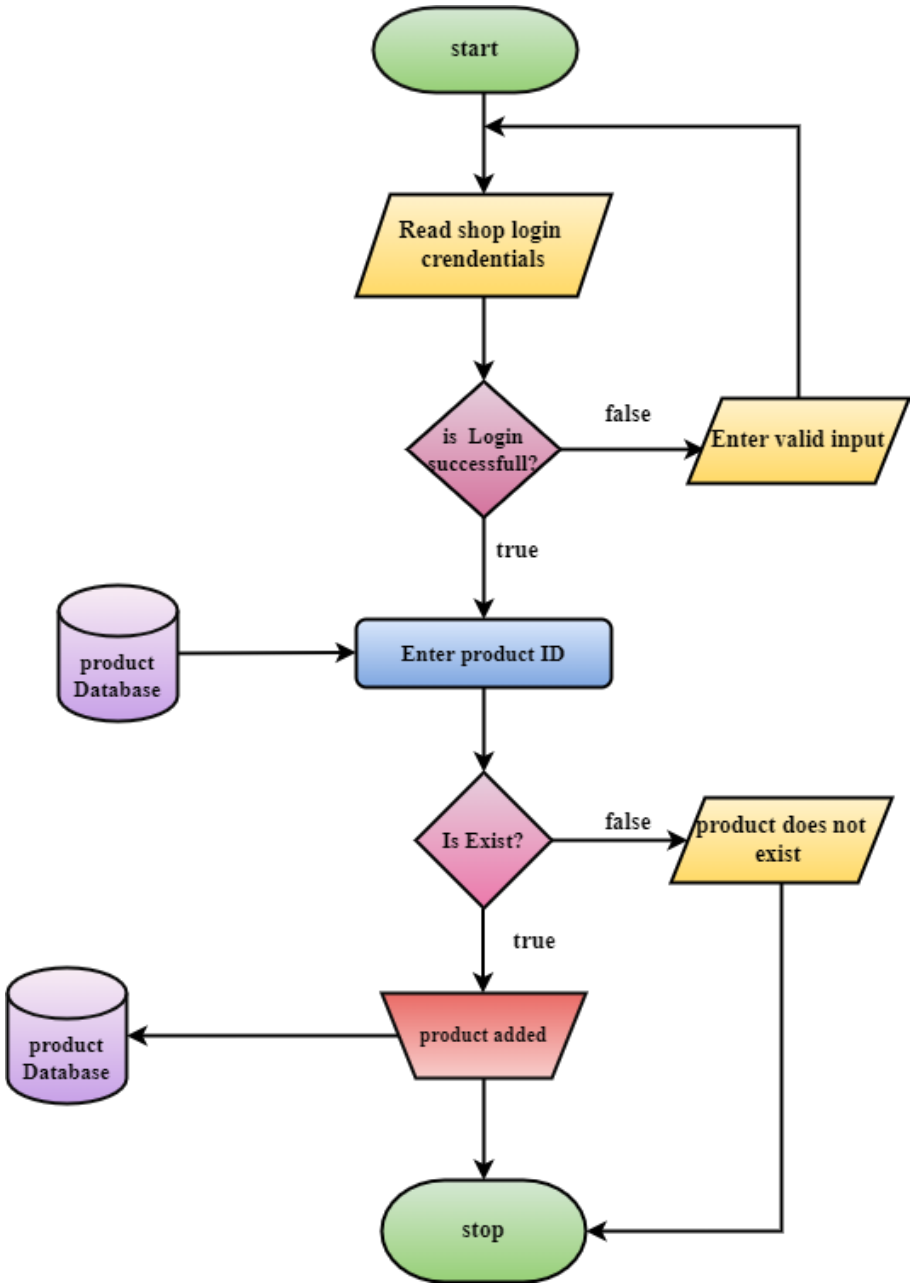

Figure 5.28 modular description of shop Add Product

- **Database I/O interfaces:** System has interface to receive add product.
- **Outputs:** Product added details are displayed.

5.3.3.4: Shop view product

- **Inputs:** Email Id, password, product ID.
- **Procedural Details** (Flow Chart):

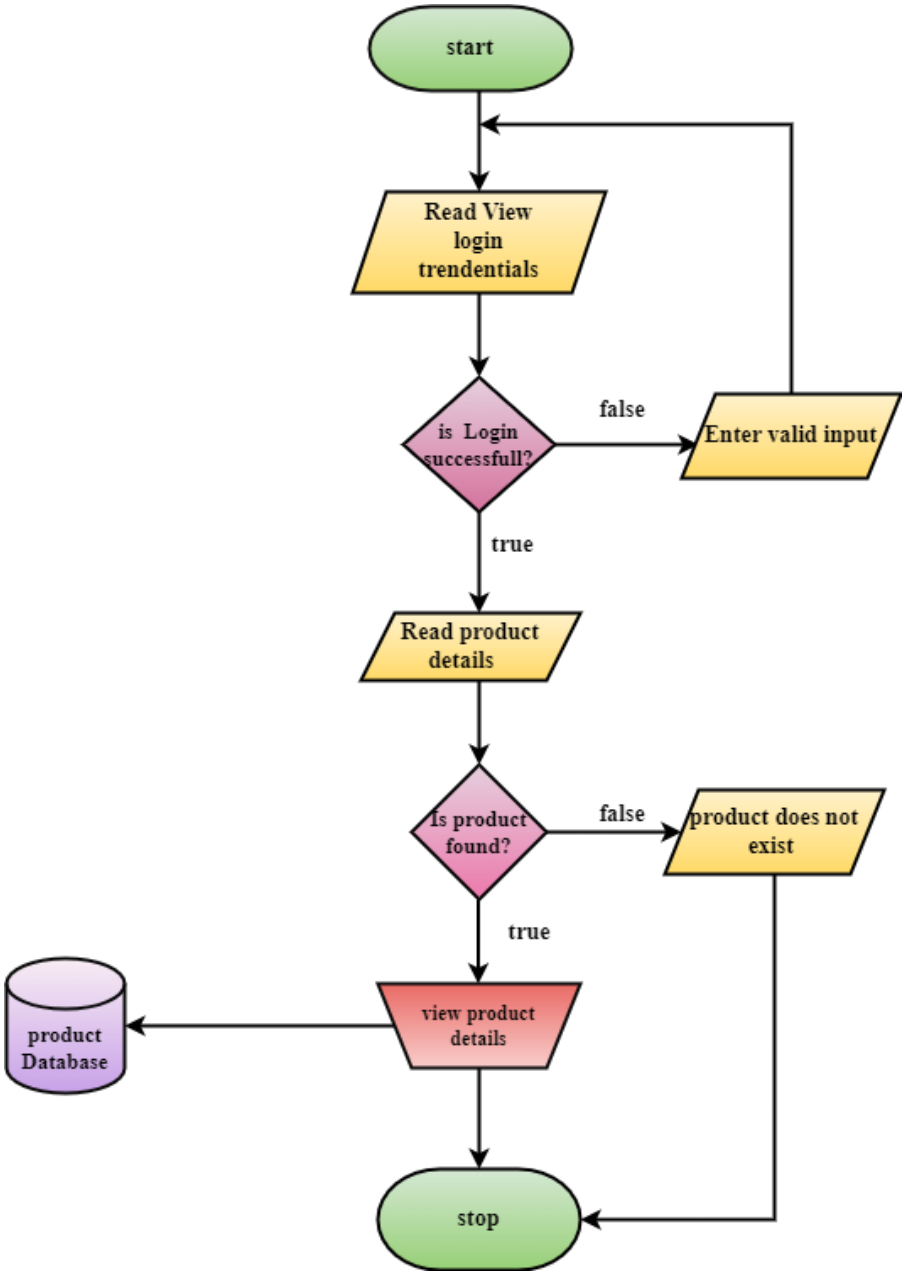

Figure 5.29 modular description of shop View Product

- **Database I/O Interfaces:** System has interface to receive product details.
- **Output:** Product details are displayed.

5.3.3.5: shop delete product

- **Inputs:** Email ID, password, product ID
- **Procedural Details** (Flow Chart):

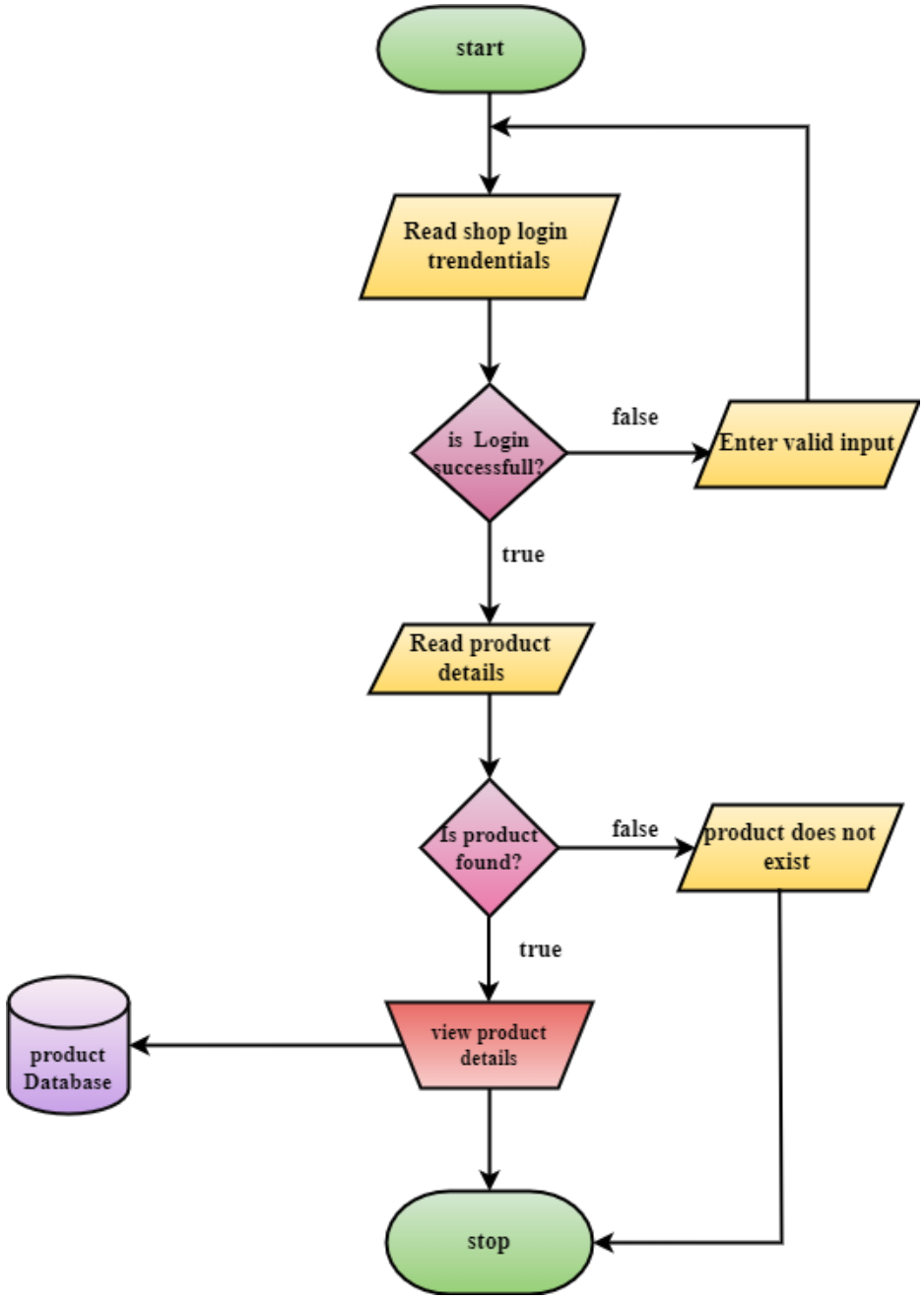

Figure 5.30: modular description of shop deletes Product

- **Database I/O Interfaces:** System has interface to shop delete the product.
- **Output:** shop can view the delete product details.

5.3.3.6: shop View order

- **Inputs:** Order details
- **Procedural Details** (Flow Chart):

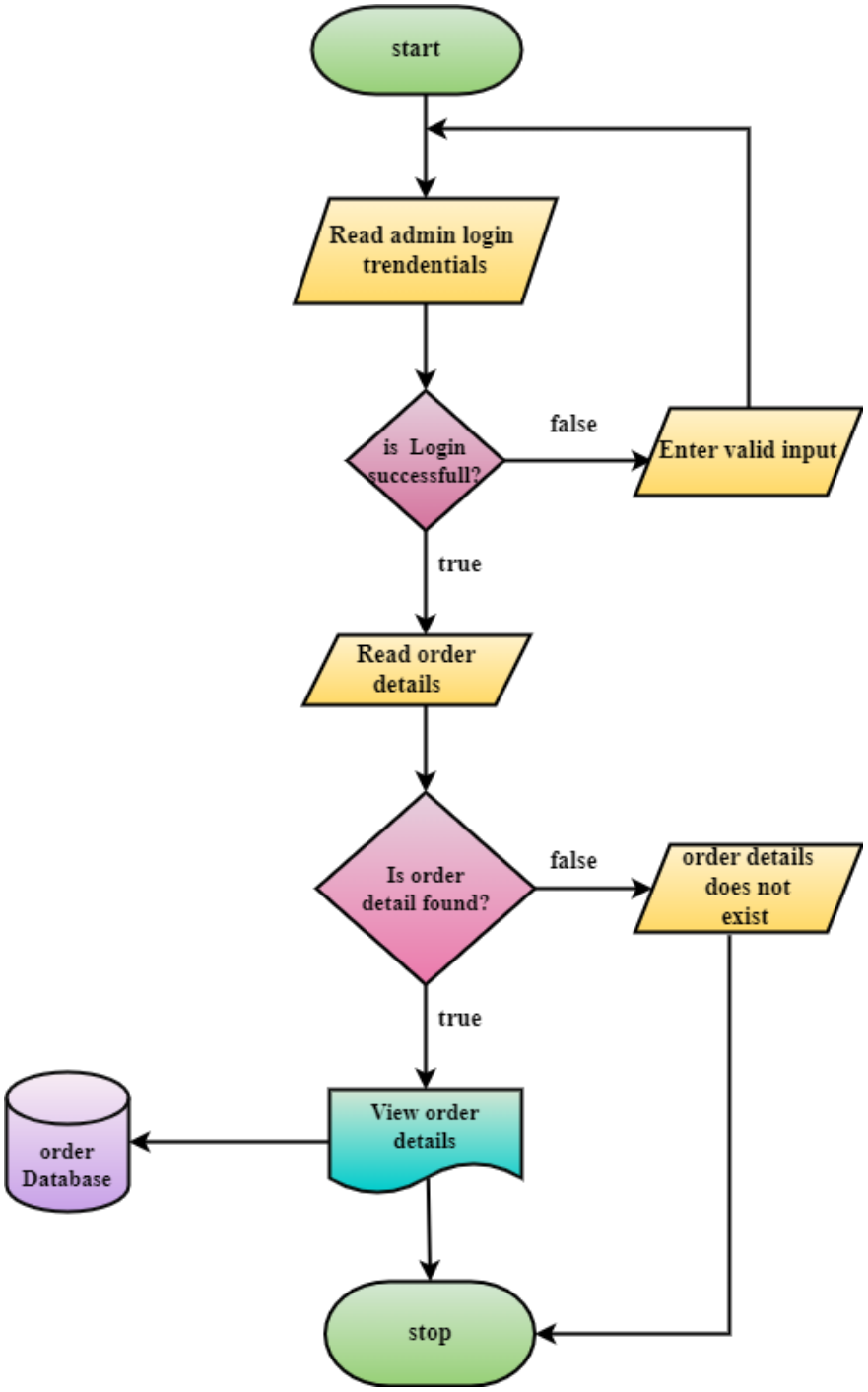

Figure 5.31 modular description of view Order

- **Database I/O Interfaces:** System has well defined interface for order.
- **Output:** Order data stored in database

5.3.3.6: shop Delete order

- **Inputs:** Order details
- **Procedural Details** (Flow Chart):

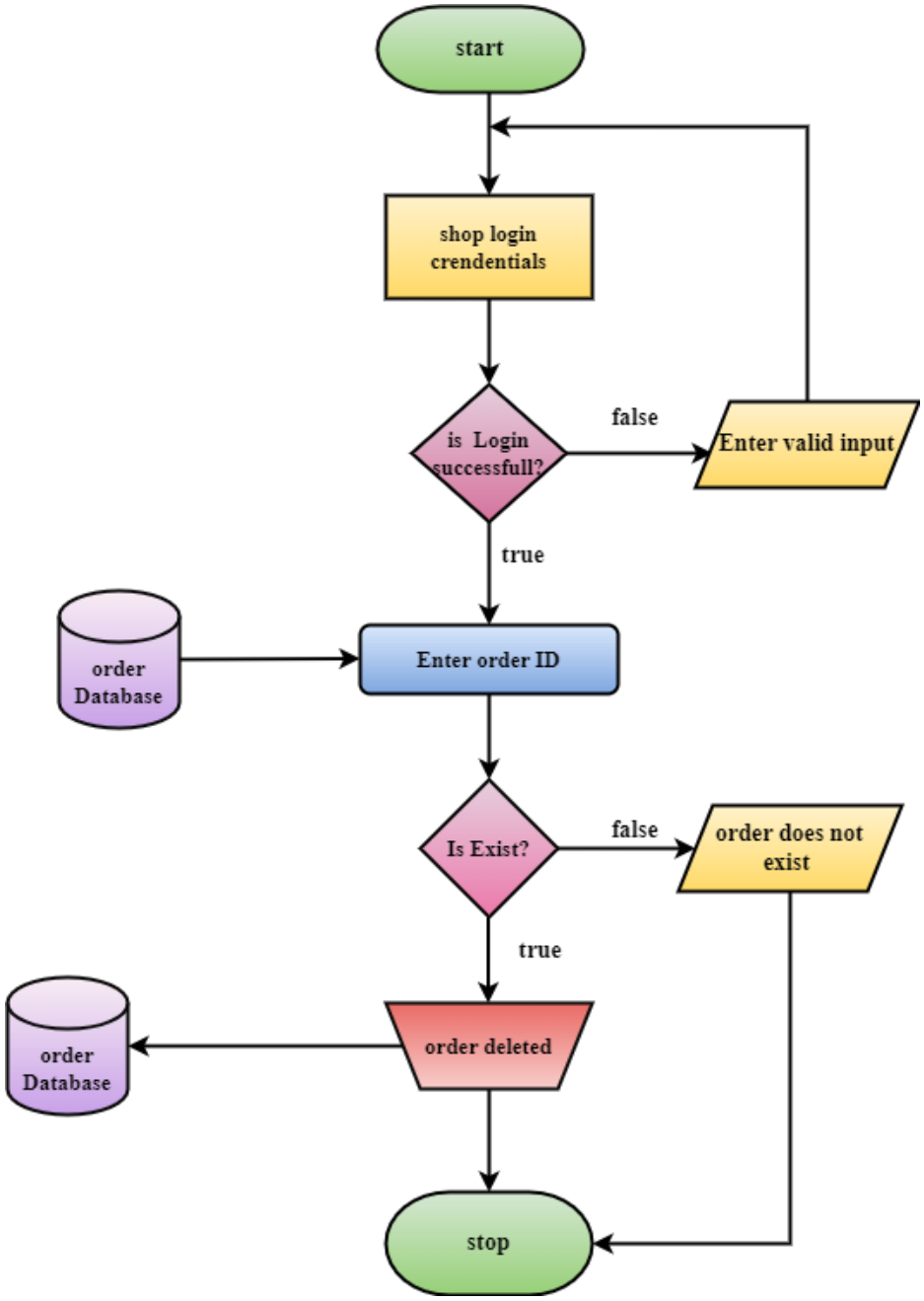

Figure 5.32 modular description of delete Order

- **Database I/O Interfaces:** System has well defined interface for order.
- **Output:** Order data stored in data

5.3.3.8: shop view payment

- **Inputs:** Payment details
- **Procedural Details** (Flow Chart):

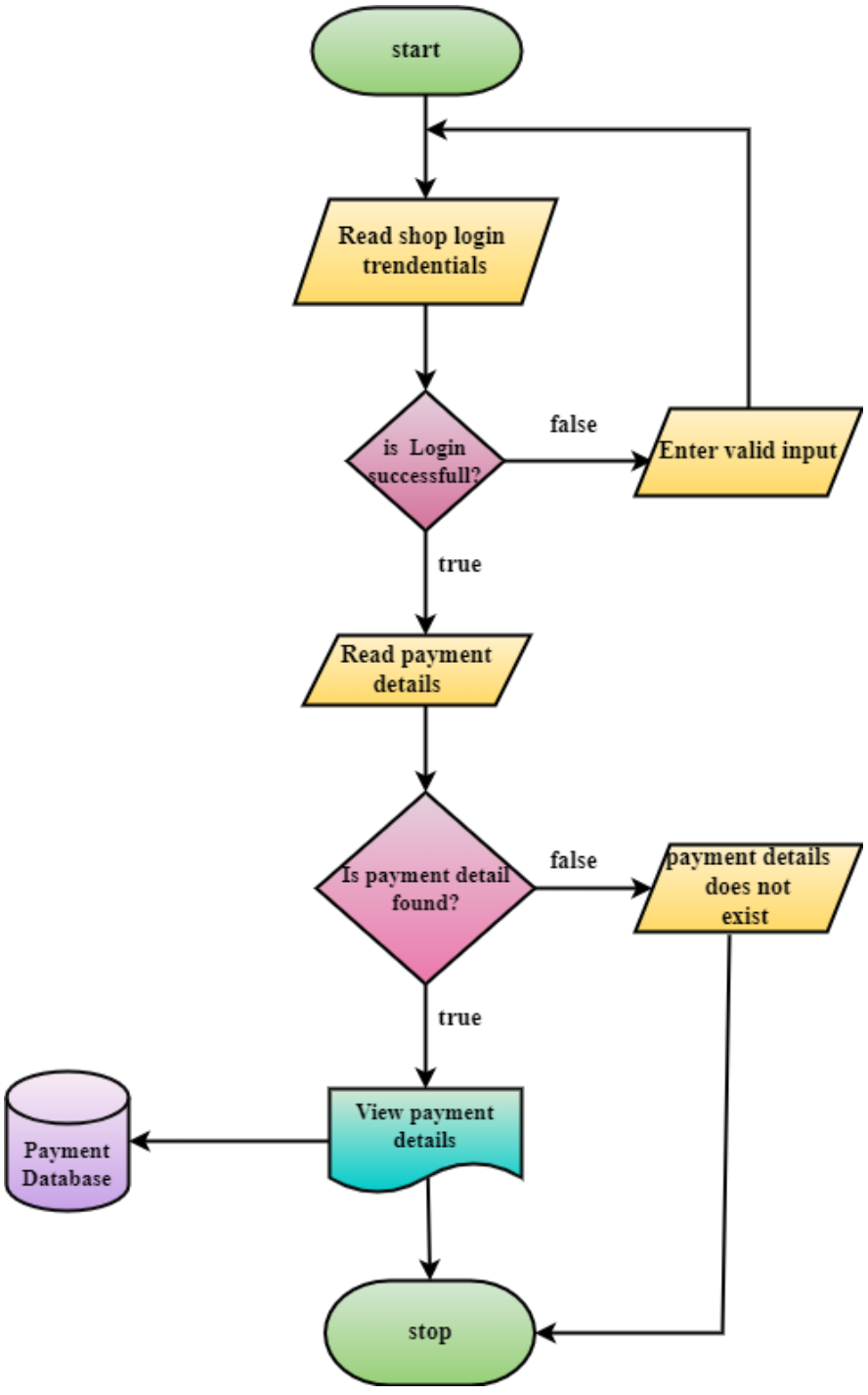

Figure 5.33 Detailed Design of view payment

- **Database I/O Interfaces:** System has well defined interface display payment.
- **Output:** payment stored in database.

5.3.3.9: Shop add category

- **Inputs:** Username, Password, c_id
- **Procedural Details** (Flow Chart):

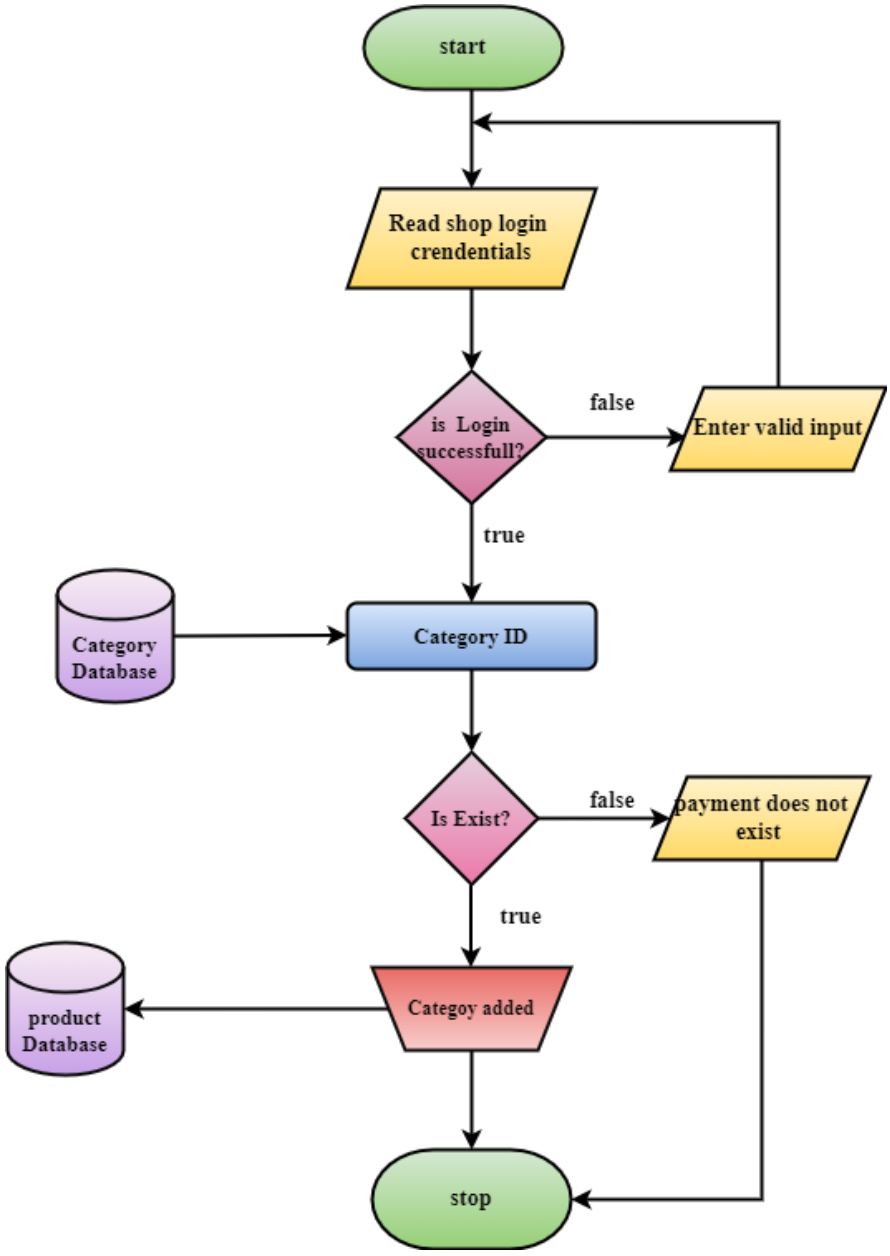

Figure 5.34 modular description of shop Add category

- **Database I/O interfaces:** System has interface to receive add category.
- **Outputs:** category added details are displayed.

5.3.3.10: Shop view Category

- **Inputs:** Email Id, password, category ID.
- **Procedural Details** (Flow Chart):

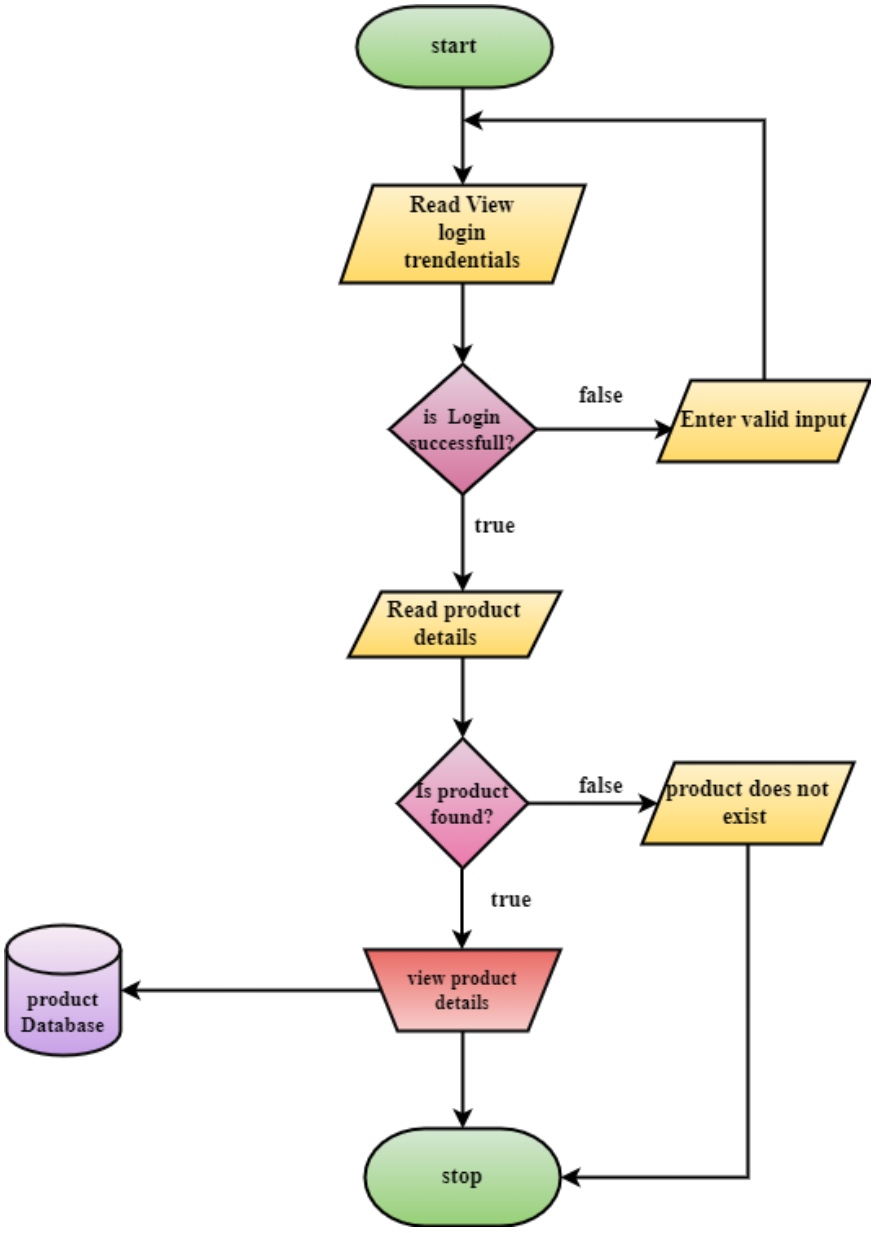

Figure 5.35 modular description of shop View category

- **Database I/O Interfaces:** System has interface to receive category details.
- **Output:** Product details are displayed.

5.3.3.11: shop delete category

- **Inputs:** Email ID, password, category ID
- **Procedural Details** (Flow Chart):

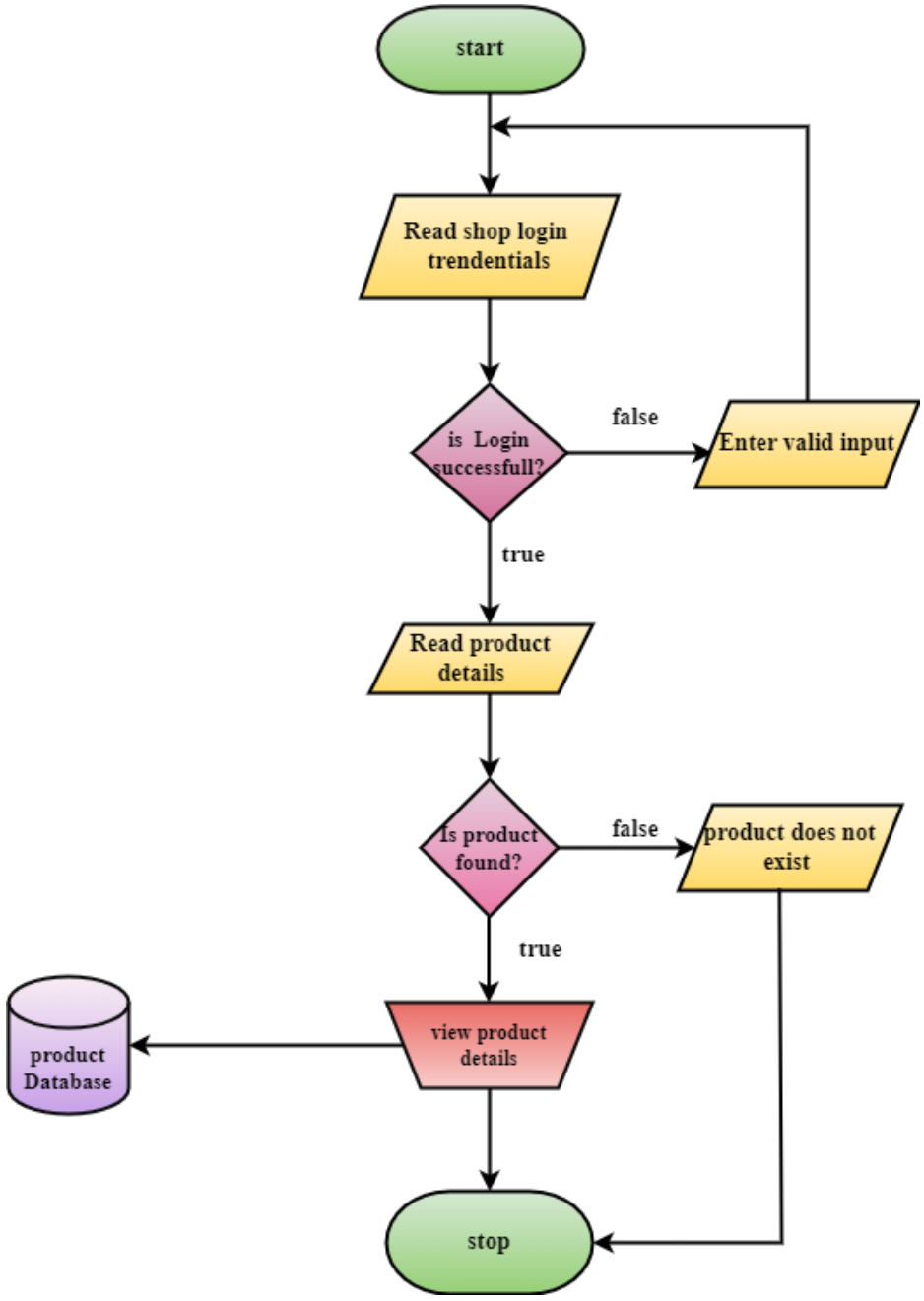

Figure 5.36: modular description of shop deletes category

- **Database I/O Interfaces:** System has interface to shop delete the category.
- **Output:** shop can view the delete category details.

5.3.3.12: Shop add report

- **Inputs:** Username, Password, pr_id
- **Procedural Details** (Flow Chart):

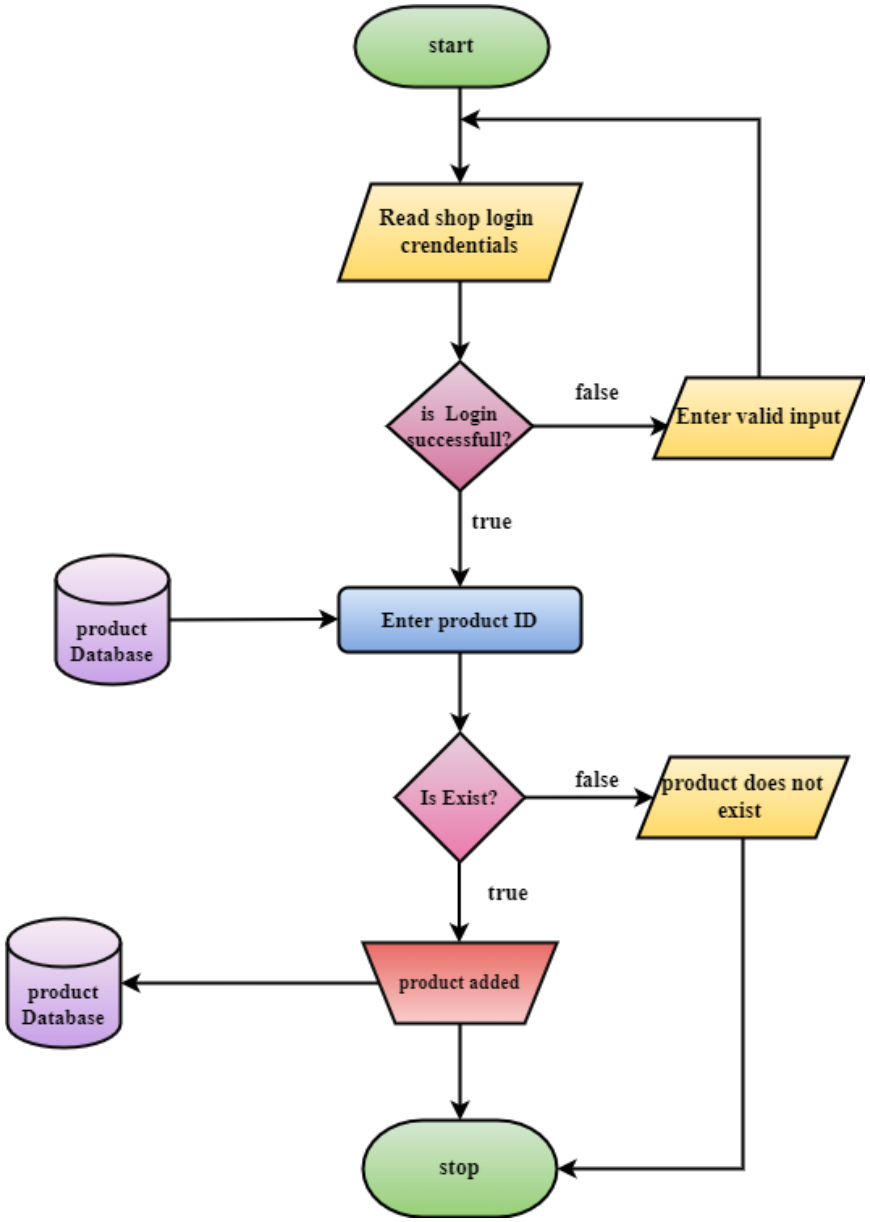

Figure 5.37 modular description of shop Add report

- **Database I/O interfaces:** System has interface to receive add report
- **Outputs:** report added details are displayed.

5.3.3.13: Shop view Report

- **Inputs:** Email Id, password, product ID.
- **Procedural Details** (Flow Chart):

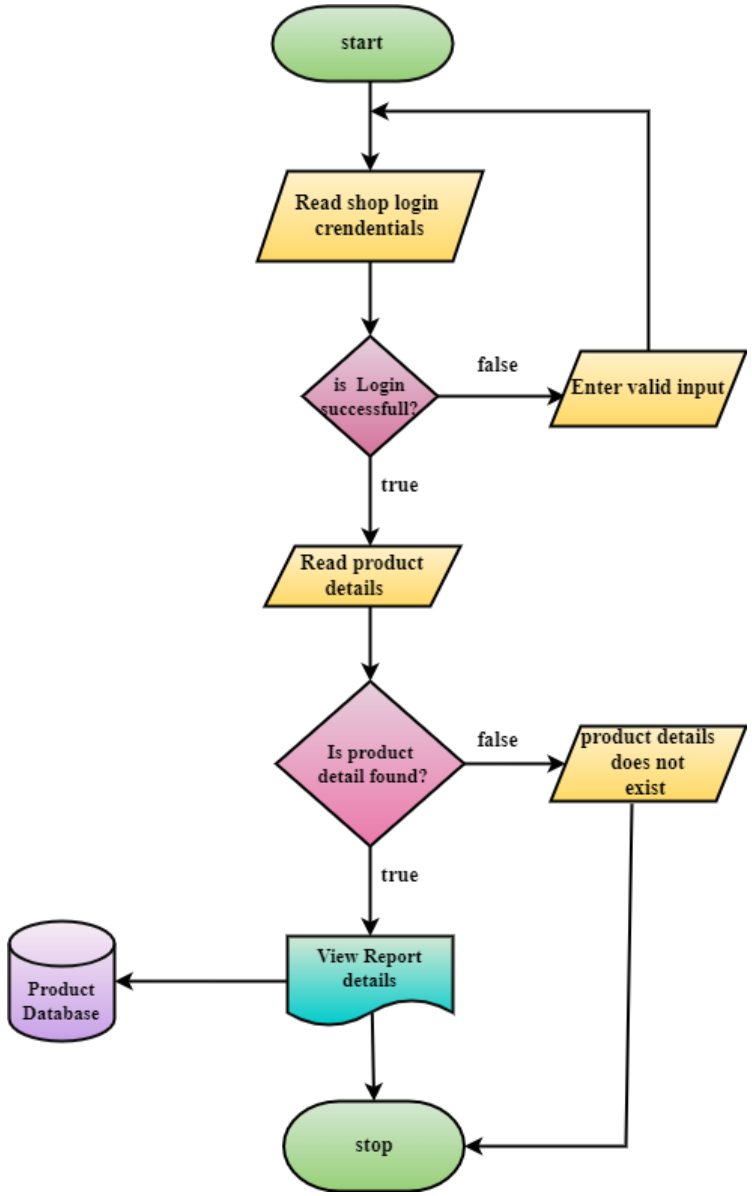

Figure 5.38 modular description of shop View report

- **Database I/O Interfaces:** System has interface to receive report details.
- **Output:** report details are displayed.

CHAPTER-6

PROGRAM CODING LISTING

Introduction

The goal of the coding or program phase is to translate the designerof the system are can produced during the design phase into code in a given programming language which can be executed by a computer and that performs the computation specified by the design. The coding phase affects both testing and maintenance profoundly. As we saw earlier the time spent in coding is a small percentage of the total software cost while testing and coding maintenance consume the major percentages. Thus, it should be clear that the goal during coding should not be to reduce the implementation cost but the goal should to be reduce the cost of later phases even if means that a cost of this phase has to increase. In other words, the goal during this phase is not to simplify the job of programmer.Rather the goal should be to simplify the job of tester and maintainer.

The coding is done with the following characteristics:

- Ease of Design to Code translation
- Code Efficiency
- Memory Efficiency
- Response Time
- Maintainability
- Security
- Simple Ease to Understand Code
- Efficient and Consistent Logic

6.1 Database Connection

```
<?php ob_start();

session_start();


//-----database connection variable-------

define('DB_SERVER', "localhost");
```

```
define('DB_USER', "root");

define('DB_DATABASE', "galaxy_fashion");

 define('DB_PASSWORD', "");

define('DB_TYPE', 'mysql');

define('SITE_TITLE', 'demo.com');

// tag line of the project

define('SITE_TAG_LINE', 'give your tag line of your project here');


 //Site contact information

define('SITE_CONTACT', 'your number');

//email  information

define('SITE_EMAIL_INFO', 'your mail id');

define('BASE_URL', 'http://localhost/cud opertaion/');


// included main class

require_once 'app/Main.php'; require_once

'app/Admin.php'; require_once

'app/Controller.php';


/**

* @param $class

*
```

6.2 Authorization/Authentication

6.2.1 Login

```
<?php
include '../config.php';
$admin=new Admin();
if(isset($_POST['Login']))
{
// username and password giving
   $name=$_POST['name'];
   $password=$_POST['Password'];
   $stmt=$admin->ret("SELECT * FROM `admin` WHERE `a_name`='$name' ");
   $num=$stmt->row Count ();
      if($num>0){
         $row=$stmt->fetch(PDO::FETCH_ASSOC);
         $dbpassword=$row['a_password'];
         if(password_verify($password,$dbpassword)){
            $_SESSION['aid']=$row['a_id'];
            echo "<script>alert('Login Successfull');window.location='../index.php';</script>";
         }else{
            echo "<script>alert('You have entered wrong
   password');window.location='../login.php';</script>";
         }
      }else{
         echo "<script>alert('You are not Valid USer');window.location='../login.php';</script>";
      }
   }
   ?>
```

6.2.2 Shop login

```
<?php

include '../config.php';

$admin=new Admin();

//giving login information

if(isset($_POST['Login']))
{
   $email=$_POST['Email'];
   $password=$_POST['Password'];

        $stmt=$admin->ret("SELECT * FROM `shop` WHERE `o_email`='$email' AND `ps_status`='Accepted'");
   $num=$stmt->rowCount();

   if($num>0){

      $row=$stmt->fetch(PDO::FETCH_ASSOC);

      $dbpassword=$row['ps_password'];

      if(password_verify($password,$dbpassword)){

         $_SESSION['psid']=$row['ps_id'];

         echo "<script>alert('Login Successfull');window.location='../index.php';</script>";

      }else{

       echo"<script>alert('You have entered wrong b password');window.location='../login.php';</script>";

      }

}else{

                    echo "<script>alert('You are not Approved By Admin');window.location='../login.php';</script>";
```

```
    }
}
?>
```

6.2.3 User login

```
<?php
include '../config.php';
$admin=new Admin();
if(isset($_POST['login'])){
    $email=$_POST['email'];
    $password=$_POST['password'];
    $stmt=$admin->ret("SELECT * FROM  `user` WHERE `u_email`='$email'");
    $num=$stmt->rowCount();
    if($num>0){
        $row=$stmt->fetch(PDO::FETCH_ASSOC);
        $dbpass=$row['u_password'];
        if(password_verify($password,$dbpass)){
            $_SESSION['uid']=$row['u_id'];
            echo "<script>alert('Login Successfull ');window.location='../index.php';</script>";
        }else{
            echo "<script>alert('You Have Entered Worng
Password');window.location='../login.php';</script>";
        }
    }else{
        echo "<script>alert('You are Not a Valid User');window.location='../login.php';</script>";
```

```
    }
        }
    ?>
```

6.3 Data Store/Retrieval/Update

6.3.1 Add shop by Admin

```
<?php
include '../config.php';
$admin=new Admin();
if(isset($_GET['psid'])){
    $ps_id=$_GET['psid'];
    $stmt=$admin->cud("UPDATE `shop` SET `ps_status`='Accepted' WHERE
`ps_id`='$ps_id'",'updated');
    echo "<script>alert('Accepted Successfull');window.location='../managepetshop.php'</script>";
}
if(isset($_GET['ps_id'])){
    $ps_id=$_GET['ps_id'];
    $stmt=$admin->cud("UPDATE `shop` SET `ps_status`='Rejected' WHERE
`ps_id`='$ps_id'",'updated');
    echo "<script>alert('Rejected  Successfull');window.location='../managepetshop.php'</script>";
}
?>
```

6.3.2 Add product by shop

```
<?php
include '../config.php';
$admin=new Admin();
$psid=$_SESSION['psid'];
if(isset($_POST['product'])){
    $pet_id=$_POST['maincat_id'];
    $cat_id=$_POST['cat_id'];
    $pname=$_POST['pname'];
    $pprice=$_POST['pprice'];
    $details1=$_POST['var'];
    $details=implode(",",$details1);
    $image='upload/'.basename($_FILES['img']['name']);
    move_uploaded_file($_FILES['img']['tmp_name'],$image);
    $stmt1=$admin->ret("SELECT * FROM `product` WHERE `p_name`='$pname'");
    $num=$stmt1->rowCount();
    if($num>0){
        echo "<script>alert('product Name Alredy
Exists');window.location='../add_product.php';</script>";
    }else{

    $stmt=$admin->cud("INSERT INTO `product`
(`cat_id`,`ps_id`,`maincat_id`,`p_name`,`p_price`,`p_details`,`p_img`)VALUES('$cat_id','$psid','$p
et_id','$pname','$pprice','$details','$image')","inserted');
    echo "<script>alert('product added
successfully');window.location='../add_product.php';</script>";
```

```
  }

  }
```

6.3.3 Retrievals

Admin view User

```
<?php

include 'config.php';
$admin=new Admin();
?>
<!DOCTYPE html>
<html lang="en">


<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Admin</title>
  <!-- base:css -->
  <link rel="stylesheet" href="vendors/mdi/css/materialdesignicons.min.css">
  <link rel="stylesheet" href="vendors/css/vendor.bundle.base.css">
  <!-- endinject -->
  <!-- inject:css -->
  <link rel="stylesheet" href="css/style.css">
  <!-- endinject -->
  <link rel="shortcut icon" href="images/favicon.png" />
</head>


<body>
  <div class="container-scroller d-flex">
    <!-- partial:partials/_sidebar.html -->
    <?php include 'sidebar.php'?>
    <!-- partial -->
    <div class="container-fluid page-body-wrapper">
```

```
<!-- partial:partials/_navbar.html -->
<?php include 'header.php'?>
<!-- partial -->
<div class="main-panel">
  <div class="content-wrapper">
    <div class="row">

      <div class="col-lg-12 grid-margin stretch-card">
        <div class="card">
          <div class="card-body">
            <h4 class="card-title">All Users</h4>

            <div class="table-responsive pt-3">
              <table class="table table-bordered">
                <thead>

                  <tr>
                    <th>SL.NO</th>
                    <th>NAME</th>
                    <th>Email</th>
                    <th>Phone</th>
                    <th>Address</th>
                  </tr>

                </thead>
                <tbody>
                <?php
                      $i=1;

                      $stmt=$admin->ret("SELECT * FROM `user` ");
                      while($row=$stmt->fetch(PDO::FETCH_ASSOC)){
                      ?>
                 <tr>
```

```
                    <td><?php echo $i++?></td>
                    <td><?php echo $row['u_name']?></td>
                    <td><?php echo $row['u_email']?></td>
                    <td><?php echo $row['u_phno']?></td>
                    <td><?php echo $row['u_address']?></td>


                </tr>
                <?php } ?>
            ?>
```

View Feedback

```
<?php
include 'config.php';
$admin=new Admin();

?>
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>view Feedback</title>
  <!-- base:css -->
  <link rel="stylesheet" href="vendors/mdi/css/materialdesignicons.min.css">
  <link rel="stylesheet" href="vendors/css/vendor.bundle.base.css">
  <!-- endinject -->
  <!-- inject:css -->
  <link rel="stylesheet" href="css/style.css">
  <!-- endinject -->
  <link rel="shortcut icon" href="images/ga-logo.png" />
</head>
```

```
<body>
  <div class="container-scroller d-flex">
    <!-- partial:partials/_sidebar.html -->
    <?php include 'sidebar.php'?>
    <!-- partial -->
    <div class="container-fluid page-body-wrapper">
      <!-- partial:partials/_navbar.html -->
      <?php include 'header.php'?>
      <!-- partial -->
      <div class="main-panel">
        <div class="content-wrapper">
          <div class="row">

            <div class="col-lg-12 grid-margin stretch-card">
              <div class="card">
                <div class="card-body">
                  <h4 class="card-title">feedbacks</h4>

                  <div class="table-responsive pt-3">
                    <table class="table table-bordered">
                      <thead>

                        <tr>
                          <th>SL.NO</th>
                          <th>NAME</th>
                          <th>Messgae</th>
                          <th>Date</th>
                        </tr>

                      </thead>
                      <tbody>
                      <?php
                            $i=1;
```

```
                $stmt=$admin->ret("SELECT * FROM `feedback` INNER JOIN `shop` ON
shop.ps_id=feedback.ps_id ");
                while($row=$stmt->fetch(PDO::FETCH_ASSOC)){
                ?>
            <tr>

                <td><?php echo $i++?></td>
                <td><?php echo $row['ps_name']?></td>
                <td><?php echo $row['message']?></td>
                <td><?php echo $row['date']?></td>


             </tr>
            <?php } ?>
```

6.3.4 Shop Update product

```
<?php
include '../config.php';
$admin=new Admin();
$psid=$_SESSION['psid'];
if(isset($_POST['product'])){
   $pid=$_POST['pid'];
   $p_name=$_POST['pname'];
   $p_price=$_POST['pprice'];
   $image='upload/'.basename($_FILES['img']['name']);
   move_uploaded_file($_FILES['img']['tmp_name'],$image);
   $p_details=$_POST['var'];
   $p_details=implode(",",$p_details);
   $stmt=$admin->cud("UPDATE `product` SET
`p_name`='$p_name',`p_price`='$p_price',`p_details`='$p_details',`p_img`='$image' WHERE
`p_id`='$pid'" , 'updated');
```

```
    echo "<script>alert('product updated
successfully');window.location='../view_product.php';</script>";


}
?>
```

6.4 Data Validation

```
    <?php
include '../config.php';
$psid=$_SESSION['psid'];
$admin=new Admin();
if(isset($_POST['pet'])){
    $pname=$_POST['maincat_name'];
    $stmt1=$admin->ret("SELECT * FROM `maincat` WHERE `maincat_name` ='$pname'");
    $num=$stmt1->rowCount();
    if($num>0){
        echo "<script>alert('Category Name Alredy
Exists');window.location='../add_product.php';</script>";
    }else{

    $stmt=$admin->cud("INSERT INTO `maincat` (`ps_id`,`maincat_name`
)VALUES('$psid','$pname')",'inserted');
    echo "<script>alert('Category Added Successfully');window.location='../category.php';</script>";
}
}
if(isset($_POST['category'])){
    $p_id=$_POST['p_id'];
    $catname=$_POST['cat_name'];
    $stmt1=$admin->ret("SELECT * FROM `category` WHERE `cat_name`='$catname'");
    $num=$stmt1->rowCount();
    if($num>0){
        echo "<script>alert('product Name Alredy
Exists');window.location='../add_product.php';</script>";
```

```
}else{

$stmt=$admin->cud("INSERT INTO
`category`(`ps_id`,`maincat_id`,`cat_name`)VALUES('$psid','$p_id','$catname')","inseted');
echo "<script>alert('category Added Successfully');window.location='../category.php';</script>";
}
}
?>
```

6.5 Search

```
// ---Not applicable---
```

6.6Passing Of Parameters

```
<?php
include '../config.php';
$admin=new Admin();
$stm=$admin->ret("SELECT * FROM `service_provider`");
while($row1=$stm->fetch(PDO::FETCH_ASSOC)):
$d_arr[$row1['sp_id']]=$row1['name'];
endwhile;

?>
```

6.7Internal Documentation

```
<?php
include '../config.php';
$admin=new Admin();
//Posted values are assigned to the local variables
```

```
$username=$_POST['username'];

$password=$_POST['password'];

$name=$_POST['fullname'];

$email=$_POST['email'];

$address=$_POST['address'];

$contact=$_POST['contact'];

$status='pending';

//Calling the cud function to insert user registration information to the user table

$stmt=$admin->cud("INSERT INTO `User`
(`username`,`password`,`name`,`email`,`address`,`contact`,`status`) VALUES
('$username','$password','$name','$email','$address','$contact','$status')","saved");

//Message Successful

echo                    "<script>alert('Registration
successful.... ');window.location='../login/login.php';</script>";

?>
```

*****

CHAPTER-7

USER INTRERFACES

Introduction

The “Galaxy Fashion” is a user-friendly GUI based website which is developed using PHP, CSS, HTML, as front-end tool and My SQL server and PHP as back end. The users of this website are Users, shop’s and Admin.

7.1 Login:

7.1.1 Purpose: Admin Login:

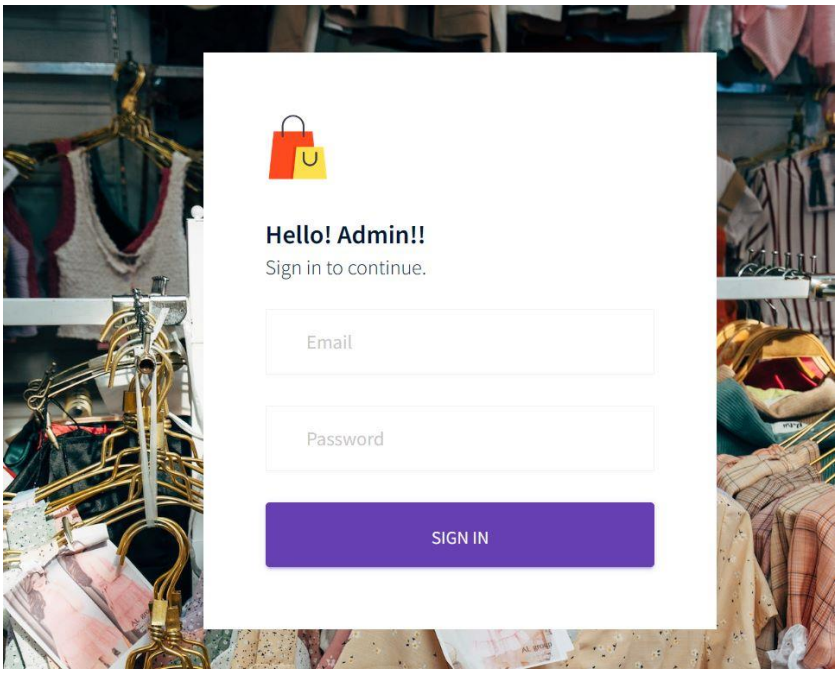

Figure 7.1: Admin login Form

7.1.1.1 Description

The Admin begins all his activities in the project by Entering valid email and password in the login form. Here two textboxes for entering email and password are placed. Button is placed in the name of sign in.

7.1.1.2 Navigation

The login icon is located in the admin login page. To view this screen, the admin must click on that.

7.1.1.3 Elements

- **Email:**
    - **Type:** textbox, **Label:** Email
    - **Content:** To enter Email.
- **Password:**
    - **Type:** password**, Label:** password
    - **Content:** To enter password.
- **Login:**
    - **Type:** button, **Label:** Login
    - **Content:** It is used to submit the form and check if username and password are valid.

7.1.2 Purpose: Shop Login

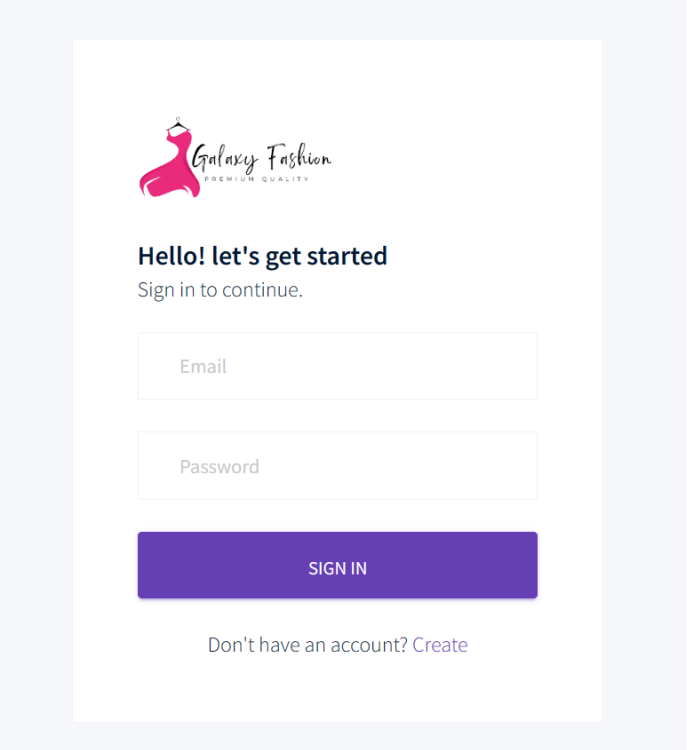

Figure 7..2: Shop login Form

7.1.2.1 Description

This interface permits supplier to log in to the system. The supplier uses this GUI to sign in. It collects password and username. Collected credentials will be checked by gathering data from the

supplier database and continuing to the shop home page if successful.

7.1.2.2 Navigation

The Login code is located in the bottom Home page. To view this screen, the supplier must click on that.

7.1.2.3 Elements

- **Email:**
    - **Type:** textbox, **Label:** Email
    - **Content:** To enter Email.
- **Password:**
    - **Type:** password**, Label:** password
    - **Content:** To enter password.
- **Login:**
    - **Type:** button, **Label:** Login
    - **Content:** It is used to submit the form and check if username and password are valid.

7.1.3 Purpose: User login

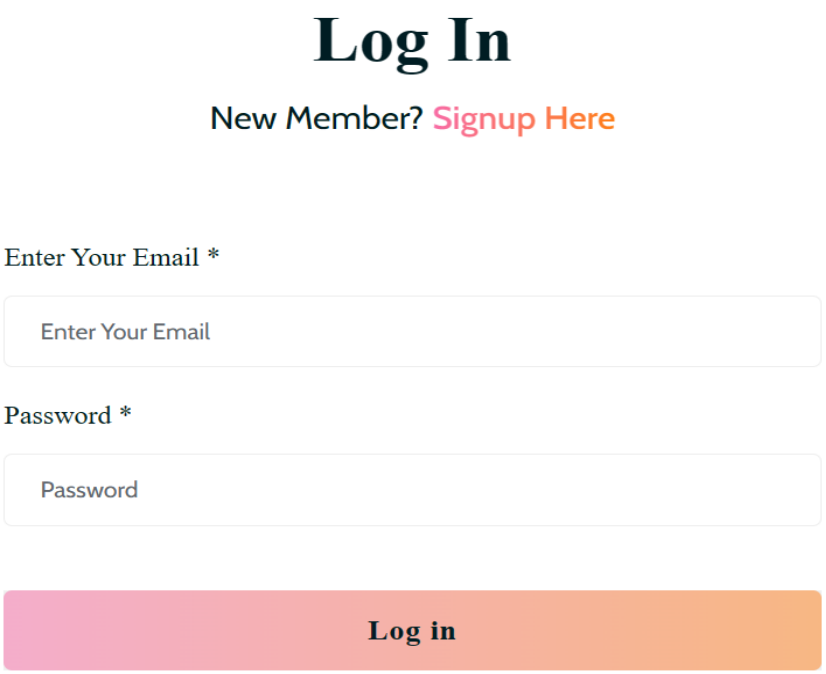

Figure 7.3: User login Form

7.1.2.1 Description

This interface permits user to log in to the system. The user uses this GUI to sign in. It collects password and email. Collected credentials will be checked by gathering data from the user database and continuing to the home page if successful.

7.1.2.2 Navigation

The Login code is located in the bottom Home page. To view this screen, the user mustclick on that.

7.1.2.3 Elements

- **Email:**
    - **Type:** textbox, **Label:** Email
    - **Content:** To enter Email.
- **Password:**

- **Type:** password**, Label:** password
- **Content:** To enter password.

- **Login:**
  - **Type:** button, **Label:** Login
  - **Content:** It is used to submit the form and check if username and password are valid.

7.2 Main Home Screen

7.2.1 Purpose: User Home Screen

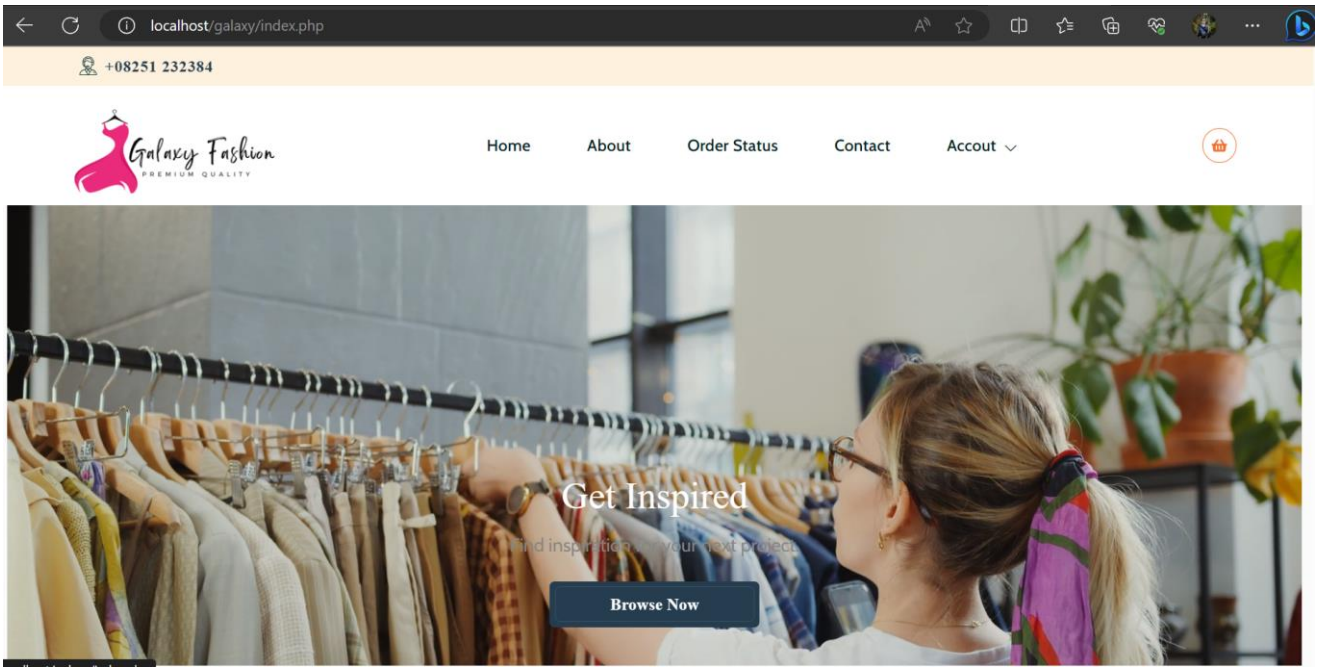

Figure 7.4: User Home Screen

Description

After entering valid Username and Password, the user is redirected to User Home page. If the User is not registered, he will never get home page. User Home page contains menus such as View products; View shops orders, carts and Contact.

Navigation

This page is home page of user.

Elements

Home page links different pages. For example, if you want order the product or view cart.

7.2.2 Purpose: Admin Home page

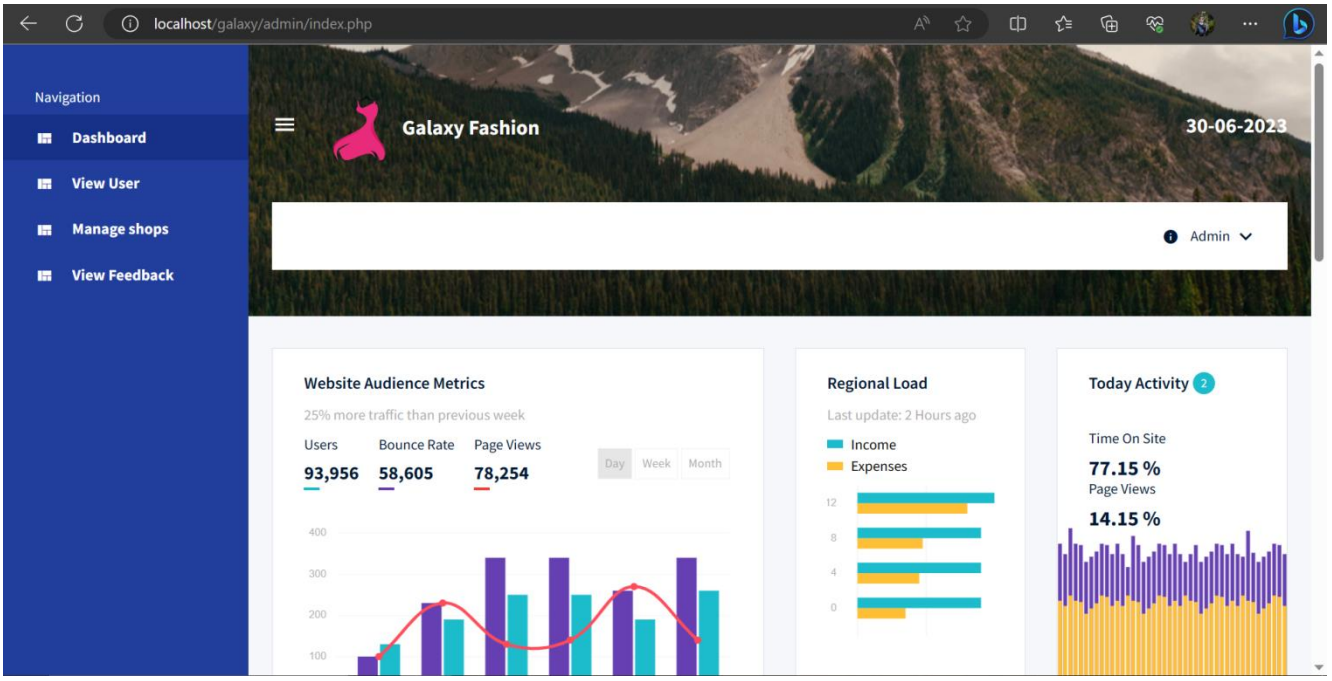

Figure 7.5: Admin Home page

Description

After receiving proper username and password admin is redirected to this page. He can view all the activities of project. It contains menus such as Home, View Users, Feedback, Manage shops.

Navigation

This page is main page of admin.

Elements

Admin dash board links to different pages. For example, if you want to view Feedback, manage shop and view user.

7.2.2 Purpose: Shop Dash Board

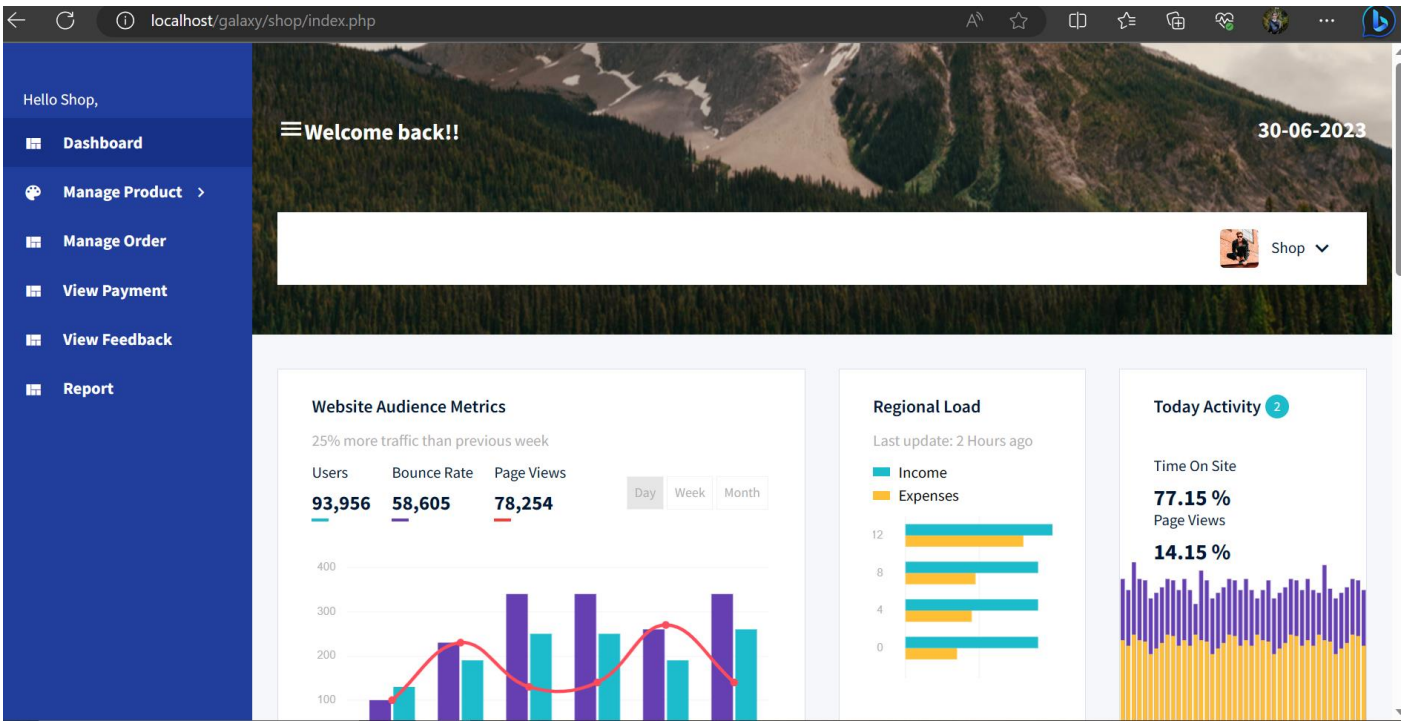

Figure 7.6: Shop home page

Description

After receiving proper username and password admin is redirected to this page. He can view all the activities of project. It contains menus such as Home, View Users, Feedback, Manage products, manage order and payment.

Navigation

This page is main page of shop.

Elements

Shop dash board links to different pages. For example, if you want to view Feedback, manage product and view user.

7.3 Menu

7.3.1 Purpose: Admin Menu

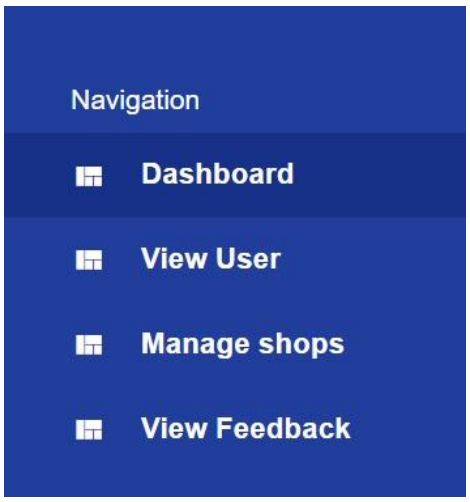

Figure 7.7: Admin menu

Description

Admin is authorized to upload and view the shops. Also, he is enabling view user and feedback.

7.3.2 Purpose: Shop Menu

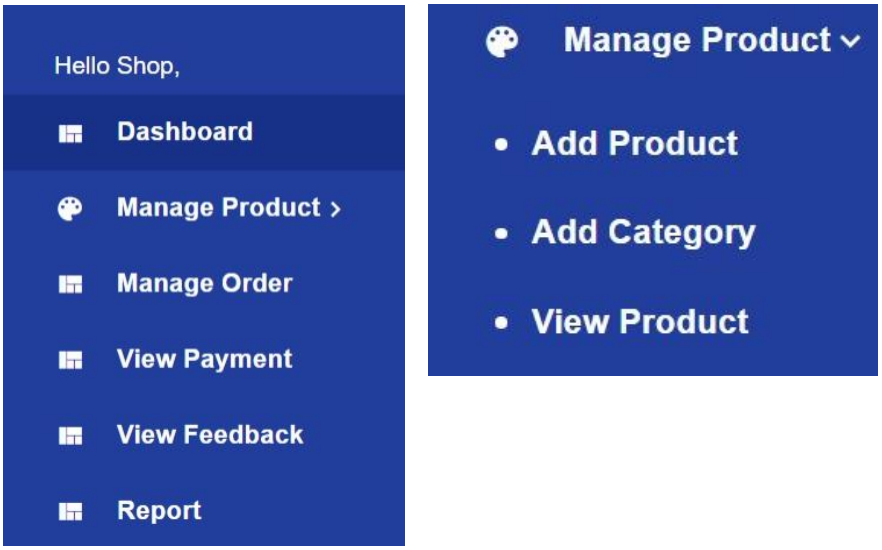

Figure 7.8: shop menu

Description

Here there is a Manage product menu to add product, add category and view product. After shop owner he add his products, delete his products and Manage Orders. He is also allowed to update his products.

Manage product: Manage product is a dropdown menu. It enables the shop owner to Add/View products also it enables to manage orders.

7.3.3 Purpose: User Menu

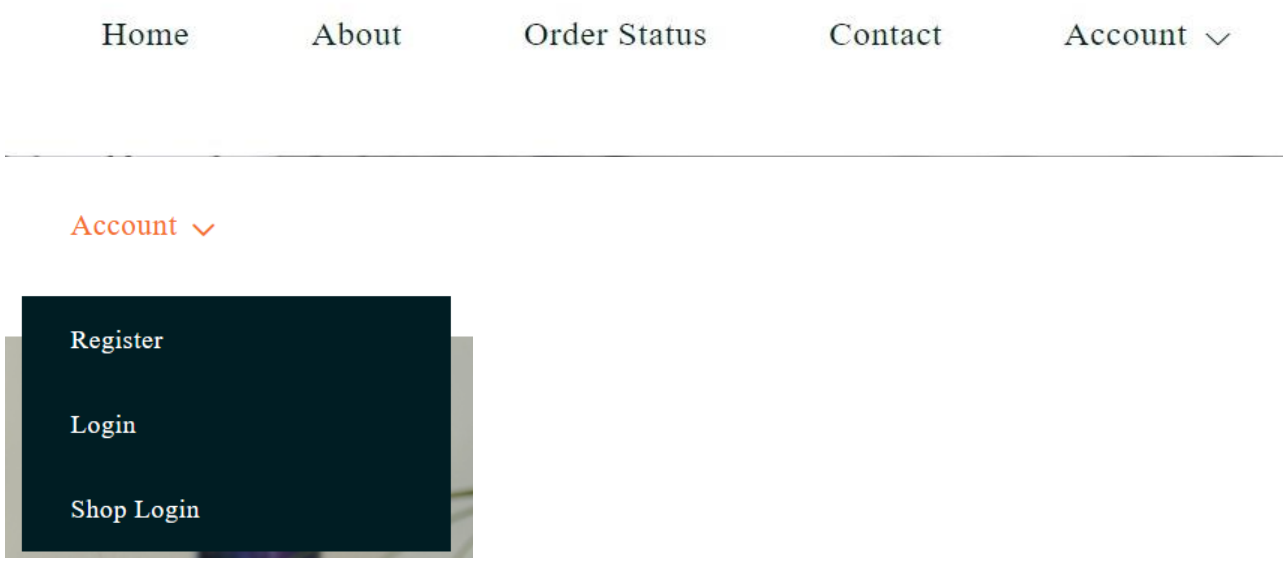

Figure 7.9: user menu

Here there is a home menu display home page, about menu tell about the website , order status menu display the order status shipping details and the contact menu tell about website contact details or the user care services.

Account: Account is a dropdown menu. It enables the user toregister or login the website and also enables to shop owner login

7.4 Data store/retrieval/update

7.4.1 Data Store –Add product

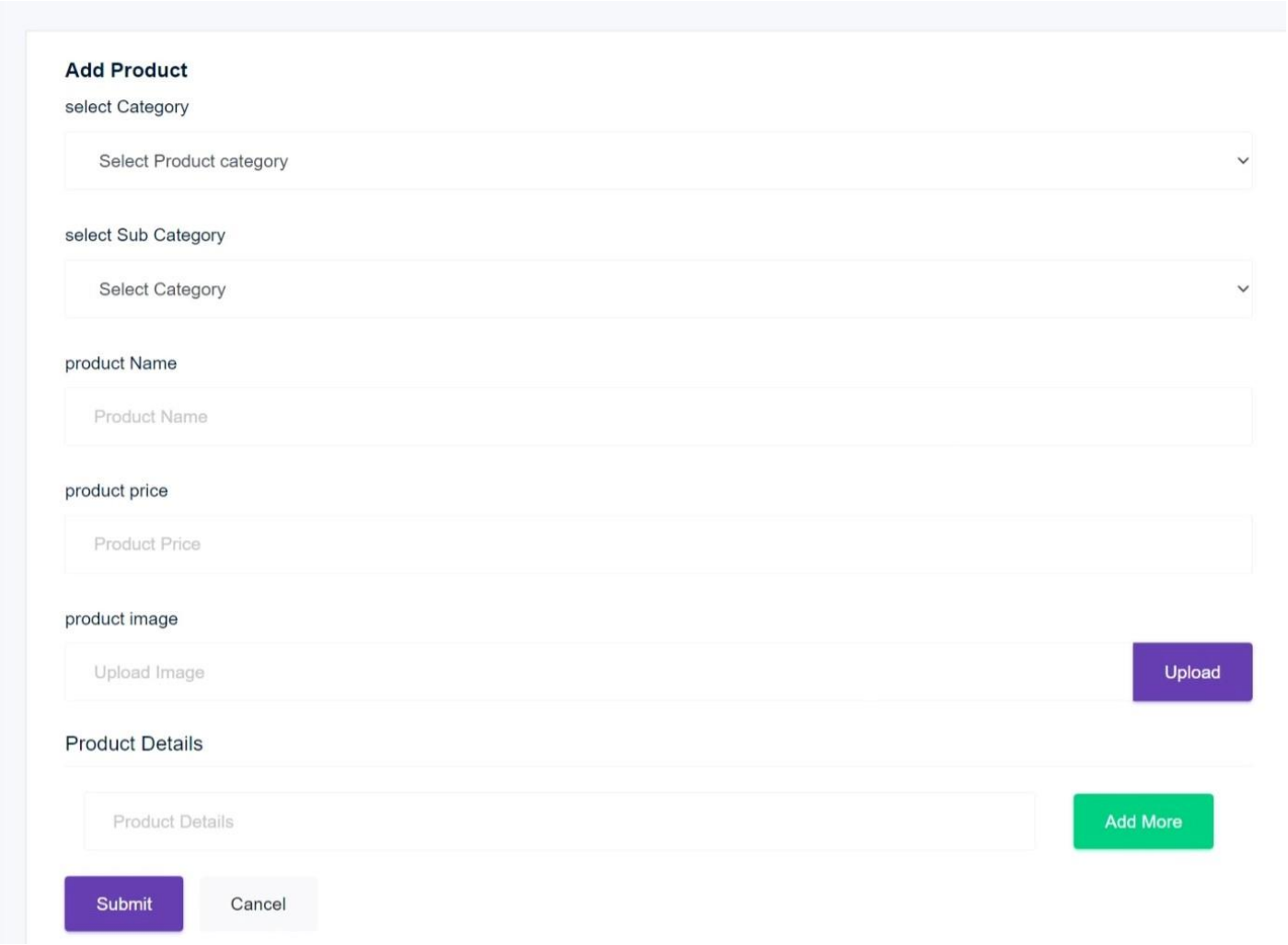

Figure 7.10: Add product

Description

In order to display the details of the product to the shop the shop owner has to add his product First. If the product is successfully added then his product are visible to users.

In the above Add product form, text box is used to fetch all the information from SP. Titles of the form is displayed with the help of labels. Save is a submit button which takes the data from shop owner and send it to database.

Navigation

This is the screen that will appear when the shop owner clicks on add submit icon.

Elements

- **Product name:**

  Type: textbox, **Label:** Machine Name

  Content: To enter machine name.

- **Product Category**

  Type: textbox, **Label:** Machine Name

  Content: To enter machine name.

- **Product sub Category**

  Type: textbox, **Label:** product Name

  Content: To enter product name.

- **Image:**

  Type: textbox**, Label:** image

  Content: To enter image of the product.

- **Price:**

  Type: textbox, **Label:** price

  Content: It is used to enter the price.

- **Product details**

  Type: textbox, **Label:** product details

  Content: To enter product details.

- **Submit**

  Type: button, **Label:** submit

  Content: It is used to add the product to the website

7.4.2 Data Retrieval

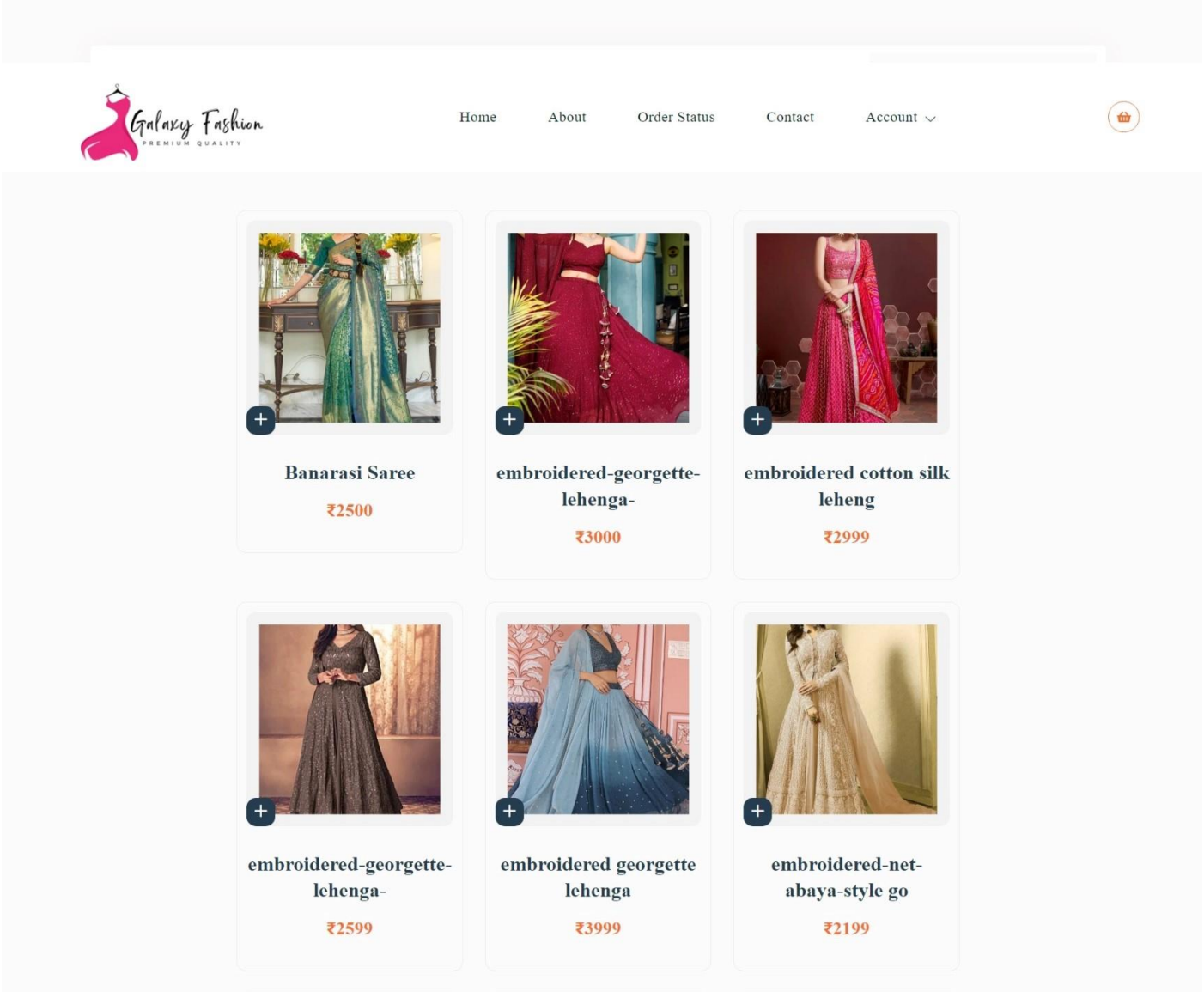

Figure 7.11: all products by shop

Description

After login if the user is willing to order the product, he has to click upon get product in + button. When he clicks + button, it allows view details of the product.

If the users wish to buy products, then he can click upon cart button and he gets product in the cart. User click to proceed to check out button then order the product.

Navigation

This page is displaying all product details.

Elements

Product: Type: button, Text: product

Content: To go back to home page.

7.5 Validation

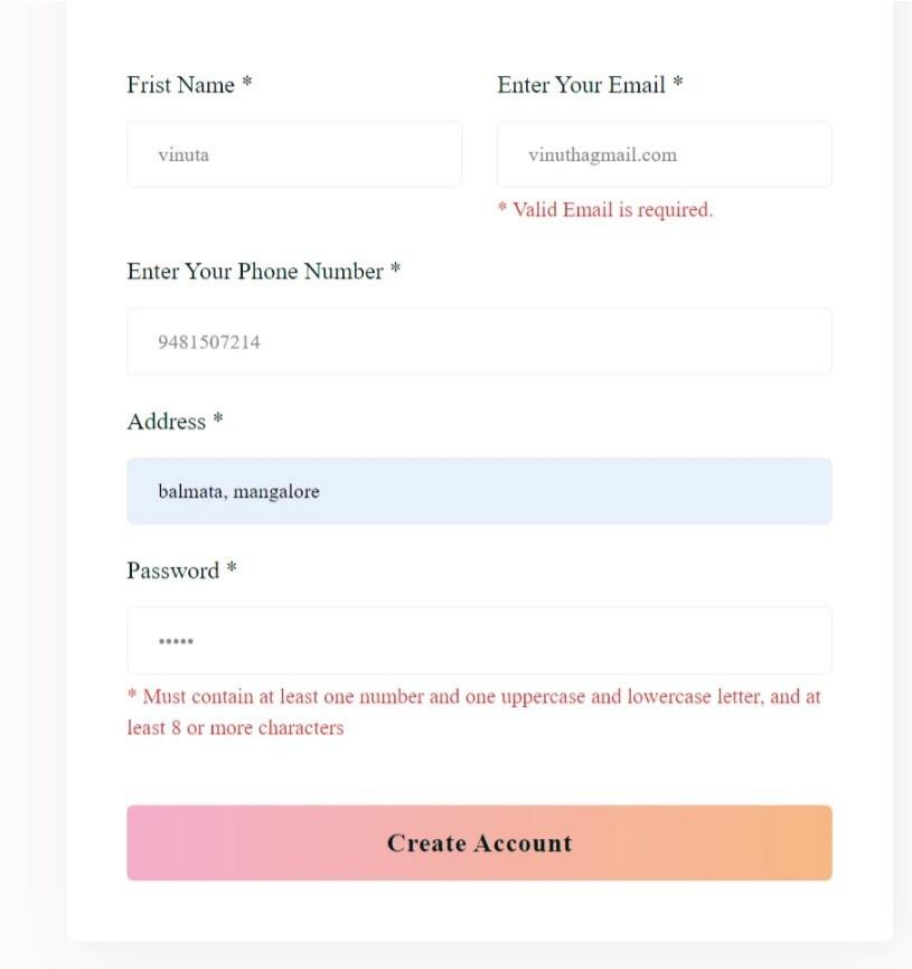

Figure 7.12: validation in creating account

Description

When user tries to register to the site using the details, if the required input is not given then it will show the invalid register credentials .In the above interface it is showing the invalid message because it is given the invalid email and invalided password.

Navigation

The sign in icon is located in the user login page. To view this screen, the user must click on the register link button.

Elements

- **Email:**

  Type: textbox, **Label:** username

  Content: To enter username.

- **Password:**

  Type: password**, Label:** password

  Content: To enter password.

- **User name:**

  Type: textbox, **Label:** Full name

  Content: To enter Full name of the user.

- **Address:**

  Type: textbox, **Label:** Address

  Content: To enter Address of the Shop.

- **Contact:**

  Type: number, **Label:** Contact

  Content: To enter Shop mobile number.

- **Sign in:**

  Type: button, **Label:** Register

  Content: It is used to store the details of user in the database

7.5 View

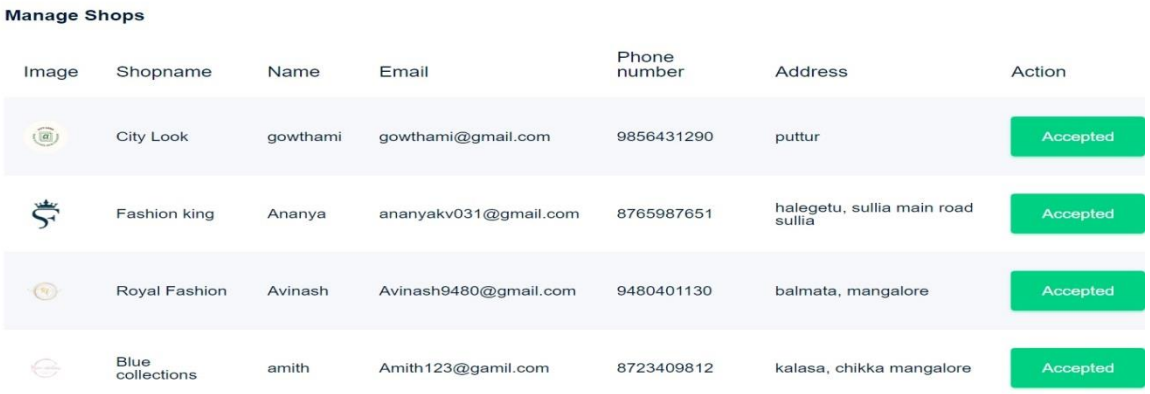

Figure 7.13: Admin view the shop

Description

Admin Can View all the shops and their details. He can mainly view SPID, Name,Image, Email, Mobile, Address, and Contact.

Navigation

The shop icon is located in the home page. To view this screen, the user must click on that.

Element

- **Accept:**

  Type: button, **Label:** accept

  Content: Admin can approve shop.

- **Reject**

  Type: button**, Label:** Reject

  Content: Admin can reject shop.

7.6 Error message

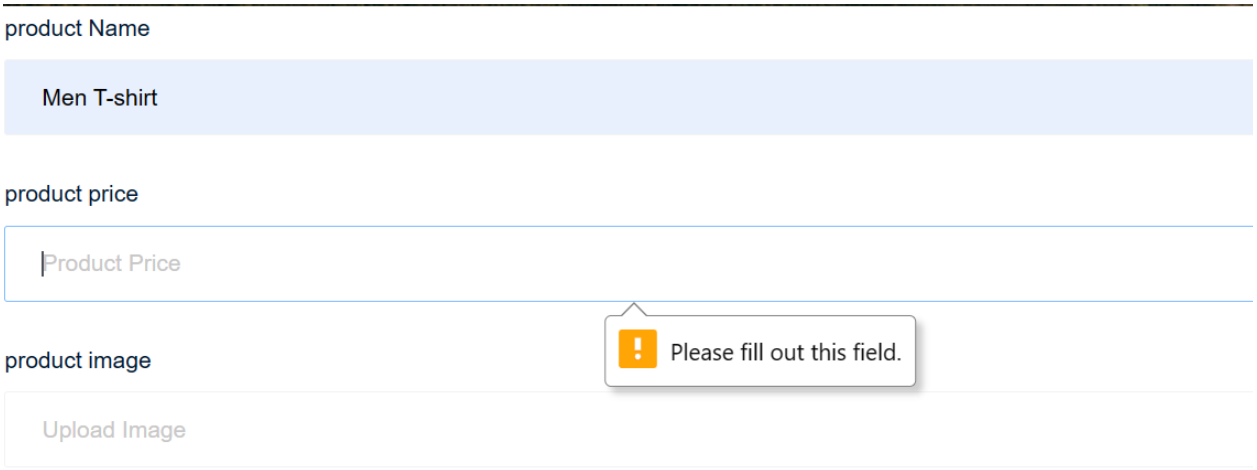

Figure 7.14: error massages in add product

Description

Error message Prompts when the things happen that should not happen. In the above page when the shop owner tried to add product without mentioning the price. So this message is triggered.

Navigation

This page is displayed as an error message.

Elements

OK Type: button, **Text**: OK

Content: To go back to add product page.

7.7 Alerts

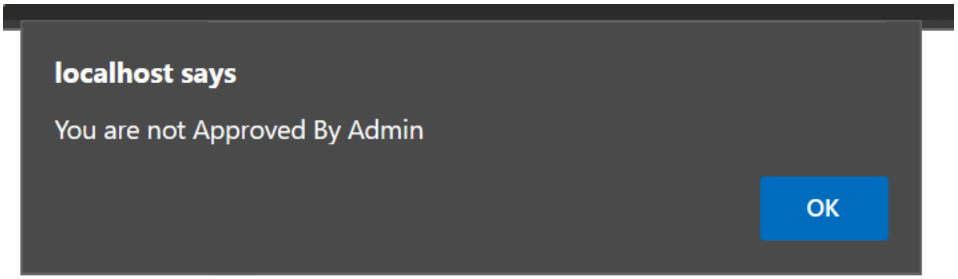

Figure 7.9: Alerts in shop login

7.9.1 Description

When the shop not approved by admin clicks login button, if he enters user name and password this message is prompted. By clicking ok shop owner can jump to login page.

7.9.2 Navigation

This page is displayed as alert message.

7.9.3 Element

- **OK Type**: button, **Text:** OK
- **Content:** To go back to login page

****

CHAPTER-8

TESTING

8.1 Introduction

Software Testing is an investigation conducted to provide stack holders with information about the quality of the product or productunder test. Testing has been defined as the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Software testing is the process used to assess the quality of computer software. It involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should. It is obtained to 'detection'.

There are two types of software testing

- **White Box testing**

White box testing strategy deals with the internal logic and structure of the code. It is also called as glass, structural, open and clear box testing. The test that are written based on the white box testing strategy incorporate coverage of the code written, branches, statements and internal logic of the code etc.

- **Black Box testing**

Black box takes the internal perspective of the test object to derived test case. These tests can be functional and non-functional though usually functional. The test Designer selects valid and invalid inputs and determines the correct input. There is no knowledge of the test object's internal structure. This method of test design is applicable to all levels of software testing: unit, internal, functional and system and acceptance.

Software testing has three main purposes:

- The verification process confirms that the software meets its technical specifications. A "specification" is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
- The validation process confirms that the software meets the business requirements.
- A defect is a variance between the expected and actual result. The defect's ultimate source may

be traced to a fault introduced in the specification, design, or development phases. Not all the defects will necessarily result in failures.

8.2 Test Reports

Table 8.1: Test Reports

Sl. No	Unit Testing	Integration Testing
1	It does not occur after and before of Anything.	It occurs after Unit Testing and before System Testing.
2	It is not abbreviated by any name	It is abbreviated as "I&T" that is why sometimes also called Integration and Testing.
3	It is not further divided into any.	It is further divided into Top-down Integration, Bottom-Up Integration and so on.
4	It may not catch integration errors, or other system-wide issues because unit testing only tests the functionality of the Units themselves.	Integration testing uncovers an error that arises when modules -are integrated to build the overall system.
5	The goal of unit testing is to isolate each part of the program and show that the individual Parts are correct.	The goal of Integration Testing is to combined modules in the application and tested as a group to see that they are working fine.
6	It does not follow Anything.	It follows unit testing and precedes System testing.
7	It obviously starts from the module Specification.	It obviously starts from the interface specification.

8	It is used to check whether the system meets the defined Specifications of not.	It is used to check whether it meets the defined User Requirements or not.
9	It requires complex Scaffolding means frame.	It requires some scaffolding means frame.
10	It definitely pays attention to the behavior of single Modules.	It definitely pays attention to the integration among modules.
11	It is only the kind of White Box Testing.	It is both the kind of Black Box and White Box Testing.

Sl. No	**System Testing**	**Acceptance Testing**
1	System Testing does not include any thing means it is not known by any other name.	Acceptance Testing Include Alpha Testing means it is also sometimes known as alpha testing.
2	User is not involved in System Testing.	User is completely involved in Acceptance Testing.
3	It is performed before Acceptance Testing.	It is performed after Acceptance Testing.
4	It is not the final stage of validation.	It is the final stage of Validation.
5	System testing of software or hardware testing conducted on a whole, integrated system to estimate the systems compliance with its specified set of Requirements?	Acceptance Testing of software or hardware is testing conducted to evaluate the system compliance with its specified set of user requirements.

8.2.1 Unit Testing

In computer programming, unit testing is a method by which individual units of source Code, sets of one or more computer programs modules together with associated control data, usage producers, are tested to determine if they are fit to use. Intuitively, one can view a unit as the smallest testable part of an application, In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object oriented programming a unit is often an entire interface, such as class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

Login Form

Table 8.2: Unit Testing – Login Form

Test Scenario ID	Login-1	**Test Case ID**	Login-1A
Test Case Description	Login– successful or not	**Test Priority**	High
Pre-Requisite	A valid user account	**Post - Requisite**	NA

Test Execution Steps:

Sl. No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	If user clicks on login button without entering username And password	Inputs are not given	Please enter username and password	Please enter username and password	Microsoft Edge	Pass	Invalid Login attempt

2	Enter correct username & password and hit login button	Username: admin Password: ******	Login Successful	Login Successful	Chrome	Pass	Login Successful
3	If username is blank but password is entered	Password: *******	Please enter username	Please enter username	Chrome	Pass	invalid login attempt
4	If password is blank but username is entered	Username: admin	Please enter password	Please enter password	Chrome	Pass	invalid login attempt
5	If the username or password is incorrect	Username: admin Password: ******	Invalid username and password	Invalid username and password	Chrome	Pass	invalid login attempt

Registration

Table 8.3: Unit Testing – Registration

Test Scenario ID	Registration-1	**Test Case ID**	Registrtion-1A
Test Case Description	Registration– successful or not	**Test Priority**	High
Pre-Requisite	A valid user account	**Post-Requisite**	NA

Test Execution Steps:

Sl.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	When user click on the register button	Inputs are not given	Displays registration form	Displays registration form	Chrome	Pass	Invalid Registration
2	User Enters Username Password Full Name Email, Address, Phone Number, And clicks on Register Button	Username: Sunil Password ****** full Name: Sunil Kumar Email: Sunil@gmail.com address: Mangalore Phone Number: 7996546345	Successfully registered	Successfully registered	Chrome	Pass	Registration Successful

3	If any fields are blank	User: Sunil Password: *******	All fields are mandatory	All fields are mandatory	Chrome	Pass	invalid registratio nattempt
4	If usernam ealready exists	Usernam e:Sunil	User already exists	User already exists	Chrome	Pass	invalid registratio nattempt
5	If email alread yexists	User Sunil Sunil@gmail.co mPassword ****** Usernam e:Sunil	Email already exists!	Email already exists!	Chrome	Pass	invalid registratio nattempt

User Home Page

Table 8.4: Unit Testing –user home page

Test ScenarioID	User Home Page	**Test Case ID**	User Home Page-1A
Test Case Description	User home page for User	**Test Priority**	High
Pre-Requisite	A valid user account	**Post-Requisite**	NA

Test Execution Steps:

Sl.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	When Useradd profile and Userenters Username, Full name, Email, Password, address,	Username: bhuvi Password ****** Full Name:	Profile added successfull y	Profile added successfully	Chrome	Pass	Profileadded

	contact number	Bhuvanesh Kumar Email: bhuvi@gmail.com address: Thokottu Phone Number: 9995434567					
2	If User clicks on product	Click button	New product page will displayed	New product page will displayed	Chrome	Pass	Successful
3	If Userclick proceed to checkout button oncart page	Click button	Ordered successfully	Ordered successfully	Chrome	Pass	Successful
4	If user clicks on cart	Click button	New Cart page will displayed	New Cart page will displayed	Chrome	Pass	Successful
5	If User clicks on logout	Click button	User will be redirected to the homepage of the Galaxy	User will be redirected to the homepage of the Galaxy	Chrome	Pass	Successful

			Fashion	Fashion			

Shop dash board

Table 8.5: Unit Testing – Shop Home Page

Test ScenarioID	Shop Home Page	**Test Case ID**	ShopPage-1A
Test Case Description	Shop Home page for Shop	**Test Priority**	High
Pre-Requisite	A valid user account	**Post-Requisite**	NA

Test Execution Steps:

Sl.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	When Shop add profile and Shop enters Username, Password, Full name Email, Address, Contact, Upload Photo and click on submit Button.	Username:anoop Password ****** Full Name: Anoop Chandra Email: anoop23@gmail.com address: Mangalore Phone Number: 9895654234	Profile added successfully	Profile added successfully	Chrome	Pass	Profile added

2	When ShopAdd ServicesAnd Shopenters Name, Add work, Price, andclick on add service	Anoop Grass cutting 850/- Spraying Pesticides500/-	Shop added successfully	Shop added successfully	Chrome	Pass	Shop added
3	When Shop Click onview Orders	Click button	Order page will displayed	Order page will displayed	Chrome	Pass	View Successful
4	If Shop clicks on logout	Click button	Shop will be redirect edto the homepa geof the Galaxy Fashion	Shop will be redirected to the homepage of the Galaxy Fashion	Chrome	Pass	Logout Successful

8.2.2 Integration testing

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

User view products

Table 8.6: Integration Testing – View products

Test ScenarioID	User view Product	**Test Case ID**	View Product-1A
Test Case Description	User view Product by login	**Test Priority**	High
Pre-Requisite	A valid user account	**Post-Requisite**	NA

Test Execution Steps:

Sl.No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1	If User view products	Click Button	View Products page will be displayed	View Product page will be displayed	Chrome	Pass	successfully
2	If User click book button on the Product page	Click button	Booked successfully	Booked successfully	Chrome	Pass	Successful

8.2.3 System Testing

System testing refers to the testing of complete system as a whole that is carried out by testers sometimes by the developers to check whether it meets the system specifications or not. System testing involves both functional and non-functional testing. In the system the behaviour of whole system product is tested as defined by the scope of the development project.

It may include tests based on risks or requirement specification, business process, use cases or other high level description of the system behaviour, interaction with the operating system, and system resources.

Table 8.7: System Testing

SL.NO	Test Condition	Test Result
1	System Loading	Successful
2	System Run Procedure	Successful
3	File I/O Operation	Successful
4	Database Communication	Successful
5	Server/Client Interaction	Successful
6	Memory Usage	Normal
7	System Processor Usage	Normal
8	Authentication/Authorization	Successful

****

CHAPTER-9
CONCLUSION

9.1 Conclusion

The “Galaxy Fashion” (Online Ready wear shop) is an e-commerce site that which supplies various kind of clothing material. User will order the clothes that they need from this Galaxy Fashion and Administrator who is the Galaxy Fashion owner can manage the user orders. Admin can get order required materials from the suppliers. Suppliers can view and manage orders from the Admin can assign the work of order. With database connectivity and capability, this application manages the resources efficiently and effectively through the control of the database, this application will able to reduce paperwork involve for the product browsing and purchasing. The main objective of this project is to develop an attractive, interactive and user-friendly online selling website.

This web-based software can be accessed from anywherewith internet and has a good scope. It is flexible enough to incorporate new features and modules as per the user requirements. The module has fulfilled all the objectives identified. The system is developed in such a way that the user with common knowledge of computers can handle it easily. The module has a user-friendly interface.The reports requested by the client have been generated and all documentation requiredfor operation and maintenance of the module has been provided. The future enhancement to the system can be made as technology improves or changes

9.2 Limitations

- Because of lack of knowledge users may not able to handle the website properly.
- The information on website might be unreliable if not updated on a regular basis.
- Android or IOS apps yet to be develop.
- We only delivery product with in 50km from the shop.
- The information on website might be unreliable if not updated on a regular basis.

9.3 Scope and Enhancement

. This web application can be extended to support different types of the payments modes that help the user. An android application can be developed to help the users to access the system easily and effectively.

- The system is designed in such a way that any new features can be added easily.
- Highly Succeeded and Achieved users in the specific field is allowed to given

correct product information users through online video.

- The system is designed in such a way that any new features can be added easily.
- It will reduce work and time to find products compare to traditional shopping
- Android application can be developed

9.4 Abbreviation and Acronyms

- DB-Database
- DBA-Database Administrator
- DBMS-Database Management System
- DCL-Data Control Language
- DDL-Data Definition Language
- DML- Data Manipulation Language
- CRUD-Create, Read, Update and Delete
- EAV-Entity-Attribute-Value
- ERD-Entity Relationship Diagram
- FK-Foreign Key
- NF-Normal From
- OS-Operating System
- PK-Primary Key
- RDBMS-Relational Database Management System
- 1NF – First Normal Form
- 2NF – Second Normal Form
- 3NF – Third Normal Form

9.4 References

9.4.1 Books Reference

- PANKAJ JALOTTED" S SOFTWARE ENGINEERING: A PRECISE APPROACH By Pankaj Jalote (Author).
- Learning PHP, MySQL, JavaScript, CSS & HTML 5: A Step-by-Step Guide to Creating Dynamic websites by robin Nixon (Author).

9.4.2 Web Reference

- Pankaj Jalote, (2004). *An Integrated Approach to Software Engineering*, (3rd ed.). Narosa Publishing House.
- D. K. K. K. Prasad. *Software Testing Tools*. Dreamtech Press.
- Ivan Bayross. *HTML5 and CSS3 made simple*. BPB Publications.
- Ivan Bayross.*SQL/PL/SQL- the Programming language of Oracle* (1st ed.). BPBPublications.
- Get started with Bootstrap. https://getbootstrap.com/docs/5.3/getting-started/introduction
- PHP Tutorial. https://www.geeksforgeeks.org/php-tutorial/
- CSS Tutorial. https://www.w3schools.com/css/default.asp
- JavaScript Tutorial. https://www.w3schools.com/js/default.asp
- HTML Tutorial. https://www.w3schools.com/html/default.asp
- [Vishweb Design]. https://youtu.be/iNnLHlpdrao
- [FundaOfWebIT]. https://www.youtube.com/@FundaofWebIT
- [BoostMyTool]. https://www.youtube.com/@BoostMyTool
- Draw.io. https://www.draw.io/app.digrams.net

****