ONLINE FOOD ORDERING AND TABLE RESERVATION SYSTEM

A PROJECT REPORT

Submitted by

K. MEGHANA CHOWDARY	**316126510147**
K. HARITHA	**316126510149**
M. SRI MANJARI	**316126510152**
M. NAVEEN	**316126510156**
MD. MEHAR NAAZ	**316126510158**
K. NAREN SAI KRISHNA	**316126510164**
R. SREE MEGHANA	**316126510170**

In fulfillment for the mini project

Of

OPEN SOURCE TECHNONOLOGIES LAB

IN

COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Autonomous)

SANGIVALASA, VISAKHAPATNAM - 531 162

2016-2020

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Autonomous)

SANGIVALASA, VISAKHAPATNAM-531 162

BONAFIDE CERTIFICATE

Certified that this project report **"ONLINE FOOD ORDERING AND TABLE RESERVATION SYSTEM**" is the bonafide work of **"K. Meghana Chowdary (316126510147), K. Haritha (316126510149), M. Sri Manjari (316126510152), M.Naveen (316126510156), MD. Mehar Naaz (316126510158), K. Naren Sai Krishna (316126510164), R.Sree Meghana (316126510170) "**who carried out the project work under my supervision**.**

Mr. S. JOSHUA JOHNSON	**Dr. R. SIVARANJANI**
(ASSISTANT PROFESSOR)	**(HEAD OF THE DEPARTMENT)**
(SUPERVISOR)	
COMPUTER SCIENCE AND ENGINEERING	COMPUTER SCIENCE AND ENGINEERING
ANITS	ANITS

DECLARATION

We,***K.MEGHANACHOWDARY***(316126510147)**,*****K.HARITHA***(316126510149)**,** ***M.SRIMANJARI***(316126510147)**,** ***M.NAVEEN***(316126510147)**,** ***MD.MEHAR NAAZ***(316126510147)**,** ***K.NAREN SAI KRISHNA***(316126510147)**,** ***R.SREE MEGHANA***(316126510147) students of third year second semester **B.TECH, Computer Science & Engineering from Andhra University, Visakhapatnam**, hereby declare that the mini project work entitled" **ONLINE FOOD ORDERING AND TABLE RESERVATION SYSTEM** " is carried out by us and submitted in fulfillment of the reqirements for "**Mini Project of Open Source Technologies Lab**" ,under Anil Neerukonda Institute of Technology & Sciences during the Academic year 2016-2020 and has not been submitted to any other university.

K. MEGHANA CHOWDARY	**316126510147**
K. HARITHA	**316126510149**
M. SRIMANJARI	**316126510152**
M. NAVEEN	**316126510156**
MD. MEHAR NAAZ	**316126510158**
K. NAREN SAI KRISHNA	**316126510164**
R. SREE MEGHANA	**316126510170**

ACKNOWLEDGEMENT

An endeavor over a long period can be advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

We owe our tributes to **Dr.R.Sivaranjani, Head of the Department, Computer Science & Engineering for** her valuable support and guidance during the period of project implementation.

We wish to express our sincere thanks and gratitude to our project guides Mr. **S. JOSHUA JOHNSON** (Assistant professor) Department of **COMPUTER SCIENCE AND ENGINEERING, ANITS,** for the simulating discussions, in analyzing problems associated with our project work and for guiding us throughout the project. Project meeting were highly informative. We express our sincere thanks for the encouragement, untiring guidance and the confidence they had shown in us. We are immensely indebted for their valuable guidance throughout our project.

We also thank all the staff members of CSE department for their valuable advices.

We also thank Principal and supporting staff for providing resources as and when required.

K.MEGHANA CHOWDARY	**316126510147**
K.HARITHA	**316126510149**
M.SRI MANJARI	**316126510152**
M.NAVEEN	**316126510156**
MD.MEHAR NAAZ	**316126510158**
K.NAREN SAI KRISHNA	**316126510164**
R.SREE MEGHANA	**316126510170**

TABLE OF CONTENTS

ABSTRACT

The application is about the online food ordering and table reservation mechanism of the restaurant "Relish, A Foodie's Paradise". The application is user-friendly and allows the user to order food by sitting relaxed at their home. It also allows them to reserve their favorite table in the restaurant, which cuts down waiting time. This application thereby improves the revenue of the restaurant since the customer finds this way easy to satisfy their food cravings.

There are different components in the application: User Account, Menu Card, Details, Table Reservation.

1) *Details*: This tab provides the address and phone number of the restaurant for the sake of customer.

2) *Menu Card*: The menu card presents a variety of food items in different cuisines like Mexican, Italian, Chinese, etc. The customer can customize the items they order.

3) *User Account:* A user can place order only after logging into his Relish application account.

4) *Table Reservation*: Customer can know the availability of space, select from the available seating having different outdoor views.

Payment: When the customer chooses what they want to eat, they choose the payment mode:

a) Debit Card b) Credit Card c) Net banking d) Cash on delivery

Coupon Code: The customer can enter coupon code they received through their account for discounts. The coupon codes may be specific to customers, based on how frequently they order.

Rating: The application allows customers to rate the food delivery experience for the different categories: 1) Taste 2) Quality 3) Value for money 4) Delay of delivery. The user may also give suggestions for the benefit of the restaurant.

Hence the application is beneficial to the customer and also promotes the food business of Relish.

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Online Food Ordering and Table Reserving System website is based on the strategy,provides customers to order food and reserve table efficiently. The website is implemented as software application, gives the advantage to the customer does all of this in one step, selecting the items from the menu and mentioning any other details. The order gets placed on the system and can be processed by the restaurant very quickly. It is an advantage for the customer to know in advance that he will not have to go through the trouble of waiting until a table is available, or being put on a waiting list. The main purpose of this SRS is to illustrate the requirements of the project Online Food Ordering and Table Reservation System application and how the application is developed using certain hardware and software specification.

1.1 Overall Description

Online Food Ordering and Table Reservation System is a web application. This system is developed to provide service facility to restaurant and also to the customer.

Online ordering system is an application greatly simplifies the ordering process for both the customer and the restaurant. This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find free tables or place orders. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

2.System Analysis

System analysis is the description of a system into its component pieces to study how the component pieces are study and work.

2.1. Software Requirement Specification

Software Requirement Specification is the starting point of software developing activity.As system grew more complex it became evident that the goal of the entire system cannot beEasily comprehended. Hence the needs for the requirements phase are use. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds ofclients (the input) into a formal document. The purpose of the software requirement specification is to reduce the communication gap between the clients and developers. Software RequirementSpecification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties of involved in the system.

2.1.1. Purpose

The purpose of Online Food Ordering and Table Reservation is to show the software requirements of the food ordering and table reservation software.This provides customer to order food and reserve the table efficiently.This provides accurate information about the food orders.

2.1.2. Scope

This project will help to manage and run the restaurant business systematically. In this, we will provide an application that can be used by customers to order food and reserve table. Customers can also give feedback or rating through this application, so that admin can evaluate whole system.

The benefits are:

a) It is a unique software which helps to organize event without any paper work.

b) It has a wide variety of modules.

2.1.3. Objective:

The main objective of the system is to simplify and improve the efficiency of the ordering food and table reserving process for both customer and restaurant.

2.1.4. Proposed System

This online application enables the end users to:

- Register online,
- Select the food items from the list available,
- Order food,
- Pay for the order, and
- Reserve Table at a specified date and time.

2.2 Functional Requirements:

- **Table Reservation**: User can reserve any table through this application any time and from anywhere.
- **Food Ordering**: User can also order their favorite food using this application.
- **Feedback:** User can tell his experience for the services provided by giving rating in the "Rate Us" section.

2.3 Non-Functional Requirements:

1. Performance requirements:

- The system need to be reliable.
- If unable to process the request then appropriate error message.
- Web pages are loaded within few seconds.

2. Safety requirements:

- The details need to be maintained properly.
- Users must be authenticated.
- The database must be kept backed up.

3. Security Requirements:

- After entering the password and user id only, user can access his profile.
- The details of user must be safe and secure.
- Sharing of details.

Software Requirements:

- **Operating System** **:** Windows, Mac, Linux.
- **Web Technologies** **:** Html, JavaScript, CSS, PHP
- **Web Server** **:** Wamp
- **Database** **:** My SQL
- **Web Browser** **:** IE, Mozilla Firefox or Google Chrome
- **Browser Configuration:** JavaScript must be enabled,Cookies must be enabled, Pop-up windows must be enabled.

Hardware Requirements:

- **Processor** **:** Intel Pentium 4 or higher
- **RAM** **:** 512 MB or higher
- **Hard-drive space** **:** 60 MB or higher
- **Internet Connection** **:** 4 Mbps or higher
- **Controller** **:** Keyboard and a Mouse

3. System Design

Object oriented design is concerned with developing an object-oriented model of a software system to implement requirements. It is the process of designing the components, interfaces, objects, classes, attributes and operations that will satisfy the requirements. Typically start with the candidate objects defined during the analysis, but add much more rigor to their definition. Then add or change as needed to refine a solution.

The designer's goal is how the outputs to be produced and in what format samples of the output are also presented. The processing phases are handled through the program construction and testing. Finally details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step towards implementation.

The importance of software design can be stated in a single word "Quality". Design provides us with representations of software that can be accessed for quality. Design is the only way that can be able to accurately translate a customer's requirements into a finished software product or system without design risk, building an unstable system, that might fail it if small changes are made or may be difficult to test, or one who's quality can't be tested. So it is essential phase in the development of a software product.

Object-oriented design can yield the following benefits:

Maintainability through simplified mapping to the problem domain, which provides for less analysis effort, less complexity in system design, easier verification by the user.

Reusability of the design artifacts, which saves time and cost.

Productivity gains through direct mapping of features of Object-Oriented Programming languages.

Architecture diagram:

Software Architecture:

The architecture of a system describes its major components, their relationships and how they interact with each other. Architecture serves as a blueprint for a system.

It provides an abstraction to manage the system complexity and establish a communication and coordination among components. It defines a structured solution to meet all the technical and operational requirements, while optimizing the common quality attributes like performance and security.

Component and Connector View:

It is a collection of run time entities are called components. A component is a unit which has a identity in the executing system. While executing the components they need to interact with other to support system services. Connectors provide interaction. Component and Connector view is the most common view of architecture. Components are units of data stores in the system. A component has a name which is generally chosen to represent the role of the component (or) the function it performs. The name provides a unique identity to the component. A connector has a name that should describes the nature of interactions the connector supports.

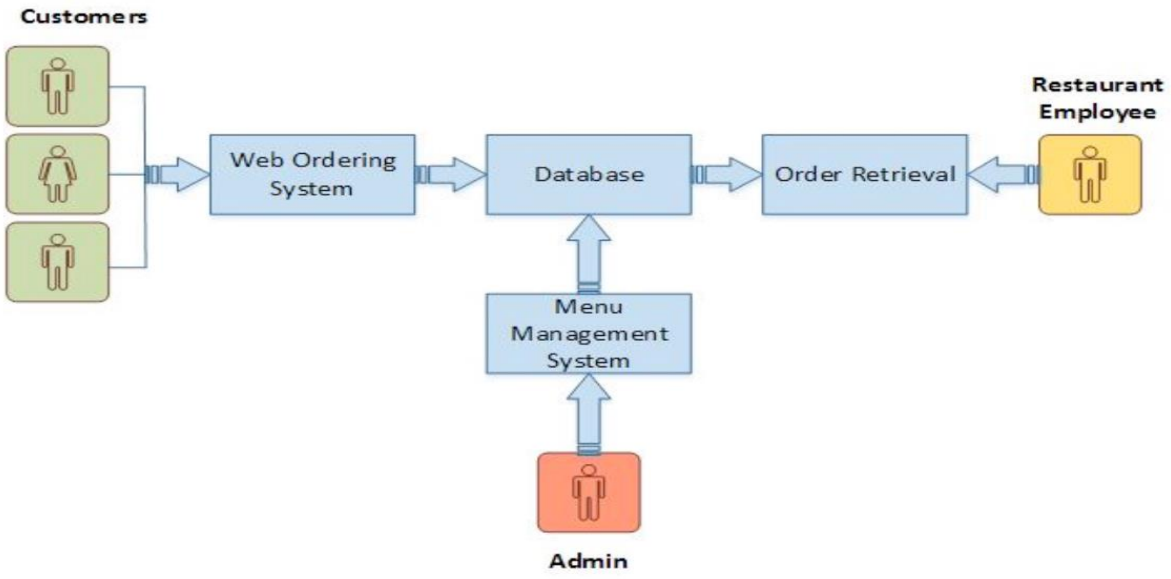


Fig:3.0 Architecture diagram

3.1 Data Flow Diagram

The DFD is also known as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. It maps out the flow of information for any process or system, how data is processed in terms of inputs and outputs. It uses defined symbols like rectangles, circles and arrows to show data inputs, outputs, storage points and the routes between each destination. They can be used to analyze an existing system or model of a new one. A DFD can often visually "say" things that would be hard to explain in words and they work for both technical and non-technical.

There are four components in DFD: 1. External Entity

2. Process

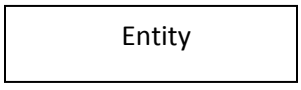
3. Data Flow

4. data Store

1. External Entity:

It is an outside system that sends or receives data, communicating with the system. They are the sources and destinations of information entering and leaving the system. They might be an outside organization or person, a computer system or a business system. They are known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram. These are sources and destinations of the system's input and output.

Representation:

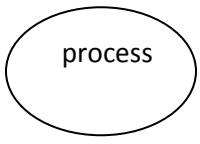


In the DFD shown below USER and ADMIN are the external entities.

2. Process:

It is just like a function that changes the data, producing an output. It might perform computations for sort data based on logic or direct the dataflow based on business rules.

Representation:



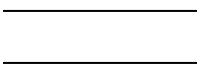
3. Data Flow:

A dataflow represents a package of information flowing between two objects in the data-flow diagram, Data flows are used to model the flow of information into the system, out of the system and between the elements within the system.

Representation:

4. Data Store:

These are the files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label.

Representation: 

DFD Levels:

A data flow diagram can dive into progressively more detail by using levels. DFD levels are numbered 0, 1 or 2 and occasionally go to even level 3 or beyond. The necessary level of detail depends on the scope of the task.

- **DFD Level 0:**

  It is also called a context diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to entities. It should be easily understood.

- **DFD Level 1:**

It provides a more detailed breakout of pieces of the Context Level Diagram. The main functions carried out by the system, break-down of the high-level process of the Context Diagram into its sub-process.

- **DFD Level 2:**

This goes one step deeper into parts of level 1. It may require more text to reach the necessary level of detail about the system's functioning.

Data-Flow Diagram

Level 0 or Context Diagram:

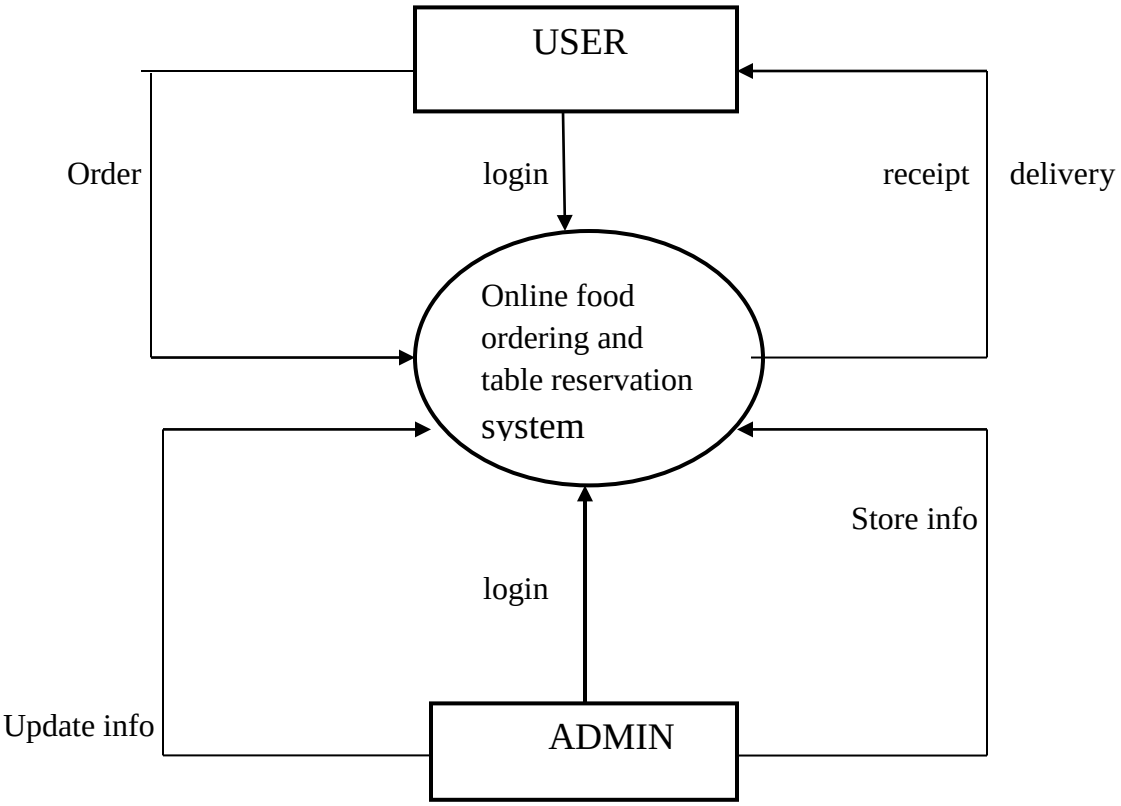


Fig: 3.1(a) level 0 DFD for online food ordering and table reservation system

Level 1 diagram:

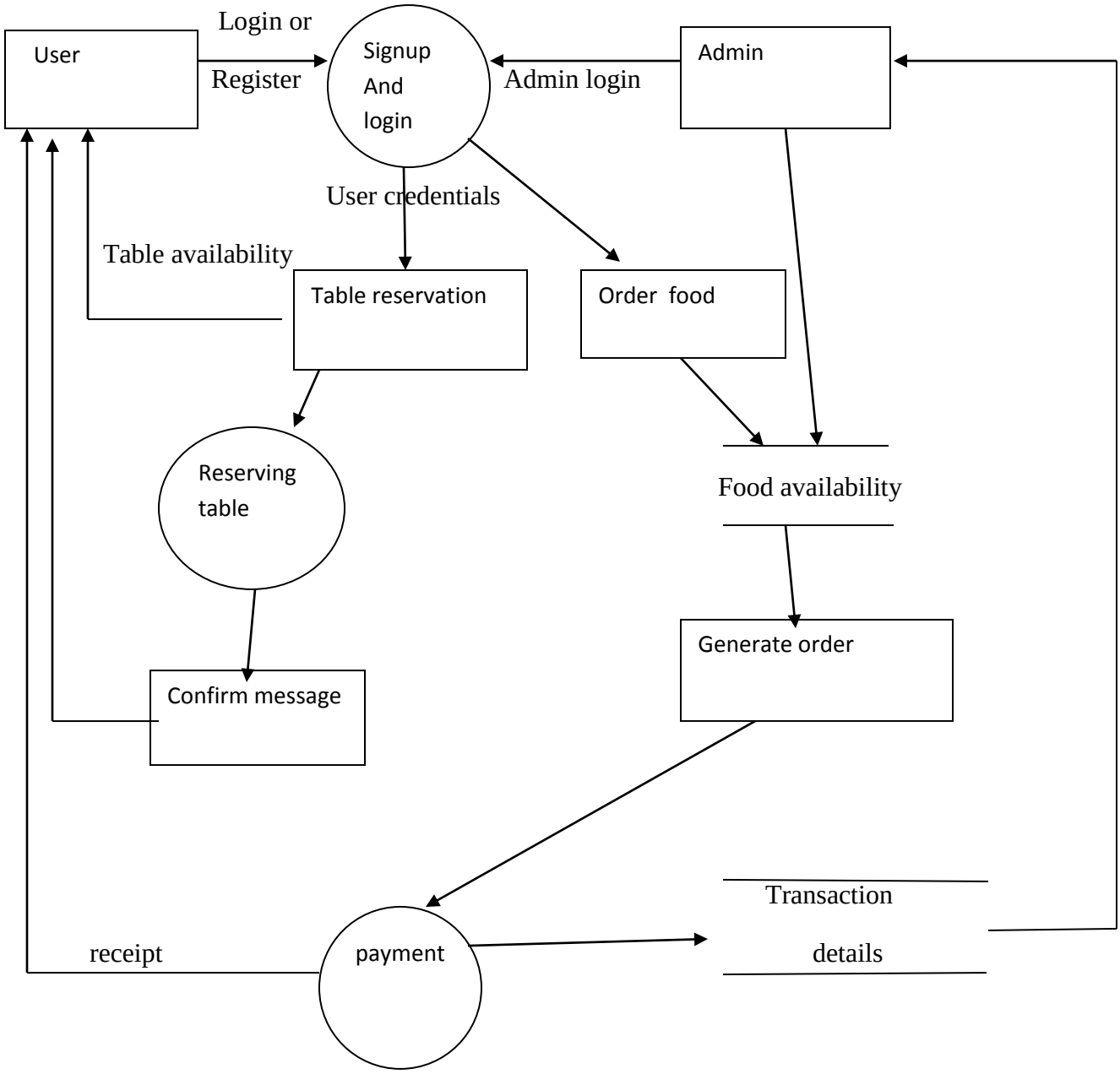


Fig: 3.1(b) level 1 DFD for online food ordering and table reservation system

Level 2 diagram:

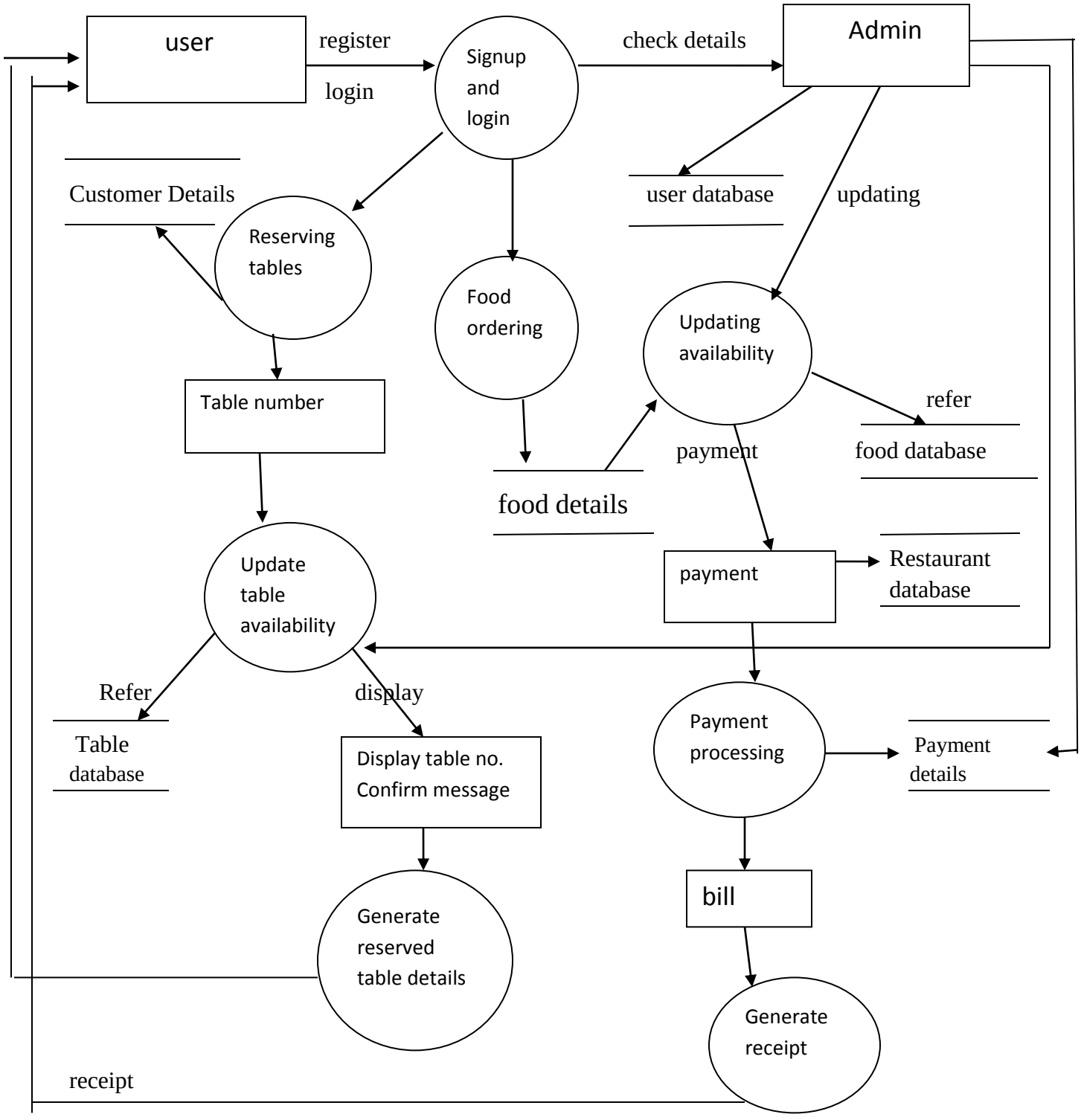


Fig: 3.1(c) level 2 DFD for online food ordering and table reservation system

3.2 UML Design

UML stands for Unified Modeling Language. Taking SRS document of analysis as input to the design phase drawn UML diagrams. The UML is only language so is just one part of the software development method. The UML is process independent, although optimally it should be used in a process that should be driven, architecture-centric, iterative, and incremental. The UML is language for visualizing, specifying, constructing, documenting the articles in a software-intensive system.

A modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of the system. A modeling language such as the UML is thus a standard language for software blueprints.

The UML is a graphical language, which consists of all interesting systems. There are also different structures that can transcend what can be represented in a programming language.

These are different diagrams in UML.

3.2.1 Use Case Diagram

Use Case during requirement elicitation and analysis to represent the functionality of the system. Use case describes a function by the system that yields a visible result for an actor.

The identification of actors and use cases result in the definitions of the boundary of the system i.e., differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system. Use case describes the behavior of the system as seen from the actor's point of view. It describes the function provided by the system as a set of events that yield a visible result for the actor.

Use case for placing order:

Use case scenario:

Use case name	Place order
Participating actors	Customer, admin
Flow of events	Enter login id(c) Enter password(c) Submit(c) Check credentials(a) Select type of cuisine(c) Select food items(c) Check availability(a) Enter submit(c) Generate order(a) Display order(a)
Entry condition	Login, password
Exit condition	logout
Quality requirements	-

Table 1(a): Use case scenario for placing food order

Use case diagram:

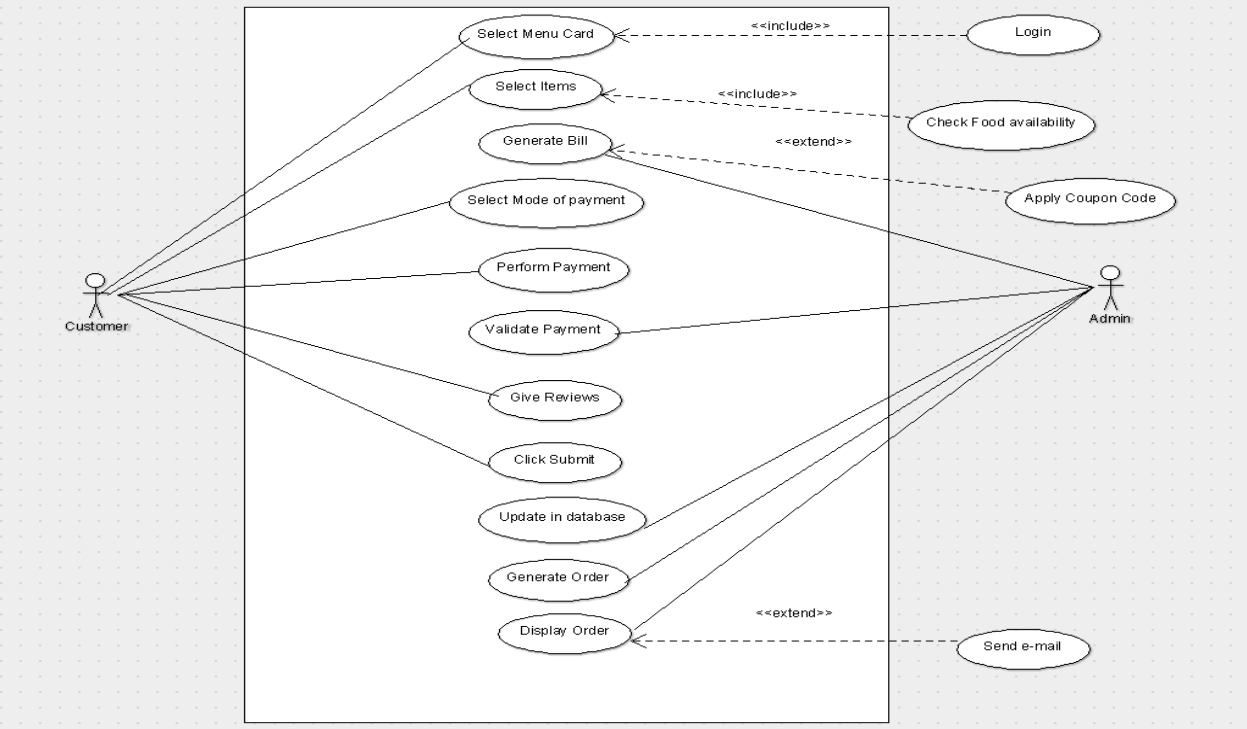


Fig: 3.2.1(a) Use case for placing order

Use Case diagram for Table reservation

Use case scenario:

Use case name	Login
Participating actors	Customer, admin
Flow of events	Select table reservation(c) Check availability of seats(a) Select seat number(c) Display table no.'s(a) Select table(c) Click submit(c) Update availability(a) Display message(a)
Entry condition	Login, password
Exit condition	Logout
Quality requirements	-

Table 1(b): Use case scenario for table reservation

Use case diagram:

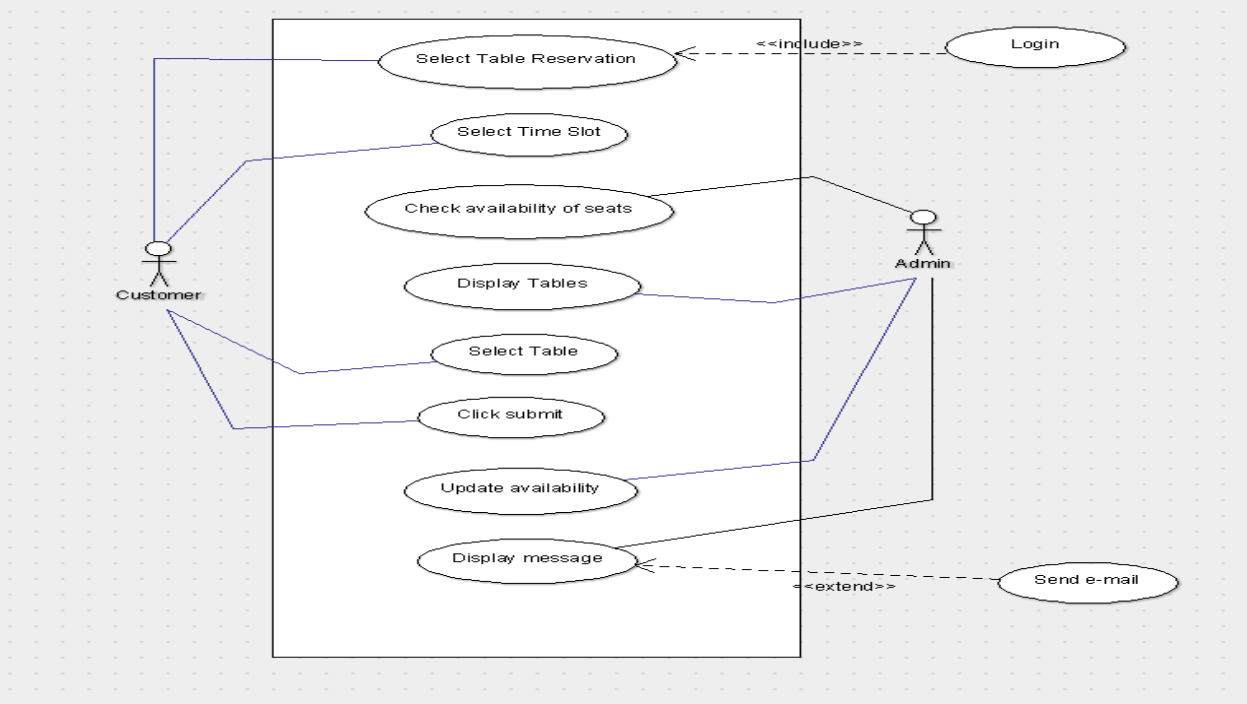


Fig: 3.2.1(b) Use case for table reservation

Use Case for Payment and Feedback

Use case scenario:

Use case name	Payment and Feedback
Participating actors	Customer, Server
Flow of events	Enter login id(c) Check user id(c) Enter Password(c) Submit(c) Check credentials(s) Perform Payment Validate Payment Give Feedback or Rating Click Submit
Entry condition	Login, password, Card details
Exit condition	Logout
Quality requirements	Secure Internet Connection

Table 1(c): Use case scenario for payment and feedback

Use case diagram:

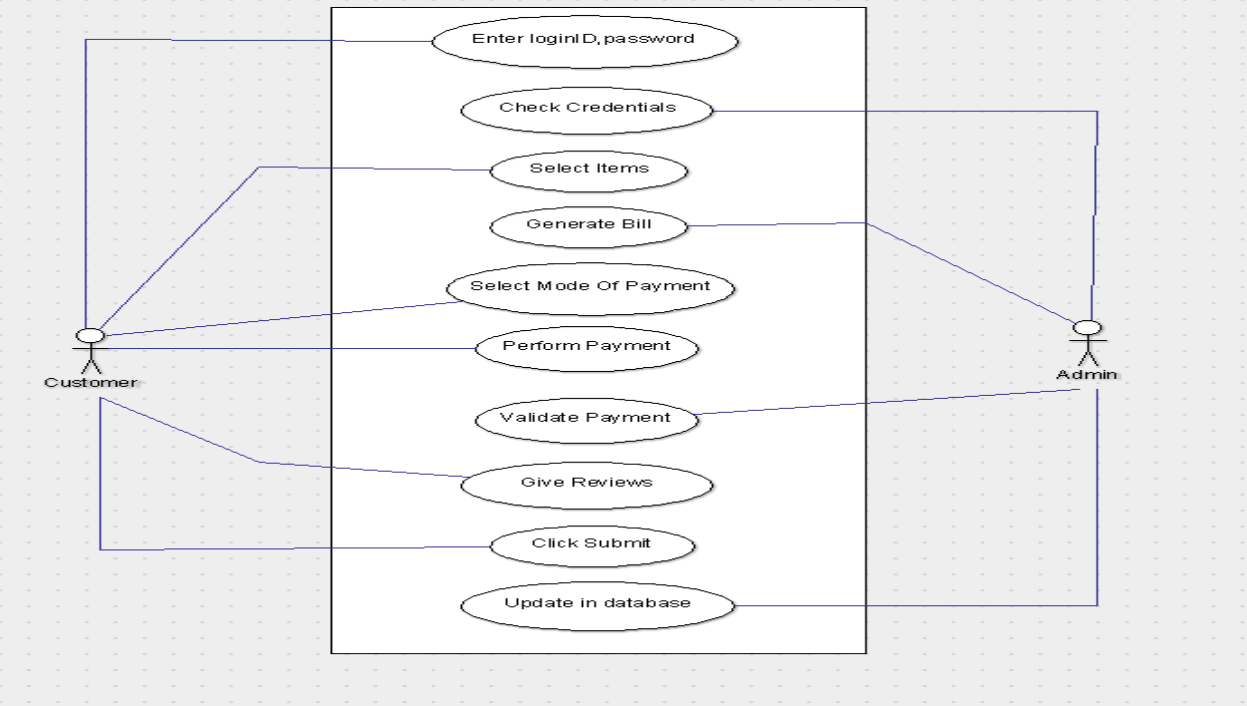


Fig: 3.2.1(c) Use case for payment and feedback

3.2.2 Class diagram

Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagram describe the different perspective when designing a system-conceptual, specification and implementation. Classes are composed of three things: name, attributes, and operations. Class diagram also display relationships such as containment, inheritance, association etc. The association relationship is most common relationship in a class diagram. The association shows the relationship between instances of classes.

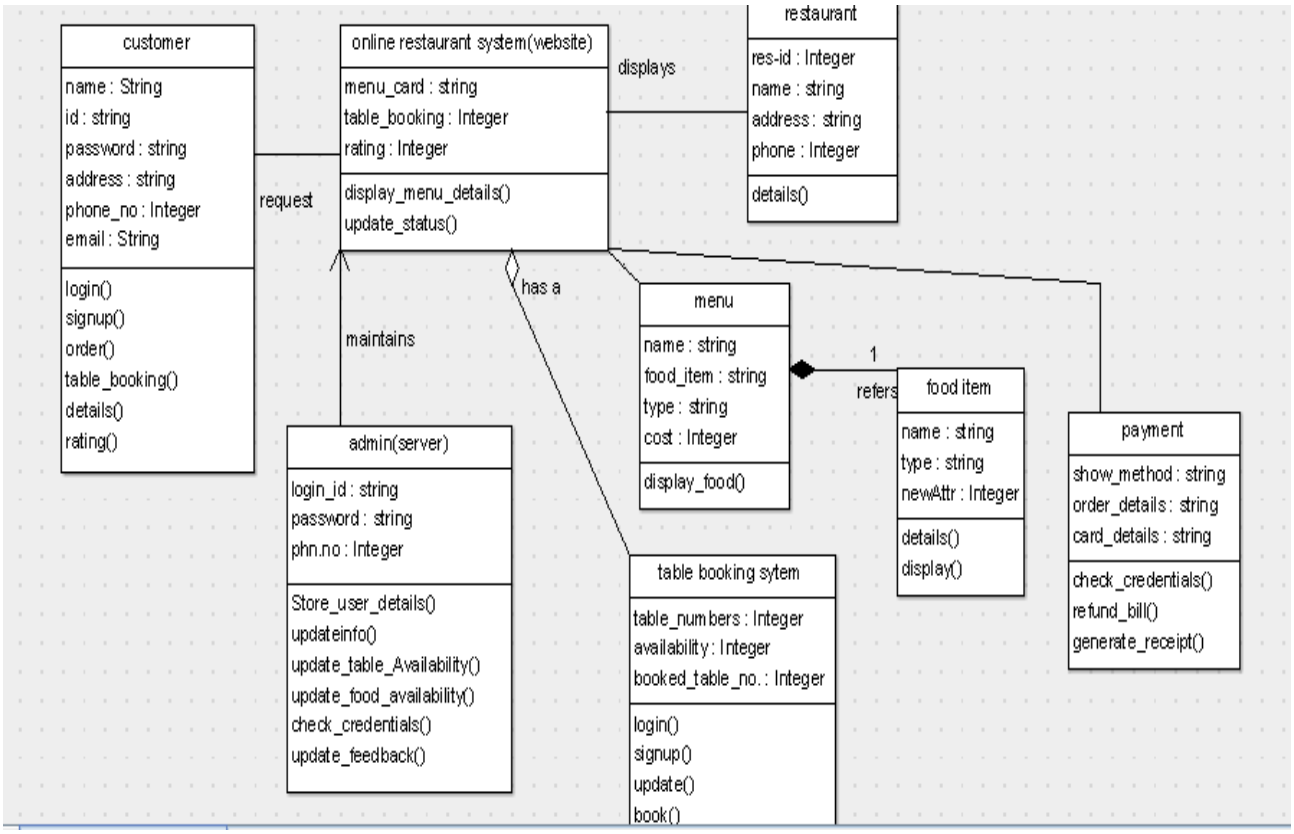


Fig: 3.2.2 Class diagram for online food ordering and table reservation system.

3.2.3 Sequence Diagram:

Sequence diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).

Objects: Object can be viewed as an entity at a particular point in time with specific value and as a holder of identity.

Sequence diagram for placing Order:

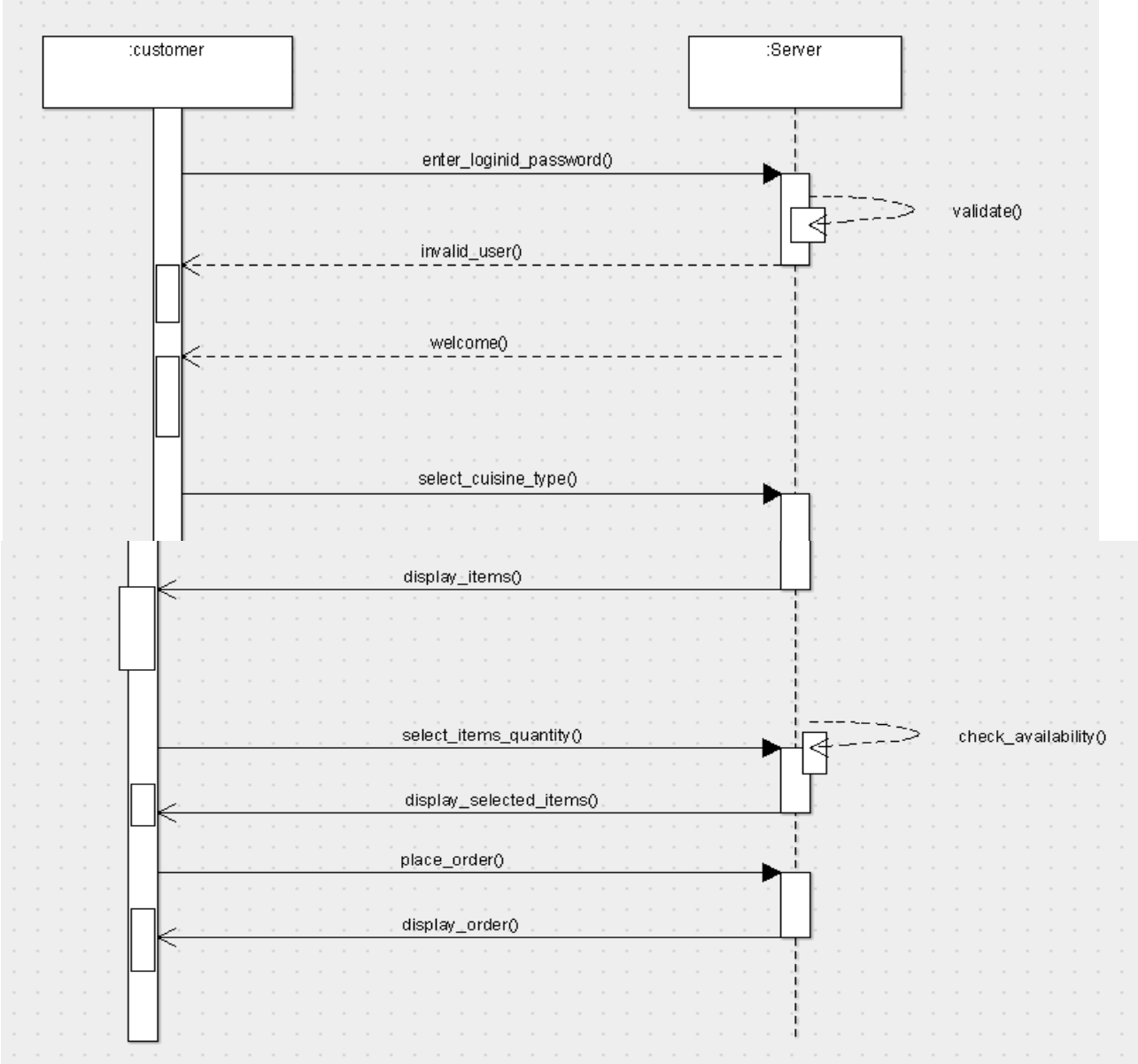


Fig: 3.2.3(a) sequence diagram for placing order

Sequence diagram for Table Reservation module:

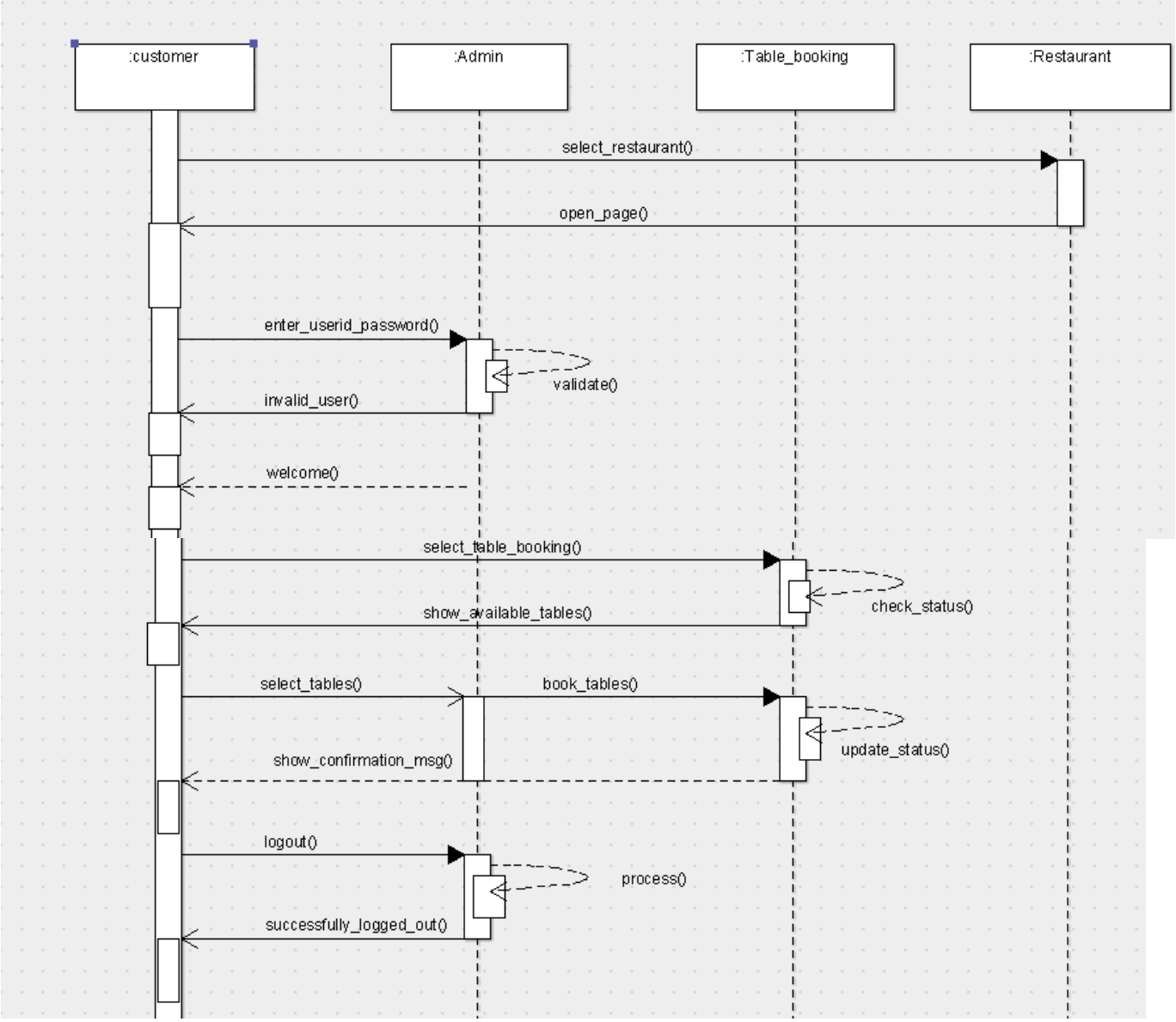


Fig: 3.2.3(b) sequence diagram for table reservation

Sequence diagram for Rating module:

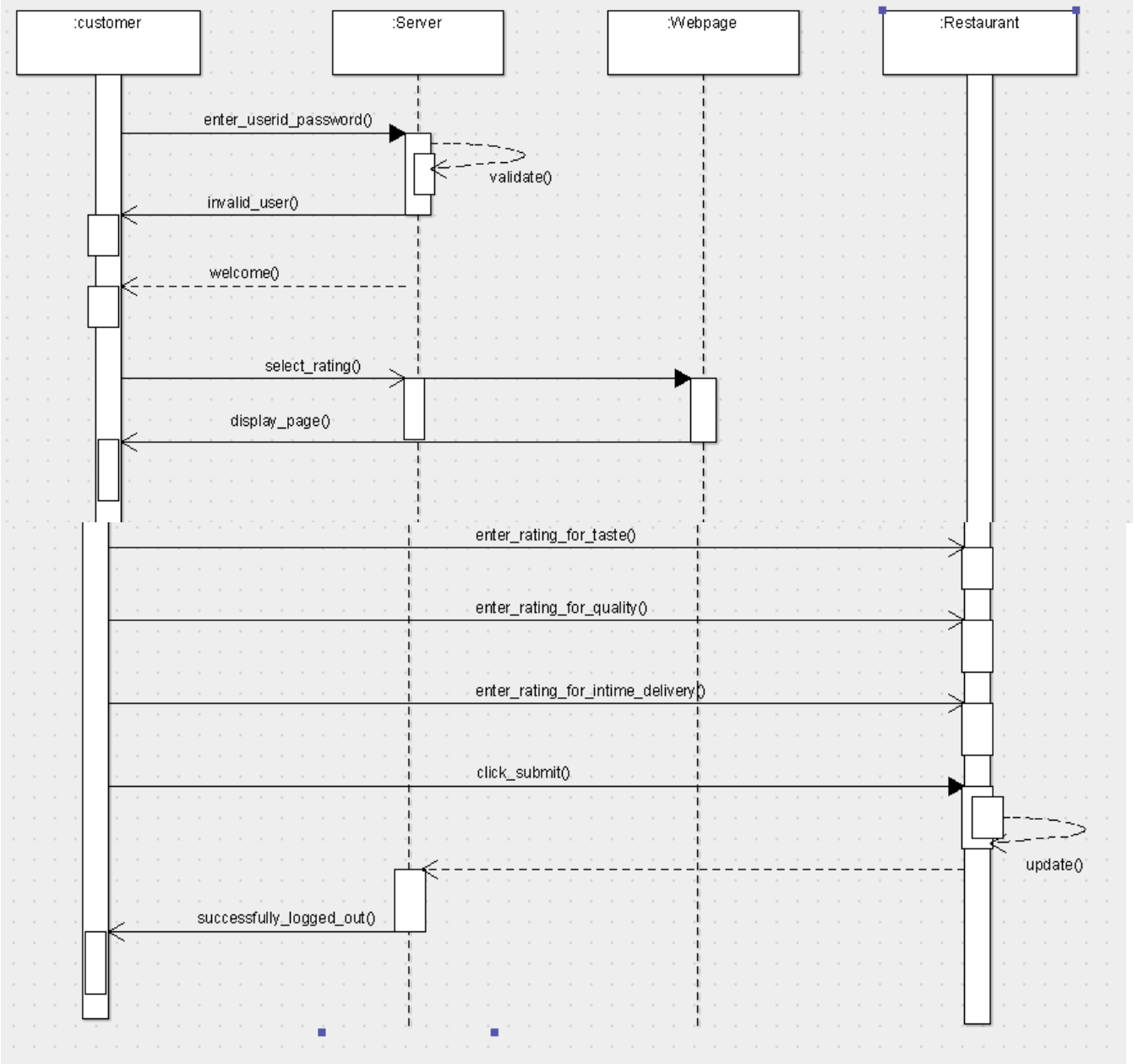


Fig: 3.2.3(c) sequence diagram for rating

3.2.4 Collaboration Diagram

Collaboration diagram displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages. Collaboration diagram is the dynamic behavior of objects in addition to sequence diagram. The transformation form of a sequence diagram into a collaboration diagram is a bi-directional function. The difference between the sequence diagrams and collaboration diagrams is that collaboration diagram emphasizes more on the structure than the sequence of interactions.

Collaboration diagram have two features:

1. There is a path to indicate how one object is linked to another.

2. There is a sequence number to indicate the time order of messages.

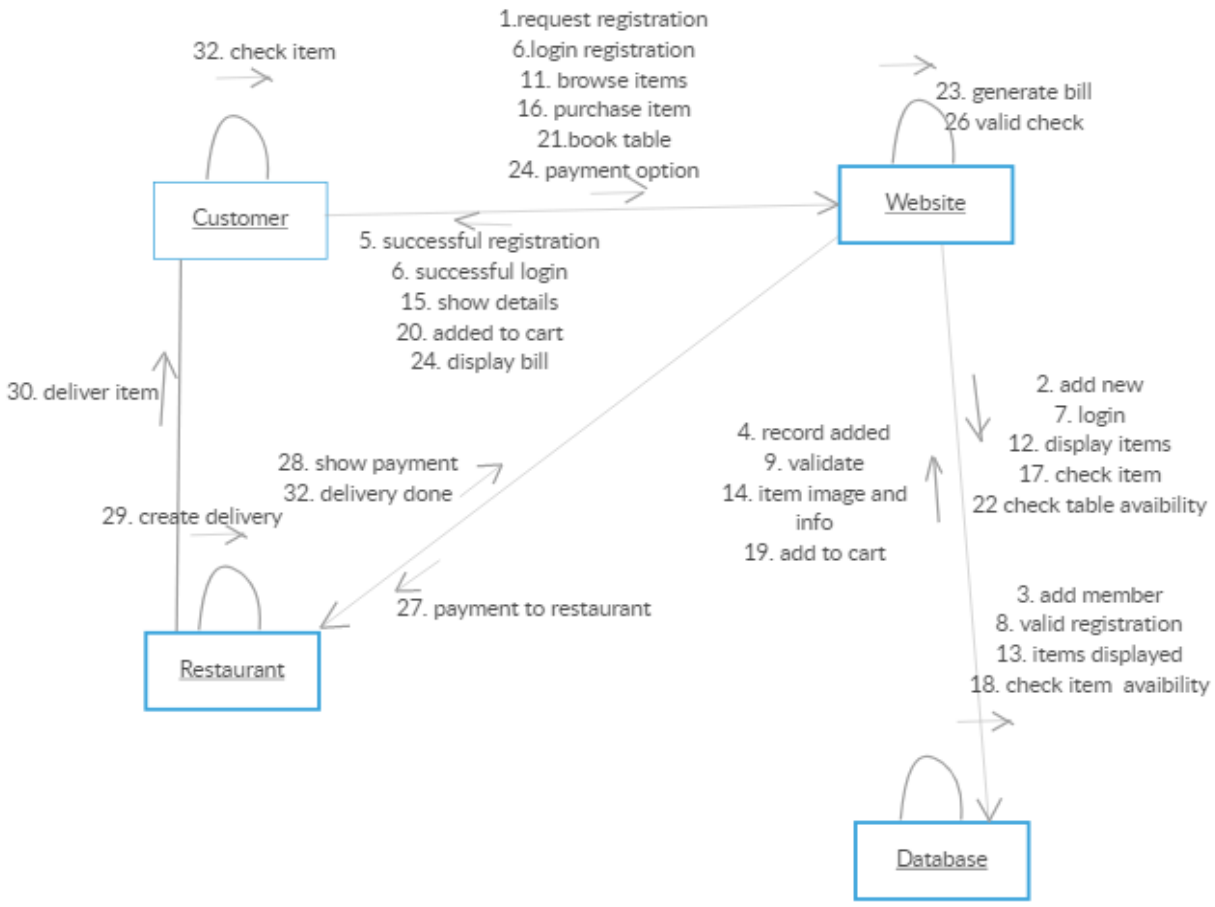


Fig: 3.2.4 collaboration diagram for online food ordering and table reservation system.

3.2.5 State Chart diagrams

A state chart diagram describes a state machine which shows the behavior of classes. It shows the actual changes in state not processes or commands that create those changes and is the dynamic behavior of objects over time by model1ing the life cycle of objects of each class.

It describes how an object is changing from one state to another state. There are mainly two states in State Chart Diagram: **1**. Initial State **2**. Final-State.

Some of the components of State Chart Diagram are:

State: It is a condition or situation in life cycle of an object during which it's satisfies same condition or performs some activity or waits for some event.

Transition: It is a relationship between two states indicating that object in first state performs some actions and enters into the next state or event.

Event: An event is specification of significant occurrence that has a location in time and space.

State chart diagram for login:

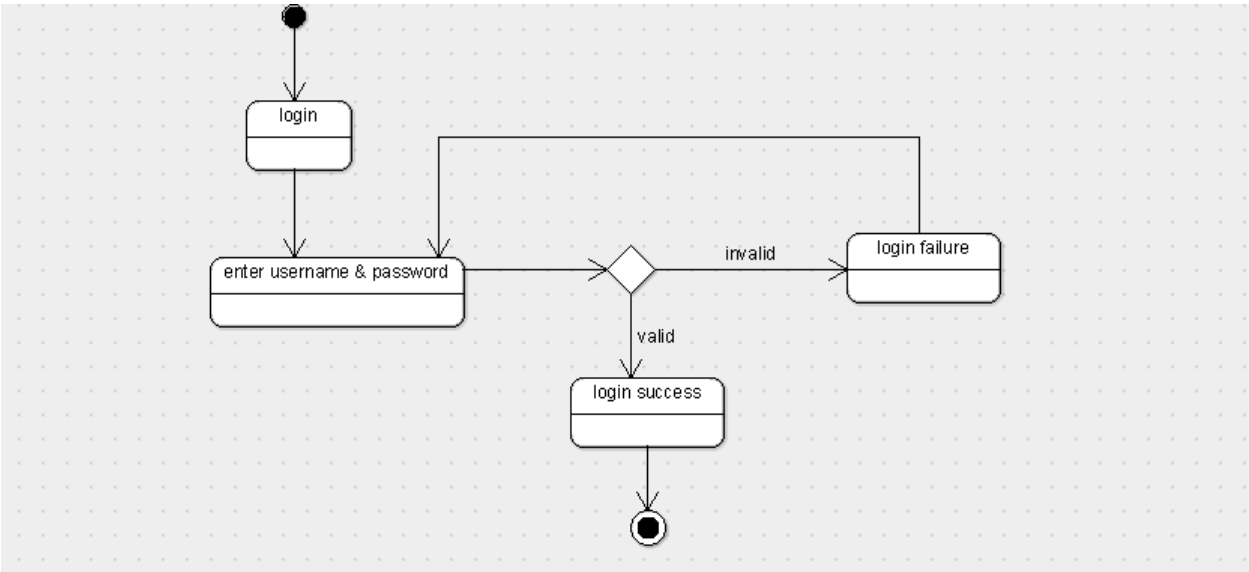


Fig: 3.2.5(a) state chart diagram for login

State chart diagram for placing order:

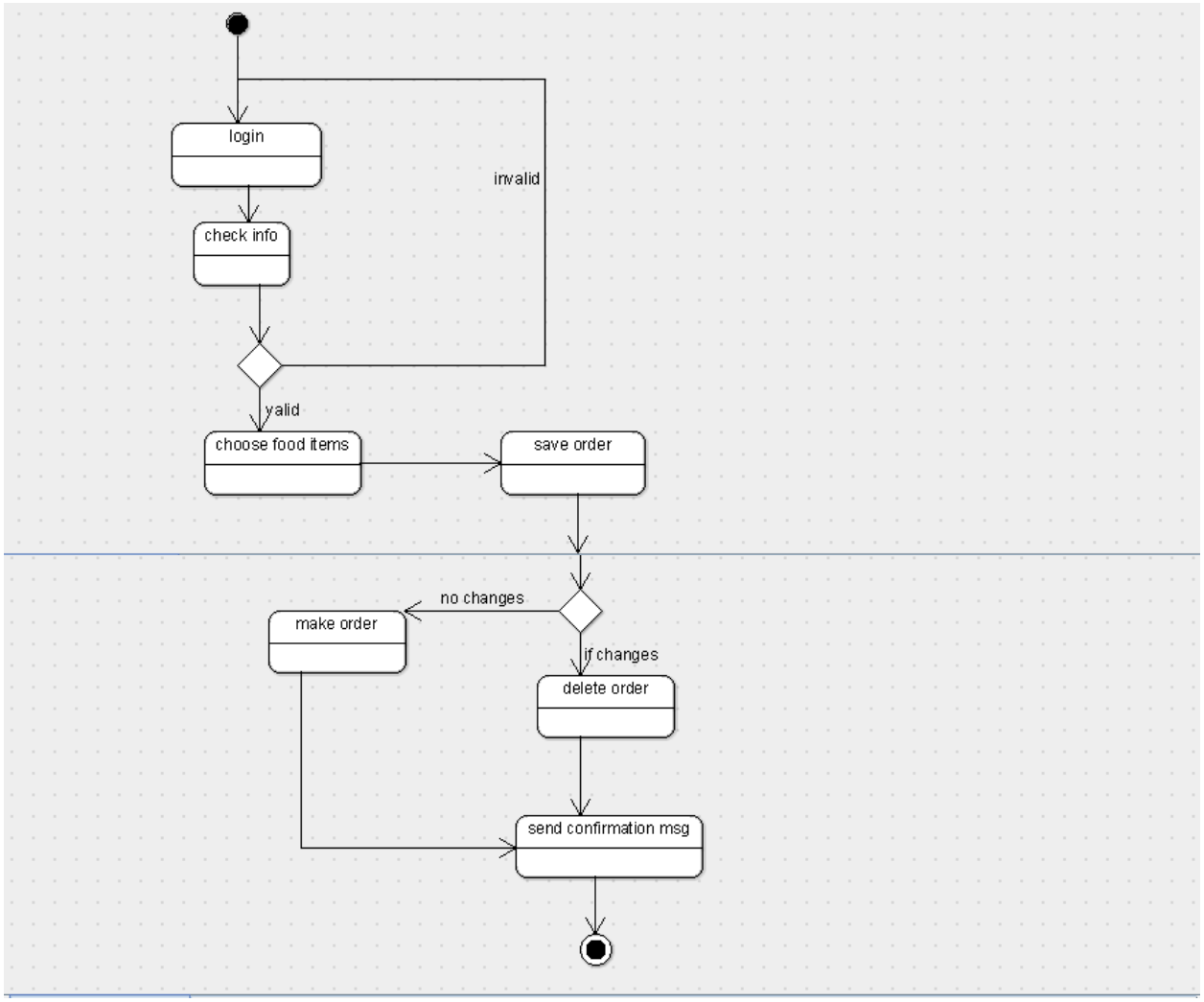


Fig: 3.2.5(b) state chart diagram for placing order.

State chart diagram for payment:

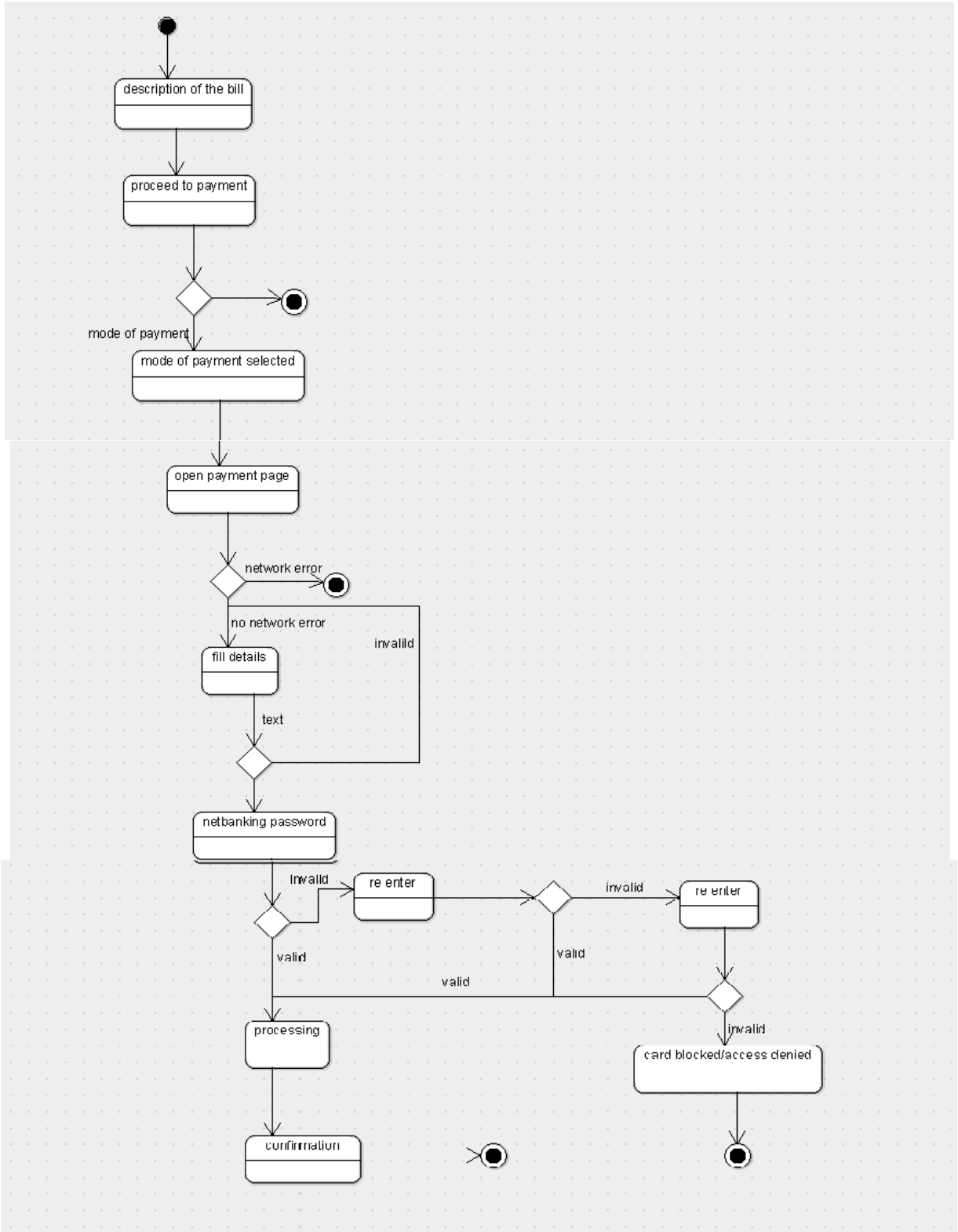


Fig: 3.2.5(c) state chart diagram for payment

3.2.6. Activity diagram

An activity diagram shows the flow from activity to activity. An activity is a going, on-atomic execution within a state machine. An activity results in some action, results in a change of state or return of a value.

Activity Diagram commonly contains

1. Activity states and action states.

2. Transition.

3. Objects, it may contain nodes and constraints.

Activity states and action states: An executable atomic computation is called action state, which cannot be decomposed. Activity state is non-atomic, decomposable and takes some duration to execute.

Transition: It is the path from one state to the next state, represented as simple directed line.

Branching: When an alternate path exists, branching arises which is represented by diamond. It has and incoming transition, two or more outgoing transitions.

Forking and Joining: The synchronization bar when split one flow into two or more flows is called fork. When two or more flows are combined at synchronization bar, the bar is called join.

Swim Lanes: Group work flow is called swim lanes. All groups are portioned by vertical solid lines. Each swim lane specifies locus of activities and has a unique name. Each swim lane is implemented by one or more classes. Transition may occur between objects across swim lanes.

Activity diagram for placing order:

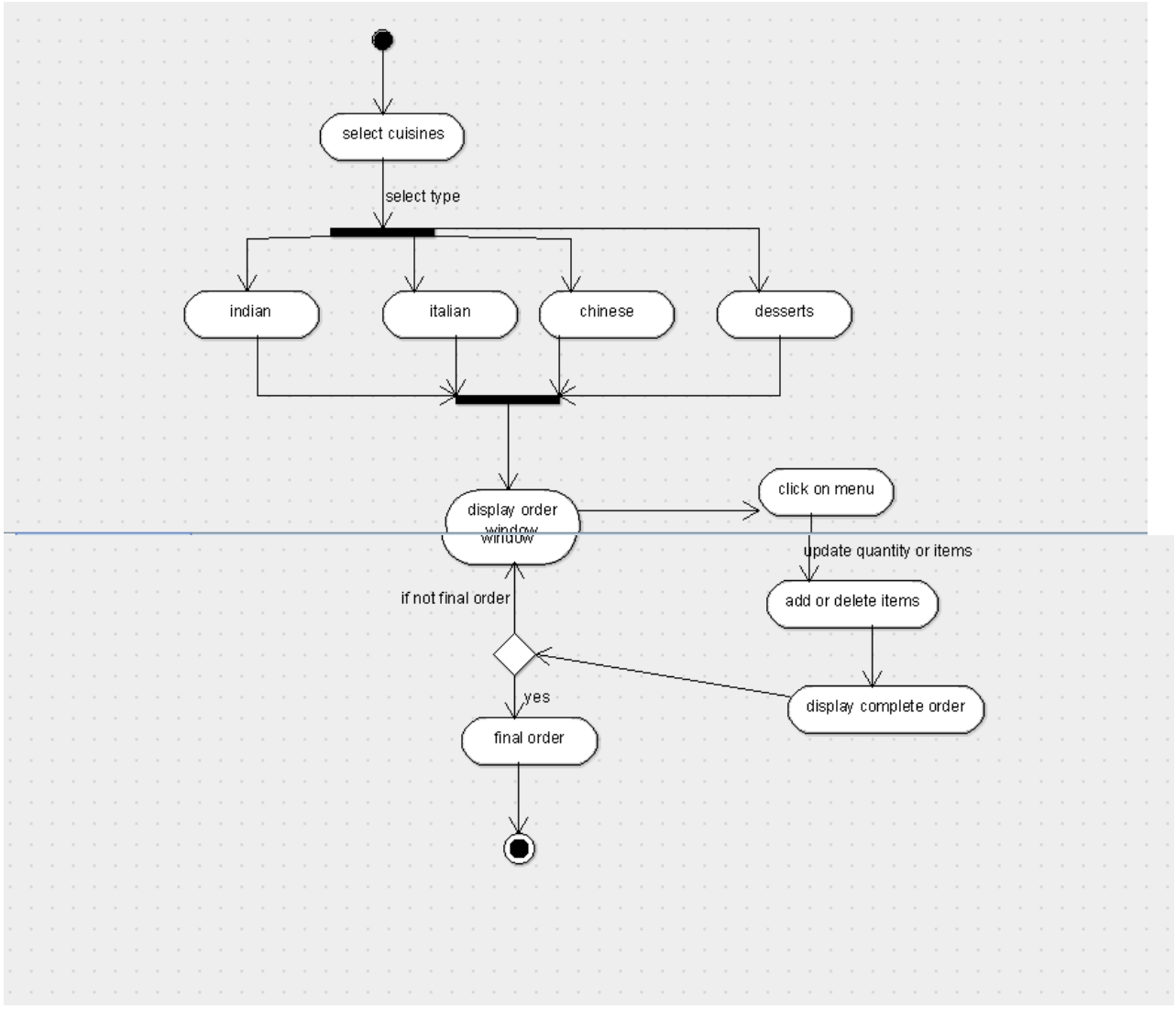


Fig: 3.2.6(a) Activity diagram for placing order

Activity diagram for update menu:

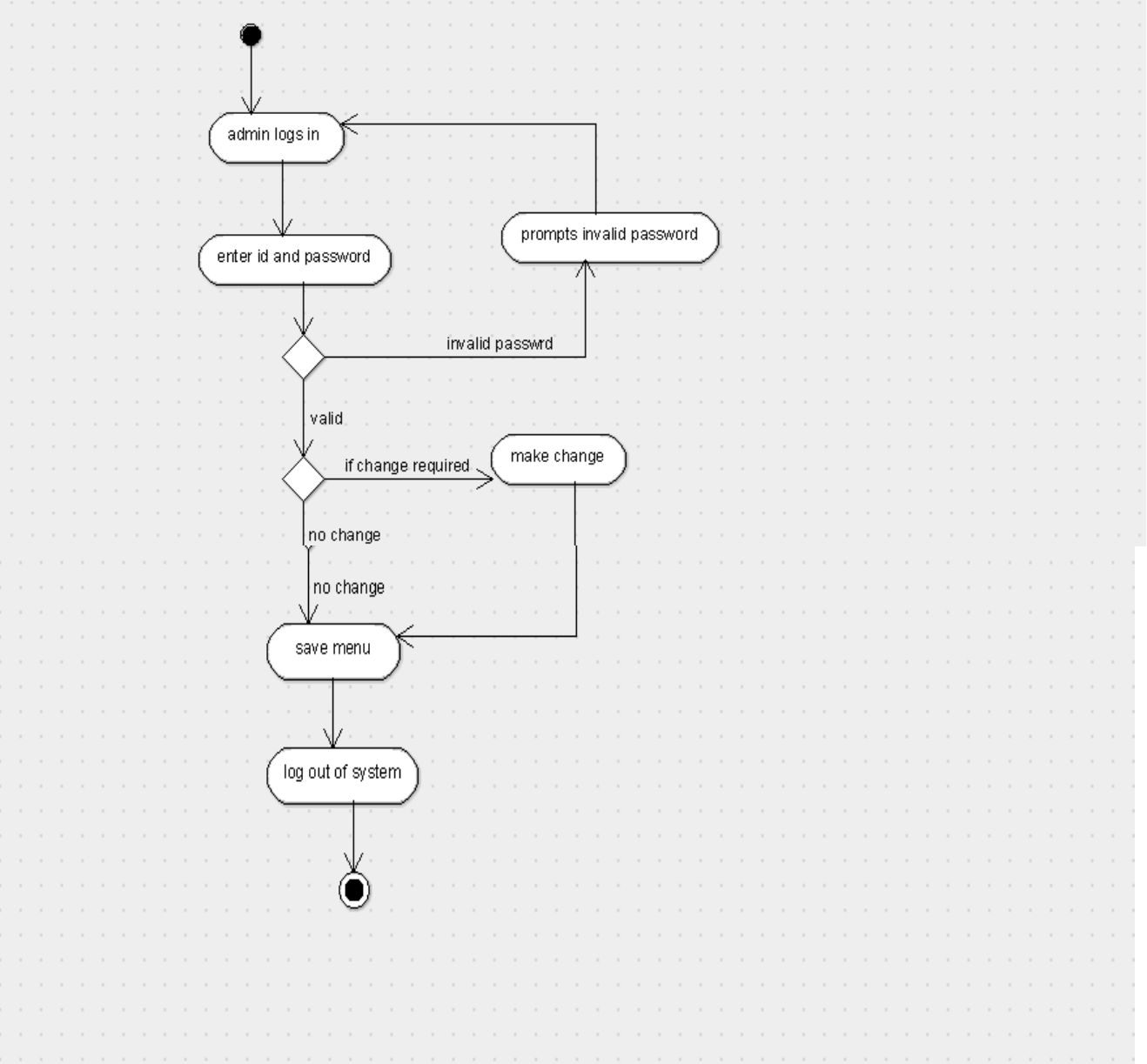


Fig: 3.2.6(b) Activity diagram for update menu

Activity diagram for payment:

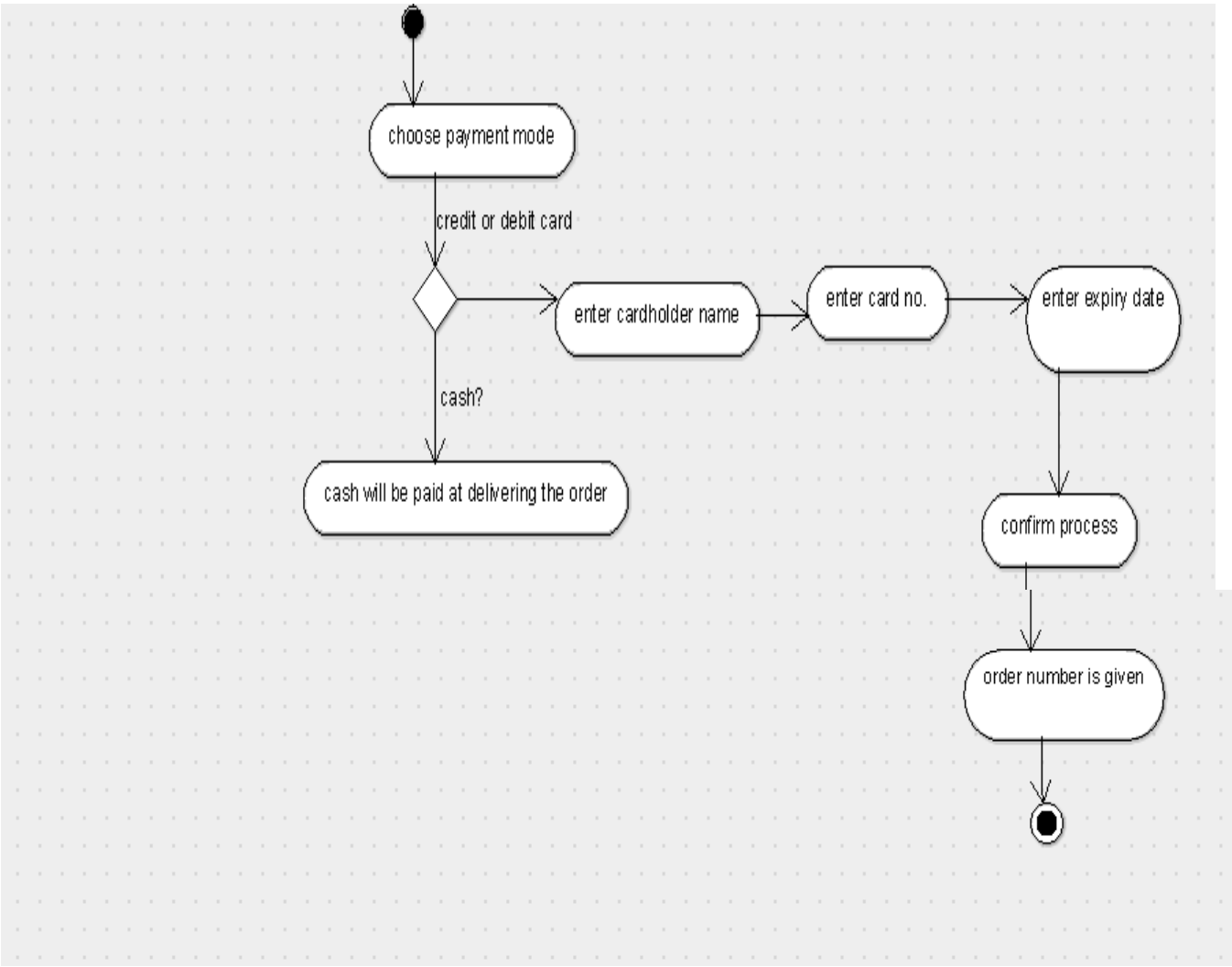


Fig: 3.2.6(c) Activity diagram for payment

3.2.7 Component diagram:

It describes the organization and wiring of the physical components in a system. The diagrams can be presented to key project state holders and implementation staff.
Component diagram can be used to model
A) The components of a system
B) The database schema
C) The executable of an application
D) A system's source code

Notation is rectangle with two smaller rectangles extending out to the boundary from its left side.

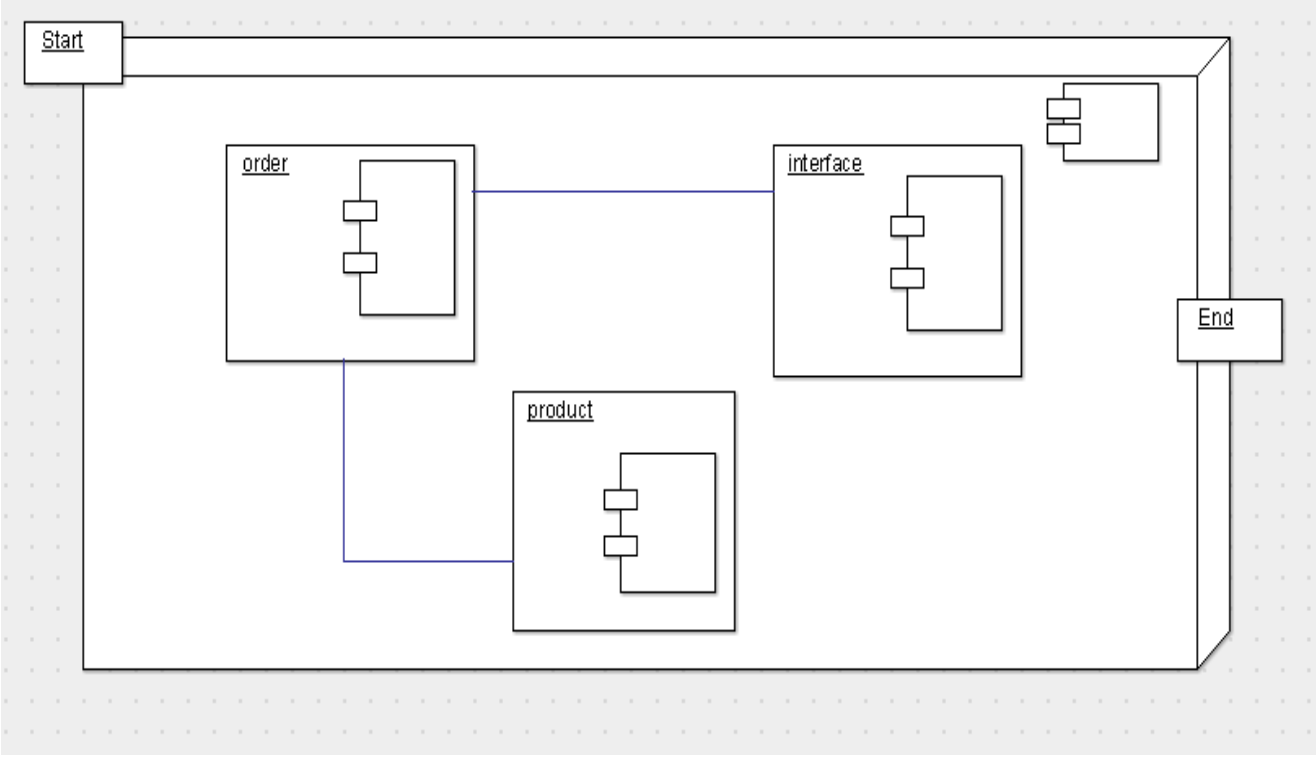


Fig: 3.2.7 Component diagram

3.2.8 Deployment diagram:

Deployment:

It is used to visualize the topology of the physical component of a system where the software components are deployed. It consists of nodes and their relationships.

Component diagrams are used to describe the components and the deployment diagrams show how they are deployed in hardware. These are useful for system engineers.

An efficient deployment diagram is very important as it controls the performance, maintainability and portability.

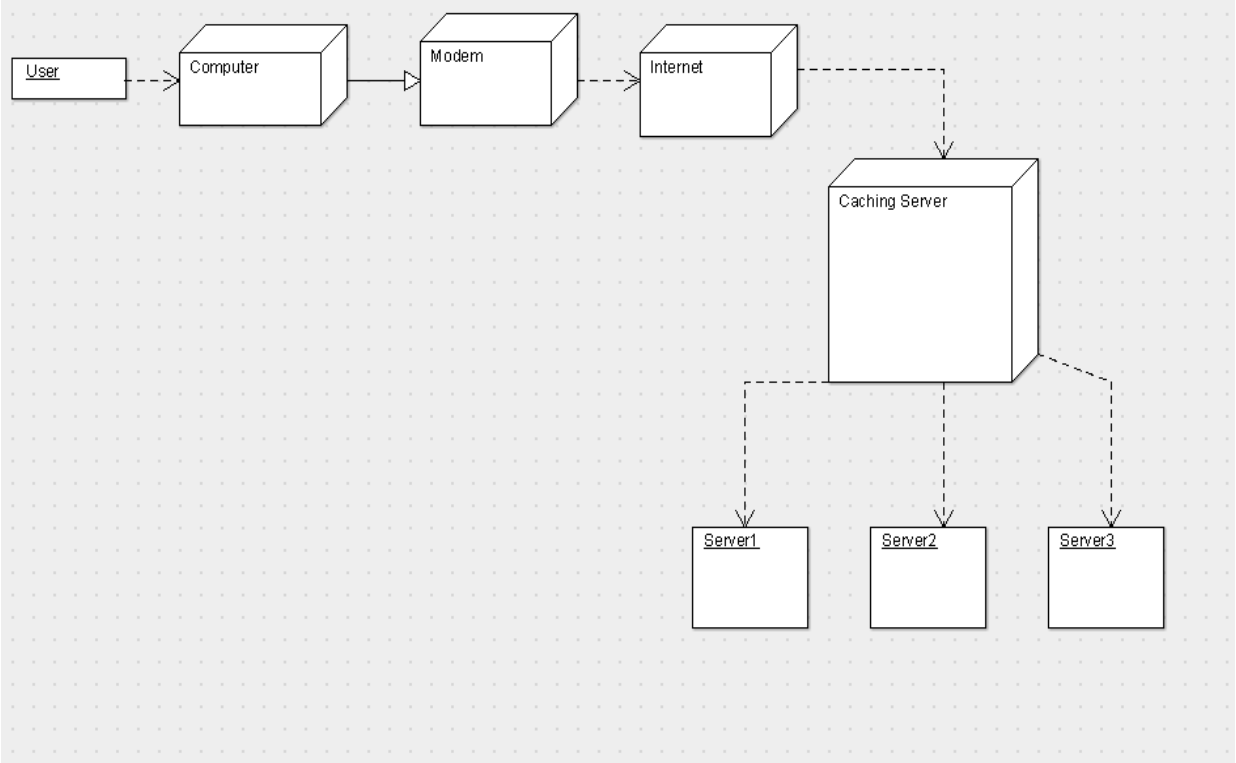


Fig: 3.2.8 Deployment diagram

4. Database Design

ER Diagram:

An Entity-relationship model describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram. An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.ER diagram shows the complete logical structure of database.

In ER Diagram:

- *Rectangle*: Represents Entity sets
- *Ellipses*: Attributes
- *Diamonds*: Relationship set
- *Lines*: They link attributes to entity sets and entity sets to relationship set
- *Double Ellipses*: Multi valued Attributes
- *Dashed Ellipses*: Derived Attributes
- *Double Rectangles*: Weak Entity Sets
- *Double Lines*: Total participation of an entity in relationship set

ER diagram has three main components:

1.Entity

2.Attribute

3.Relationship

Entity: An entity is an object or component of data.

Weak Entity: An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity.

Attribute: An attribute describes the property of an entity.

There are four types:

1. Key attribute: A key attribute can uniquely identify an entity from an entity set.

2. Composite attribute: An attribute that is a combination of other attribues is known as composite attribute.

3. Multi-valued attribute: An attribute that can hold multiple values is known as multi-valued attribute.

4. Derived attribute: A derived attribute is one whose value is dynamic and derived from another attribute.

Relationship: It shows the relationship among entities. There are four types of relationships:

1. *One to One*: When a single instance of an entity is associated with a single instance of another entity is called one to one relationship.

2. *One to Many*: When a single instance of an entity is associated with more than one instances of another entity is called one to many relationship.

3. *Many to One*: When more than one instances of an entity is associated with a single instance of another entity is called many to one relationship.

4. *Many to many*: When more than one instances of an entity is associated with more than one instances of another entity is called many to many relationship.

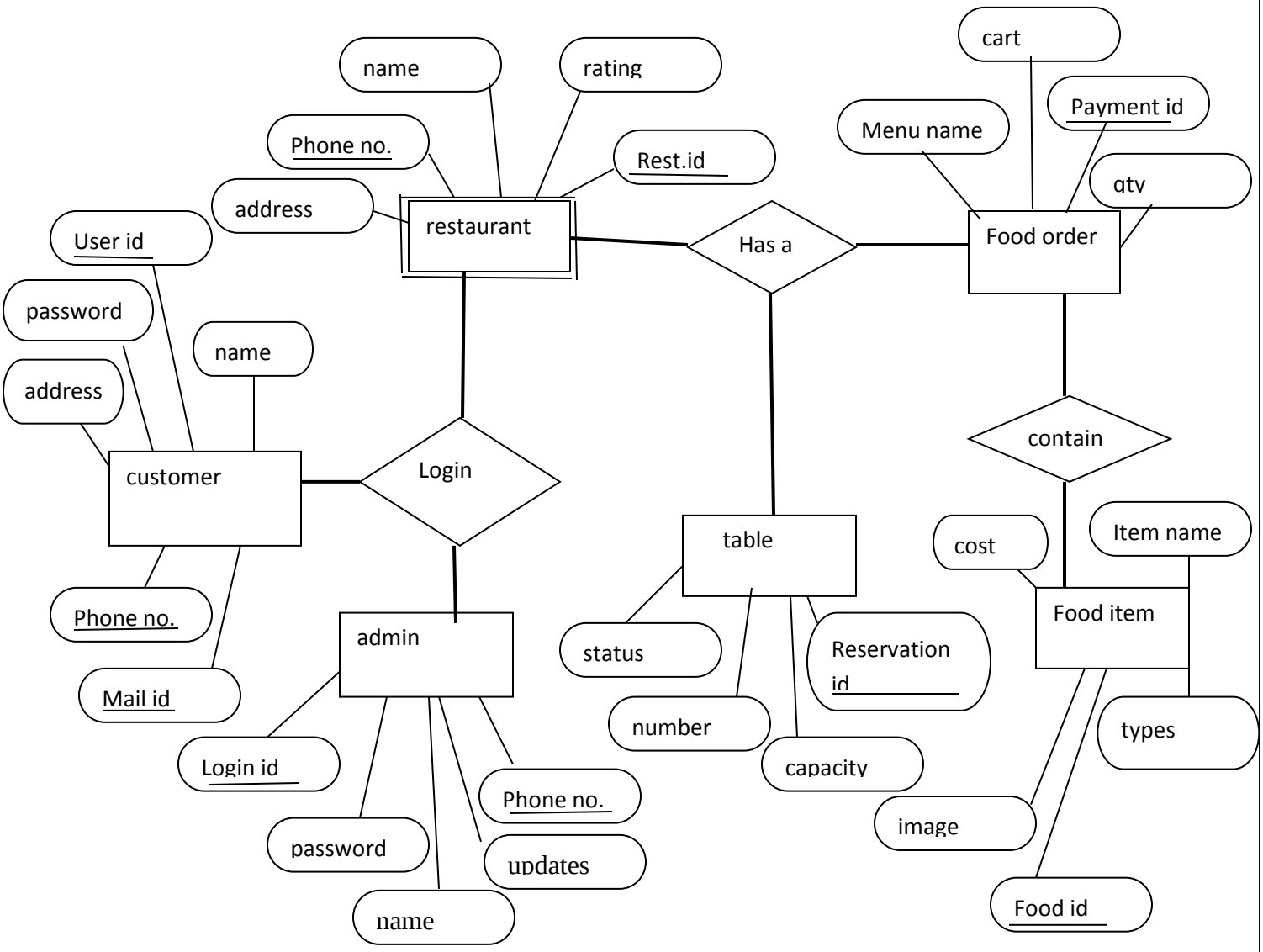


Fig: 4. ER Diagram

Databases:

```
| Tables_in_food-ordering |
+-------------------------+
| orders                  |
| orders_items            |
| products                |
| rating                  |
| show_orders             |
| table_booking           |
| user_data               |
+-------------------------+
7 rows in set (0.00 sec)

mysql> select * from user_data;
+-----+----------+------------+------------+---------------------------+--------------------------------------------------------------+------------------------------+
| sno | firstname | lastname   | phone      | email                     | pass                                                         | address_                     |
+-----+----------+------------+------------+---------------------------+--------------------------------------------------------------+------------------------------+
|  12 | ADMIN    | ADMIN      |          0 | narensaikrishna@gmail.com | admin                                                        | NULL                         |
|  18 | Hello    | kumar      | 7416047302 | td5e4q456@gmail.com       | $2y$10$Vp9zEUL0TEfYd09h3TZpHuk3gUwoQTR7cfxq8LQs2STW.6lnMCceu | Flat no. 401, Raghavendra Apartm |
|   1 | Naren    | Sai Krishna | 7416047416 | narensaikrishna@gmail.com | $2y$10$CnhliExErnDqfvdBxuP7Y.EEZuApKfavcZyboaicBkRJvWXxCBkDu | Anits College, Vizag         |
|  17 | Naren    | Naren      | 7416047430 | naren@naren.com           | $2y$10$myD29KJZjPEJMmS/yNN44.fUhsNgia9G2ANF4907YLfTbTTPcGVeu | Flat no. 401, Raghavendra Apartm |
|  10 | madhu    | kumar      | 8297389668 | td5e4q456@gmail.com       | $2y$10$.WCkEcFnORtsRSrBnTQslupGtuRFfQSfBBbHd9D/1Vw7.cwD7pRdK | t6rwe56ud4e56a               |
|   8 | manjari  | manju      | 8317522217 | manjari3993@gmail.com     | $2y$10$OUTwkyb42/ePsL.6w0PTQOsESMsF7DMBedasYFVqjHv8GOKEsFr5O | mmm                          |
|  15 | admin    | admin      | 9000000000 | relish@relish.com         | $2y$10$oOdrsDhS6XTEcoz8Sjc2De7t7WxGdTUVVPE9ZZvD4PO6YvdXLevPa | Relish                       |
|   9 | haritha  | kunisetti  | 9010434230 | haritha.kunisetti@gmail.com | $2y$10$gTdu8lDsvsKTA9rgmTn1jOEFt2IQkilu70r1.WiCPva2p.mO.YJCi | paderu                       |
|  14 | ananan   | jfdcjxxc   | 9988112233 | td5e4q456@gmail.com       | $2y$10$3XU6nlqKXHKRzLyTkTVwOQDeWrCKcwnWBAkcF2TsLZhScVRNLX1W | Flat no. 401, Raghavendra Apartm |
|  13 | Naren    | Hello      | 9988998891 | td5e4q456@gmail.com       | $2y$10$ai0FfmWVVTgqLvNHFiKLuDM0JfCWD89tWGq2cupchBAFo1ALPWLy | Flat no. 401, Raghavendra Apartm |
|  16 | ADMIN    | ADMIN      | 9999999999 | relish@relish.com         | $2y$10$5SfFa9OhDTHp/6bYHcmWxue3PvWmDOLUXHjB6PZB6U/mZJPt9yHmG | Visakhapatnam                |
+-----+----------+------------+------------+---------------------------+--------------------------------------------------------------+------------------------------+

mysql> select * from table_booking;
+----+------------+----------------+-------------+---------------------+-----------------------+-----------------------+-----------------------+-----------------------+-------+
| id | phone      | date_of_booking | time_       | created_date        | f1t1                  | f1t2                  | f1t3                  | f1t4                  | f1t5  |
+----+------------+----------------+-------------+---------------------+-----------------------+-----------------------+-----------------------+-----------------------+-------+
| 19 | 9999999999 | 2019-03-16     | 8 A.M - 9 A.M | 2019-03-11 20:15:08 | images/available.png | images/reserved.png  | images/reserved.png  | images/available.png | image |
| 20 | 9999999999 | 2019-03-17     | 8 A.M - 9 A.M | 2019-03-11 20:16:52 | images/available.png | images/available.png | images/reserved.png  | images/available.png | image |
| 21 | 9999999999 | 2019-03-14     | 8 A.M - 9 A.M | 2019-03-12 17:28:08 | images/available.png | images/available.png | images/reserved.png  | images/available.png | image |
| 22 | 9999999999 | 2019-03-11     | 8 A.M - 9 A.M | 2019-03-11 20:28:54 | images/available.png | images/available.png | images/reserved.png  | images/available.png | image |
| 23 | 7416047416 | 2019-03-27     | 8 A.M - 9 A.M | 2019-03-12 19:09:47 | images/available.png | images/available.png | images/reserved.png  | images/available.png | image |
+----+------------+----------------+-------------+---------------------+-----------------------+-----------------------+-----------------------+-----------------------+-------+
5 rows in set (0.01 sec)

mysql> select * from show_orders;
+----------+------------+------------+---------------------+---------------+----------+-------+
| order_id | order_name | product_id | product_name        | product_price | quantity | total |
+----------+------------+------------+---------------------+---------------+----------+-------+
|        1 | joshua     |          3 | BUTTER NAAN         |            90 |        2 |   180 |
|        1 | joshua     |          8 | SPECIAL FRIED RICE  |           150 |        1 |   150 |
|        1 | joshua     |          9 | VEGETABLE BIRYANI   |           150 |        3 |   450 |
|        1 | joshua     |         12 | MUTTON BIRYANI      |           230 |        1 |   230 |
|        1 | joshua     |         15 | VEGETABLE SOUP      |            85 |        1 |    85 |
|        1 | joshua     |         24 | CAESAR SALAD        |           170 |        1 |   170 |
|        1 | joshua     |          2 | PLAIN NAAN          |            70 |        1 |    70 |
|        2 | haritha    |          2 | PLAIN NAAN          |            70 |        1 |    70 |
|        2 | haritha    |         32 | GARLIC PIZZA        |           290 |        1 |   290 |
|        2 | haritha    |          3 | BUTTER NAAN         |            90 |        1 |    90 |
|        2 | haritha    |         33 | GRILLED SALMON      |           340 |        1 |   340 |
|        2 | haritha    |          7 | VEGETABLE FRIED RICE |           110 |        1 |   110 |
|        2 | haritha    |         34 | CRAB ALFREDO        |           380 |        1 |   380 |
|        2 | haritha    |         11 | CHICKEN BIRYANI     |           190 |        1 |   190 |
|        2 | haritha    |         35 | SHRIMP ALFREDO      |           290 |        1 |   290 |
|        2 | haritha    |         14 | CHICKEN RICE SOUP   |           100 |        1 |   100 |
|        2 | haritha    |         37 | FRUIT PLATTER       |           250 |        1 |   250 |
```

```
mysql> select * from rating;
+------------+-------+-------------+---------------+-----------------+
| phone      | taste | foodquality | intimedelivery | customerservice |
+------------+-------+-------------+---------------+-----------------+
| 7416047416 |   4.5 |         2.5 |           1.5 |             1.5 |
| 8317522217 |   4.5 |         4.5 |             5 |             4.5 |
| 9010434230 |   0.5 |           1 |             1 |               1 |
| 8297389668 |     5 |         4.5 |           4.5 |               5 |
|          0 |   3.5 |           2 |             2 |               5 |
| 9999999999 |   0.5 |         0.5 |             1 |             0.5 |
+------------+-------+-------------+---------------+-----------------+
6 rows in set (0.01 sec)

mysql> select * from products;
+------------+----------------------+-----------------------+---------------------+---------------+
| product_id | product_name         | product_image         | product_description | product_price |
+------------+----------------------+-----------------------+---------------------+---------------+
|          1 | GARLIC NAAN          | garlicnaan.jpg        |                     |         50.00 |
|          2 | PLAIN NAAN           | plainnaan.jpg         |                     |         70.00 |
|          3 | BUTTER NAAN          | butternaan.jpg        |                     |         90.00 |
|          4 | MEALS                | meals.jpg             |                     |        130.00 |
|          5 | CHICKEN FRIED RICE   | chickenfriedrice.jpg  |                     |        170.00 |
|          6 | EGG FRIED RICE       | eggfriedrice.jpg      |                     |        150.00 |
|          7 | VEGETABLE FRIED RICE | vegfriedrice.jpg      |                     |        110.00 |
|          8 | SPECIAL FRIED RICE   | specialfriedrice.jpg  |                     |        150.00 |
|          9 | VEGETABLE BIRYANI    | vegbiryani.jpg        |                     |        150.00 |
|         10 | EGG BIRYANI          | eggbiryani.jpg        |                     |        170.00 |
|         11 | CHICKEN BIRYANI      | chickenbiryani.jpg    |                     |        190.00 |
|         12 | MUTTON BIRYANI       | muttonbiryani.jpg     |                     |        230.00 |
|         13 | CHICKEN NOODLE SOUP  | chickennoodlesoup.jpg |                     |         90.00 |
|         14 | CHICKEN RICE SOUP    | chickenricesoup.jpg   |                     |        100.00 |
|         15 | VEGETABLE SOUP       | vegetablesoup.jpg     |                     |         85.00 |
|         16 | HOT & SOUR SOUP      | hot&soursoup.jpg      |                     |        110.00 |
|         17 | VEG CHOWMEIN         | vegchowmein.jpg       |                     |        200.00 |
|         18 | CHICKEN CHOWMEIN     | chickenchowmein.jpg   |                     |        170.00 |
|         19 | SHRIMP CHOWMEIN      | shrimpchowmein.jpg    |                     |        195.00 |
|         20 | CHICKEN NOODLES      | chickennoodles.jpg    |                     |         95.00 |
|         21 | VEG NOODLES          | vegnoodles.jpg        |                     |         75.00 |
|         22 | EGG NOODLES          | eggnoodles.jpg        |                     |         85.00 |
|         23 | TOSSED SALAD         | tossedsalad.jpg       |                     |        150.00 |
|         24 | CAESAR SALAD         | caesarsalad.jpg       |                     |        170.00 |
|         25 | CHICKEN SALAD        | chickensalad.jpg      |                     |        190.00 |
|         26 | GREEK SALAD          | greeksalad.jpg        |                     |        150.00 |
|         27 | STROMBOLI            | stromboli.jpg         |                     |        200.00 |
|         28 | MEATBALL             | meatballparmigiana.jpg|                     |        170.00 |
|         29 | CHICKEN PARMIGIANA   | chickenparmigiana.jpg |                     |        195.00 |
|         30 | CLASSIC ITALIAN      | classicitalian.jpg    |                     |        250.00 |
|         31 | MARGHARITA           | margharita.jpg        |                     |        270.00 |
|         32 | GARLIC PIZZA         | garlicpizza.jpg       |                     |        290.00 |
|         33 | GRILLED SALMON       | grilledsalmon.jpg     |                     |        340.00 |
|         34 | CRAB ALFREDO         | crabalfredo.jpg       |                     |        380.00 |
|         35 | SHRIMP ALFREDO       | shrimpalfredo.jpg     |                     |        290.00 |
|         36 | LOBSTER RAVIOLI      | lobsterravioli.jpg    |                     |        230.00 |
|         37 | FRUIT PLATTER        | fruitplatter.jpg      |                     |        250.00 |
|         38 | MANGO JUBLIEE        | mangojubliee.jpg      |                     |        150.00 |
|         39 | ICE CREAM            | icecream.jpg          |                     |         90.00 |
|         40 | FRIED ICE CREAM      | friedicecream.jpg     |                     |        125.00 |
|         41 | BANANA SPLIT         | bananasplit.jpg       |                     |        180.00 |
|         42 | PANNA COTTA          | pannacotta.jpg        |                     |         95.00 |
|         43 | ICEBOX CAKE          | iceboxcake.jpg        |                     |         80.00 |
|         44 | COAST LIME PIE       | coastlimepie.jpg      |                     |        250.00 |
|         45 | CHOCOLATE CAKE       | chocolatecake.jpg     |                     |        220.00 |
+------------+----------------------+-----------------------+---------------------+---------------+
45 rows in set (0.01 sec)
```

```
mysql> select * from orders;
+----------+---------------------+------------+-----------------------------+
| order_id | order_date          | order_name | order_email                 |
+----------+---------------------+------------+-----------------------------+
|        1 | 2019-02-22 11:09:41 | joshua     | joshua.cse@anits.edu.in     |
|        2 | 2019-02-22 11:25:23 | haritha    | haritha.kunisetti@gmail.com |
|        3 | 2019-02-23 21:51:16 | naren      | naren@naren.com             |
|        4 | 2019-02-28 11:04:05 | fv         | narensaikrishna@gmail.com   |
|        5 | 2019-03-05 13:06:56 | Naren      | narensaikrishna@gmail.com   |
|        6 | 2019-03-07 22:45:46 | Naren      | narensaikrishna@gmail.com   |
|        7 | 2019-03-07 22:47:28 | n          | narensaikrishna@gmail.com   |
|        8 | 2019-03-07 22:50:41 | naren      | naren@naren.com             |
|        9 | 2019-03-07 22:53:13 | n          | Hello@hello.com             |
|       10 | 2019-03-07 22:54:05 | Naren      | narensaikrishna@gmail.com   |
|       11 | 2019-03-08 09:45:45 | Naren      | narensaikrishna@gmail.com   |
|       12 | 2019-03-08 09:47:26 | Naren      | narensaikrishna@gmail.com   |
|       13 | 2019-03-08 10:25:39 | Naren      | narensaikrishna@gmail.com   |
|       14 | 2019-03-08 10:28:39 | Naren      | naren@naren.com             |
|       15 | 2019-03-08 10:29:10 | a          | narensaikrishna@gmail.com   |
|       16 | 2019-03-08 10:34:54 | naren      | naren@naren.com             |
|       17 | 2019-03-08 10:35:58 | naren      | narensaikrishna@gmail.com   |
|       18 | 2019-03-08 11:29:46 | joshua     | joshua.sirasapalli@gmail.com |
|       19 | 2019-03-11 20:23:14 | n          | nn@nn.com                   |
+----------+---------------------+------------+-----------------------------+
```

```
mysql> select * from orders_items;
+----------+------------+----------+
| order_id | product_id | quantity |
+----------+------------+----------+
|        1 |          2 |        1 |
|        1 |          3 |        2 |
|        1 |          8 |        1 |
|        1 |          9 |        3 |
|        1 |         12 |        1 |
|        1 |         15 |        1 |
|        1 |         24 |        1 |
|        2 |          1 |        2 |
|        2 |          2 |        1 |
|        2 |          3 |        1 |
|        2 |          7 |        1 |
|        2 |         11 |        1 |
|        2 |         14 |        1 |
|        2 |         18 |        1 |
|        2 |         23 |        1 |
```

5. Implementation

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the system for the users that will work efficiently and effectively. The system will be implemented only after thorough testing and if it is found to work according to the specification.

Wamp Server:

Wamp Server refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySql database and PHP programming language. Wamp Server automatically installs everything you need to intuitively develop Web applications. You will be able to tune your server without even touching its setting files. Best of all, Wamp Server is available for free (under GPML license) in both 32 and 64 bit versions. Wamp server is not compatible with Windows XP, SP3, or Windows Server 2003.

PHP :

PHP: Hypertext Pre-processor (or simply PHP) is a general-purpose programming Language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994 the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive Initialism PHP: Hypertext Pre-processor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and Executed PHP code, which may be any type of data, including images, with the generated Web page. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control

HTML:

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items

CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript
CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Java script:

JavaScript often abbreviated as JS, is a high-level, interpreted programming language that confirms to the ECMA Script specification. It is a programming language that is characterized as dynamic, weakly typed, prototype-based and multi-paradigm.
Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications.

The vast majority of websites use it, [10] and major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object

oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

MYSQL:

MYSQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MYSQL is developed, marketed and supported by MYSQL AB, which is a Swedish company. MYSQL is becoming so popular because of many good reasons-

1. MySQL is released under an open source license. So you have nothing to pay to use it.
2. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
3. MySQL uses a standard form of the well-known SQL, data language.
4. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++,JAVA etc.
5. MySQL works very quickly and works well even with large data sets.
6. MySQL is very friendly to PHP, the most appreciated language for web development.
7. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments

CODE FOR TABLE RESERVATION:

```
<?php
require('php/db.php');
session_start();
if(!isset($_SESSION['sfirstname'])){
	header('Location: homepage_before_login.php');
}
if(!isset($_SESSION['date1'])){
	header('Location: tablereserve1.php');
}
$date=$_SESSION['date1'];
$time=$_SESSION['time1'];
$queryRole = "SELECT * FROM table_booking WHERE date_of_booking='$date' and time_='$time'";
$result1=mysqli_query($con,$queryRole) or die(mysqli_error());
$rows1 = mysqli_num_rows($result1);
$row1=mysqli_fetch_assoc($result1);
if(!isset($_SESSION['sfirstname'])){
	header('Location: homepage_before_login.php');
}
if(isset($_POST['submit']))
{
	// function checkImage($id){
	//	if($id=='')
	//	$q="Update table_booking set f1t1='images/selected.png' where date_of_booking='$date' and time_='$time'";
	//	$qq=mysqli_query($con,$q);

	// }
	$f1t1=$_POST['f1t1'];$f1t2=$_POST['f1t2'];$f1t3=$_POST['f1t3'];$f1t4=$_POST['f1t4'];$f1t5=$_POST['f1t5'];$f1t6=$_POST['f1t6'];$f1t7=$_POST['f1t7'];$f1t8=$_POST['f1t8'];$f1t9=$_POST['f1t9'];
	$f2t1=$_POST['f2t1'];$f2t2=$_POST['f2t2'];$f2t3=$_POST['f2t3'];$f2t4=$_POST['f2t4'];$f2t5=$_POST['f2t5'];$f2t6=$_POST['f2t6'];$f2t7=$_POST['f2t7'];$f2t8=$_POST['f2t8'];$f2t9=$_POST['f2t9'];
	$f3t1=$_POST['f3t1'];$f3t2=$_POST['f3t2'];$f3t3=$_POST['f3t3'];$f3t4=$_POST['f3t4'];$f3t5=$_POST['f3t5'];$f3t6=$_POST['f3t6'];$f3t7=$_POST['f3t7'];$f3t8=$_POST['f3t8'];$f3t9=$_POST['f3t9'];
	$f4t1=$_POST['f4t1'];$f4t2=$_POST['f4t2'];$f4t3=$_POST['f4t3'];$f4t4=$_POST['f4t4'];$f4t5=$_POST['f4t5'];$f4t6=$_POST['f4t6'];$f4t7=$_POST['f4t7'];$f4t8=$_POST['f4t8'];$f4t9=$_POST['f4t9'];

	$update=mysqli_query($con,"UPDATE table_booking set f1t1='$f1t1', f1t2='$f1t2', f1t3='$f1t3', f1t4='$f1t4', f1t5='$f1t5', f1t6='$f1t6', f1t7='$f1t7', f1t8='$f1t8', f1t9='$f1t9',
```

```
        if($update)
                {
                        $_SESSION['beep']="1";
                        ?>
                <script>
                alert("Your table has been sucessfully booked!. Thank You");

                </script>
<?php
}
header("refresh:0; url=tablereserve.php");
}
?>

<!DOCTYPE html>

<html lang="en">

<head>
                <title>Reserve your Tables</title>
                <link rel="icon" href="images/Logo.png">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet">
                <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
        <style>
                *{

                                        margin: 0px;
                                        padding:0px;
                                        box-sizing: border-box;
                                        font-family: 'Flamenco';
                                        font-weight:bold;
                                        }
    .Logo{
        height: 130px;
        width: auto;
        float: left;
        margin-top: 20px;
        transition: 5s linear;
    }
                body{
```

```
background-image: linear-gradient(rgba(0, 0, 0, 0.8),rgba(0,0,0,0.8)), url('images/i2.jpeg');
            background-size: cover;
            background-repeat: no-repeat;
            background-position: center;
            height: 100vh;
                                        overflow: hidden;
                        }
        h4{
            font-family: 'Pacifico';
            color:white;

        }
        h1{
            font-family: 'Pacifico';
            color:white;

        }
        table{
            padding-left:5%;
            height:30%;
            width:40%;
            background-color:#0009;
            margin-bottom:2%;
            margin-left:5%;
            margin-right:5%;
            margin-top:8%;

            border-spacing:0px;
        }
        .btn {
  border-radius: 12px;
  border-color:purple;
  background-color: transparent;
  border: 2px solid;
  color: purple;
  padding: 16px 32px;
  text-align: center;
  font-size: 16px;
  margin: 1%;
  position: absolute;
  opacity: 1;
  transition: 0.3s;
  z-index:2;

}
```

```
.booking-form {
	position: relative;
	max-width: 500px;
	width: 100%;
	margin: 0px auto;
	padding: 15px;
	overflow: hidden;
	background-color: transparent;
}


.booking-form .form-header h1 {
	font-weight: 700;
	text-transform: capitalize;
	font-size: 42px;
	margin: 0px;
	color: #fff;

}

.booking-form .form-group {
	position: relative;
	margin-bottom: 30px;
}

.booking-form .form-control {
	background-color: white;
	height: 60px;
	padding: 0px 25px;
	border: none;
	border-radius: 40px;
	color: black;
	-webkit-box-shadow: 0px 0px 0px 2px transparent;
	box-shadow: 0px 0px 0px 2px transparent;
	-webkit-transition: 0.2s;
	transition: 0.2s;
}

.booking-form .form-control::-webkit-input-placeholder {
	color: black;
}
```

```
.booking-form .form-control:-ms-input-placeholder {
	color: black;
}

.booking-form .form-control::placeholder {
	color: black;
}
input:focus,
select:focus,
textarea:focus,
button:focus {
   outline: none;
}
.booking-form .form-control:focus {
	-webkit-box-shadow: 0px 0px 0px 2px #ff8846;
	box-shadow: 0px 0px 0px 2px #ff8846;
}

.booking-form input[type="date"].form-control {
	padding-top: 16px;
}

.booking-form input[type="date"].form-control:invalid {
	color: black;
}

.booking-form input[type="date"].form-control+.form-label {
	opacity: 1;
	top: 10px;
}

.booking-form select.form-control {
	-webkit-appearance: none;
	-moz-appearance: none;
	appearance: none;
}

.booking-form select.form-control:invalid {
	color: black;
}

.booking-form select.form-control+.select-arrow {
	position: absolute;
	right: 15px;
	top: 50%;
	-webkit-transform: translateY(-50%);
```

```
        transform: translateY(-50%);
        width: 32px;
        line-height: 32px;
        height: 32px;
        text-align: center;
        pointer-events: none;
        color: rgba(255, 255, 255, 0.5);

        font-size: 14px;
        text-decoration: none;
}



.booking-form select.form-control option {
        color: #000;
        font-size:20px;

}

.booking-form .form-label {
        position: absolute;
        top: -10px;
        left: 25px;
        opacity: 0;
        color: #ff8846;

        font-size: 13px;
        text-decoration: none;
        font-weight: 700;
        text-transform: uppercase;
        letter-spacing: 1.3px;
        height: 15px;
        line-height: 15px;
        -webkit-transition: 0.2s all;
        transition: 0.2s all;
        opacity: 1;
        top: 10px;


}

.booking-form .form-group.input-not-empty .form-control {
        padding-top: 0px;

}
```

```
.booking-form .form-group.input-not-empty .form-label {
	opacity: 1;
	top: 10px;
}
.submit-btn {

	color: #fff;
	background-color: #e35e0a;
	font-weight: 700;
	height: 60px;
	padding: 10px 30px;
	width: 100%;
	border-radius: 40px;
	border: none;
	text-transform: uppercase;
	font-size: 16px;
	letter-spacing: 1.3px;
	-webkit-transition: 0.2s all;
	transition: 0.2s all;
}
.form-inline {
  display: flex;
  flex-flow: row wrap;
  align-items: center;
}

.booking-form .submit-btn:hover,
.booking-form .submit-btn:focus {
	opacity: 0.9;

}
   </style>
</head>

<body>
<button class="btn" style="position:absolute" onclick="goBack()">Back</button>


<div class="info" style="position:absolute;top:2%;right:5%">
   <img src="images/reserved.png" alt="table">   
   <img src="images/selected.png" alt="table">   
   <img src="images/available.png" alt="table">
</div>

<div class="info" style="color:white;position:absolute;top:10%;right:5%">
```

```
   <span>Reserved</span>   
   <span>Selected</span>   
   <span>Available</span>
</div>


<form method="post" class="form-inline" >


<div class="rest">
<div class="form-btn" style="margin:20px;">
<input type="submit" name="submit" value="Book" style="margin-left:500%;width:100%;padding-left:30px;padding-right:30px" class="submit-btn">
</div>

<img src="images/Logo.png" onload="Func()" style="margin-top:150%;margin-left:375%"class="Logo" id="Logo" alt="LOGO">
</div>

<table style="width:40%;position:absolute;top:0;left:0;">
<caption> <h1>Floor 1</h1></caption>

  <tr>
        <td><img src="<?php echo $row1['f1t1']?>" class="table-logo" alt="LOGO" id="f1t1" onclick="change(id)">
        <input hidden type="text" name="f1t1" id="f1t11" value="<?php echo $row1['f1t1']?>"> </td>
   <td><img src="<?php echo $row1['f1t2']?>" class="table-logo" alt="LOGO" id="f1t2" onclick="change(id)">
        <input type="hidden" name="f1t2" id="f1t21" value="<?php echo $row1['f1t2']?>">
</td>
   <td><img src="<?php echo $row1['f1t3']?>" class="table-logo" alt="LOGO" id="f1t3" onclick="change(id)">
        <input type="hidden" name="f1t3" id="f1t31" value="<?php echo $row1['f1t3']?>">
</td>
  </tr>
  <tr>
   <td><img src="<?php echo $row1['f1t4']?>" class="table-logo" alt="LOGO" id="f1t4" onclick="change(id)">
        <input type="hidden" name="f1t4" id="f1t41" value="<?php echo $row1['f1t4']?>">
</td>
   <td><img src="<?php echo $row1['f1t5']?>" class="table-logo" alt="LOGO" id="f1t5" onclick="change(id)">
        <input type="hidden" name="f1t5" id="f1t51" value="<?php echo $row1['f1t5']?>">
</td>
```

```
  <td><img src="<?php echo $row1['f1t6']?>" class="table-logo" alt="LOGO" id="f1t6" onclick="change(id)">
        <input type="hidden" name="f1t6" id="f1t61" value="<?php echo $row1['f1t6']?>">
</td>
 </tr>
 <tr>
        <td><img src="<?php echo $row1['f1t7']?>" class="table-logo" alt="LOGO" id="f1t7" onclick="change(id)">
        <input type="hidden" name="f1t7" id="f1t71" value="<?php echo $row1['f1t7']?>">
</td>
  <td><img src="<?php echo $row1['f1t8']?>" class="table-logo" alt="LOGO" id="f1t8" onclick="change(id)">
        <input type="hidden" name="f1t8" id="f1t81" value="<?php echo $row1['f1t8']?>">
</td>
  <td><img src="<?php echo $row1['f1t9']?>" class="table-logo" alt="LOGO" id="f1t9" onclick="change(id)">
        <input type="hidden" name="f1t9" id="f1t91" value="<?php echo $row1['f1t9']?>">
</td>
 </tr>
</table>



<table style="width:40%;position:absolute;bottom:0;left:0;">
<caption> <h1>Floor 3</h1></caption>

 <tr>
  <td><img src="<?php echo $row1['f3t1']?>" class="table-logo" alt="LOGO" id="f3t1" onclick="change(id)">
        <input type="hidden" name="f3t1" id="f3t11" value="<?php echo $row1['f3t1']?>">
</td>
  <td><img src="<?php echo $row1['f3t2']?>" class="table-logo" alt="LOGO" id="f3t2" onclick="change(id)">
        <input type="hidden" name="f3t2" id="f3t21" value="<?php echo $row1['f3t2']?>">
</td>
  <td><img src="<?php echo $row1['f3t3']?>" class="table-logo" alt="LOGO" id="f3t3" onclick="change(id)">
        <input type="hidden" name="f3t3" id="f3t31" value="<?php echo $row1['f3t3']?>">
</td>
 </tr>
 <tr>
  <td><img src="<?php echo $row1['f3t4']?>" class="table-logo" alt="LOGO" id="f3t4" onclick="change(id)">
        <input type="hidden" name="f3t4" id="f3t41" value="<?php echo $row1['f3t4']?>">
</td>
```

```
  <td><img src="<?php echo $row1['f3t5']?>" class="table-logo" alt="LOGO" id="f3t5" onclick="change(id)">
        <input type="hidden" name="f3t5" id="f3t51" value="<?php echo $row1['f3t5']?>">
</td>
  <td><img src="<?php echo $row1['f3t6']?>" class="table-logo" alt="LOGO" id="f3t6" onclick="change(id)">
        <input type="hidden" name="f3t6" id="f3t61" value="<?php echo $row1['f3t6']?>">
</td>
 </tr>
 <tr>
  <td><img src="<?php echo $row1['f3t7']?>" class="table-logo" alt="LOGO" id="f3t7" onclick="change(id)">
        <input type="hidden" name="f3t7" id="f3t71" value="<?php echo $row1['f3t7']?>">
</td>
  <td><img src="<?php echo $row1['f3t8']?>" class="table-logo" alt="LOGO" id="f3t8" onclick="change(id)">
        <input type="hidden" name="f3t8" id="f3t81" value="<?php echo $row1['f3t8']?>">
</td>
  <td><img src="<?php echo $row1['f3t9']?>" class="table-logo" alt="LOGO" id="f3t9" onclick="change(id)">
        <input type="hidden" name="f3t9" id="f3t91" value="<?php echo $row1['f3t9']?>">
</td>
 </tr>
</table>

<table style="width:40%;position:absolute;bottom:0;right:0;">
<caption> <h1>Floor 4</h1></caption>

 <tr>
  <td><img src="<?php echo $row1['f4t1']?>" class="table-logo" alt="LOGO" id="f4t1" onclick="change(id)">
        <input type="hidden" name="f4t1" id="f4t11" value="<?php echo $row1['f4t1']?>">
</td>
  <td><img src="<?php echo $row1['f4t2']?>" class="table-logo" alt="LOGO" id="f4t2" onclick="change(id)">
        <input type="hidden" name="f4t2" id="f4t21" value="<?php echo $row1['f4t2']?>">
</td>
  <td><img src="<?php echo $row1['f4t3']?>" class="table-logo" alt="LOGO" id="f4t3" onclick="change(id)">
        <input type="hidden" name="f4t3" id="f4t31" value="<?php echo $row1['f4t3']?>">
</td>
 </tr>
 <tr>
  <td><img src="<?php echo $row1['f4t4']?>" class="table-logo" alt="LOGO" id="f4t4" onclick="change(id)">
```

```
        <input type="hidden" name="f4t4" id="f4t41" value="<?php echo $row1['f4t4']?>">
</td>
  <td><img src="<?php echo $row1['f4t5']?>" class="table-logo" alt="LOGO" id="f4t5"
onclick="change(id)">
        <input type="hidden" name="f4t5" id="f4t51" value="<?php echo $row1['f4t5']?>">
</td>
  <td><img src="<?php echo $row1['f4t6']?>" class="table-logo" alt="LOGO" id="f4t6"
onclick="change(id)">
        <input type="hidden" name="f4t6" id="f4t61" value="<?php echo $row1['f4t6']?>">
</td>
 </tr>
 <tr>
  <td><img src="<?php echo $row1['f4t7']?>" class="table-logo" alt="LOGO" id="f4t7"
onclick="change(id)">
        <input type="hidden" name="f4t7" id="f4t71" value="<?php echo $row1['f4t7']?>">
</td>
  <td><img src="<?php echo $row1['f4t8']?>" class="table-logo" alt="LOGO" id="f4t8"
onclick="change(id)">
        <input type="hidden" name="f4t8" id="f4t81" value="<?php echo $row1['f4t8']?>">
</td>
  <td><img src="<?php echo $row1['f4t9']?>" class="table-logo" alt="LOGO" id="f4t9"
onclick="change(id)">
        <input type="hidden" name="f4t9" id="f4t91" value="<?php echo $row1['f4t9']?>">
</td>
 </tr>
</table>

<table style="width:40%;position:absolute;top:0;right:0;">
<caption> <h1>Floor 2</h1></caption>

 <tr>
  <td><img src="<?php echo $row1['f2t1']?>" class="table-logo" alt="LOGO" id="f2t1"
onclick="change(id)">
        <input type="hidden" name="f2t1" id="f2t11" value="<?php echo $row1['f2t1']?>">
</td>
  <td><img src="<?php echo $row1['f2t2']?>" class="table-logo" alt="LOGO" id="f2t2"
onclick="change(id)">
        <input type="hidden" name="f2t2" id="f2t21" value="<?php echo $row1['f2t2']?>">
</td>
  <td><img src="<?php echo $row1['f2t3']?>" class="table-logo" alt="LOGO" id="f2t3"
onclick="change(id)">
        <input type="hidden" name="f2t3" id="f2t31" value="<?php echo $row1['f2t3']?>">
</td>
 </tr>
 <tr>
```

```
  <td><img src="<?php echo $row1['f2t4']?>" class="table-logo" alt="LOGO" id="f2t4" onclick="change(id)">
        <input type="hidden" name="f2t4" id="f2t41" value="<?php echo $row1['f2t4']?>">
</td>
  <td><img src="<?php echo $row1['f2t5']?>" class="table-logo" alt="LOGO" id="f2t5" onclick="change(id)">
        <input type="hidden" name="f2t5" id="f2t51" value="<?php echo $row1['f2t5']?>">
</td>
  <td><img src="<?php echo $row1['f2t6']?>" class="table-logo" alt="LOGO" id="f2t6" onclick="change(id)">
        <input type="hidden" name="f2t6" id="f2t61" value="<?php echo $row1['f2t6']?>">
</td>
 </tr>
 <tr>
  <td><img src="<?php echo $row1['f2t7']?>" class="table-logo" alt="LOGO" id="f2t7" onclick="change(id)">
        <input type="hidden" name="f2t7" id="f2t71" value="<?php echo $row1['f2t7']?>">
</td>
  <td><img src="<?php echo $row1['f2t8']?>" class="table-logo" alt="LOGO" id="f2t8" onclick="change(id)">
        <input type="hidden" name="f2t8" id="f2t81" value="<?php echo $row1['f2t8']?>">
</td>
  <td><img src="<?php echo $row1['f2t9']?>" class="table-logo" alt="LOGO" id="f2t9" onclick="change(id)">
        <input type="hidden" name="f2t9" id="f2t91" value="<?php echo $row1['f2t9']?>">
</td>
 </tr>
</table>




<script>
  function goBack() {
      window.history.back();
    }
                $('.form-control').each(function () {
                        floatedLabel($(this));
                });

                $('.form-control').on('input', function () {
                        floatedLabel($(this));
                });

                function floatedLabel(input) {
```

```
                                var $field = input.closest('.form-group');
                                if (input.val()) {
                                        $field.addClass('input-not-empty');
                                } else {
                                        $field.removeClass('input-not-empty');
                                }
                        }
   function change(element)
{
        var x = document.getElementById(element);
        var x11= document.getElementById(element+"1");
        var x1=x11.getAttribute("value");
        var v = x.getAttribute("src");

  if(v == "images/reserved.png"){
     alert("TABLE RESERVED ! PLEASE SELECT ANOTHER.")
          v1= "<?php echo 'images/reserved.png'?>";
  }
  else if(v == "images/available.png"){
        v = "images/selected.png";
        v1= "<?php echo 'images/reserved.png'?>";
  }
  else{
        v = "images/available.png";
        v1= "<?php echo 'images/available.png'?>";
  }
  x11.setAttribute("value", v1);
  x.setAttribute("src", v);
//        var x = document.getElementById(element).src;
//   $.post("aa.php",{aaa:x}, function(data){
//        alert(data);
//        ("#"+element).html(data);

//   });
}
function Func(){
   Logo=document.getElementById('Logo');
   Logo.style.transform="rotate(10000deg)";

}

}
   </script>

</form>
</body>
```

6. Testing

The purpose of testing is to discover errors. Testing is a process of trying to discover every conceivable fault or weakness in a work product.
It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

Software testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for well planned through testing.

Testing Objectives:

There are several rules that can serve as testing objectives they are:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is the one that has a high probability of finding an undiscovered error.

6.1 Types of Testing:

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at different phases of software development are:

6.1.1 Unit Testing:

Unit testing is done on individual models as they are completed and becomes executable. It is confined only to the designer's requirements. Unit testing is different from and should be preceded by other techniques, including:

- Inform Debugging
- Code Inspection

Each module can be testing using the following two strategies:

Black Box testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program.
This testing has been used to find error in the following categories:
Incorrect or missing functions

- Interface errors
- Errors in data structures are external database access
- Performance error
- Initialization and termination of errors
- In this testing only the output is checked for correctness
- The logical flow of data is not checked

White Box testing

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.
It has been used to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

6.1.2 Integration Testing

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together. It is typically performed by developers, especially at the lower, module to module level. Testers become involved in higher levels

6.1.3 System Testing

Involves in house testing of the entire system before delivery to the user. The aim is to satisfy the user the system meets all requirements of the client's specifications. It is conducted by the testing organization if a company has one. Test data may range from and generated to production.

Requires test scheduling to plan and organize:

1. Inclusion of changes/fixes. 2. Test data to use

One common approach is graduated testing: as system testing progresses and (hopefully) fewer and fewer defects are found, the code is frozen for testing for increasingly longer time periods.

6.1.4 Acceptance Testing

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

User Acceptance Test (UAT)

"Beta testing": Acceptance testing in the customer environment.

Requirements traceability:

- Match requirements to test cases.
- Every requirement has to be cleared by at least one test case.
- Display in a matrix of requirements vs. test cases.

6.2 Test Cases

In general a test case is a set of test data and test programs and their expected results. A test case in software engineering normally consists of a unique identifier, requirements references from a design specification, preconditions, events, a series of steps (also known as actions) to follow, input, output and it validates one or more systems requirements and generates a pass or fail.

Test case 6.2.1 –for login

Test plan id: C01

Test case id: l01

Feature to be tested: Login

Preconditions: 1.Website must be running.

Test script: 1.Verify user id

2. Verify password

3. Display the appropriate page

Test data: 1.valid user id

2. Invalid user id

3. Valid password

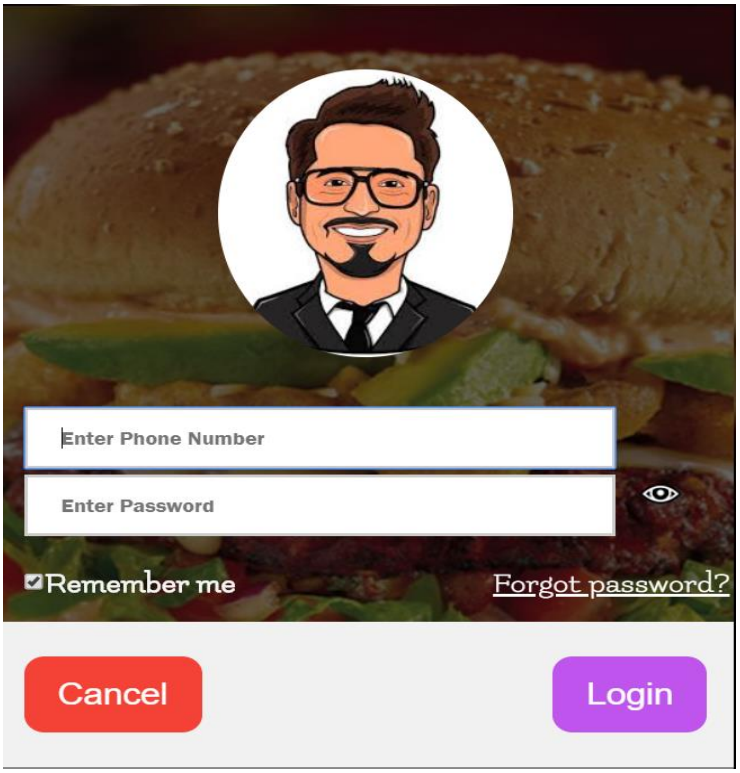
4. Invalid password

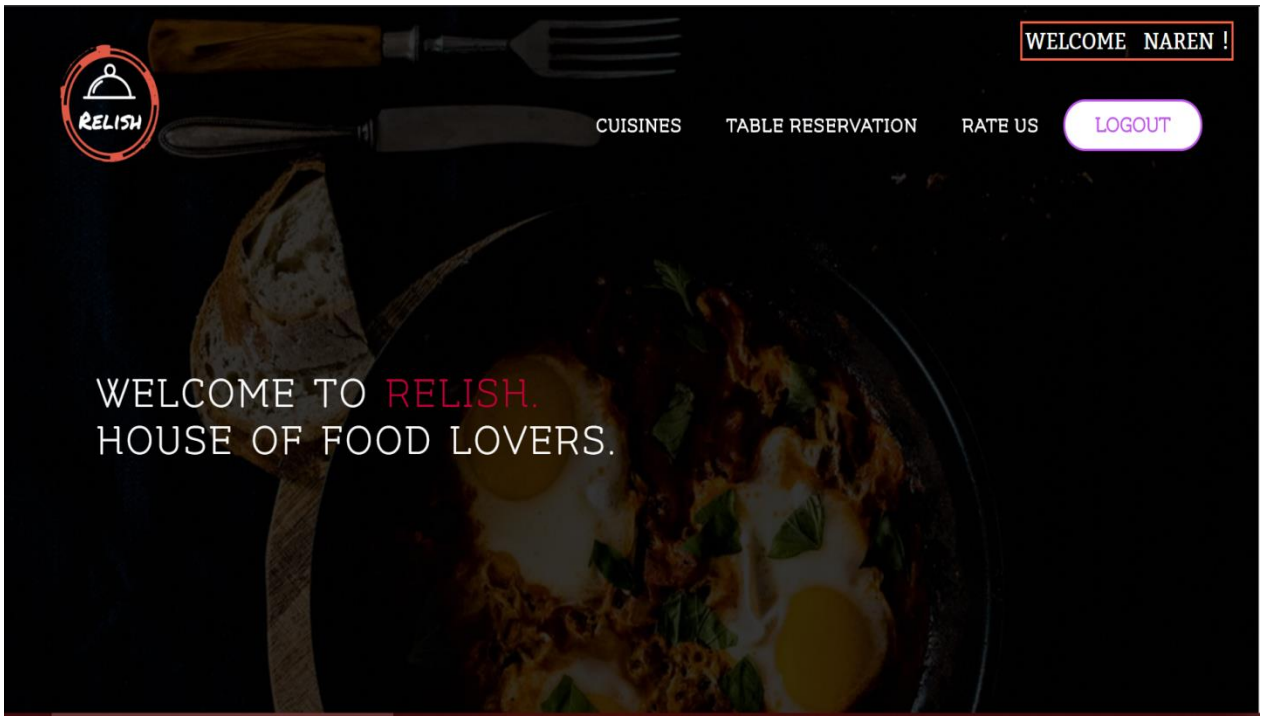
Expected results:

1. Display the homepage.

2. Display message re-enter the details in case of invalid.

Test status: Pass





Error report:

Test Plan ID	Test case ID	Error ID	Error
C01	101	E01	Database Connection Unsuccessful
C01	101	E02	Incorrect Username
C01	101	E03	Incorrect Password

Table 2(a): Error report for login

Test case 6.2.2– for ordering food

Test plan id: C02

Test case id: of01

Feature to be tested: Ordering food

Preconditions: 1.user must be logged in.

Test script: 1. Verify login details

2. Select cuisines
3. Select items
4. Select quantity
5. Checkout the cart
6. Enter name and mail
7. Display order details and order id.

Test data: 1.valid user id

2. Invalid user id
3. Valid password
4. Invalid password
5. Valid order id
6. Invalid order id

Expected results:

1. Order must be inserted in table by valid user.
2. Display the details of order including order id.

Test status: Pass

Order ID	Order Name	Product ID	Product Name	Product Price	Quantity	Total
1	joshua	2	PLAIN NAAN	70.00	1	70.00
1	joshua	3	BUTTER NAAN	90.00	2	180.00
1	joshua	8	SPECIAL FRIED RICE	150.00	1	150.00
1	joshua	9	VEGETABLE BIRYANI	150.00	3	450.00
1	joshua	12	MUTTON BIRYANI	230.00	1	230.00
1	joshua	15	VEGETABLE SOUP	85.00	1	85.00
1	joshua	24	CAESAR SALAD	170.00	1	170.00

Error report:

Test Plan ID	**Test case ID**	**Error ID**	**Error**
C02	Of01	E001	Database Connection Unsuccessful
C02	Of01	E002	Incorrect Bill Computation
C02	Of01	E003	Wrong Order Id
C02	Of01	E004	Database update failed

Table 2(b): Error report for ordering food

Test case 6.2.3–for Reserving table

Test plan id: C03

Test case id: rt01

Feature to be tested: Table reservation

Preconditions: 1.user must be logged in.

Test script: 1. Verify login details

2. Select table reservation
3. Enter date & time
4. Select tables
5. Display confirmation message.

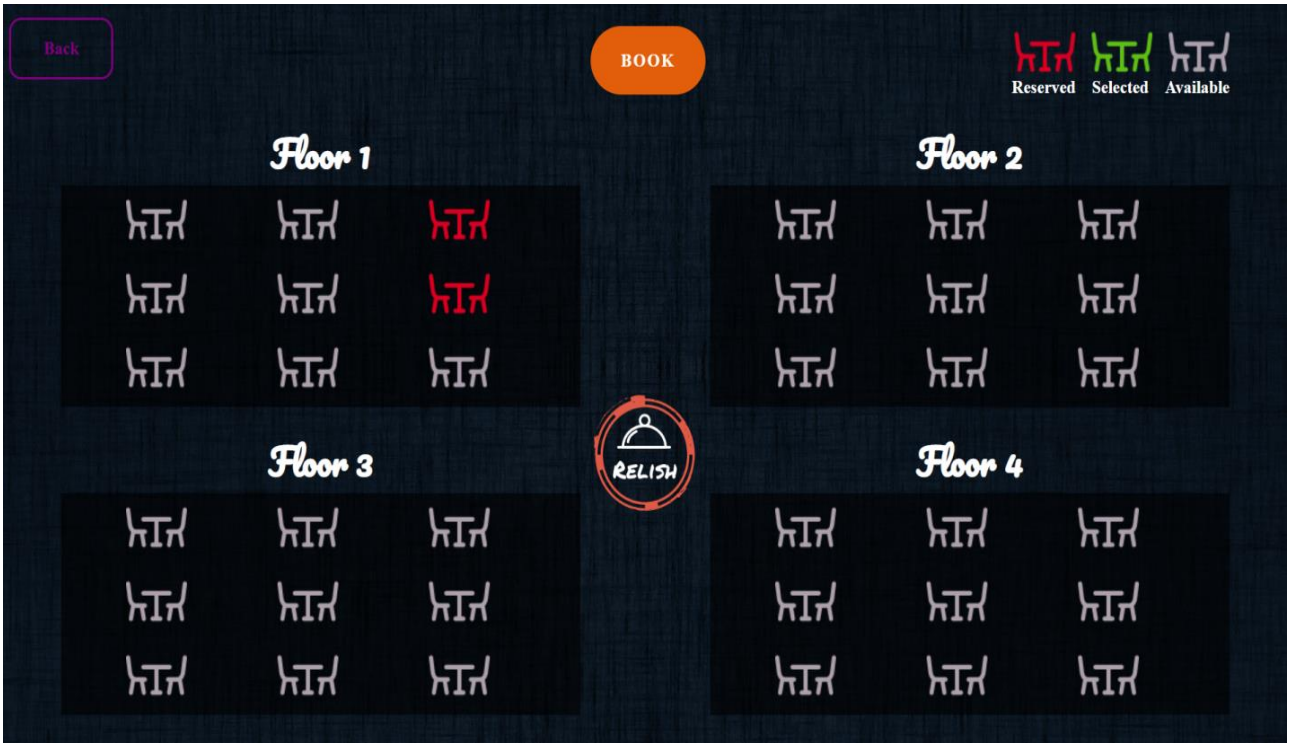
Test data: 1.valid user id

2. Invalid user id
3. Valid password
4. Invalid password
5. Valid confirmation message
6. Invalid confirmation message

Expected results:

1. Tables must be selected by valid user.
2. Display the details of selected table or confirm message.

Test status: Pass



Error report:

Test Plan ID	Test case ID	Error ID	Error
C02	Of01	E001	Database Connection Unsuccessful
C02	Of01	E002	Incorrect Bill Computation
C02	Of01	E003	Wrong Order Id
C02	Of01	E004	Database update failed

Table 2(c): Error report for table reservation

Test case 6.2.4–for updating availability of table

Test plan id: C04

Test case id: ut01

Feature to be tested: Update status of table

Preconditions: 1.Admin must be logged in

2. The table data along with date and time must exist in database.

Test script: 1.verify login details

2. Check table id

3. Select status category

4. Update table status in database

Test data: 1.valid user id

2. Invalid user id

3. Valid password

4. Invalid password

5. Valid table id

6. Invalid table id

Expected results:

2. Table data must be updated in table by valid admin and valid table id.
3. Display changes / updation made.
4. Display message re-enter the details in case of invalid.

Test status: Pass

Before Updating

After Updating

Error report:

Test Plan ID	Test case ID	Error ID	Error
C02	Of01	E001	Database Connection Unsuccessful
C02	Of01	E002	Incorrect Bill Computation
C02	Of01	E003	Wrong Order Id
C02	Of01	E004	Database update failed

Table 2(d): Error report for updating availability of table

Test report

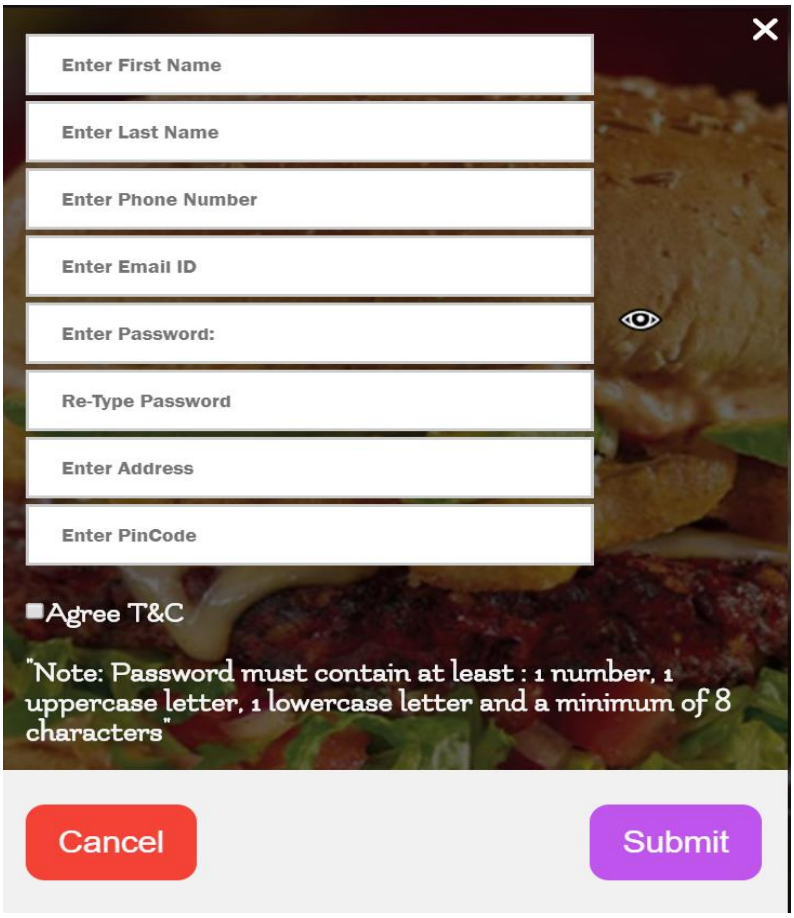
Test case ID	Test condition	Sample input	Expected output	Sample output	Test status
C01	To login	Login id, password	Login into the page	Login into the page	pass
C02	For ordering food	Food items added to cart	Order placed and display order id	Order placed and display order id	pass
C03	For reserving table	Select date and time and tables	Details of table	Details of table	pass
C04	Updating availability of table	Admin login, update table	Display updated tables	Display updated tables	pass

Table 3: Test report

7.Results

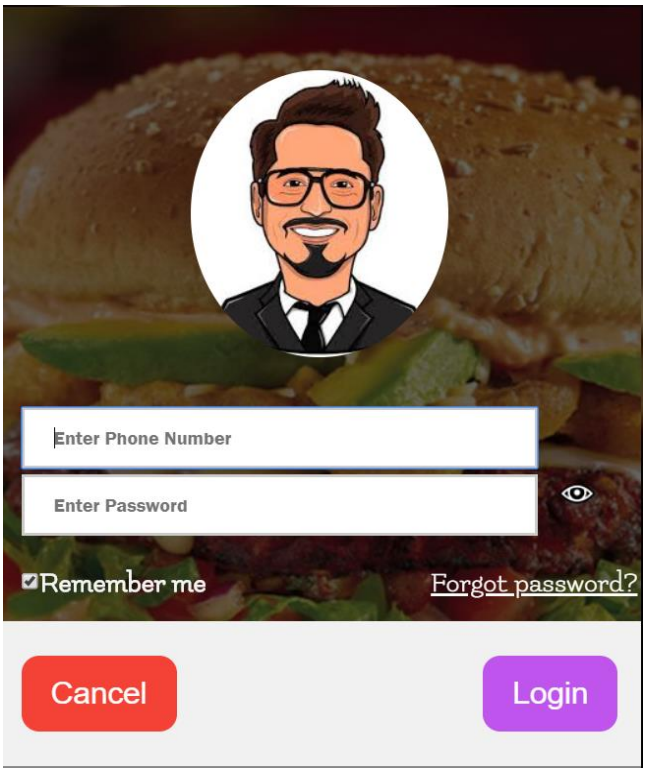
Signup:

This is a modal / popup window which allows the users to create a new account in our website. Only the users from selective pincodes can register.



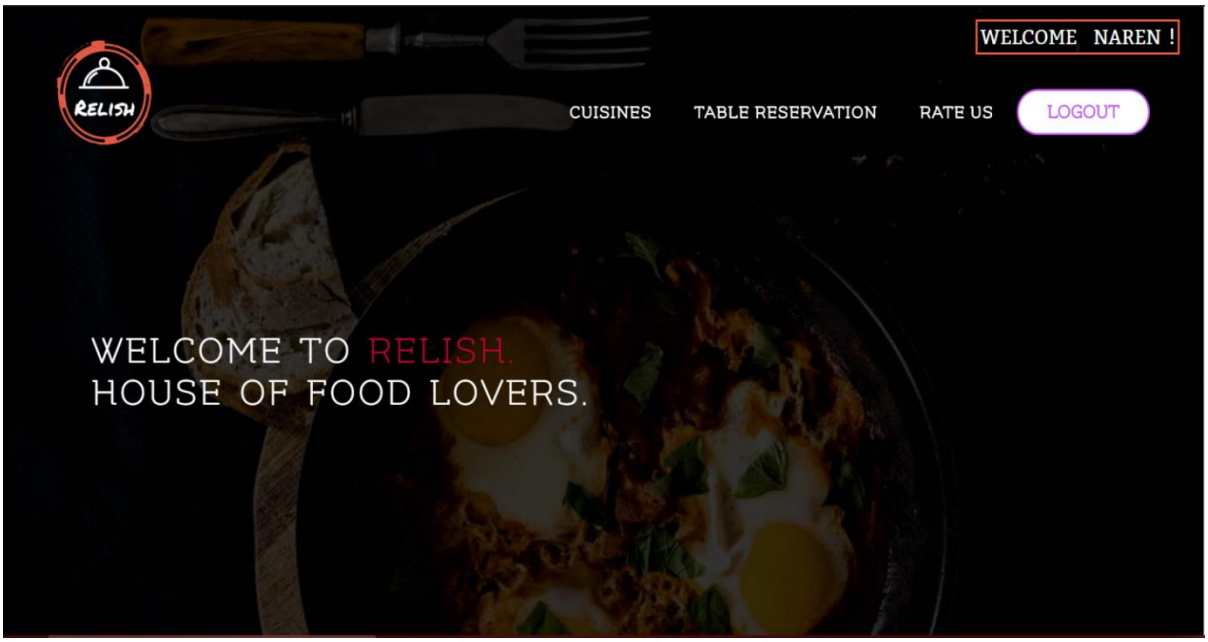
Login:

This is a modal / popup in which each user must login with his/her phone number and password set in the signup phase.



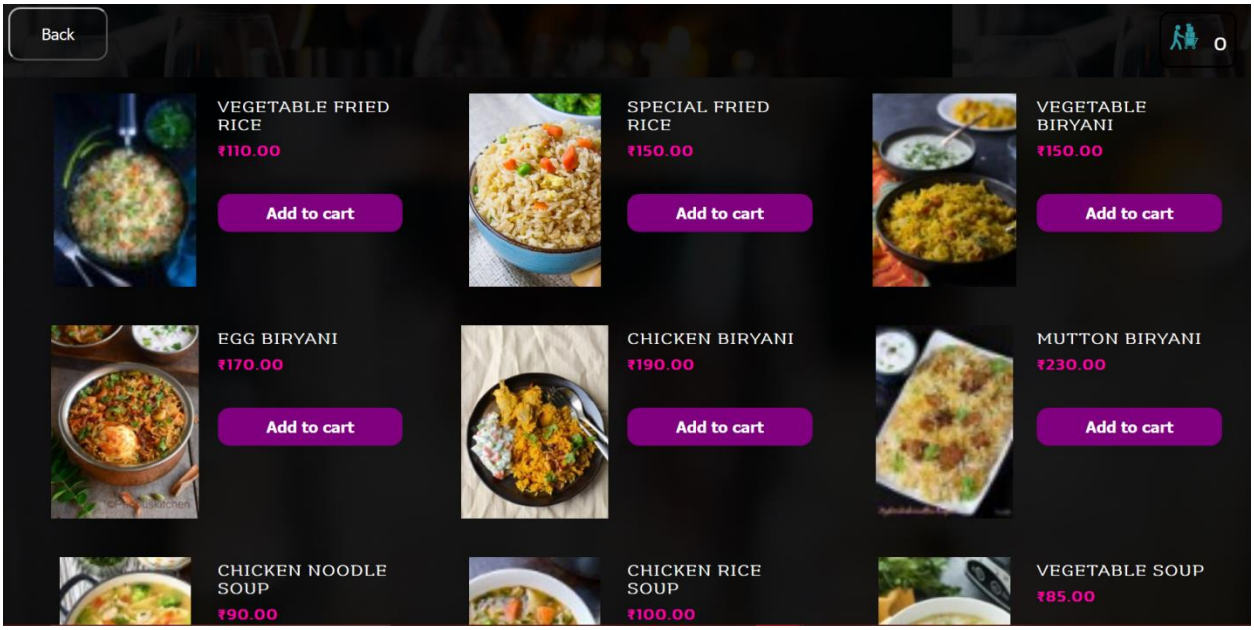
Homepage:

The homepage is the first page which appears after login. It can further navigate to several other pages like table reservation, food-ordering, etc.



Menu:

The menu page displays several items which the user can order. It also displays a user's cart which can be toggled any time to see the final bill.



Cart:

The cart displays the user's order along quantity and with the final total price which is to be paid.

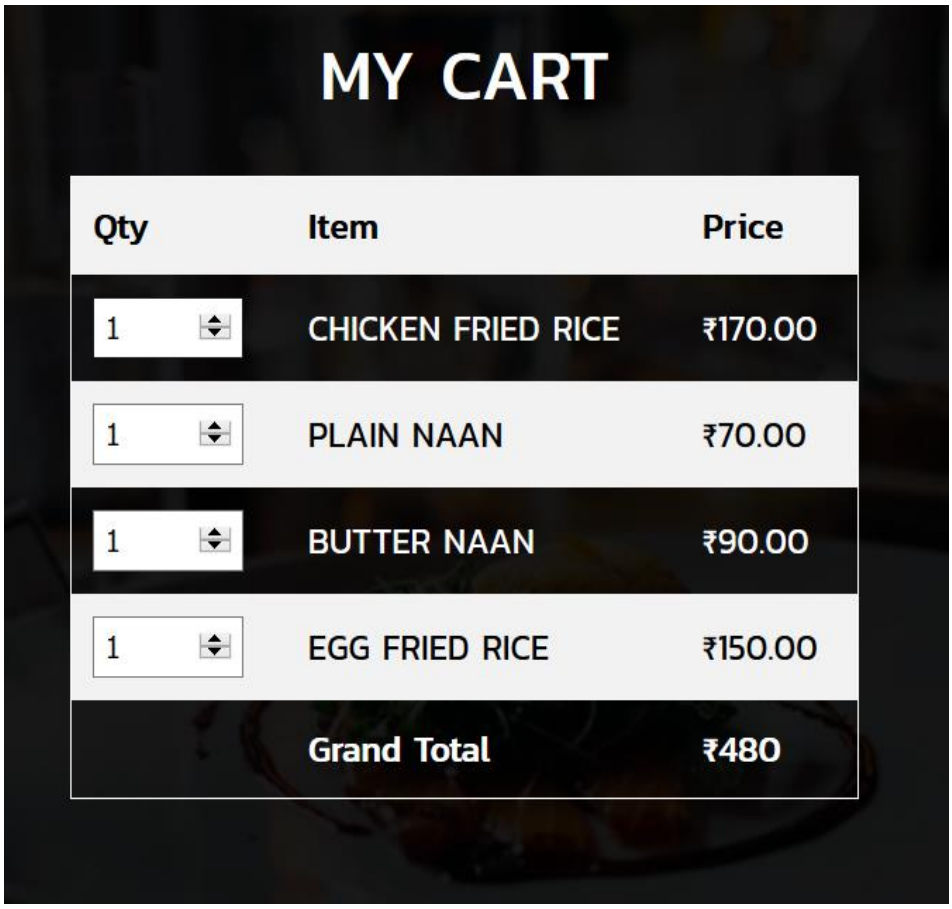
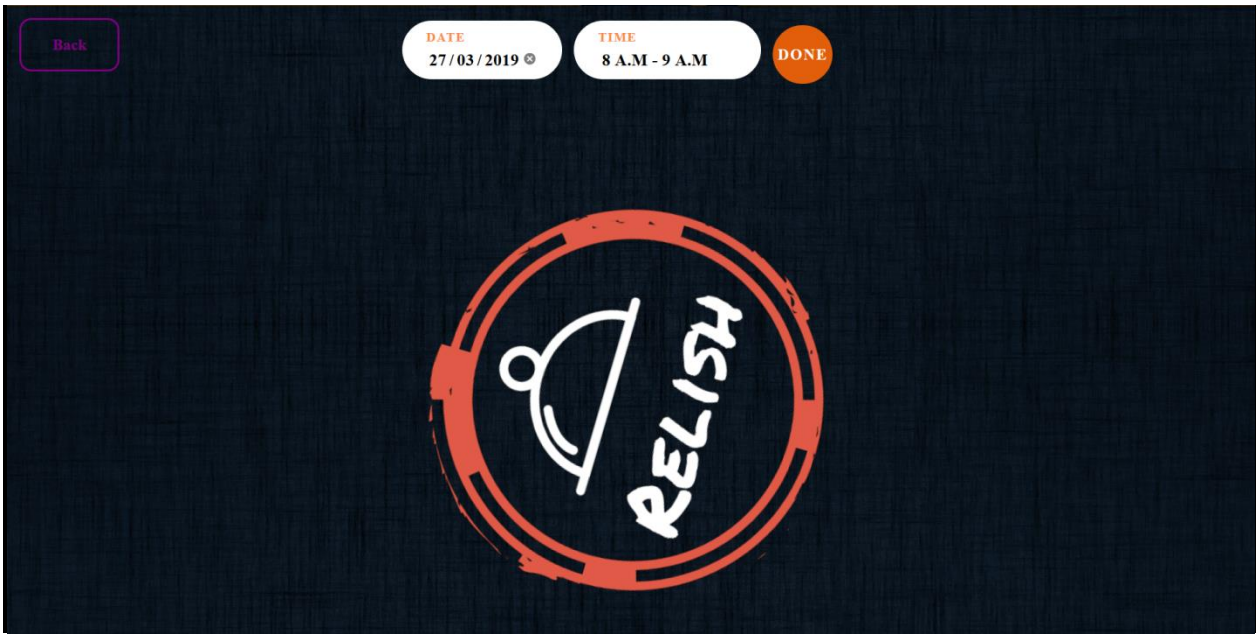


Table reservation:

This is the first page which appears after selecting table reservation. In this page, the user selects the time slot to book the tables. Then, he will be redirected to the main table booking page where he can select from the available tables.



Rating:

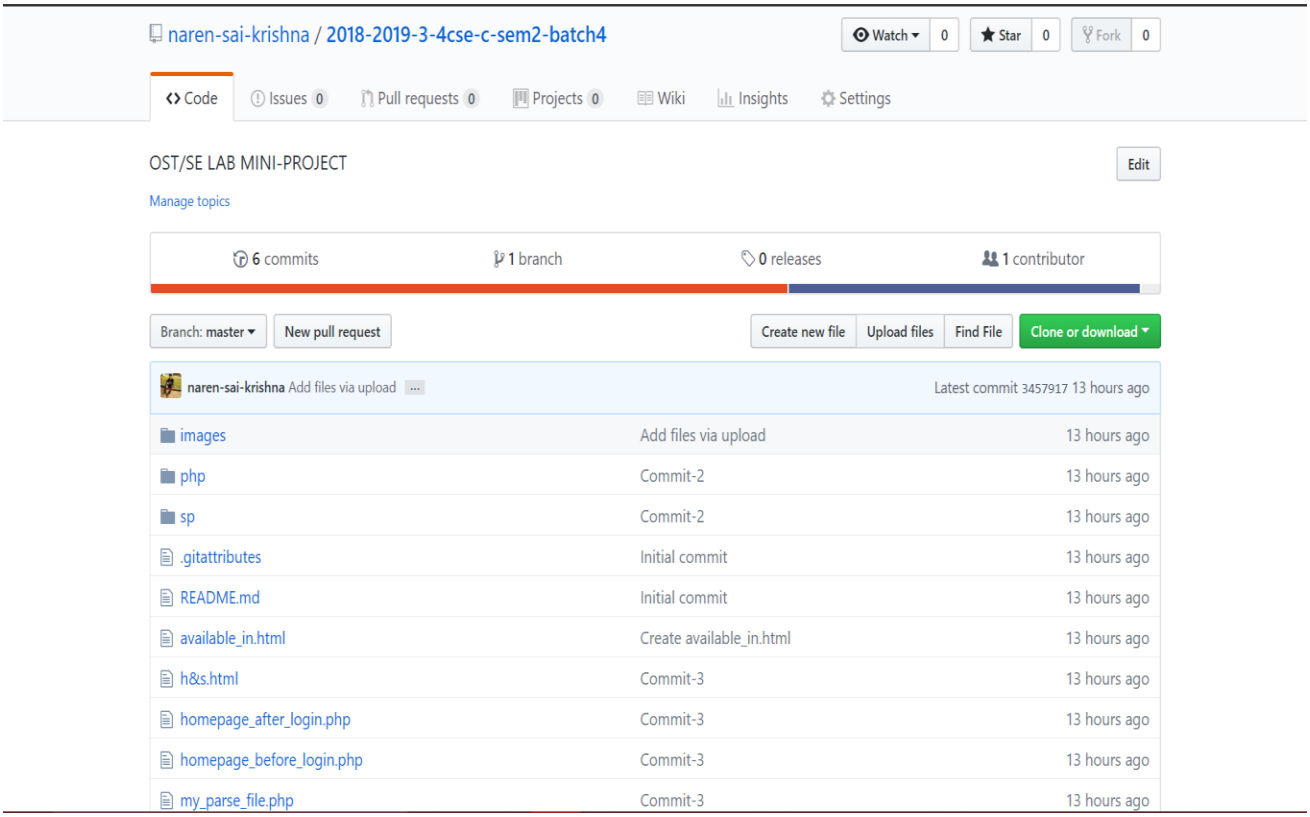
Customer's can provide rating for the service provided to them by clicking on the rating tab in the navigation bar.

8.References

1. https://www.w3schools.com/
2. https://stackoverflow.com/
3. Learning PHP, MySQL & JavaScript, 4th Edition
   With jQuery, CSS & HTML5, Robin Nixon, O'Reilly Media

Github Link

https://github.com/naren-sai-krishna/2018-2019-3-4cse-c-sem2-batch4



APPENDIX:

SAMPLE I/O CODE:

(CUISINES PAGE)

```
<?php
/* [INIT] */
session_start();
if(!isset($_SESSION['sfirstname'])){
        header('Location: ../homepage_before_login.php');
}
require __DIR__ . DIRECTORY_SEPARATOR . "lib" . DIRECTORY_SEPARATOR . "config.php";
require PATH_LIB . "lib-db.php";
require PATH_LIB . "lib-cart.php";
$cartLib = new Cart();
$products = $cartLib->pGet();

/* [DRAW HTML] */
?>
<!DOCTYPE html>
<html>
  <head>
  <link rel="icon" href="Logo.png">
    <!-- [META] -->
```

```
    <title>Cuisines</title>

    <meta name="description" content="Cart">
    <meta name="author" content="Naren">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- [SCRIPTS & STYLES] -->
    <link rel="stylesheet" href="public/theme.css">
    <link href="https://fonts.googleapis.com/css?family=Kanit" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <link href="https://fonts.googleapis.com/css?family=Flamenco" rel="stylesheet">
    <script src="public/general.js"></script>
    <script src="public/cart.js"></script>
  </head>
  <body>
    <script>
      function goBack() {
          window.history.back();
        }
      </script>
    <!-- [NOTIFICATION BOX] -->
    <div id="noteOut"><div id="noteIn"></div></div>

    <!-- [HEADER] -->
```

```
<header id="page-header">

<button class="btn" onclick="goBack()">Back</button>
<!-- <div id="list" style="margin-left:20%;">
   <a href="" >Indian</a> 
   <a href="" >Italian</a> 
   <a href="" >Chinese</a> 
   <a href="" >Desserts</a> 
</div> -->
  <div id="page-cart" onclick="cart.toggle();">

  <img src="images/cart.png" alt="Shopping Cart" width="30" height="30">  <span id="page-cart-count">0</span>
  </div>
</header>

<!-- [PRODUCTS] -->
<div id="products" >
<?php
  if (is_array($products)) {
    foreach ($products as $id => $row) {
      ?>
      <div class="pdt">

        <img src="images/<?= $row['product_image'] ?>" height="200px" width="250px" />
```

```
            <h3 class="pdtName"><?= $row['product_name'] ?></h3>

            <div class="pdtPrice">&#8377;<?= $row['product_price'] ?></div>

            <div class="pdtDesc"><?= $row['product_description'] ?></div><br>

            <input class="pdtAdd" type="button" value="Add to cart" style=".pdtAdd:hover{opacity:0.2;}" onclick="cart.add(<?= $row['product_id'] ?>);" style="position:relative;"/>

          </div>

         <?php

         }

       } else {

         echo "No products found.";

       }

       ?></div>


    <!-- [CART] -->

    <div id="cart" class="ninja"></div>


    <footer id="page-footer">

      &copy; RELISH 2019

    </footer>

  </body>

</html>
```