Technical Report

Food Delivery website



Submitted By

**Muhammad Tayyab**

2019-ag-6483

Advised By

**Hafiz M. Haroon**

**Bachelor of Science**

**In**

**Computer Science**

**-----------------------------------------------------------------**

## SUB-CAMPUS BUREWALA

# UNIVERSITY OF AGRICULTURE

# FAISALABAD, PAKISTAN

# 2023

# TITLE: FOOD DELIVERY WEBSITE

# SIGNATURES:

# Student \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# (Muhammad Tayyab)

# EVALUATION COMMITTEE:

# Advisor \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# (Hafiz M. Haroon)

# Member \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# (Sidra Habib)

# Member \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# (Dr. Rana M. Saleem)

**DEDICATION**

I dedicate this effort to the Almighty Allah, thank you for the guidance, strength, Power of Mind, Protection and skills and for giving me healthy Life and the Last Prophet of Allah Hazrat Muhammad (PBUH)

**TO MY DIVINELY FAMILY**

I dedicate my work to my loved parents and Brothers who have always teach me to trust in Allah, believe in hard work and teach me that so much could be done with little.

**TO MY RESPECTED TEACHERS**

Teachers are always great source of inspiration and motivation to me. However, my teacher’s remained beacon of light for me. Their sincere guidance and prudent leadership guided my way clearly not only to excel in achieving this dissertation but also definite directions for professional career too.

**TO MY SWEET FRIENDS**

My friends mean world to me. I am proud to say that save their cooperation, collaboration and team work, I would not be able to achieve this target easily. I’m more than thankful to all for their Support and encouragement during the time, I need them most.

**DECLARATION**

This technical report submitted in partial fulfillment of the requirements for my Final Year Project of BS (Computer Science) I declare that this work has been completed according to the guidelines established by the Advisor and has not been submitted for any other purpose.

Muhammad Tayyab

(2019-ag-6483)

**ACKNOWLEDGEMENT**

All praises be to **Allah, the Lord of the world,** the Master of the Day After, who has given unlimited mercy to His creations which show Allah’s love is spread around. He is the one who has love, and teaches mankind of the power of His love and from Allah’s love, He sent the Messenger, Muhammad (peace be upon him) to guide mankind from wickedness to the truth of slam. Thus, peace and salutation are uttered to beloved Prophet Muhammad SAWW (peace upon him) who has brought mankind from cruelty and stupidity ages to peaceful and safety ages, Islam period.

I would like to acknowledge and give my warmest thanks to my supervisor **Hafiz Muhammad Haroon** who made this work possible. His guidance and advice carried me through all the stages of writing my Report. I would also like to thank my friends for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would also like to give special thanks to my elder Brothers **Faisal Jamil** and **Muhammad Shakeel** and my family as a whole for their continuous support and understanding when undertaking my report and my project. Your prayer for me was what sustained me this far.

# ABSTRACT

An Online Food Ordering System is proposed here which simplifies the food ordering process. The proposed system shows a user interface and updates the menu with all available options so that it eases the customer work. Customer can choose more than one item to make an order and can view order details before logging off. The order confirmation is sent to the customer. The order is placed in the queue and updated in the database and returned in real time. This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

**TABLE CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Content** | **Page No.** |
| Ch# 1 | **Introduction** |  |
| 1 | History | 1 |
| 2 | Existing System | 3 |
| 3 | Disadvantages of existing system | 3 |
| 4 | Proposed system | 4 |
| 5 | Goal and objectives | 5 |
| 6 | Problem Statement | 5 |
| 7 | Solution of Problems | 6 |
| 8 | Functional Requirements | 7 |
| 9 | Non-Functional Requirements | 8 |
| 10 | Tools and Technologies | 8 |
| 11 | Gantt Chart | 8 |
| Ch# 2 | **Methodology** |  |
| 1 | Use case Diagram | 9 |
| 2 | ER Diagram | 10 |
| 3 | Data Flow Diagram | 11 |
| 3.1 | Level Zero Data Flow Diagram | 11 |
| 3.2 | Level One Data Flow Diagram | 12 |
| 3.3 | Level Two Data Flow Diagram | 14 |
| 4 | Sequence Diagram | 15 |
| Ch# 3 | **Design** |  |
| 1 | Login Page | 16 |
| 2 | Signup Page | 17 |
| 3 | First Food Item | 18 |
| 4 | Second Food Item | 19 |
| 5 | Third Food Item | 20 |
| 6 | Fourth Food Item | 21 |
| 7 | HTML Code | 22 |
| Ch# 4 | **Testing** |  |
| 1 | Unit Testing | 33 |
| 2 | Black Box Testing | 34 |
| 3 | White Box Testing | 34 |
| 4 | Integration Testing | 36 |
|  | **References** | 38 |

**Chapter 1**

**Introduction**

The labor rates are increasing steadily year on year thus making it difficult to find employees. The food industry is highly labor intensive and the biggest expense in the food industry is the cost of employing the right kind of people to do the work. One of the ways to reduce this expense is to use modern technology to replace some of the jobs done by human beings and make machines do the work process delivery website is proposed here which simplifies the food ordering process. A food delivery website is a place where customers can directly order from the restaurant or hotel instead of going through a third-party food delivery business. It is a web- based ordering system where customers using internet can use the food delivery website to order online This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

# History

The first online food order was a pizza from Pizza Hut in 1994.This is a picture from a 2018 Pizza Hut pizza box, which describes the first online food sale. The online food ordering market has increased in the U.S with 40 percent of U.S adults having ordered their food online once. The online food ordering market includes foods prepared by Restaurants prepared by independent people and groceries being ordered online and then picked up or delivered the first online food ordering service, World Wide Waiter (now known as Waiter.com), was founded in 1995. The site originally serviced only northern California, later expanding to several additional cities in the United States



**First online food order 1994 (Fig 1)**

By the late 2000s, major pizza chains had created their own mobile applications and started doing 20–30 percent of their business online. With increased Smartphone penetration, and the growth of both Uber and the sharing economy, food delivery startups started to receive more attention. In 2010, Snapfinger, who is a multi-restaurant ordering website, had a growth in their mobile food orders by 17 percent in one year.

By 2015, online ordering began overtaking phone ordering. In 2015, China’s online food ordering and delivery market grew from 0.15 billion Yuan to 44.25 billion Yuan. As of September 2016, online delivery accounted for about 3 percent of the 61 billion U.S. restaurant transactions.

In a 2019 market study of restaurant delivery services, the global market for online-ordered prepared food delivery was estimated at $94 billion and is estimated to grow at just over 9 percent a year, reaching $134.5 billion in 2023. The study defined the market as 1)”meals ordered online which are directly delivered by the restaurant, no matter if ordered via a platform (e.g. Delivery Hero) or a restaurant website (e.g. Domino’s)”; 2) online meal orders and deliveries “both carried out by a platform” (Deliveroo, Uber Eats, e.g.); 3) “online orders that are picked up in the restaurant” by the customer. It does not include phone orders.

After 2020, COVID-19 significantly boosted online food delivery usage worldwide. According to research conducted by the NDP Group, online restaurant ordering is growing 300% faster than dine-in traffic.“Online ordering has started to become the norm, thanks to the convenience, accuracy, and ability to integrate payments. At scale, ubiquitous on-demand and subscription delivery of prepared food could potentially spell the end of cooking at home.”

**Meituan food delivery worker uniform yellow in China (Fig 2)**

1. **Existing System.**

The existing system happens to be a non-computerized operating system were all operations are done manually by the waiter carrying paper and to take down the order of the customer or making an order over the counter. This leads to mistakes because the waiter might not understand what the customer had ordered therefore serving him/her a different menu. This could be so embarrassing because the customer might not take it lightly with the waiter which may lead to misunderstanding.

1. **Disadvantages of existing system.**

Due to manual means being employed by the fast-food restaurants, it is very difficult to satisfy the wants and needs of the customers. Most of the problems include:

1. Mistakes are made when taking the orders of the customers.

2. The process of collecting customers’ purchases order is very tedious. This makes it impossible to deliver goods on time.

3. It leads to lack of understanding between the customers and the employees.

4. The record keeping system is poor. Losses of vital records have been reported in the past consequently. Besides, protecting the file system from unauthorized access is a problem that has defiled solution.

5. Unnecessary time is wasted conveying information through the ladder of authority. Management at times seeks to get a copy of the customer’s order form and this may take a lot of time to obtain it.

6. It causes reduction of production flow. These are the major problems facing the existing system and would be corrected with the help of the proposed system.

1. **Proposed system**.

The proposed system is developed to manage ordering activities in fast food restaurant. It helps to record customer submitted orders. The system should cover the following functions in order to support the restaurant’s business process for achieving the objectives:

1. To allow the customer to make order, view order and make changes before submitting their order and allow them make payment through prepayment card or credit card or debit card.

2. To provide interface that allows promotion and menu.

3. To prevent interface that shows customers’ orders detail to front-end and kitchen staffs for delivering customers’ orders

4. Tools that generate reports that can be used for decision making

5. A tool that allows the management to modify the food information such as price, add a new menu and many others as well as tools for managing user, system menu and promotion records.

# Goals and Objectives.

The purpose of an [online ordering system](https://dinamic.io/restaurant-software/online-food-ordering-system.html) is to make itself beneficial for the customer and the business so that they can stay afloat while also serving customers their favorite dishes. With Food Aggregators increasing their commission every quarter, it is unsustainable for the restaurant to manage their restaurant while depending on food delivery orders. The online ordering market is expanding and, especially the [online food ordering](https://dinamic.io/restaurant-software/online-food-ordering-system.html) segment is growing at a very rapid pace. [Food Delivery](https://dinamic.io/restaurant-software/online-food-ordering-system.html) is the preferred way for customers to enjoy food these days. This change has been roughly owed to the pandemic where customers prefer to [order food](https://dinamic.io/restaurant-software/online-food-ordering-system.html) [online](https://dinamic.io/restaurant-software/online-food-ordering-system.html) instead of dining out.

# Problem Statement.

In recent years, with the rapid developments in food delivery services, more and more restaurants have launched their own food delivery applications, in order to attract more customers as well as make it convenient for those who do not want to go out for dinner. Therefore, these kinds of application services have brought good news to customers, especially for salaried employees, because it can provide more choices on the application’s food menu and it allows for less wait time. It brings people a lot of convenience, but at the same time, problems may arise. For the restaurants, not only are they bearing all of the delivery cost, but they also must guarantee the food’s freshness and quality and whether they can deliver on time. Furthermore, food delivery service is a big challenge for restaurant delivery persons that are without professional training.

**According to market research, we can draw the following conclusions.**

1. The delivery capacity is very limited for normal food outlets. Generally, each person is only able to carry about two to three servings of food at one time,
2. Being unfamiliar with the geographical position, even if they are only able to carry about two to three servings of food at one time, they are unable to outline a reasonable timetable and trips.
3. They are not professional drivers, so they don’t have security during the period of delivery (Massachusetts safety offices league, 2012), The efficiency is not high. Each delivery person is employed by a restaurant and they are unable to work for another restaurant until their contract ends and as a result, they earn very little money each day.
4. In addition, customers have to apply for different account numbers for different restaurants. Not only it is very inconvenient, but also, they might often forget the account number or password. It is really, in a sense, reducing the customer’s information security. In summary, for these reasons, it is essential for a food delivery application to develop a study on how to improve production and management efficiency, and reduce agency cost in competitive markets to maximize profit.

# Solution of Problems.

What we want to do is to develop a mobile application that could solve these problems. We want to reduce the cost restaurants are paying to their drivers, but we are not able to establish our own delivery team at the moment. We referenced UBER and Airbnb with their sharing economy model, and we decided to adopt that mode into our concept. Matthew Crosby, the manager of the Rocky Mountain Institute, offers us clues about the advantage of doing so: “Shared economy companies unbundle existing assets and enable value exchange out of those assets, with close to zero marginal capital cost since the users themselves own the actual physical assets, whether a car or a home” (2015). This mode would allow our application to offer lower costs and better service. That is, a driver does not need to be someone employed by the restaurant. Anyone who wants to be a driver could register for our application and commit to deliver food.

1. The restaurant could cut down the budget of hiring many drivers.
2. There usually could be more than 10 restaurants within a few blocks. This gives drivers’ broader business opportunities. They now are not limited in a unique restaurant but they have choices to accept the delivery orders they want.
3. A restaurant usually sets a range of delivery service, which if exceeded; they would refuse to deliver that order. While a driver could select whichever order he wants to work on, even though it would cost him 20 minutes getting there ─ they could get more tips by the customer.
4. Working time becomes more flexible. Drivers could get online at any time when they want to accept orders and deliver food.

**This also brings various benefits to customers while ordering food.**

1. Being able to choose from a list of restaurants. This saves time for them to figure out from where they could order food,
2. Remembering a set of account numbers and passwords for different websites would be tough for elderly people, but now, this can all be on one application, many restaurants have their own reward system, while we might develop a new one that could be applied to every restaurant listed in our application. After solving our potential
3. problems, the next step is to design the application and its features.

# Functional Requirements.

### Login to Website.

First of all user login to the website if he/she is already registered. If he/she is not registered they will create their account on this website.

### Visit the website.

For order any food from This Website Customer Needs to Visit This Website. Customer Only Visit This Website When She/he has Access to Internet.

### Navigate the restaurant’s menus.

After Opening The Website Customer Will See The Food Menus Of Restaurant or Hotel. And The Prices Of Foods Will Be Written Along With The Food Items.

### Select food from menus.

Then Customer Will Select the Food Of His/her Desire The Restaurant Or Hotel Menu. The Prices of Food Will Be Written Along With Foods This Will Help the Customer to select The Food According to His/her Money.

### Written the payment and delivery details.

After Selecting Food Customer Will Tell The Payment Method Of Food. It’s Mean She/he Will Pay for His/her Food via Cash or via Debit Card or via Mobile Account. And Customer Will Also Tell The His/her Address.

### Place an order.

Then Customer Will Order His/her Food By Clicking The Order Button On Website. And Restaurant or Hotel Will Receive the Order.

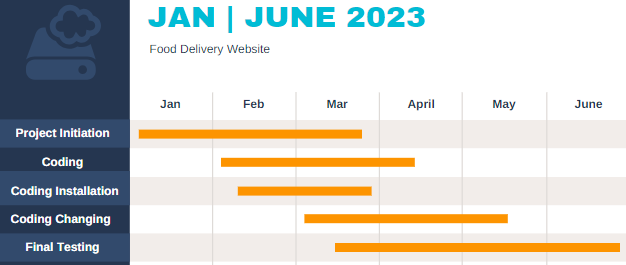
### Receive the confirmation of order.

When Customer Order His/her Food. Website Will Check All Customers Details if The Detail Given By Customer Is Correct then Website Will Show on Screen “**Your Food Is Successfully Order**”. If The Detail is wrongs, then it Will Show “**Please Give Correct Details**”.

# Non-Functional Requirements.

* + Performance.
  + Scalability.
  + Portability.
  + Compatibility.
  + Reliability.
  + Availability.
  + Maintainability.
  + Security.
  + Localization.
  + Usability.

1. **Tools and Technologies.**
   * Html.
   * CSS.
   * JavaScript.
   * Bootstrap
   * JQuery
   * Visual studio code.
   * Web Browser: IE 10 or above, Mozilla FF 31 and above or Google Chrome.
   * Operating System: Windows (Vista/7 or above).
2. **Gantt Chart**



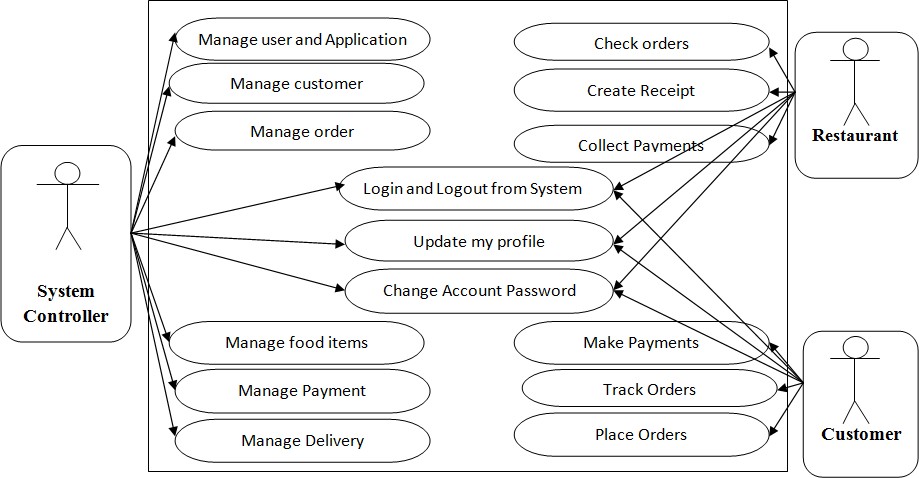
# Chapter 2

# Methodology

# A methodology is collection of methods, practices, processes, techniques, procedures, and rules. In project management, methodologies are specific and strict and usually contain a series of steps and activities for each phase of the project’s life cycle. They have defined approaches showing us precisely what steps to take next, the motivation behind each step, and how a project stage should be performed.

# Use Case diagram of Food Delivery Website.

The use case in the diagram represents the main processes in an Online Food Delivery Website. then they will be broken down into more specific use cases depending on the included processes of the main use case. Each of these use cases explains how the system handles the actions or scenarios requested by the user.

Food Delivery Website Use Case Diagram is a design used as one of the Methodology on Food Delivery. Website development It represents the main functions or processes of the system as well as the specific processes included. They were also labeled properly to guide programmers and users about the structure of Food Delivery Website.

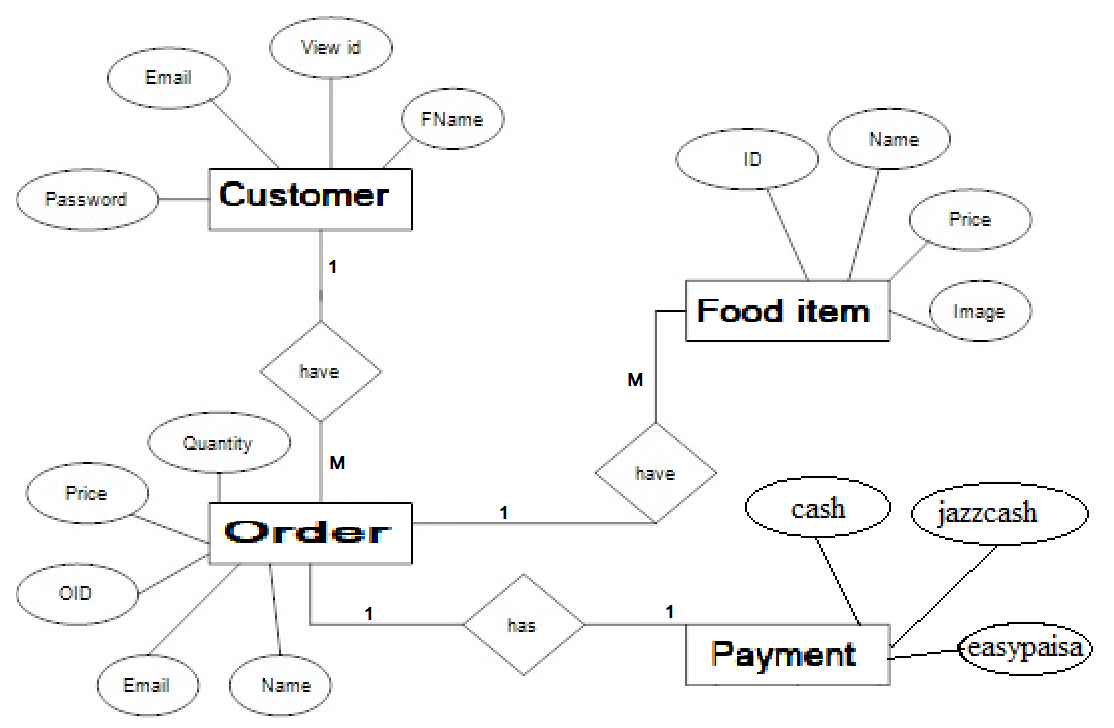
**Use Case diagram (Fig 3)**

1. **ER Diagram for Food Delivery Website.**

Food Delivery Website System Entity-Relationship (ER) diagram can showcase all the entities that work together to make sure the ordering of any fast food is successful. There are entities like Customer, Order, Food Item, and Payment in an ER diagram for a Food Delivery Website system.

Entity Customer have attributes like view id, First Name, Email, Password.Entity Order has attributes like quantity, Price, Order id. Email, Order Name.Entity Food item have attributes like Order id, Name, Price, Image.Entity Payment has attributes like Cash, Jazzcash, Easypaisa.

As the below online Food delivery website ER diagram suggests, rectangles illustrate entity types, whereas ellipses represent attributes. On the other hand, lines link attributes to entity types and entity types with different relationship types. (1) To (M) shows one To Many Relationships and (1) to (1) shows one to one Relationship. The relationship between Customer to order is one To many order to food is one To Many and order to Payment is one to one.

****

**ER Diagram (Fig 4)**

1. **DFD for Food Delivery Website.**

**Food Delivery websites** are very dominant these days. To emphasize how it works, a DFD for the food ordering system is applied. The **DFD** (Data Flow Diagram) represents the **system’s flow of data and transformations.** This transformation occurs when data enters and exits a system. **DFD**represents and describes the **food delivery system as a whole,** including **input, processing,** and **output.**

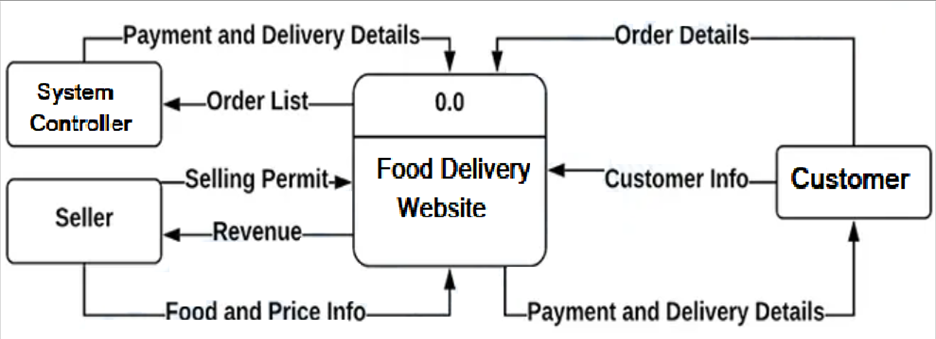
The DFD diagram of the food ordering system consists of the following diagrams:

* Context Diagram (Level 0)
* DFD Level 1
* DFD Level 2

Let us know first the system’s inputs and outputs, processing, and databases.

* 1. **Level 0 DFD of Food Ordering System.**

A **context diagram (level 0 data-flow diagram)** defines the **boundaries**of the **food Delivery processes**. It demonstrates how data flows between the system and external entities. Through the **DFD level 0**, a single step is applied to illustrate the system’s entire notion. The arrows in the diagram indicate the flow direction of the data input. **Customers** (buyers), **sellers**, and the **system Controller** are external entities that cause the system to perform a certain function. Food delivery website” is the name of the primary operation. This means that 0.0 initiate the idea to further discuss the main function. In general, the food Delivery system DFD level 0 serves as the starting point for the illustrations that follow.



**Level Zero DFD (Fig 5)**

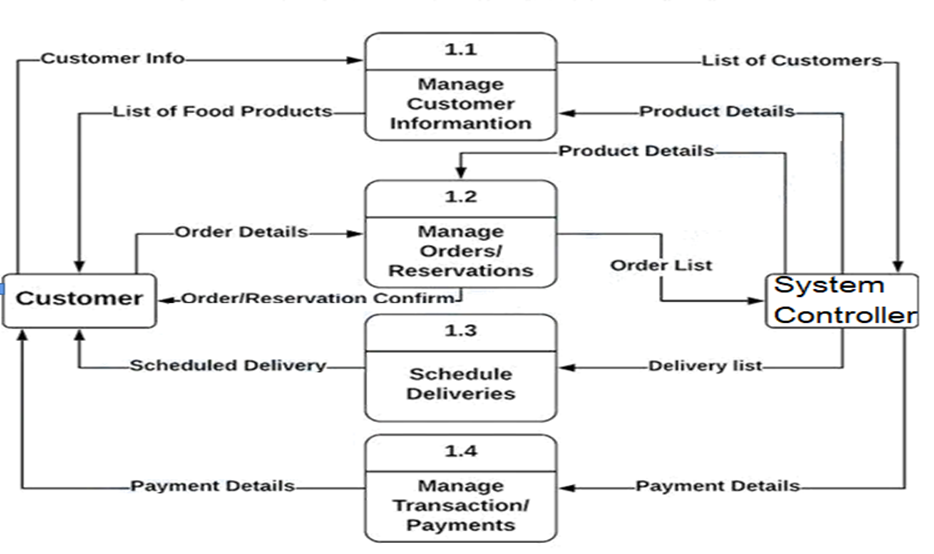
## ****Level 1 DFD of Food Ordering System****

The **1st level DFD** of the **food delivery system**describes each of the system’s primary sub-processes. This level represents the context diagram’s “extended viewpoint.” basic data. These data are used for orders, deliveries, and payment purposes. Now the system also needs to manage orders and reservations from customers. This occurs when a customer orders food using the system or reserves a meal and grabs it from the store. For some customers who want to have their food delivered to their houses, the system will automatically schedule deliveries. This process is optional since there are other customers who pick up their orders from the store. The system also manages the transactions and payments of each customer to complete the overall function. Then the transactions were saved in the database for inventory purposes. Take note that these ideas were derived from the common activities of actual food ordering management. You can also modify the given idea to meet your desired functions or use it as it is. At this level, you already understand the project’s overall work and you may end your work here. However, the next level might also intrigue you about what is and how does it work?

At this level, a single process is explained by focusing on its smaller parts, which include:

* Manage Customer Information
* Manage Orders and Reservations
* Schedule Deliveries
* Manage Transaction and Payments

The process of managing customer information includes gathering customers’

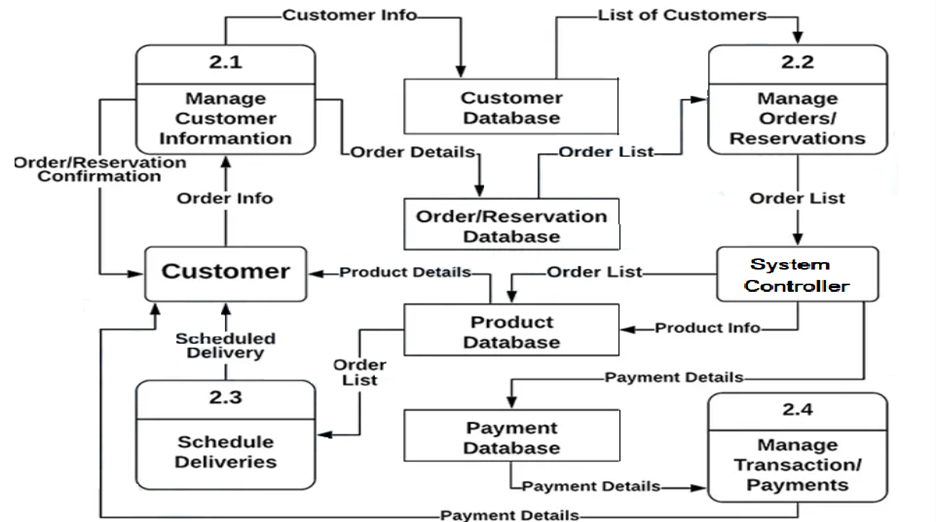
 **Level 1 DFD (Fig 6)**

**Level 2 DFD of Food Ordering System**

The DFD level 2 is the highest concept abstraction among the mentioned levels. The reason why is that this level specifies the processes (if there are any) under the sub-process in level 1. But let’s talk about an important part of the data flow diagram at this level. Supposedly, the example should explain one of the sub-processes in level 1. Instead, it gives more emphasis to the system’s database (data store). The data stores that the system needs are as follows:

* Customer Database
* Orders Database
* Product Database
* Payment Database

These data stores are responsible for keeping the data secure and available when needed. The information that the system gives will only go to the authorized user and owner of the information. The food delivery website is a project that provides efficient processing of orders through the internet. This means that system should work as how manual food ordering does. The only difference is that the system enables customers to order whenever and wherever they are.

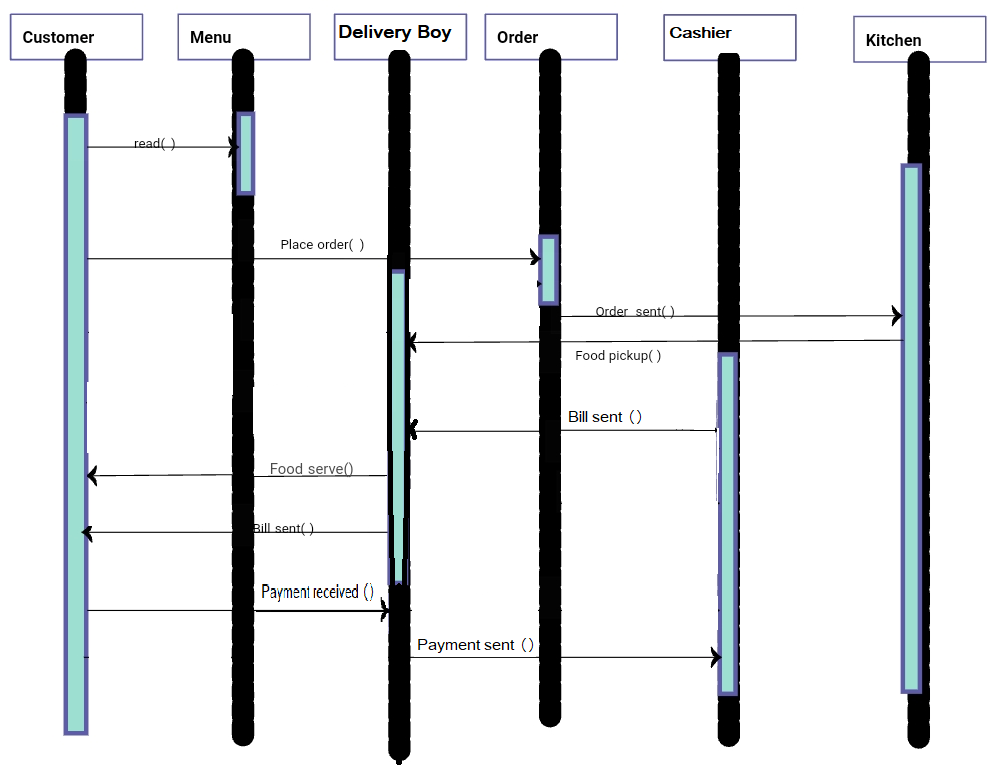


**Level 2 DFD (Fig 7)**

1. **Sequence Diagram Of Food Delivery Website.**

The sequence diagram example shows six participating objects: Customer, Menu, Delivery Boy, Order, Cashier and the kitchen.

* Step 1: Customer reads menu.
* Step 2: Customer Place Order.
* Step 3: order sent to kitchen.
* Step 4: Food Pickup from Kitchen to Delivery Boy.
* Step 5: Cashier Sent Bill To Delivery Boy.
* Step 6: Delivery Boy Delivered Food To Customer
* Step 7: Delivery Boy gave the bill to Customer.
* Step 8: If the product is available, it is added to the order.
* Step 9: Customer Gave The Payment of Food to Delivery Boy.
* Step 10: Delivery Boy Gave The Payment Of Food To Cashier.



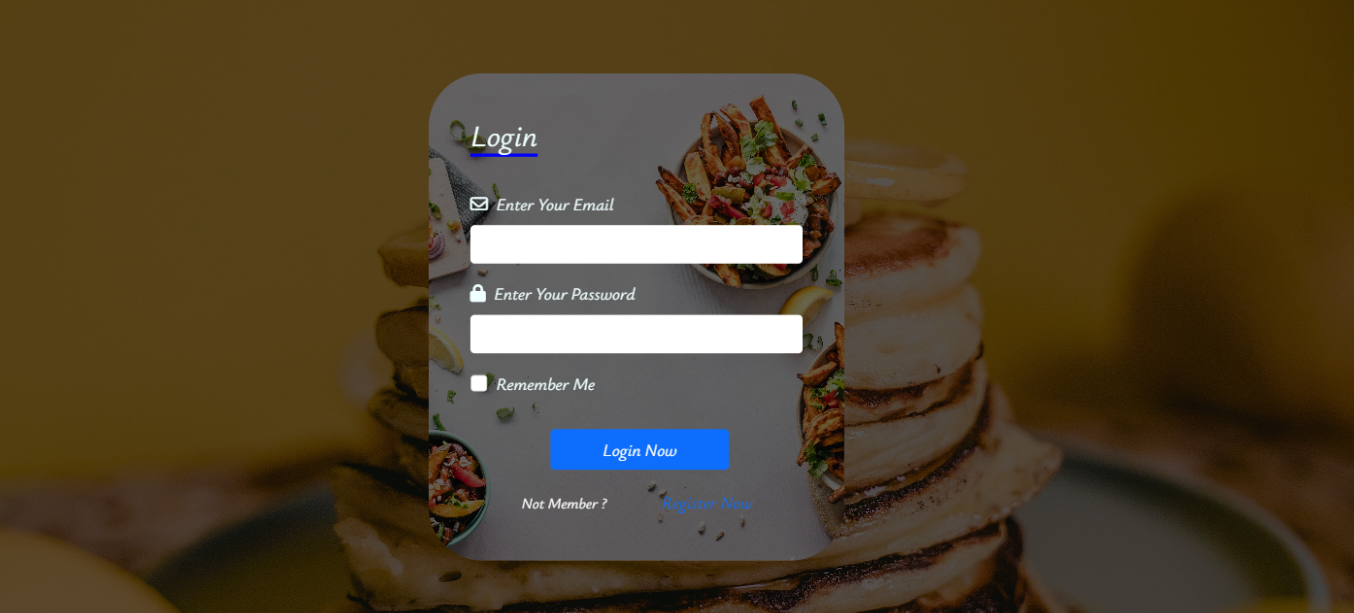
**Sequence Diagram (Fig 8)**

**Chapter 3**

**Design**

* 1. **Login Page.**

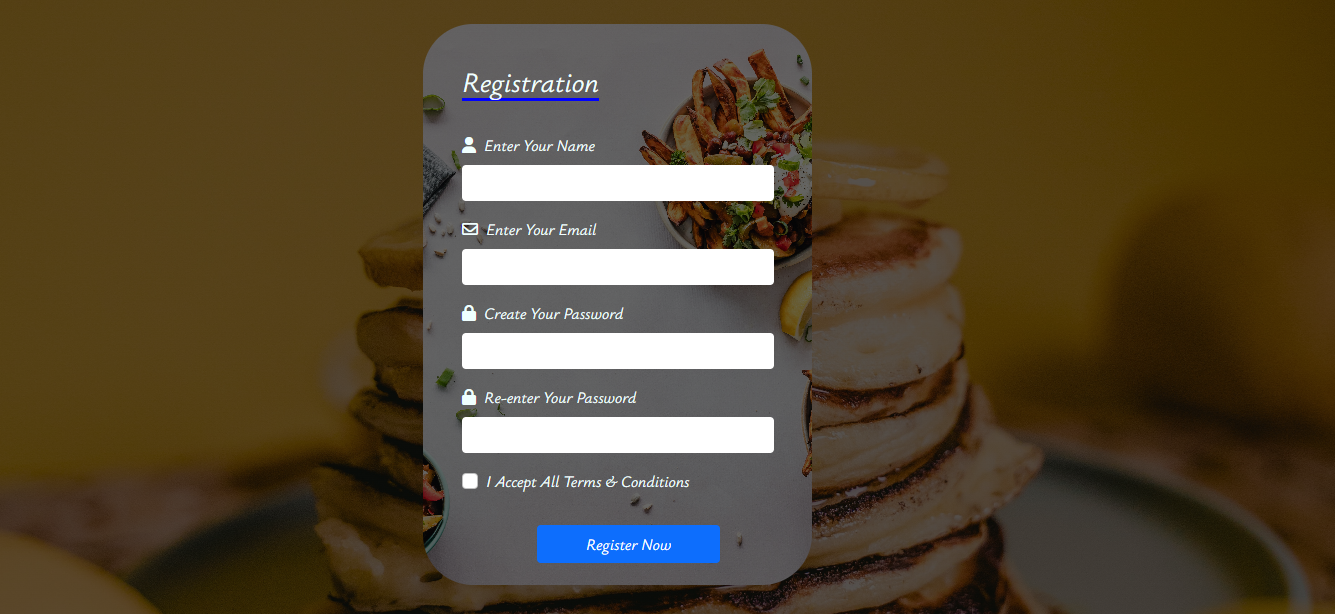
This is First Page of website when user will open the website if user is already registered on website, then she/he will login using email and Password is user not already registered she/he will click on signup now and signup page will be open.

****

**Login Page (Fig 9)**

* 1. **Signup Page.**

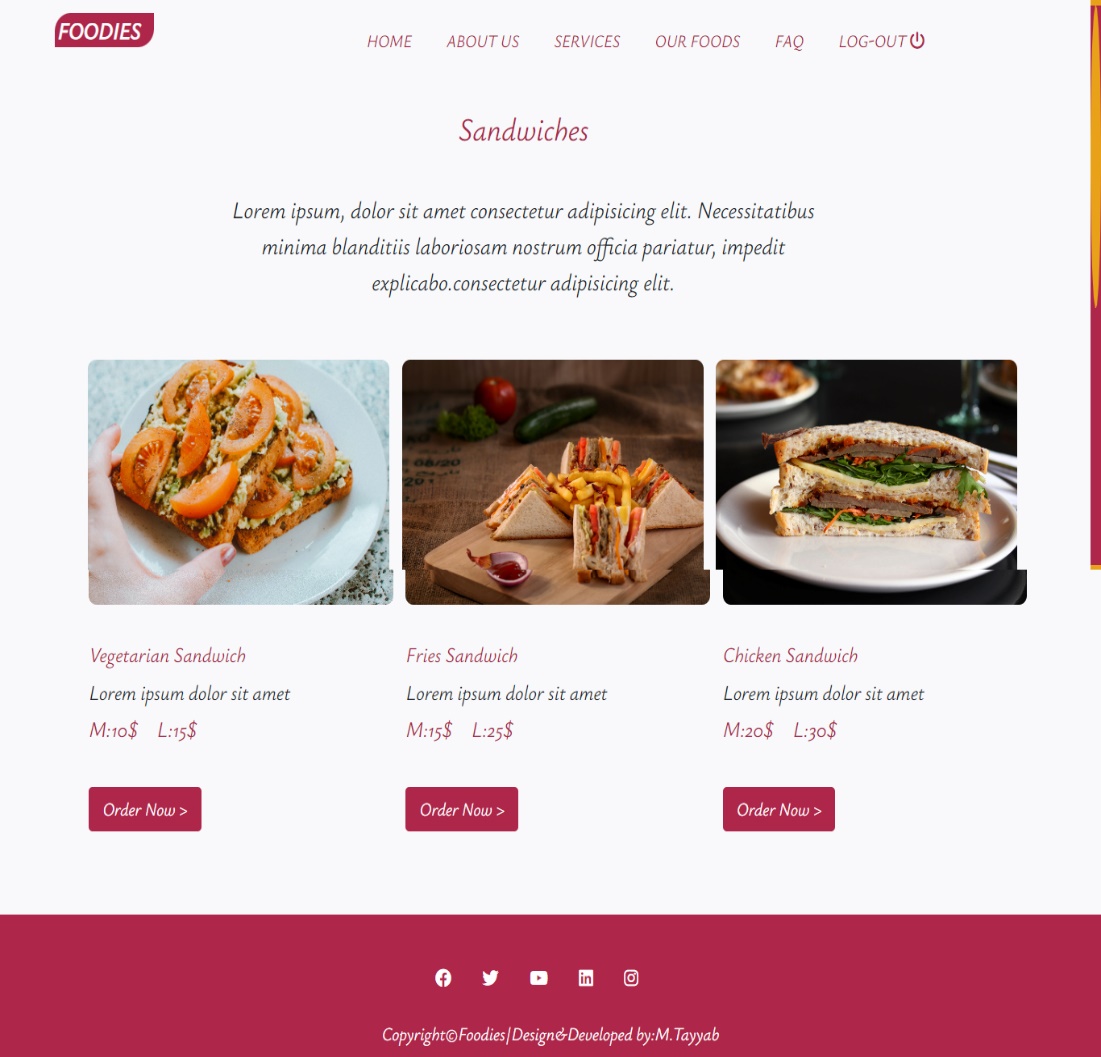
If user is new on this website the user has to register his/her account on this website. For registration user will enter his/her name email address and password and confirm Password by enter password again. Click on given blank of (I accept all terms & conditions) and when user will click on Register Now Button, she/he will be Registered on this website.



**Signup Page (Fig 10)**

* 1. **First Food item.**

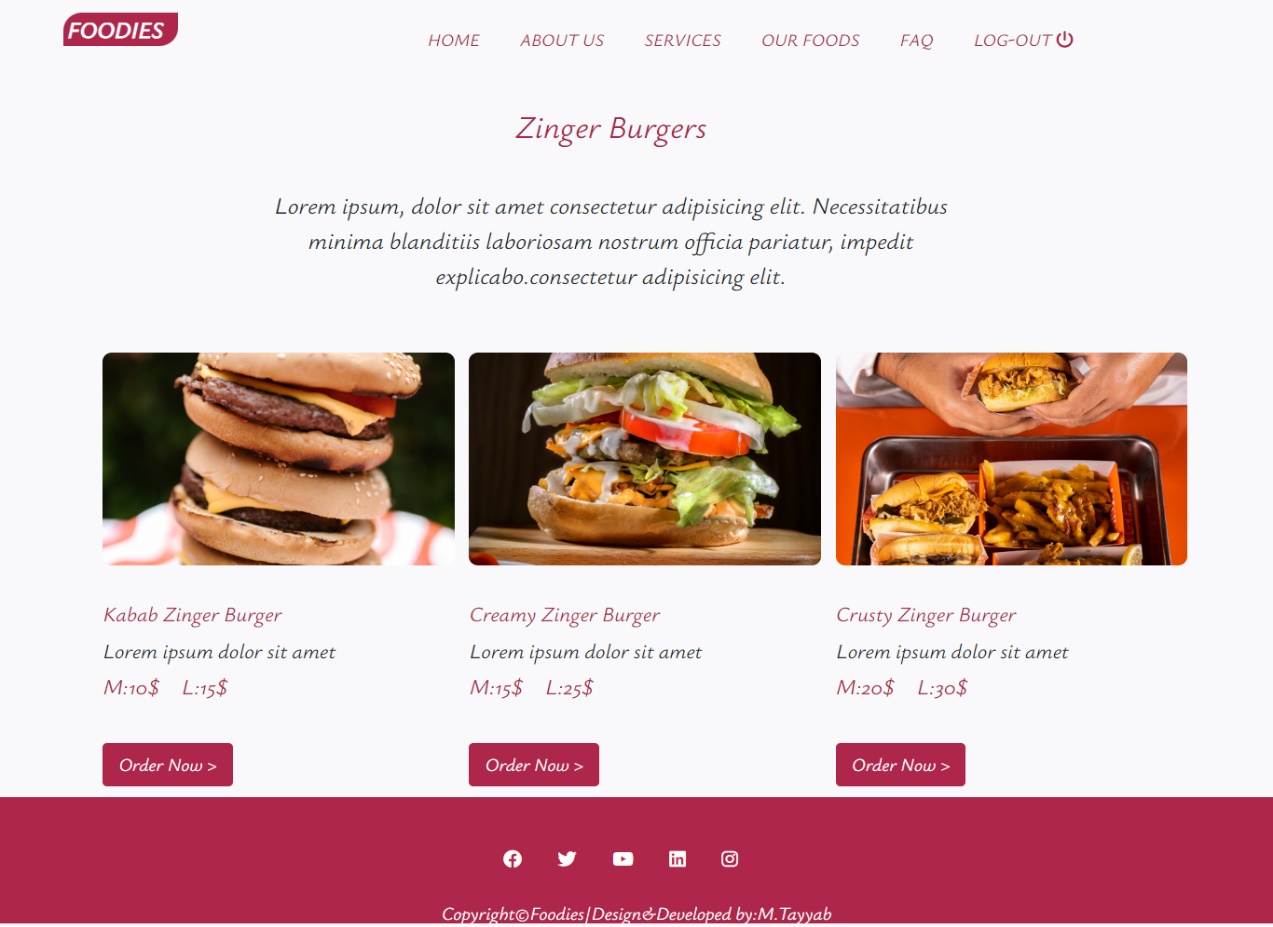
In Our Front food menus are given customer will click on food she/he want to order. on the first number of list Sandwich is given if she/he want to order Sandwich she/he will click on View More Button than all type of Sandwiches will be shown with price. Customer will click on any type of Sandwich for delivery process.



**First Food item (Fig 11)**

* 1. **Second food item.**

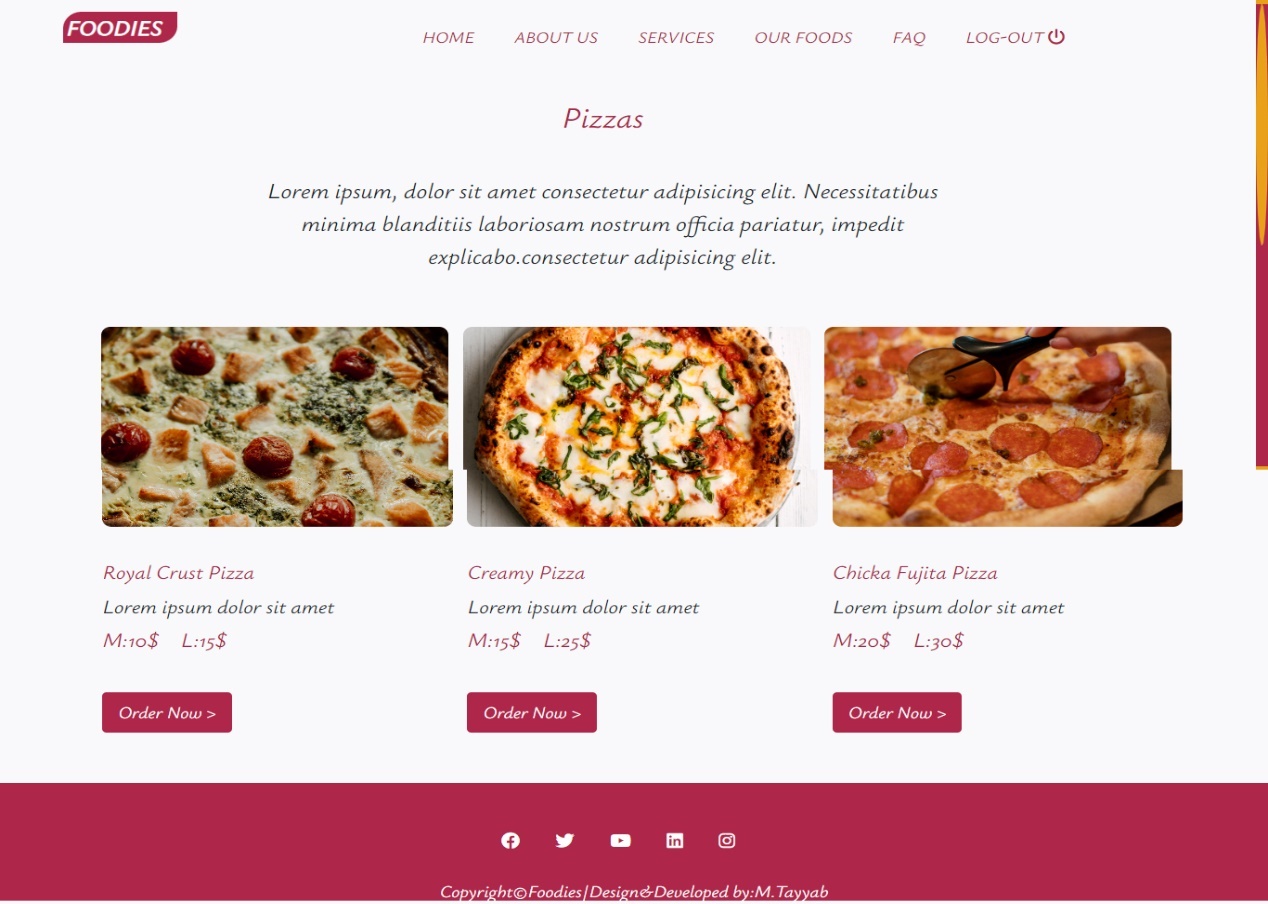
In Our Front food menus are given customer will click on food she/he want to order. on the second number of list Zinger Burger is given if she/he want to order Zinger Burger she/he will click on View More Button than all type of Zinger Burgers will be shown with price. Customer will click on any type of Zinger Burgers for delivery process.



**Second Food item (Fig 12)**

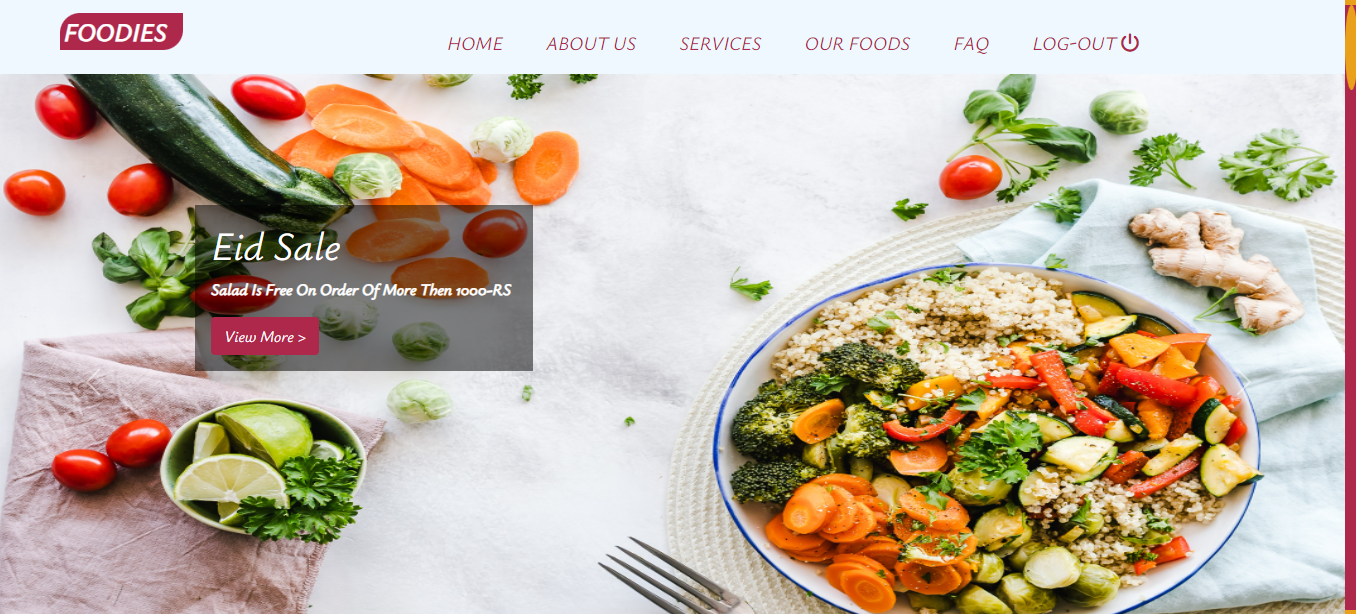
* 1. **Third Food Item.**

In Our Front food menus are given customer will click on food she/he want to order. on the Third number of list Pizza is given if she/he want to order Pizza she/he will click on View More Button than all type of Pizzas will be shown with price. Customer will click on any type of Pizza for delivery process.

****  
**Third Food item (Fig 13)**

* 1. **Landing Page.**

This is First Page When a User Login to Website it has Five Sections First Section Is Home Second is About Us Third is Services Fourth is Our Foods and Last Is Frequ-ently Asked Questions.

****

**Landing Page (Fig 14)**

* 1. **HTML Coding**

<!DOCTYPE *html*>

<html *lang*="en">

<head>

  <meta *charset*="UTF-8">

  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

  <title>Foodies</title>

  <link *rel*="icon" *href*="2.png">

  <link *rel*="stylesheet" *href*="index.css">

  <link *rel*="stylesheet" *href*="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"

*integrity*="sha512-iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46ZmdtAc9eNBvH0H/ZpiBw=="

*crossorigin*="anonymous" *referrerpolicy*="no-referrer" />

  <link *rel*="preconnect" *href*="https://fonts.googleapis.com">

  <link *rel*="preconnect" *href*="https://fonts.gstatic.com" *crossorigin*>

  <link

*href*="https://fonts.googleapis.com/css2?family=Kaushan+Script&family=Poppins:wght@100;400&family=Sigmar&family=Ysabeau:ital@1&display=swap"

*rel*="stylesheet">

*<!-- Bootstrap CSS -->*

  <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" *rel*="stylesheet"

*integrity*="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" *crossorigin*="anonymous">

</head>

<body>

  <div *class*="container\_fluid">

    <div *class*="containerr">

      <nav>

        <div *class*="left">

          FOODIES

        </div>

        <ul *class*="right">

          <li *class*="btn btn-lg"><a *href*="#body1">HOME</a></li>

          <li *class*="btn btn-lg"><a *href*="#body2">ABOUT US</a></li>

          <li *class*="btn btn-lg"><a *href*="#body3">SERVICES</a></li>

          <li *class*="btn btn-lg"><a *href*="#body5">OUR FOODS</a></li>

          <li *class*="btn btn-lg"><a *href*="#body6">FAQ</a></li>

          <li *class*="btn btn-lg" *id*="ORDER"><a

*href*="file:///E:/Web%20Developmwnet%20DP/CSS/Projects/FOODIES/form.html">LOG-OUT <i

*class*="fa-solid fa-power-off"></i></a></li>

*<!-- <li id="ORDER">LOG-OUT <i class="fa-solid fa-power-off"></i></li>  -->*

        </ul>

      </nav>

      <div *class*="mt-4">

        <div *id*="carouselExampleIndicators" *class*="carousel slide " *data-bs-ride*="carousel">

          <div *class*="carousel-indicators pb-5">

            <button *type*="button" *data-bs-target*="#carouselExampleIndicators" *data-bs-slide-to*="0" *class*="active"

*aria-current*="true" *aria-label*="Slide 1"></button>

            <button *type*="button" *data-bs-target*="#carouselExampleIndicators" *data-bs-slide-to*="1"

*aria-label*="Slide 2"></button>

            <button *type*="button" *data-bs-target*="#carouselExampleIndicators" *data-bs-slide-to*="2"

*aria-label*="Slide 3"></button>

          </div>

          <div *class*="carousel-inner">

            <div *class*="carousel-item active " *data-bs-interval*="3000">

              <img *src*="slider-img/1.jpg" *class*="d-block img-fluid " *alt*="...">

              <div *class*="carousel-caption text-start d-none d-md-block p-3 w-25 ">

                <h1>Eid Sale</h1>

                <p *class*="fw-bolder">Buy 2 Zinger Burger Get 1 Zinger Burger Free</p>

                <a *class*="btn " *href*="burgers.html" *id*="button1">View More ></a>

              </div>

            </div>

            <div *class*="carousel-item" *data-bs-interval*="3000">

              <img *src*="slider-img/2.jpg" *class*="d-block img-fluid " *alt*="...">

*<!--  -->*

              <div *class*="carousel-caption text-start d-none d-md-block p-3 w-25 ">

                <h1>Eid Sale</h1>

                <p *class*="fw-bolder">Salad Is Free On Order Of More Then 1000-RS</p>

                <a *class*="btn " *href*="Sandwich.html" *id*="button1">View More ></a>

              </div>

            </div>

            <div *class*="carousel-item" *data-bs-interval*="3000">

              <img *src*="slider-img/3.jpg" *class*="d-block img-fluid " *alt*="...">

*<!--  -->*

              <div *class*="carousel-caption text-start d-none d-md-block p-3 w-25 ">

                <h1>Eid Sale</h1>

                <p *class*="fw-bolder">1 Pasta Is Free on Every Deal Of More Then 1500-RS</p>

                <a *class*="btn " *href*="pizza.html" *id*="button1">View More ></a>

              </div>

            </div>

          </div>

          <button *class*="carousel-control-prev" *type*="button" *data-bs-target*="#carouselExampleIndicators"

*data-bs-slide*="prev">

            <span *class*="carousel-control-prev-icon" *aria-hidden*="true"></span>

            <span *class*="visually-hidden">Previous</span>

          </button>

          <button *class*="carousel-control-next" *type*="button" *data-bs-target*="#carouselExampleIndicators"

*data-bs-slide*="next">

            <span *class*="carousel-control-next-icon" *aria-hidden*="true"></span>

            <span *class*="visually-hidden">Next</span>

          </button>

        </div>

      </div>

      <div *class*="body1" *id*="body1">

        <div *class*="bodyleft">

          <h2>Good Food Choices Are <br> Good Investments</h2>

          <br>

          <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus minima blanditiis laboriosam

            nostrum officia pariatur, impedit explicabo.</p>

          <br>

          <a *href*="#body5" *class*="btn btn-lg">Check Now</a> *<!-- Button trigger modal -->*

          <button *type*="button" *class*="btn btn-lg " *id*="button1" *data-bs-toggle*="modal" *data-bs-target*="#exampleModal">

            Order Now

          </button>

*<!-- Modal -->*

          <div *class*="modal fade" *id*="exampleModal" *tabindex*="-1" *aria-labelledby*="exampleModalLabel"

*aria-hidden*="true">

            <div *class*="modal-dialog">

              <div *class*="modal-content">

                <div *class*="modal-header">

                  <h5 *class*="modal-title" *id*="exampleModalLabel">Order Details</h5>

                  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>

                </div>

                <div *class*="modal-body">

                  <form>

                    <div *class*="mb-3">

                      <label *for*="exampleInputEmail1" *class*="form-label">Enter Food Name</label>

                      <input *type*="Text" *class*="form-control" *id*="exampleInputEmail1" *aria-describedby*="emailHelp">

                    </div>

                    <div *class*="dropdown">

                      <button *class*="btn w-50 dropdown-toggle mb-3 dropbtn" *type*="button" *id*="dropdownMenuButton1"

*data-bs-toggle*="dropdown" *aria-expanded*="false">

                        Select item Size

                      </button>

                      <ul *class*="dropdown-menu w-50 " *aria-labelledby*="dropdownMenuButton1">

                        <li><a *class*="dropdown-item w-50" *href*="#">Small</a></li>

                        <li><a *class*="dropdown-item w-50" *href*="#">Medium</a></li>

                        <li><a *class*="dropdown-item w-50" *href*="#">Large</a></li>

                      </ul>

                    </div>

                    <div *class*="mb-3">

                      <label *for*="exampleInputPassword1" *class*="form-label">Enter Phone Number</label>

                      <input *type*="tel" *class*="form-control" *id*="exampleInputPassword1">

                    </div>

                    <div *class*="mb-3">

                      <label *for*="exampleInputEmail1" *class*="form-label">Enter Complete Address</label>

                      <input *type*="Text" *class*="form-control" *id*="exampleInputEmail1" *aria-describedby*="emailHelp">

                    </div>

                    <div *class*="mb-3 form-check">

                      <input *type*="checkbox" *class*="form-check-input" *id*="exampleCheck1">

                      <label *class*="form-check-label" *for*="exampleCheck1">Check me out</label>

                    </div>

                    <button *type*="submit" *class*="btn btn-lg " *id*="button1">Order</button>

                  </form>

                </div>

                <div *class*="modal-footer">

                  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Close</button>

                </div>

              </div>

            </div>

          </div>

        </div>

        <div *class*="bodyright">

        </div>

      </div>

      <div *class*="body2" *id*="body2">

        <div *class*="body2right">

          <h2>We Provide Ourselves On <br>Making Real Food From The <br> Best Ingridiants</h2>

          <br>

          <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus minima blanditiis laboriosam

            nostrum officia pariatur, impedit explicabo.</p>

          <br>

          <a *href*="#" *class*="btn btn-lg">Read More</a>

        </div>

        <div *class*="body2left">

        </div>

      </div>

      <div *class*="body3" *id*="body3">

        <div *class*="body3left">

          <h2>We Make Everything By Hand<br>With the Best Possible<br>Ingrediants</h2>

          <br>

          <p>minima blanditiis laboriosam nostrum officia pariatur, impedit explicabo.</p>

          <br>

          <p *class*="menu"><i *class*="fa-solid fa-check"></i>Sandwich</p>

          <p *class*="menu"><i *class*="fa-solid fa-check"></i>Zinger Burger</p>

          <p *class*="menu"><i *class*="fa-solid fa-check"></i>Pizza</p>

*<!-- Button trigger modal -->*

          <button *type*="button" *class*="btn btn-lg" *id*="button1" *data-bs-toggle*="modal" *data-bs-target*="#exampleModal">

            Order Now

          </button>

*<!-- Modal -->*

          <div *class*="modal fade" *id*="exampleModal" *tabindex*="-1" *aria-labelledby*="exampleModalLabel"

*aria-hidden*="true">

            <div *class*="modal-dialog">

              <div *class*="modal-content">

                <div *class*="modal-header">

                  <h5 *class*="modal-title" *id*="exampleModalLabel">Order Details</h5>

                  <button *type*="button" *class*="btn-close" *data-bs-dismiss*="modal" *aria-label*="Close"></button>

                </div>

                <div *class*="modal-body">

                  <form>

                    <div *class*="mb-3">

                      <label *for*="exampleInputEmail1" *class*="form-label">Enter Food Name</label>

                      <input *type*="Text" *class*="form-control" *id*="exampleInputEmail1" *aria-describedby*="emailHelp">

                    </div>

                    <div *class*="dropdown">

                      <button *class*="btn w-50 dropdown-toggle mb-3 dropbtn" *type*="button" *id*="dropdownMenuButton1"

*data-bs-toggle*="dropdown" *aria-expanded*="false">

                        Select item Size

                      </button>

                      <ul *class*="dropdown-menu w-50 " *aria-labelledby*="dropdownMenuButton1">

                        <li><a *class*="dropdown-item w-50" *href*="#">Small</a></li>

                        <li><a *class*="dropdown-item w-50" *href*="#">Medium</a></li>

                        <li><a *class*="dropdown-item w-50" *href*="#">Large</a></li>

                      </ul>

                    </div>

                    <div *class*="mb-3">

                      <label *for*="exampleInputPassword1" *class*="form-label">Enter Phone Number</label>

                      <input *type*="tel" *class*="form-control" *id*="exampleInputPassword1">

                    </div>

                    <div *class*="mb-3">

                      <label *for*="exampleInputEmail1" *class*="form-label">Enter Complete Address</label>

                      <input *type*="Text" *class*="form-control" *id*="exampleInputEmail1" *aria-describedby*="emailHelp">

                    </div>

                    <div *class*="mb-3 form-check">

                      <input *type*="checkbox" *class*="form-check-input" *id*="exampleCheck1">

                      <label *class*="form-check-label" *for*="exampleCheck1">Check me out</label>

                    </div>

                    <button *type*="submit" *class*="btn btn-lg " *id*="button1">Order</button>

                  </form>

                </div>

                <div *class*="modal-footer">

                  <button *type*="button" *class*="btn btn-secondary" *data-bs-dismiss*="modal">Close</button>

                </div>

              </div>

            </div>

          </div>

        </div>

        <div *class*="body3right">

          <div *class*="box31"></div>

          <div *class*="box32"></div>

          <div *class*="box33"></div>

          <div *class*="box34"></div>

        </div>

      </div>

      <div *class*="body4" *id*="body4">

        <h2>When A Man Stomach Is Full Its Make No Different<br>Whether He Is Rich and Poor</h2>

        <br>

        <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus minima blanditiis laboriosam nostrum

          officia pariatur, impedit explicabo.</p>

        <br>

        <a *href*="#" *class*="btn btn-lg">Read More ></a>

        <br>

        <br>

      </div>

      <div *class*="body5" *id*="body5">

        <h2>Explore Our Foods</h2>

        <br>

        <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Necessitatibus minima blanditiis laboriosam nostrum

          officia pariatur, impedit explicabo.consectetur adipisicing elit. </p>

        <div *class*="body5cards">

          <div *class*="body5cards1">

            <div *class*="body5cards11">

            </div>

            <br>

            <h3>Sandwich</h3>

            Lorem ipsum dolor sit amet

            <br>

            <h4>M:10$</h4>

            <h4>L:15$</h4>

            <br>

            <a *href*="Sandwich.html" *class*="btn btn-lg pt-0" *id*="button1"> View More ></a>

          </div>

          <div *class*="body5cards1">

            <div *class*="body5cards12">

            </div>

            <br>

            <h3>Zinger Burger</h3>

            Lorem ipsum dolor sit amet

            <br>

            <h4>M:15$</h4>

            <h4>L:25$</h4>

            <br>

            <a *href*="burgers.html" *class*="btn btn-lg pt-0" *id*="button1"> View More ></a>

          </div>

          <div *class*="body5cards1">

            <div *class*="body5cards13">

            </div>

            <br>

            <h3>Pizza</h3>

            Lorem ipsum dolor sit amet

            <br>

            <h4>M:20$</h4>

            <h4>L:30$</h4>

            <br>

            <a *href*="pizza.html" *class*="btn btn-lg pt-0" *id*="button1"> View More ></a>

          </div>

        </div>

      </div>

      <div *class*="body6" *id*="body6">

        <br>

        <h2>Frequently Asked Questions</h2>

        <div *class*="body6left">

          <h3>~Lorem ipsum dolor sit, amet consectetur adipisicing?</h3>

          Lorem ipsum, dolor sit amet consectetur adipisicing elit. Consequuntur fugiat qui quibusdam quasi expedita.

          Suscipit.

          <br>

          <br>

          <br>

          <h3>~Lorem ipsum dolor sit, amet consectetur adipisicing?</h3>

          Lorem ipsum, dolor sit amet consectetur adipisicing elit. Consequuntur fugiat qui quibusdam quasi expedita.

          Suscipit.

        </div>

        <div *class*="body6left">

          <h3>~Lorem ipsum dolor sit, amet consectetur adipisicing?</h3>

          Lorem ipsum, dolor sit amet consectetur adipisicing elit. Consequuntur fugiat qui quibusdam quasi expedita.

          Suscipit.

          <br>

          <br>

          <br>

          <h3>~Lorem ipsum dolor sit, amet consectetur adipisicing ?</h3>

          Lorem ipsum, dolor sit amet consectetur adipisicing elit. Consequuntur fugiat qui quibusdam quasi expedita.

          Suscipit.

        </div>

      </div>

      <div *class*="body7">

        <h3>Is Foodies Breads Really Backed Fresh Each Day</h3>

        <a *href*="" *class*="btn btn-lg">Learn More ></a>

      </div>

      <div *class*="body8">

        <a *href*="https://web.facebook.com/profile.php?id=100091250833316"><i *class*="fa-brands fa-facebook"></i></a>

        <a *href*="https://twitter.com/FoodiesSW"><i *class*="fa-brands fa-twitter"></i></a>

        <a *href*="https://www.youtube.com/@DilsefoodieOfficial"><i *class*="fa-brands fa-youtube"></i></a>

        <a *href*="https://www.linkedin.com/company/foodie"><i *class*="fa-brands fa-linkedin"></i></a>

        <a *href*="https://www.instagram.com/explore/tags/foodie/?hl=en"><i *class*="fa-brands fa-instagram"></i></a>

        <br>

        <br>

        Copyright©Foodies|Design&Developed by:M.Tayyab

      </div>

    </div>

  </div>

*<!-- Option 1: Bootstrap Bundle with Popper -->*

  <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"

*integrity*="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

*crossorigin*="anonymous"></script>

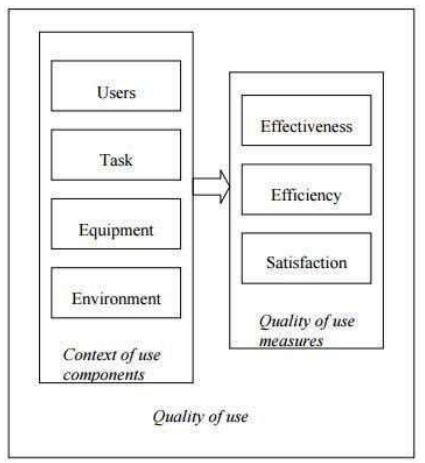
</body>

</html>

**Chapter 4**

**Testing**

Software testing is a process used to identify the correctness, completeness and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software. So that it could be corrected before the product is released to the end users. In simple words, software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed but in the RAD approaches most of the test effort is on-going. The purpose of project evaluation is to assess the software development methodology that was used throughout the development of the framework, assess the usefulness of the technologies and tools, the accuracy of the estimations and the usefulness of the reviews. The solution will be reviewed and evaluated to decide whether it accomplishes the ideas presented in the initial overview and for the quality of the product. Since the software engineering model used in this project was as incremental model, the evaluation for each delivery was obtained by the client at the time the components were delivered. They were mainly review meetings and the feedback obtained was used in corrective maintenance and/or change management. The architecture diagram for the proposed system in Figure in given bellow. Based on a prepared list of criteria along with some practical experimentation, a software evaluation makes it possible to check whether the project objectives were satisfied. Table was designed to collect feedback from the end users of the system. This was distributed among the users to get their feedback anonymously. The evaluator assessed the returned forms in order to obtain the actual feedback of the users and to assess that the project objectives have been met



**Software Evaluation (Fig 15)**

**1. Unit Testing**

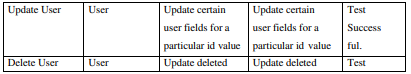
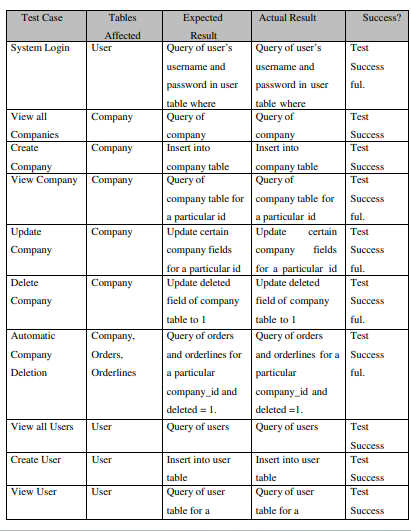
The purpose of Unit Testing is to ensure that each unit in the system functions properly. This testing strategy was applied to each object in the system. A Java test class containing methods to assess objects, calling each method with an assortment of parameters to test whether the values returned were appropriate. These test classes ensured that valid input data was accepted and invalid data was rejected. Each unit within the application was tested to verify that all links and buttons navigated as expected. Usability testing has also been incorporated to examine whether error messages are clear and understood. The database schema was also monitored to ensure insertions, updates and deletes were occurring and with the expected changes. All the tests have proved successful. For those tests which failed initially, the causing factor was identified and remedial action was taken in order to pass the test concerned. The detail of remedial action taken has been documented. Security testing has been incorporated to verify that the expected result is in face, the actual result.

**2. Black Box Testing**

The purpose of Black Box or functional testing is to assess the systems internal workings. However the overall aim of this test strategy is to examine results without knowing how the system arrived at that result. In effect, a tester only requires knowledge of the system specification rather than underlying architecture. This caters for such testing to be performed from an end users perspective rather than a designer perspective. Furthermore, any ambiguities that may exist between the Black Box test results and the original specification are easily detectable, as an unexpected output would occur. The tests performed here are based around users input and actions. The expected and actual system output for each test case has been documented.

**3. White Box Testing**

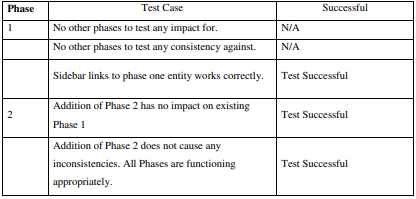
The purpose of White Box testing is to certify that the underlying system architecture functions correctly. This contrasts to Black Box testing which examines system output from knowledge about the systems use of syntax. White Box testing has been undertaken for this system, in order to examine the changes of state for each of the database tables within the system. White Box testing has two immediate benefits. Firstly, by examining the table states during update, create and delete processes for example, the developer is able to verify that the right tables are being queried or affected. Furthermore, the developer can query the record contents, to ensure the correct and relevant fields in one or more tables have been correctly queried or affected, as appropriate. The second benefit is more aimed at the client; as such testing can simulate system stress that it may endure once deployed to see how it reacts. The results of the White Box testing can be found in figure



**Table: Test Cases for Whitebox testing**

**4. Integration Testing**

The purpose of performing Integration Testing is to ensure that the different portions that make up the system function correctly, when combined to form a single working application. Integration Testing was performed at the end of each phase to prove that the system still maintained functionality. The benefits become more apparent as the system increases in size and functionality. For example, at the completion of phase two, Companies and Users were two separate system entities (at that point in development). However, the addition of Access Controls in phase three relies on successful integration with the Users entity, for login and customizable access rights to be successful in their operations. For such reasons, testing occurred at each phase completion to verify that new additions have not affected existing functionality. Navigation from one part of the system to another was also examined, by testing the links within the systems sidebar menu. As mentioned in Section 5, the implementation of CSS helps to uphold a consistent layout format for each page in the system. This was also examined. The outcome of Integration Testing for each phase of the system is shown in table. This phase was executed until no international errors were found.





**Table: Test Cases for integrated test**

**References**

1. Linima Sahu , Rahul Sahu , Sagar Sahu , Prof. Vivek Kumar Sinha, "Click to Eat with Mess Facility" May 2023
2. ["Pizza Hut Tells Twitter It Made The First Online Sale In 1994"](https://www.huffingtonpost.com/2013/09/09/pizza-hut_n_3894981.html). Huffingtonpost.com 2013-09-09. Retrieved 2018-08-14.
3. Anderson, R.G. (1978), “Data processing and Management Information Systems”, McDonald and Evans Ltd
4. [Akaash Amalraj](https://dinamic.io/restaurant-software/authors/akaash-amalraj) , "Purpose 0f Ordering System", 1st November 2021 DiNAMIC : <https://dinamic.io/restaurant-software/blogs/what-is-the-purpose-of-an-online-ordering-system/>