A Mini Project Report
on

ONLINE FOOD ORDERING SYSTEM

By

SVS PRAHARSHITHA

1602-19-733-176

T. SAI VARSHINI

1602-19-733-160

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31

2020-21

ACKNOWLEDGEMENT

We would like to express our special thanks for gratitude of our teacher Mrs. P. Nagaratna hedge for the able guidance and support in completing my project and gave us this golden opportunity to do this project.

TABLE OF CONTENTS:

ABSTRACT:

Online Food Ordering System simplifies the food ordering process. The proposed system shows an user interface and update the menu with all available dishes. Customer can choose more than one item to make an order and can view order details before logging off. The order is placed in the queue. This system benefits both the customer and the business. Customers can easily customize their favorite dishes and place their order.

Our proposed system is an online food ordering system that enables ease for the customers. It overcomes the disadvantages of the traditional queueing system. Our proposed system is a medium to order online food hassle free from restaurants as well as mess service. This system improves the method of taking the order from customer. The online food ordering system sets up a food menu online and customers can easily place the order as per their wish. Also with a food menu, customers can easily track the orders. This system also provides a feedback system in which user can rate the food items. Also, the proposed system can recommend hotels, food, based on the ratings given by the user, the hotel staff will be informed for the improvements along with the quality. The payment can be made online or pay-on-delivery system. For more secured ordering separate accounts are maintained for each user by providing them an ID and a password.

INTRODUCTION:

Online food ordering system is a software which allows various restaurants to accept and manage orders through internet. It can be a website or a mobile app through which hungry customers can easily select their favourite dishes and place an order.

The online food ordering system sets up a food menu online and customers can easily place the order as per they like. Also with a food menu, online customers can easily track the orders. There are various facilities provided so that the users of the system will get service effectively. Again, the idea comes that mostly mess users are person who are shifted for various reason in new cities. So, they are interrelated. Increasing use of smart phones is also considered as a motivation, so that any users of this system get all service on single click. Another motivation can be considered as the system will be designed to avoid users doing fatal errors, users can change their own profile, users can track their food items through GPS, users can provide feedback and recommendations and can give ratings, it will give appropriate feedbacks to Restaurants. Due to lack of a full fledge application that can fulfil the customer requirements by providing him food from restaurants, there is a need for the system. This proposed system will be used by the people who keep shifting from cities to cites. As well as, it will be useful for the students studying in different cities. The proposed system will provide the flexibility to the Customers/Users to order from either Restaurants. It will also provide Recommendations to the customers from the restaurants/mess owners uploaded on a daily basis. In the proposed system, there will be no limitation on the amount of order the customer wants.

The proposed system is designed to avoid users doing fatal errors and inappropriate action. Scope of proposed system is justifiable because in large amount peoples are shifting to different cities so wide range of people can make a use of proposed system. The system/interface will take input from the user. The major attributes that will give input to the dataset are: name, address, email-Id, mobile no, other personal related values, etc. The output will include user/customer's Order, Bill, Feedback and Payment options. Initially there will be 4 restaurants. The reason why to choose this project is the idea behind project that is to solve problem of people which they are facing when they shift to different city. The system is not only for user but also for provider who provides food service. This system is for making efficient communication between consumer and producer of the food system.

Problem Statement:

Imagine you meet a small startup company planning to launch an Online Food Delivery System like Swiggy/Zomato that allows consumers & service providers to interact in real-time. They want to support both Mobile App and Web-based applications. Like many small start-ups, they are confident that they will be the next big thing and expect significant, rapid growth in the next few months. With this in mind, they are concerned about the following:

- Scaling to meet the demand, but they are not sure when and how the demand will grow — they are very concerned about buying too much infrastructure too soon or not enough too late!
- Effective distribution of load.
- The ability for Service Providers to send notifications to consumer.
- configurable database/s and data access layer to yield high performance and throughout.
- Allocate food delivery personal efficiently.
- Design for easy onboarding and searchability of restaurants.
- Prediction of order delivery time.

Purpose:

The purpose of this SRS is to outline both the functional and non-functional requirements of this subject. It is the intention that the presented set of requirements possesses the following qualities; correctness, unambiguousness, completeness, consistency, verifiability, modifiability and traceability. Consequently, the document should act as a foundation for efficient and well-managed project completion and further serve as an accurate reference in the future. To this audience group, this SRS

should convey and confirm the required functionality and represent a contractual agreement between the involved parties.

Scope:

In current formal dining environments, some form of physical static menu is utilised to convey the available food choices to customers. Said menus are generally paper based and hence impose restrictions on the textual real estate available and the ability a restaurateur has to update them. Three related concepts are encompassed by the general scope of the online food ordering system. The first pertains to the replacement of paper-based menus using an electronic format, the second relates to a complementary electronic strategy for the front of house handling of a customer's order and the third surrounds the process of transferring said electronic orders to the kitchen for preparation.

Overview:

The Online Food Ordering System is a software package to facilitate ordering within a traditional restaurant. The customer is able to view the menu, place orders and organise the final bill through the surface computer interface built into their table. The system contains full accountability and logging systems, and supports supervisor actions to account for exceptional circumstances, such as a meal being refunded or walked out on. Customers are presented with an attractive and easy-to-use surface computer GUI with a single option in their menus.

Benefits:

Greater flexibility in menus, an increase in restaurant productivity and capacity for extensive business auditing are the primary benefits associated with this system. Menu updates can be rolled out at any time with no extra labour from printing and distributing new menus, allowing for more dynamic pricing and content changes. With the underlying software system taking responsibility for a customer's order throughout its lifecycle, accuracy is ensured.

APLLICATION REQUIREMENT SPECIFICATION:

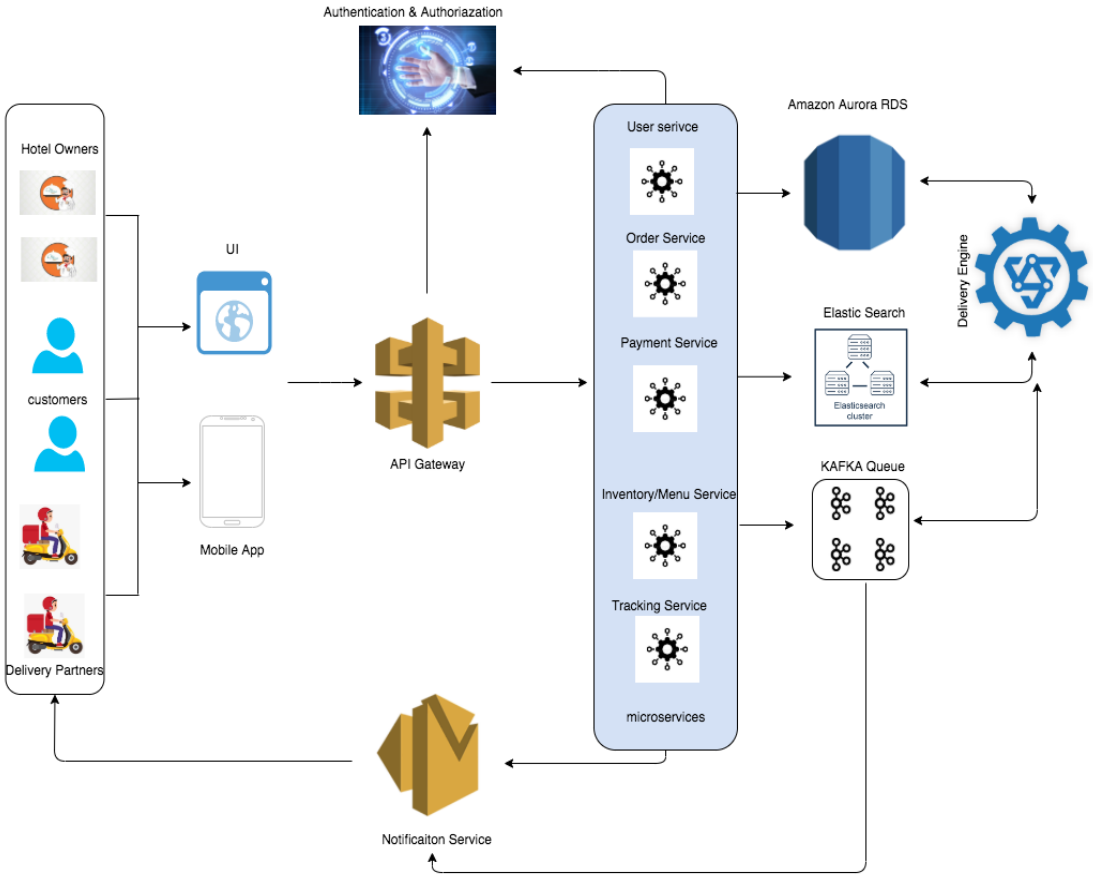
Hardware:

- Minimum RAM required: 512 mb
- Minimum disk space required: 50 mb
- Input devices: Mouse, Keyboard
- Output devices: Monitor

Software:

- Eclipse IDE
- Windows 7 or above
- JAVA platform SE 8 or above

SYSTEM ARCHITECTURE:



CODE:

Menu:

```
package miniProject;


import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import java.text.SimpleDateFormat;


public class restaurant
{
public void menu() {
while(true)
{
        System.out.println("Select a restaurant near your location:");
        System.out.println("1.Dominos Pizza");
        System.out.println("2.MC Donalds");
        System.out.println("3.Baskins and Robins");
        System.out.println("4.Cream stone");
        Scanner sc = new Scanner(System.in);
        int hotel;
        int totalAmount = 0;
        int item;
```

```
        hotel=sc.nextInt();
switch(hotel)
{
case 1:
        int flag=1;
        while(flag!=0)
        {
                System.out.println("select item from menu:");
                System.out.println("1.Margherita-Rs.200\n2.Farmhouse-
Rs.300\n3.PeppyPanner-Rs.400\n4.Mexican Green Wave-Rs.500\n ");
                Scanner s = new Scanner(System.in);
                item=s.nextInt();
                switch(item)
                {
                case 1:
                        System.out.println("Please enter the quantity");
                        int n = s.nextInt();
                        totalAmount = totalAmount + (n*200);
                        break;
                case 2:
                        System.out.println("Please enter the quantity");
                        int m = s.nextInt();
                        totalAmount = totalAmount + (m*300);
                        break;
                case 3:
```

```
            System.out.println("Please enter the quantity");
            int p = s.nextInt();
            totalAmount = totalAmount + (p*400);
            break;
case 4:
            System.out.println("Please enter the quantity");
            int q = s.nextInt();
            totalAmount = totalAmount + (q*500);
            break;
default:
           System.out.println("Invalid input");
}
System.out.println("Do you need anything else?\n0 or 1");
Scanner cs = new Scanner(System.in);
int more = cs.nextInt();
if(more == 0)
{
           flag=0;
           payment(totalAmount);
}
else
{
           flag=1;
}
//break;
```

```
        }
case 2:
        int flag1=1;
        while(flag1!=0)
        {
                System.out.println("select item from menu:");
                System.out.println("1.Maharaja Mac-Rs.195\n2.McSpicy Paneer
Burger-Rs.160\n3.Mexican Cheesy Friesr-Rs.122\n4.veg combo-Rs.500");
                Scanner s = new Scanner(System.in);
                item=s.nextInt();

        switch(item)
        {
        case 1:
                 System.out.println("Please enter the quantity");
                 int n = s.nextInt();
                 totalAmount = totalAmount + (n*195);
                 break;
        case 2:
                 System.out.println("Please enter the quantity");
                 int m = s.nextInt();
                 totalAmount = totalAmount + (m*160);
                 break;
        case 3:
                 System.out.println("Please enter the quantity");
```

```
                int p = s.nextInt();
                totalAmount = totalAmount + (p*122);
                break;
        case 4:
                System.out.println("Please enter the quantity");
                int q = s.nextInt();
                totalAmount = totalAmount + (q*500);
                break;
        default:
               System.out.println("Invalid input");
        }
        System.out.println("Do you need anything else?\n0 or 1");
        Scanner cs = new Scanner(System.in);
        int more = cs.nextInt();
        if(more == 0)
        {
                flag1=0;
                payment(totalAmount);
        }
        else
        {
                flag1=1;
        }
        //break;
}
```

```
case 3:
        int flag3=1;
        while(flag3!=0)
        {
                System.out.println("select item from menu:");
                System.out.println("1.Fresh Fruit Chikoo-Rs.281\n2.Honey nut
Crunch-Rs.334\n3.Pina Colada Ice Cream-Rs.329\n4.Hazelnut Rocher-Rs.500");
                Scanner s = new Scanner(System.in);
                item=s.nextInt();

        switch(item)
        {
        case 1:
                 System.out.println("Please enter the quantity");
                 int n = s.nextInt();
                 totalAmount = totalAmount + (n*281);
                 break;
        case 2:
                 System.out.println("Please enter the quantity");
                 int m = s.nextInt();
                 totalAmount = totalAmount + (m*334);
                 break;
        case 3:
                 System.out.println("Please enter the quantity");
                 int p = s.nextInt();
```

```
                totalAmount = totalAmount + (p*329);
                break;
        case 4:
                System.out.println("Please enter the quantity");
                int q = s.nextInt();
                totalAmount = totalAmount + (q*500);
                break;
        default:
               System.out.println("Invalid input");
        }
        System.out.println("Do you need anything else?\n0 or 1");
        Scanner cs = new Scanner(System.in);
        int more = cs.nextInt();
        if(more == 0)
        {
                flag1=0;
                payment(totalAmount);
        }
        else
        {
                flag1=1;
        }
        //break;
}
```

```
case 4:
        int flag4=0;
        while(flag4!=0)
        {
                System.out.println("select item from menu:");
                System.out.println("1.Malai Tub-Rs.100\n2.Willi wonka-
Rs.200\n3.Rich Dark Chocolate-Rs.399\n4.Oreo Shot-Rs.150");
                Scanner s = new Scanner(System.in);
                item=s.nextInt();

        switch(item)
        {
        case 1:
                 System.out.println("Please enter the quantity");
                 int n = s.nextInt();
                 totalAmount = totalAmount + (n*100);
                 break;
        case 2:
                 System.out.println("Please enter the quantity");
                 int m = s.nextInt();
                 totalAmount = totalAmount + (m*200);
                 break;
        case 3:
                 System.out.println("Please enter the quantity");
                 int p = s.nextInt();
```

```
            totalAmount = totalAmount + (p*399);
            break;
case 4:
            System.out.println("Please enter the quantity");
            int q = s.nextInt();
            totalAmount = totalAmount + (q*150);
            break;
default:
           System.out.println("Invalid input");
}
System.out.println("Do you need anything else?\n0 or 1");
Scanner cs = new Scanner(System.in);
int more = cs.nextInt();
if(more == 0)
{
           flag1=0;
           payment(totalAmount);
}
else
{
           flag1=1;
}
//break;
}
default :
```

```
                System.out.println("Invalid input");
                break;
}
}
}

public void payment(int totalAmount)
{
        double taxOnGood=0.0;
        double initialBill=0;
        double totaltax=0;

        if(totalAmount<=1000)
        {
                taxOnGood=0.05;
                totaltax=0.05*totalAmount;
                initialBill=totaltax+totalAmount;
                System.out.println("Total is: "+ initialBill);
        }
        else if(totalAmount>1000 && totalAmount<=5000) {
                taxOnGood=0.1;
                totaltax=0.1*totalAmount;
                initialBill=totaltax+totalAmount;
                System.out.println("Total is: "+ initialBill);
        }
```

```
        else
        {
                taxOnGood=1.8;
                totaltax=1.8*totalAmount;
                initialBill=totaltax+totalAmount;
                System.out.println("Total is: "+ initialBill);
        }
        delivery(initialBill);

}
public void delivery(double initialBill)
{
        int deliveryCharges = 50;
        double totalBill=deliveryCharges + initialBill;
        System.out.println("Total bill to be paid with delivery charges is: "+ totalBill);
        paymentMode();
        System.exit(0);
}
static boolean flag1 = true;

public static boolean isValidCardNumber()
{
        Scanner sc = new Scanner(System.in);
        String cardNumber = sc.nextLine();
        Pattern p = Pattern.compile("(?:\\d[ -]*?){13,16}");
```

```
        Matcher m = p.matcher(cardNumber);
        flag1=m.find() && m.group().equals(cardNumber);
    return (m.find() && m.group().equals(cardNumber));

}

public static boolean isValidCvv()
{
        Scanner sc = new Scanner(System.in);
        String cvv = sc.nextLine();
        Pattern p = Pattern.compile("^[0-9]{3,4}$");
        Matcher m = p.matcher(cvv);
        flag1=m.find() && m.group().equals(cvv);
    return (m.find() && m.group().equals(cvv));

}

public static boolean isValidDate()
{
        Scanner sc = new Scanner(System.in);
        String expiryDate = sc.nextLine();
        Pattern p = Pattern.compile("(0[1-9]|1[0-2])\\/?(([0-9]{4})|[0-9]{2}$)");
        Matcher m = p.matcher(expiryDate);
        flag1=m.find() && m.group().equals(expiryDate);
    return (m.find() && m.group().equals(expiryDate));
```

```
}
```

Login Details:

```
package miniProject;
import java.util.Scanner;
import java.util.regex.*;
class LoginDetails
{
String number = null;
   static boolean flag=true;

public static boolean isValid(String number)
{
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your phone number:");
        number = sc.nextLine();
        Pattern p = Pattern.compile("(?:0\\/91)?([7-9][0-9]{9})");
        Matcher m = p.matcher(number);
        flag=m.find() && m.group().equals(number);
   return (m.find() && m.group().equals(number));

}
public String phoneNumber()
{
        int i=0;
```

```
        if (isValid(number))
        {
                System.out.println("Valid Number");
        }
        else
        {
                while(!flag)
                {
                        System.out.println("Enter phone number again cuz it was
invalid: ");
                        isValid(number);
                        //continue;

                }
        }
        return number;
}

}
class emailIdDetails
{
static String email= null;
static boolean flag1=true;
```

```
public  boolean isValid(String email)
   {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your emailId:");
        email = sc.nextLine();
      String emailRegex ="^.+[^@]+@[A-z]+\\.(?:)com$";
      Pattern pat = Pattern.compile(emailRegex);
      if (email == null)
         return false;
      flag1=pat.matcher(email).matches();
      return pat.matcher(email).matches();
   }

public String emailId()
{
if (isValid(email))
      System.out.println("valid information");
   else
   {
        while(!flag1)
        {
                System.out.println("Enter email again cuz it was invalid: ");
                isValid(email);
                //continue;
```

```
            }


    }

return null;
}
}
class passwordDetails
{
static String password="";
static boolean flag2=true;


public static boolean isValidPassword(String password)
    {
            Scanner ps = new Scanner(System.in);
            System.out.println("Enter your password:");
            password = ps.nextLine();
        String regex = "^(?=.*[0-9])"
                    + "(?=.[a-z])(?=.[A-Z])"
                    + "(?=.*[@#$%^&+=])"
                    + "(?=\\S+$).{8,20}$";
        Pattern p = Pattern.compile(regex);


        if (password == null) {
            return false;
```

```
        }
        //flag2=m.matches();
        Matcher m = p.matcher(password);
        return m.matches();
    }
public String passwordvalid()
{
        //String password="";
        if (isValidPassword(password))
        System.out.println("Valid password");
    else
    {
        while(!flag2)
                {
                        System.out.println("Enter password again cuz it was invalid: ");
                        isValidPassword(password);
                }
        //return null;

}
        return password;

}
}
class nameDetails
```

```
{
Scanner sc = new Scanner(System.in);
nameDetails()
{
        System.out.println("Enter your name:");
        String name = sc.nextLine();
}

}


class addressDetails{

Scanner sc = new Scanner(System.in);
public void address(){

        System.out.println("Enter your Address:");
        String address = sc.nextLine();

}

}


public class onlineFoodOrdering extends restaurant
```

```
{
public static void main(String[] args)
{

        LoginDetails l = new LoginDetails();
        emailIdDetails e = new emailIdDetails();
        nameDetails n = new nameDetails();
        passwordDetails p= new passwordDetails();
        l.phoneNumber();
        e.emailId();
        p.passwordvalid();
        addressDetails ad = new addressDetails();
        ad.address();
        restaurant res= new restaurant();
        res.menu();

}
}
```

Payment Mode:

```
public void paymentMode(){

        Scanner sc = new Scanner(System.in);
        System.out.println("Select the mode of payment:");
        System.out.println("1.Cash on delivery\n2.Card\n3.PayU");
```

```
int mode = sc.nextInt();
switch(mode)
{
case 1:
        System.out.println("Please keep the exact change ready");
        break;
case 2:
        System.out.println("Enter the Card Details:");
        System.out.println("Enter the Card Number:");
        if(isValidCardNumber())
        {
                System.out.println("Proceeding further...");
        }
        else
        {
                while(!flag1)
                {
                        System.out.println("Please enter the valid number");
                        if(isValidCardNumber())
                        {
                                break;
                        }

                }
        }
```

```
System.out.println("Enter the name of card holder:");
Scanner sc1 = new Scanner(System.in);
String name = sc1.nextLine();
System.out.println("Enter the CVV:");
if(isValidCvv())
{
	System.out.println("Proceeding further...");
}
else
{
	while(!flag1)
	{
		System.out.println("Please enter the valid number");
		if(isValidCvv())
		{
			break;
		}

	}
}
System.out.println("Enter the expiry date:");
if(isValidDate())
{
	System.out.println("Card Details Verified");
}
```

```
else
{
    while(!flag1)
    {
        System.out.println("Please enter the correct expiry
date");
        if(isValidDate())
        {
            break;
        }

    }
}
System.out.println("Please wait...");
System.out.println("Proceeding for payment");
try
{
    TimeUnit.SECONDS.sleep(20);
}
catch(Exception e)
{
    System.out.println("Exception handled");
}

System.out.println("Payment successful");
```

```
                break;
        case 3:
                System.out.println("Bill payment proceeding through PayU...");
                try
                {
                        TimeUnit.SECONDS.sleep(30);
                }
                catch(Exception e)
                {
                        System.out.println("Exception handled");
                }

                System.out.println("Payment successful");
                break;
        default:
                System.out.println("Invalid input");
                break;
        }

        SimpleDateFormat formatter = new SimpleDateFormat("HH:mm:ss");
        System.out.println("Your order will be delivered within 40 minutes by
"+formatter.format(System.currentTimeMillis()+(40*60*1000)));



}
```

```
}
```

OUTPUT:

Payment Through Cash:

Enter your name:
Varshini
Enter your phone number:
9999998888
Enter your emailId:
saivarshini98gmail
Enter email again cuz it was invalid:
Enter your emailId:
saivarshini@gmail.com
Enter your password:
varsh14
Enter your Address:
Sri Sai Enclave-2,Boduppal,Hyderabad
Select a restaurant near your location:
1.Dominos Pizza
2.MC Donalds
3.Baskins and Robins
4.Cream stone
1
select item from menu:
1.Margherita-Rs.200
2.Farmhouse-Rs.300
3.PeppyPanner-Rs.400
4.Mexican Green Wave-Rs.500

1
Please enter the quantity
2
Do you need anything else?
0 or 1
1
select item from menu:
1.Margherita-Rs.200
2.Farmhouse-Rs.300
3.PeppyPanner-Rs.400
4.Mexican Green Wave-Rs.500

4
Please enter the quantity
3
Do you need anything else?

0 or 1
0
Total is: 2090.0
Total bill to be paid with delivery charges is: 2140.0
Select the mode of payment:
1.Cash on delivery
2.Card
3.PayU
1
Please keep the exact change ready
Your order will be delivered within 40 minutes by 21:13:10

Payment Through Card:

Enter your name:
Praharshitha
Enter your phone number:
9875558809
Enter your emailId:
svs123@gmail.com
valid information
Enter your password:
svs65
Enter your Address:
Tricolor aparments,Narayanaguda
Select a restaurant near your location:
1.Dominos Pizza
2.MC Donalds
3.Baskins and Robins
4.Cream stone
2
select item from menu:
1.Maharaja Mac-Rs.195
2.McSpicy Paneer Burger-Rs.160
3.Mexican Cheesy Friesr-Rs.122
4.veg combo-Rs.500
1
Please enter the quantity
3
Do you need anything else?
0 or 1
1
select item from menu:
1.Maharaja Mac-Rs.195

2.McSpicy Paneer Burger-Rs.160
3.Mexican Cheesy Friesr-Rs.122
4.veg combo-Rs.500
2
Please enter the quantity
1
Do you need anything else?
0 or 1
0
Total is: 782.25
Total bill to be paid with delivery charges is: 832.25
Select the mode of payment:
1.Cash on delivery
2.Card
3.PayU
2
Enter the Card Details:
Enter the Card Number:
879023
Please enter the valid number
4567 8976 3345 8734
Enter the name of card holder:
SVS Praharshitha
Enter the CVV:
8765
Enter the expiry date:

Please enter the correct expiry date
22/22
Please enter the correct expiry date
12/23
Please wait...
Proceeding for payment
Payment successful
Your order will be delivered within 40 minutes by 21:24:20

Payment Through PayU:

Enter your name:
Prahari
Enter your phone number:
2356748
Enter phone number again cuz it was invalid:

Enter your phone number:
9885555555
Enter your emailId:
prahari@gmail.com
valid information
Enter your password:
svschubby
Enter your Address:
Avatar apartments, Uppal, Hyderabad
Select a restaurant near your location:
1.Dominos Pizza
2.MC Donalds
3.Baskins and Robins
4.Cream stone
3
select item from menu:
1.Fresh Fruit Chikoo-Rs.281
2.Honey nut Crunch-Rs.334
3.Pina Colada Ice Cream-Rs.329
4.Hazelnut Rocher-Rs.500
2
Please enter the quantity
4
Do you need anything else?
0 or 1
1
select item from menu:
1.Fresh Fruit Chikoo-Rs.281
2.Honey nut Crunch-Rs.334
3.Pina Colada Ice Cream-Rs.329
4.Hazelnut Rocher-Rs.500
4
Please enter the quantity
2
Do you need anything else?
0 or 1
1
select item from menu:
1.Fresh Fruit Chikoo-Rs.281
2.Honey nut Crunch-Rs.334
3.Pina Colada Ice Cream-Rs.329
4.Hazelnut Rocher-Rs.500
1
Please enter the quantity
1
Do you need anything else?

0 or 1
0
Total is: 2878.7
Total bill to be paid with delivery charges is: 2928.7
Select the mode of payment:
1.Cash on delivery
2.Card
3.PayU
3
Bill payment proceeding through PayU...
Payment successful
Your order will be delivered within 40 minutes by 21:29:42

CONCLUSION and FUTURE WORK:

Therefore, conclusion of the proposed system is based on user's need. The system is developed in considering all issues related to all user which are included in this system. Wide range of people can use this if they know how to operate android smart phone. Various issues related to Food Service will be solved by providing them a full-fledged system. Thus, implementation of Online Food Ordering system is done to help and solve one of the important problems of people. Based on the result of this research, it can be concluded: It helps customer in making order easily; It gives information needed in making order to customer. The Food website application made for restaurant can help restaurant in receiving orders and modifying its data and it is also made for admin so that it helps admin in controlling all the Food system. With online food ordering system, a restaurant online can be set up and the customers can easily place order. Also with a food menu online, tracking the orders is done easily and improve the food delivery service. The restaurants can even customize online restaurant menu. Having a restaurant menu on internet, potential customers can easily access it and place order at their convenience. Thus, an automated food ordering system is presented with features wireless communication. The proposed system would attract customers and adds to the efficiency of maintaining the restaurant and billing sections.

REFERENCES:

[1] Ashutosh Bhargave, Niranjan Jadhav, Apurva Joshi, Prachi Oke, S. R Lahane,"Digital Ordering System for Restaurant Using Android", International Journal of Scientific and Research Publications 2013.

[2] https://www.geeksforgeeks.org/write-regular-expressions/