RESTAURANT BILLING SYSTEM

A MINI PROJECT REPORT

18CSC207J - ADVANCED PROGRAMMING PRACTICE

Submitted by

BHAVYA MALHOTRA (RA2111003010951)
ABHIGYAN KASHYAP (RA2111003010936)

Under the guidance of
Dr. Sivakumar S
Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"RESTAURANT BILLING SYSTEM"** is the bonafide work of **Bhavya Malhotra (RA2111003010951)** and **Abhigyan Kashyap (RA2111003010936)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Balamurugan	Dr. M. Pushpalatha
FACULTY	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor & Head
Department of Computing Technologies	Department of Computing Technologies

ABSTRACT

The Restaurant Billing System project is a software solution designed to simplify the process of generating bills for customers at a restaurant. The system provides an intuitive interface for managing orders, generating bills, and handling payments. The software allows restaurant to display menu items. The system also includes features for calculating taxes based on predefined rules. The proposed system is suitable for small-size restaurants, tiffin shops, and eateries that require an efficient and reliable way to manage their billing process. With this software, restaurant owners can reduce manual errors, save time, and enhance their customers' dining experience by providing accurate and timely bills. Additionally, the system provides reporting features that enable restaurant managers to monitor sales, revenue, and trends.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CAPEX	**Capital Expenditure**
OPEX	**Operating Expenditure**
E1R1A1T1	**Effort Requirement Activity Task**
UX	**User Experience**
UAT	**User Acceptance Testing**
IR	**Infrastructure Requirement**
API	**Application Development Interface**
WBS	**Work Breakdown Structure**
SWOT	**Strength, Weakness, Opportunity, Threats**
RMMM	**Remote Monitoring and Management**
GANTT	**Generalized Activity Normalization Time Table**
DFD	**Data Flow Diagram**
ER	**Entity Relationship**
ISA	**Specialization-Generalization**
DW	**Data Base**

CHAPTER 1

INTRODUCTION

A restaurant billing system is a type of point-of-sale software specifically designed for small sale restaurants, food trucks and others in the food services. As many of the restaurants are lacking the technology and infrastructure for generation of bill. many restaurant billing systems are designed to integrate easily with other software applications, which suits and works according to your business.

This Case study focuses on restaurant billing systems. These systems are cutting-edge technology solutions that help streamline operations and make running a restaurant business more efficient and cost-effective. Many restaurants today are turning to RMS as a comprehensive solution for all aspects of their business. From payment processing to communicating with suppliers and customers to inventory management.

- This project aims to solve the problem of billing system in traditional and small-scale restaurants.
- It saves time and effort by reducing the amount of work you must do.
- This project is planned to give a simpler look and user-friendly user interface and supporting easy operation.
- Detail requirement details from the customer.
- Creative and Coding to make an attractive and user-friendly application.

CHAPTER 2

LITERATURE SURVEY

THE HISTORY

In bullet points, describe the current situation.

- Nowadays, restaurant billing is operated manually.
- it becomes hectic for the invoice maker to make the bills for the guests.
- It is most difficult to deal at the time of the rush of the guests in the restaurant.

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- It only deals with the billing system of the restaurant.
- This was a simple project and does not suit high-investment hotels.

APPROACH

List what is needed to complete the project.

- Detail requirement details from the customer.
- Creative and Coding to make an attractive and user-friendly application.

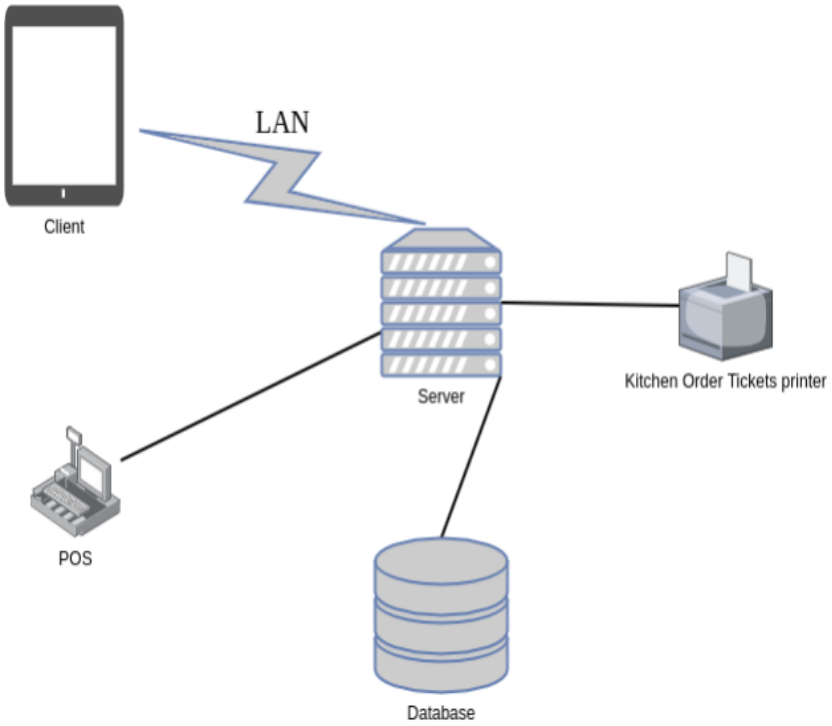
BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

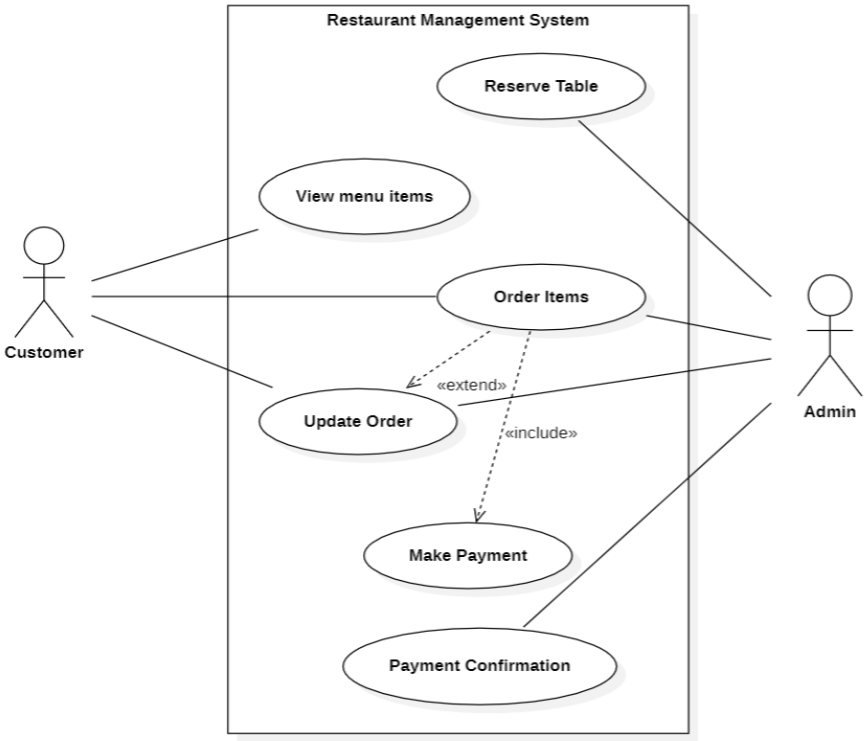
- Smooth, organized and user-friendly interface.
- Simple, easy to operate with Intuitive Menu Configuration.
- Reduce the requirements of human resources.

CHAPTER 3

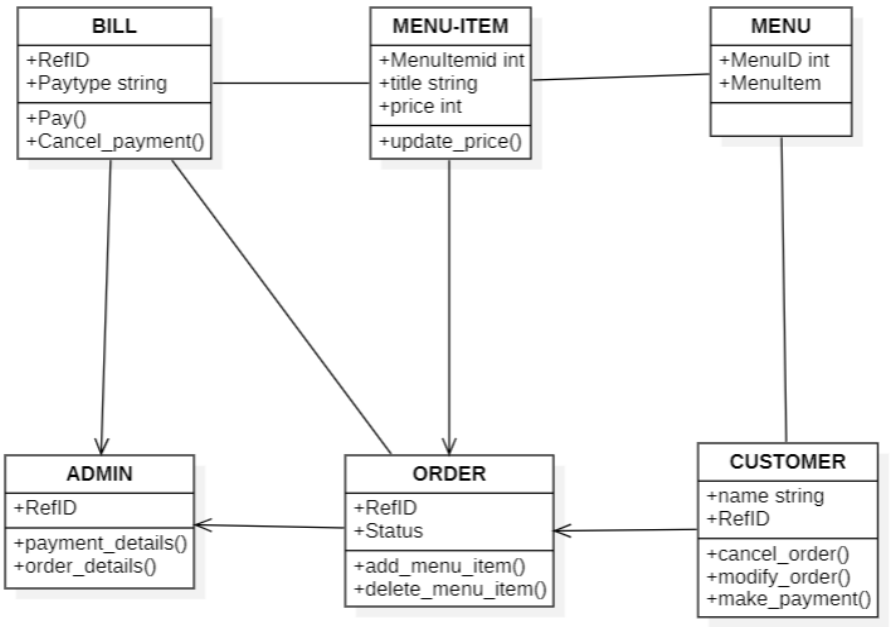
SYSTEM ARCHITECTURE AND DESIGN



USE CASE DIAGRAM:

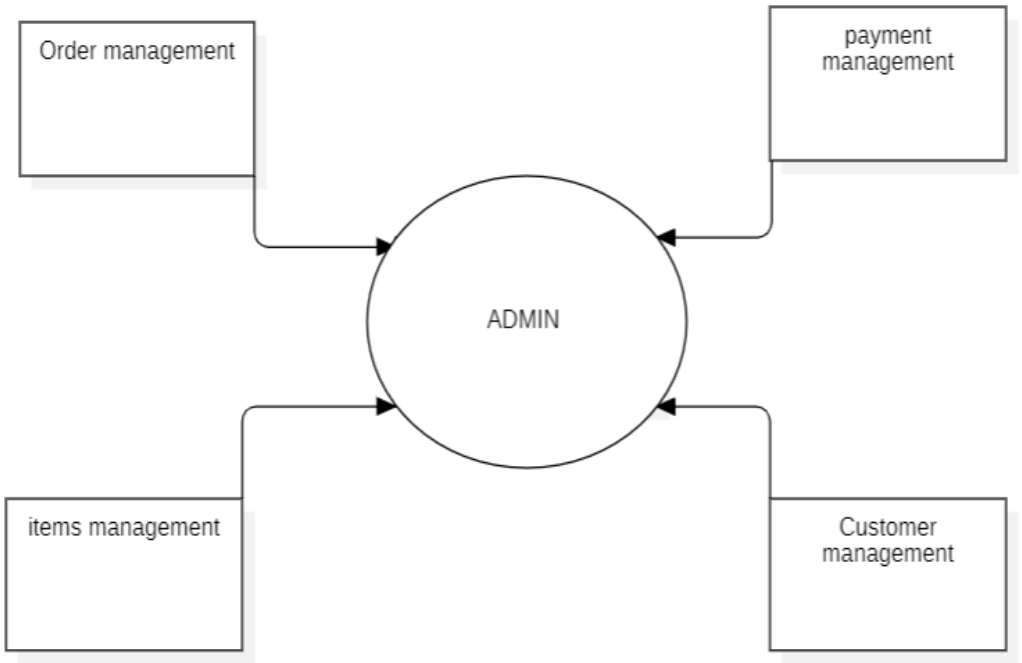


CLASS DIAGRAM:

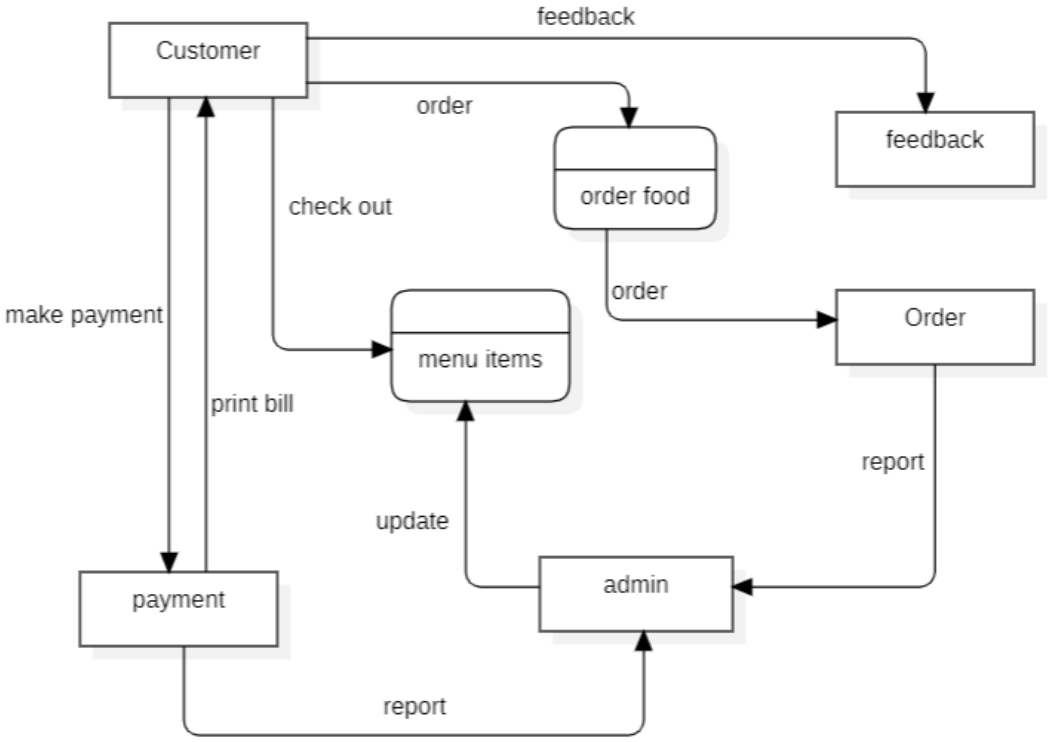


DATA FLOW DIAGRAM:

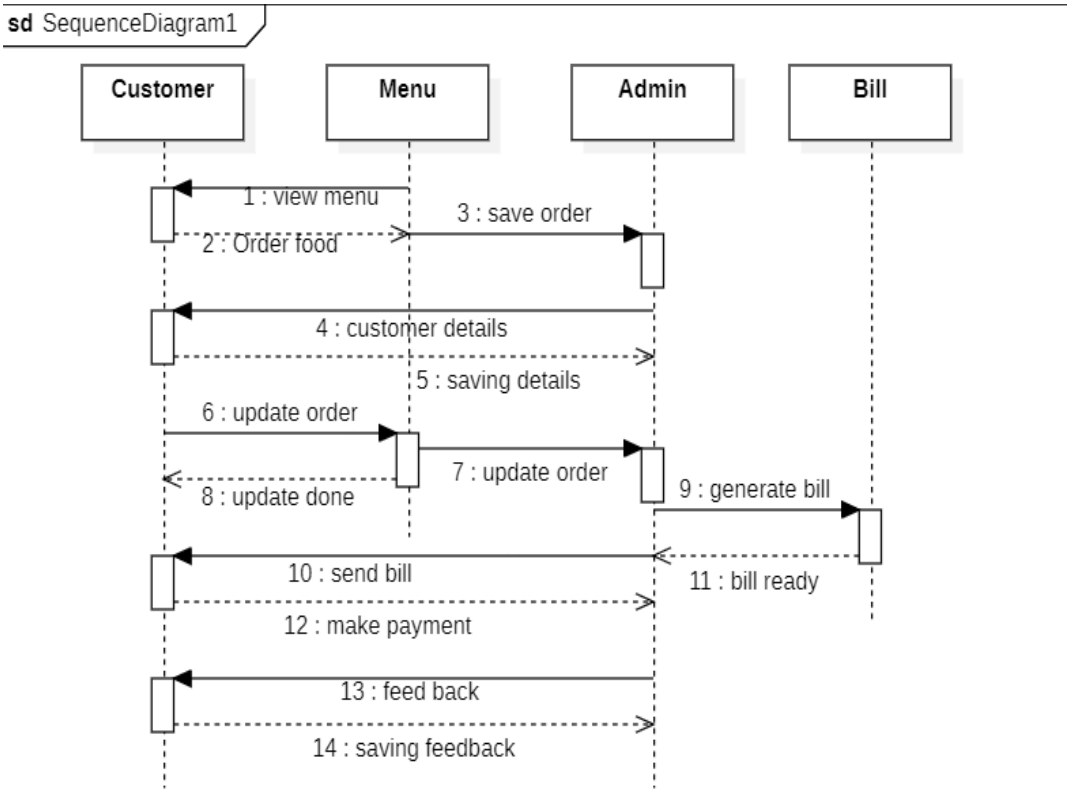
DFD LEVEL 0



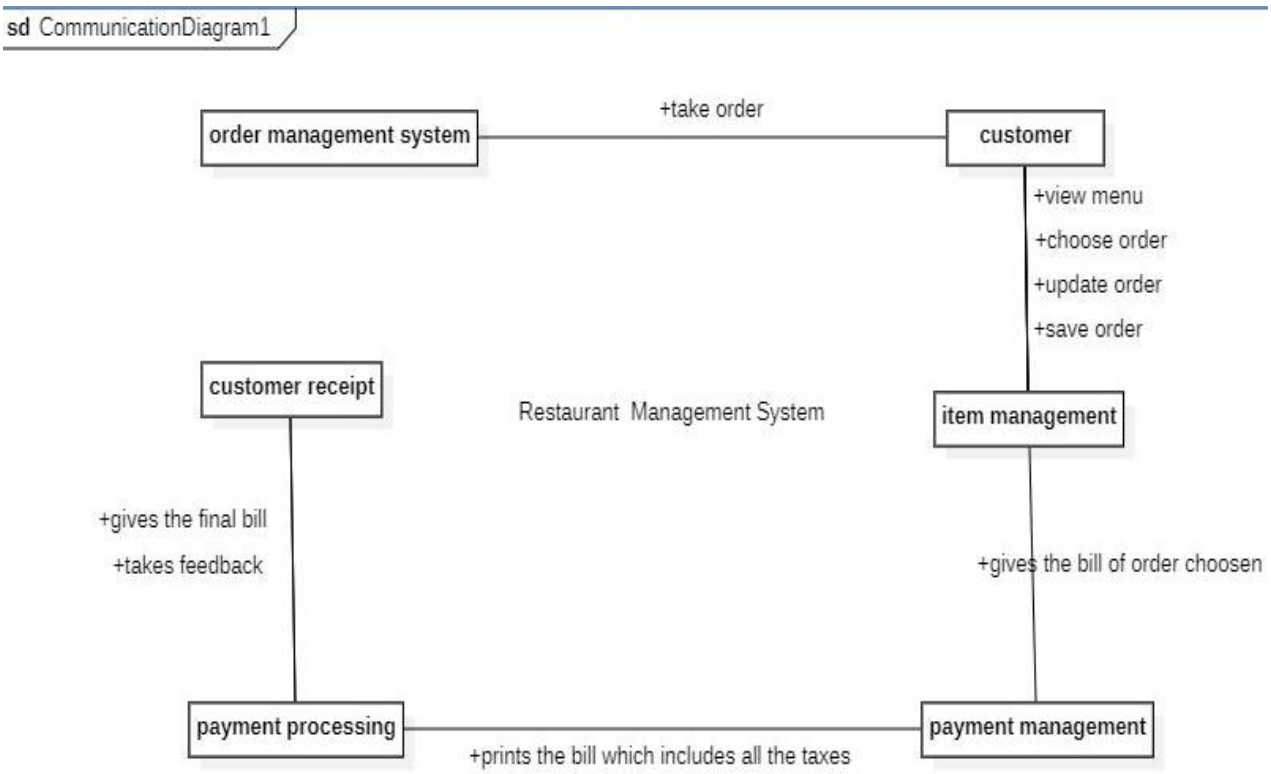
DFD LEVEL 1



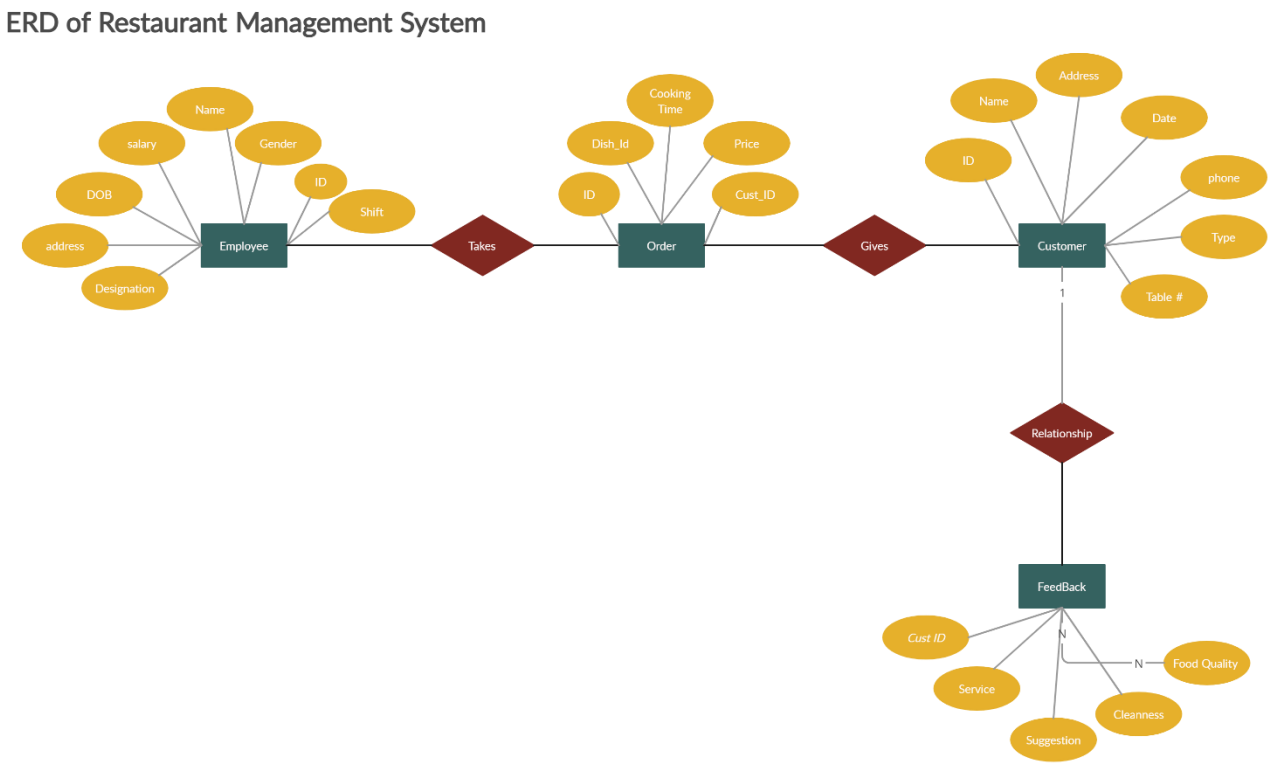
SEQUENCE DIAGRAM:



COLLABORATION DIAGRAM:



ER DIAGRAM:



CHAPTER 4

METHODOLOGY

- The project uses the “Agile process method” as it is faster, and the risk factor is low.
  Agile model breaks down the tasks into smaller steps or iterations which no longer requires long duration for processing each step. The breakdown of steps benefits by increasing the speed of the functioning. The requirements are finalized at the start of the development process.
- This project is broken down into steps like requirement gathering, developing, designing, managing, reviewing and implementation.
- The basic requirements of the Restaurant billing system is first analyzed and then the developers and designers make sure that the project coding and interface is made in a user friendly manner. The tester makes sure that the project is having no errors to debug and then finally it is sent for implementation.
- In this way of processing a project, the advantages include – Speeder results, risk factor is low, transparency in the process procedure and the requirements and implementation methodology is analyzed at the start.

CHAPTER 5

CODING AND TESTING

Import and time module:

```
from tkinter import *
import random
import time
import datetime

root = Tk()
root.geometry("1600x8000")
root.title("Restaurant Management System")
#root.option_add("*Font", "helvetica 20")

Tops = Frame(root, width=1600, relief=SUNKEN)
Tops.pack(side=TOP)

f1 = Frame(root, width=800, height=700, relief=GROOVE)
f1.pack(side=LEFT)

# ================================================================================
#                  TIME MODULE
# ================================================================================
localtime = time.asctime(time.localtime(time.time()))

lblInfo = Label(Tops, font=('perpetua', 50, 'bold'), text="Telugu Bro's Restaurant", fg="aquamarine4", bd=10, anchor='w')
lblInfo.grid(row=0, column=0)

lblInfo = Label(Tops, font=('castellar', 20, 'bold'), text=localtime, fg="aquamarine4", bd=10, anchor='w')
lblInfo.grid(row=1, column=0)
```

Internal logic:

```
def Ref():
    x = random.randint(10908, 500876)
    randomRef = str(x)
    rand.set(randomRef)

    if (Idly.get() == ""):
        CoIdly = 0
    else:
        CoIdly = float(Idly.get())

    if (Dosa.get() == ""):
        CoDosa = 0
    else:
        CoDosa = float(Dosa.get())

    if (Poori.get() == ""):
        CoPoori = 0
    else:
        CoPoori = float(Poori.get())

    if (Pulav.get() == ""):
        CoPulav = 0
    else:
        CoPulav = float(Pulav.get())

    if (Pongal.get() == ""):
        CoPongal = 0
    else:
        CoPongal = float(Pongal.get())
```

```
    CostofMeal = "Rs", str('%.2f' % (CostofIdly + CostofDrinks + CostofDosa + CostofPoori + CostPulav + CostPongal))

    PayTax = ((CostofIdly + CostofDrinks + CostofDosa + CostofPoori + CostPulav + CostPongal) * 0.2)

    TotalCost = (CostofIdly + CostofDrinks + CostofDosa + CostofPoori + CostPulav + CostPongal)

    Ser_Charge = ((CostofIdly + CostofDrinks + CostofDosa + CostofPoori + CostPulav + CostPongal) / 99)

    Service = "Rs", str('%.2f' % Ser_Charge)

    OverAllCost = "Rs", str('%.2f' % (PayTax + TotalCost + Ser_Charge))

    PaidTax = "Rs", str('%.2f' % PayTax)

    Service_Charge.set(Service)
    Cost.set(CostofMeal)
    Tax.set(PaidTax)
    SubTotal.set(CostofMeal)
    Total.set(OverAllCost)
```

Restaurant info 1:

```
lblReference = Label(f1, font=('corbel', 16, 'bold'), text="Reference", bd=16, anchor="w",fg="aquamarine4")
lblReference.grid(row=0, column=1)
txtReference = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=rand, bd=10, insertwidth=4, bg="grey80",
                     justify='right')
txtReference.grid(row=0, column=2)

lblIdly = Label(f1, font=('candara', 16, 'bold'), text="Idly", bd=16, anchor="w")
lblIdly.grid(row=1, column=1)
txtIdly = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Idly, bd=10, insertwidth=4, bg="grey80",
                justify='right')
txtIdly.grid(row=1, column=2)

lblDosa = Label(f1, font=('candara', 16, 'bold'), text="Dosa", bd=16, anchor="w")
lblDosa.grid(row=2, column=1)
txtDosa = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Dosa, bd=10, insertwidth=4, bg="grey80",
                justify='right')
txtDosa.grid(row=2, column=2)

lblPoori = Label(f1, font=('candara', 16, 'bold'), text="Poori", bd=16, anchor="w")
lblPoori.grid(row=3, column=1)
txtPoori = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Poori, bd=10, insertwidth=4, bg="grey80",
                 justify='right')
txtPoori.grid(row=3, column=2)

lblPulav = Label(f1, font=('candara', 16, 'bold'), text="Pulav", bd=16, anchor="w")
lblPulav.grid(row=4, column=1)
txtPulav = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Pulav, bd=10, insertwidth=4, bg="grey80",
                 justify='right')
```

Restaurant info 2:

```
lblDrinks = Label(f1, font=('candara', 16, 'bold'), text="Drinks", bd=16, anchor="w")
lblDrinks.grid(row=0, column=3)
txtDrinks = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Drinks, bd=10, insertwidth=4, bg="grey80",
                  justify='right')
txtDrinks.grid(row=0, column=4)

lblCost = Label(f1, font=('candara', 16, 'bold'), text="Cost of Meal", bd=16, anchor="w")
lblCost.grid(row=1, column=3)
txtCost = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Cost, bd=10, insertwidth=4, bg="grey80",
                justify='right')
txtCost.grid(row=1, column=4)

lblService = Label(f1, font=('candara', 16, 'bold'), text="Service Charge", bd=16, anchor="w")
lblService.grid(row=2, column=3)
txtService = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Service_Charge, bd=10, insertwidth=4, bg="grey80",
                   justify='right')
txtService.grid(row=2, column=4)

lblStateTax = Label(f1, font=('candara', 16, 'bold'), text="State Tax", bd=16, anchor="w")
lblStateTax.grid(row=3, column=3)
txtStateTax = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=Tax, bd=10, insertwidth=4, bg="grey80",
                    justify='right')
txtStateTax.grid(row=3, column=4)

lblSubTotal = Label(f1, font=('corbel', 16, 'bold'), text="Sub Total", bd=16, anchor="w",fg="aquamarine4")
lblSubTotal.grid(row=4, column=3)
txtSubTotal = Entry(f1, font=('Times New Roman', 16, 'bold'), textvariable=SubTotal, bd=10, insertwidth=4, bg="grey80",
                    justify='right')
txtSubTotal.grid(row=4, column=4)
```

Functional test-case report:

Category	Progress Against Plan	Status
Ordered food should be in menu	GREEN	COMPLETED
Proper tax should be entered	GREEN	COMPLETED
Generate reference number	GREEN	COMPLETED
Reset the previous results	GREEN	COMPLETED
To calculate the total bill	GREEN	COMPLETED

Non-functional test-case report:

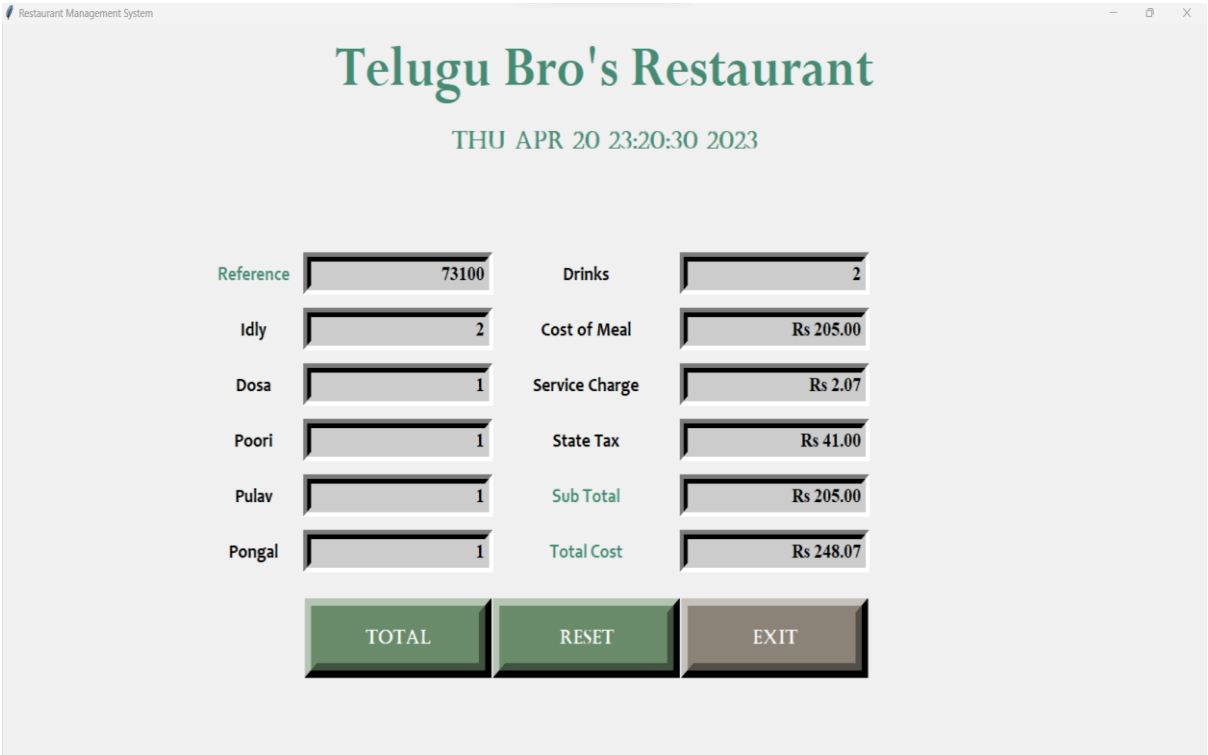
Category	Progress Against Plan	Status
Immediate response from the software	GREEN	COMPLETED
GUI Testing	GREEN	COMPLETED
IDLE testing	GREEN	COMPLETED
Closing the software	GREEN	COMPLETED
To calculate the total bill	GREEN	COMPLETED

Functional	Test Case Coverage (%)	Status
MODULE-1	100%	COMPLETED
BILL-CALCULATION	100%	COMPLETED

CHAPTER 6

SCREENSHOTS AND RESULTS

RESTAURANT BILLING SYSTEM:



CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, the restaurant billing system project with menu management is a critical tool that can simplify the ordering and payment process for both customers and restaurant staff. The project should focus on the functionalities of menu management and billing, ensuring that customers can easily place orders, and staff can efficiently process and calculate bills. Overall, a restaurant billing system with menu management can significantly improve the efficiency and customer experience of any restaurant. It can reduce errors, minimize wait times, and streamline the ordering and payment process, leading to increased customer satisfaction and revenue for the business.

Future Enhancements: Here are some potential enhancements that could be made to the restaurant billing system project in the future:

- Integration with other systems: The restaurant billing system can be integrated with other restaurant billing systems such as inventory management, employee management, and customer relationship management systems. This integration can help improve the efficiency of the restaurant's overall operations.
- Mobile app development: Developing a mobile app for the restaurant billing system can enable customers to order food and make payments from their mobile devices. This feature can improve the customer experience and increase sales.
- Loyalty programs: Implementing a loyalty program that rewards customers for their frequent visits and purchases can help increase customer retention and encourage repeat business
- Analytics and reporting: Adding analytics and reporting features to the billing system can help restaurant owners track their sales, customer behavior, and revenue. This data can help them make informed decisions to improve their business.

REFERENCES

1. https://github.com/
2. https://chat.openai.com/
3. https://www.wikipedia.org/