CHAPTER 3
Analysis & Design

3.1 System Requirements (Functional and Non-Functional Requirements)

Functional Requirements

User Registration and Login:

Users should be able to register on the Surbhi Pure Veg Restaurant website using their email addresses.

User accounts should be protected with secure authentication mechanisms to prevent unauthorized access.

Upon successful registration, users should receive a verification email to activate their accounts.

Menu Management:

Admins should have the ability to add, update, and remove menu items from the online inventory.

Menu items should be categorized and organized for easy browsing by customers.

Admins should be able to set attributes such as item name, description, price, and availability status.

Sales and Inventory Management:

Admins should be able to monitor sales data and track inventory levels in real-time.

Alerts should be generated for low stock levels to prompt restocking activities.

Admins should have the ability to analyze item performance and identify popular dishes for promotional activities.

Reporting:

The admin dashboard should include various reports such as sales reports, inventory reports, and revenue analysis.

Reports should be customizable and provide insights into key metrics to support decision-making processes.

User Account Management:

Admins should have the authority to manage user accounts, including account modification and removal.

User accounts should store relevant information such as contact details, order history, and preferences.

Data Security:

Admins should implement robust data security measures to protect sensitive customer and business information.

Encryption techniques should be employed to safeguard data during transmission and storage.

Regular security audits and updates should be conducted to mitigate potential risks and vulnerabilities.

Feedback Management:

Admins should have access to a feedback management system to review and respond to user-provided feedback.

Feedback should be categorized and analyzed to identify areas for improvement and address customer concerns effectively.

Dashboard and Analytics:

Admins should have access to a comprehensive dashboard providing key insights into website performance, customer behavior, and sales trends.

Analytics tools should be integrated to track user engagement, conversion rates, and other relevant metrics.

Payment Integration:

Users should be able to make payments securely for menu items using various payment methods such as credit/debit cards, digital wallets, and bank transfers.

Payment gateways should be integrated to facilitate seamless transactions and ensure data privacy and security.

Search and Filtering:

Users should be able to search for menu items based on various criteria such as category, price range, and specific dietary preferences.

Advanced filtering options should be available to refine search results and facilitate menu discovery.

Non-Functional Requirements

User Interface:

The system should have a user-friendly interface that is visually appealing and intuitive to navigate.

Responsive design principles should be employed to ensure compatibility with different screen sizes and devices.

Performance:

The website should load quickly and respond promptly to user interactions to provide a seamless browsing experience.

Page load times should be optimized, and caching mechanisms should be implemented to enhance performance.

Security:

The system should adhere to industry-standard security protocols and best practices to protect user data and prevent unauthorized access.

Secure Sockets Layer (SSL) encryption should be implemented to secure data transmission over the internet.

Scalability:

The system architecture should be designed to scale horizontally and vertically to accommodate increasing user traffic and data volumes.

Load balancing and clustering techniques should be employed to distribute workload and optimize resource utilization.

Reliability:

The system should be highly available and resilient to minimize downtime and service interruptions.

Redundancy and failover mechanisms should be implemented to ensure continuous operation in the event of hardware or software failures.

Compatibility:

The system should be compatible with a wide range of web browsers, devices, and operating systems to reach a broader audience.

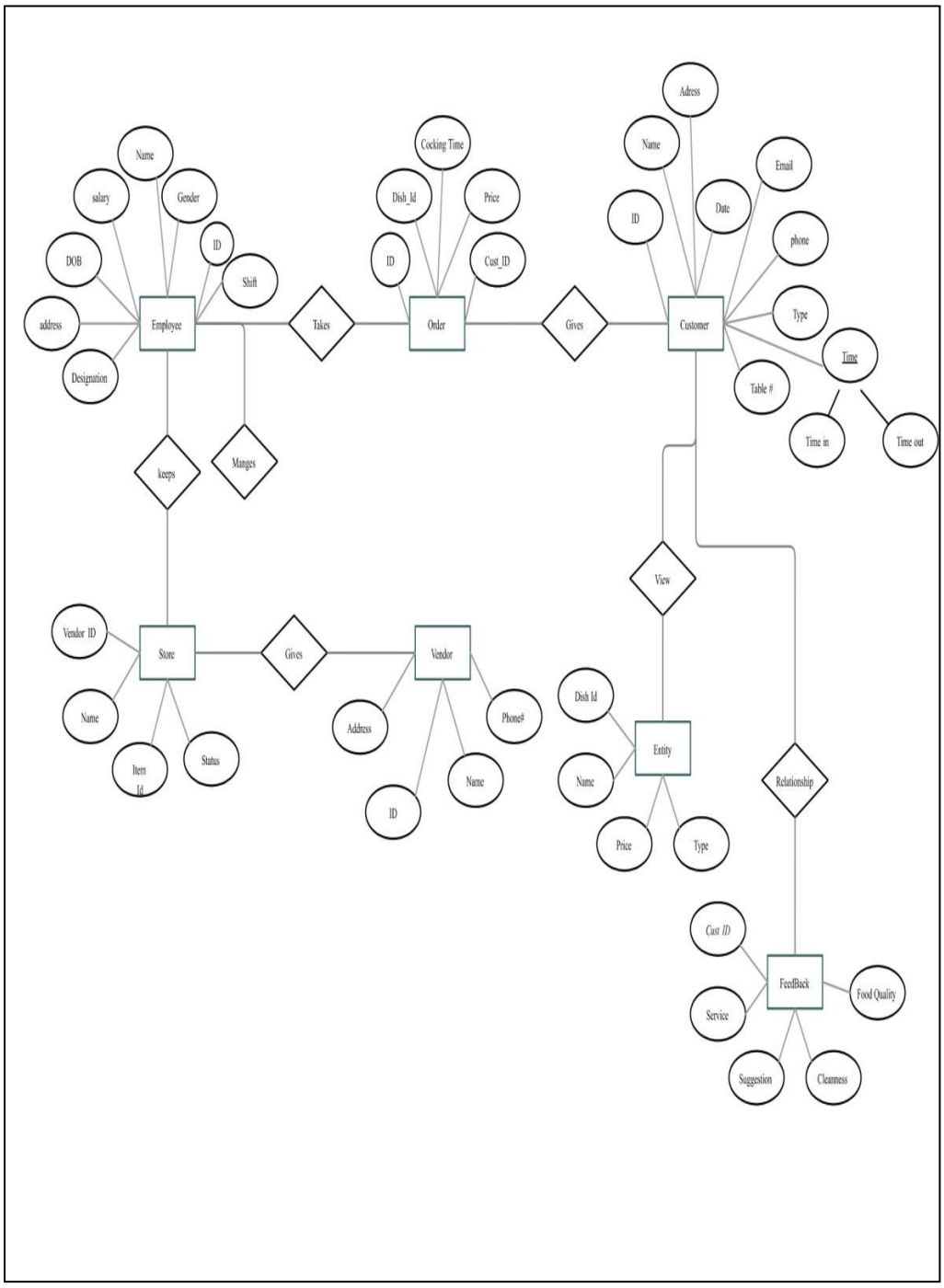
Compatibility testing should be performed regularly to identify and address any compatibility issues.

Maintenance and Support:

The system should be easy to maintain and update, with regular software patches and updates released to address security vulnerabilities and bugs.

Technical support channels should be available to assist users with inquiries, issues, and troubleshooting steps.

3.2 Entity-Relation Diagram:



3.3.1

Table Name	**orders**
Primary Key	**orderid**
Foreign Key	**productid**
Description of Table	**User information is stored in this table**

Sr. No	**Field Name**	**Datatype with size**	**Constraints**	**Description**
1	orderid	varchar (45)	Primary Key	Order unique number to differentiate
2	prodid	varchar (45)	Foreign Key	Product unique number to differentiate which is present in the user's cart
3	quantity	int	Not Null	Quantity of product in the cart
4	amount	Decimal (10,2)	Not Null	Total amount of product from cart
5	shipped	int	Not Null	Order status

3.3.2

Table Name	**orders**
Primary Key	orderid
Foreign Key	productid
Description of Table	User information is stored in this table

Sr. No	**Field Name**	**Datatype with size**	**Constraints**	**Description**
1	orderid	varchar (45)	Primary Key	Order unique number to differentiate
2	prodid	varchar (45)	Foreign Key	Product unique number to differentiate which is present in the user's cart
3	quantity	int	Not Null	Quantity of product in the cart
4	amount	Decimal (10,2)	Not Null	Total amount of product from cart
5	shipped	int	Not Null	Order status

3.3.3

Table Name	**orders**
Primary Key	**orderid**
Foreign Key	**productid**
Description of Table	**User information is stored in this table**

Sr. No	**Field Name**	**Datatype with size**	**Constraints**	**Description**
1	transid	varchar (45)	Primary Key	Transaction unique number to evaluate the necessary details
2	transorderid	varchar (45)	Foreign Key	Order unique number related to this transaction
3	truserid	varchar (60)	Foreign Key	User unique number related to this transaction
4	time	datetime	Not Null	Display date and time of transaction
u5	amount	decimal (10,2)	Not Null	Amount to be paid while checkout

3.3.4

Table Name	**user**
Primary Key	email
Foreign Key	-
Description of Table	The order information will be stored in this table

Sr. No	**Field Name**	**Datatype with size**	**Constraints**	**Description**
1	email	varchar (60)	Primary Key	Email of a user
2	password	varchar (20)	Not Null	Password of a user
3	name	varchar (30)	Not Null	Name of a user
4	mobile	bigint	Not Null	Contact number of users
5	address	varchar (250)	Not Null	Delivery addresses of users
6	pincode	int	Not Null	Pin code to check availability of delivery or not

3.3.5

Table Name	usercart
Primary Key	-
Foreign Key	prodid, email
Description of Table	The order information will be stored in this table

Sr. No	Field Name	Datatype with size	Constraints	Description
1	username	varchar (60)	Not Null	Username of the user
2	prodid	varchar (45)	Not Null	Product unique number to differentiate items in the user's cart
3	quantity	int	Not Null	Quantity of the product

3.3.5

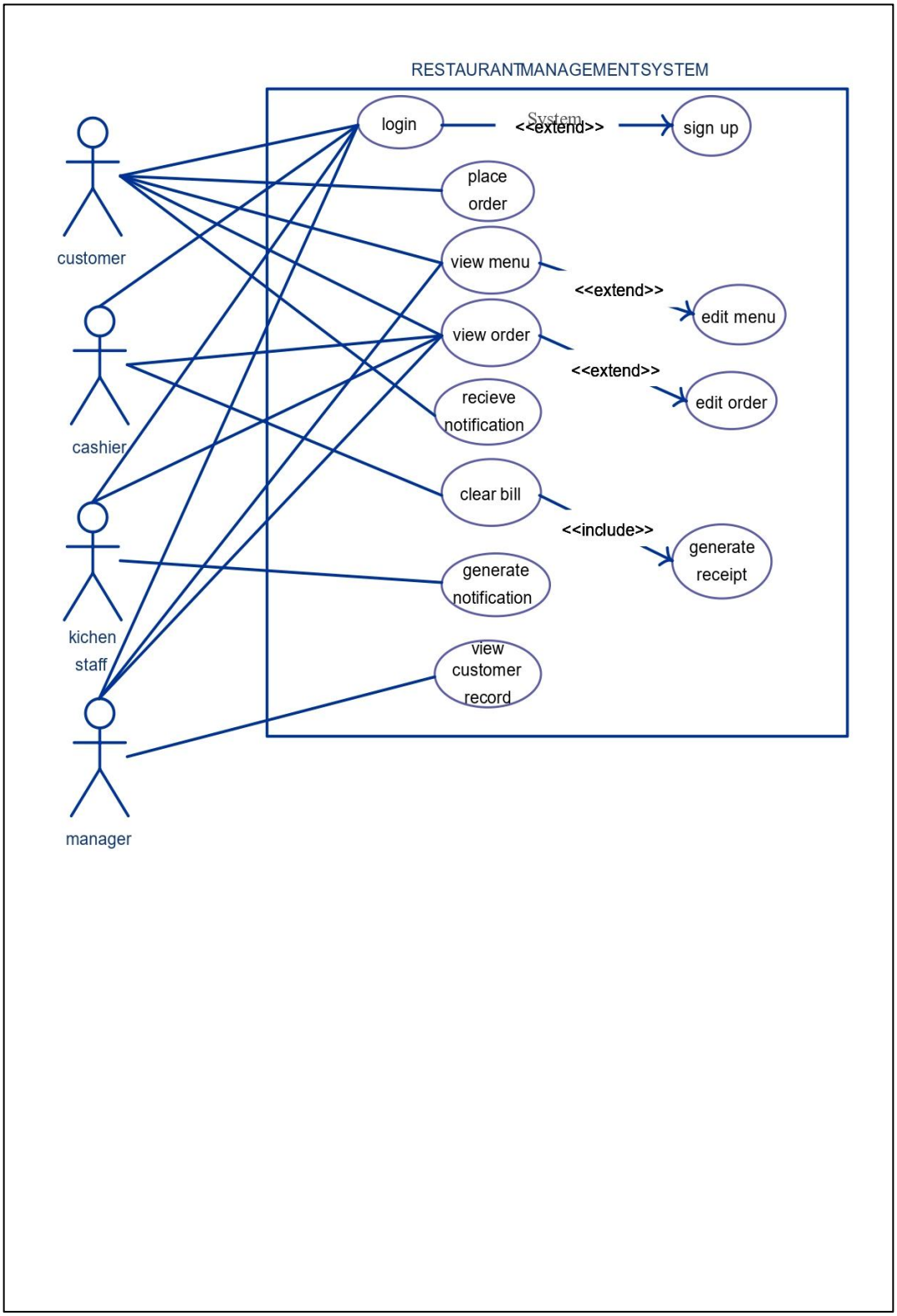
Table Name	user_demand
Primary Key	username
Foreign Key	prodid, email
Description of Table	The order information will be stored in this table

Sr. No	Field Name	Datatype with size	Constraints	Description
1	username	varchar (60)	Primary Key	Username of the customer
2	prodid	varchar (45)	Foreign Key	Product unique number
3	quantity	int	Not Null	Quantity of the product

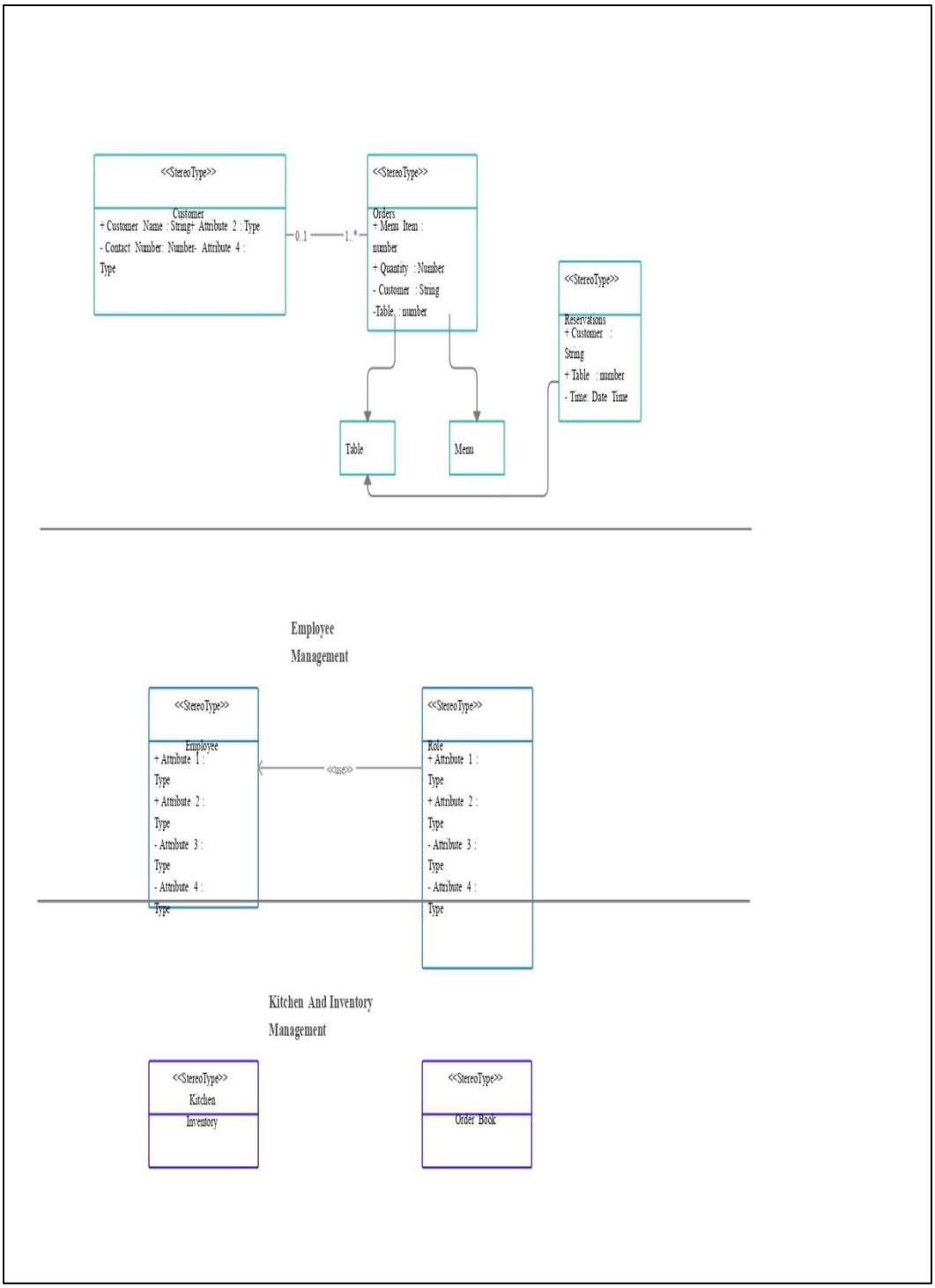
3.3.6 Data Dictionary:

Sr. No	Field Name	Data Type	Description
1	address	varchar (250)	Delivery addresses of users
2	amount	Decimal (10,2)	Total amount of product from cart
3	email	varchar (60)	Email of a user
4	image	longblob	Images of product
5	mobile	bigint	Contact number of users
6	name	varchar (30)	Name of a user
7	orderid	varchar (45)	Order unique number
8	password	varchar (20)	Password of a user
9	pid	varchar (45)	Product unique number present in inventory
10	pincode	int	Pin code to check availability of delivery or not
11	pinfo	varchar (350)	Information / description of the product
12	pname	varchar (100)	Name of the product
13	pprice	decimal (12,2)	Price of the product
14	pquantity	int	Product quantity in stock
15	prodid	varchar (45)	Product unique number to differentiate items in the user's cart
16	ptype	varchar (20)	Type of the product
17	quantity	int	Quantity of product in the cart
18	shipped	int	Order status
19	time	datetime	Display date and time of transaction
20	transid	varchar (45)	Transactions unique number to evaluate the necessary details
21	username	varchar (60)	Username of the customer

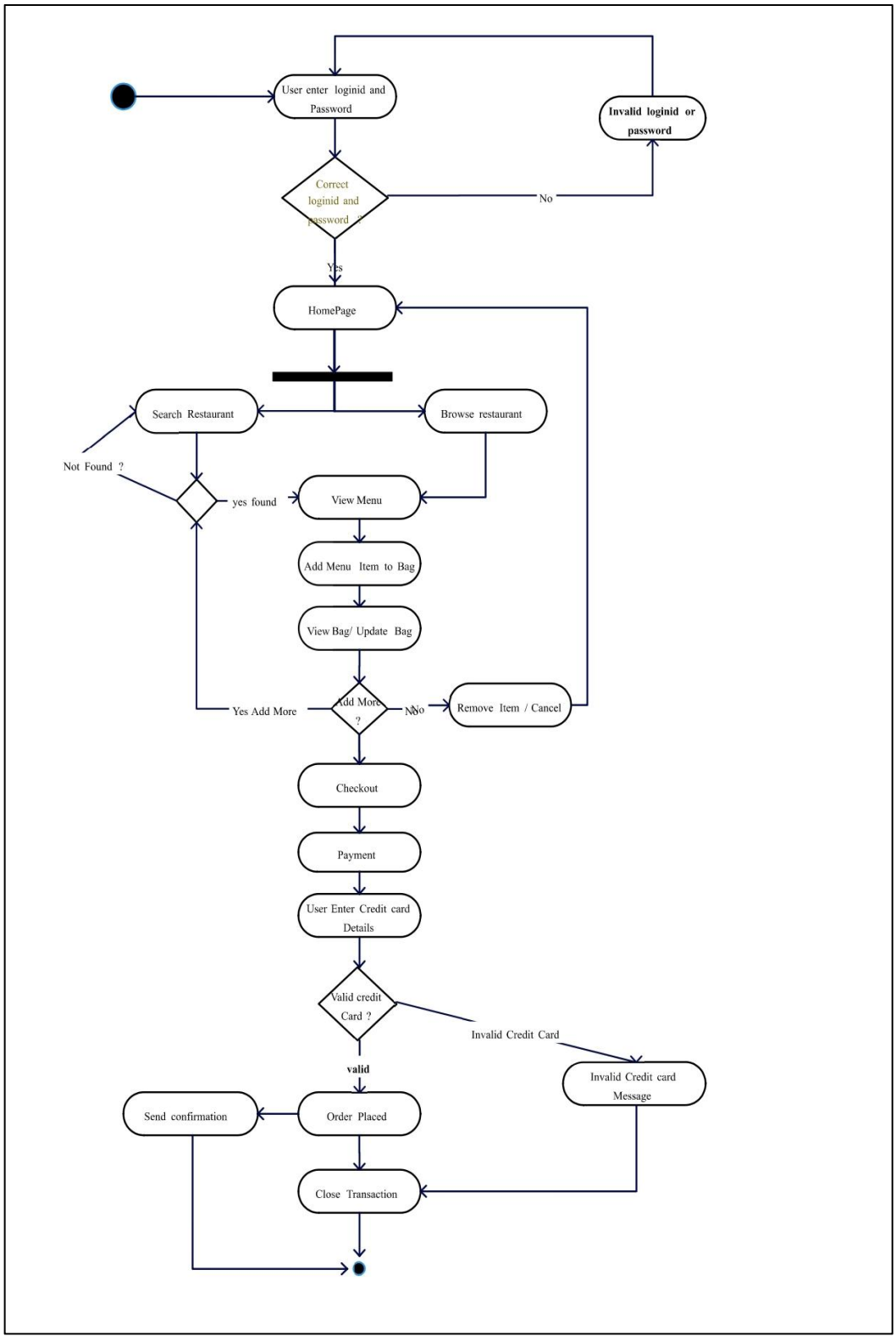
3.4 Use Case Diagrams:



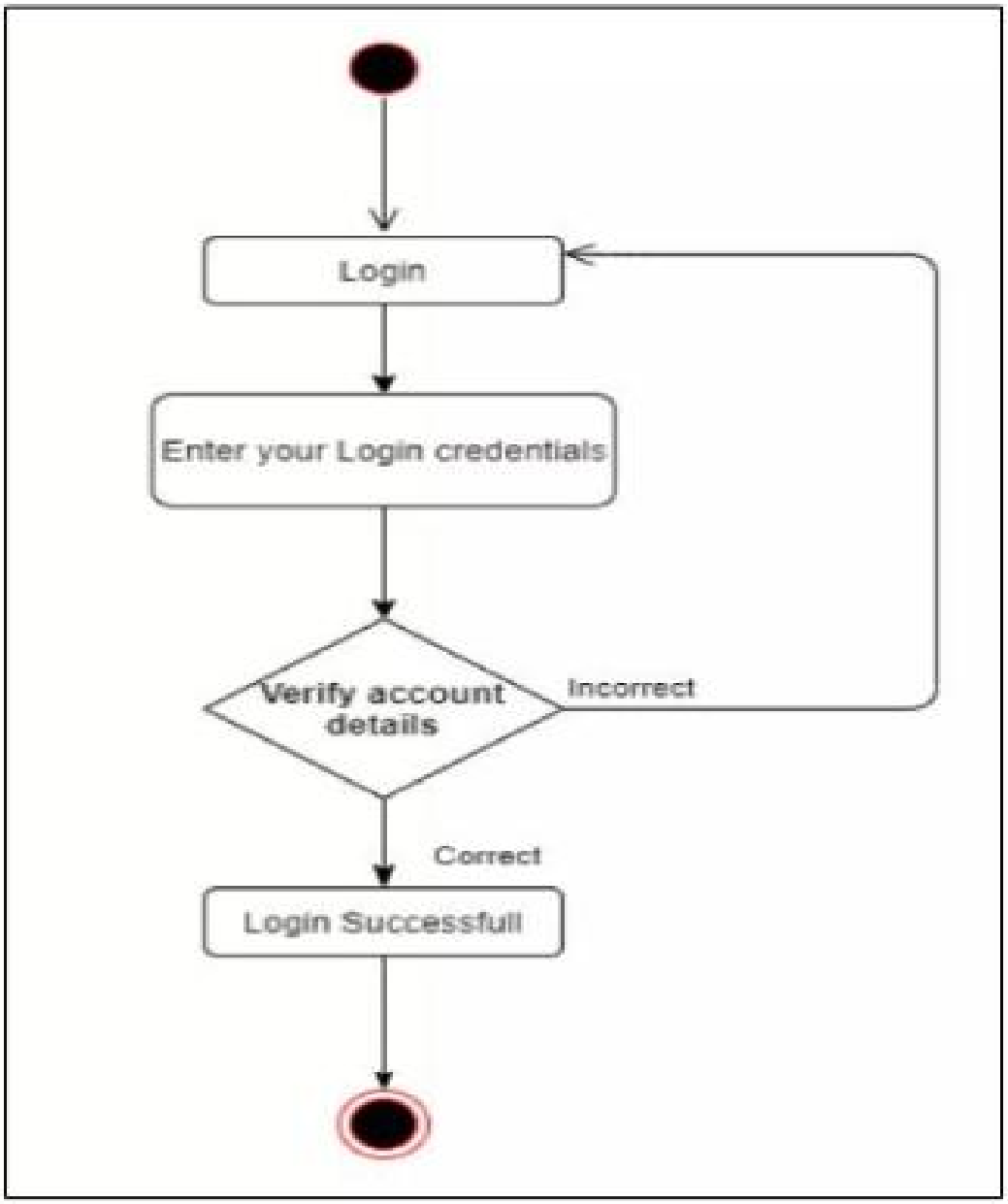
3.5 Class Diagram:



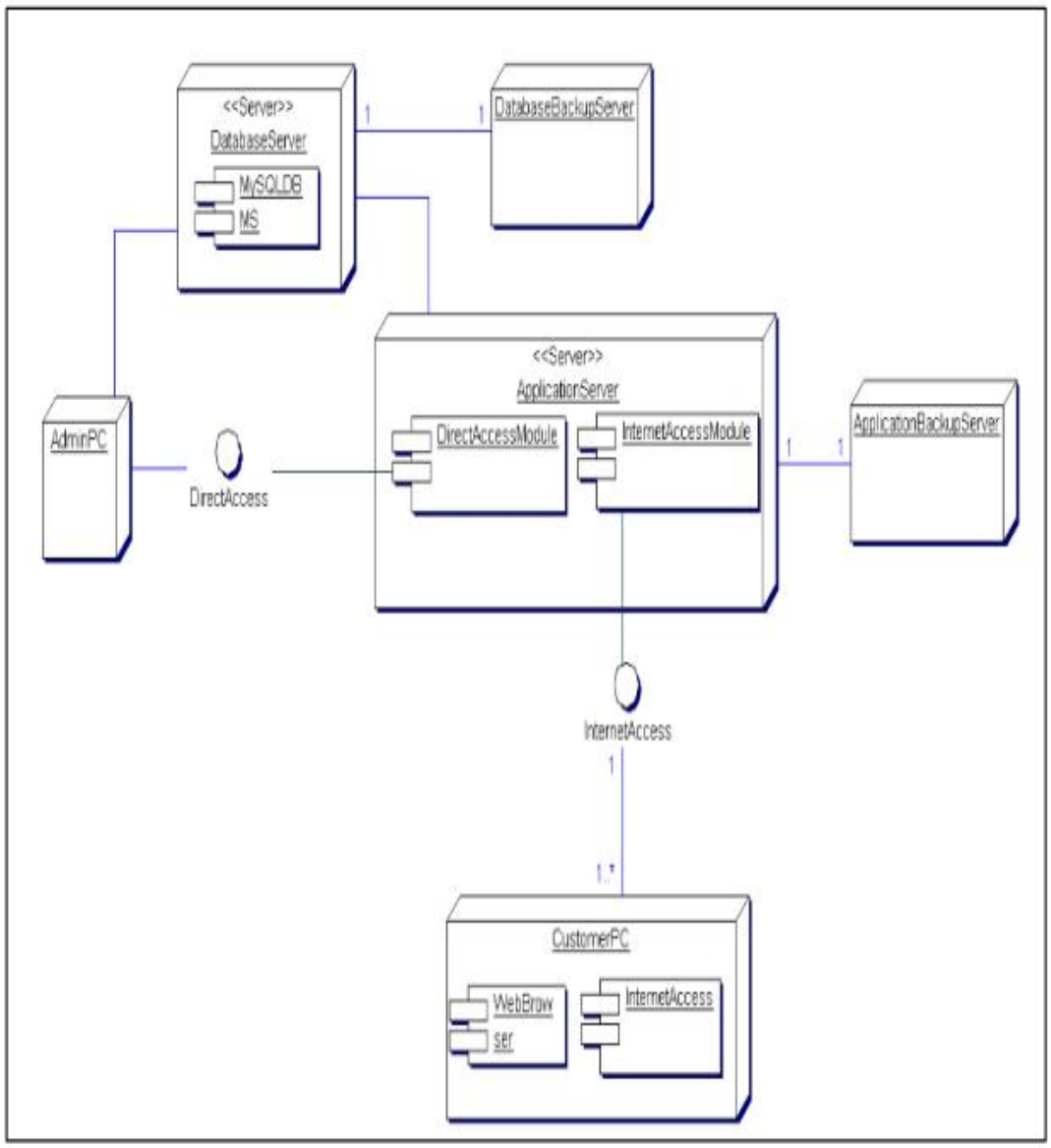
3.6 Activity Diagram:



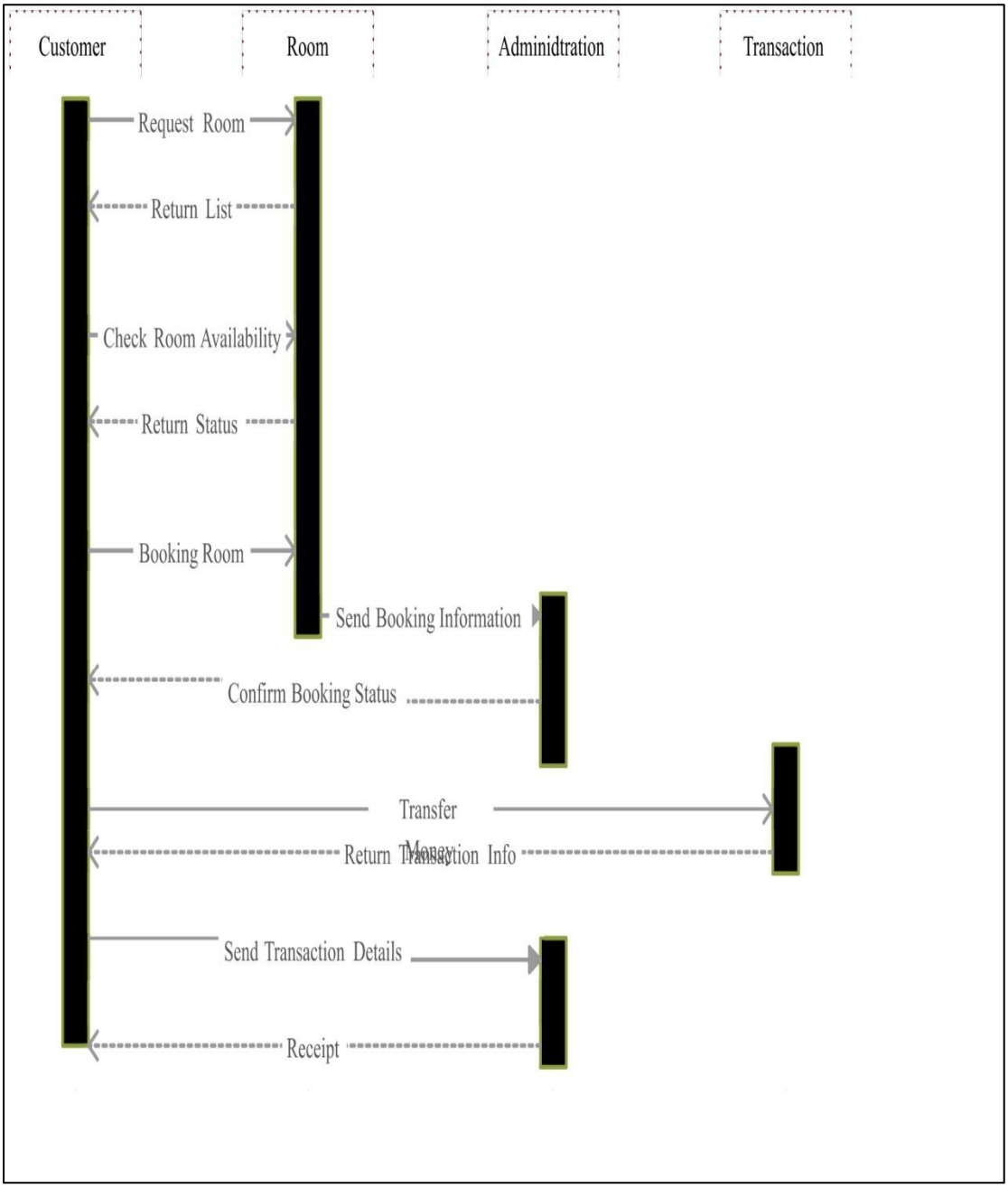
Activity Diagram for Login:



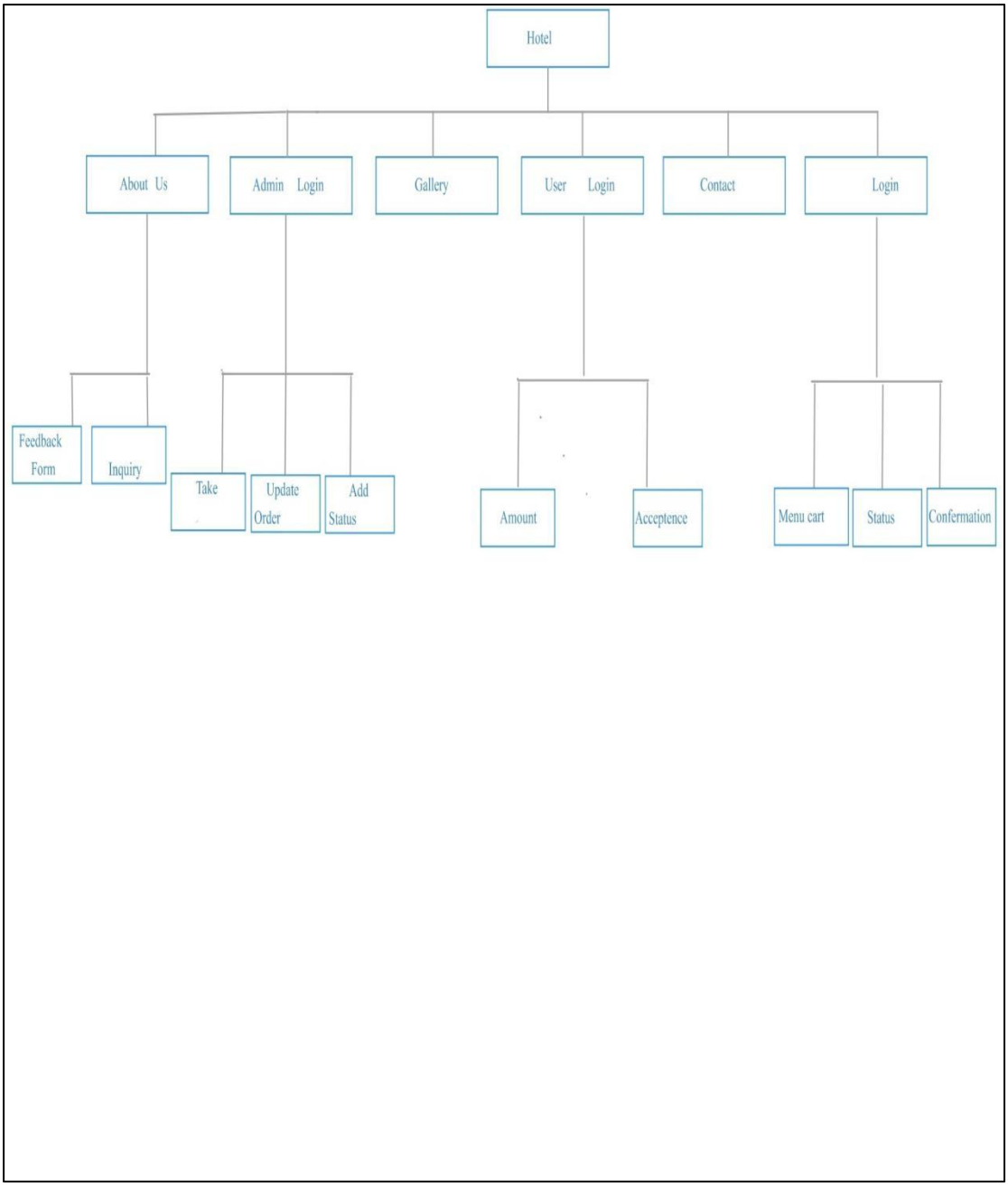
3.7 Deployment Diagram:



Sequence Diagram of System:

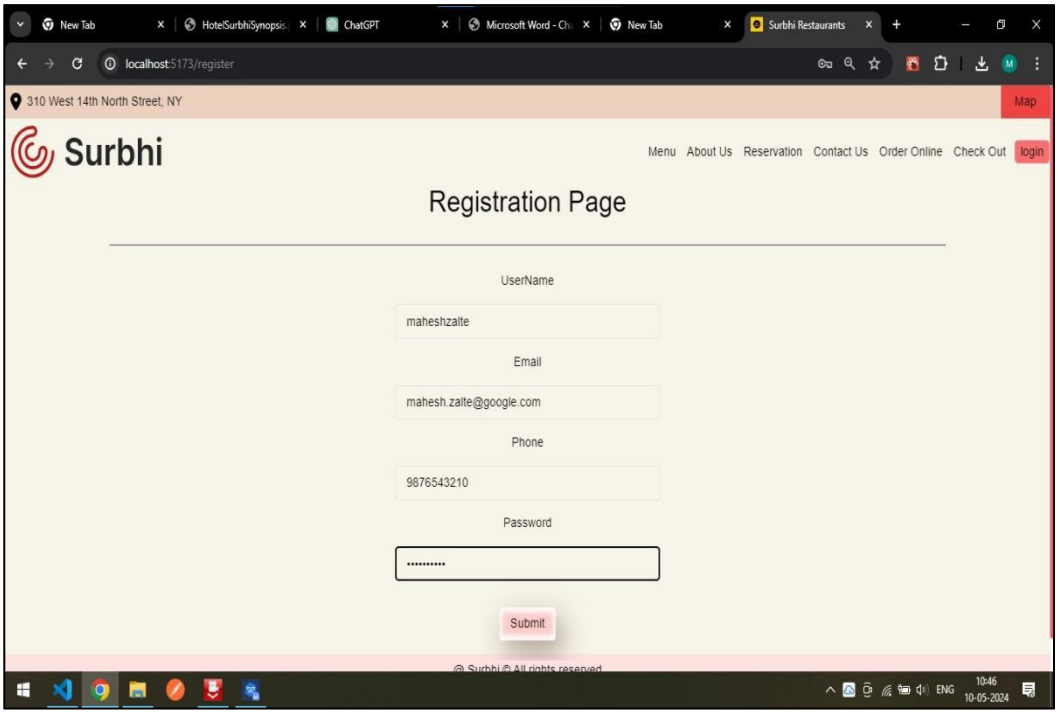


3.8 Module Hierarchy Diagram:

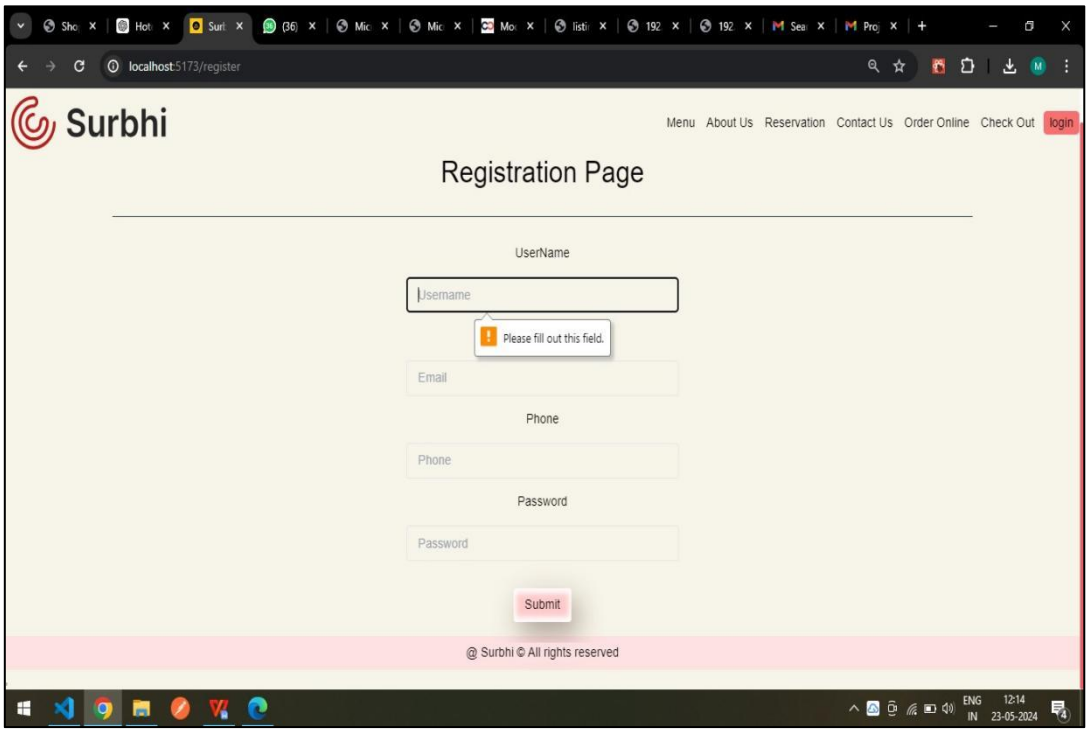


3.9 Sample Input and Output Screens:

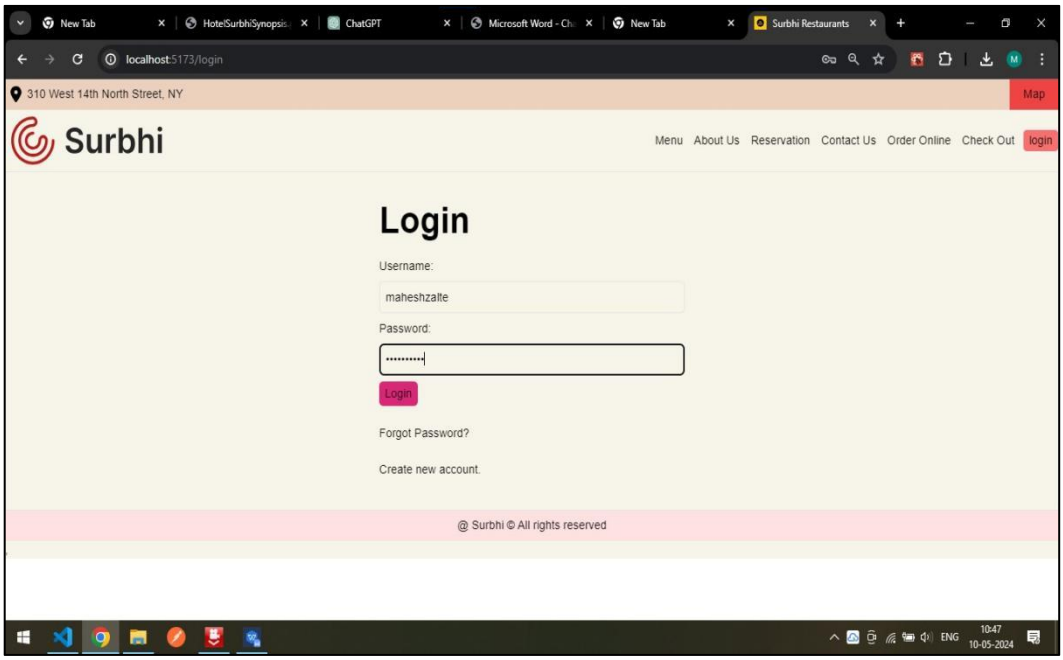
Registration Page: This page allows users to register to the platform using credentials



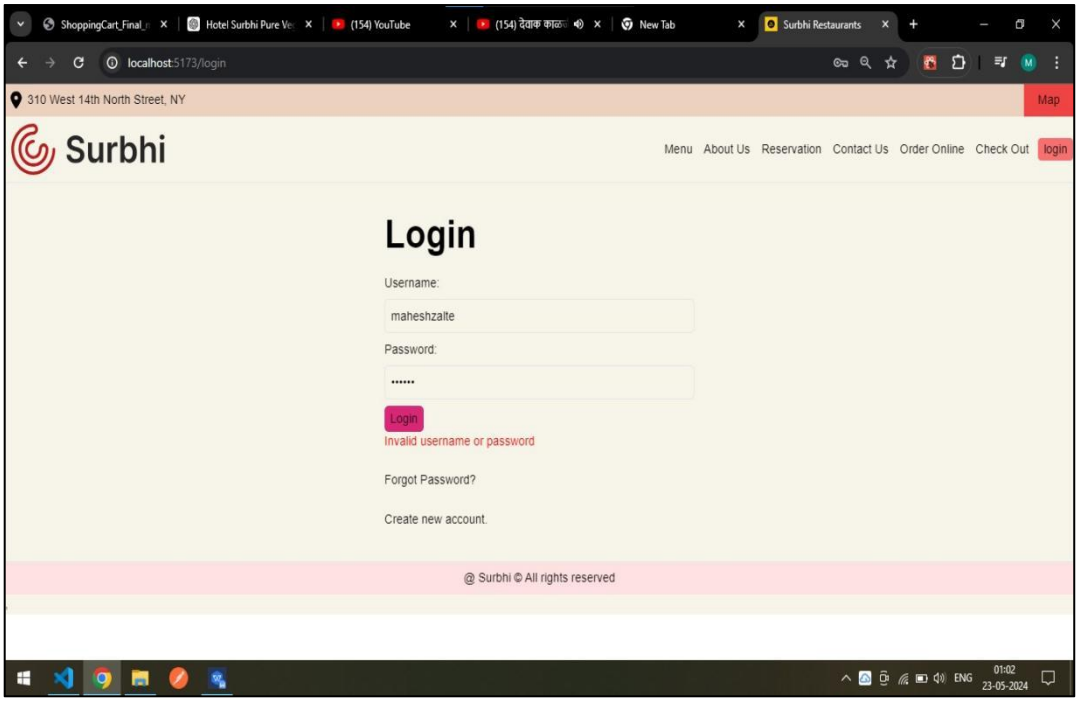
Validation:



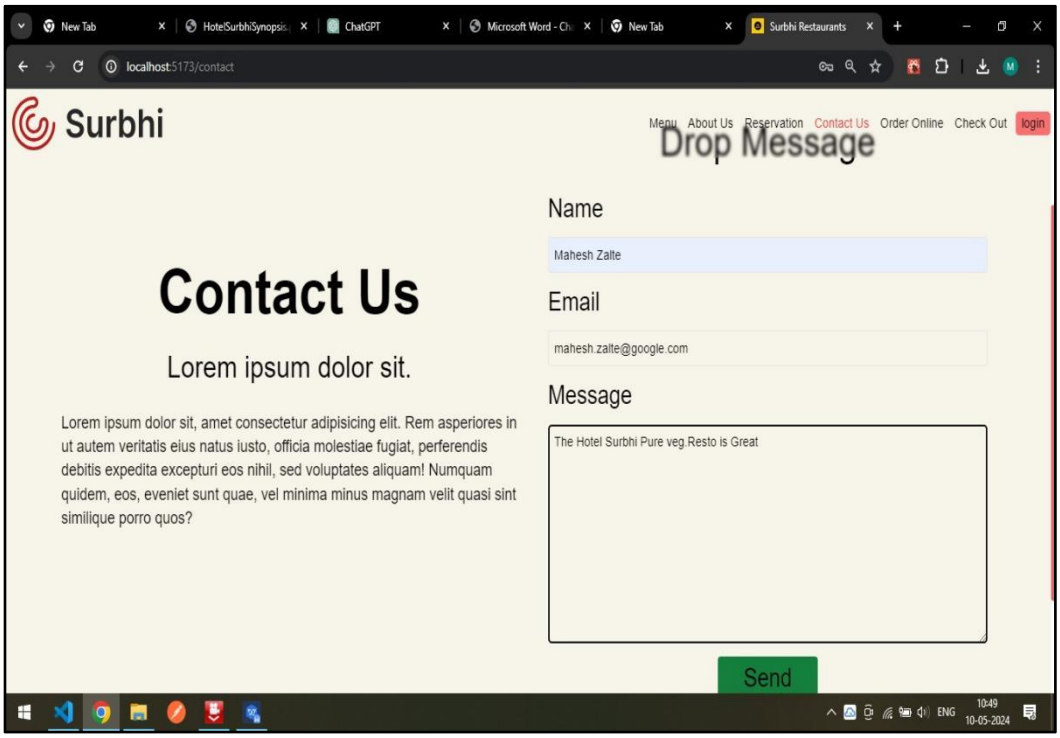
Login Page: This page allows users to login into the platform using credentials.



Validation:



Contact Us Form:



This is a Contact Us Page where the user is able to submit his feedback