



Integration Of Stores

A Project of Capstone Project - 2

Submitted by

Tanisha Kumari

(1613107064 /16SCSE107035)

In partial fulfillment for the award of the degree of

Bachelor of technology

IN

**Computer Science & engineering with specialization in cloud
computing&virtualization**

School Of Computer Science And Engineering

Under the Supervision of

Dr.Pallavi Murghai Goel

Assoc.professor

2019-2020



**SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING
BONAFIDE CERTIFICATE**

Certified that this project report “..integration of stores..” is the
bonafide work oftanishakumar(1613107064).....” who carried out the
project work under my supervision.

<<Signature of the Head of the Department>>

SIGNATURE

<<Name>>

HEAD OF THE DEPARTMENT

<Academic Designation>>

<<Department Details>>

<<Department Details>>

<<Signatur of th SuperVISIO
e e R

SIGNATURE

<<Name>

SUPERVISior

ABSTRACT

The project is web-based system for an existing shop. The project objective is to deliver the online shopping application with a java platform.

This project is an attempt to provide the online shopping to customers from a real shop. It helps buying the products in the shops anywhere through internet by using a java-based application supported devices. Thus, the customer will get the service of online shopping and home deliver from the favorite shop. This system integrates the small shops of localities and put them into online. the system provides an online portal where their customers can enjoy easy shopping from anywhere, with there favorite shop from locality .to use the service the shop has to registered. The registration step involves the intake of data which include product detail, prices, discount etc. and put them into online database.to update of shop moderator are used.

On the other-hand to use the system the guests have to registered first for using the services. Once the guest become a user, they can easily order the product from there favorite shop in their localities which are registered on the system.

LIST OF FIGURES

ADMIN MODULE

Manage moderators

Manage products

Manage users

Manage orders

Moderator modules

User modules

E-R DIAGRAMS & DFDS

Login

User details

Product details

Product orders

Complete diagrams

Login DFD

Registration DFD

Moderator DFD

LIST OF TABLES

Login table

User detail table

Product detail table

Product order table

	TABLE OF CONTENT	Page No
ABSTRACT		iii
List Of Figures		iv
List Of Tables		iv,v
List Of Symbols		v
INTRODUCTION		vi
1.1Project objective		
1.2Project over-view		
1.3Project scope		
1.4Study of system		
2. MODULES		
2.1Admin		
2.2Moderator		
2.3User		
3. SYSTEM ANALYSIS		
3.1Existing system		
3.2Proposed system		
3.3SYSTEM REQUIREMENT		
4 .General description		
5.System objective		
6.System requirement		
6.1 Non-functional		

6.2Functional

7 SYSTEM DESIGN

7.1Input and output design

7.2 Input design

7.3Output design

8.DATABASES

8.1SYSTEM TOOLS

8.2FRONT-END TOOLS

8.3BACK-END TOOL

9.TABLES

9.1E-R diagram

9.2Data flow diagram

INTRODUCTION

This project is a web-based shopping system for an existing shop. The objective of this project is to deliver the online shopping system into a Java platform. So that users can easily order products from their favorite shop.

Online shopping is the process whereby consumers can directly buy goods or services from a seller in real-time, without an intermediary service, over the internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through the internet by using Java-enabled devices. Thus, the customer will get the services of shopping and home delivery from his favorite shop.

PROJECT OBJECTIVE

The objective of this project is to make an application in a Java platform to purchase items from an existing shop. In order to build such an application, complete web support needs to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of a Java application with a web view.

PROJECT OVER-VIEW

The central concept of this is to allow customers to shop virtually using the internet and allows the customer to buy the items and articles of their desire from their favorite store. The information pertaining to the products are stored on an RDBMS at the server side.

The server processes the customer and the items are shipped to the address submitted by item. The application was designed into two modules first. One for the customer who wishes to buy the product second for the shopkeeper who maintain and update the information pertaining to the articles and those of the customers. The end user of this products is a departmental store where the application is hosted on the web and the administrator maintains the database. The application which deployed at the customer database, the details of the item are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various screen designed for level of users. Once the authorized personal feed the relevant data into the system, several reports could be generated could be generated as per the security.

STUDY OF THE SYSTEM

MODULES:

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules are:

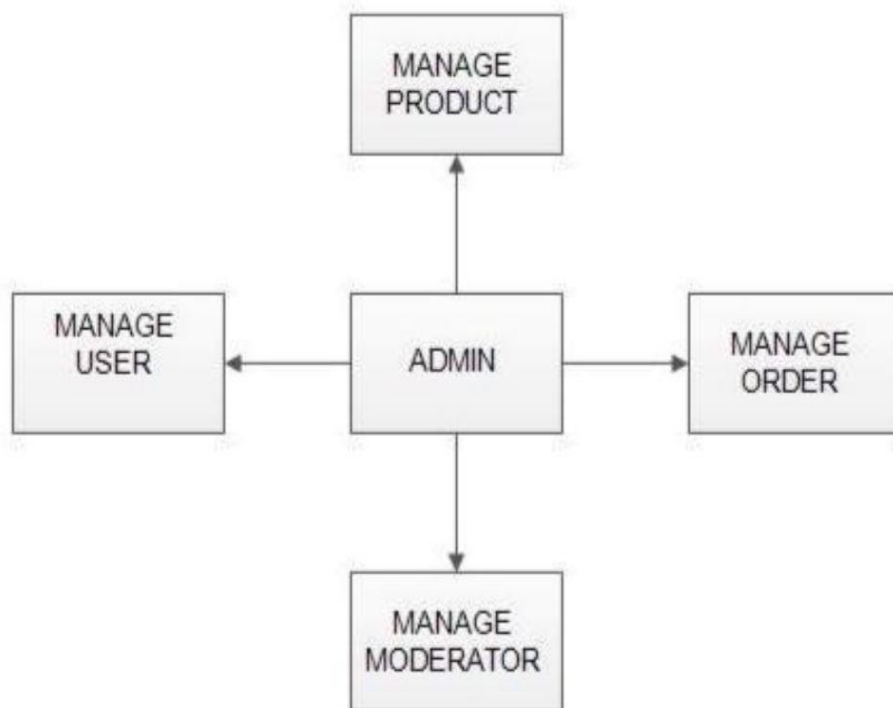
- ADMINISTRATOR
- MODERATOR
- USER

The administrator is the super user of the application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about all the user and all products.

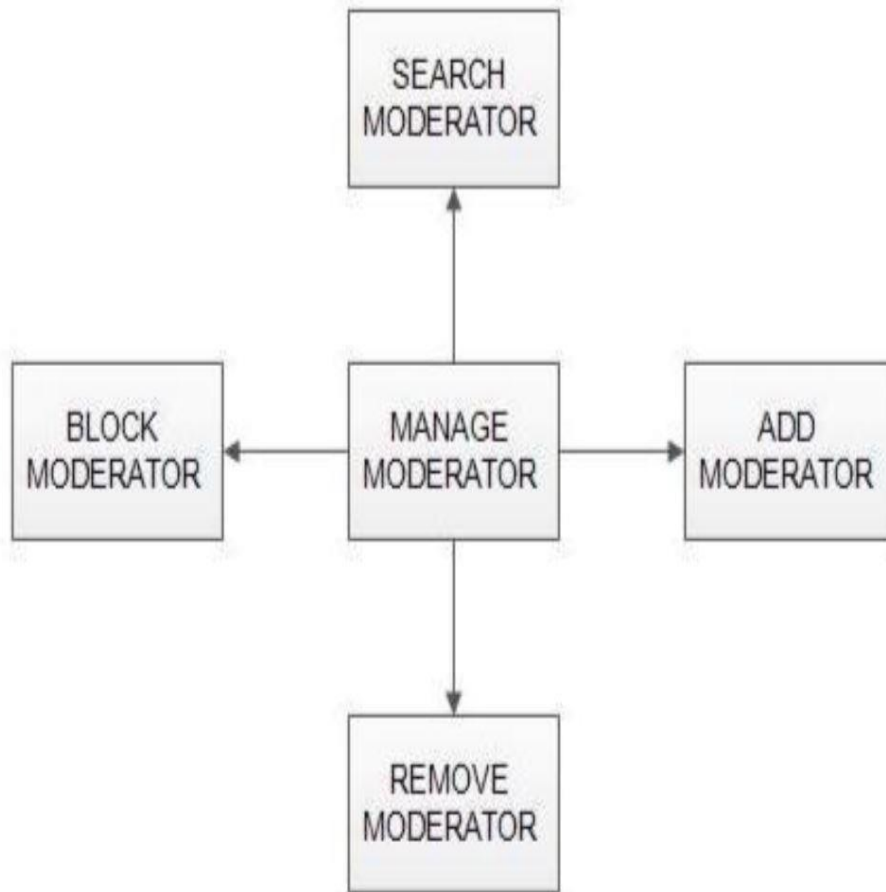
This module is divided into different sub-modules.

1. Manage moderator
2. Manage products
3. Manage user
4. Manage order

ADMIN MODULES



MANAGER MODERATOR



Add moderator:

Only admin is having the privilege to add a moderator. A moderator can be considered as a staff who manages or owner of a group of the product.

Block moderator

Only admin is having the privilege to add a moderator. A moderator can be considered as a staff who manages the orders or owners the orders or owner of a group of products.

Block moderator

Admin can restrict a moderator from managing the order by blocking them.

Admin can unblock a blocked user if needed.

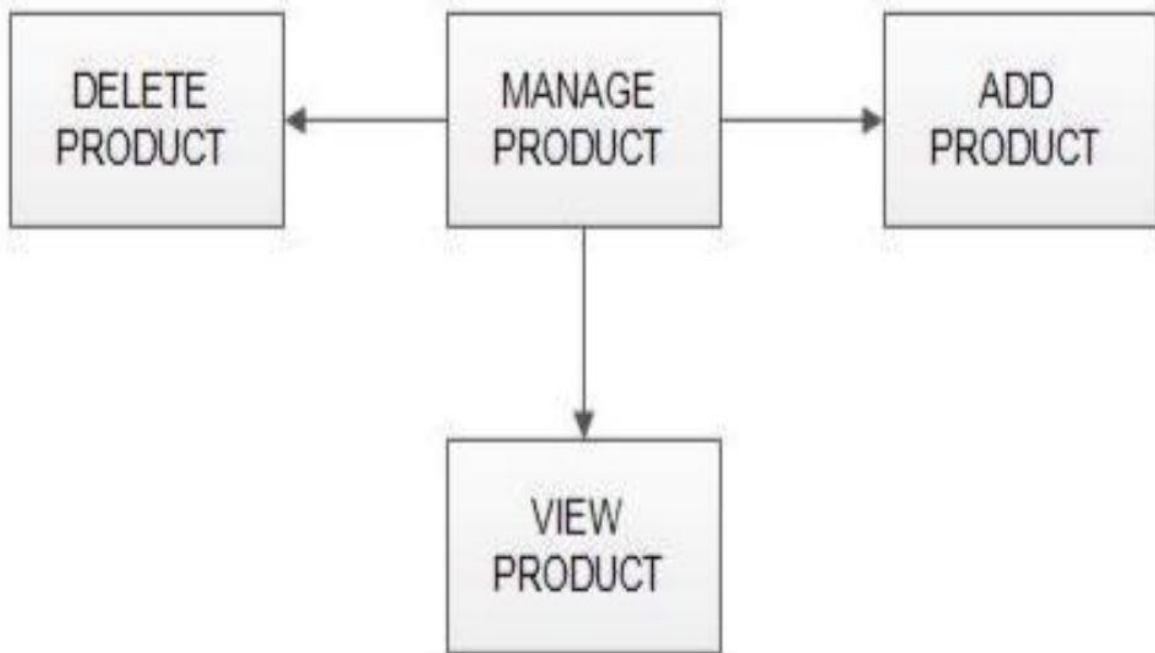
Remove moderator

Admin has privilege to delete a moderator who was added.

Search moderator

All existing moderator can be views by the administrator as a list. If there is number of moderator and admin need to find one of them, the admin can search for a moderator by name.

MANAGE PRODUCTS



Add products

The shopping cart project contains different kinds of products. The products can be classified categories by name. admin can add new products into the existing with all its details including an image.

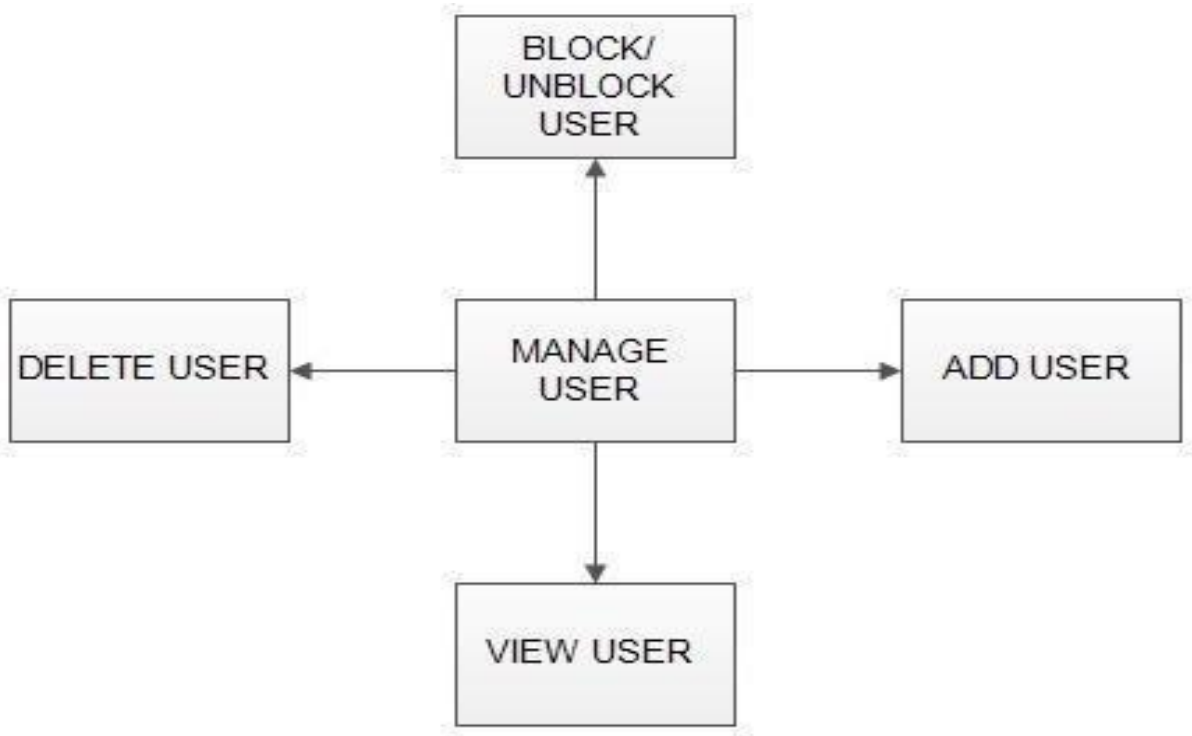
Delete products

Administrator can delete the products based on the stock of that particular product.

Search products

Admin will have a list view off all the existing products. He can also search for a particular product by name.

MANAGE USER



View users

The admin will have a list of view of all users registered in the system. Admin can view all the details of each user in the except password.

Add user

Admin has a privilege to add user directly by providing the details.

Delete and block users

Administrator has a right to delete or block a user. The defaults status of a new user registered is a set as blocked. The admin must accept the new user by unblocking him.

MANAGE ORDERS



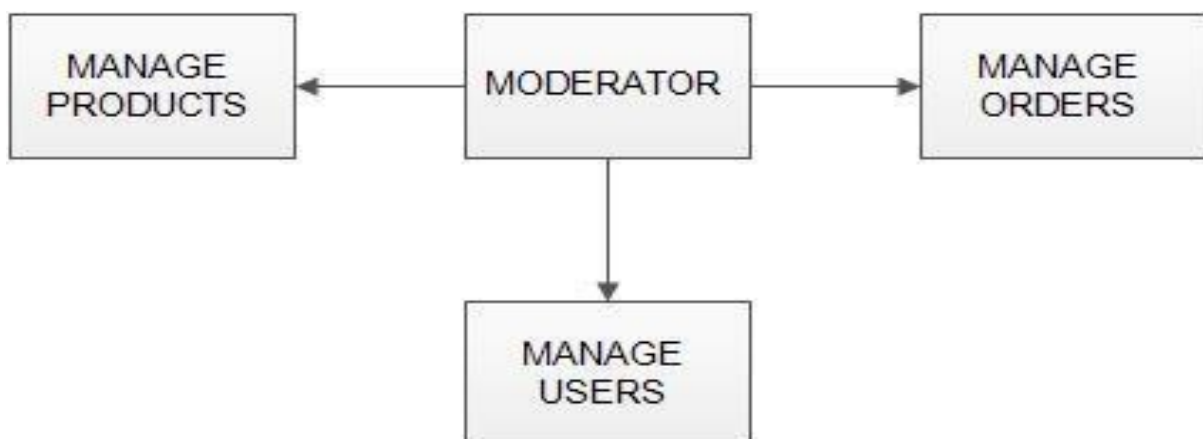
View order

Administrator can view the orders which is generated by the users. He can verify the details of the purchase.

Delete orders

Admin can delete from the order list when the product is taken for delivery.

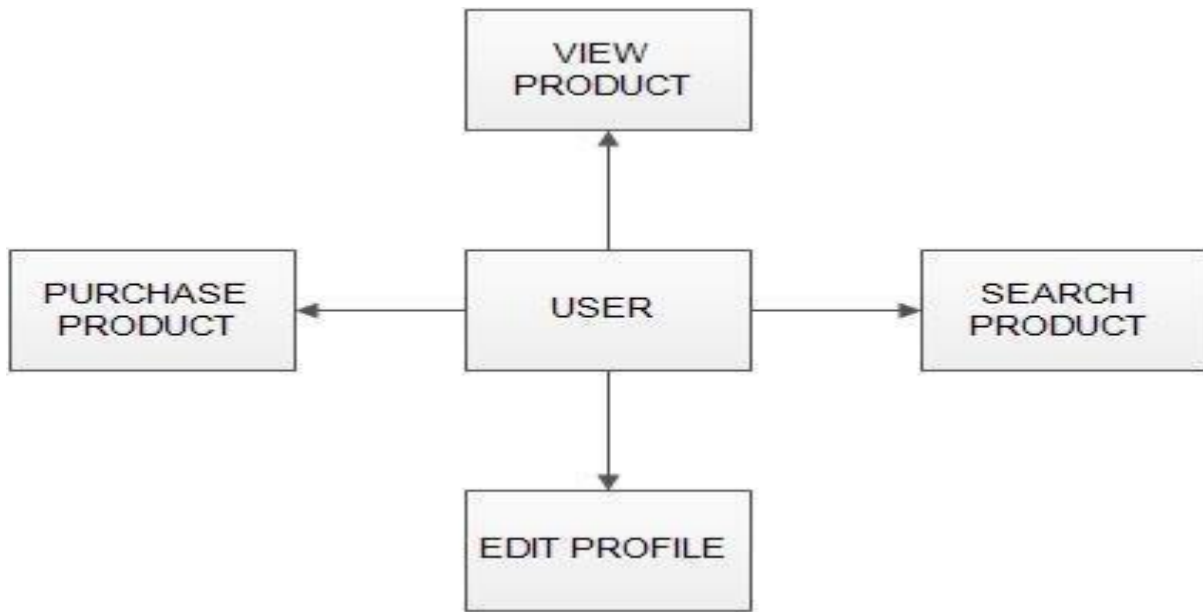
MODERATOR



A moderator is considered as a staff who can manage order for time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the works of admin. Now moderator has all the privilege having except managing other moderator. He can add products and users. He can also check products and users. He can also check the orders and edit his profile.

- Manage products
- Manage users
- Manage orders

USER MODULE



REGISTRATION

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

LOGIN

A user must login with his user-name and password to the system after registration.

VIEW PRODUCT

User can view the list of products based on their names after successful login. A detailed description of a particular product with product name, product details, product images, price can be viewed by users.

SEARCH PRODUCT

User can search for a particular product in the list by name.

ADD TO CART

The user can add the desired products into the cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

SUBMIT CART

After confirming the items in cart user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

HISTORY

In the history the user will have a view of pending orders.

EDIT PROFILE

The user can view can edit the profile.

SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

EXISTING SYSTEM

The current system for shopping is to visit shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item.

- It is less user-friendly
- User must go to shop and select products.
- It is difficult to identify the required products.
- Description to identify the required product.
- It is a time-consuming process.

PROPOSED SYSTEM

In the proposed system customer need not go the for buying the products. He can order the product the customer and product orders.

The system also recommends a home delivery system for the purchased products.

SYSTEM REQUIREMENT SPECIFICATION

GENERAL DESCRIPTION

Product Description

The system consists of two parts. A web application which can provide the online shopping from his smartphone. Web application should be able to help the customer for selecting his item add to the owner in managing the order.

Motivation

as online shopping become a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. for competing with those online brands, if shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

SYSTEM OBJECTIVE

To provide a java application for online shopping of products in an existing shop.

To provide an online shopping web site for the same shop present in area.

SYSTEM REQUIRENMENT

Non-Functional requirement

Functional requirement

Efficiency requirement

When an online shopping java application is implemented customer can purchase product in an efficient manner. It takes less time to shop a product as we know shopping takes large amount of time to buy any stuff.

Reliability Requirement

The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

Usability Requirement

The java application is designed for user friendly environment and ease of use.

Implementation Requirement

Implementation of the system using CSS and html in front end with JSP as back end and it will be used for database connectivity. And the database part is developed in MYSQL. Responsive web designing is used for making the website compatible for any type of screen.

Delivery Requirement

The whole system is expected to be delivered in for months of time with a weekly evaluation by the project guide.

FUNCTIONAL REQUIREMNET

USER

User login:

Description of features

This feature is used by the user to login into system. A user must login with his user name and password to the system after registration. If they are invalid, the user not allowed to enter the system.

Functional requirement

Username and password will be provided after user registration is confirmed.

Password should be hidden from others while typing it in the field.

ADMIN

Manage user

Description of feature:

The administrator can add users, delete users, view users and block users.

Manage moderator

Description of feature

The administrator can add moderators, remove moderators, view moderators.

Manage orders

Description of feature

The administrator can view orders and delete orders.

Functional requirement

The system must identify the login of admin.

Admin account should be secured so that only owner of the shop can access that account.

MODERATOR

Description of features

A moderator is considered as a staff who can manage for the time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the work load of admin. Now moderator has all the privilege of an admin having except managing other moderators. He can manage users and manage products. He can also check the orders and edit his profile.

Functional requirement

The system must identify the login of a moderator

The moderator can easily update the products

SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase is focus on the detailed implementation of the feasible system. Its emphasis on the translating design. Specification to performance specification. System design has two phases of development-

Logical design

Physical design

During logical design phase the analyst describe inputs (sources), outputs, database and procedure all in a format that meets the user requirements. The analyst also specifies the needs of a user at a level that are virtually determines the information flow in and out of the system and the data resources. Here the logical design is followed by physical design or coding. Physical design produces the working system by defining the design specification, which specify exactly what the system must do. The programmers write the necessary program that accept input from the user, perform necessary processing on accepted data and produce the required report on the hard copy or display it in the screen.

INPUT AND OUTPUT DESIGN

Input design:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validation the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operator can be controlled by input design. The user-oriented inputs are converted to as computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input

data are validated and if any data violates any condition, the user warned by a message. If the data satisfy all the condition, it is transferred to the appropriate table in the database. In this project the student details are to be appropriate tables in the database. In this project the student details are to be enter at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exception occurs.

Output design:

Computer output is the most important and direct source of information to the user. Output design is very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the relationship with the users and help on decision making. Allowing the user to view a sample screen is important because the user is the ultimate judge of the quality of output. The output modules of this system is the selected notification.

DATABASE

Database design:

Databases are the storehouse of the data used in the software systems. The data is store in a table and the table is stored in database.

Several tables are the data for the manipulation of setting for a system.

Primary key

The field which is unique for all record occurrences. This field is used to set relationship between tables.

Foreign key

It is a technique to reduce redundancy of data.

SYSTEM TOOLS

The various system tools that has been used in the development of both the front and back end are discussed in this chapter.

FRONT END

JSP, HTML, CSS are utilized to implement the front end.

JSP

Different pages in the application are designed using JSP. A java server pages component is a type of java servlet that is designed to fulfill the role of a user interface for a java web-based application. web developers write JSPS as text files that combines HTML, XML element, and embedded JSP actions and commands. Using JSP, one can collect input form users through web pages.

HTML

Html is a syntax used to format text document on the web.

CSS

CSS Is a style sheet language used for describing the look and formatting of a document written in markup language.

JAVA

This application is delivered as a java application. So, java platform is used to develop the user application.

BACK-END

The back end is implemented using MYSQL which is used to design databases.

MYSQL

MYSQL is the world second most widely used open source relational database management system. The SQL stand for structured query language.

An application known as MYSQL workbench is used to create database.

TABLES

login

login						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
uid	int(11) FIRST		NO	YES	auto_increment	
uname	varchar(50) AFTER `uid`		NO	NO		
upass	varchar(50) AFTER `uname`		NO	NO		
utype	varchar(20) AFTER `upass`		NO	NO		
enabled	varchar(10) AFTER `utype`		NO	NO		

user details

userdetails						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
uid	int(11) FIRST		NO	YES		
uloginname	varchar(50) AFTER `uid`		NO	NO		
uemailid	varchar(100) AFTER `uloginname`		NO	NO		
umobno	varchar(30) AFTER `uemailid`		NO	NO		

product details

product_details

Field Name	Field Type	Default	AllowNull	PriKey	Extra
id	int(11) FIRST		NO	YES	auto_increment
name	varchar(100) AFTER `id`		NO	NO	
description	varchar(500) AFTER `name`		YES	NO	
price	varchar(50) AFTER `description`		YES	NO	
photo	varchar(100) AFTER `price`		YES	NO	

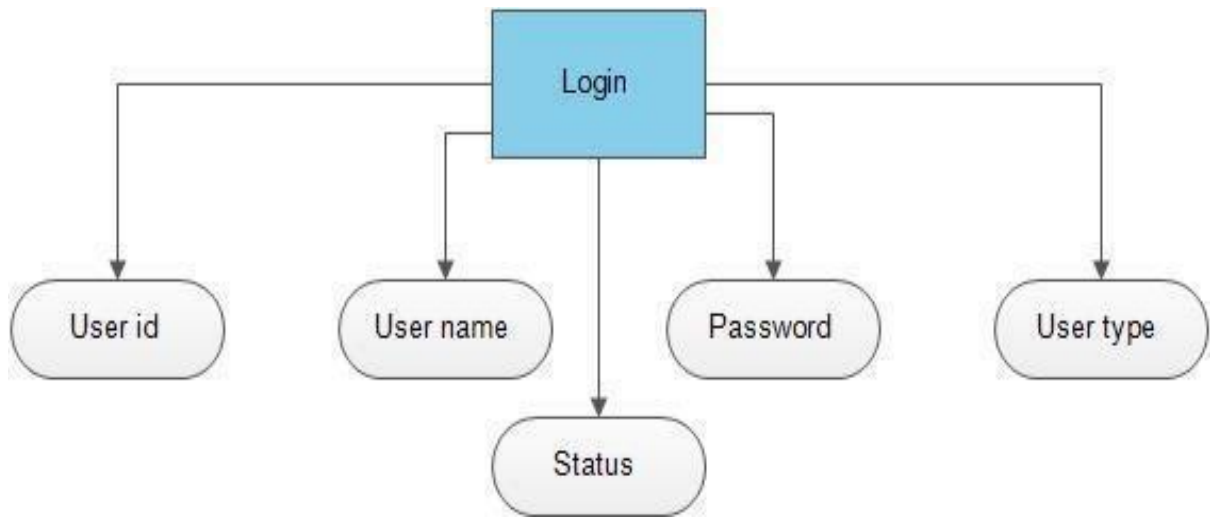
Product orders

product_orders

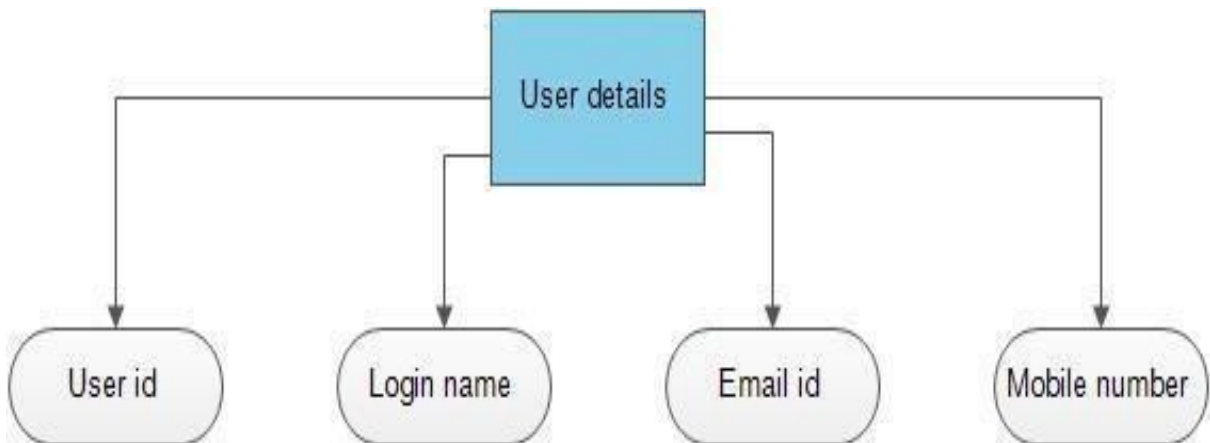
Field Name	Field Type	Default	AllowNull	PriKey	Extra
id	int(11) FIRST		NO	YES	auto_increment
product_id	int(11) AFTER `id`		NO	NO	
user_id	int(11) AFTER `product_id`		NO	NO	
deliver_address	varchar(500) AFTER `user_id`		YES	NO	

E-R DIAGRAMS

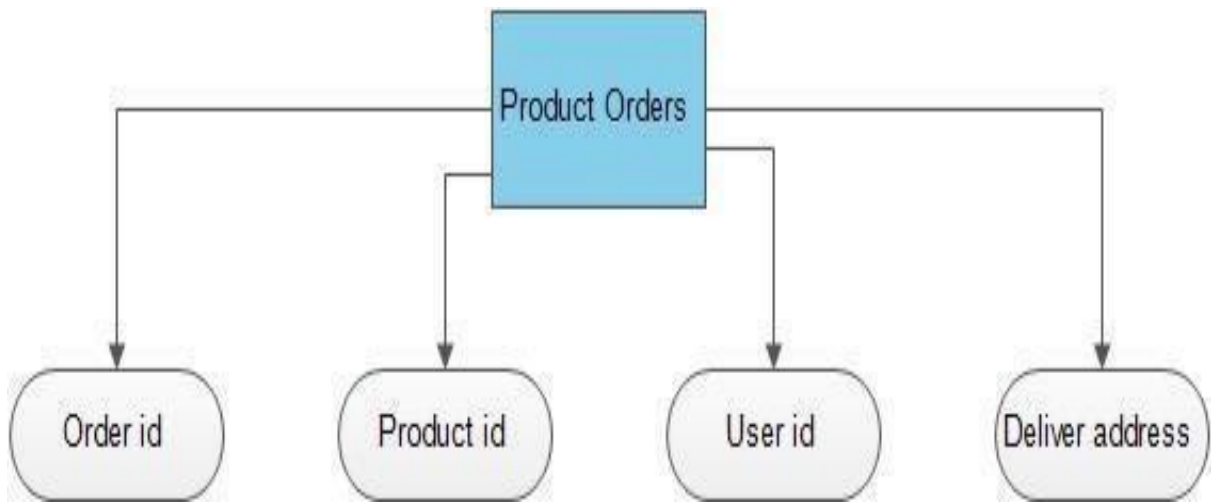
Login



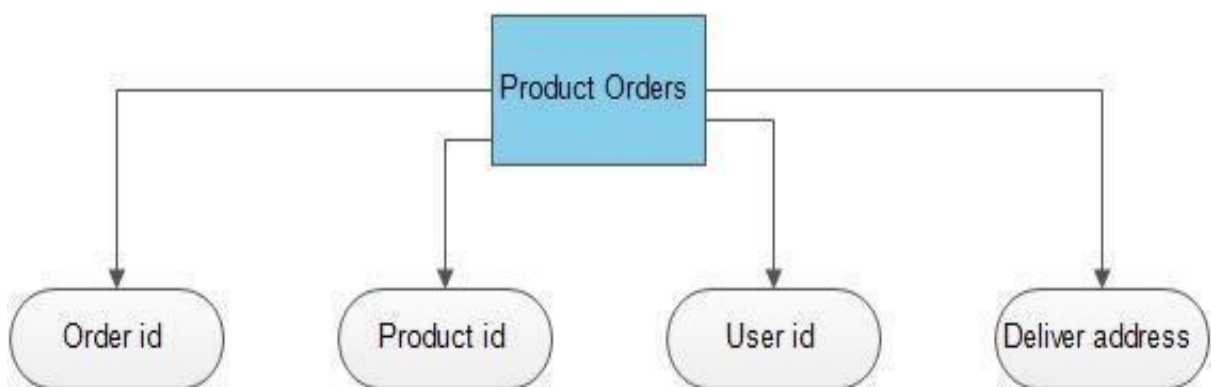
User details



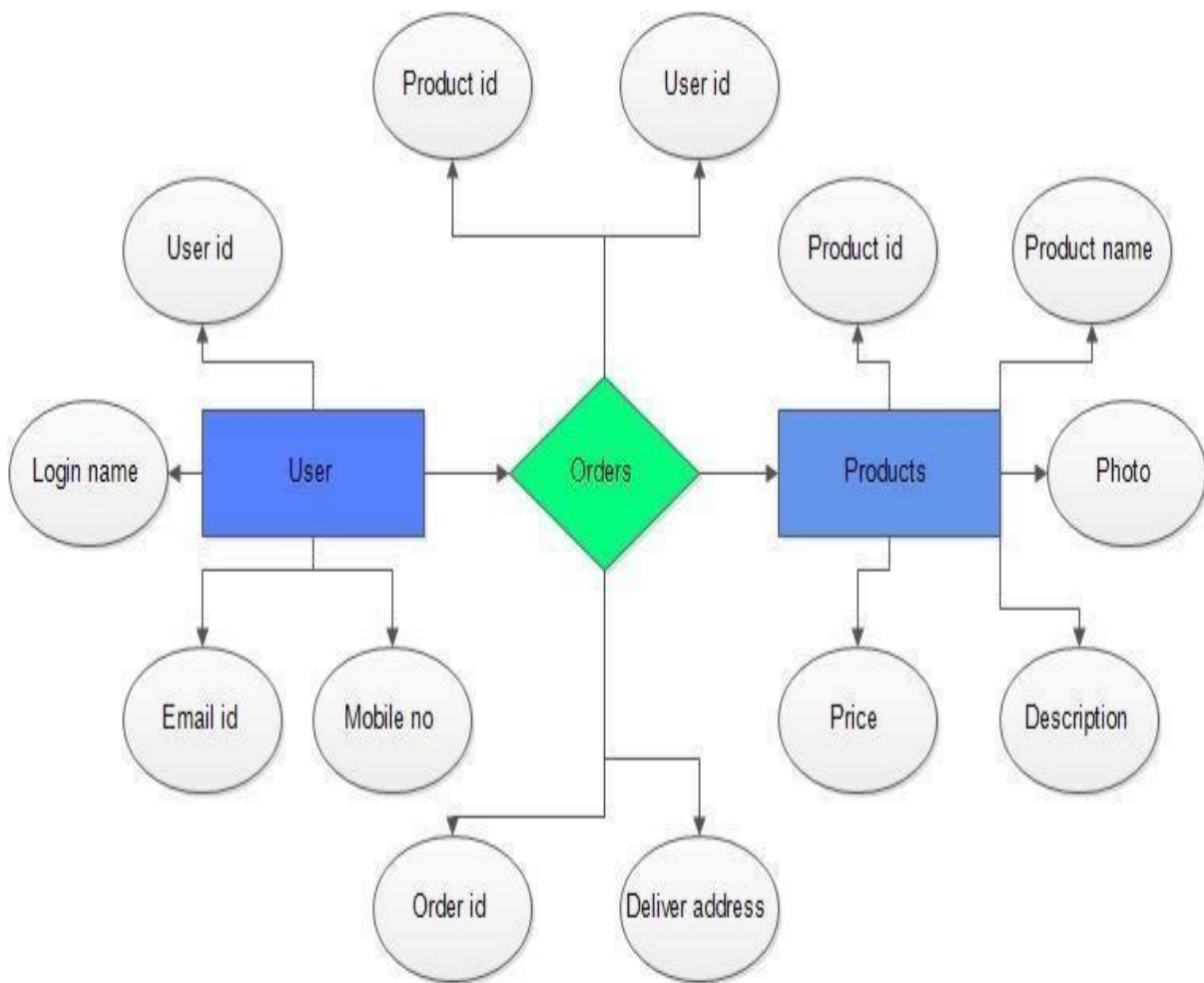
Product details



Product orders



complete diagram



DATA FLOW DIAGRAM

A data flow diagram is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describe the flow of data and the process that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply and physical implement. It has the purpose of clarifying system.

Requirements and identifying major transformation. So, it is the string point of the design phase that functional decompose the

requirement specification down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons often referred to as logical data flow chart diagram.

EXTERNAL ENTITY

An external entity is a source or destination of a data flow. Only those entities which originate or receive data are represented on a data flow diagram. The symbol used is a regular box.

PROCESS

A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.

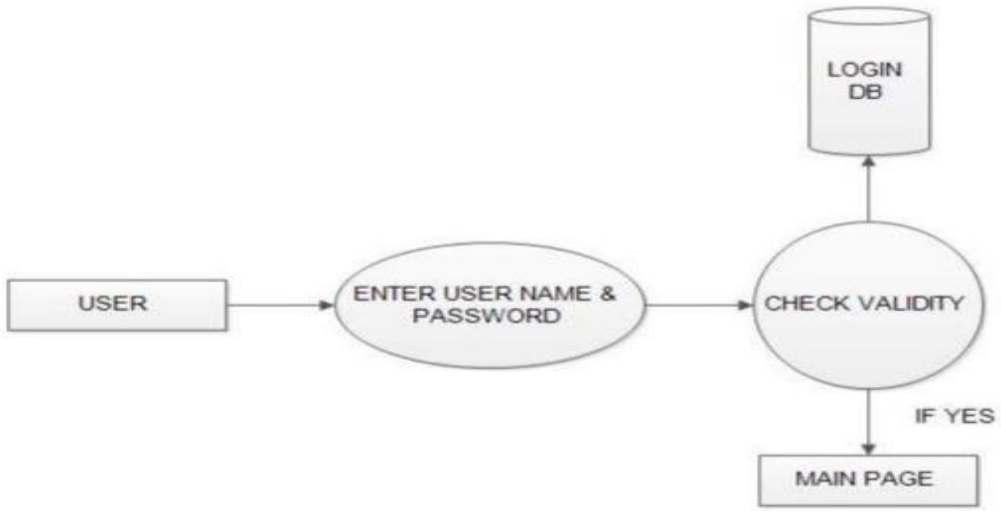
DATAFLOW

The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowhead showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referred by the process or data stores at its head and tail, or by a description of its contents.

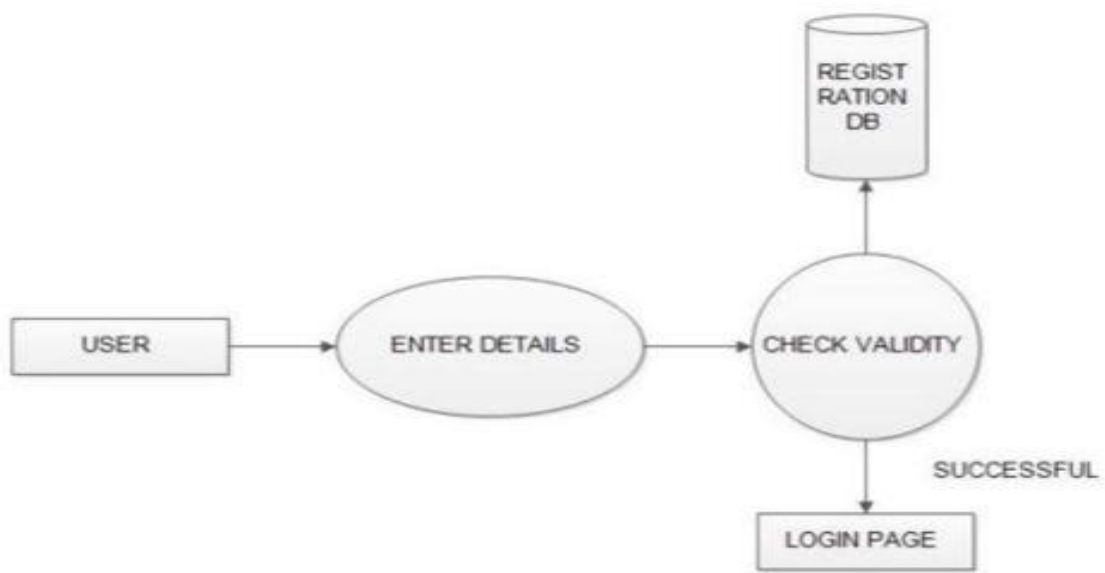
DATA STORE

A data store is a holding place for information within the system. It represented by an open-ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or ma be short-term accumulations. For example, batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

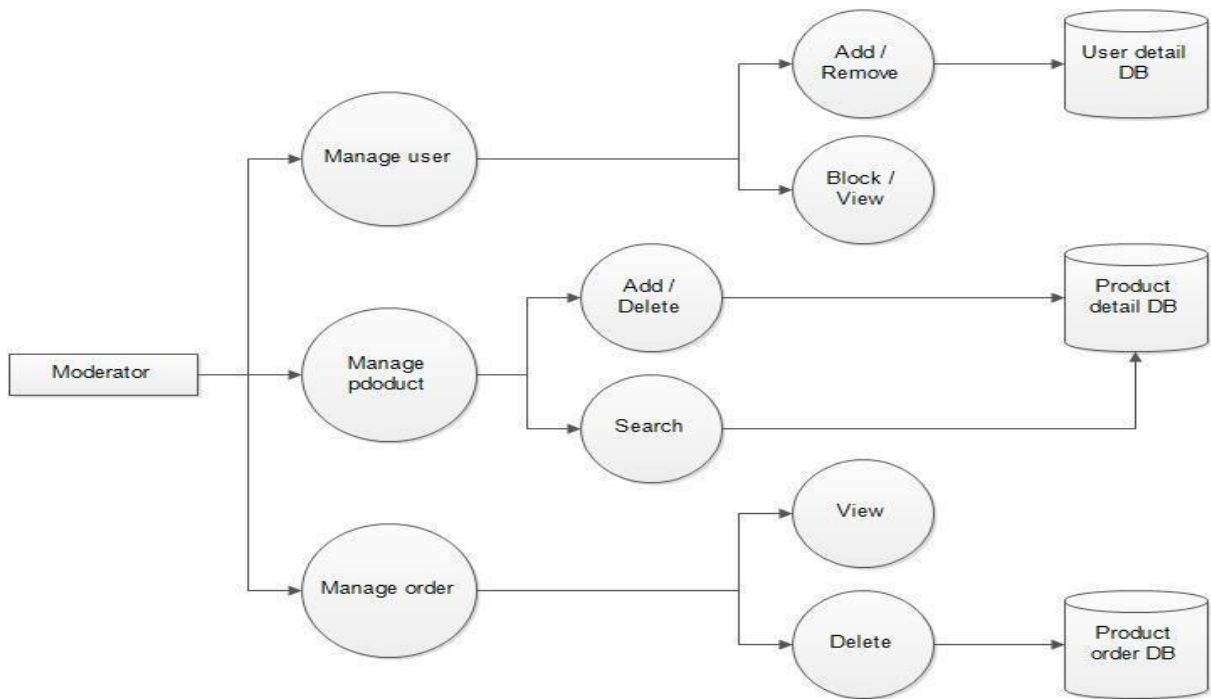
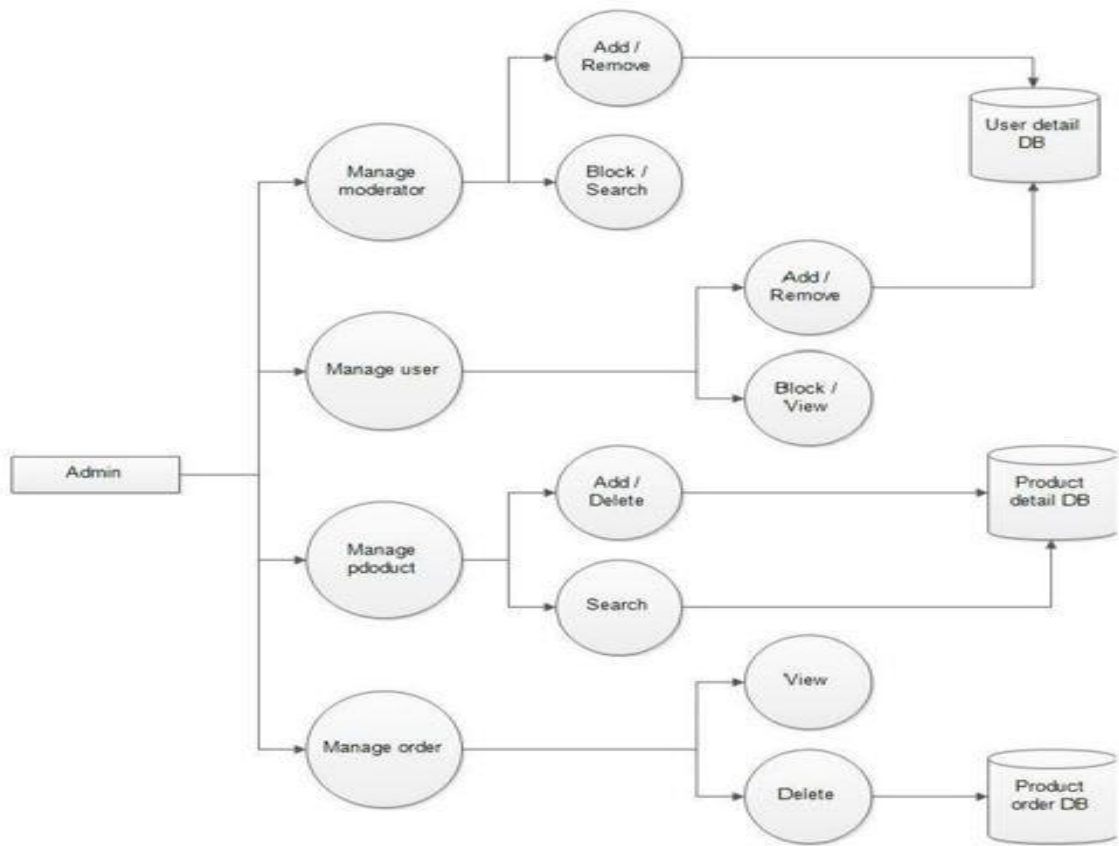
Login DFD:



Registration DFD:



Admin DFD



Moderator DFD:

SAMPLE CODE

```
import javax.servlet.*;
import java.sql.*;

public class ConListener implements ServletContextListener
{
    public void contextInitialized(ServletContextEvent event)
    {
        try
        {
            //Class.forName("oracle.jdbc.driver.OracleDriver");

            //Connection
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","onlineshop","onlineshop");

            Class.forName("com.mysql.jdbc.Driver");

            Connection
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/onlineshop","root","");

            PreparedStatement insertUser=con.prepareStatement("insert into userinfo values(?,?,?,?,?)");

            PreparedStatement
            getUserWhereID=con.prepareStatement("select * from userinfo where email=?");

            PreparedStatement
            updateUser=con.prepareStatement("update userinfo set name=?,ph=?,addr=? where email=?");
```

```
PreparedStatement getItem=con.prepareStatement("select *  
from itemdetails");
```

```
PreparedStatement  
getItemWhereCateg=con.prepareStatement("select * from  
itemdetails where categ=?");
```

```
PreparedStatement  
getMaxOrderID=con.prepareStatement("select max(oid) from  
orderbook");
```

```
PreparedStatement  
insertOrder=con.prepareStatement("insert into orderbook  
values(?,?,?,?)");
```

```
PreparedStatement  
insertOrderItem=con.prepareStatement("insert into orderitem  
values(?,?,?,?)");
```

```
PreparedStatement getOrders=con.prepareStatement("select  
* from orderbook where odate=?");
```

```
PreparedStatement  
getOrderItems=con.prepareStatement("select * from orderitem  
where oid=?");
```

```
PreparedStatement insertItem=con.prepareStatement("insert  
into itemdetails values(?,?,?,?)");
```

```
PreparedStatement  
getItemWhereName=con.prepareStatement("select * from  
itemdetails where iname=?");
```

```
PreparedStatement  
updateItem=con.prepareStatement("update itemdetails set  
iname=?,price=?,image=?,categ=? where iname=?");
```

PreparedStatement

```
updateItem2=con.prepareStatement("update itemdetails set  
iname=?,price=?,categ=? where iname=?");
```

PreparedStatement

```
deleteItem=con.prepareStatement("delete from itemdetails where  
iname=?");
```

```
Statement st=con.createStatement();
```

```
ServletContext ctx=event.getServletContext();
```

```
ctx.setAttribute("stmt", st);
```

```
ctx.setAttribute("dbcon", con);
```

```
ctx.setAttribute("getItem", getItem);
```

```
ctx.setAttribute("insertUser", insertUser);
```

```
ctx.setAttribute("getUserWhereID", getUserWhereID);
```

```
ctx.setAttribute("updateUser", updateUser);
```

```
ctx.setAttribute("getItemWhereCateg", getItemWhereCateg);
```

```
ctx.setAttribute("getMaxOrderID", getMaxOrderID);
```

```
ctx.setAttribute("insertOrder", insertOrder);
```

```
ctx.setAttribute("insertOrderItem", insertOrderItem);
```

```
ctx.setAttribute("getOrders", getOrders);
```

```
ctx.setAttribute("getOrderItems", getOrderItems);
```

```
ctx.setAttribute("insertItem", insertItem);
```

```
ctx.setAttribute("getItemWhereName", getItemWhereName);
```

```
ctx.setAttribute("updateItem", updateItem);
```

```
ctx.setAttribute("updateItem2", updateItem2);
```

```
ctx.setAttribute("deleteItem", deleteItem);
```

```

    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}

public void contextDestroyed(ServletContextEvent e) {}
}

```

SAMPLE JSP:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8" />
<title>Online Shop</title>
<style type="text/css">
</style>
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div class="container">
  <div class="header">
    <h5>Online Shop</h5>
    <p><a href="index.html">Home</a></p>
  </div>
  <div class="block5"></div>
  <iframe src="slide.html" style="width:960px;height:365px;max-
width:100%;overflow:hidden;border:none;padding:0;margin:0 auto;

```

```

display:block;" marginheight="0"
marginwidth="0"></iframe>
<div class="content">
  <div class="welcome">
    <h3>WELCOME TO OUR SITE</h3>
    
  </div>
  <p>ERestaurant, offers guests a fun and casual dining experience.
The locally owned neighborhood favorite boasts a menu full of
mouth-watering appetizers, comfort all-American favorites, and
delectable desserts. In the evenings, Geogeske transforms into a hip
lounge with a full-service bar and a mix of sultry beats by local
Deejays.</p>
  <div class="login">
    <h4>Admin Login Form</h4>
    <div class="login1">
      Admin ID:<br/><br/><br/>
      Password:<br/><br/>
    </div>
    <form action="adminprocess.jsp" method="post">
      <div class="login2">
        <input class="lform" name="aid" type="text" /><br/><br/>
        <input class="lform" name="pass" type="password"
/><br/><br/>
      </div>
      <div class="login3">
        <input class="bluebutton" value="Login" type="submit" />
      </form>
    </div>
  </div>
  <div class="footer">Design& Develop by Tanisha</div>
</div>
</body></html>

```

REFERENCES

MYSQL cookbook by Paul Dubois

MC. GrawHILL'S java

Compete CSS guide, Maxine Sherrin and John Allsopp.

SQL in 21 days by Sams.