
Online Food Plaza

1906802

1906874

T.Y. B.Sc. IT (Semester VI)

Academic Year 2019-2020



K. J. SOMAIYA COLLEGE OF SCIENCE AND COMMERCE



Re-accredited with 'A' Grade by NAAC
Vidyanagar, Vidyavihar, Mumbai 400 077

DEPARTMENT OF INFORMATION TECHNOLOGY
PROJECT CERTIFICATE FOR B.Sc. (I.T.) STUDENTS

2019 – 2020

This is to certify- that the project entitled

'Online Food Plaza'

Undertaken by

Sister. **RUTUJA BANE(19-06802)**

Sister. **PRIYANKA SONI(19-06874)**

In fulfilment for B.Sc. I.T. Degree (Semester 6) Examination has been completed by him/her. This project had not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

This is to further certify that he/she has completed all required phases of the project

(Ms. Anushree Sukhi)

Project Guide

(Mr. Prakash Patil)

Head of the Department

College Seal

External Examiner

ACKNOWLEDGEMENT:

The success of this project would not have been possible without the kind support and assistance of many individuals and organization, and we are immensely blessed to have got this all along the duration of our project. We would like to extend our profound gratitude to each and every of them.

We are highly indebted to KJ SOMAIYA COLLEGE OF SCIENCE AND COMMERCE for constant guidance and supervision, as well as for providing all the necessary ICT infrastructure and friendly environment for the successful completion of the project. We are also appreciative of the efforts of B.Sc. (IT) coordinator **Mrs. MINAL D.** without his supporting role, the project would have been nowhere near completion.

We would like to express our gratitude to out project supervisor **MS. ANUSHREE SUKHI** who interest on our project and guided us throughout the project by providing all the necessary idea, information and knowledge for the developing a functional android application.

We are thankful and fortunate enough to get constant support from our seniors and every teaching staff of B.Sc. (IT) department which helped us successfully complete our project. We would also like to extend our regards to all the non-teaching staff of B.Sc.-IT department for their timely support.

Last but not the least, our thanks and appreciations also go to each and every one of our friends for their encouragement and support in developing the project.

RUTUJA BANE.

PRIYANKA SONI.

TABLE OF CONTENTS

Sr. No.	Particular	Page No.
1	Chapter 1 : Introduction	8 – 10
1.1	Background	9
1.2	Objectives	9
1.3	Purpose , Scope and Applicability	10
2	Chapter 2 : Survey of Technologies	11-12
3	Chapter 3 : Requirements and Analysis	13-29
3.1	Problem Definition	14
3.2	Requirements Specification	14-15
3.3	Planning and Scheduling	16
3.4	Software and Hardware Requirements	16
3.5	Preliminary Product Description	17
3.6	Conceptual Models	
	3.6.1 Use case Diagram	17-22
	3.6.2 E-R Diagram	23
	3.6.3 Data Flow Diagram	24-26
	3.6.4 Flowchart	27
	3.6.5 Class Diagram	28-29
4	Chapter 4 : System Design	30
4.1	Basic Modules in project	31

4.2	Data Design	31-34
4.3	User Interface Design screenshots	35-45
4.4	Security Issues	45-46
4.5	Test Case Design	46
5	Chapter 5 : Future Enhancements	47
5.1	Implementation Approaches	48
5.2	Coding Details and Code Efficiency	48-53
	5.2.1 Code Efficiency	53
5.3	Testing Approach	54
	5.3.1 Unit Testing	54
	5.3.2 Integrated Testing	55
	5.3.3 Performance Testing	55
5.4	Modifications and Improvements	56
5.5	Test Cases	56
6	Chapter 6: Results and Discussion	57
6.1	6.1 Test Reports	58
6.2	User Documentation	58-59
7	Chapter 7: Conclusions	60
7.1	Conclusion	61
	7.1.1 Significance of the System	61
7.2	Limitations of the System	61-62
7.3	Future Scope of the Project	62
	REFERENCES	63
	GLOSSARY	64

Table of Figures

Fig No.	Particular	Page No.
Fig 3.1	Gantt Chart	16
Fig 3.2	Use case Diagram	17
	Fig 3.2.1 Login Module Use case	17
	Fig 3.2.2 Sign Up Module Use case	18
	Fig 3.2.3 Customer Home Page Use case	19
	Fig 3.2.4 Admin Home Page Use case	20
	Fig 3.2.5 My Cart Use case	21
	Fig 3.2.6 Manage Food Use case	22
Fig 3.3	E-R Diagram	23
Fig 3.4	Data Flow Diagram	24
	Fig 3.4.1 Context level DFD	24
	Fig 3.4.2 0 level DFD(User)	24
	Fig 3.4.3 0 level DFD (Admin)	25
	Fig 3.4.4 1 level DFD User	26
Fig 3.5	Flowchart	27
Fig 3.6	Class Diagram	28
Fig 3.7	Spiral Model	29
Fig 4.1	User Interface	35
	Fig 4.1.1 Home Page Header	35
	Fig 4.1.2 Home Page Footer	35
	Fig 4.1.3 Login Page	36
	Fig 4.1.4 Invalid Admin Login	36

Fig 4.1.5 Invalid Customer Login	37
Fig 4.1.6 Register Page	37
Fig 4.1.7 Customer Home Page After Login	38
Fig 4.1.8 Show Food List	39
Fig 4.1.9 Update Profile	39
Fig 4.1.10 After Updating Profile	40
Fig 4.1.11 My Cart	40
Fig 4.1.12 After deleting pasta from cart	41
Fig 4.1.13 Admin Home Page	41
Fig 4.1.14 Add Food Page	42
Fig 4.1.15 After French Fries is added	43
Fig 4.1.16 Updating French Fries Cost	43
Fig 4.1.17 After Updating French Fries price	44
Fig 4.1.18 Deleting Food	44
Fig 4.1.19 Customer List	45

Chapter 1:

Introduction Chapter

INTRODUCTION:

Food ordering website is a website which help restaurants may be large or small to optimize and have complete control over their business and customers. Nowadays these websites are grabbing the market like anything. This website helps the restaurants to do all functionalities more accurately and faster way. It reduces the manual works and improves the efficiency of restaurants. This website is helping food orderings to maintain the stock and cash flows. The website helps food orders to maintain day to day records in the system. It's very useful in keeping a proper record of the database.

1.1 BACKGROUND:

This Case study looks at the problem of setting up a fast food restaurant. In existing system there are few problems:

- For placing any orders customers have to visit hotels or restaurants to know about food items and then place order and pay. In this method time and manual work is required.
- While placing an order over the phone, customer lacks the physical copy of the menu item, lack of visual confirmation that the order was placed correctly.
- Every restaurant needs certain employees to take the order over phone or in-person, to offer a rich dining experience and process the payment. In today's market, labor rates are increasing day by day making it difficult to find employees when needed.

Hence, to solve this issue, what I propose is an "Online Food Order System, originally designed for small scale business like College Cafeterias, Fast Food restaurant or Take-Out, but this system is just as applicable in any food delivery industry.

The main advantage of my system is that it greatly simplifies the ordering process for both the customer and the restaurant and also greatly lightens the load on the restaurant's end, as the entire process of taking orders is automated.

Anticipated Benefits are:

1. This will minimize the number of employees at the back of the counter.
2. The system will help to reduce labor cost involved.
3. The system will be less probable to make mistake, since it's a machine.

4. This will avoid long queues at the counter due to the speed of execution and number of optimum screens to accommodate the maximum throughput.

1.2 OBJECTIVE:

The main objective of this project is to computerize the manual system and reduce the time consumption. In other words, we can say that our project has the following objectives: -

Make all the system computerize

- Reduce time consumption
- Reduce error scope
- All system managements are automated
- Centralized database management
- Easy operations for operator of the system
- No paper work requirement.

1.3 PURPOSE, SCOPE AND APPLICABILITY:

1.3.1 . PURPOSE:

Manage and add food item and food item related informationally.

Having an up-to-date website, though isn't just about appearances. It's more about communicating with our users and surrounding community.

In this database will be maintained by the database management system. The system can easily access data from the database.

The Administrator can easily access and update the data on the website.

The main purpose of this project is to get the satisfaction with the user by providing this platform.

1.3.2. SCOPE:

More than one people can access the food information from the website.

All users allocated with food related information.

There are many features which helps user and administrator keep record of available information.

Home based search for internet search engines in order to find all type of food items.

1.3.2 APPLICABILITY:

Everyone can use this website for getting food information and to order food items.

The administrator can keep the track of the user database.

The user can avail the information regarding the food quantity.

Chapter 2:

Survey of Technologies

SURVEY OF TECHNOLOGIES:

❖ JAVA

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

❖ HTML

Hypertext Mark-up Language (HTML) is the standard markup for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

❖ PHP

PHP is a server-side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Preprocessor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only.

❖ SQL Database

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as updating data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

Chapter 3:

Requirements and Analysis

3.1 PROBLEM DEFINITION:

Main idea is to design a web application which is useful for users to buy food items through website. Using this application we provide all features for users to search for products, add product to cart, purchase using online methods like net banking, credit card, debit card.

3.2 REQUIREMENTS SPECIFICATION:

A .FUNCTIONAL REQUIREMENTS:

USER:

- The user shall be able to login to the website.
- The user shall be able to view food item information.
- The user shall be able to navigate from one page to another.
- The user shall be able to locate different categories.
- The user shall be able to add food item to cart.
- The user shall be able to view price.

PAYMENT:

- Once food item is added to the cart, Payment option will appear.
- After clicking on payment, the user will be redirected to payment module.
- Once Payment module is open options for payments appears like credit card, PAYTM.
- Payment can be done with the help of credit card.
- It can also be done with the help of PAYTM.

SIGN UP:

Main idea is to design a web application which is useful for users to buy products through website. Using this application we provide all features for users to search for products, add product to cart, purchase using online methods like net banking, credit card, debit card.

- After clicking on register in the listing page the user will be redirected to the signup page.
- Signup includes user credentials like username, name, email id, contact, Address.
- After the login is done user module gets open.

ADMIN:

- The admin shall be able to login to the website.
- The admin shall be able add food item and images to the website.
- The admin shall be able to remove user and food item from database.
- The admin shall be able to view applied forms from the customer.
- The admin shall be able to change the password.

B. NON FUNCTIONAL REQUIREMENTS:

AVAILABILITY:

The website should be made available 24 hours per week.

SECURITY:

The website should encrypt user passwords.

MAINTAINABILITY:

The website should be closed once in 6 months for maintenance and elaboration.

RELIABILITY:

The website should never crash more than 10 minutes per month.

DESIGN CONSTRAINS:

The website design should give an attractive look.

PLANNING AND SCHEDULING:

Task	Date Start	End Date	Status
Requirement gathering	23 rd July	30 th July	Completed
Finalize UI and use case diagram	1 st August	9 th August	Completed
Making ER diagram	13 th August	25 th August	Completed
Login module	28 th August	15 th September	Completed
Manage food Module	20 th September	1 st October	Completed
Manage customer Module	12 th October	15 th November	Completed
Manage cart Module	16 th November	1st December	Completed
Update Customer Module	3 rd December	25 th December	Completed
Testing validation and deployment of application	26 th December	5 th January	Completed

Fig 3.1 Gantt chart

3.4 SOFTWARE AND HARDWARE REQUIREMENTS:

SOFTWARE REQUIREMENTS:

- Windows 7 or more
- Front-End :- HTML , CSS
- Back-End :- PHP , MySQL
- Programming Language: Java

HARDWARE REQUIREMENTS:

- GB RAM
- i3 processor or above
- 50 GB Hard disk drive

3.5 PRELIMINARY PRODUCT DESCRIPTION:

Online ordering system that I am proposing here greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

3.6 CONCEPTUAL MODELS:

A conceptual model is are presentation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents. It is also a set of concepts. Some models are physical objects; for example, a toy model which may be assembled, and maybe made to work like the object it represents.

3.6.1. USECASE DIAGRAM:

LOGIN:

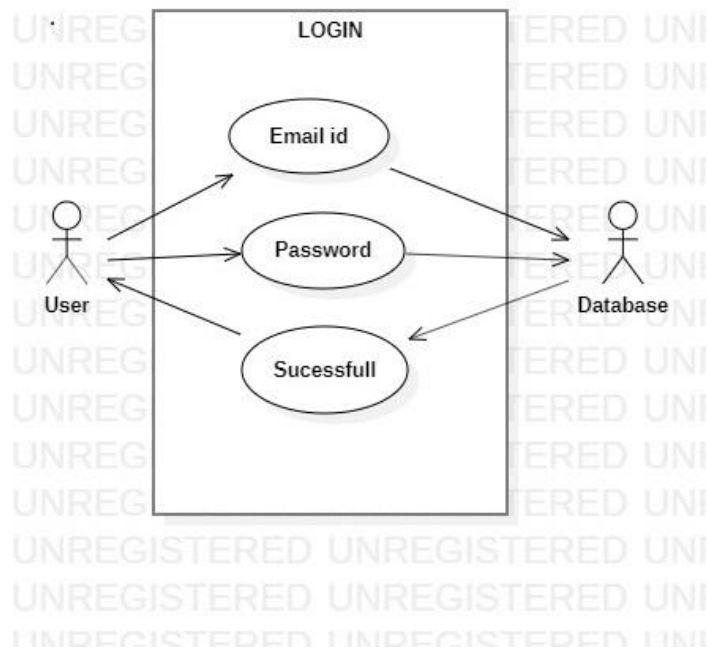


Fig 3.2.1 login module

SIGN UP:

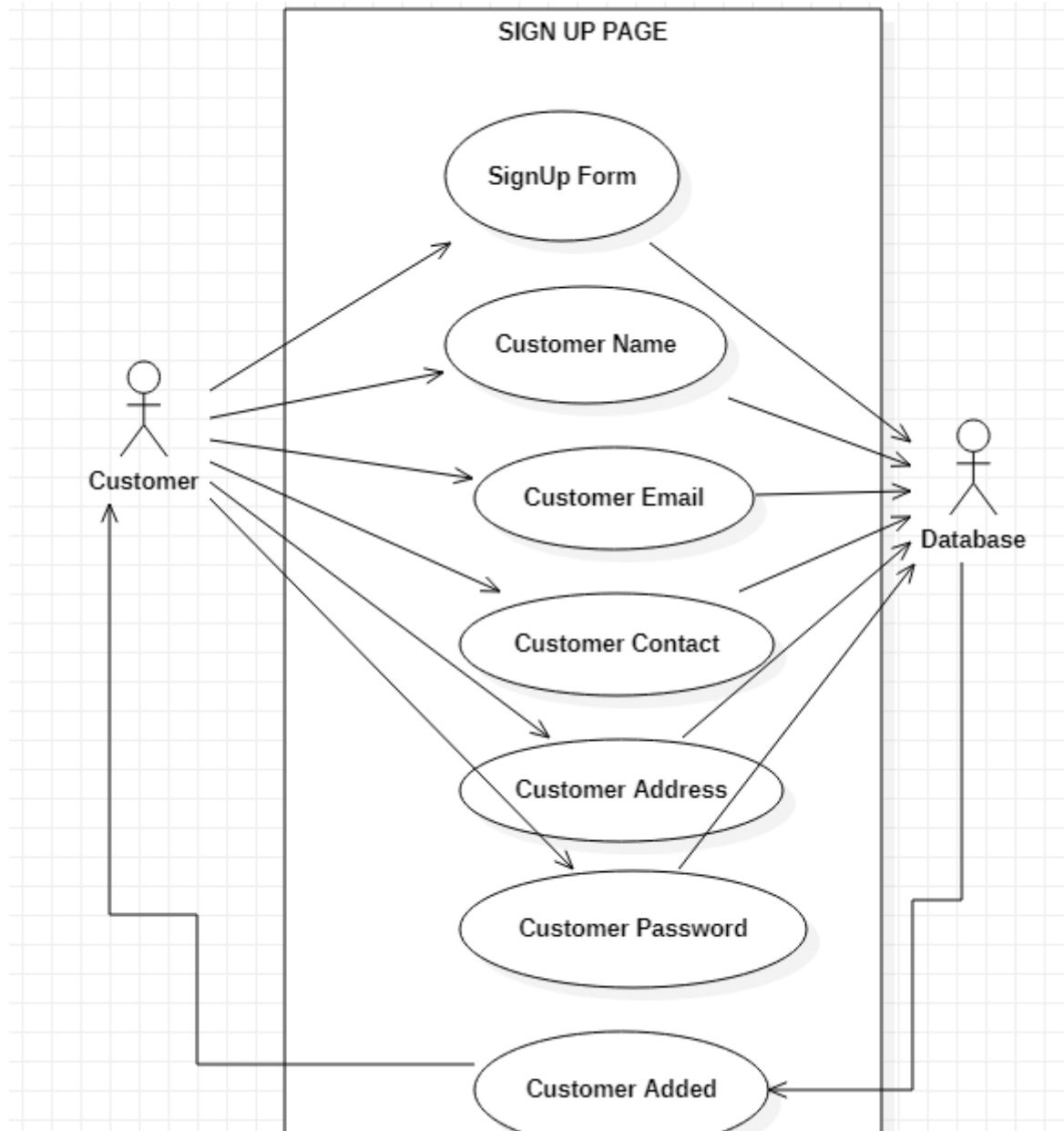


Fig 3.2.2

CUSTOMER HOME PAGE:

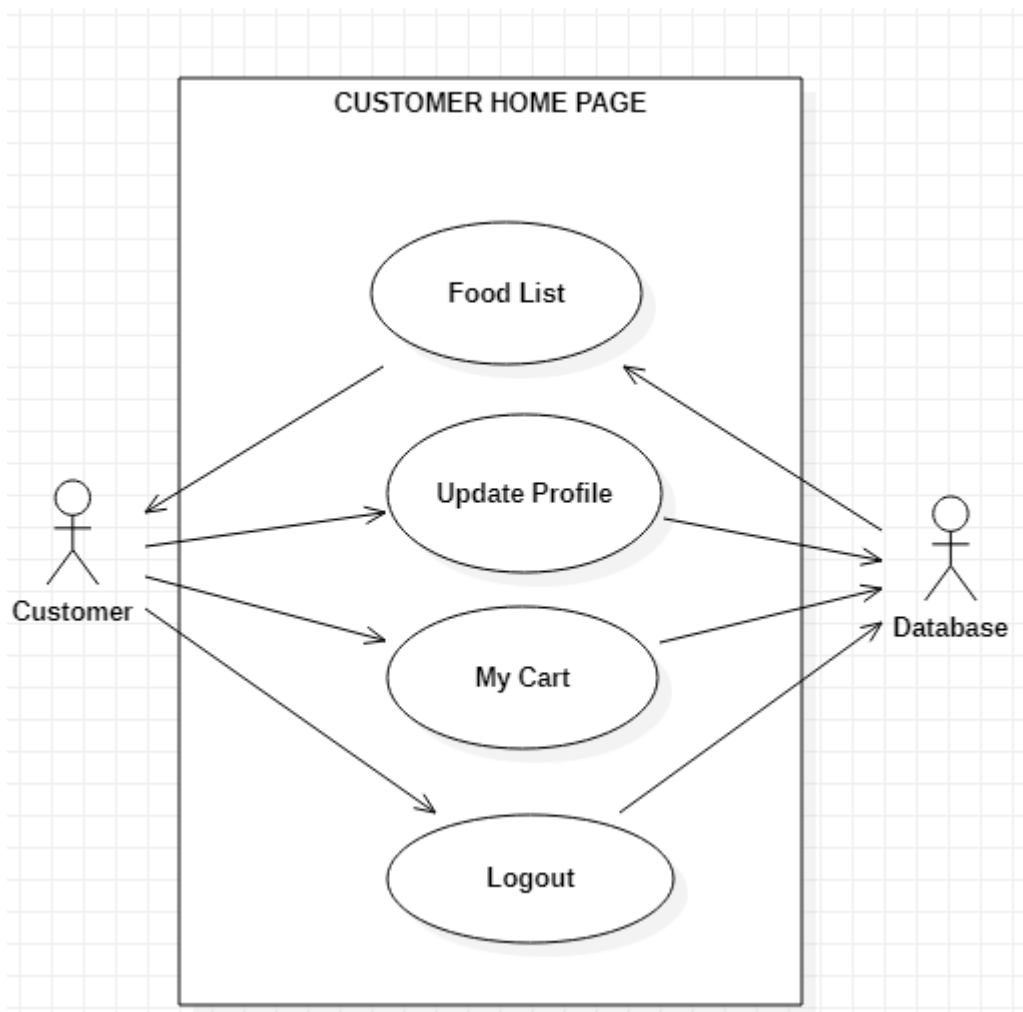


Fig 3.2.3

ADMIN HOME PAGE:

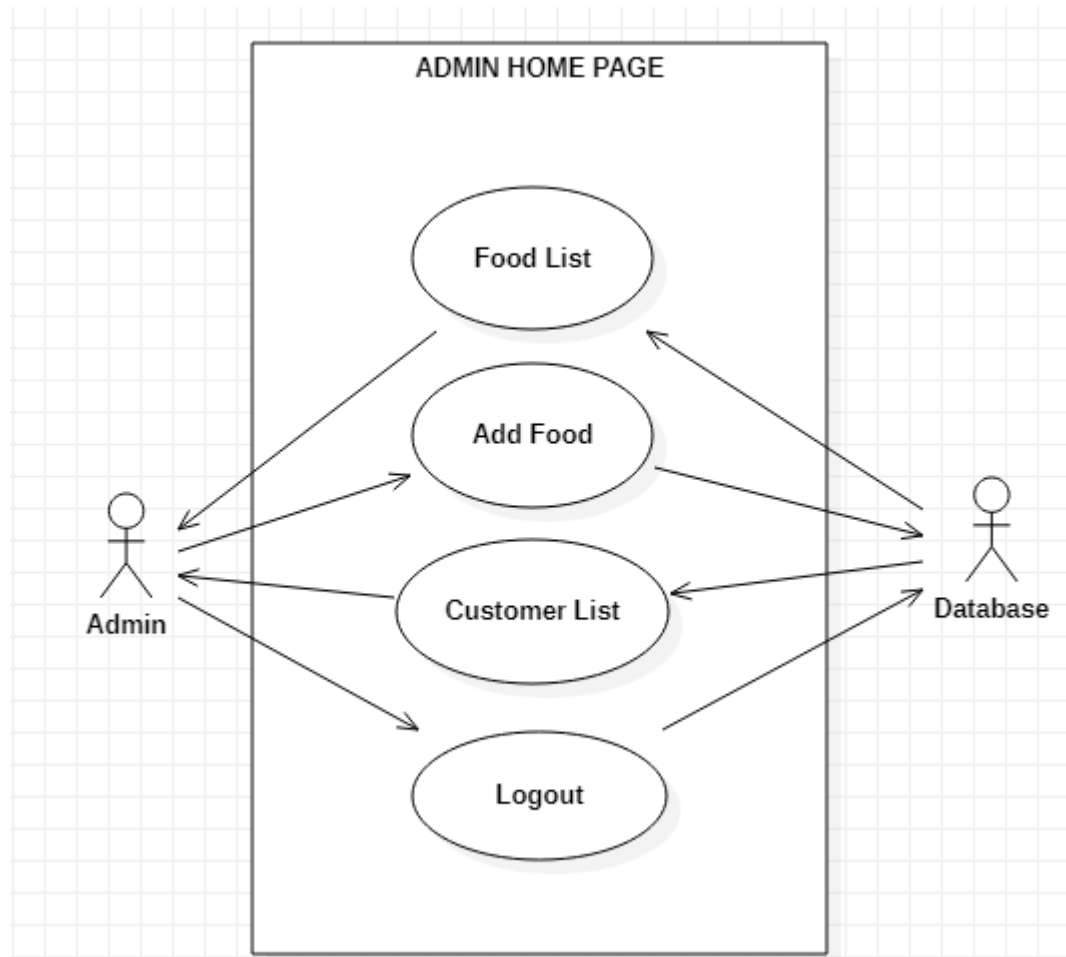


Fig 3.2.4

MY CART:

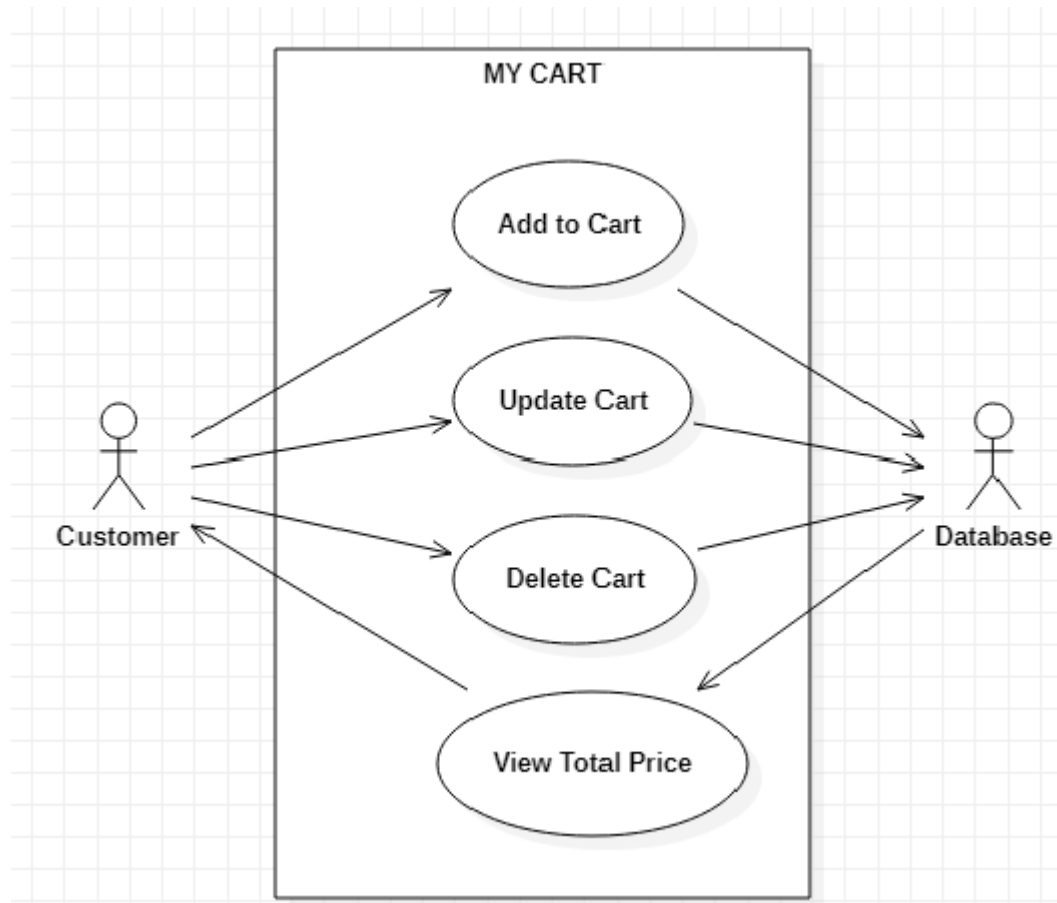


Fig 3.2.5

MANAGE FOOD:

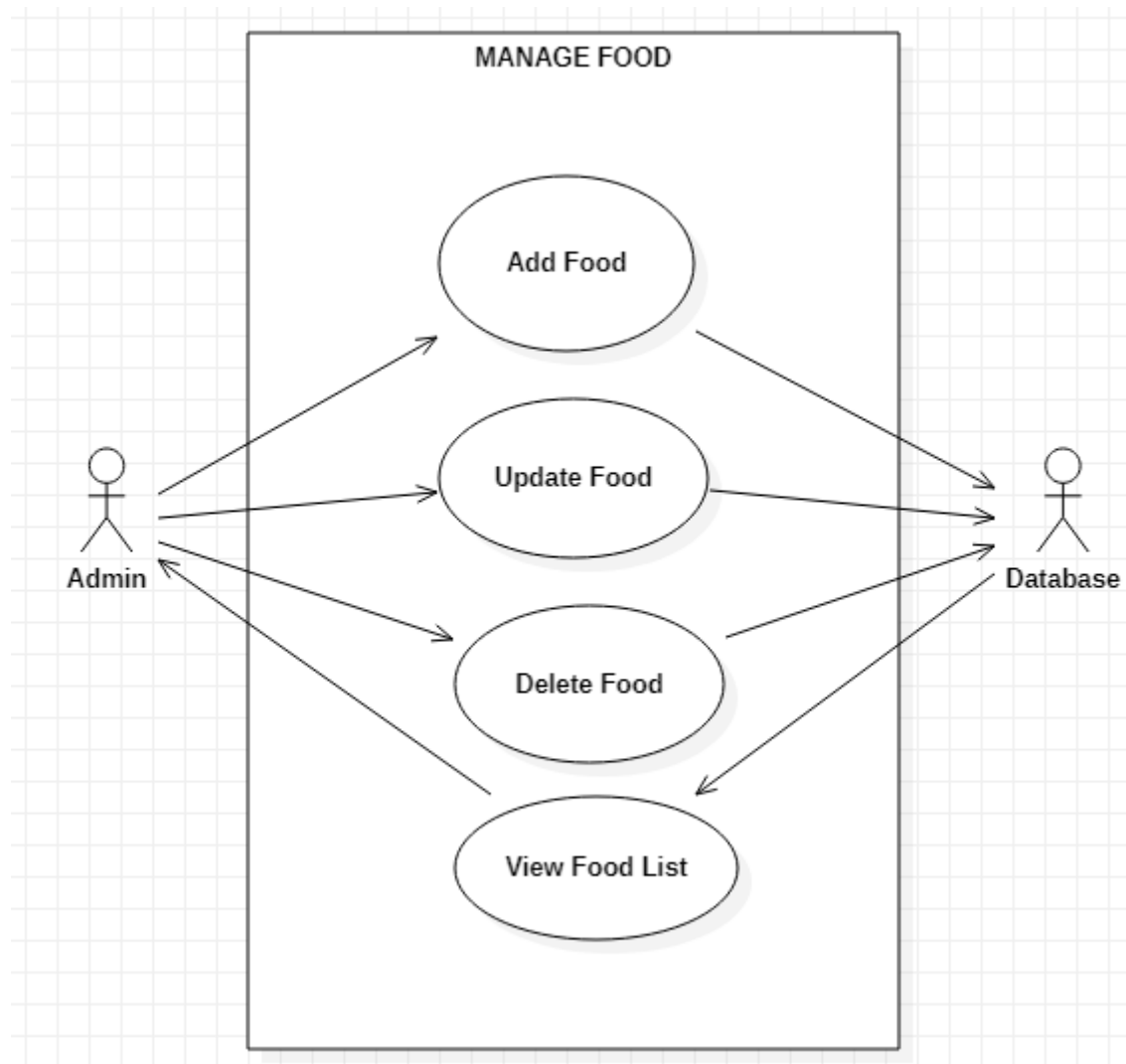


Fig 3.2.6

3.6.2. E-R DIAGRAM:

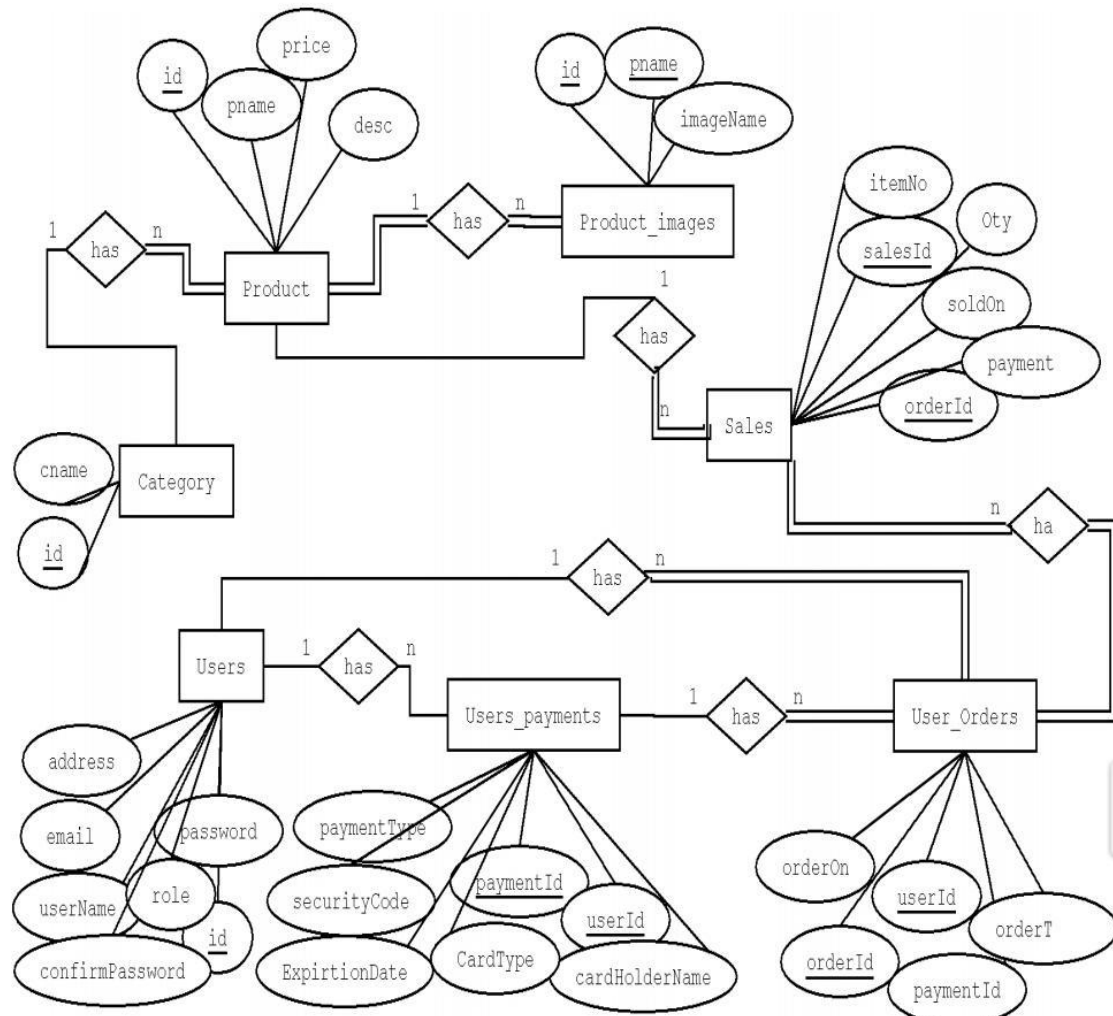


Fig 3.3

3.6.3 Context Level Diagram

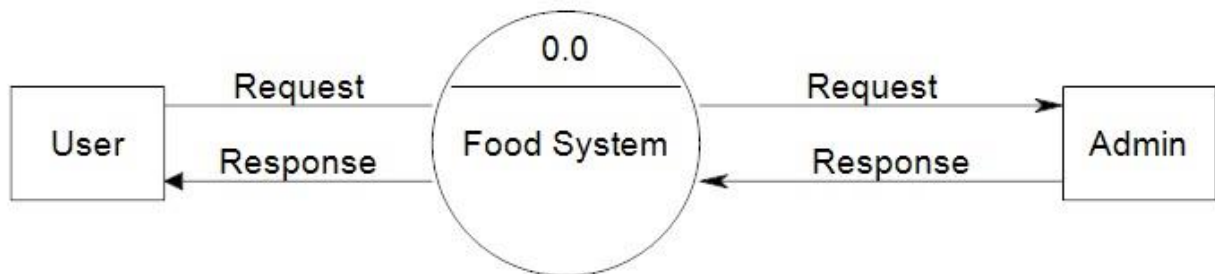


Fig 3.4.1 Context Level DFD

Level 0 DFD (Identifies manageable sub process)

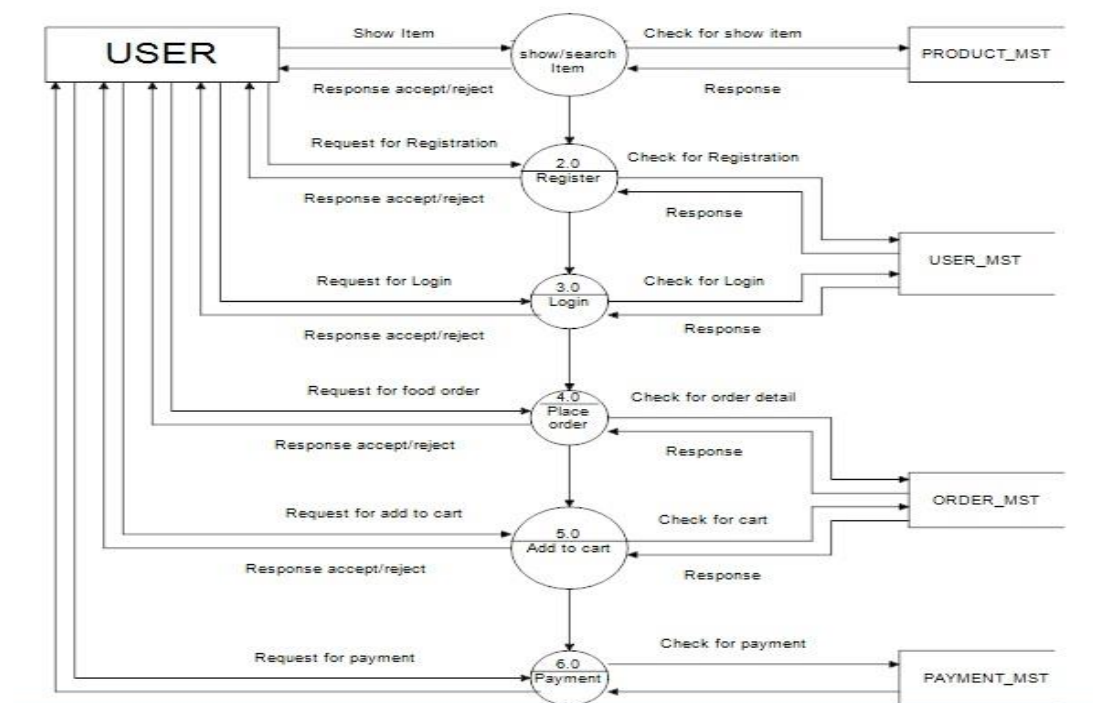


Fig 3.4.2 0 Level DFD (User)

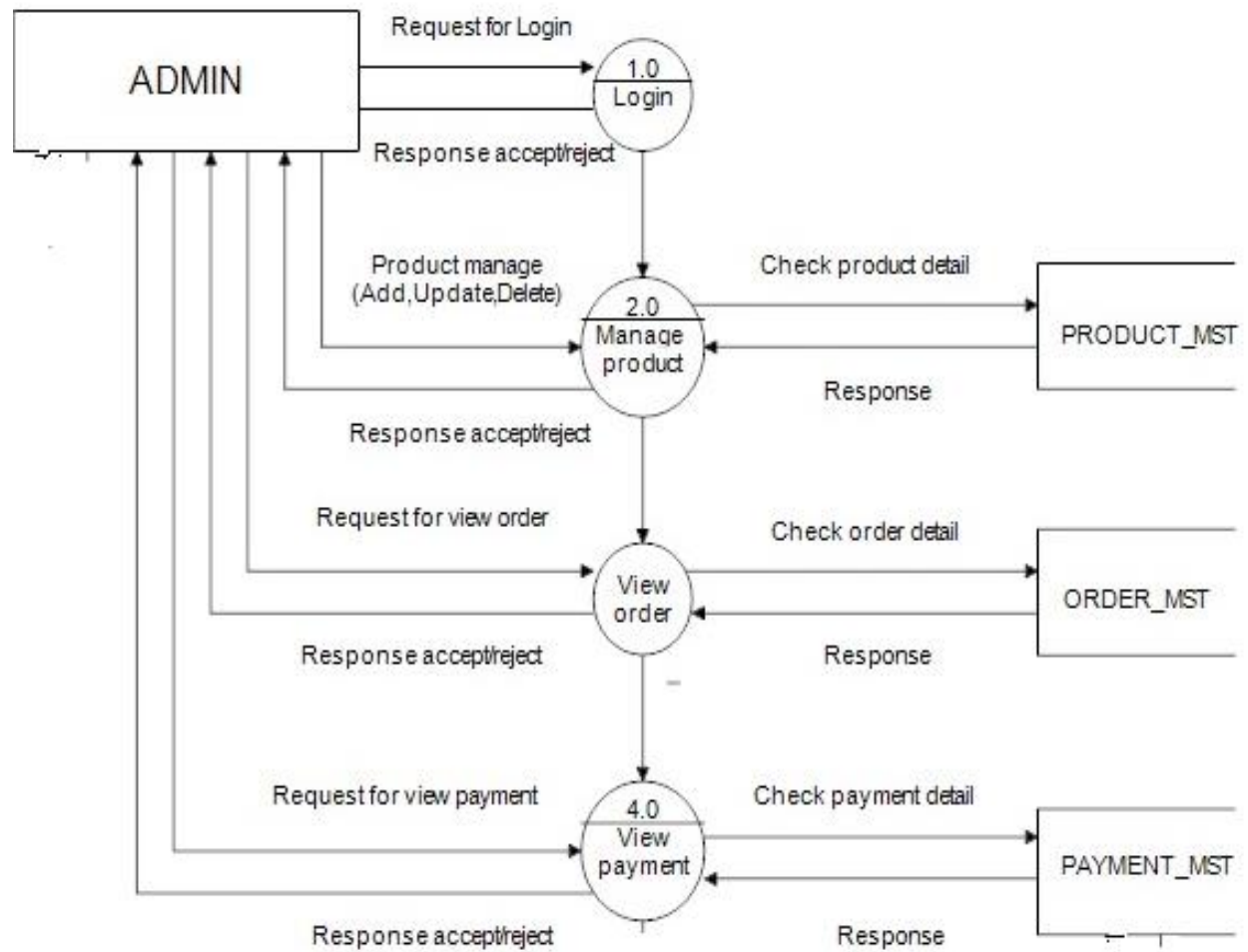


Fig 3.4.3 0 Level DFD (Admin)

1 level DFD n Level 1-n DFD (Identifies actual data flows and data stores):

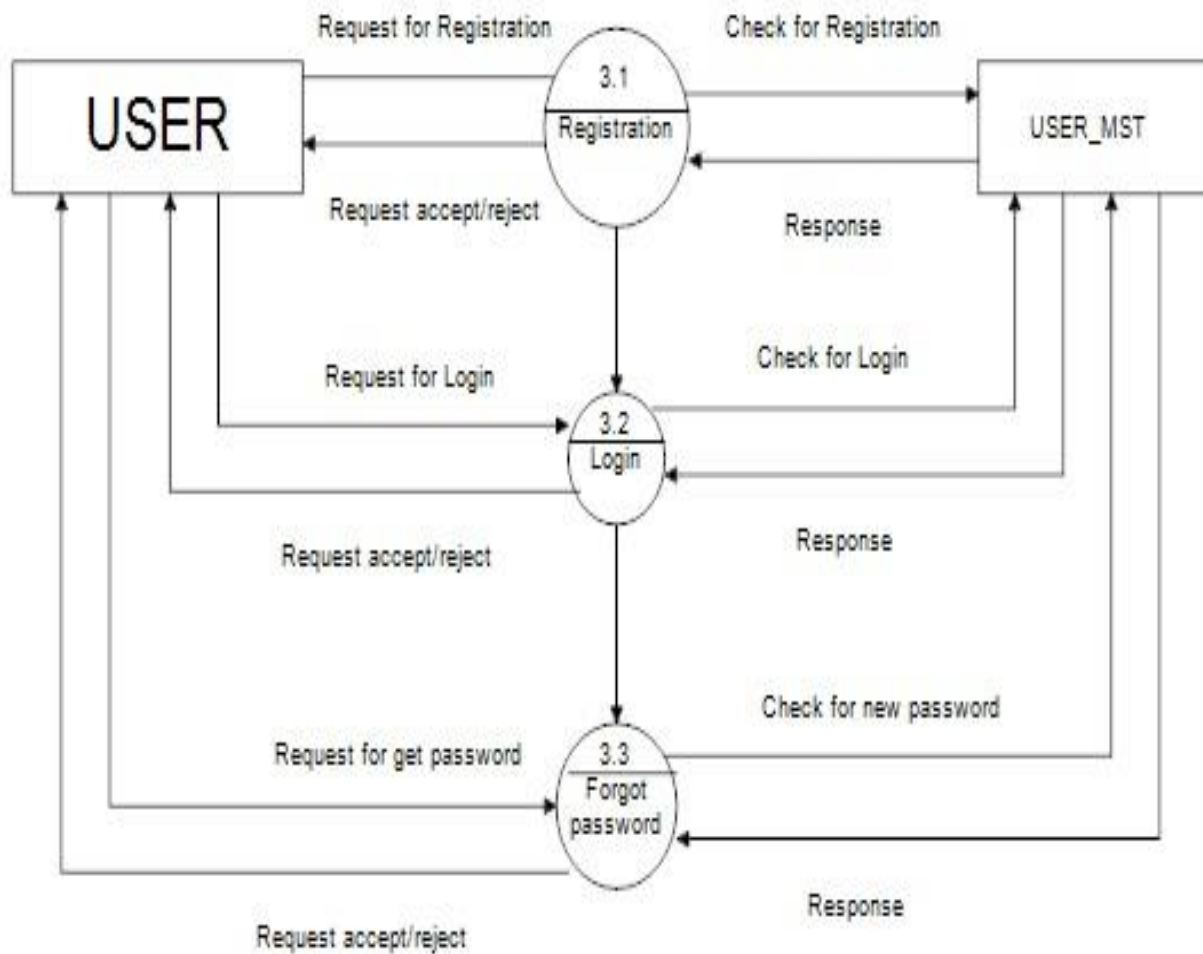


Fig 3.4.4 1 Level DFD (User)

3.6.4. FLOWCHART:

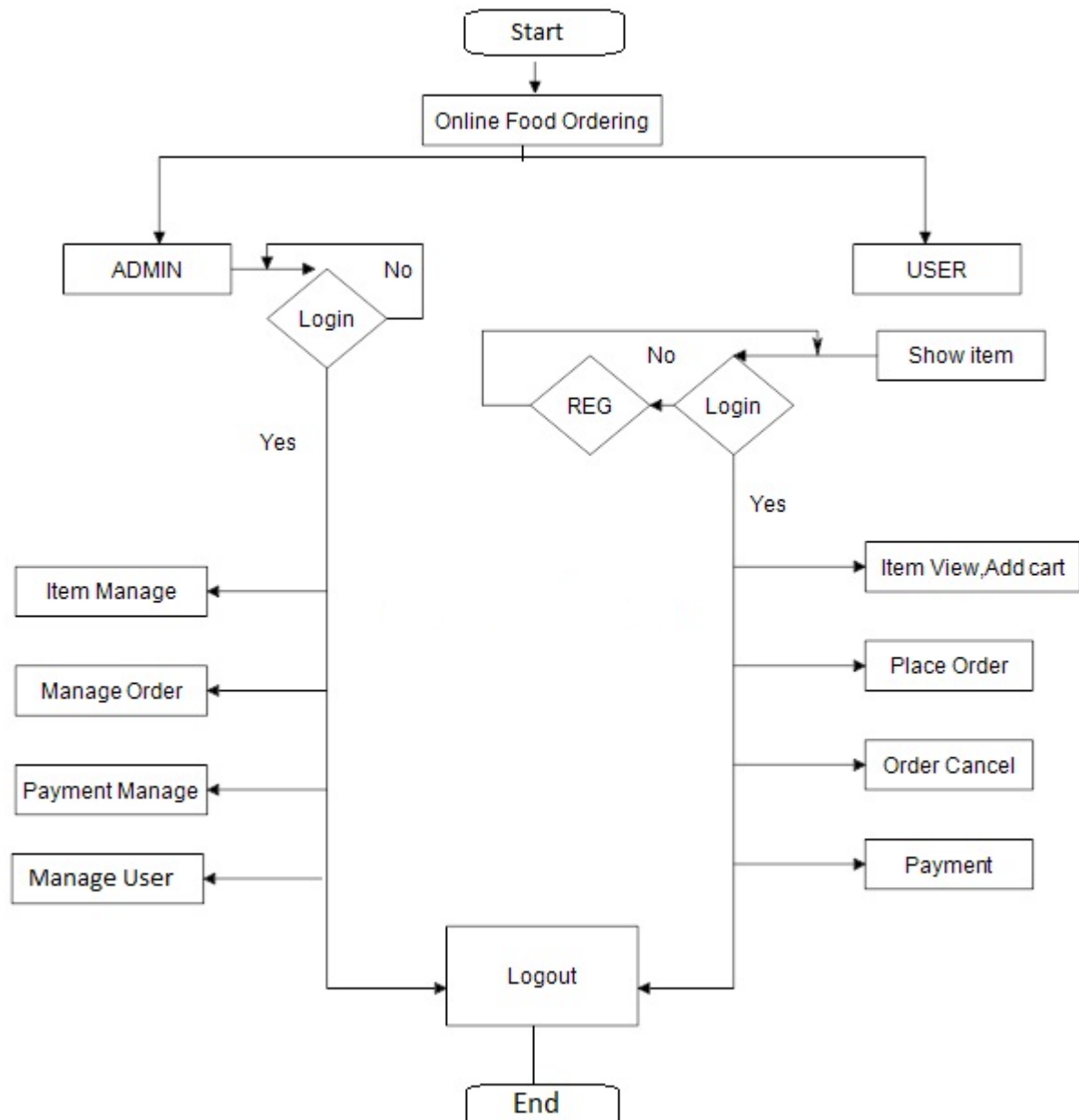


Fig 3.5 Flowchart

3.6.5. CLASS DIAGRAM:

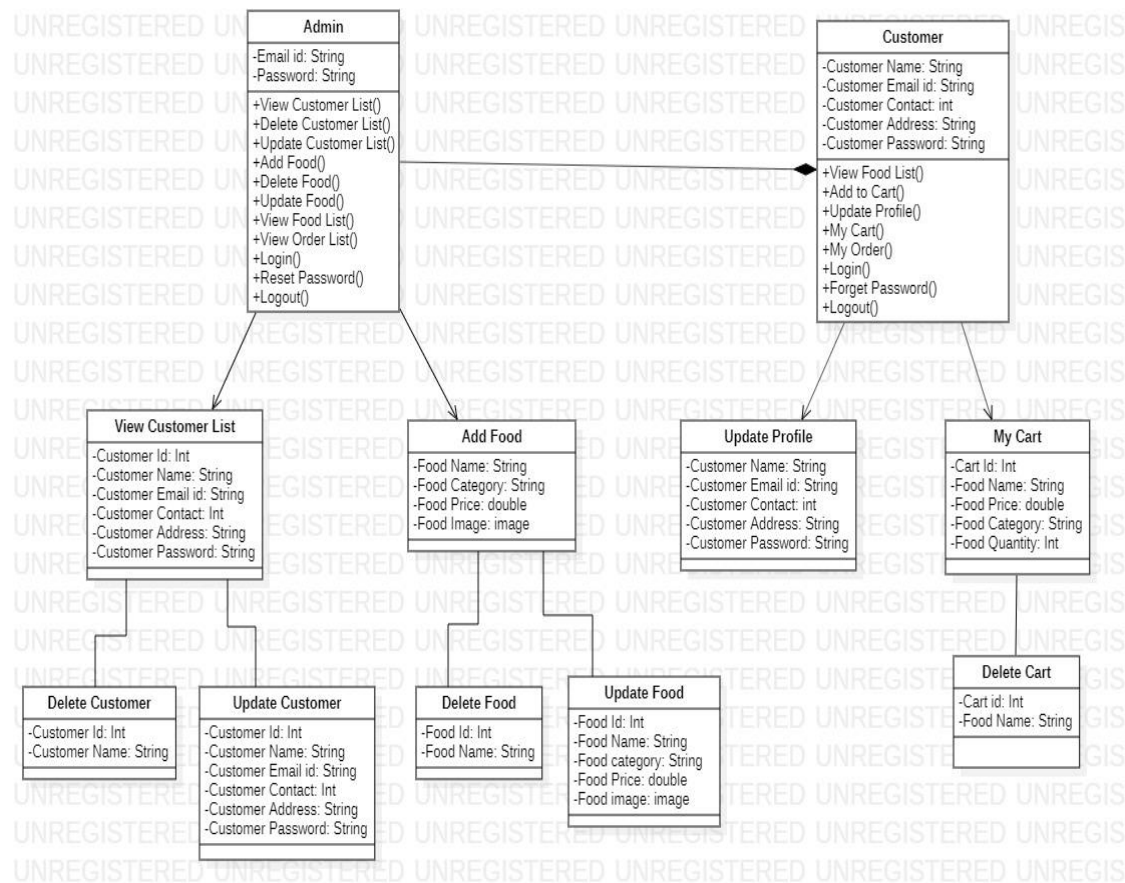


Fig 3.6 Class Diagram

3.7. Project SDLC Model:

Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model.

The Radius of the spiral at any point represents the expenses (cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

Below diagram shows the different phases of the Spiral Model:

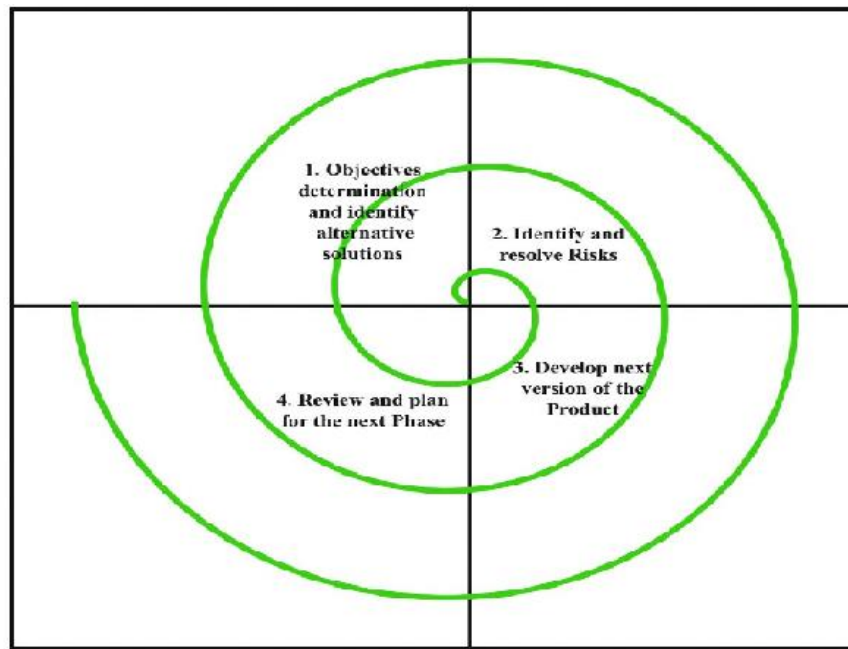


Fig 3.7. Spiral Model

Chapter 4:

System Design

4.1. BASIC MODULE IN PROJECT:

Login Module:

Admin can login with email-id and password. Customer can login with valid email-id and password which get after registration.

Sign Up Module:

Customer can register by adding information like customer name, customer email-id, customer contact, customer Address, customer Password.

Customer Home Page Module:

Customer can view the food list, customer can add to cart, update and delete the cart by own wish. Customer can update his/her profile by login. Customer can logout.

Admin Home Page Module:

Admin can view the food list and customer list by login and they can add the food items, update or delete. Admin can view customer list and they can update or delete customer. Admin can reset the password if the admin feels unsafe.

Manage Food:

Admin can view Food list and they can add new food to the Food Plaza food list by adding information like food name, food category, food price, food image. If the admin want to update the food list then admin can update the food information like Food-id, food-name, food category, food Price and food image. Admin can delete the food in the food list.

4.2. DATA DESIGN:

Database is critical for all business and educational system. A good database does not allow any form of anomalies and stores only relevant information in an ordered manner. If a database has anomalies, it is affecting the efficiency and data integrity. For example, delete anomaly arise upon the deletion of a row which also forces other useful data to be lost.

4.2.1. DATA DICTIONARY:

TABLES IN FOODPLAZA:

Tables_in_foodplaza_22072
admin_22072
cart_22072
customer_22072
food_22072
order1_22072

ADMIN TABLE:

adminEmail	adminPass
admin@gmail.com	admin123

CUSTOMER TABLE:

```
select * from customer_22072;
```

id	custName	custAddress	custContact	custEmail	custPass
1	neha	powai	9869051516	nehayadav@gmail.com	nom
2	soniya	ftu	8652595538	soniya@somaiya.edu	123

CART TABLE:

```
mysql> select * from cart_22072;
```

cartId	foodId	custEmail	foodQuantity
2	1	piyu@gmail.com	1
3	1	piyu@gmail.com	1
4	1	soni.pk@somaiya.edu	1
5	3	neha@gmail.com	1

4.2.2. DATA INTEGRITY AND CONSTRAINTS:

INTEGRITY AND CONSTRAINTS OF ADMIN TABLE:

```
mysql> desc admin_22072;
```

Field	Type	Null	Key	Default	Extra
adminEmail	varchar(100)	YES		NULL	
adminPass	varchar(15)	YES		NULL	

INTEGRITY AND CONSTRAINTS OF CUSTOMER TABLE:

```
mysql> desc customer_22072;
```

Field	Type	Null	Key	Default	Extra
custId	int(11)	NO	UNI	NULL	auto_increment
custName	varchar(400)	YES		NULL	
custAddress	varchar(500)	YES		NULL	
custContact	varchar(10)	YES	UNI	NULL	
custEmail	varchar(40)	NO	PRI	NULL	
custPassword	varchar(15)	YES		NULL	
conPassword	varchar(15)	YES		NULL	

INTEGRITY AND CONSTRAINTS OF FOOD TABLE:

```
mysql> desc food_22072;
```

Field	Type	Null	Key	Default	Extra
foodId	int(11)	NO	PRI	NULL	auto_increment
foodName	varchar(500)	YES		NULL	
foodCategeory	varchar(100)	YES		NULL	
foodPrice	double	YES		NULL	
FoodImage	blob	YES		NULL	

INTEGRITY AND CONSTRAINTS OF CART TABLE:

```
mysql> desc cart_22072;
```

Field	Type	Null	Key	Default	Extra
cartId	int(11)	NO	PRI	NULL	auto_increment
foodId	int(11)	YES		NULL	
custEmail	varchar(100)	YES		NULL	
foodQuantity	int(11)	YES		NULL	

INTEGRITY AND CONSTRAINTS OF ORDER:

```
mysql> desc order1_22072;
```

Field	Type	Null	Key	Default	Extra
orderId	int(11)	NO	PRI	NULL	auto_increment
custEmail	varchar(100)	YES		NULL	
totalPrice	double	YES		NULL	
orderStatus	varchar(100)	YES		processing	
orderDate	varchar(100)	YES		NULL	

4.3 USER INTERFACE:

HomePage:

Homepage Header:



Fig 4.1.1 Homepage Header

Homepage Footer:

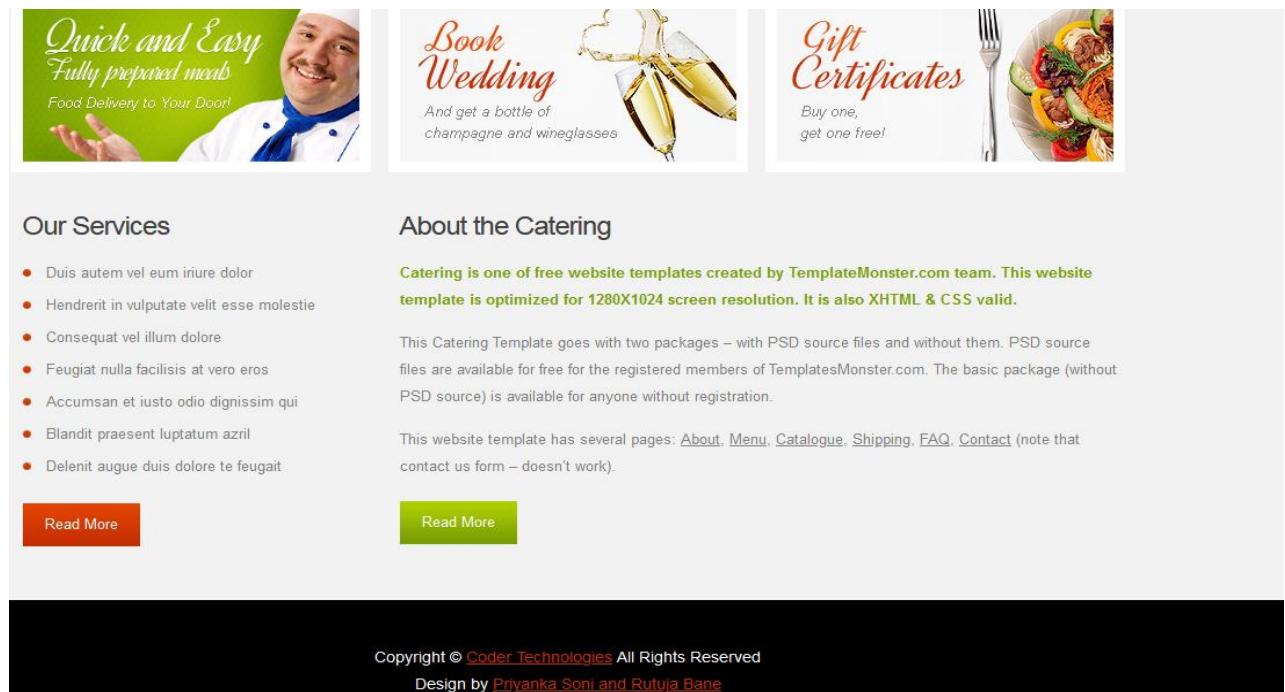


Fig 4.1.2 Homepage Footer

Login Page:

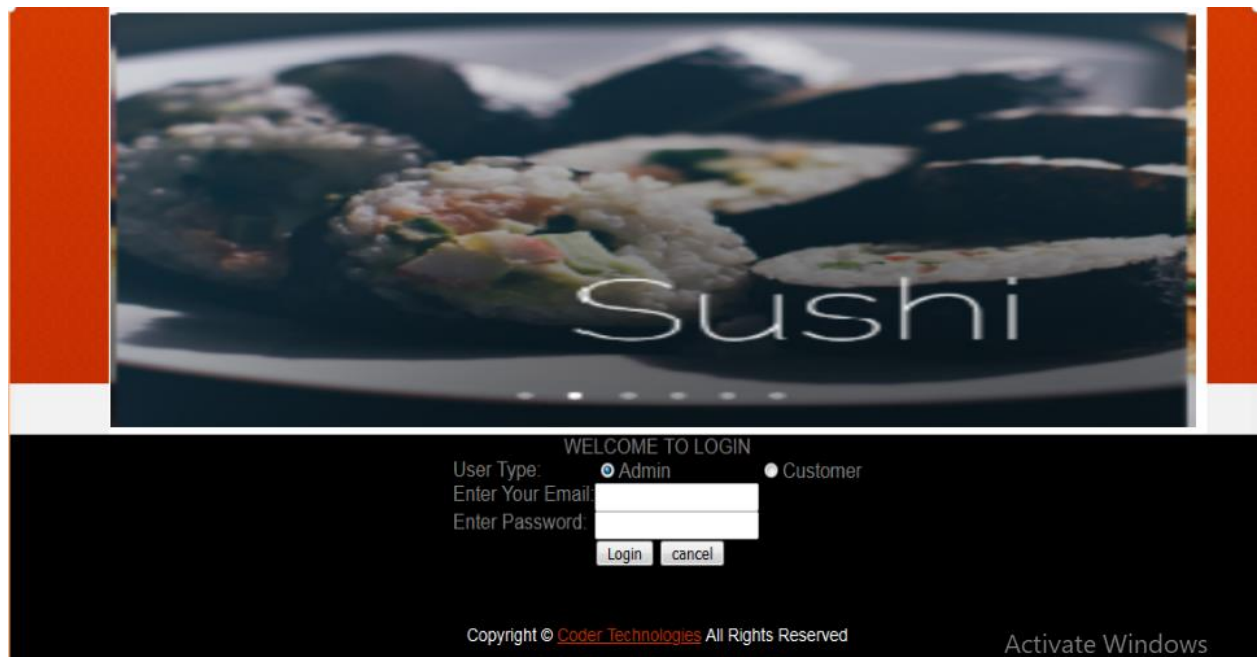


Fig 4.1.3 Login Page

Invalid Admin Login:



Fig 4.1.4 Invalid Admin Login

Invalid Customer Login:



Fig 4.1.5 Invalid Customer Login

Register:



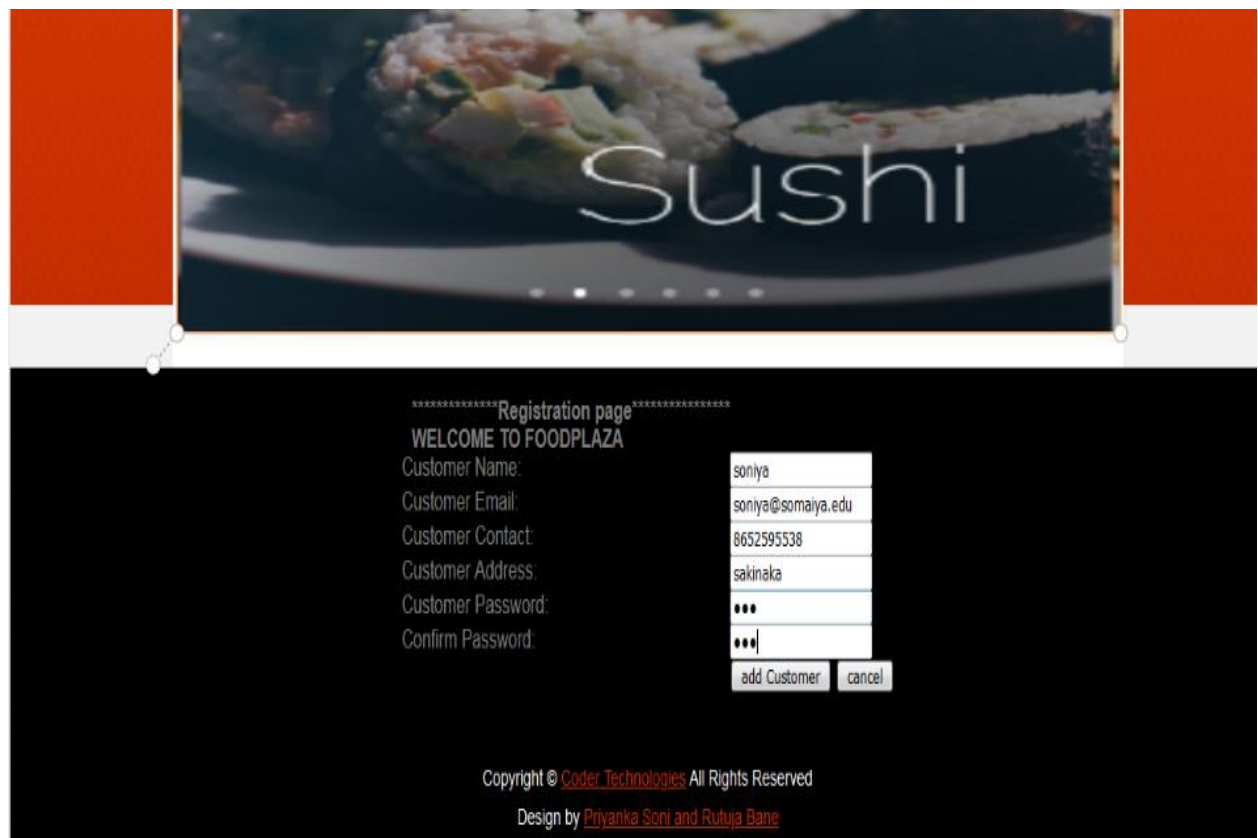


Fig 4.1.6 Register Page

Customer Home page after login:



Fig 4.1.7 Customer Home Page After Login

Show Food List Page:



Fig 4.1.8 Show Food List

Update Profile:

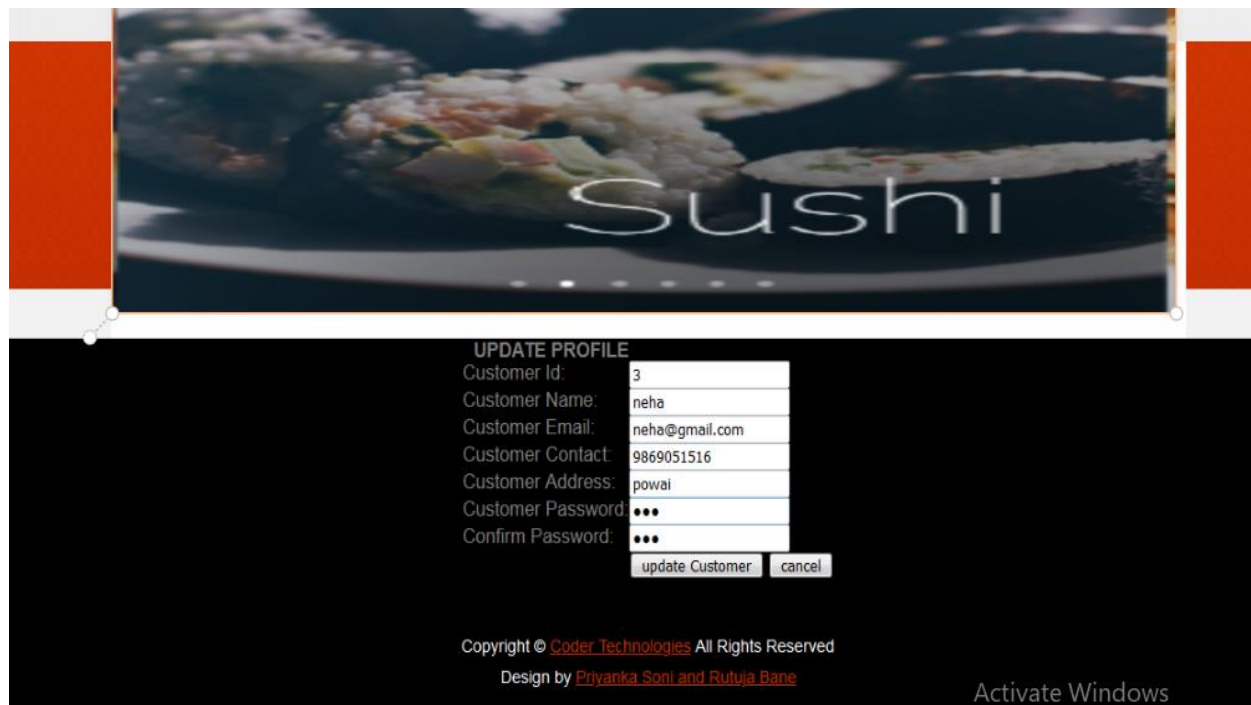


Fig 4.1.9 Update Profile

After updating profile:

Here the email neha@gmail.com is updated to nehayadav@gmail.com



Fig 4.1.10. After updating profile

My cart:

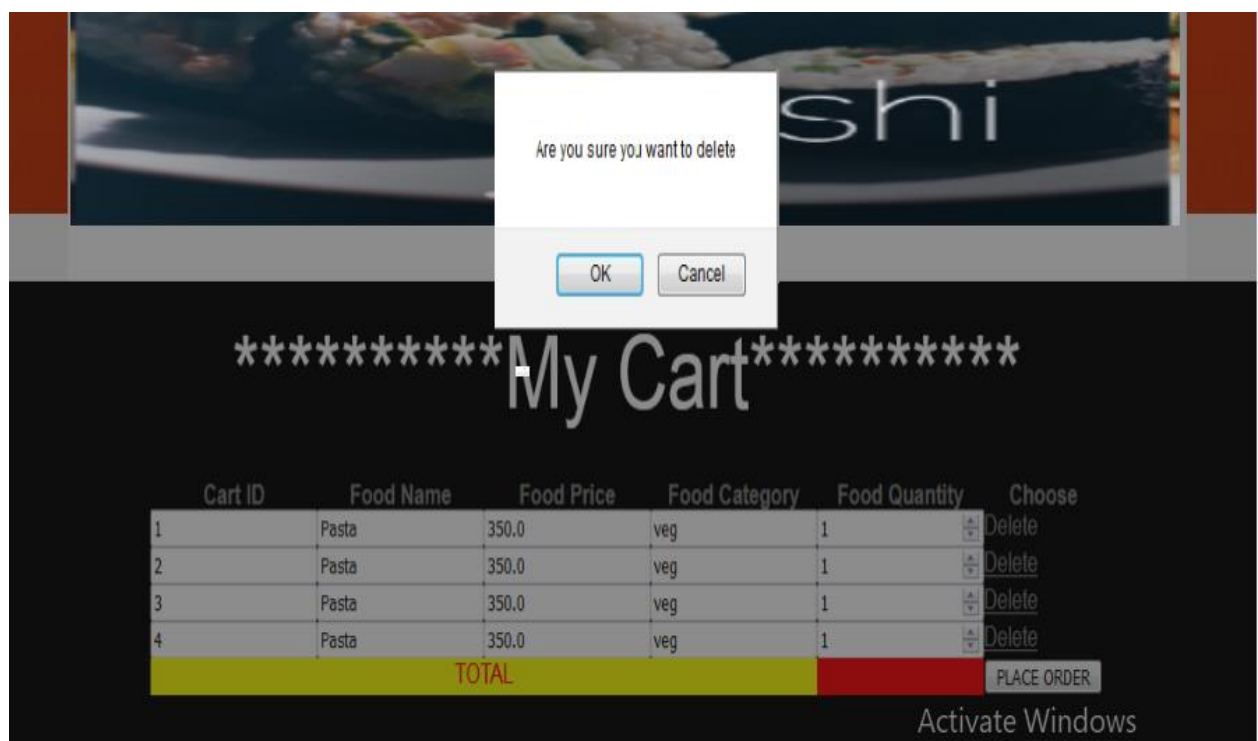
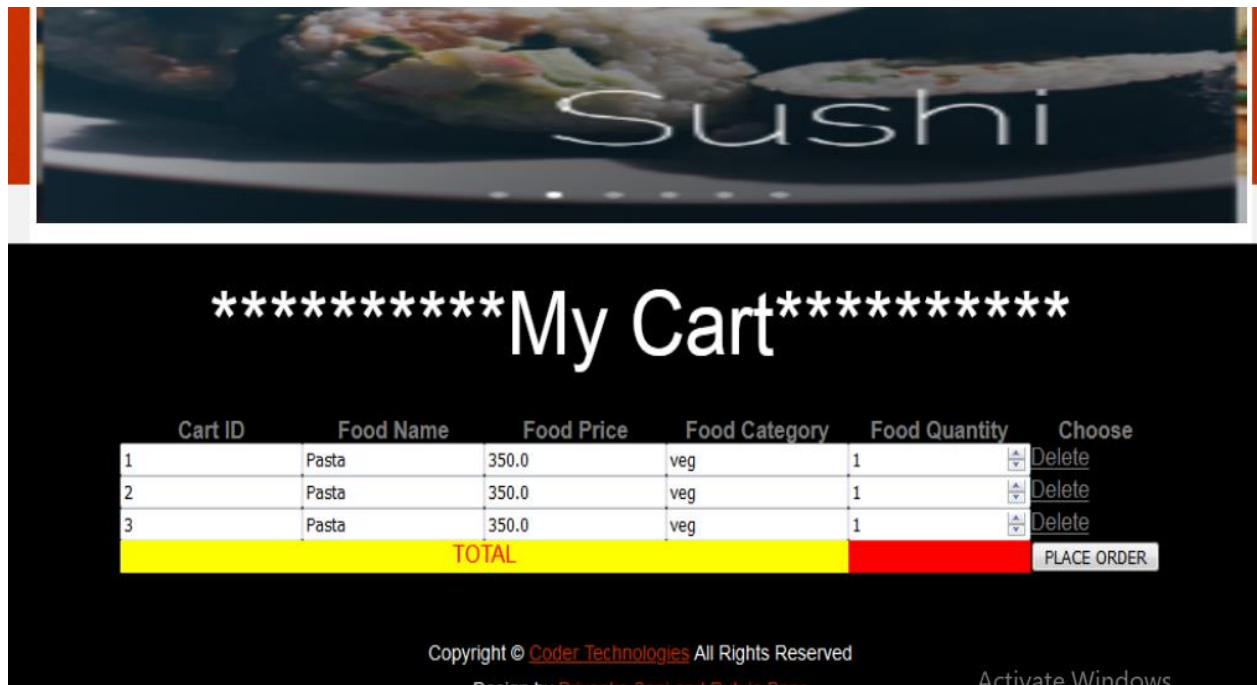


Fig 4.1.11. My Cart

After deleting pasta from cart:



*****My Cart*****

Cart ID	Food Name	Food Price	Food Category	Food Quantity	Choose
1	Pasta	350.0	veg	1	Delete
2	Pasta	350.0	veg	1	Delete
3	Pasta	350.0	veg	1	Delete
TOTAL					PLACE ORDER

Copyright © [Coder Technologies](#) All Rights Reserved
Design by [Rajendra Singh](#) and [Rajendra Singh](#)
Activate Windows

Fig 4.1.12. After deleting pasta from cart

Admin Home page:



admin

[coder'skitchen.com](#)

[HOME](#) [FOOD LIST](#) [ADD FOOD](#) [CUSTOMER LIST](#) [ORDER LIST](#) [RESET PASSWORD](#) [LOGOUT](#)

coder's kitchen

Sushi

Activate Windows

Fig 4.1.13. Admin Home Page

Add Food Page:



Fig 4.1.14. Add Food Page

Food Name, Food category, Food Price, Food Image of French fries id added

The screenshot shows the 'Add Food' form in the application. At the top, there is a banner image of sushi with the word 'Sushi' overlaid. Below the banner, there is a dark background with the text '50% discount on G-pay' on the left. The form itself is titled '****ADD FOOD****' and contains the following fields: 'Food Name:' with the value 'Frenchfries', 'Food Category:' with radio buttons for 'veg' (selected) and 'Non-Veg', 'Food Price:' with the value '50', and 'Food Image:' with a 'Browse...' button and the filename 'frenchfries.jpg'. Below the form are two buttons: 'addfood' and 'cancel'. At the bottom of the page, there is a copyright notice: 'Copyright © Coder Technologies All Rights Reserved', the design credit: 'Design by [Pryanka Soni and Rutuja Bane](#)', and a watermark: 'Activate Windows'.

After the French fries is added:

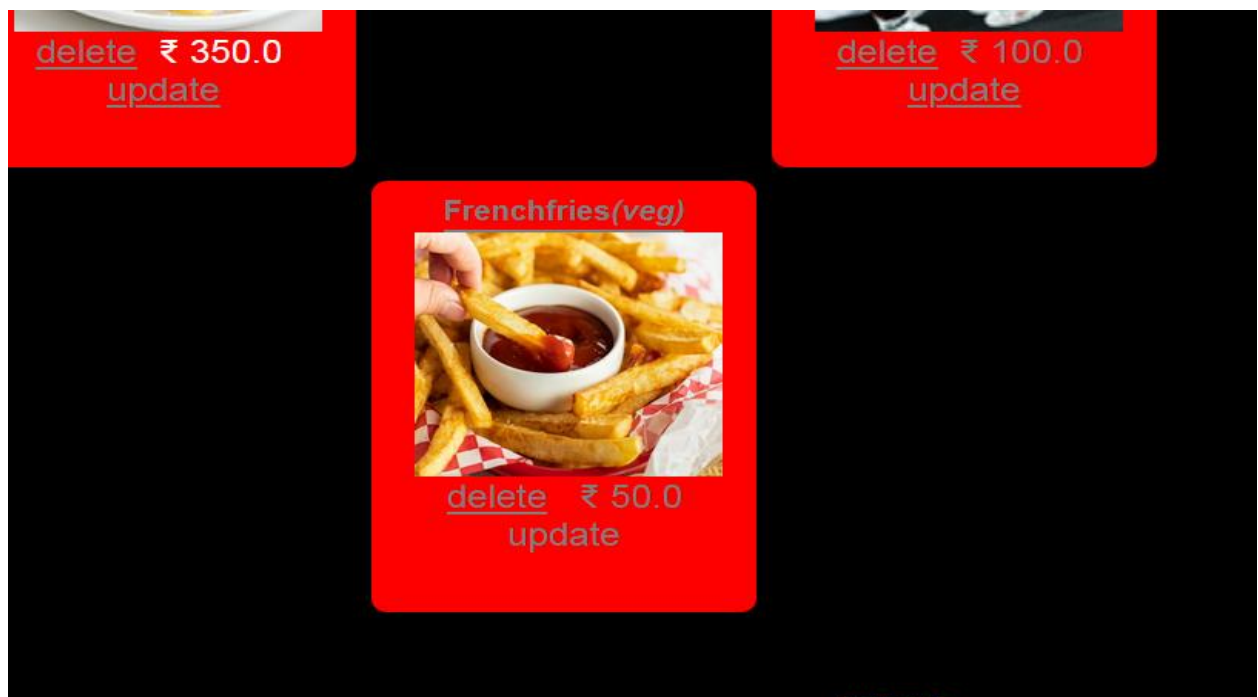


Fig 4.1.15. After the French fries is added

Updating French fries cost:

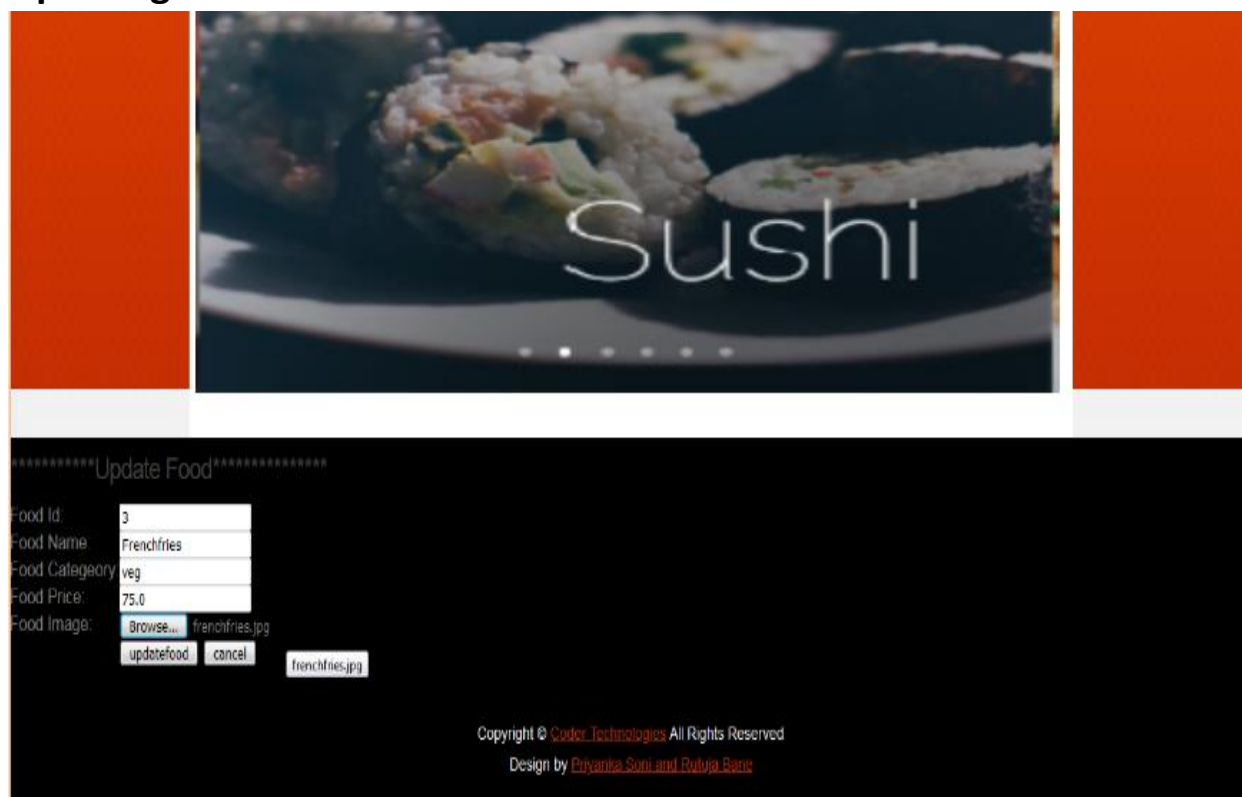


Fig 4.1.16. Updating French fries Cost

After updating French fries price: Admin and Customer can view the updated Information hence here the updated price of french-fries are updated in the food List.



Fig 4.1.17.After Updating French Fries Price

Deleting Food: After deleting the pasta and falooda the admin and customer can view the updated Food list.



Fig 4.1.18. Deleting Food

Customer list: Admin can view, update and delete the Customer.

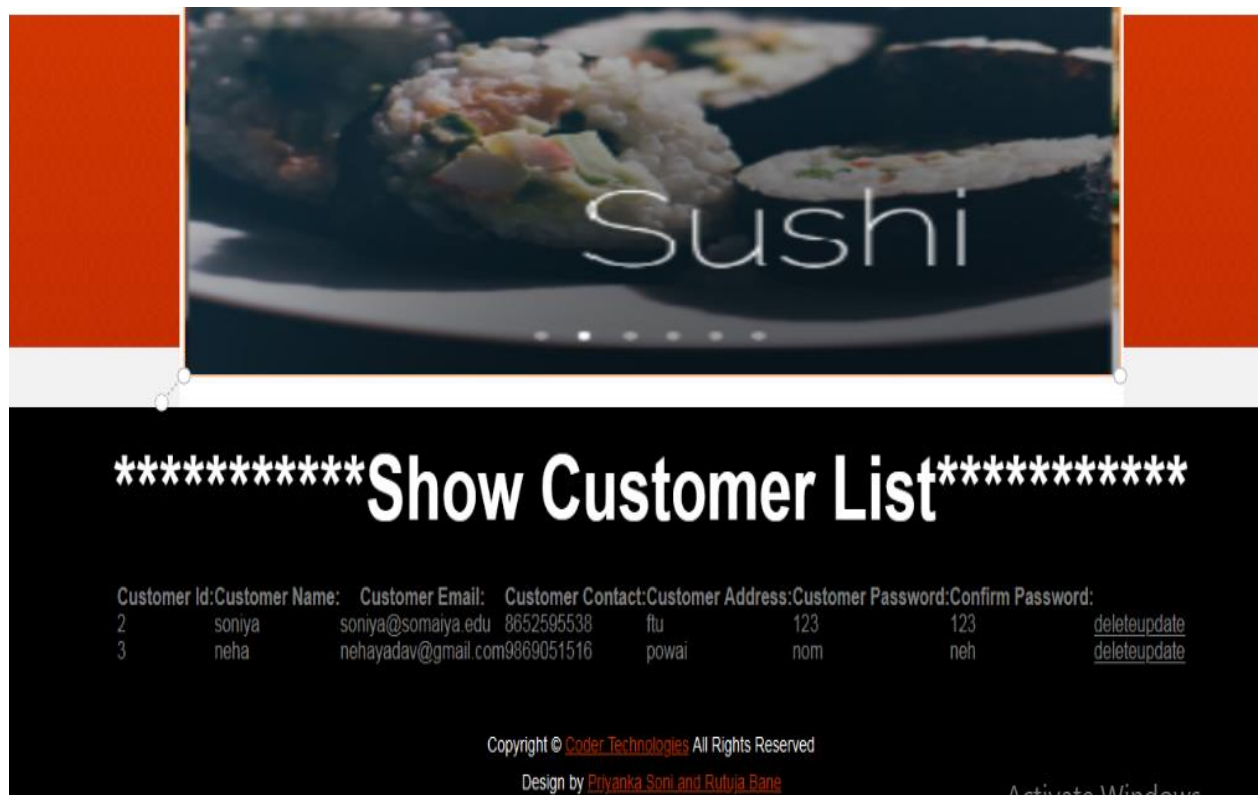


Fig 4.1.19. Customer List

4.4. SECURITY ISSUES:

1. Injection:

These attacks occur when the hacker sends untrusted data to a web application through a data submission channel, such as form input. The data input causes the web application to execute unintended commands.

For instance, a hacker can submit SQL database code into a data submission channel that is expecting a username in plain text. If the form input is insecure, the web application will execute the SQL code. Examples of injections include:

- LDAP queries.
- PHP queries.
- SQL queries.
- OS commands.

2. Broken Authentication:

This occurs when vulnerabilities in the web application's authentication system are exploited, giving attacker access to user or admin accounts. In some cases, the web application mismanages session-related information, enabling hackers to compromise the

user's identity. This information could be in the form of secret keys, passwords, session cookies, or others.

3. Sensitive Data Exposure:

If a web application doesn't provide adequate security for sensitive data, attackers could modify or sniff out the data and use it for their own purposes. For instance, applications that use weak Transport Layer Security (TLS) or weak encryption keys are susceptible to man-in-the-middle attacks.

The exposure of sensitive data can be mitigated by encrypting all sensitive data, disabling caching of responses with sensitive data, and using secure protocols and algorithms

4.5. TEST CASES DESIGN:

PHASES:

The Structure of the system can be divide into three logical components, plus the database, which is invisible to end user. Each of these component must be tested individually, and the approaches which will be used for each component described in the following sections.

DATABASE:

Testing of the database component is very straightforward, and has actually already been mostly completed. The database was the first component designed and before beginning work on any of the application, I wrote all of the SQL statements I expected to need and executed them directly, essentially isolating the database, using the SQL client. By doing this I was able to reveal and promptly fix a large percentage of error within the database itself.

WEB ORDERING SYSTEM:

Testing of the web ordering system will be the most strenuous, as it is component that will see the highest frequency of use and will be exposed to the most users, which leads to a higher potential of failure. Testing here will be divided into two phases. During normal use case testing I will execute all of the functions available through the user interface using a broad spectrum of reasonable values that a user would be expected to input. In addition to simply observing the in-application effects.

In phase two I will perform exceptional use case testing, where I will artificially generate that shouldn't arise, but possibly could, and monitor how the system handle these cases. These cases fall into one of the two categories-when the mistake happens in the browser and the server has to deal with it, or the other way around. I have tried to place appropriate checks on all values being sent back and forth so system realizes something is wrong before going to the database and potentially changing the state of the system, but it will very important to see if there is anything I have not accounted for.

Chapter 5:

Future Enhancements

5.1. IMPLEMENTATION APPROACHES:

System implementation is the important stage of project when these Theoretical design is turned into practical system.

The main stages in the implementation are as follows:

- #1 – Understand Why You Are **Implementing** an OMS. This is the most essential step to the process
- #2 – Planning
- #3 – Document Every **Stage** of the Process. ...
- #4 – Test, Test, Test
- #5 – Training

The implementation of the system application is done in Java, jQuery, HTML, JSP and the datasets are stored in MySQL database.

We have developed a web-based application. The hardware required for our application includes a desktop or laptop with browser and internet connection. For the initial implementation of the system we have added the common foods and beverages. The Food list with higher recommendation will showcase to the customer.

5.2. CODING DETAILS AND CODE EFFICIENCY:

➤ Libraries used:

```
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import com.foodplaza.pojo.Cart;
import com.foodplaza.pojo.Customer;
import com.foodplaza.pojo.Food;
import com.foodplaza.utility.DBConnection;
```

➤ SQL Connection:

To establish SQL Connection:

```
import java.sql.*;

public class DBConnection {

    static Connection conn=null;
    public static Connection getConnection() throws ClassNotFoundException , SQLException
    {
        Class.forName("com.mysql.jdbc.Driver");
        conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/foodplaza_22072","root","root");
        return conn;
    }
}
```

➤ User without Login Add to Cart:

If the user try to add the food in the Add to cart without login then then the user will not able to add the food to the cart and a Message "Please login first"

```
if(usertype==null)
{
    req.setAttribute("status","Please login first");
    RequestDispatcher rd=req.getRequestDispatcher("Login.jsp");
    rd.forward(req, res);
}
else
{
    String uname=(String)session.getAttribute("username");
    if(action!=null && action.equals("MyCart"))
    {
        int foodid=Integer.parseInt(req.getParameter("foodid"));
        System.out.println(foodid);
        ct=new Cart();
        ct.setFoodId(foodid);
        ct.setCustEmail(uname);
        ct.setFoodQuantity("1");

        ans=cartdi.addToCart(ct);
        if(ans)
        {
            System.out.println("added");
            List<Cart> cartlist=cartdi.showCart(uname);
            session.setAttribute("cartlist",cartlist);
            System.out.println(cartlist);
            res.sendRedirect("Cart.jsp");
        }
        else
        {
            System.out.println("not added");
        }
    }
    else
    {
        System.out.println("added");
        List<Cart> cartlist=cartdi.showCart(uname);
```

User Type:

When the user want to login the user will check the radio button admin or Customer. If the admin email and password is valid then the admin will redirect to the Index.jsp page or else error message "invalid Admin!! Please enter valid username and password". If the Customer email and password is valid then the admin will redirect to the login.jsp page or else error message "invalid Customer!! Please enter valid username and password".

```
String Ustertype=req.getParameter("user");
String Emailid=req.getParameter("Email");
String Password=req.getParameter("pass");
System.out.println(Ustertype);
if(Ustertype!=null && Ustertype.equals("admin"));
{
    ans=ldi.isAdmin(Emailid,Password);
    System.out.println(ans);
    if(ans)
    {
        session.setAttribute("ustertype","admin");
        RequestDispatcher rs=req.getRequestDispatcher("Index.jsp");
        rs.forward(req,res);
    }
    else
    {
        out.print("Invalid Admin!! Please Enter Valid Username And Password");
        RequestDispatcher rs=req.getRequestDispatcher("/Login.jsp");
        rs.include(req,res); } }
if(Ustertype!=null && Ustertype.equals("Customer"));
{
    ans=ldi.isCustomer(Emailid,Password);
    if(ans)
    {
        session.setAttribute("username", Emailid);
        //out.print("Welcome Customer");
        session.setAttribute("ustertype","customer");
        RequestDispatcher rs=req.getRequestDispatcher("Index.jsp");
        rs.forward(req,res); }
    else{ out.print("Invalid Customer!! Please Enter Valid Username And Password");
        RequestDispatcher rs=req.getRequestDispatcher("/Login.jsp");
        rs.include(req,res); } }
```

➤ Add and Update the Customer:

```
if(action!=null && action.equals("addcustomer"))
{
    String name=req.getParameter("customerName");
    String CustEmail=req.getParameter("custEmail");
    String CustomerContact=req.getParameter("CustomerContact");
    String customerAddress=req.getParameter("customerAddress");
    String pass=req.getParameter("pass");
    String pass1=req.getParameter("pass1");
    c.setCustName(name);
    c.setCustEmail(CustEmail);
    c.setCustContact(CustomerContact);
    c.setCustAddress(customerAddress);
    c.setCustPassword(pass);
    c.setConPassword(pass1);
    ans=cdi.addCustomer(c);
    if(ans)
    { out.print("Customer Added"); }
    else { out.print("Customer Not Added"); }
} if(action!=null && action.equals("updatecustomer"))
{ System.out.println("inside update");
    int id=Integer.parseInt(req.getParameter("customerid"));
    String name=req.getParameter("customerName");
    String CustEmail=req.getParameter("custEmail");
    String CustomerContact=req.getParameter("CustomerContact");
    String customerAddress=req.getParameter("customerAddress");
    String pass=req.getParameter("pass");
    String pass1=req.getParameter("pass1");
    c.setCustName(name);
    c.setCustEmail(CustEmail);
    c.setCustContact(CustomerContact);
    c.setCustAddress(customerAddress);
    c.setCustPassword(pass);
    c.setConPassword(pass1);
    System.out.println(c);
    ans=cdi.updateCustomer(c);
```

Add Food:

Request Dispatcher is an interface whose implementation defines an object which can **dispatch** the **request** to any resources on the server. If the admin have an action of add food then food will be added. If something goes wrong then an error message " Food Not Added" will display to Admin

```
protected void doPost(HttpServletRequest req, HttpServletResponse res) throws
    ServletException, IOException
{
    InputStream foodimage;
    Part part;
    out=res.getWriter();
    action=req.getParameter("action");
    System.out.println("action:"+action);
    if(action!=null && action.equals("addfood"))
    {
        String name=req.getParameter("foodname");
        String categeory=req.getParameter("categeory");
        double price=Double.parseDouble(req.getParameter("foodprice"));
        part= req.getPart("foodimage");
        foodimage = part.getInputStream();
        f.setFoodName(name);
        f.setFoodCategeory(categeory);
        f.setFoodPrice(price);
        f.setFoodImage(foodimage);
        ans=fdi.addFood(f);
        if(ans) { out.print("Food Added");
            List<Food> flist=fdi.showFoodList();
            session=req.getSession();
            session.setAttribute("Foodlist", flist);
            RequestDispatcher rs=req.getRequestDispatcher("ShowFoodList.jsp");
            rs.forward(req, res); }
        {out.print("Food Not Added"); }
    }
}
```

Update and Delete Food:

If the admin have an action of delete food then food will be deleted. If something goes wrong then an error message "Food Not Deleted" will display to Admin. If the admin have an action of update food then admin have to write the food id no then current page will redirect to updatefood.jsp. Else the Admin will redirect to the foodlist.jsp page

```
if(action!=null&&action.equals("delete"))
{ int foodId=Integer.parseInt(req.getParameter("foodid"));
  ans=fdi.deleteFood(foodId);
  if(ans)
  { List<Food> flist=fdi.showFoodList();
    session.setAttribute("Foodlist", flist);
    res.sendRedirect("ShowFoodList.jsp"); }
  else { out.print("Food Not Deleted "); }
} // delete action
if(action!=null && action.equals("update"))
{ int foodId=Integer.parseInt(req.getParameter("foodid"));
  f=fdi.searchFoodById(foodId);
  session.setAttribute("Food", f);
  RequestDispatcher rs=req.getRequestDispatcher("/updatefood.jsp");
  rs.forward(req, res); }
else {
  List<Food> flist=fdi.showFoodList();
  session=req.getSession();
  session.setAttribute("Foodlist", flist);
  RequestDispatcher rs=req.getRequestDispatcher("ShowFoodList.jsp");
  rs.forward(req, res);
} //searchfood by id close
```

5.2.1. Code Efficiency:

Code efficiency plays a significant role in applications in a high-execution-speed environment where performance and scalability are paramount. One of the recommended best practices in coding is to ensure good code efficiency. Well-developed programming codes should be able to handle complex algorithms.

Recommendations for code efficiency include:

- To remove unnecessary code or code that goes to redundant processing
- To make use of optimal memory and non-volatile storage
- To ensure the best speed or run time for completing the algorithm
- To make use of reusable components wherever possible
- To make use of error and exception handling at all layers of software, such as the user interface, logic and data flow
- To create programming code that ensures data integrity and consistency
- To develop programming code that's compliant with the design logic and flow
- To make use of coding practices applicable to the related software
- To optimize the use of data access and data management practices
- To use the best keywords, data types and variables, and other available programming concepts to implement the related algorithm

5.3. TESTING APPROACH:

System testing is to check each step of the program to make sure that the designed software is working properly. The system was tested by inserting the admin username and password, then registering some people where the system excellently saved their information in the database, and it was saved successfully.

5.3.1. Unit Testing:

Unit testing focuses verification effort on the smallest part of the module. The unit testing implemented in every module of Online Food plaza. By giving correct manual input to the system, the data are stored in database and retrieved. If you want required module to access input or get the output to the Customer. Any error will accrued the time will provide handler to show what type of error will occurred

Test Cases	Explanation	Result
LOGIN	To check if only admin and register customer will be able to login after filling in the correct details in the available fields. Check if it was successful	Pass
PREVIEW	To check if the admin and customer will be able to visit their respective profile	Pass
ADD FOOD	To check if the admin can add food. Check if it was successful	Pass
CHECK EVENT	To check if the admin can view customer list, food list, delete or update customer and food list, customer can add cart, update or delete cart, view food list Check if it was successful.	Pass

Fig 5.1 Unit testing

5.3.2 Integrated Testing:

Integration testing is the next phase in system testing. The links interactions between MySQL or one activity and another were tested and everything was working well.

Test Case ID	Test Case Objectives	Test Case Description	Input	Expected Output	Results
	To check the interface link between the login page and the home page.	Click on the LOGIN button.	Button clicked.	To be directed to the home page.	Pass
			Button not clicked.	Remains unchanged.	Pass
	To check the interface link between the home page and other menu page	Click on the Menu button.	Button clicked.	To be directed to the respective activity page	Pass
			Button not clicked	Remains unchanged.	Pass

Fig 5.2 Integrated testing

5.3.3. Beta Testing:

Beta Testing is one of the Acceptance Testing types, which adds value to the product as the end-user (intended real user) validates the product for functionality, usability, reliability, and compatibility. Inputs provided by the end-users helps in enhancing the quality of the product further and leads to its success. This also helps in decision making to invest further in the future products or the same product for improvisation. Since Beta Testing happens at the end user's side, it cannot be controlled activity.

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

5.4. MODIFICATIONS AND IMPROVEMENTS:

The following section describes the work that will be implemented with future releases of the software.

- **Customize orders:** Allow customers to customize food orders
- Enhance User Interface by adding more user interactive features. Provide Deals and promotional

Offer details to home page. Provide Recipes of the Week/Day to Home Page

- **Payment Options:** Add different payment options such as PayPal, Cash, Gift, Cards etc. Allow to save payment details for future use.

- Allow to process an order as a Guest
- **Delivery Options:** Add delivery option
- **Order Process Estimate:** Provide customer a visual graphical order status bar
- **Order Status:** Show only Active orders to Restaurant Employees.
- **Order Ready notification:** Send an Order Ready notification to the customer
- **Restaurant Locator:** Allow to find and choose a nearby restaurant
- Integrate with in store touch screen devices like iPad

5.5. TEST CASES:

Web page content should be correct without any spelling or grammatical errors

- All fonts should be same as per the requirements.
- All the text should be properly aligned.
- All the error messages should be correct without any spelling or grammatical errors and the error message should match with the field label.
- All the fields should be properly aligned.
- Enough space should be provided between field labels, columns, rows, and error messages.
- All the buttons should be in a standard format and size.
- Home link should be there on every single page.
- Check for broken links and images.
- Confirmation message should be displayed for any kind of update operation.
- Check the end user can run the system without frustration.
- Check the tab should work properly.
- Scroll bar should appear if required.
- Title should display on each web

Chapter 6:

Results and Discussion

6.1. Test Reports:

Test Case ID	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when valid email and password is entered	Email: soni.pk@somaiya.edu Password: <u>soni.pk@7</u>	Login should be successful	Login was successful	Pass
2	Check response when invalid email or password is entered	Email: soni.pk@somaiya.edu Password: soni123	Login should not be successful	Login was unsuccessful	Pass
3	Check Every Functionality	Perform Various Operations	All operation should be perform properly.	All operation should be perform properly	Pass

Fig 6.1 Test Reports

6.2. User Documentation:

User documentation refers to the documentation for a product or service provided to the end users. The user documentation is designed to assist end users to use the product or service. This is often referred to as user assistance. The user documentation is a part of the overall product delivered to the customer.

Admin can login can go to the admin panel. Admin can access Menu's Home, Add Food, Food List, Customer List, Order List, Reset password and Logout. Admin can add the food by adding food name, food category, food price and food image. Admin can update the food items by viewing food id and food name by then update the food information like food id, food name, food category, food price and food image and they can also delete the food by using food id and food name. Admin can view the food list and customer list. They can also reset the password.

Customer should register/login to go to the Customer panel. Customer should login with the registered email-id and password. Customer can access Menu's Home, Food List,

Update Profile, My Cart, Forget Password and Logout. Customer first can see the food list then they can add to cart the food items, user can register by using customer name, customer email, customer contact, customer Address and customer password. Customer can update his/her profile by itself. Customer can add to cart, delete the cart and update the cart. If the customer forget his/her password then they can click on forget password to get new password by the server.

Chapter 7: Conclusions

7.1. Conclusion:

The main objective of the application is to help IT students understand the basics of Java,

JavaScript and HTML. The following results have been achieved after completing the system and relate back to the system's objective.

- Should allow IT students to browse through the code and application:

This can be achieved when students are able to run and install the application. When they run the application, they can browse through the implementation of different objects.

- Should allow users to browse through different product categories: This is achieved through an easy to use graphical interface menu options.

- Should allow users to save items to the cart and view detailed information about the order:

The users can add any number of items to the cart from any of the available food categories by simply clicking the Add to Cart button for each item. Once item is added to the cart, user is presented with detailed order to review or continue shopping.

- Should allow the user to Check Out the item (s): This is achieved using the "Proceed to checkout button" in the cart initially and then "Check Out" button at last step after "review Order" step.

Button is disabled when there are no items in the cart.

- Should allow the user to process the payment: This is achieved when user selects "Processed to

Checkout" button and fill up the Payment information details.

- Should allow the user to see Success message after placing an order: This is achieved when user successfully places an order. The user is given the order conformation number along with success message.

7.1.1. Significance of the System:

ONLINE FOOD ORDER SYSTEM is a website designed primarily for use in the food delivery industry. This system will allow hotels and restaurants to increase scope of business by reducing the labor cost involved. The system also allows to quickly and easily manage an online menu which customers can browse and use to place orders with just few clicks. Restaurant employees then use these orders through an easy to navigate graphical interface for efficient processing.

7.2. Limitations of the System:

Limitation of project on online ordering system:

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by me. Though the software presents a

broad range of options to its user some intricate option could not be covered into it; partly because of logistic and partly due to lack of sophistication. Paucity of time was also measure constraints, does it was not possible to make software full proof and dynamic. Lack of time also compelled me to ignore some part such as storing old result of the candidate etc. Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it acknowledge that a layman May find it a bit problematic at first instance .The user is provided help at its step for his convenience in working with the software.

List of limitations which is available in the online food ordering system:

- Excel export has not been developed for Food item, Category due to some criticality.
- The transactions are executed in off-line mode, hence on-line data for Cart, Order capture and modification is not possible.
- Off-line reports of Food item, Customer, Cart cannot be generated due to batch mode execution.

7.3. Future Scope of the Project:

- Enhance user interface by adding more user interactive features. Provide deals and Promotional Offer details to home page. Provide recipes of the week/Day to home page.
- Payment Options: Add different payment options such as PayPal, Cash, Gift Cards etc. Allow to save payment details for future use.
- This order food online system project aimed at developing an online food ordering system which can be used in small places, and medium cities firstly and then on a large scale.
- The system enables staff to let update and make changes to their food and beverage list information based on the orders placed and the orders completed.

Chapter 8:

References

References:

<https://www.slideshare.net/armanreza161/food-ordering-system-75284735>

<https://dev.mysql.com/downloads/connector/j/>

<https://www.youtube.com/watch?v=7TOmdDJc14s&list=PLsyebzWxl7pUPF2xijJiG4BKC9xGY46>

<https://www.javatpoint.com/servlet-tutorial>

<https://www.udemy.com/course/complete-java-course-beginner-to-advanced-k/>

Glossary:

Food Plaza: Food Plaza is our Online Food Ordering System name.

OMS: OMS is abbreviation of Online Management System.