



E-RESTAURANT

A Report for the Evaluation 3 of Project 2

Submitted by

RAHUL RUHELA

(18032030076 / 18SCSE2030059)

HARSH BHARDWAJ

(18032030095 / 18SCSE2030082)

in partial fulfillment for the award of the degree of

MASTER IN COMPUTER APPLICATION

IN

Computer Science and Engineering in Computer Application

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Mr. Umesh Kumar Gupta,

Assistant Professor

APRIL / MAY- 2020



SCHOOL OF COMPUTING AND SCIENCE AND
ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report “E - RESTAURANT” is the bonafide work of
“RAHUL RUHELA, HARSH BHARDWAJ” who carried out the project work
under y supervision.

SIGNATURE OF HEAD

Dr. Munish Shabarwal
Dean Assistant Professor
School of Computer Science
& Engineering
Galgotias University Uttar Pradesh

SIGNATURE OF SUPERVISOR

Mr.Umesh Kumar Gupta Professor &
School of Computer Science
& Engineering
Galgotias University Uttar Pradesh

ABSTRACT

E-Restaurant is an online Java web based application that assesses students by conducting online objective exams. This project will enable market level to conduct all needs of users and have automated checking of answers based on the response by the candidates.

The project allows admin to schedule a exam specifying time and set of areas the Food. It would enable many dishes with suitable price here you can book your Table also. The result of the set would be immediately available to the users. Further the User can also check the Table and their dishes and many Items .The admin can also upload related meals on the server which will be made available to the users for download. This project would be helpful for booking your order whatever you want and booking Tble.

TABLE OF CONTENTS

S No	Topic	Page No
1	To Whom It May Concern	1
2	Certification	2
3	Declaration	3
4	Acknowlegment	4
5	Abstract	5
6	Tables of contents	6-7
7	List of Symbols	7
Sr. No.	Chapter	PageNo
CHAPTER 1	INTRODUCTION	08-11
CHAPTER 2	SOFTWARE AND HARDWARE REQUIREMENTS	12

CHAPTER 3	PROJECT ANALYSIS	13-15
CHAPTER 4	LITERATURE REVIEW	16-19
CHAPTER 5	SOFTWARE REQUIREMENT SPECIFICATION	20-28
CHAPTER 6	SOFTWARE DESIGN	29-39
CHAPTER 7	CODE IMPLEMENTATION	40-47
CHAPTER 8	SOFTWARE TESTING	48-51
CHAPTER 9	OUTPUT SCREENS	52-62
CHAPTER 10	LIMITATIONSAND CONCLUSION	63
CHAPTER 11	FUTURE IMPROVEMENTS	64
CHAPTER 12	REFERENCES	65-66

CHAPTER 1

INTRODUCTION

1.1 Introduction

E-Restaurant is the online food ordering and table booking system. It is very easily manageable online food ordering system that allows your restaurant to manage your food order through website where your website visitors can browse food menu in easiest way and order directly to your restaurant with home delivery or dine in options which provides online table booking options for your customers and allow them to book their table for lunch or dinner on their convenient time with their food orders.

In simple words E-Restaurant makes online food ordering very easy as restaurant's customer can easily order their food via restaurant website and allows restaurants to generate more revenue through online table bookings and food orders.

1.2 Problem Statement

Food is one of main factors of daily life so we can see restaurants almost everywhere. Some restaurants are popular, many people want to eat at that restaurant but sometimes there are no enough tables or food for customers. Therefore, such restaurants need to have a system to manage these problems.

Online food ordering and table reservation system to manage the restaurant business. 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. Nowadays, many restaurants manage their business by manual method. Take for example, customer ordering today, a restaurant waiter takes the customer order by manual system with pen and paper.

This is a problem for restaurant waiter; there is the probability of losing and duplicating customer information. Additionally, it would affect to reputation restaurant in operate management of ordering. Hides, the restaurant waiter information also by manual system kept use papered

this is difficult for restaurant administrator to find waiter information, probability missing the paper and difficult to arrange the schedule.

Sometime, waiter information and customer information is important to restaurant administrator for reference in the furthermore, restaurant side needs management in the section menu. This is important to restaurant waiter to manage the menu beside this section is for customer viewer the menu that restaurant prepared and make their ordering. As a result, the current system (manual system) is not effective and efficient to use anymore because the current system cannot save, manage and monitor the restaurant customer information, menu, customer ordering and generate report well.

1.3 Aims and Objective of the Study

Based on the problem stated above, the aims and objective of the project are:

- To develop user interface for Sea Palace online restaurant management system.
- To provide online menu information for customer.
- To develop a database system that will store the information of Sea Palace Restaurant's customers, menu and even customer order and reservation.

1.4 Motivation

What really goads me in to this project ranges from the benefit of networking, webpage development to the concept of internet architecture. I have the optimisms that this will afford me an insight into website.

With the modern development and technology innovation sky-rocketing worldwide, the web augments the ability to communicate throughout the world. It further enhances an unprecedented access of information. It also provides better means of getting information, sharing resources just as it serves as a great research tool in institution of learning, this project basically considers improving on ordering system, reservation and retrieval of food.

1.5 Scope of the Study

The scope of this project is to develop an online food ordering and table reservation system. Consist of three components which is target user, target area and project deliverables. The

target user: the group of users that had been identified to use the system is customer and administrator.

Customer: this user will register to be a member to use the online system of this Sea Palace Restaurant management system. After customer makes online ordering, customer will be confirmed some minute before printing slip whether by email or for dine-in customer, a customer will have to reserve a table for dine-in and will be confirmed some minute before printing slip whether by email or phone.

Administrator: is the person who will manage the entire system. The type of user will also do maintenance and control the application of this system.

1.6 Significance of the Study

This system will be going to help customer and administrator in restaurant especially part of online ordering and reservation table. Most of restaurant has a problem of the ordering and reservation table. The result of online ordering and reservation table will give customer easy to make ordering reservation table online and hopefully can smoothly up the job of administrator and waiter. This system also produces a computerized system in defining the best solution in each ordering and reservation problem faces by customer and administrator.

1.7 Existing System

In old system, we can't do reservation online and old system has following problems.

- More time taking.
- Before reservation it is complex to find all information about the hotel.
- Lot of paper work
- Hard to analyze data real time.
- Difficult to process history data
- Difficult communication b/w different branches
- Hard to introduce new processes.

1.8 Proposed System

The basic aim of the proposed system was to provide all improvised functionality and flavors of the existing system minus the entire drawbacks or shortcomings analyzed. With a front end like JSP, JavaScript, CSS, JQUERY and back end like MySQL most of the major irritants have already vanished. All records of this system are stored in separate databases which are regularly updated, so whenever required these databases are used to respond user queries.

The following points were kept in mind while designing this system.

- The system should be user friendly.
- Data validation whenever necessary to ensure correctness of input data.
- Data security should be taken care of.
- Reduce the redundancy of data.
- Maintaining and updating the database easily.
- This system is based on the very popular Model-View-Controller (MVC) Architecture. MVC is most commonly used in websites, very popular and tried and test.

CHAPTER 2

SOFTWARE AND HARDWARE REQUIREMENTS

2.1 Software Requirements

- OS : Windows 7 or above
- Tool : Net Beans 8.0.2
- Web Server : Tomcat Apache 8.0.15 or Glassfish 4.0
- Platform : Java
- Scripting : JSP
- Backend : MySQL

2.2 Hardware Requirements

- Processor : Intel Dual core and all above
- Main Mem : 1 GB DDR3
- Hard Disk : Approximate of 10 GB of Disk Space
- Keyboard : 108 keys
- Monitor : 20" Color LCD

CHAPTER 3

PROJECT ANALYSIS

3.1 Project Instructions

- Based on the given requirements, conceptualize the Solution Architecture, depict the various architectural components, show interactions and connectedness and show internal and external elements. Design the web services, web methods and database infrastructure needed both client and server.
- Provide an environment for up gradation of application for newer versions that are available in the same domain as web service target.

3.2 Modules

3.2.1 Admin

- Admin Login
- Admin Home

3.2.2 Customer

- Customer Registration
- Customer Login
- Forgot Password
- Customer Home
- Ordering Food Items

- Booking Tables
- Cancellation of Booked Tables
-

3.3 Modules Description

3.3.1 Admin

- Admin Login

Admin should Login the system.

- Admin Home

The admin should be able to perform the following actions-

1. View the list of orders placed.
2. View the tables booked in various slots.
3. Should be able to cancel booked tables when required.
4. Should be able to update total number of tables.
5. Should be able to add new items to the menu.
6. Should be able to modify the items in the menu.

3.3.2 Customer

- Customer Registration

The user should enter valid credentials. An auto generated password should be sent to the entered email address for verification purpose.

- Customer Login

The Customer should enter his email address as user id and the auto generated password received during registration to login the system. The customer can change this password later.

- Forgot Password

In case the customer forgets his password, a new password should be sent to his registered email address which can be used to login the system.

- Customer Home

The customer should be able to perform the following actions-

1. Order Food Items
2. Book Table
3. Cancel Booked Tables.
4. Edit Profile
5. View the Chefs, Gallery and Contact information of the restaurant.

- Ordering Food Items

The customer should be able to order food items online by

1. Going to the menu and view the items in different categories.
2. Choose item's quantity and add them to the cart.
3. Go to the cart.
4. Edit the delivery address or use the default address.
5. Click Order Now.
6. After redirection to payment gateway, choose payment mode and pay the

billing amount.

- Booking Tables

The customer should be able to book tables online by-

1. Going to book table module.
2. Choose date, slot and number of tables.
3. Click Book Now and confirm book.
4. After redirection to payment gateway, choose payment mode and pay

the billing amount.

- Cancellation of Booked Tables

The customer should be able to cancel booked tables through booking id.

The refund amount should be initiated to his account.

- Payment

The customer should choose the payment mode and pay the billing amount.

CHAPTER 4

LITERATURE REVIEW

4.1 Study of the Background

Taking a glance into the world of computing, one can see that advent of the information technology has changed the way things are done today. Every such aspect of the life is dancing to tune this system, the internet with the world wide web (WWW) as engine to the basic operation of the information system gave room for information gathering, information exchange and access control, with these problems solving tool, we look at corner to see where the introduction of such system plays a significant role and thus the idea of online food ordering and table reservation came into begin in our immediate environment.

The online ordering and table reservation system is a system that allows customer to reserve tables space or food ordering online in advance. It is a system that makes it convenient for customers to enjoy the most comfortable and unfailing booking of seat at the most important times of their choices without the need to move from restaurant to restaurant in search of food availability.

Before computerization of record keepings, end users suffer unconditional time wastage in processing a food ordering or table reservation and most unfortunately, miss such spaces

even after bearing the burden. This is mostly due to failures in the manual system which range from losing of data, misrepresentation of information, or even a result of human nature.

With the advent of online system software, companies took the task of managing the reservations of restaurant into their hands by simply registering interested member's online addresses of restaurants into their system. It was made possible for restaurateurs to attend to their customers online and on real time. Since its inception, many restaurant owners embrace the idea and completely returned control of their booking to these software owners through their local website. With this process, many companies started joining the train in providing similar service so much such that the online restaurant table reservation becomes competitive market.

Now restaurateurs have option of an online dining reservation software solution that comes without the high cost of per seat transactions fees. This system allows restaurateurs to offer dining patrons the ability to book real-time dining reservations through the restaurant website without the worry of being charge the per seat transaction fees.

With the online technology, reservation data such as contact information and dining preferences is captured via the restaurants website, and then owned it and maintained by the restaurant at no charge as each reservation is saved into a proprietary database, thus making it possible for restaurant owners to efficiently and effectively manage its reservations on deck.

Nowadays it is no longer news to hear restaurant users talk about online reservations and thus in a system like our own where people are too busy to even maintain a daily requirement, many of the socio- economic problems of the environment are solved.

Sea Palace Restaurant was chosen as a case study, because it is one of the most patronized restaurants in Mumbai metropolis as it gives the best restaurant service.

Moreover, the daily demand for food has risen up and is expected to rise more tremendously in schools. Sea Palace Restaurant offers more than 1000 people their daily

food requirement in all categories (breakfast, lunch and dinners) the system gives room for reservation and souvenirs at occasions.

The unending waiting time of the system in Sea Palace Restaurant for the need to computerize the system. The online food reservation system has the ability to accept input data, guarantee response on time and fulfill all characteristics of real time processes. Also, as a sector of e-commerce, the system will capture rules and regulations governing online business. Thus, this work will create and provide the entire environment with a comfortable and enabling condition for feeding and many other restaurant services.

4.2 Related Work

The existing systems that inspired and motivated me to develop this system are Zomato and Foodmingo. The concepts and the business models are same as the existing systems except that we also provide dedicated system for restaurants.

4.2.1 Zomato

Its mission is to ensure nobody has a bad meal.

It does this by

- Helping people discover great places around them.
- Their team gathers information from every restaurant on a regular basis to ensure their data is fresh. Their vast community of food lovers share their reviews and photos, so you have all that you need to make an informed choice.
- Building amazing experiences around dining.
- Starting with information for over 1 million restaurants (and counting) globally, they are making dining smoother and more enjoyable with services like online ordering and table reservations.
- Enabling restaurants to create amazing experiences.
- With dedicated engagement and management tools, they are enabling restaurants to spend more time focusing on food itself, which translates directly to

better dining experiences.

- And they are doing it globally

From Vancouver to Auckland, Zomato is used by millions every day to decide where to eat in over 10,000 cities across 23 countries. In a few years, they would be able help point you to a great place to eat no matter what part of the world you're in.

4.2.2 Foodmingo

One Stop Shop for all your food needs

- Foodmingo is an online portal to facilitate
- Food ordering and delivery from restaurant of choice;
- Restaurant table booking;
- Restaurant discounts and deals;
- Banquet booking.

Foodmingo is been piloted in Milton Keynes, UK and now is ready to be launched in India.

4.3 Problem Definition

What I observed is there are various problems with these existing systems

- It is needed if you have so many requests which cannot be handled by one web server. Typically, 10-15k requests per second can be handled by one web server for a dynamic website, but it depends totally on complexity of website/web application. So, it becomes difficult sometimes to manage all the requests at the same time which often leads to server failure and eventually bad customer experience.
- Due to several options available for restaurants, the profit margins of the registered restaurants are very low and eventually the need arises that they have their own dedicated system that can manage food orders and table bookings from their restaurants exclusively.

4.4 How are we different from our competitors?

- Our dedicated system exclusively for your restaurant.
- We provide complete solution that includes, food ordering on web/call, real time table booking and cost optimization consultancy.
- We have allocated generous marketing budget for promoting our portal, hence greater publicity for your restaurant also.
- Investing in state of the art technology.
- Our full focus till we achieve 100% coverage.
- Founded by dedicated and committed professionals to serve your needs.
- Know your customer first.

CHAPTER 5

SOFTWARE REQUIREMENT SPECIFICATION

5.1 Product Description

Sea Palace is one of Mumbai's most prestigious and stylist Indian restaurant which served authentic Indian cuisine. With its simple, elegant yet friendly and vibrant environment, Sea Palace has become a common social gathering place for many friends and families. Therefore, this has helped to boost the growth of its business.

Currently, Sea Palace is using a completely manual based system to carry out some of their day to day operations. Due to the business growth, this system has become inadequate to meet its business requirements.

Some problems it encountered using a manual system when its business is getting busier each day

- More manpower is needed to serve the customers which lead to space constraint in the restaurant.
- Servers complained that they have too much to do within the fastest time possible else customers will get impatient with their service. This may lead to more human error such as carelessness.
- Inefficiency caused returning customers to decrease as the wait time for seats, food to be served, servers' response and billing are getting too long.
- Food quality degrades as food processing time is shortened to satisfy customers' impatience.
- Customers can only reserve seats and order food through phone calls and this means that payment cannot be made beforehand. Some of the food ordered by customers may have special ingredients that need advance purchasing. Last minute cancellation by customers or customers who do not turn up, result in wastage of food and staff effort is put to a waste. Thus, the restaurant will eventually make losses.
- High expenses incurred.

With the aim for solving the above problems, Sea Palace has decided to engage us to develop a portal to

1. Reduce the workload of the staff.
2. Have online payment via credit/debit card.
3. Receive order in real time

Therefore, our software development team will introduce an Online Reservation and Food Ordering System (E-Restaurant) whereby customers can browse the food menu online, which order can be placed and payment can be made through the system and reserve seats

based on restaurant floor plan to pick the exact seat location in the restaurant that the customers prefer to dine at.

5.1.1 Product Vision

The new system (E-Restaurant) aims to increase efficiency to smoother work flow of the restaurant so as to provide top-notch dining experience and service to the customers. It also aims to reduce overheads caused by the manual system and solve the current problems mentioned in Section 1 – Product Description.

5.1.2 Business Requirements

The first version of the E-Restaurant must be available within three months.

E-Restaurant must demonstrate cost saving of at least 20% on labor within a year after the introduction. The reduction of manpower would mean that the restaurant need not have to activate as many staff as before during peak hours or days of the week.

Labor productivity must be improved by 15% at least.

Revenue must result in 20% increase after a year.

New and existing customers patrolling the restaurant must result in 15% increase at least.

5.1.3 Stakeholders and Users

Management – The Board of Directors as the controlling interest in E-Restaurant. Weekly status update meeting will be held to communicate the progress of the project to the management.

Purchaser – Sea Palace who will invest money to develop the system.

User – Customers who use E-Restaurant to interact with Sea Palace.

Developer – Anand Kumar

Staff – Restaurant Manager and Restaurant Supervisor who maintain and update the portal such as adding new items to the menu, making changes to the prices, introducing promotions.

5.1.4 Project Scope

The scope of this project is to develop an Online Reservation and Food Ordering System(E-Restaurant) which will be integrated on Sea Palace website. This system allows reservation making and food ordering services that will provide a convenient dining experience to the customers. Customers can also raise special requests to cater to their needs. In addition, the E-Restaurant system allows customers to choose their desire seats online based on the restaurant floor plan and order food. Then, payment can also be done online through E-Restaurant system. Database will be created to keep track of customers' information and requests.

5.1.5 Assumptions

Order ID will be issued to customers after each transaction with E-Restaurant.

The payment modes will be through PayPal, credit/debit cards or internet banking.

An invoice will automatically be generated after each transaction.

5.1.6 Constraints

The system should support various payment modes.

5.2 Functional Requirements

5.2.1 General

The user shall only be able to perform the following operations:

- For customer
 - Make a reservation
 - Browse menu and place order
 - Make payment
 - Cancel reservation
- For restaurant personnel
 - View orders placed
 - View table(s) booked/cancelled.

5.2.2 Make a Reservation

- When the user initiates 'Book Table, he/she must be taken to the 'Book Table' page.
- The user must be able to enter the following information
 - i. Time and date
 - ii. Number of people
- Once the user submits the information, he/she will be taken to the payment page to choose the payment mode and confirm payment.
- Once the payment succeeds, the table(s) and the information must be updated in the database and the confirmation message mailed to the user.
- If the payment fails, the E-Restaurant must show payment failure message to the user notifying him/her of the reservation failure.

5.2.3 Browse Menu and place order

- The E-Restaurant must display the menu that is updated to the current day and must be coherent to the menu used in the restaurant.
- The menu page must display the menu in tabs with the following categories:
 - i. Veg Main course
 - ii. Non-Veg Main course
 - iii. Rice
 - iv. Roti
 - v. Salad/raita
 - vi. Beverages
 - vii. Sweets
 - viii. Chinese
 - ix. Snacks
- The E-Restaurant must allow the user to view the following about a single item from the menu:
 - i. Item name

- ii. Item image
 - iii. Item price
- The user must be able to select the item by choosing the quantity and clicking on add to cart button and the item will be added to cart, refer to 2.3.7.
- The E-Restaurant must display a cart that contains the items selected by the user at the top of the page.
- The cart must contain the following information
 - i. Items list
 - ii. Items price
 - iii. Items quantity
 - iv. Net amount
 - v. Delivery address
- The user must be able to delete the item from the cart by selecting the “cross button” beside the item name.
- If there is at least one item in the cart, the user must be able to click the “Order Now” button to proceed.
- Once the “Order Now” button is clicked, it must be redirected to the payment gateway and after successful payment of the billing amount the selected items information must be updated in the database.
- If the update fails, the E-Restaurant must show an error message to the user notifying him/her of the failure.

5.2.4 Cancel Reservation

- The E-Restaurant must allow the user to modify or cancel reservation at “Cancel Reservation” Page.
- The user must enter the booking id for verification.
- If the verification is successful, the user must be able to make the cancellation.
- The E-Restaurant must update the database with the cancellation and initiate the refund amount to the user.

5.2.5 View orders placed

The Restaurant Manager can view the list of orders placed and accordingly initiate the preparation and delivery of orders.

5.2.6 View table(s) booked/cancelled

- The Restaurant Manager can view the table(s) reservation and cancellation and accordingly assign or de-assign the tables on given date and slot.
- The Restaurant Manager may also cancel the booking on customer's request.

5.3 Data Requirements

Data requirements describe the format, structure, type, and allowable values of data entering, leaving, or stored by the product.

- The system will only accept data which are correct and not ambiguous. Eg. Mobile number should only be 8 digits long and credit card numbers should be 16 digits long.
- The booking can only be submitted and processed by the system when all required fields of data have been filled up.
- The system should display all times in the 24-hour clock format.
- The system must store customer names in fields recording first and last name.
- When a customer has selected a table of choice for booking but has not yet confirmed his booking, the system will lock out that particular table to other customers.

5.4 Non-functional requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within.

5.4.1 Compatibility

The website should be compatible with both Internet Explorer and Mozilla Firefox, the 2 most widely used browser currently.

5.4.2 User interface

The user interface should be as familiar as possible to users who have used other web applications and Windows desktop applications. E.g., we will follow the UI guidelines for naming menus, buttons, and dialog boxes whenever possible.

5.4.3 Security

- Access will be controlled with usernames and passwords
- Only administrator users will have access to administrative functions, average users will not.
- Database should be reasonably secured to prevent leak or loss of confidential information such as credit card details from customers.

5.4.4 Performance

- The system should be up and running 24/7.
- It should support at least 100 users using the online booking concurrently without any lag.

5.4.5 Backup and Recovery

- There should be a backup server and database to prevent service interruption or loss of data when the main server and database are down.
- Downtime should not last more than 30sec when switching from main server to the backup server in case of a breakdown.

5.4.6 Reliability

- The whole online booking system should achieve a 99% success rate. i.e. downtime should not be more than 1% of its total operating time.

- System review will take place monthly. Any lack in performance or reliability will be addressed and improved on after each review.

5.4.7 System Maintenance

Maintenance of the system will be conducted weekly. Maintenance will be conducted during off-peak hours e.g. between 12am - 6am.

5.5 Interface Requirements

5.5.1 Overview

The user interface of this restaurant booking system is a web site which can be viewed using popular web browsers. This high accessibility made it easier and more convenient for users to use the system. Users don't need to set up any additional software for the purpose of running the system. As long as an Internet connection is available, the system can be easily accessed using their mobile devices. Multi-platforms operation is also an additional advantage of this design.

One more advantage of this design is the power of the Hyper Text Markup Language (HTML). HTML provides nicer features with simple modification and configuration compared to the GUI of other languages. HTML language supports the use of other languages and technique to make dynamic objects, which can improve the vividness of the application.

5.5.2 User Interfaces

These are the fundamental features of the GUI that should be included in the websites:

- A login box comprises of an account and a password text field. Users can sign in using their credentials to check their bookings and orders. We can provide the signup function for long-term users so that they don't have to refill the information every time booking is made.
- A dynamic menu including the links to the homepage, the menu page, the booking page and the information page. The menu page will have the list of foods with their

respective images. It can be divided into many pages to ease up the navigation. The booking page will have a shopping cart function for the booking of food and a clickable map for the reservation of seats. After booking is submitted, the webpage will automatically redirect to the payment page. The information page will provide additional information about the restaurant.

- A slideshow or a flash of the images of the restaurant.
- Images of the top ordered dishes and their respective information (e.g. price).
- A panel for advertisements coming from our own restaurant or from other parties.

5.6 Hardware Interfaces

Description of how the software application interfaces with hardware *exists outside the scope of the system*.

5.7 Software Interfaces

The use of web design tools such as Bootstrap is employed to make a more professional and nicer design of the system. The code editor and the design editor is integrated in one tool-NetBeans, which allows easy modification as well as addition of elements onto the web pages. Interactive and dynamic objects can be created more easily within a few clicks. The platform to implement the webpage is JSP and MySQL with the support of Apache. JSP is chosen due to its popularity, ease in coding and the availability of free scripts online.

CHAPTER 6

SOFTWARE DESIGN

6.1 Use Case Diagram

A Use Case diagram describes the functionality and users involved in the system. It contains the following-

Actors: They are the users involved in the system including human beings and other system components.

Use Cases: It includes the services provided to users of the system.

The Use Case Diagram for my system is as shown below

FIGURE 6.1 Use Case Diagram

The actors involved in this system are Customer's, Restaurant Manager and Merchant.

The services they get are as described below.

➤ Customer

- Login

After logging into the system, the customer can

- Browse Food Items
- Checks slots available for table booking

- Manage Account

The customer can

- Register
- Login

- Manage Profile
- Place Order

The customer can place order for food items by

 - Adding items to cart
 - Editing item's quantity
 - Viewing cart details
 - Confirming Delivery Address
 - Checking Out
- Book Tables

The customer can choose number of tables on desired date and slot and then

 - Check Out
- Cancel Table Booking

The customer may cancel booked tables on behalf of booking id and

 - Confirm Cancellation
- Merchant
 - Payment Gateway

The merchant authenticates the payment from customer on the billing amount. The payment confirmation is forwarded to restaurant manager.
- Restaurant Manager
 - Manage Catalog

The restaurant manager can

 - View the list of orders placed
 - View Booked Tables
 - Cancel Booked Tables
 - Add new items to the menu
 - Modify items

6.2 Data Flow Diagrams(DFDs)

Data Flow Diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destination.

The data flow diagram is analogous to a road map. It is a network model of all possibilities with different detail shown on different hierarchical levels. This processes of representing different details level is called “leveling” or “partitioning” by some data flow diagram.

Data Flow Diagram (DFD) uses a number of symbols to represent the systems. Data Flow Diagram also known as Bubble Chart is used to clarify system requirements and identifying the major transformations that will become programs in system design.

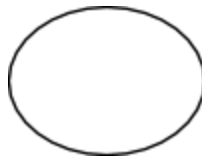
DFD Symbols:

1. A square defines a source (originator) or destination of system data.



2. An arrow identifies data flow-data in motion. It is a pipeline through which data flows.

3. A circle or “bubble” represents a process that transforms incoming data flow into outgoing data flow.



4. An open rectangle is a temporary repository of data.



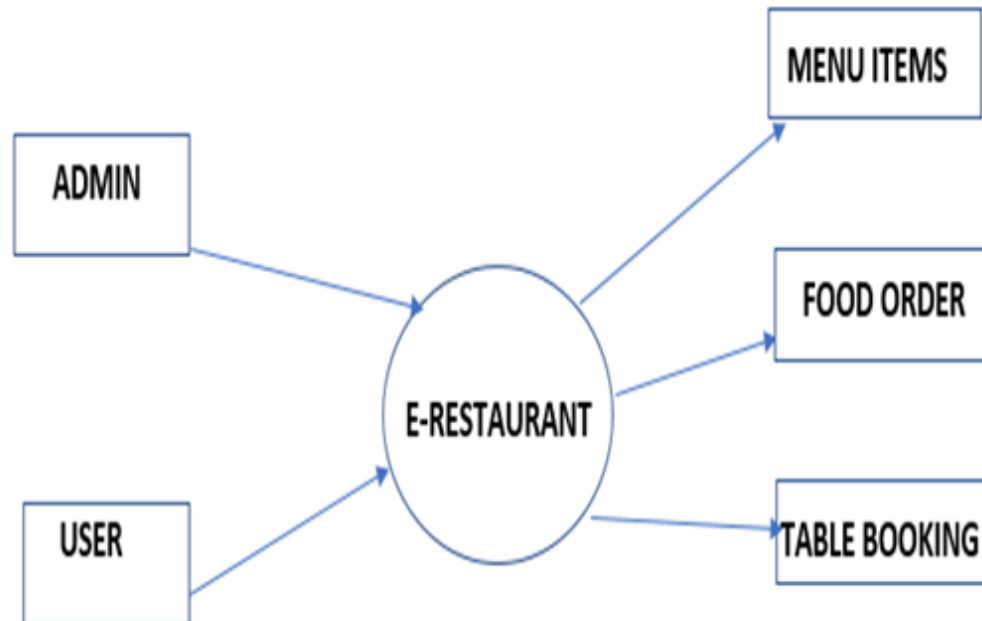


FIGURE 6.2 Context Free Diagram for my system

6.3 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

In order to further describe the behavior of systems, these class diagrams can be complemented by a state diagram or UML state machine.

UML provides mechanisms to represent class members, such as attributes and methods, and additional information about them.

6.4 Object Diagrams

An object diagram in the Unified Modeling Language (UML), is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time.

In the Unified Modeling Language (UML), an object diagram focuses on some particular set of objects and attributes, and the links between these instances. A correlated set of object diagrams provides insight into how an arbitrary view of a system is expected to evolve over time. In early UML specifications the object diagram is described as:

An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time. The use of object diagrams is fairly limited, namely to show examples of data structure.

The latest UML 2.5 specification does not explicitly define object diagrams, but provides a notation for instances of classifiers.

Object diagrams and class diagrams are closely related and use almost identical notation. Both diagrams are meant to visualize static structure of a system. While class diagrams show classes, object diagrams display instances of classes (objects). Object diagrams are more concrete than class diagrams. They are often used to provide examples or act as test cases for class diagrams. Only aspects of current interest in a model are typically shown on an object diagram.

Each object and link on an object diagram is represented by an Instance Specification. This can show an object's classifier (e.g. an abstract or concrete class) and instance name, as well as attributes and other structural features using slots. Each slot corresponds to a single attribute or feature, and may include a value for that entity.

The name on an instance specification optionally shows an instance name, a ':' separator, and optionally one or more classifier names separated by commas. The contents of slots, if any, are included below the names, in a separate attribute compartment. A link is shown as a solid line, and represents an instance of an association.

If you are using a UML modeling tool, you will typically draw object diagrams using some other diagram type, such as on a class diagram. An object instance may be called an instance specification or just an instance.

6.5 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances. Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn't have to wait for a response. Asynchronous calls are present in multithreaded applications and in message-oriented middleware. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message (Execution Specifications in UML).

Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. If an object is destroyed (removed from memory), an X is drawn on bottom of the lifeline, and the dashed line ceases to be drawn below it. It should be the result of a message, either from the object itself, or another. A message sent from outside the diagram can be represented by a message originating from a filled-in circle (found message in UML) or from a border of the sequence diagram (gate in UML).

6.7 Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects. As the number of objects and messages grows, a collaboration diagram can become difficult to read. Several vendors offer software for creating and editing collaboration diagrams.

A Collaboration diagram models the interactions between objects or parts in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.

However, Collaboration diagrams use the free-form arrangement of objects and links as used in Object diagrams. In order to maintain the ordering of messages in such a free-form diagram, messages are labeled with a chronological number and placed near the link the message is sent over. Reading a collaboration diagram involves starting at message 1.0, and following the messages from object to object.

Collaboration diagrams show a lot of the same information as sequence diagrams, but because of how the information is presented, some of it is easier to find in one diagram than the other. Collaboration diagrams show which elements each one interacts with better, but sequence diagrams show the order in which the interactions take place more clearly.

6.8 E-R Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

- **Entities**, which are represented by rectangles. An entity is an object or concept about



which you want to store information.

A weak entity is an entity that must be defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

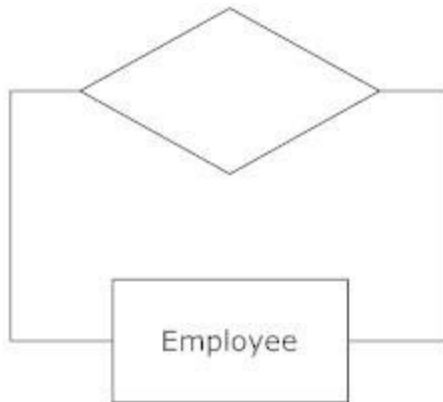


- **Actions**, which are represented by diamond shapes, show how two entities share



information in the database.

In some cases, entities can be self-linked. For example, employees can supervise other employees.



- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



A multivalued attribute can have more than one value. For example, an employee entity



can have multiple skill values.

A derived attribute is based on another attribute. For example, an employee's monthly



salary is based on the employee's annual salary.

- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional.

E-R Diagram for my system

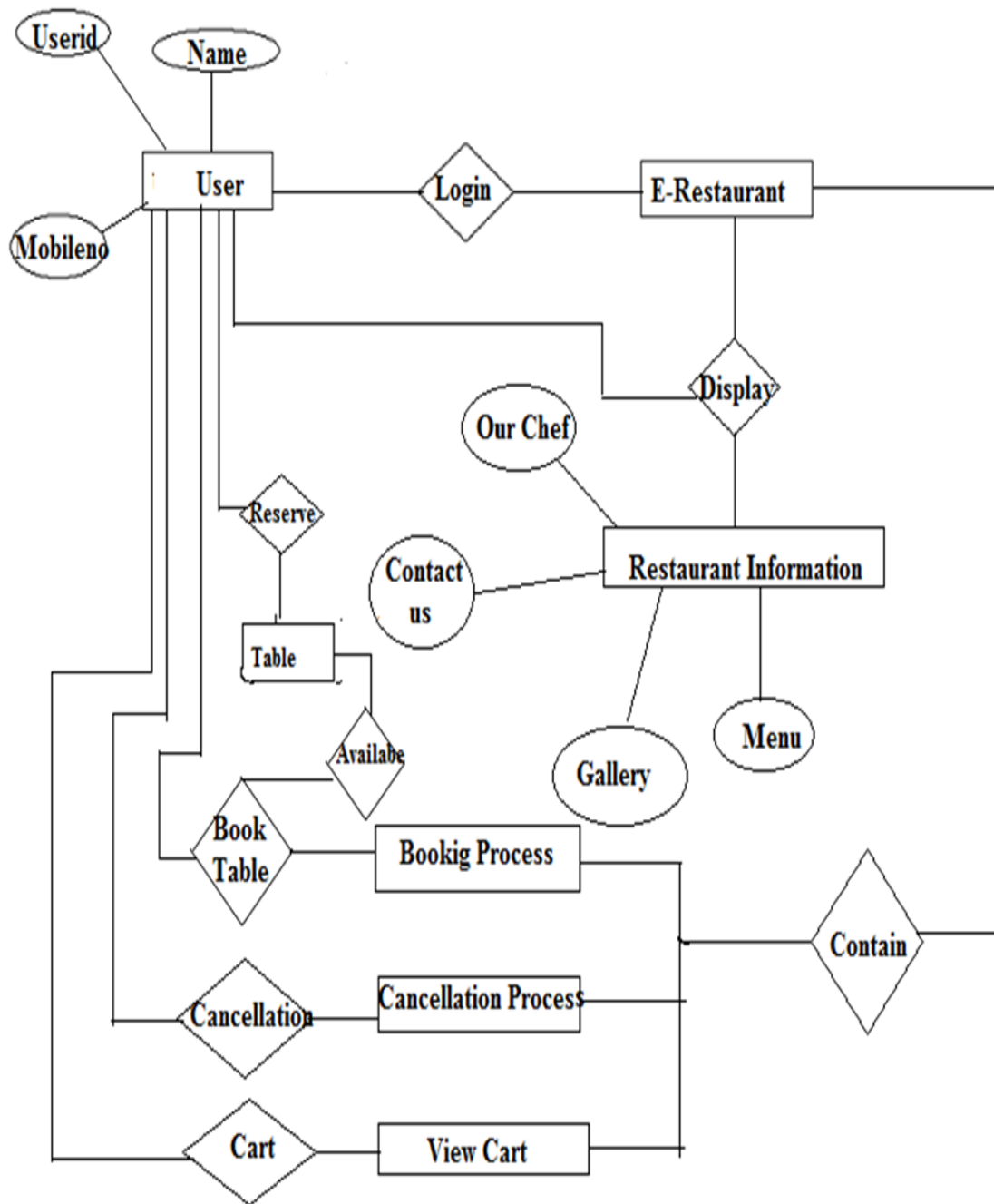


FIGURE 6.12 E-R Diagram

CHAPTER 7

CODE IMPLEMENTATION

7.1 LoginProcess.jsp

```
<%@page import="java.util.*"%>
<%@page import="java.sql.*"%>
<%
    String u=request.getParameter("uid");
    String p=request.getParameter("pass");
    Statement st=(Statement)application.getAttribute("stmt");
    ResultSets=st.executeQuery("select * from userinfo where email='"+u+"' and
pass='"+p+"'");
    if(rs.next())
    {
        session.setAttribute("uid", u);
        session.setAttribute("uname", rs.getString(2));
        LinkedHashMap tm=new LinkedHashMap();
        session.setAttribute("itemStore", tm);
        response.sendRedirect("profile.jsp");
    }
    else
    {
        response.sendRedirect("index.jsp?msg=Wrong Enteries!!");
    }
%>
```

7.2 Order.jsp

```
<%@page import="my.ItemStore"%>
```



```

<%@page import="javax.mail.internet.*"%>
<%@page import="javax.mail.*"%>
<%@page import="java.util.*"%>
<%@page import="java.sql.*"%>
<%
    String e=(String)session.getAttribute("uid");
    String n=(String)session.getAttribute("uname");
    if(e==null)
    {
response.sendRedirect("index.jsp?msg=Please Login First!!");
    }
PreparedStatement
getUserWhereID=(PreparedStatement)application.getAttribute("getUserWhereID");
getUserWhereID.setString(1,e);
ResultSet rs1=getUserWhereID.executeQuery();
rs1.next();
    String addr=rs1.getString(5);
int total=(Integer)session.getAttribute("total");
java.text.DateFormatdateFormat = new java.text.SimpleDateFormat("yyyy-MM-dd");
    String cd=dateFormat.format(new java.util.Date());
PreparedStatement
getMaxOrderID=(PreparedStatement)application.getAttribute("getMaxOrderID");
ResultSetrs=getMaxOrderID.executeQuery();
intoid=1;
if(rs.next())
oid=rs.getInt(1)+1;
LinkedHashMap tm=(LinkedHashMap)session.getAttribute("itemStore");
    Set set = tm.entrySet();

```

```

    Iterator i = set.iterator();
PreparedStatement
insertOrderItem=(PreparedStatement)application.getAttribute("insertOrderItem");
    String item="";
while(i.hasNext())
    {
Map.Entry me = (Map.Entry)i.next();
ItemStore is=(ItemStore)tm.get(me.getKey());
insertOrderItem.setInt(1, oid);
insertOrderItem.setString(2, is.iname);
insertOrderItem.setInt(3, is.price);
insertOrderItem.setInt(4, is.qty);
insertOrderItem.executeUpdate();
        item= item+ "Item: "+is.iname+" Price: "+is.price+" Quantity: "+is.qty+"\n";
    }
PreparedStatement
insertOrder=(PreparedStatement)application.getAttribute("insertOrder");
insertOrder.setInt(1, oid);
insertOrder.setString(2, e);
insertOrder.setString(3, cd);
insertOrder.setInt(4, total);
insertOrder.setString(5, addr);
insertOrder.executeUpdate();
tm=new LinkedHashMap();
session.setAttribute("itemStore", tm);
    String msg="Order Placed Successfull. Your Order ID: "+oid;
    String subject=msg;
    String body="Dear "+n+",your order Successfullly placed and Your Order Id: "+oid+

```

```

        "\n\n"+item+"\n Thank You!!";
final String aemail=application.getInitParameter("aemail");
final String apass=application.getInitParameter("apass");
try
    {
        Properties properties = new Properties();
properties.put("mail.smtp.host", "smtp.gmail.com");
properties.put("mail.smtp.socketFactory.port", "465");
properties.put("mail.smtp.socketFactory.class",
        "javax.net.ssl.SSLSocketFactory");
properties.put("mail.smtp.auth", "true");
properties.put("mail.smtp.port", "465");
        Session ses = Session.getInstance(properties,
newjavax.mail.Authenticator() {
protectedPasswordAuthentication getPasswordAuthentication() {
return new PasswordAuthentication(aemail, apass); }
        });
        Message message = new MimeMessage(ses);
message.setFrom(new InternetAddress(aemail));
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(e));
message.setSubject(subject);
message.setText(body);
Transport.send(message);
    }
catch(Exception ex)
    {
msg=msg+" Mail Sending Fail<br/>" +ex.toString();
    }

```

```

    //response.sendRedirect("cart.jsp?msg=success");
    session.setAttribute("cartno",0);
    response.sendRedirect("cart.jsp?msg="+msg+".<br/>Order delivered within 30 minutes.
    Get other details from your Mail box.");
    %>

```

7.3 BookTableprocess.jsp

```

<%@page import="javax.mail.internet.*"%>
<%@page import="javax.mail.*"%>
<%@page import="java.util.*"%>
<%@page import="java.sql.*"%>
<%
    String e=(String)session.getAttribute("uid");
    String n=(String)session.getAttribute("uname");
    if(e==null) {
        response.sendRedirect("index.jsp?msg=Please Login First!!");
    }
    String bd=request.getParameter("bdate");
    String s=request.getParameter("slot");
    String ntable=request.getParameter("ntable");
    intnt=Integer.parseInt(ntable);
    java.text.DateFormatdateFormat = new java.text.SimpleDateFormat("yyyy-MM-dd");
    String cd=dateFormat.format(new java.util.Date());
    int x=bd.compareTo(cd);
    if(x>=0)
    {
        PreparedStatement
        getTotalTable=(PreparedStatement)application.getAttribute("getTotalTable");
        ResultSetrs=getTotalTable.executeQuery();

```

```

rs.next();
inttt=rs.getInt(1);
PreparedStatement
getTableAllocated=(PreparedStatement)application.getAttribute("getTableAllocated");
getTableAllocated.setString(1, bd);
getTableAllocated.setString(2, s);
rs=getTableAllocated.executeQuery();
int ta=0;
while(rs.next())
ta=ta+rs.getInt(1);
if(nt<=(tt-ta))
    {
        session.setAttribute("bd",bd);
session.setAttribute("s",s);
session.setAttribute("nt",nt);
    %>

```

Table Book Confirmation:

```

<form action="\${initParam['posturl']}" method="post">
<input type="hidden" name="upload" value="1" />
<input type="hidden" name="return" value="\${initParam['returnurl2']}" />
<input type="hidden" name="cmd" value="_cart" />
<input type="hidden" name="business" value="\${initParam['business']}" />
<input type="hidden" name="item_name_1" value="Table Book" /><br/>
<input type="hidden" name="amount_1" value="<%=nt*10%>" /><br/>
<input type="hidden" name="quantity_1" value="1" /><br/>
<input class="bluebutton" name="" type="submit" value="PAY"/>
</form>
<%

```

```

    }
else
    {
response.sendRedirect("booktable.jsp?msg=Unable to Book. Total table left for given date
& slot: "+(tt-ta));
    }
}
else
    {
response.sendRedirect("booktable.jsp?msg=Wrong date Entered!!");
}
%>

```

7.4 Database Tables

userinfo

```

CREATE TABLE `erestaurant`.`userinfo` (
  `email` VARCHAR(45) NOT NULL,`name` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,`phone` VARCHAR(45) NOT NULL,
  `address` VARCHAR(45) NOT NULL,PRIMARY KEY (`email`));

```

booktable

```

CREATE TABLE `erestaurant`.`booktable` (
  `tbid` INT(11) NOT NULL,`name` VARCHAR(45) NOT NULL,
  `date` DATE NOT NULL,`slot` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,`tbook` INT(11) NOT NULL,
  `status` VARCHAR(45) NOT NULL,PRIMARY KEY (`tbid`));

```

orderitem

```

CREATE TABLE `erestaurant`.`orderitem` (

```

```
`oid` INT(11) NOT NULL,`name` VARCHAR(45) NOT NULL,  
`price` INT(11) NOT NULL,`qty` INT(11) NOT NULL,PRIMARY KEY (`oid`));
```

bookorder

```
CREATE TABLE `erestaurant`.`bookorder` (  
`oid` INT(11) NOT NULL,`userid` VARCHAR(45) NOT NULL,  
`odate` DATE NOT NULL,`amount` INT(11) NOT NULL,  
`address` VARCHAR(45) NOT NULL,PRIMARY KEY (`oid`));
```

itemdetails

```
CREATE TABLE `erestaurant`.`itemdetails` (  
`iname` VARCHAR(45) NOT NULL,`price` INT(11) NOT NULL,  
`image` LONGBLOB NOT NULL,`categ` VARCHAR(45) NOT NULL);
```

admininfo

```
CREATE TABLE `erestaurant`.`admininfo` (  
`aid` VARCHAR(45) NOT NULL,`name` VARCHAR(45) NOT NULL ,  
`password` VARCHAR(45) NOT NULL,PRIMARY KEY (`aid`));
```

totaltables

```
CREATE TABLE `erestaurant`.`totaltables` (  
`no_of_tables` INT NOT NULL,PRIMARY KEY (`no_of_tables`));
```

CHAPTER 8

SOFTWARE TESTING

Testing a program consists of subjecting the program to a set of test inputs and observing if the program behaves as expected. If the program fails to behave as expected then the conditions under which failure occurs are noted for later debugging and correction. Various terms associated with Testing are:

FAILURE: It is a manifestation of the error. But the mere presence of an error may not necessarily lead to failure.

TEST CASE: It is the Triplet [I, S, O] where I is the data input to the system, S is the state of the system at which data is input, and O is the expected output of the System.

TEST SUITE: It is the set of all test cases with which a given software product is to be tested.

Software testing is the process used to measure the quality of developed computer software.

Testing is a process of technical investigation, performed on behalf of stakeholders, that is

intended to reveal quality-related information about the product with respect to the context in which it is intended to operate.

There are essentially two approaches to systematically design the Test Case:

- **Black box** testing treats the software as a black-box without any understanding as to how the internals behave. Thus, the tester inputs data and only sees the output from the test object. They are designed using only the software specification of the software.
- **White box** testing, however, is when the tester has access to the internal data structures, code, and algorithms. It is therefore also called the Structural testing.

8.1 Levels of Testing

- **Unit testing** tests the minimal software component, or module. Each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented.
- **Integration testing** exposes defects in the interfaces and interaction between integrated components (modules).
- **Functional testing** tests at any level (class, module, interface, or system) for proper functionality as defined in the specification.
- **System testing** tests a completely integrated system to verify that it meets its requirements.
 - **Alpha testing** refers to the system testing carried out by the test team within the developing organization.
 - **Beta testing** it is the system testing performed by selected group of friendly customers.
 - **Acceptance Testing** refers to the System testing performed by the customer to determine whether to accept or reject the delivery of the system.

Test cases, suites, scripts, and scenarios

- A test case is a software testing document, which consists of event, action, input, output, expected result, and actual result.

- The term test script is the combination of a test case, test procedure, and test data. Initially the term was derived from the product of work created by automated regression test tools.
- The test suite often also contains more detailed instructions or goals for each collection of test cases.
- Collections of test cases are sometimes incorrectly termed a test plan. They might correctly be called a test specification. If sequence is specified, it can be called a test script, scenario, or procedure.

8.2 Test Cases

Project		E- Restaurant System		
Module		Availability		
Test Case no.		1.1		
Test Case Description		Dates are accepted or not		
Steps to Execute test	Test Data	Required Output/Action Description	Observed Output	Pass/Fail/Pending
1. Click on the Sliding window	Tick	Display the Dates and months	Displayed	Pass
2.Click on the Selected date and month	Tick	Selected date is display	displayed	Pass
3.Check Availability	Tick	Show the room is available or not	Display	Pass

TABLE: 8.1 Availability Test Case

Project		E- Restaurant System		
Module		User details		
Test Case Number		1.2		
Test Case Description		Taking all the Details of User		
Steps to Execute test	Test Data	Required Output/Action Description	Observed Output	Pass/Fail/Pending
1.Click on the text boxes, sliding window and enter the details	Tick	Display the details filled by User	Displayed	Pass
2.Checking entries are valid or not	Tick	In case of non-valid entry message shows	Displayed	Pass
2. Short key (ctrl +3)	Press	Show or hide navigation bar	Show or hide	Pass

TABLE: 8.2 User Details Test Case

Project		E- Restaurant System		
Module		Data Base		
Test Case Number		1.2		
Test Case Description		All details are stored in Data Base.		
Steps to Execute test	Test Data	Required Output/Action Description	Observed Output	Pass/Fail/Pending
1.All Entry Enter by User store in the Database	Tick	Entries stored in database	Only check by Administrator	Pass

TABLE: 8.3 Database Test Case

CHAPTER 9


OUTPUT SCREENS

Home Page

Admin Login


E-Restaurant

Menu Chefs Gallery Contact



1 2 3 4 5 6

WELCOME TO OUR SITE



E-Restaurant offers guests a fun and casual dining experience. The locally owned neighborhood favorite boasts a menu full of mouth-watering appetizers, comfort all Indo-Chinese favorites, and delectable desserts. In the evenings, E-Restaurant transforms into a hip lounge with a full-service bar and a mix of sultry beats by local DJ's.

Login Form

User ID:

Password:

Registration Page



WELCOME TO OUR SITE



E-Restaurant offers guests a fun and casual dining experience. The locally owned neighborhood favorite boasts a menu full of mouth-watering appetizers, comfort all Indo-Chinese favorites, and delectable desserts. In the evenings, E-Restaurant transforms into a hip lounge with a full-service bar and a mix of sultry beats by local DJ's.

Registration Form

Name:

Email:

Phone:

Address:

[Register](#)

Menu Page

E-Restaurant

[Profile](#) [BookTable](#) [Menu](#) [Chefs](#) [Gallery](#) [Contact](#) [Cart\[0\]](#) Welcome **Shailesh Britia**

[logout](#)

Veg Main Course

Non-Veg Main Course

Rice

Roti

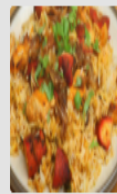
Salad/Raita

Beverages

Sweets

Chinese

Snacks



Chicken Biryani

Quantity:

Price: 240/-

[Add to Cart](#)



Egg Biryani

Quantity:

Price: 160/-

[Add to Cart](#)

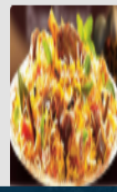


Mutton Biryani

Quantity:

Price: 360/-

[Add to Cart](#)



Hyderabadi biryani

Quantity:


Price: 400/-

[Add to Cart](#)

Cart Page




ant X

localhost:20406/ERestaurant/cart.jsp



1 2 3 4 5 6

My Cart

Item List	Price	Qty	Total
 pastry	120	1	120
 beer	300	3	900
 Chicken Biryani	240	3	720

Net Amount: **1740** /-


Delivery Address

D-16, Meera Road, Navi
Mumbai

[Order Now](#)

Designed & Developed by Shailesh Britia

pe here to search





Order Confirmation Page

Server Open Sc X E-Restaurant X

localhost:20406/ERestaurant_New2/cart.jsp?msg=Order%20Placed%20Successfull.%20Your%20Order%20ID:%2029.
Order%20delivered%20within%2030%20minutes.5

E-Restaurant

[Profile](#) [BookTable](#) [Menu](#) [Chefs](#) [Gallery](#) [Contact](#) [Cart\[0\]](#) Welcome **Shailesh Britia** [logout](#)




1 2 3 4 5 6

My Cart

Order Placed Successfull. Your Order ID: 29.
Order delivered within 30 minutes. Get other details from your Mail box.

pe here to search



Book Table Page



Book Table

Booking Date:

Slot:

No of Tables:

Book

Cancel Table

Booking ID:


Cancel Booking

Designed & Developed by Shailesh Britia

Booked Table Confirmation Page

Server Open Sc x E-Restaurant x

localhost:20406/ERestaurant_New2/booktable.jsp?msg=Table%20Booked%20Successfull.%20Your%20Booking%20ID:%2010



Book Table

Booking Date:

Slot:

No of Tables:


Table Booked Successfull. Your Booking ID: 10

Cancel Table

Booking ID:

Designed & Developed by Shailesh Britia


Type here to search



Cancelled Table Confirmation Page

rant X

localhost:20406/ERestaurant_New2/booktable.jsp?msg2=Table%20Canceled%20Successfull.Refund%20amount%20initiated%20to%20your%20%20account.



1 2 3 4 5 6

Book Table

Booking Date:

Slot:

No of Tables:

Cancel Table

Booking ID:

Table Canceled Successfull.Refund amount initiated to your account.

Designed & Developed by Shailesh Britia

Type here to search

Admin's View Orders Page

E-Restaurant

Orders TableBookings Items Welcome jai

logout



List of Orders Placed (2017-05-02)

Order ID	Items	Total	Date
29	kadhai Paneer(1),	200	2017-05-02
30	kadhai Paneer(1), naan(1),	210	2017-05-02

Add Items Page

int X

localhost:20406/ERestaurant_New2/aitem.jsp

1 2 3 4 5 6

Add New Item

Item Name:

Price:

Category: Veg Main Course ▾

Image: No file chosen

Modify Item

Item Name:


Designed & Developed by Shailesh Britia

be here to search

Modify Items Page

x

localhost:20406/ERestaurant_New2/aitem.jsp



Add New Item

Item Name:

Price:

Category:

Image: No file chosen

Modify Item

Item Name:

Item Name:


Price:

Category:

Image: No file chosen

Designed & Developed by Shailesh Britia

here to search



CHAPTER 10

LIMITATIONS AND CONCLUSION

Even though this dissertation could produce potential outcomes following the research question, there were some limitations, which could be improved in future research. In terms of the users' perceptions of the e-restaurant system, this case study only interviewed restaurant staff. However, there were some issues, which were brought up by the interviewees, such as enhancing customer service and the use by elderly people. Therefore, future research should carry out a case study based on this prototype to examine exact perceptions from restaurant customers. Furthermore, this research focused on developing e-restaurant only for web. Thus, future research should apply system design and source codes in this portfolio to be developed for other kinds of e-restaurant, for example, table side e-restaurant and stand-alone e-restaurant for a waiting area. Moreover, implementation on another platform, such as Android or .Net, is an alternative, which could be carried out in future research. Finally, additional features suggested by restaurant staff, for instance, integration with Point of Sale (POS) system and the advertising of new promotions during meals, could also be included in a new prototype, which should have more functionalities as well as a study of users' perceptions of those requirements.

Hereby I conclude my project report but with that I must confess that throughout the journey of converting this project into reality I have learned a lot and it has given me a face to face exposure with the real projects in the field of information technology. I would also like to mention that I am not going to leave this project here only. I will make sure that it is updated according to the changes in the field of education.

CHAPTER 11

FUTURE IMPROVEMENTS

After the application was deployed and the experiment was conducted in the restaurant, I spent more time at the restaurant to understand the customer reaction of the new system. The observation also helped to analyze the actual operations of the restaurant, and the associated improvements in the application can make it more helpful for the order processing activity. In the next two paragraphs, the improvements suggested by the customers and the restaurants are described. The customers, through the evaluation forms and some directed conversations, suggested the following improvements to the application that the project team decided to implement in the future versions of the application:

Many users felt that the scroll bar in the menu with a long list of menu items should be avoided, as it is difficult to scroll using the smartphone. Therefore, the future UI design of Customer View will be without the scroll bar. Others felt that shortcuts like ‘repeat last order’, similar to the manual scenario will increase the power of the application. The project team decided to observe the restaurant operations more closely on the current scenarios in the order taking process for identifying new features to the application.

Order repetition shortcut is a good idea for the next release. UI needs to be improved even more and the ‘add to tray’ option should be reconsidered. Few users were not able to understand this feature and could not understand why the selected item is not reflected in the order screen. Reduce the number of clicks for placing the order for default settings on most frequently used items and drinks.

CHAPTER 12

Book References

- [1] Shweta ShashikantTanpure, Priyanka R. Shidankar, Madhura M. Joshi, “Automated Food Ordering System with Real-Time Customer Feedback”,in International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 2, February 2013.
- [2] AshutoshBhargave, NiranjanaJadhav, Apurva Joshi, PrachiOke, Prof. Mr. S. R Lahane, “Digital Ordering System for Restaurant using Android”, in International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013.
- [3] Soon Nyeen Cheong, Wei Wing Chiew, Wen JiunYap,“Design and Development of Multi-Touchable E-Restaurant Management System” ,in 2010 International Conference on Science and Social Research (CSSR 2010), December 5 – 7, 2010, Kuala Lumpur, Malaysia.
- [4] Jhabuawala Mustafa, Kothari Radhika, NaikRiddhi, SlatewalaAbdulquadir,“Touch & Dine- A Multi-Touchable Restaurant System”, in UACEE International Journal of Computer Science and its Applications – Volume 2: Issue 1.
- [5] Swapna, M. Firdouse Ali Khan,“Design and Implementation of Ordering System for Restaurants”, in International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 10, December- 2012
- [6] Khairunnisa K., Ayob J., Mohd. HelmyA.Wahab, M. ErdiAyob, M. IzwanAyob, M. AfifAyob, “The Application of Wireless Food Ordering System”, in MASAUM Journal of Computing, Volume 1 Issue 2, September 2009.
- [7] “Qorder” The portable ordering system for Android devices.

[8] Cormac O'Connell, Restaurant Assignment.

[9] GHIRS: Integration of Hotel Management Systems by Web.

Online References

- **www.java.sun.com**
- **www.roseindia.com**
- **The Unified Modeling Language User Guide**
- **Java Server Page-“O reliev”**
- **Java-Complete Reference**