RESTAURANT MANAGEMENT SYSTEM

Project Report Submitted By

PRIYA THOMAS

Reg. No: AJC17MCA-I045

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, **"RESTAURANT MANAGEMENT SYSTEM"** is the bonafide work of **PRIYA THOMAS (Reg. No: AJC17MCA-I045)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year **2017 – 2022**.

Ms. Sruthimol Kurian
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **“RESTAURANT MANAGEMENT SYSTEM”** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date:	**PRIYA THOMAS**
KANJIRAPPALLY	**Reg. No: AJC17MCA-I045**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Sruthimol Kurian** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

PRIYA THOMAS

ABSTRACT

The system is implemented to reduce the manual work and enhances the accuracy of work in a restaurant. This system manages and maintains the record of customers and their order online. This system has been made in a user-friendly interface. So that staff can add and delete the food items easily. Through the place ordering menu, the customer can simply click and order the food. The billing system prepares the bill according to the delivered food. The current system is paper based. Papers are used in restaurants for displaying the traditional menu cards, writing down the orders of customers, storing the records of customers.

Online Restaurant management system is the system for manage the restaurant business. After successful login the customer can access the menu page with the Items listed according to the desired tune. The main point of developing this system is to help restaurant administrator manage the restaurant business and help customer for online ordering and reserve table. In proposed system user can search for a menu according to his choice i.e., according to price range and category of food and later he can order a meal.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	19
4.2.3	COLLABORATION DIAGRAM	21
4.2.4	STATE CHART DIAGRAM	22
4.2.5	ACTIVITY DIAGRAM	23
4.2.6	CLASS DIAGRAM	24
4.2.7	OBJECT DIAGRAM	25
4.2.8	COMPONENT DIAGRAM	26
4.2.9	DEPLOYMENT DIAGRAM	27

4.3	**USER INTERFACE DESIGN USING FIGMA**	**28**
4.4	**DATA BASE DESIGN**	**32**
5	**SYSTEM TESTING**	**39**
5.1	**INTRODUCTION**	**40**
5.2	**TEST PLAN**	**41**
5.2.1	**UNIT TESTING**	**41**
5.2.1.1	**TEST CASE**	**42**
5.2.2	**INTEGRATION TESTING**	**43**
5.2.3	**VALIDATION TESTING**	**45**
5.2.4	**USER ACCEPTANCE TESTING**	**45**
6	**IMPLEMENTATION**	**46**
6.1	**INTRODUCTION**	**47**
6.2	**IMPLEMENTATION PROCEDURES**	**47**
6.2.1	**USER TRAINING**	**48**
6.2.2	**TRAINING ON THE APPLICATION SOFTWARE**	**48**
6.2.3	**SYSTEM MAINTENANCE**	**48**
7	**CONCLUSION AND FUTURE SCOPE**	**49**
7.1	**CONCLUSION**	**50**
7.2	**FUTURE SCOPE**	**50**
8	**BIBLIOGRAPHY**	**51**
9	**APPENDIX**	**53**
9.1	**SAMPLE CODE**	**54**
9.2	**SCREENSHOTS**	**59**

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The "Restaurant Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Restaurant Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Category, Food Item, Order, Payment, Confirm Order. Every Restaurant Management System has different Food Item needs; therefore, we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

1.2 PROJECT SPECIFICATION

The proposed system has the following requirements:

- System needs store information about new entry of Food Item.
- System needs to help the internal staff to keep information of Category and find them as per various queries.
- System needs to maintain quantity record.
- System needs to keep the record of Customer.
- System needs to update and delete the record.
- System also needs a search area.

- It also needs a security system to prevent data.

Main modules are:

1. **Login Module**

In login module the customer and restaurants login will be taken while they already registered on the application. Every manager/user will have login id and password to login to the application.

2. **Registration Module**

This module is displayed to the visitors if they need to perform some order placements, and new registration for restaurants.

3. **Menu Module**

This module is for admin. Admin have rights to insert, update (modify) and delete the data in database as per his/her necessary requirements.

4. **Order Module**

The activity is performed by customer itself whose registration is already done. Once the verification is done by application, the order gets confirmed and delivery will be given to the dedicated customers address.

5. **Reservation Module**

The Restaurant Booking System is a convenient self-service table booking system that can be embedded on any website. With our restaurant reservation system, you can streamline your booking process by enabling customers to book tables through your site 24/7. At the same time, admins can manage the reservations and restaurant availability.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

The current system is paper based. Papers are used in restaurants for displaying the traditional menu cards, writing down the orders of customers, storing the records of customers. The disadvantages of paper-based system are that papers can get easily damaged by stain marks; they can be lost due to fire or accidents or can get lost in general. Hence, time and money are wasted. As traditional menu cards are paper based, any changes that need to be made in the menu will require reprinting of the entire menu card, leading to wastage.

For small changes, reprinting the entire menu card is impossible. Changes in the menu card cannot be made dynamically. It is inefficient to access a particular record from the stack of papers. This system is time consuming. One has to call a waiter number of times till he notices it, and wait for him to arrive at their table to take their order. Also, the waiter can misinterpret the customer’s order since he is writing the order on paper, and the case of serving a wrong dish is possible. For placing any orders customers have to visit hotels or restaurants to know about food items and then place order and pay. In this method time and manual work is required. While placing an order over the phone, customer lacks the physical copy of the menu item, lack of visual confirmation that the order was placed correctly. Every restaurant needs certain employees to take the order over phone or in person, to offer a rich dining experience and process the payment. In today’s market, labor rates are increasing day by day making it difficult to find employees when needed.

2.3 DRAWBACKS OF EXISTING SYSTEM

- No proper online management of system
- Human effort is needed.
- It is difficult to maintain important information in books.
- More manual hours need to generate required reports.

2.4 PROPOSED SYSTEM

There are lots of advantages of having quality restaurant management software. What it does is basically decreases the amount of work you have to do usually. Most of the restaurants these days deal larger amounts of cash along with credit cards being swiped, not to forget online orders which make it quite difficult to keep up with everything. This is when restaurant management software becomes important.

In our app first customer will create his account after that select the item which he wants to order then select the payment gateway and then process through his order. On the other hand, there would be an administrator who would add delete the items from the menu list.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- ➢ Does the existing technology sufficient for the suggested one?
- ➢ Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- ➢ Is there sufficient support for the users?
- ➢ Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, youneed a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object- oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains fourcomponents.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their

roles.

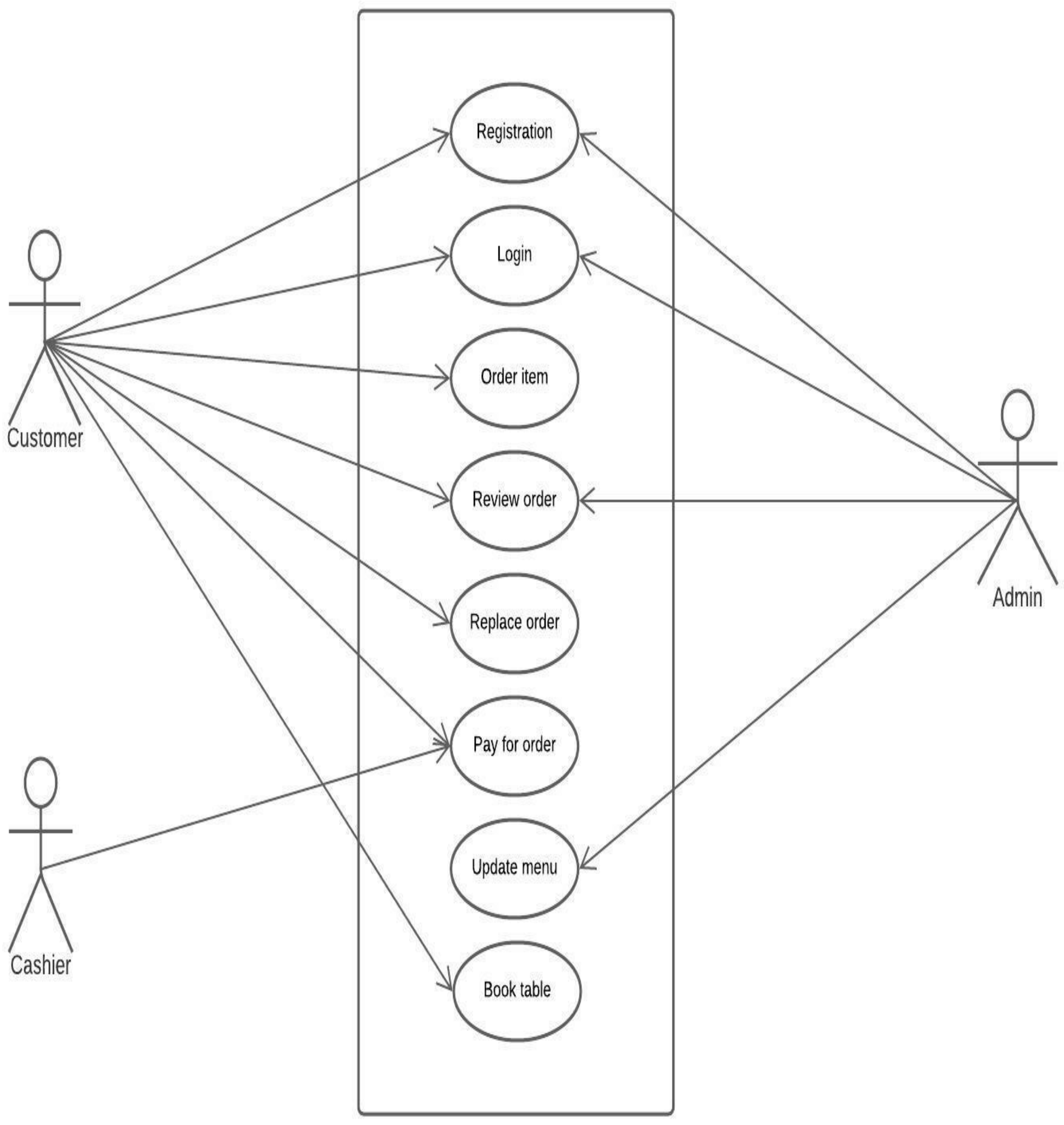
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig. 1: Use case diagram for Restaurant Management System

USE CASE DIAGRAM



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

- Lost Message

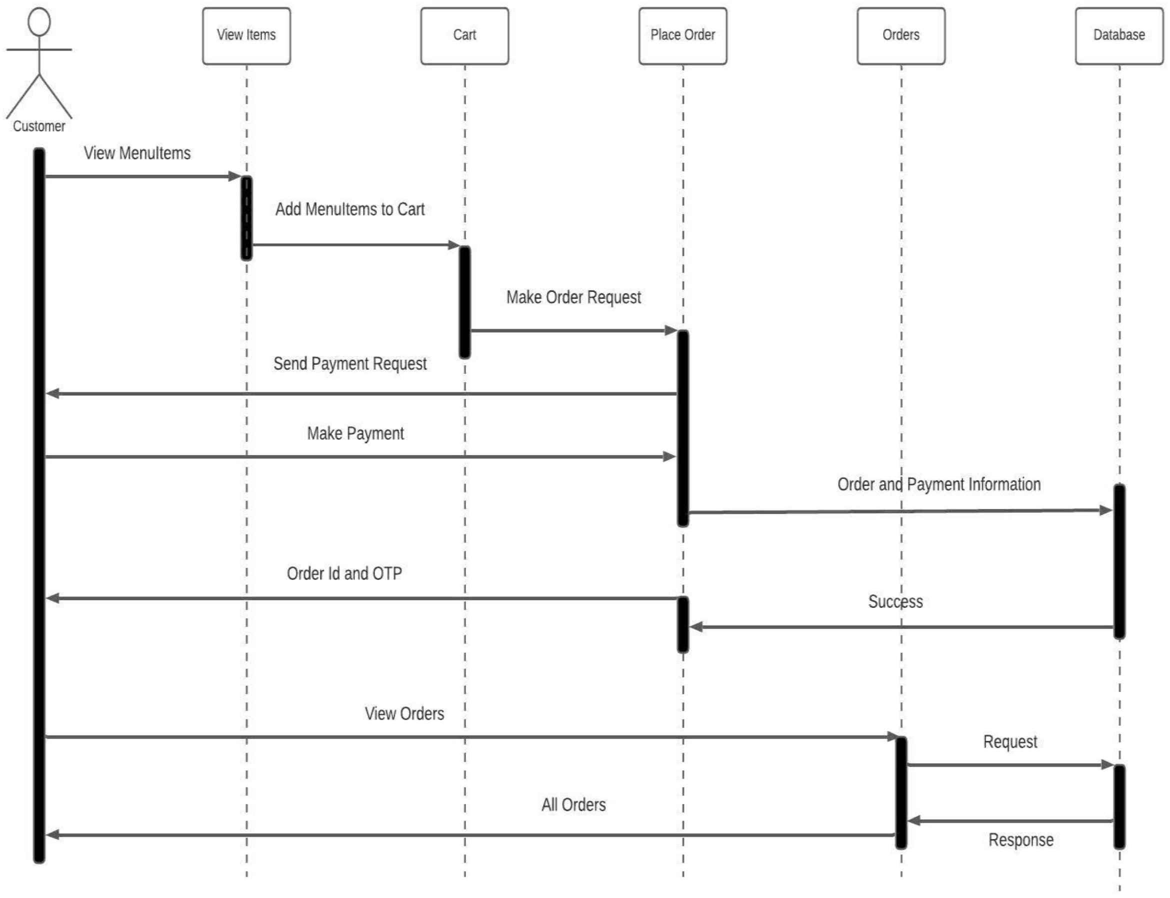
iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Fig 2: Sequence diagram for Restaurant Management System

Sequence Diagram

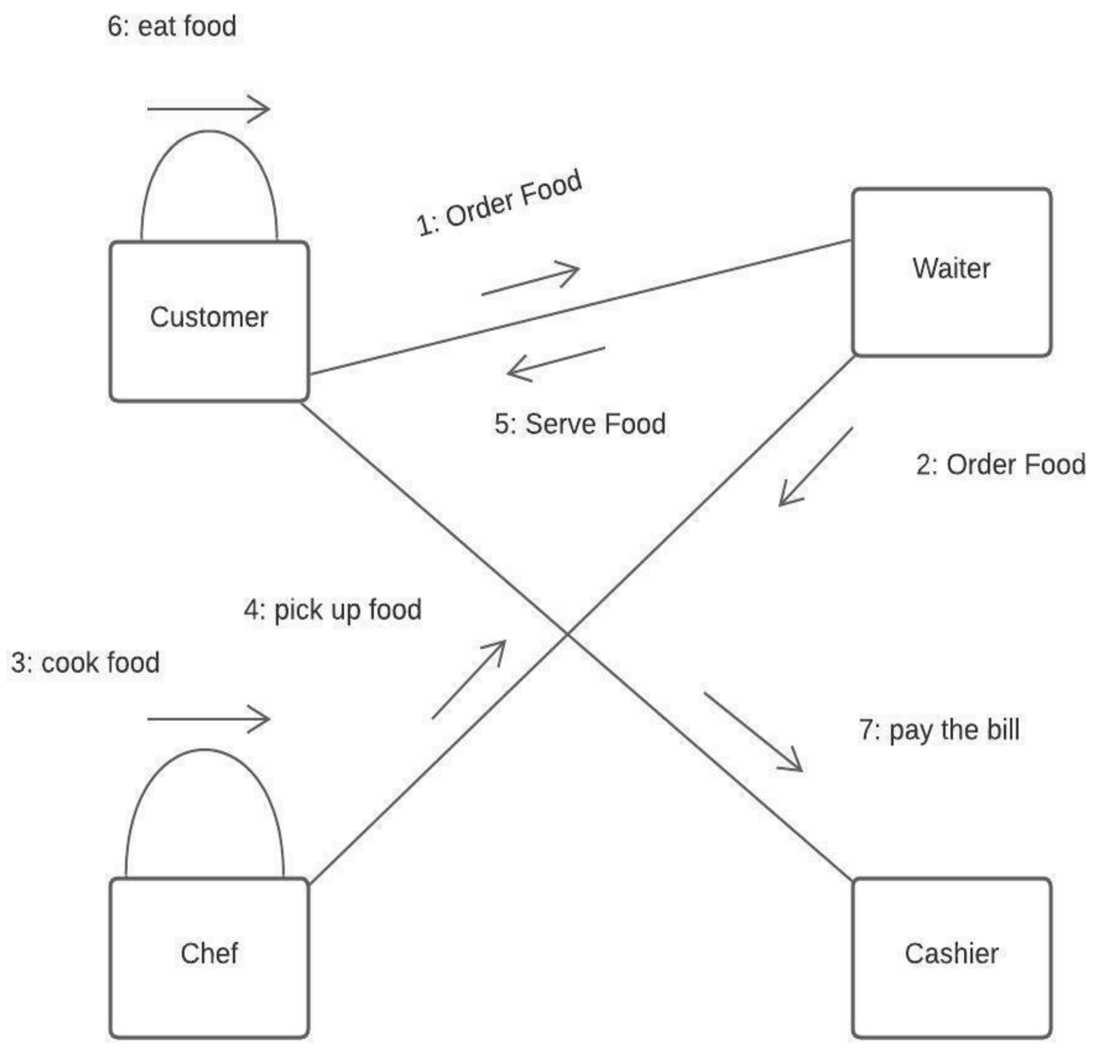


4.2.3 COLLABORATION DIAGRAM

Collaboration diagrams (known as Communication Diagram in UML 2.x) are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration is used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

Fig. 3: Collaboration diagram for Restaurant Management System

Collaboration Diagram



4.2.4 STATE CHART DIAGRAM

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

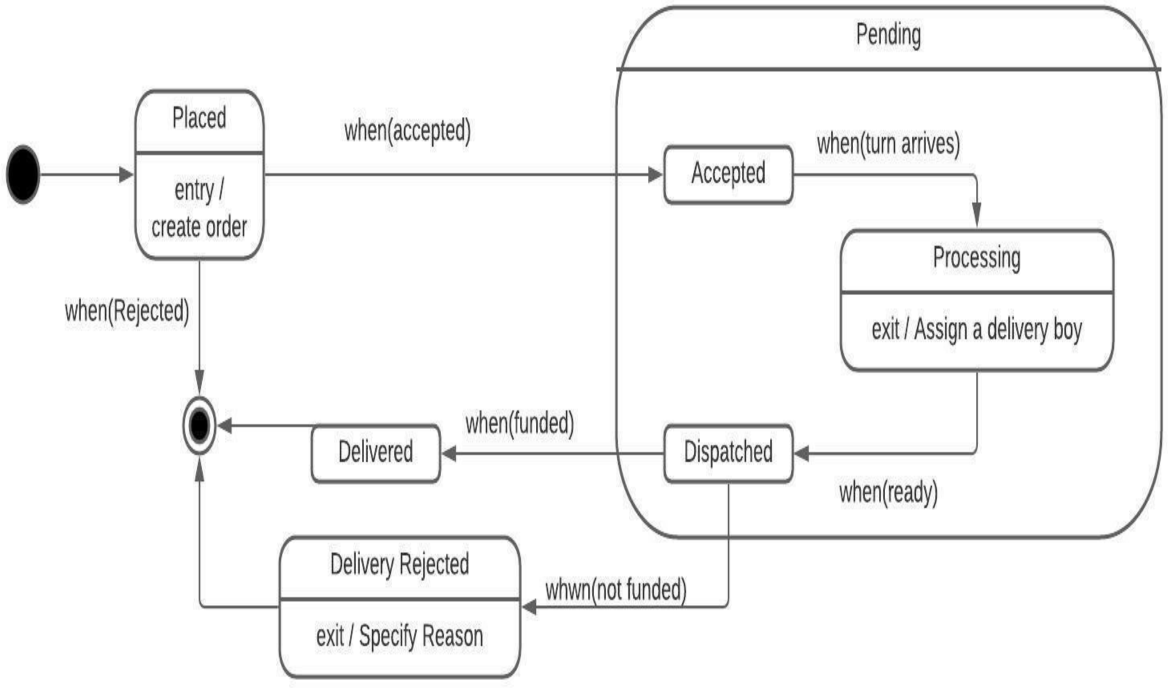
State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams −

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

Fig. 4: State chart diagram for Restaurant Management System

State Chart Diagram



4.2.5 ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

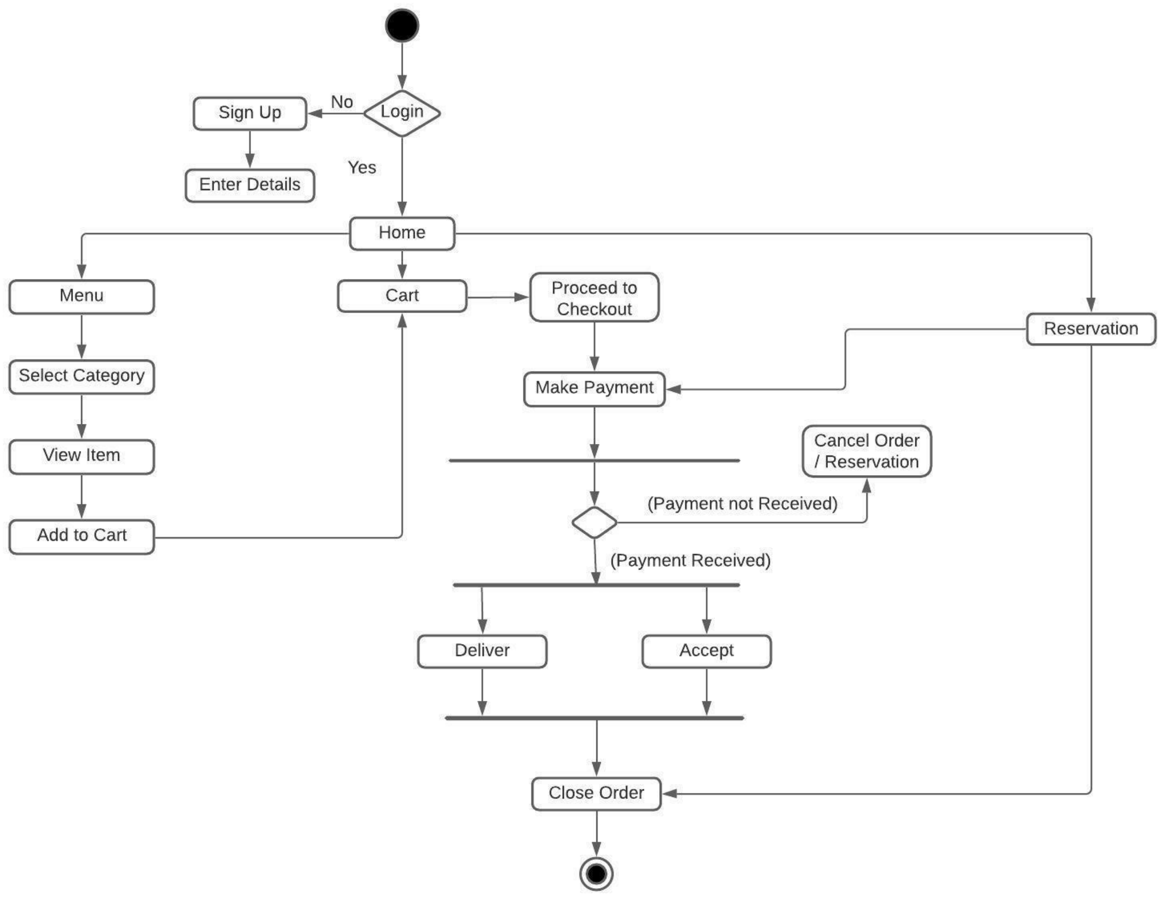
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Fig. 5: Activity diagram for Restaurant Management System

Activity Diagram



4.2.6 CLASS DIAGRAM

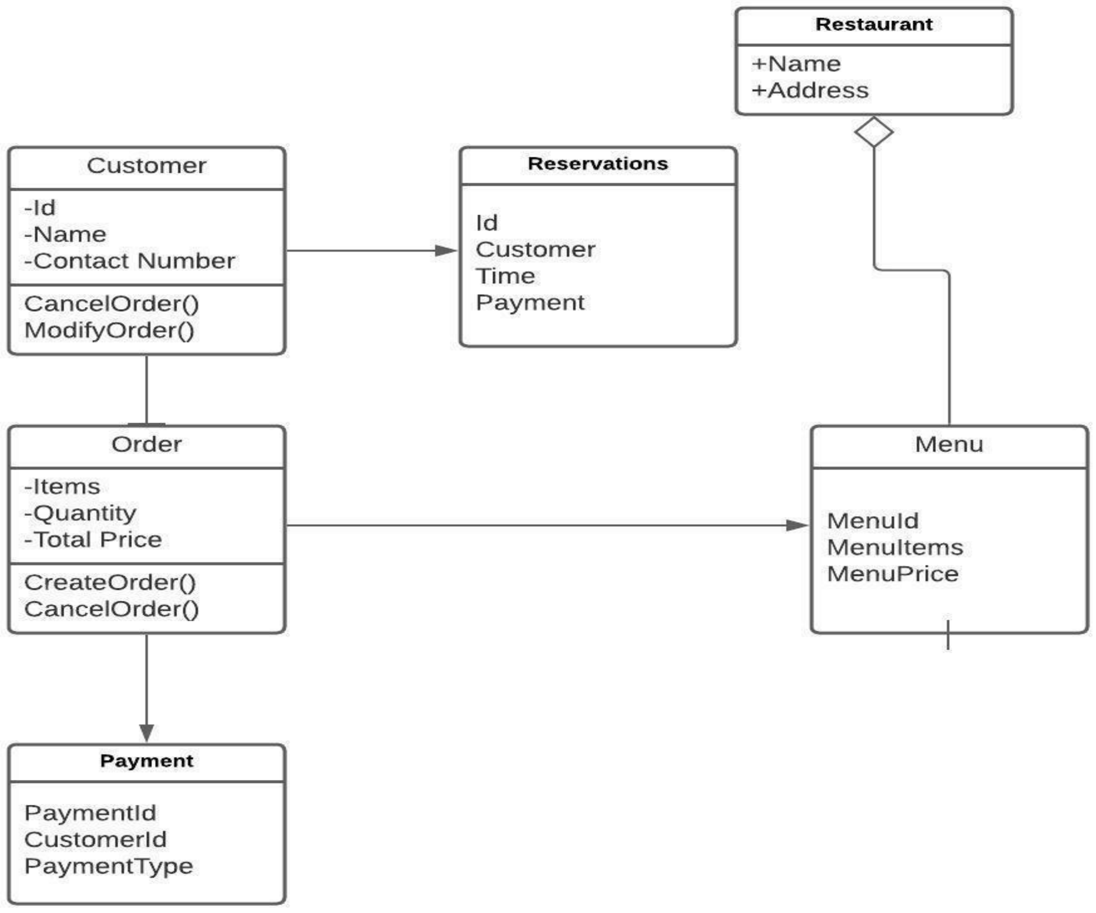
Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Fig. 6: Class diagram for Restaurant Management System

Class Diagram



4.2.7 OBJECT DIAGRAM

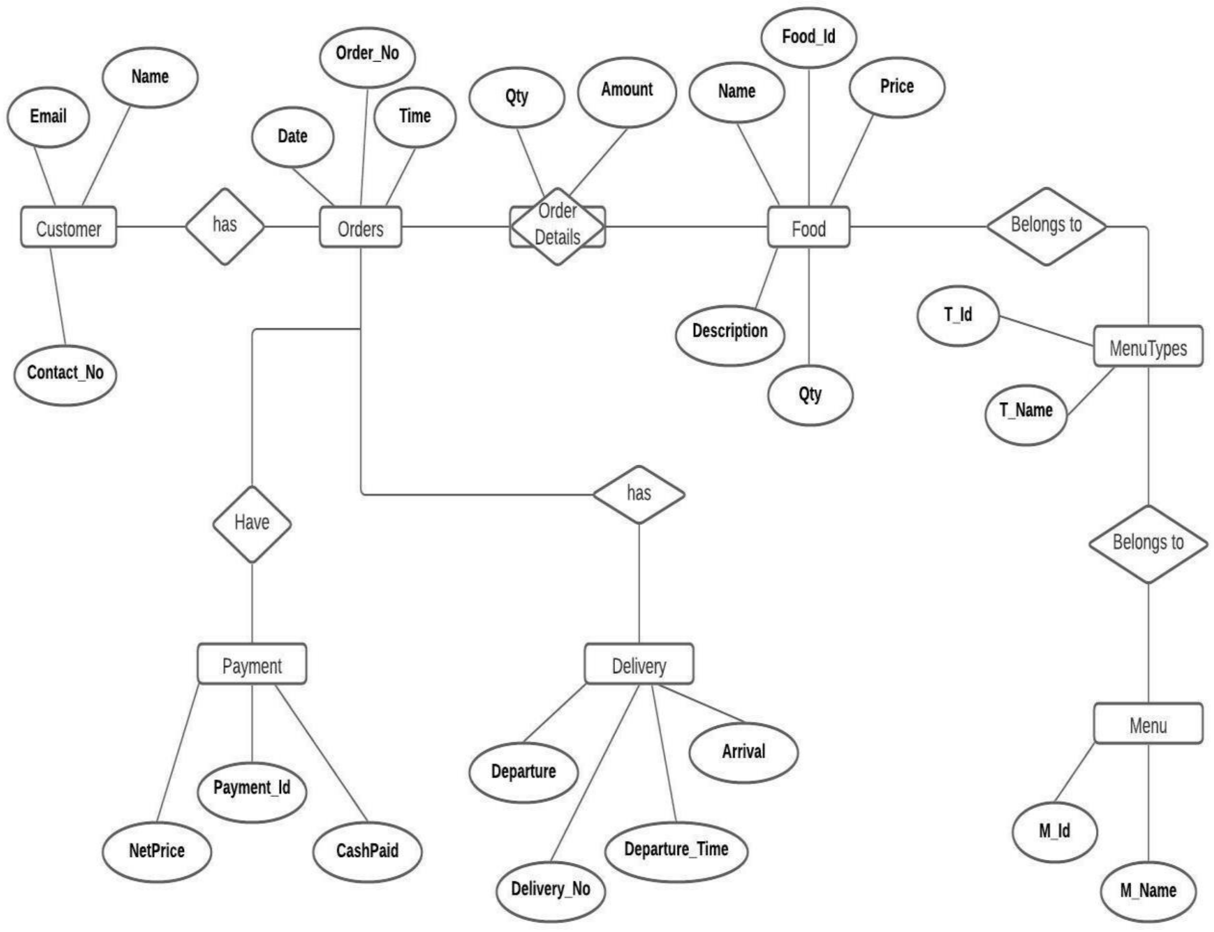
Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

Fig. 7: Object diagram for Restaurant Management System

Object Diagram



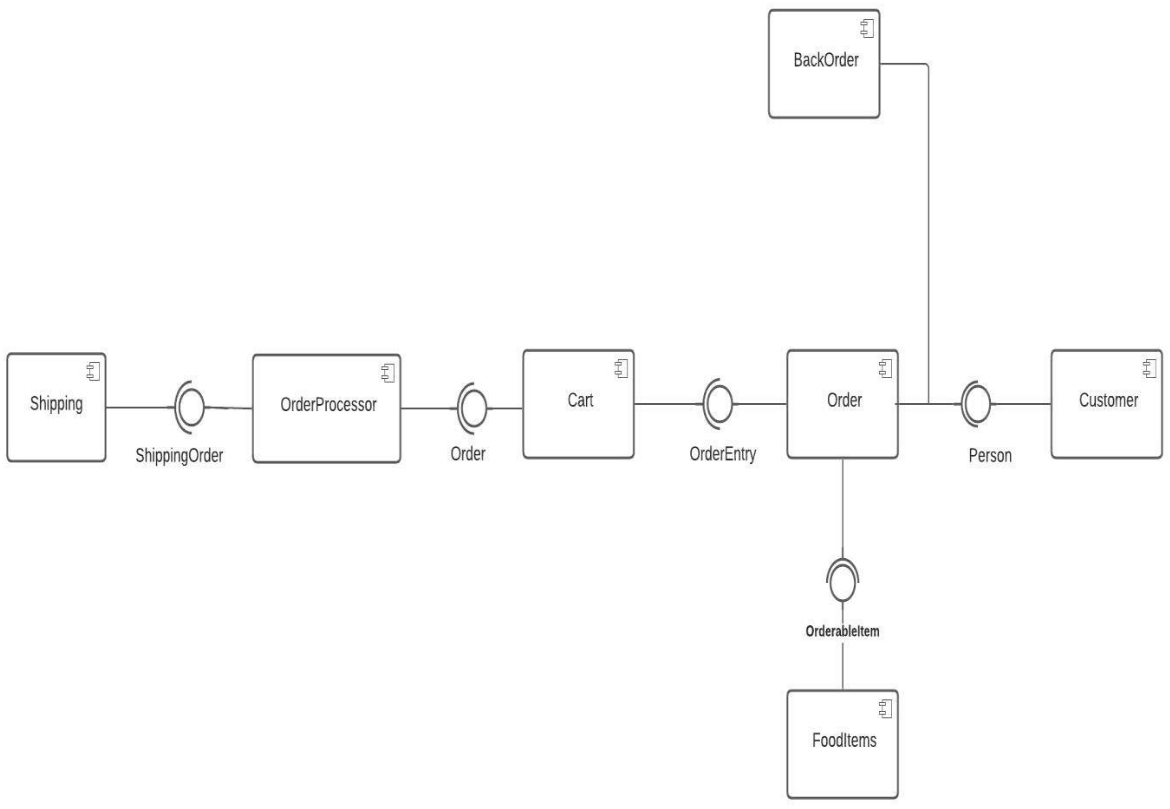
4.2.8 COMPONENT DIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

Fig. 8: Component diagram for Restaurant Management System

Component Diagram



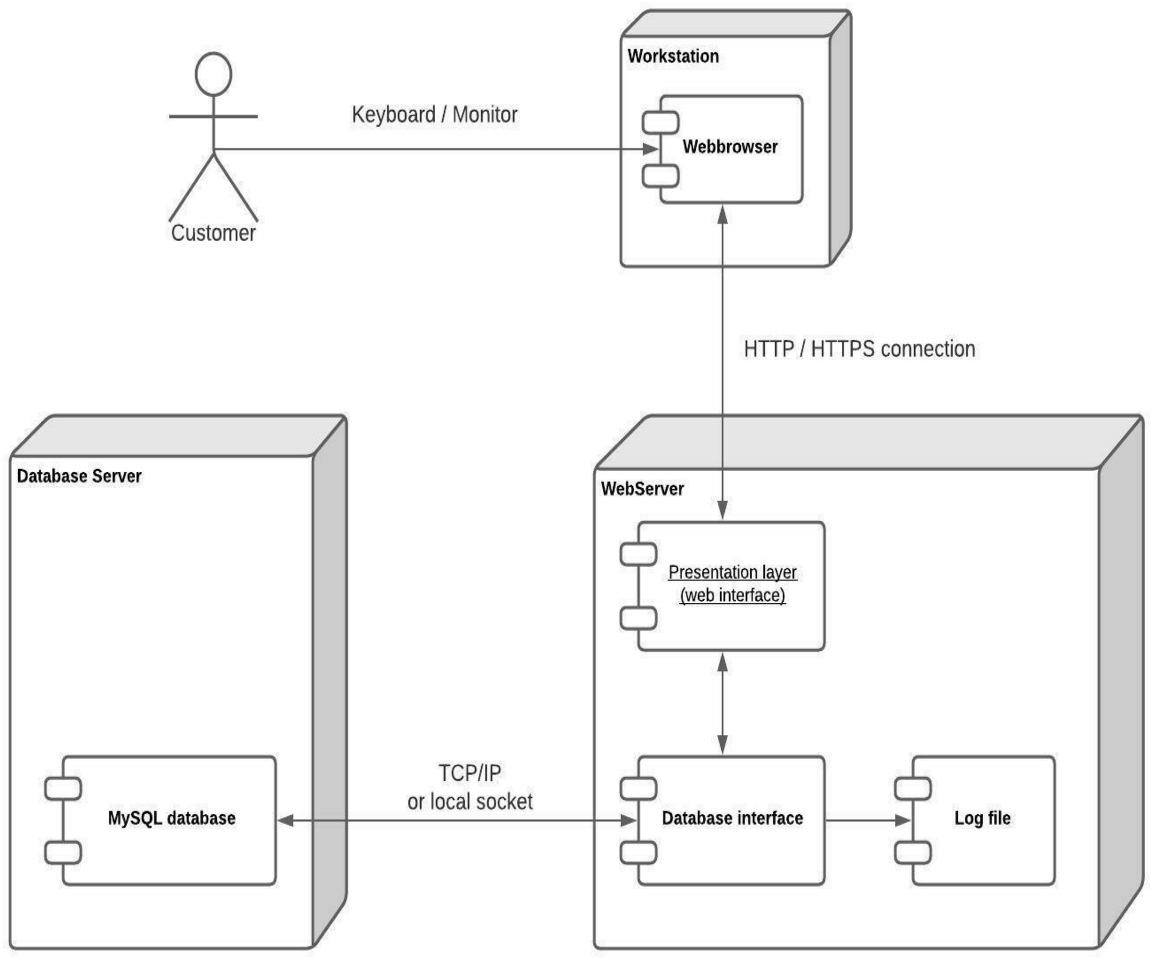
4.2.9 DEPLOYMENT DIAGRAM

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

Fig. 9: Deployment diagram for Restaurant Management System

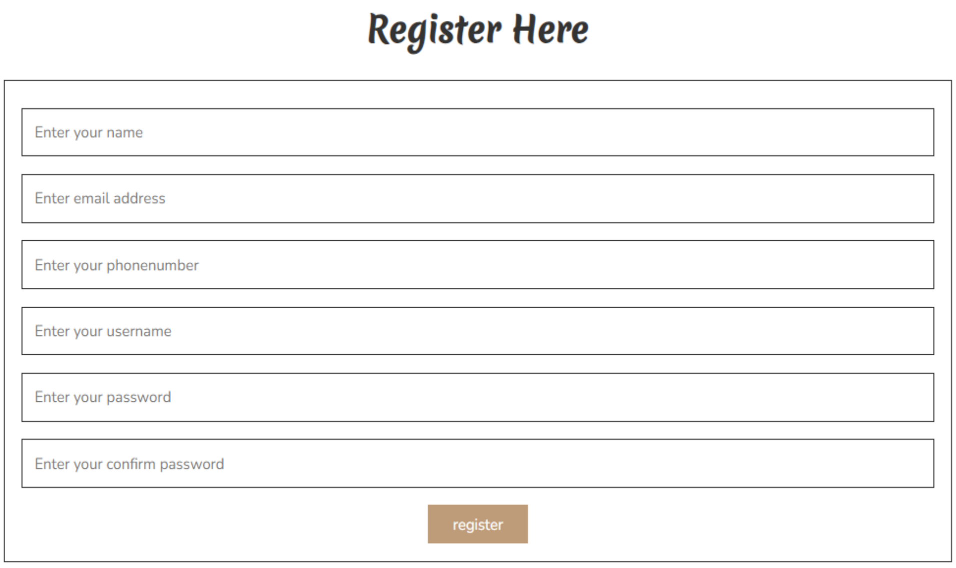
Deployment Diagram



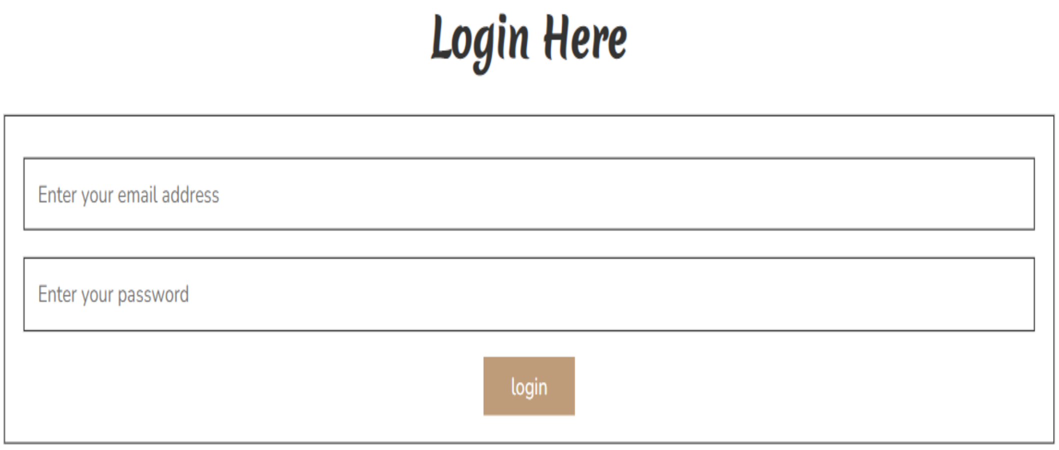
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1 - INPUT DESIGN

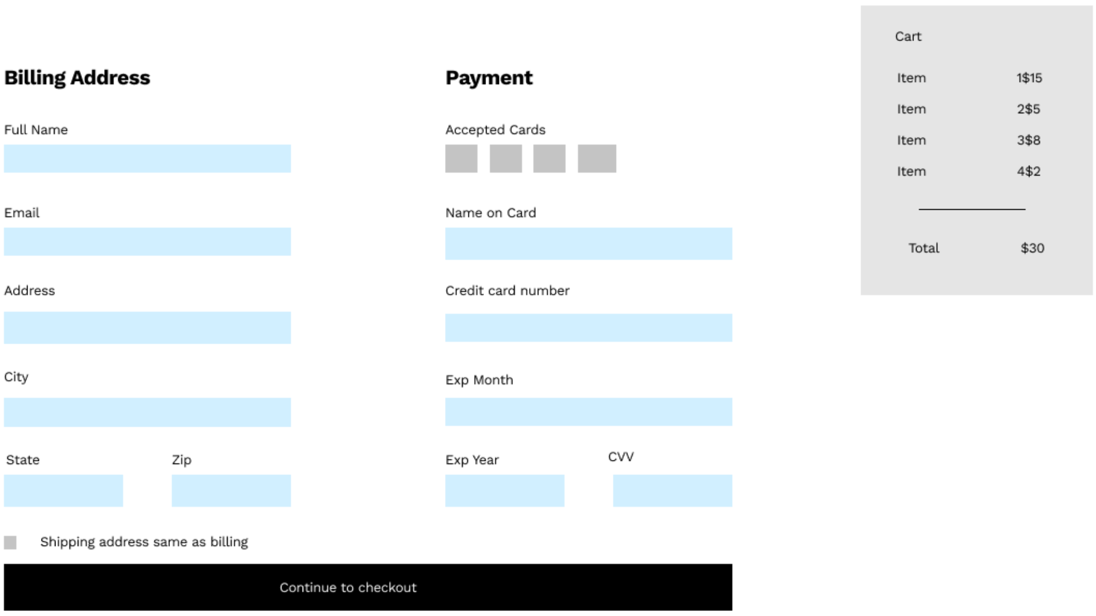
Form Name : Customer Registration



Form Name : User Login



Form Name : Checkout



Cart

Item	1$15
Item	2$5
Item	3$8
Item	4$2
Total	$30

Billing Address

Full Name

Email

Address

City

State Zip

Shipping address same as billing

Payment

Accepted Cards

Name on Card

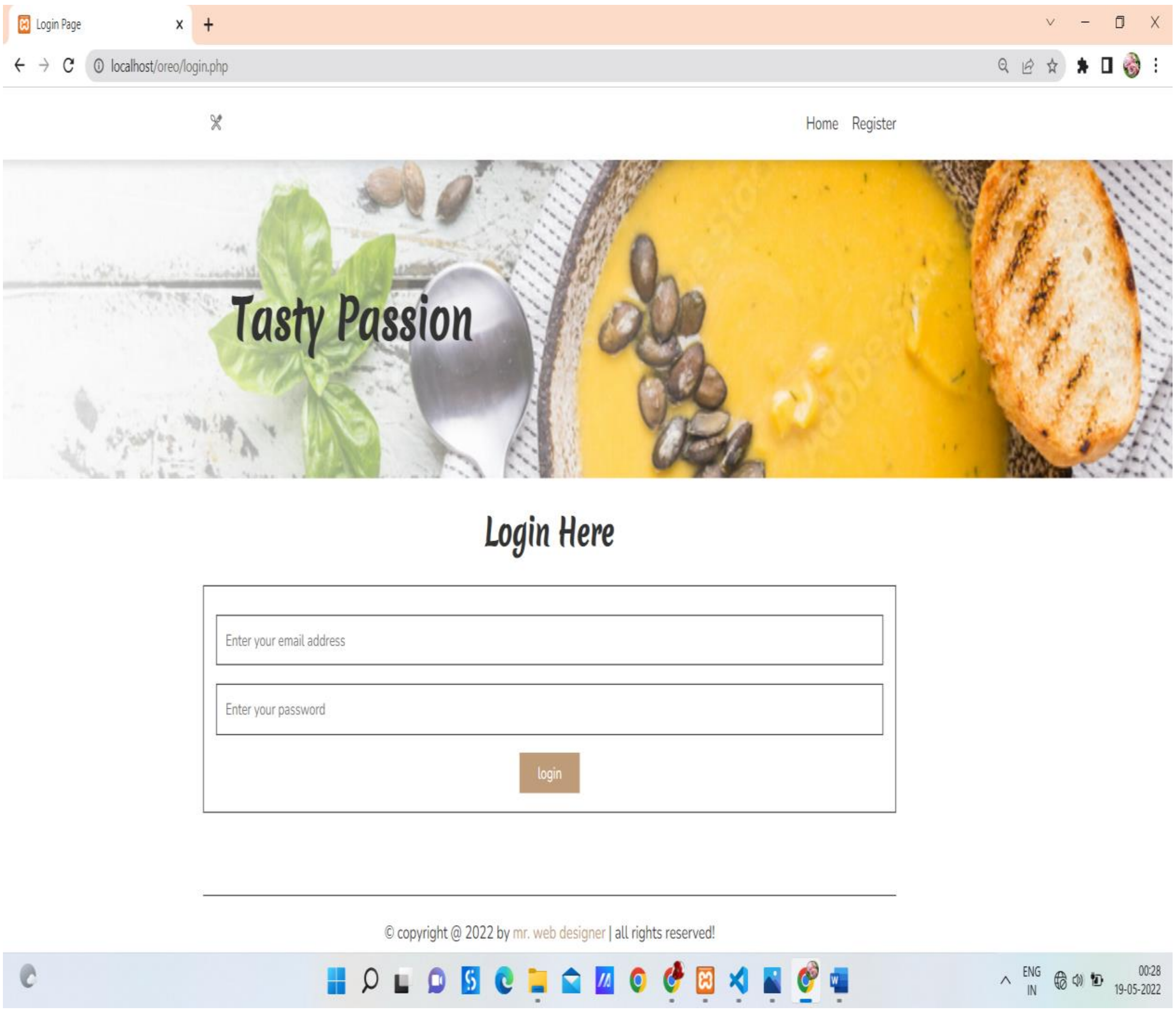
Credit card number

Exp Month

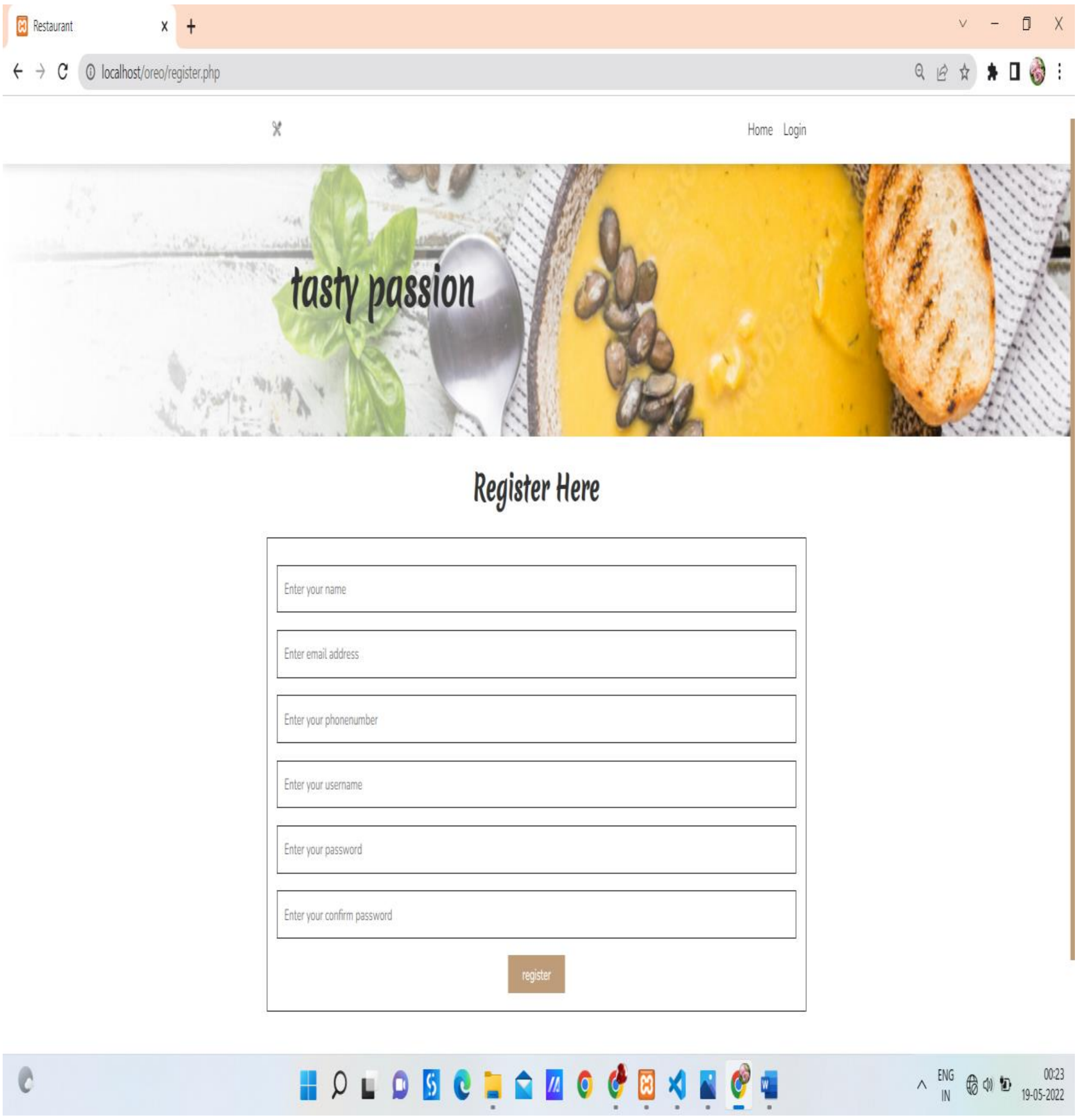
Exp Year CVV

4.3.2 OUTPUT DESIGN

User Login



Customer Registration



4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key is Super Key and Candidate Keys.

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on another non-key attribute.

TABLE DESIGN

Table No 01
Table Name : tbl_register

Primary Key : user_id

Table Description: To store user Registration information

Fieldname	Datatype	Constraints	Description
user_id	Int(20)	Primary key	Register id
name	Varchar(100)	NOT NULL	Customer's name
email	Varchar(100)	NOT NULL	Customer's email address
phone	Varchar(50)	NOT NULL	Customer's phone number
username	Varchar(100)	NOT NULL	Customer's username
password	Varchar(150)	NOT NULL	Password

Table No 02

Table Name : tbl_admin

Primary Key : a_id

Table Description: To store admin information

Fieldname	Datatype	Constraints	Description
a_id	Int(20)	Primary key	Admin id
name	Varchar(100)	NOT NULL	Admin name
email	Varchar(100)	NOT NULL	Admin email address

phone	Varchar(100)	NOT NULL	Phone number of admin
username	Varchar(100)	NOT NULL	Admin’s username
password	Varchar(100)	NOT NULL	Admin’s password for login

Table No 03

Table Name : Products

Primary key : id

Table description : To store food item’s information

Fieldname	Datatype	Constraints	Description
id	Int(100)	Primary Key	Food items id
name	Varchar(100)	NOT NULL	Food items name.
category	Varchar(20)	NOT NULL	Food items category name.
details	Varchar(500)	NOT NULL	Food items details.
Price	Int(100)	NOT NULL	Food items price.
Image	Varchar(100)	NOT NULL	Food image

Table No 04

Table Name : orders

Primary Key : id

Foreign Key : username

Table Description: To store order information

Fieldname	Datatype	Constraints	Description
id	Int(100)	Primary key	Order id
username	Varchar(100)	Foreign Key	Customer’s username
name	Varchar(100)	NOT NULL	Customer’s name who process the order
number	Varchar(12)	NOT NULL	Customer’s phone number
email	Varchar(100)	NOT NULL	Customer’s email address
address	Varchar(500)	NOT NULL	Customer’s location address
total_products	Varchar(1000)	NOT NULL	Total number of products ordered
total_price	Int(100)	NOT NULL	Total amount
placed_on	Varchar(50)	NOT NULL	Date of ordering
payment_status	Varchar(20)	NOT NULL	Amount paid or not
payment_id	Varchar(50)	NOT NULL	Payment id

Table 05

Table Name : cart

Primary Key : id

Table Description: To store cart details

Fieldname	Datatype	Constraints	Description
id	Int(100)	Primary key	Id of order details
username	Varchar(100)	Foreign Key	Foreign key of table tbl_register
pid	Int(100)	Foreign Key	Foreign key of table products
name	Varchar(100)	NOT NULL	Name of the product
price	Int(100)	NOT NULL	Price of the product
quantity	Int(100)	NOT NULL	Quantity of the selected products
image	Varchar(100)	NOT NULL	Image of the selected products

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers are always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.1.1. Test Case

Test Case

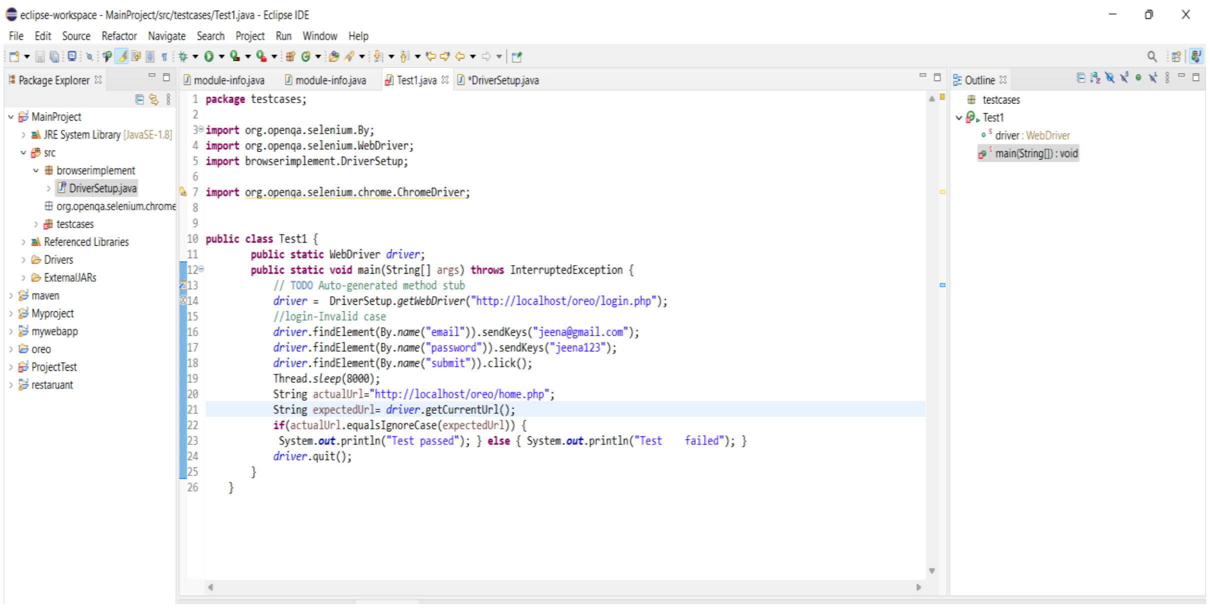
Project Name: Restaurant Management System					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By: Priya Thomas**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 19-05-2022		
Module Name: Login Screen			**Test Executed By: Ms. Sruthimol Kurian**		
Test Title: Verify login with validemail and password			**Test Execution Date:** 20-05-2022		
Description: Test the Login Page					
Pre-Condition: User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: jeena@gmail.com	User should beable to Login	User Logged in and navigated to Sub-admin Dashboard with records	Pass
3	Provide Valid Password	Password: jeena123			
4	Click on Sign In				

	button				
5	Provide Invalid Email Id or password	Email Id: userA@gmail.Com Password: User12345	User shouldnot be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

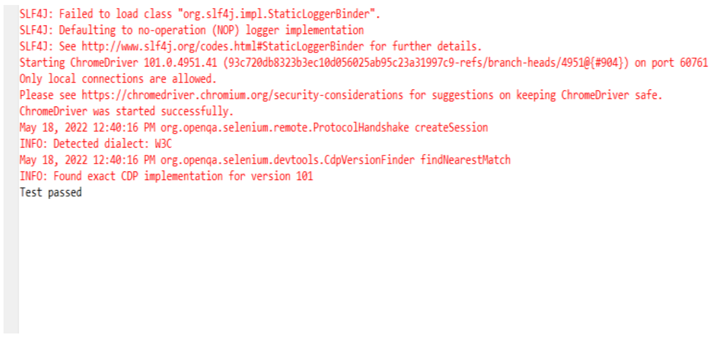
Code

```java
package testcases;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplement.DriverSetup;

import org.openqa.selenium.chrome.ChromeDriver;


public class Test1 {
            public static WebDriver driver;
            public static void main(String[] args) throws InterruptedException {
                  // TODO Auto-generated method stub
                  driver =
DriverSetup.getWebDriver("http://localhost/oreo/login.php");
                  //login-Invalid case
                  driver.findElement(By.name("email")).sendKeys("jeena@gmail.com");
                  driver.findElement(By.name("password")).sendKeys("jeena123");
                  driver.findElement(By.name("submit")).click();
                  Thread.sleep(8000);
                  String actualUrl="http://localhost/oreo/home.php";
                  String expectedUrl= driver.getCurrentUrl();
                  if(actualUrl.equalsIgnoreCase(expectedUrl)) {
                   System.out.println("Test passed"); } else {
System.out.println("Test    failed"); }
                  driver.quit();
            }

      }
```

Output



```java
package testcases;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplement.DriverSetup;

import org.openqa.selenium.chrome.ChromeDriver;


public class Test1 {
        public static WebDriver driver;
        public static void main(String[] args) throws InterruptedException {
            // TODO Auto-generated method stub
            driver =  DriverSetup.getWebDriver("http://localhost/oreo/login.php");
            //login-Invalid case
            driver.findElement(By.name("email")).sendKeys("jeena@gmail.com");
            driver.findElement(By.name("password")).sendKeys("jeena123");
            driver.findElement(By.name("submit")).click();
            Thread.sleep(8000);
            String actualUrl="http://localhost/oreo/home.php";
            String expectedUrl= driver.getCurrentUrl();
            if(actualUrl.equalsIgnoreCase(expectedUrl)) {
             System.out.println("Test passed"); } else { System.out.println("Test    failed"); }
            driver.quit();
        }
    }
```



```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions thatwill fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4. Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to upload projects by viewing profile of contractors. Provides lots of advantages like search contractor, view profile of contractors, enhanced user interface, payment options, add feedback, daily progress report option, complaint status and may more.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode.
- Customers can able to do advanced search options
- Customers can able to add complaints and feedbacks etc.
- Contractors can able to view project details and add daily progress report etc.
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- Pankaj Jalote, "So*ftware engineering*: a precise approach", 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- https://www.w3schools.com/
- www.jquery.com
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Cart.php

```
<?php
include 'db_connect.php';
session_start();

$user_id = $_SESSION['username'];
if(!isset($user_id)){
  header("Location: login.php");
};
if(isset($_GET['delete'])){
   $delete_id = $_GET['delete'];
   $delete_cart_item = mysqli_query($con, "DELETE FROM `cart` WHERE id =
'$delete_id'");
   header('location:cart.php');
 }

 if(isset($_GET['delete_all'])){
   $delete_cart_item = mysqli_query($con, "DELETE FROM `cart` WHERE username =
'$user_id'");
   header('location:cart.php');
 }

 if(isset($_POST['update_qty'])){
   $cart_id = $_POST['cart_id'];
   $p_qty = $_POST['p_qty'];
   $update_qty = mysqli_query($con, "UPDATE `cart` SET quantity = $p_qty WHERE id
= '$cart_id'");
   $message[] = 'cart quantity updated';
 }
 ?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Restaurant</title>

   <!-- font awesome cdn link  -->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

   <!-- custom css file link  -->
   <link rel="stylesheet" href="css/cart.css">

</head>
<body>

<!-- header section starts  -->
<header class="header">
   <div class="flex">
      <a href="#home" class="logo"><img src="images/logo.jpg" alt=""></a>
      <nav class="navbar">
         <a href="home.php">home</a>
         <a href="#about">about</a>
         <a href="menu.php">menu</a>
         <a href="logout.php">Logout</a>
      </nav>

      <div class="icons">
         <div id="menu-btn" class="fas fa-bars"></div>
         <div id="user-btn" class="fas fa-user"></div>
         <a href="search_page.php" class="fas fa-search"></a>
```

```
        <a href="cart.php"><i class="fas fa-shopping-cart"></i></a>
      </div>

      <div class="profile">
        <a href="user_update_profile.php" class="btn">update profile</a>
        <a href="logout.php" class="delete-btn">logout</a>
        <div class="flex-btn">
          <a href="login.php" class="option-btn">login</a>
          <a href="register.php" class="option-btn">register</a>
        </div>
      </div>

        <?php echo "<h2>".$_SESSION['username']."</h2>"; ?>

    </div>
  </header>

  <!-- header section ends -->

  <!-- home section starts  -->

  <div class="home-bg">
    <section class="home" id="home">
    </section>
  </div>

  <!-- home section ends -->

  <!-- Latest product section starts -->
  <section class="shopping-cart">
    <h1 class="title">CART</h1>
     <div class="box-container">
        <?php
```

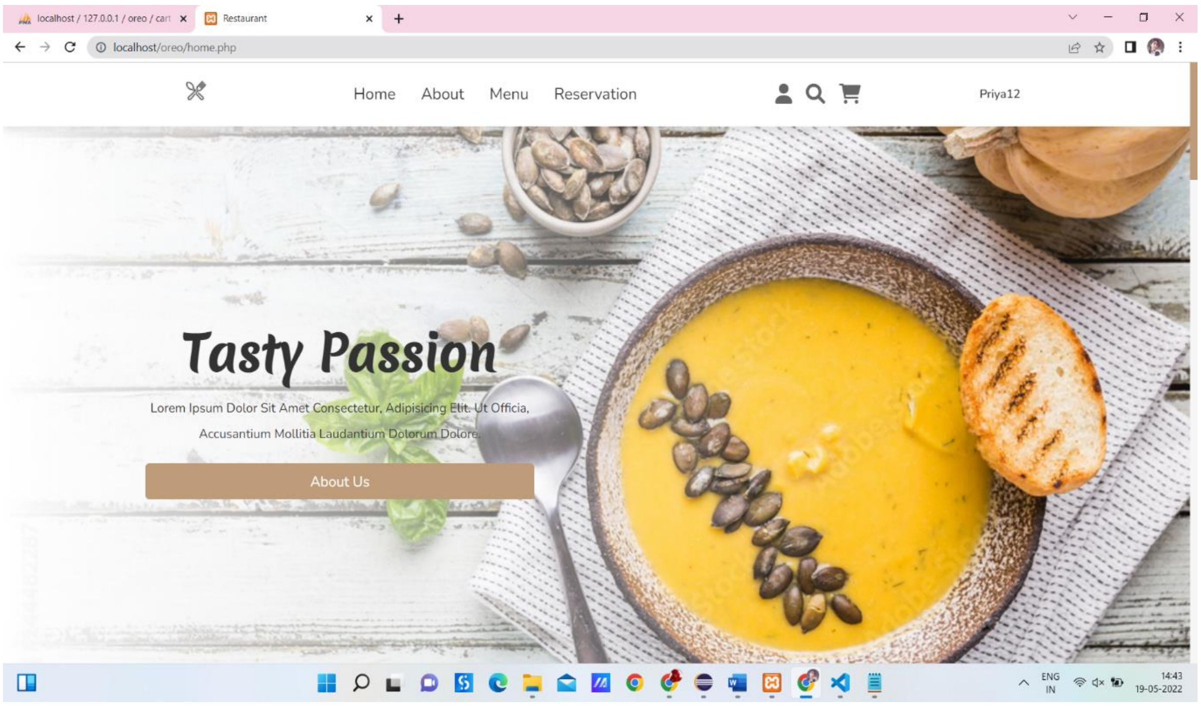
```
            $grand_total = 0;
            $select_cart = mysqli_query($con, "SELECT * FROM `cart` WHERE username =
'$user_id'");
            $check = mysqli_num_rows($select_cart) > 0;
            if($check){
        ?>
        <?php
            while($row=mysqli_fetch_array($select_cart))
            {
        ?>

        <form action="" method="POST" class="box">
            <a href="cart.php?delete=<?php echo $row['id']; ?>" class="fas fa-times"
onclick="return confirm('delete this from cart?');"></a>
            <img src="images/<?php echo $row['image']; ?>" alt="">
            <div class="name"><?php echo $row['name']; ?></div>
            <div class="price">Rs.<?php echo $row['price']; ?>/-</div>
            <input type="hidden" name="cart_id" value="<?php echo $row['id']; ?>">
            <div class="flex-btn">
               <input type="number" min="1" value="<?php echo $row['quantity']; ?>"
class="qty" name="p_qty">
               <input type="submit" value="update" name="update_qty" class="option-btn">
            </div>
            <div class="sub-total"> sub total : <span>Rs.<?php echo $sub_total = ($row['price']
* $row['quantity']); ?>/-</span> </div>
        </form>
        <?php
            $grand_total += $sub_total;
            }
        }else{
        echo '<p class="empty">your cart is empty</p>';
        }
        ?>
    </div>
```

```
     <div class="cart-total">
      <p>grand total : <span>Rs.<?php echo $grand_total; ?>/-</span></p>
      <a href="menu.php" class="option-btn">continue shopping</a>
      <a href="cart.php?delete_all" class="delete-btn <?php echo ($grand_total >
 1)?'':'disabled'; ?>">delete all</a>
      <a href="checkout.php" class="btn <?php echo ($grand_total > 1)?'':'disabled';
 ?>">proceed to checkout</a>
   </div>
 </section>

 <!-- Latest product section starts -->

 <!-- footer section starts  -->
 <section class="footer">
   <div class="credit"> &copy; copyright @ <?= date('Y'); ?> by <span>mr. web
 designer</span> | all rights reserved! </div>
 </section>

 <!-- footer section ends -->

 <!-- custom js file link  -->
 <script src="js/script2.js"></script>
</body>
</html>
```

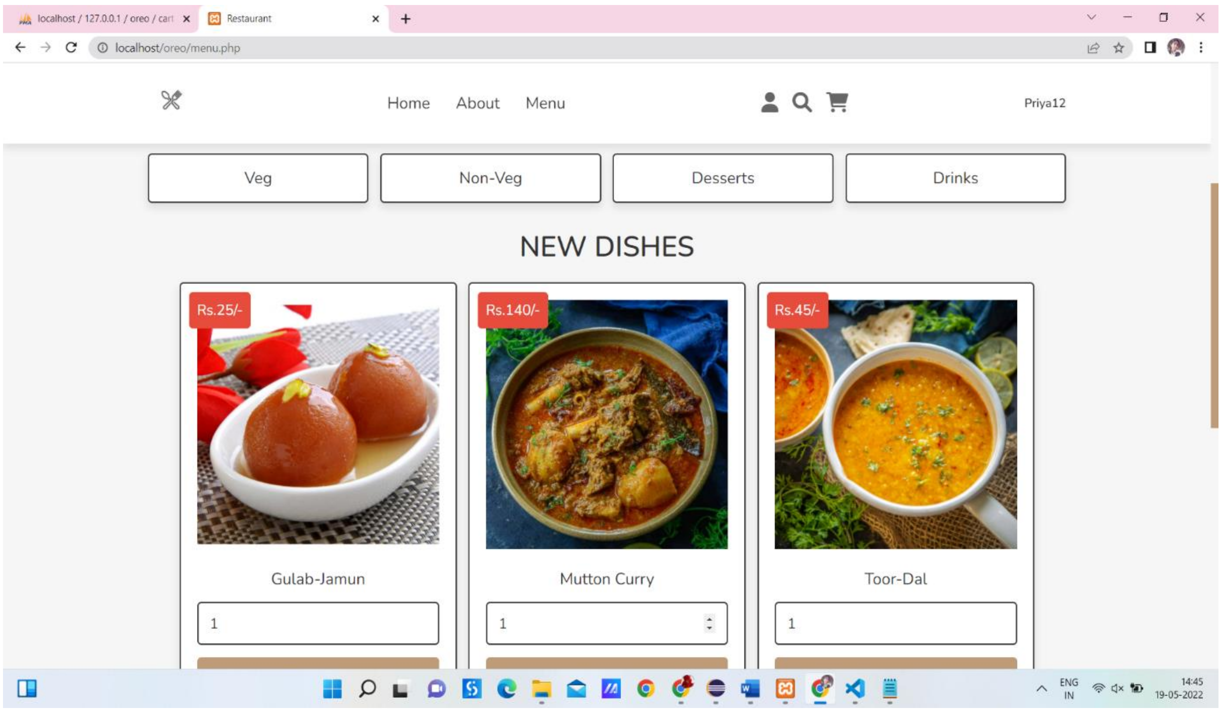
9.2 Screen Shots

CUSTOMER PAGES

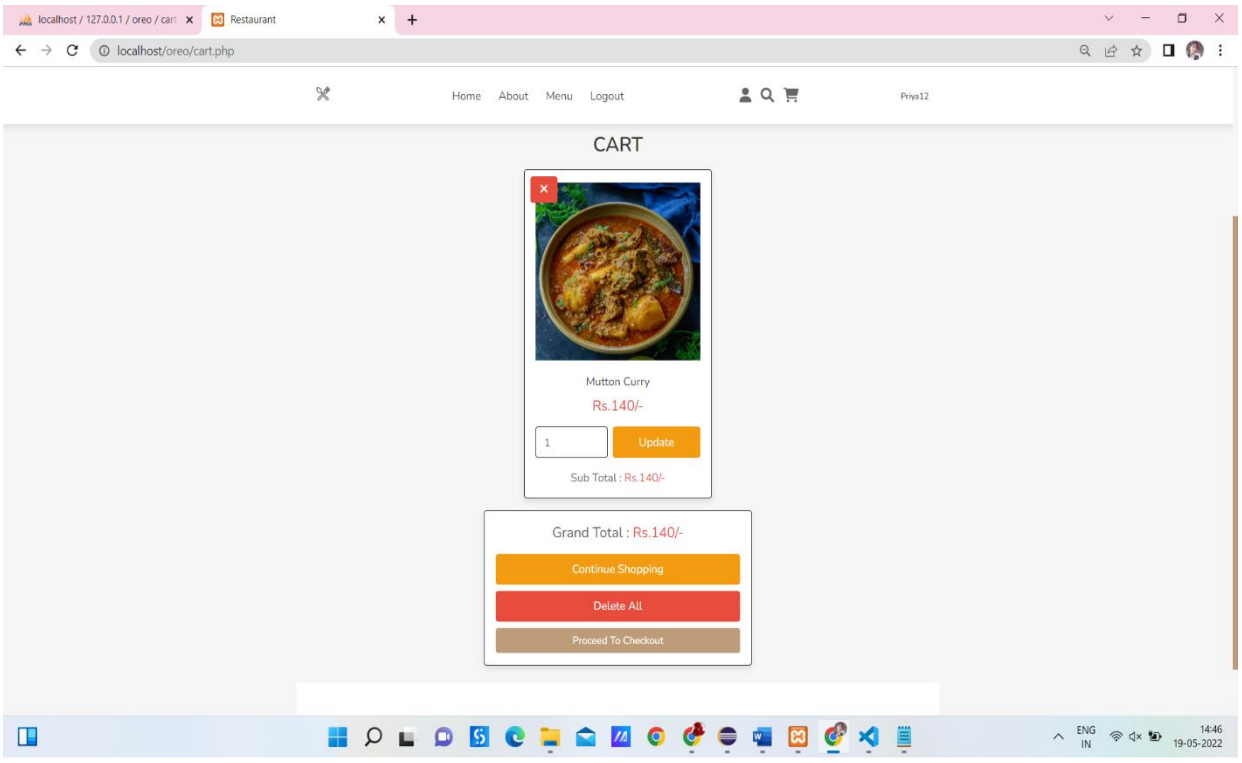
Customer Home page


localhost/oreo/home.php
Home
About
Menu
Reservation
Priya12
Tasty Passion
Lorem Ipsum Dolor Sit Amet Consectetur, Adipisicing Elit. Ut Officia, Accusantium Mollitia Laudantium Dolorum Dolore
About Us

Product page

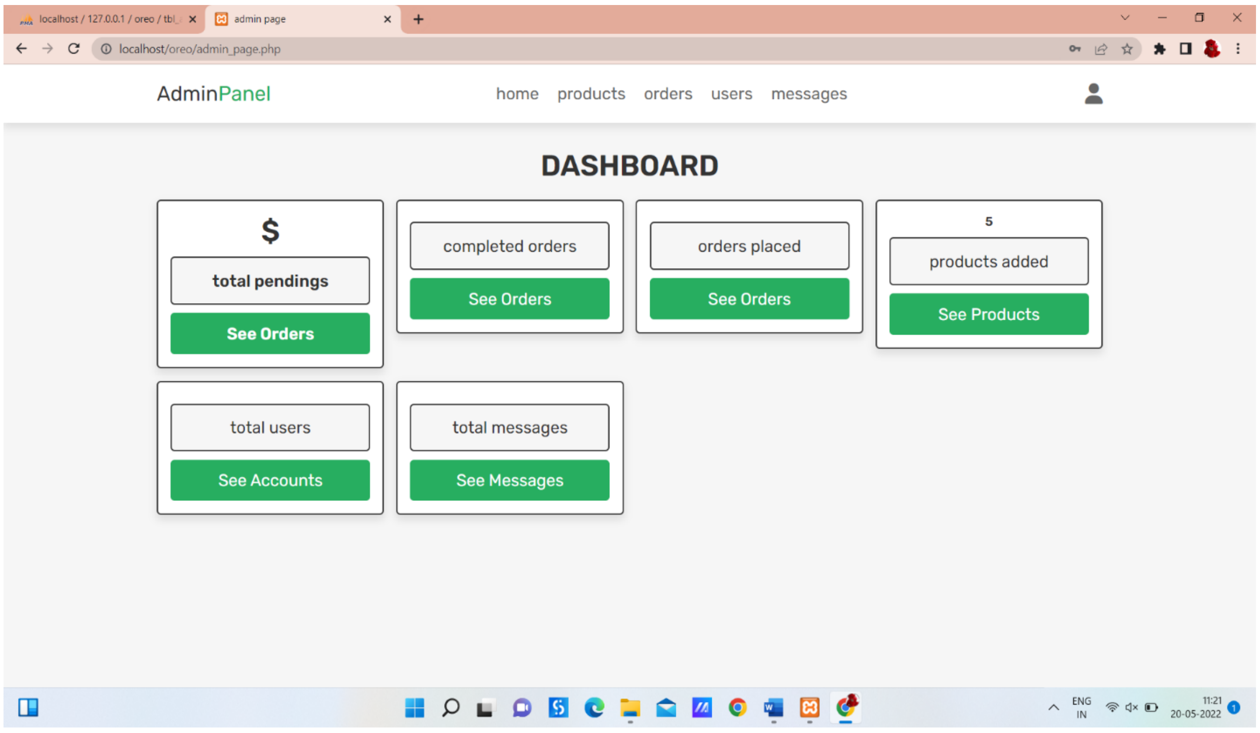


Cart page

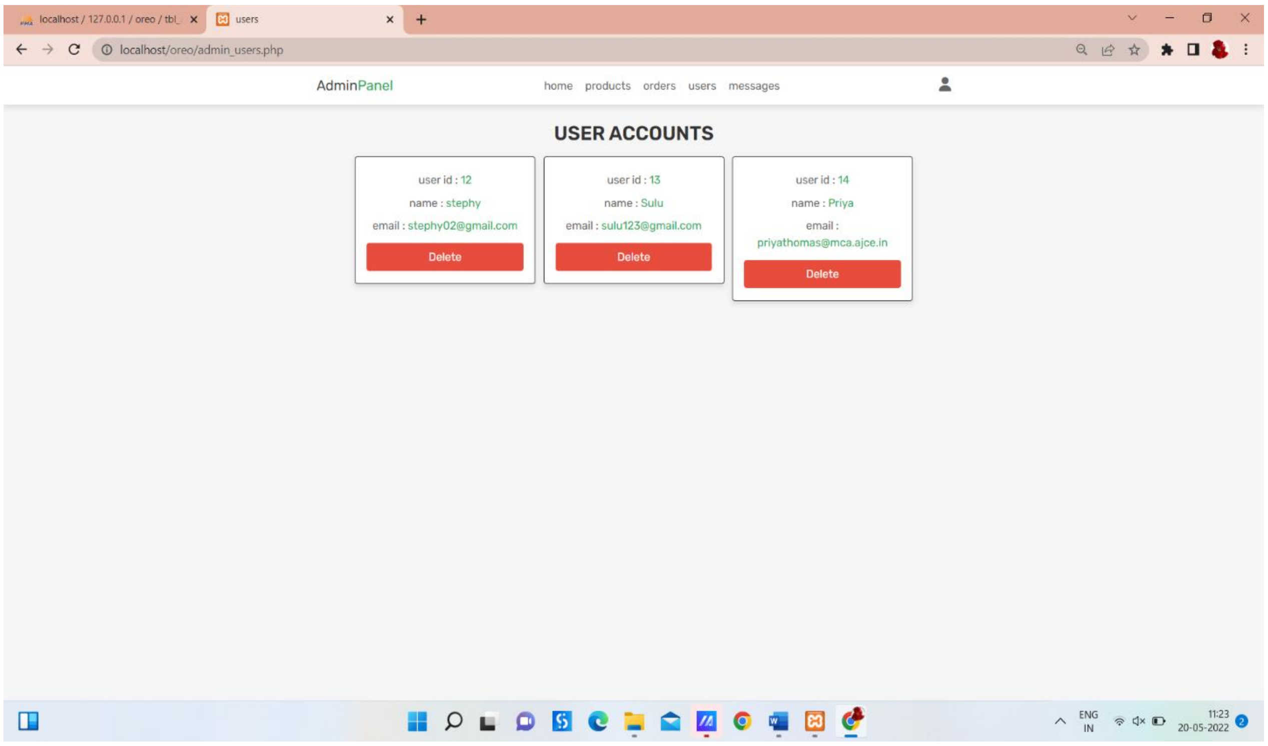

localhost / 127.0.0.1 / oreo / cart
Restaurant
localhost/oreo/cart.php
Home
About
Menu
Logout
Priya12
CART
Mutton Curry
Rs.140/-
1
Update
Sub Total : Rs.140/-
Grand Total : Rs.140/-
Continue Shopping
Delete All
Proceed To Checkout
ENG IN
14:46
19-05-2022

ADMIN PAGES

Admin Dashboard Page



View all registered customers


localhost / 127.0.0.1 / oreo / tbl_
users
localhost/oreo/admin_users.php
AdminPanel
home
products
orders
users
messages
11:23
20-05-2022

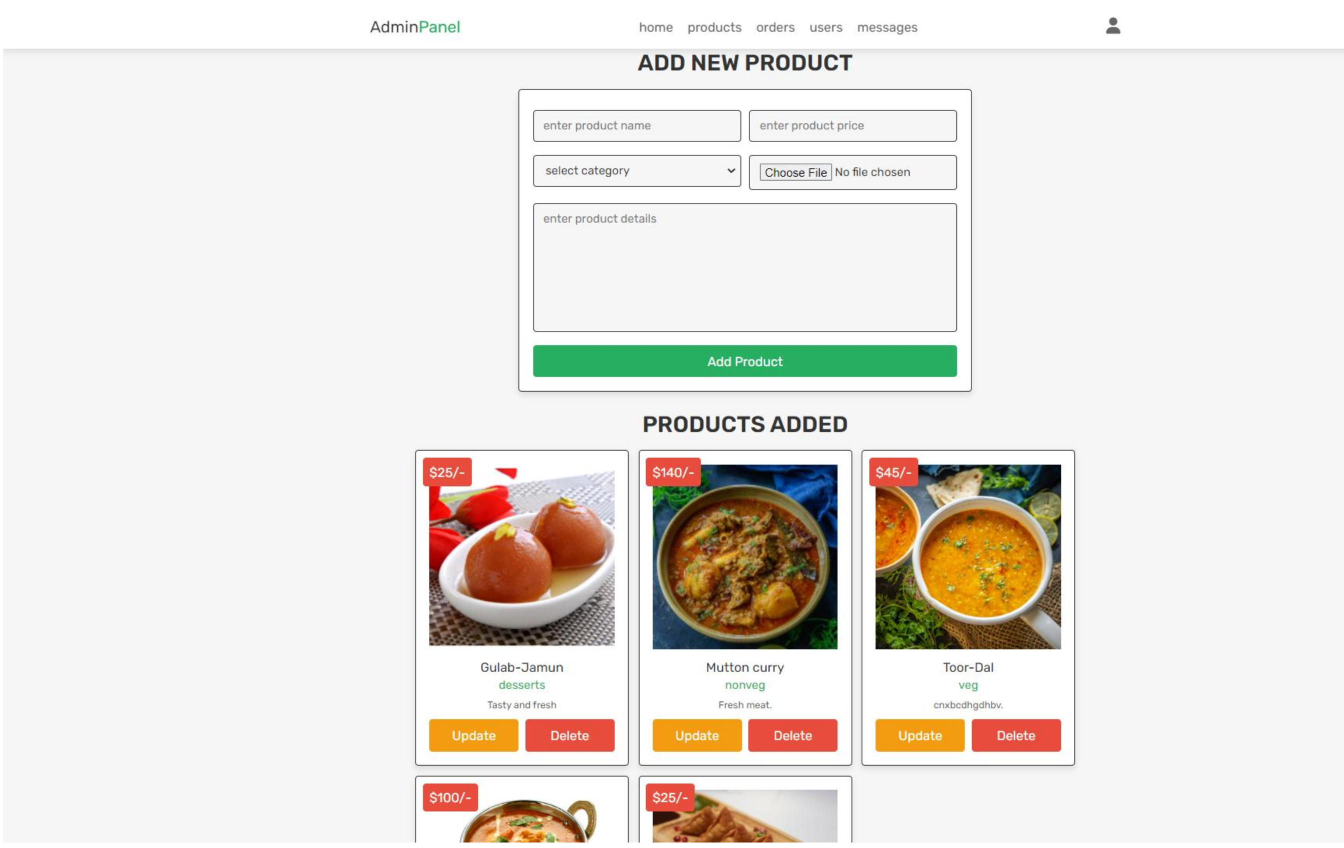
USER ACCOUNTS

user id : 12
name : stephy
email : stephy02@gmail.com
Delete

user id : 13
name : Sulu
email : sulu123@gmail.com
Delete

user id : 14
name : Priya
email : priyathomas@mca.ajce.in
Delete

Add Products & View Products


AdminPanel
home products orders users messages
ADD NEW PRODUCT
enter product name
enter product price
select category
Choose File No file chosen
enter product details
Add Product
PRODUCTS ADDED
$25/-
Gulab-Jamun
desserts
Tasty and fresh
Update
Delete
$140/-
Mutton curry
nonveg
Fresh meat.
Update
Delete
$45/-
Toor-Dal
veg
cnxbcdhgdhbv.
Update
Delete
$100/-
$25/-