Deen Dayal Upadhyaya College
(University of Delhi)

(Department of Computer Science)

A PROJECT REPORT OF SOFTWARE ENGINEERING ON

"Food Delivery System"

SUBMITTED IN PARTIAL FULFILLMENT ON THE DEGREE IN
BACHELOR OF SCIENCE COMPUTER SCIENCE HONORS (2021-2024)

Submitted by:
1.Rajveer Singh Sengar (21HCS4172)
2. Girish Kumar (21HCS4138)
3.Yogita(21HCS4196)

Submitted to:-
Mr. Anil Kumar
(Associate Professor)
Department of Computer Science
Deen dayal Upadhyaya College

Acknowledgement

We express our sincere gratitude to **Mr. Anil Kumar**, *Associate Professor*, Department of Computer Science, *Deen Dayal Upadhyaya College*, *University of Delhi*, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work. Their insights, expertise and patience have been invaluable to me.Without their support, this project would not have been possible.

CERTIFICATE

This is to certify that the project entitled, *“ONLINE FOOD DELIVERY SYSTEM"* submitted by *"RAJVEER SINGH SENGAR, GIRISH KUMAR, YOGITA"* in partial fulfillment of the requirements for the award of "Bachelor in Computer Science Honors" at the "Deen Dayal Upadhyaya College, University of Delhi" is an authentic work carried out by him under my supervision and guidance.To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Mr. Anil Kumar
(Associate Professor)
Department of Computer Science,
Deen Dayal Upadhyaya College,
University of Delhi

....
Signature

Declaration

We hereby declare that the project work entitled software Development for ***"Food Delivery System Using Web Technologies "*** submitted to the *Deen Dayal Upadhyaya College* , is a record of an original work done by me under the guidance of Mr. Anil Kumar, Associate Professor, *Department Of Computer Science*, Deen Dayal Upadhyaya College, and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of Master of *Bachelors in Computer Science Honors.*
The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Submitted by:-
Rajveer Singh Sengar (21hcs4172)
Girish kumar (21hcs4138)
Yogita (21hcs4196)

INDEX

Chapter 1

Introduction and Software Requirements Specification(SRS)

1.1 About

The "Online Food Ordering System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and,in some cases, reduce the hardships faced by this existing system.

Moreover, this system is designed for the particular need of the cafe /restaurant to carry out operations in a smooth and effective manner.

This system is designed for a particular restaurant or cafe in order to help them to expand their business and maintain their records online.

It's a personalized app for a particular restaurant listing all their products. Just like dominos or burger king it is a personalized food ordering system.It provides a user friendly interface for customers to order food from their favorite restaurant with convenience and comfort of their home without any distraction.

1.2 Problem Statement

Designing a general Food ordering website which can be customized according to the client(restaurant owner) needs.

1.3 Feasibility Study

● Economically Feasibility: The system being developed is economic with respect to a restaurant or food cafes. It is cost effective in the sense that it has eliminated the paper work completely. The system is also time effective because of its automated billing system..

. ● Technical feasibility: The technical requirement for the system is economic and it does not use any other additional Hardware and software.

● Behavioral Feasibility: The system working is quite easy to use and learn due to its simple but attractive interface. Users require no special training for operating the system.

1.4 Advantages

● Cross Platform compatibility: You can run a web application whether you are using Windows, Linux or MacOS or any of the different web browsers.

● It is more manageable as these systems need only be installed on the server placing minimal requirements on the end user workstation.

● It is highly deployable as you only simply need to send the user a website address to log in and provide them with internet access.

● Web based applications provide an added layer of security by removing access to the data and backend servers.

● Costs are dramatically lowered due to reduced support and maintenance, lower requirements on the end user system and simplified architecture.

1.5 Disadvantages

● It has a heavy reliance on the Internet. Even though we live in an Internet era, losing the Internet is still common. And no internet would directly result in losing the ability to run our website.

● A website is completely based on its web browser. Even though this is a good thing, it also has its limiting factors. If the website happens to fail or go unresponsive, then we cannot run it.

● A website operates at a slower speed than the application hosted on a local server. Also a website runs completely on internet, it frequently can feel slower

●How a website responds to a particular device also varies as in case of a desktop, android phone or a tablet.Basically which depends on responsiveness of a website.

1.6 Functional requirements:-

1. User registration and login functionality to allow users to create an account and sign in to the website.
2. A menu page that displays the restaurant's offerings and prices.
3. A Shopping cart feature that allow users to add items to cart and modify their quantity.
4. A payment gateway that accepts online payment.
5. A delivery tracking system for estimated time of arrival of food.
6. A customer rating and support system.

1.7 Non-Functional Requirements:-

1. User interface design that is user-friendly and easy to navigate.
2. Fast loading and screen responsiveness for mobile screen.
3. High level security measure to protect user data.
4. Reliable uptime and availability to ensure website is always accessible.

5. Compliance with local laws and regulations related to food delivery services such as food safety and health regulations.

1.8 Software Process Model

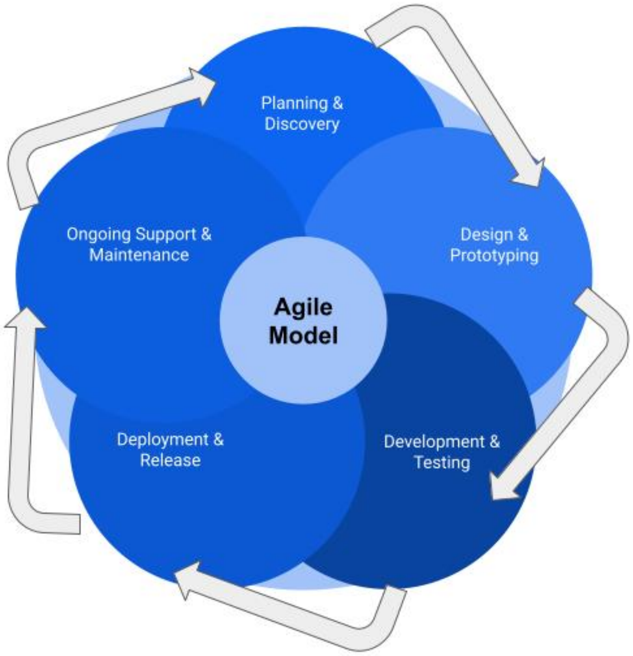
Agile Model:

The Agile model is a popular software development methodology that emphasizes collaboration, adaptability, and flexibility. It involves developing the project in iterative cycles or sprints, where requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams. This model can be ideal for a food delivery website as it allows for quick adaptability and continuous improvement based on customer feedback.

The Agile approach is based on the Agile Manifesto, which values individuals and interactions, working software, customer collaboration, and responding to change. It is designed to deliver high-quality software quickly and efficiently, while also promoting teamwork and communication.

Phases of Agile Model:

Following are the phases in the Agile model are as follows:



1.8.1. Phase 1: Planning and Discovery(Brainstorming)

During this phase, the team will identify project goals, create user stories, and define the overall product vision. The team will work with stakeholders to identify the features and functionalities that the website will need to have, and prioritize them based on business value and user needs.

1.8.2. Phase 2: Design and Prototyping

During this phase, the team will create wireframes, mockups, and prototypes to demonstrate the user interface and user experience of the website. The team will also work with stakeholders to review and iterate on the design until it meets user needs and business goals.

1.8.3. Phase 3: Development and Testing

During this phase, the team will begin developing the website using an iterative approach. The team will work on small, manageable chunks of functionality, testing each one thoroughly before moving on to the next. This approach allows for quick iteration and feedback from stakeholders, and helps to ensure that the website is functioning as intended.

1.8.4. Phase 4: Deployment and Release

During this phase, the team will deploy the website to a production environment, and conduct final testing and quality assurance checks to ensure that the website is ready for release. The team will also work with stakeholders to develop a release plan, and to ensure that all necessary documentation and support materials are in place.

1.8.5. Phase 5: Ongoing Support and Maintenance

After the website is released, the team will continue to provide ongoing support and maintenance. This may include fixing bugs, addressing user feedback, and adding new features and functionality over time.

Advantages of Agile Model

1. Frequent Delivery
2. Face-to-Face Communication with clients.
3. Efficient design and fulfills the business requirement.
4. Anytime changes are acceptable.
5. It reduces total development time.

Disadvantages of Agile Model

Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be interpreted at any time by different team members.

Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become a difficulty.

1.9 Tools Used:

- HTML
- CSS
- PHP
- JAVASCRIPT
- XAMPP
- MYSQL
- BOOTSTRAP
- VISUAL STUDIO CODE
- GOOGLE SLIDES

Chapter 2

Estimation and Scheduling

2.1 Estimation

Estimation of the size of the software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:-
Lines of Code
Number of entities in ER diagram
Total number of processes in detailed data flow diagram
Function points

2.1.1 Lines of Code (LOC): As the name suggests, LOC counts the total number of lines of source code in a project. The units of LOC are:

KLOC- Thousand lines of code
NLOC- Non-comment lines of code
KDSI- Thousands of delivered source instruction

The size is estimated by comparing it with the existing systems of the same kind. The experts use it to predict the required size of various components of software and then add them to get the total size.

It’s tough to estimate LOC by analyzing the problem definition. Only after the whole code has been developed can accurate LOC be estimated. This statistic is of little utility to project managers because project planning must be completed before development activity can begin.

Two separate source files having a similar number of lines may not require the same effort. A file with complicated logic would take longer to create than one with simple logic. Proper estimation may not be attainable based on LOC.

The length of time it takes to solve an issue is measured in LOC. This statistic will differ greatly from one programmer to the next.

Advantages:

Universally accepted and is used in many models like COCOMO.
Estimation is closer to the developer’s perspective.Simple to use.

Disadvantages:

Different programming languages contain a different number of lines.No proper industry standard exists for this technique. It is difficult to estimate the size using this technique in the early stages of the project.

2.1.2 Number of entities in ER diagram: ER model provides a static view of the project. It describes the entities and their relationships. The number of entities in ER model can be used to measure the estimation of the size of the project. The number of entities depends on the size of the project. This is because more entities needed more classes/structures thus leading to more coding.

Advantages:

Size estimation can be done during the initial stages of planning.

The number of entities is independent of the programming technologies used.

Disadvantages:

No fixed standards exist. Some entities contribute more project size than others.

Just like FPA, it is less used in the cost estimation model. Hence,
it must be converted to LOC.

2.1.3. Total number of processes in detailed data flow diagram:

Data Flow Diagram(DFD) represents the functional view of software. The model depicts the main processes/functions involved in software and the flow of data between them.Utilization of the number of functions in DFD to predict software size. Already existing processes of similar type are studied and used to estimate the size of the process. Sum of the estimated size of each process gives the final estimated size.

Advantages:

It is independent of the programming language. Each major process can be decomposed into smaller processes.This will increase the accuracy of estimation.

Disadvantages:

Studying similar kinds of processes to estimate size takes additional time and effort. All software projects are not required for the construction of DFD.

2.1.4. Function Point Analysis: In this method, the number and type of functions supported by the software are utilized to find FPC(function point count). The steps in function point analysis are:

Count the number of functions of each proposed type.

Compute the Unadjusted Function Points(UFP).

Find Total Degree of Influence(TDI).

Compute Value Adjustment Factor(VAF).

Find the Function Point Count(FPC).

The explanation of the above points is given below:

Count the number of functions of each proposed type: Find the number of functions belonging to the following types:

External Inputs: Functions related to data entering the system.
External outputs: Functions related to data exiting the system.
External Inquiries: They lead to data retrieval from the system but don’t change the system.
Internal Files: Logical files maintained within the system. Log files are not included here.
External interface Files: These are logical files for other applications which are used by our system.

Compute the Unadjusted Function Points(UFP): Categorize each of the five function types like simple, average, or complex based on their complexity. Multiply the count of each function type with its weighting factor and find the weighted sum. The weighting factors for each type based on their complexity are follows:

Function type	Simple	Average	Complex	count
External Inputs	3	4	6	5
External Output	4	5	7	5
External Inquiries	3	4	6	3
Internal Logical Files	7	10	15	6
External Interface Files	5	7	10	0

Count total for avg case complexity = 4*5 + 5*5 + 4*3 +10*6 +7*0 = 117

fi	result
1. Does the system require reliable backup and recovery?	4
2. Are specialized data communications required to transfer information to or from the application?	3
3. Are there distributed processing functions?	2
4 .Is performance critical?	2
5. Will the system run in an existing, heavily Utilized operational environments?	3
6. Does the system require online data entry?	5
7. Does the online data entry require the input transaction to be built over multiple screens or operations?	2
8. Are the ILFs updated online?	4
9. Are the inputs, outputs, files, or inquiries complex?	3
10. Is the internal processing complex?	3
11. Is code design to be reusable?	4

12. Are conversions and installation included in the design?	3
13 Is the system designed for multiple installations in different organizations?	5
14 Is the application designed to facilitate change and ease of use by the user?	5

$\sum \boldsymbol{fi} = 48$

FP = count total*[0.65 +0.01*$\sum \boldsymbol{fi}$]
= 117*[0.65 + 0.48]
= 117*1.13
= 132.21

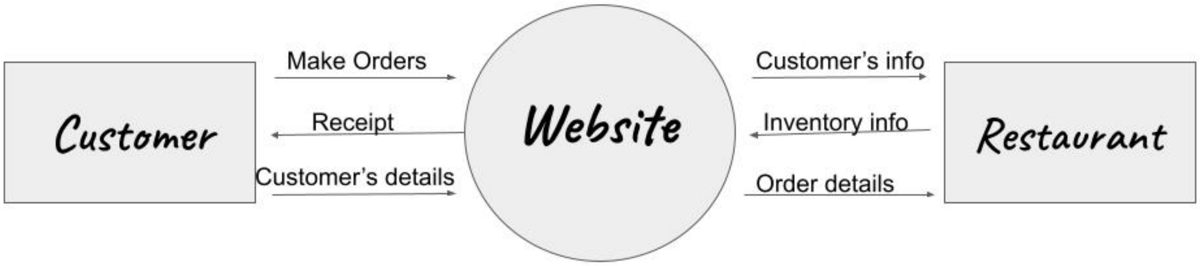
2.2 Scheduling:-

Gantt chart

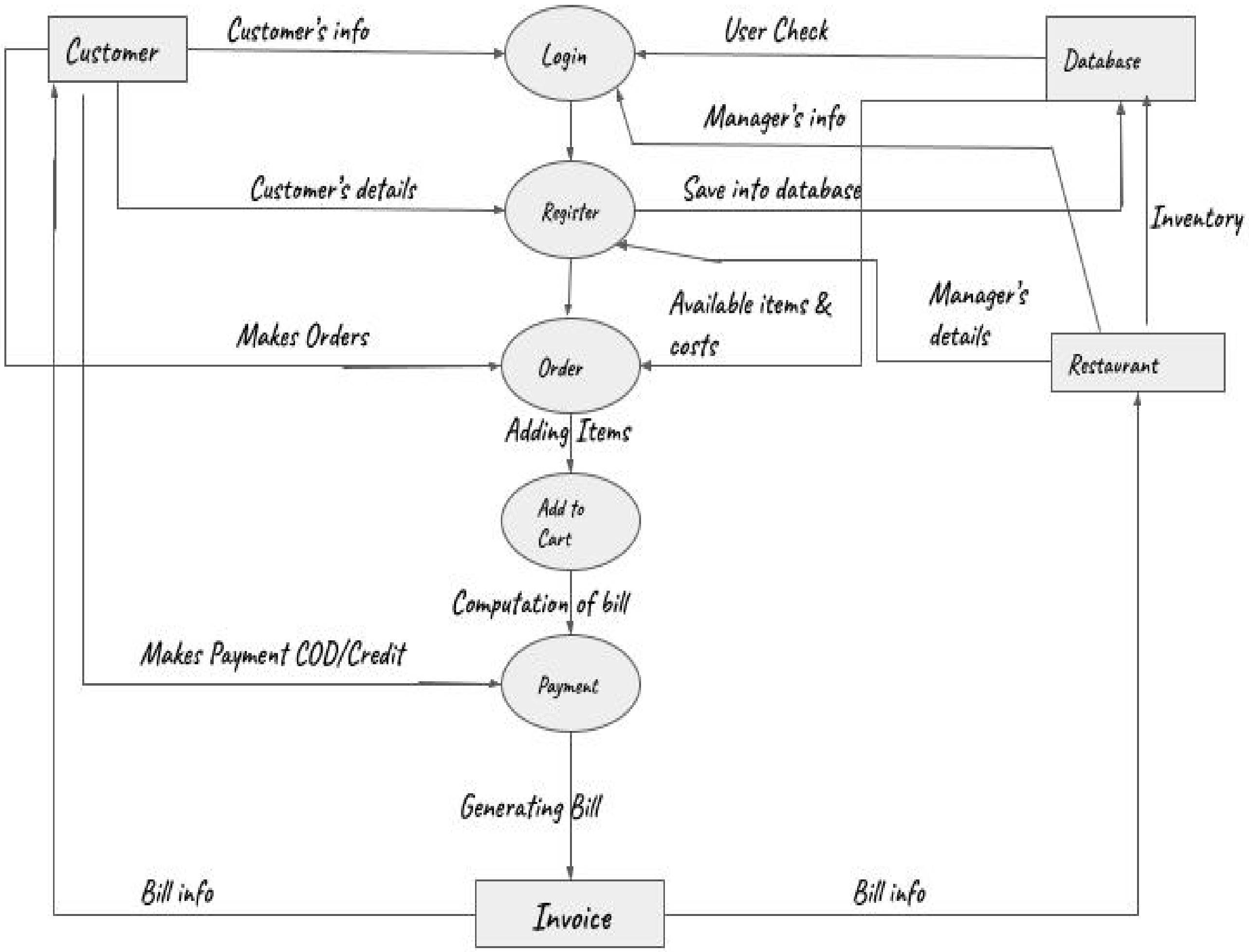
Steps	**End Date**
SRS	17th March
Cost & Size Estimation	24th March
Design	3rd April
Implementation	13th April
Testing	20th April
User Manual	22th April
Deployment	24th April

Chapter 3
ARCHITECTURE and DFD

DATA FLOW DIAGRAM



Level-0


Customer
Customer's info
Login
User Check
Database
Manager's info
Customer's details
Register
Save into database
Inventory
Available items & costs
Manager's details
Makes Orders
Order
Restaurant
Adding Items
Add to Cart
Computation of bill
Makes Payment COD/Credit
Payment
Generating Bill
Bill info
Invoice
Bill info

Level 1 DFD

Data Model

A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated. We are Using Relational Data Model in our software project.

Relational Data Model-: The relational data model is a conceptual framework used to organize and structure data in a database. It is based on the concept of a relation, which can be thought of as a table with rows and columns. Each row in a relation represents a unique entity or record, and each column represents a specific attribute or characteristic of that entity.

The key components of the relational data model are as follows:

Relation/Table: A relation is a two-dimensional table that contains rows and columns. Each row represents a single entity or record, and each column represents an attribute or property of that entity.

Tuple/Row: A tuple refers to a single row in a relation. It contains a set of values that correspond to the attributes defined for the relation.

Attribute/Column: An attribute is a named property or characteristic of the entities in a relation. It defines the type of data that can be stored in that column.

Domain: A domain defines the set of possible values for an attribute. It specifies the data type and any constraints or restrictions on the attribute values.

Primary Key: A primary key is an attribute or combination of attributes that uniquely identify each row in a relation. It ensures the uniqueness and integrity of the data in the table.

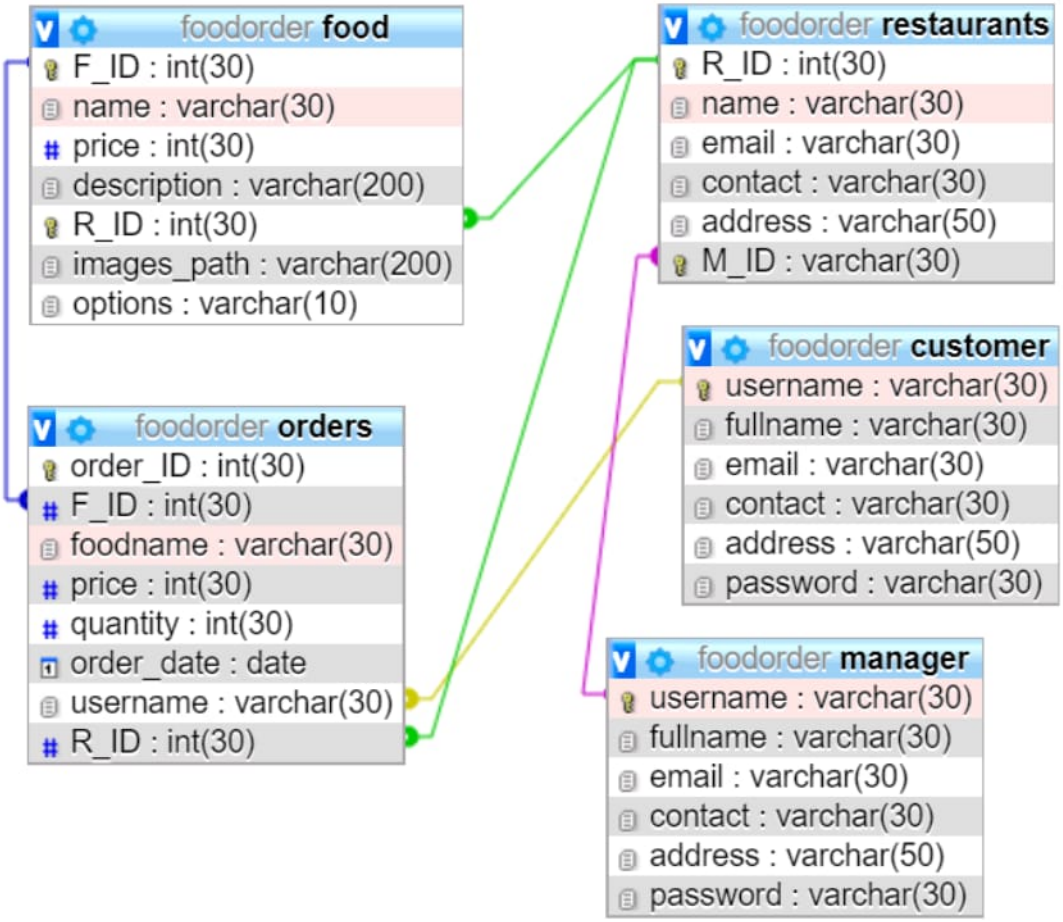
Foreign Key: A foreign key is an attribute in one relation that refers to the primary key of another relation. It establishes relationships between tables and enforces referential integrity.

Relational Operations: The relational data model provides a set of operations that can be performed on relations, such as select, project, join, and union, to retrieve, manipulate, and combine data in meaningful ways.

The relational data model offers several advantages, including simplicity, flexibility, and data integrity. It has been widely adopted and is the foundation of many popular database management systems (DBMS) like MySQL, Oracle, and Microsoft SQL Server.

Some of the tables used in our database are-:

Schema Diagram



ER Diagram

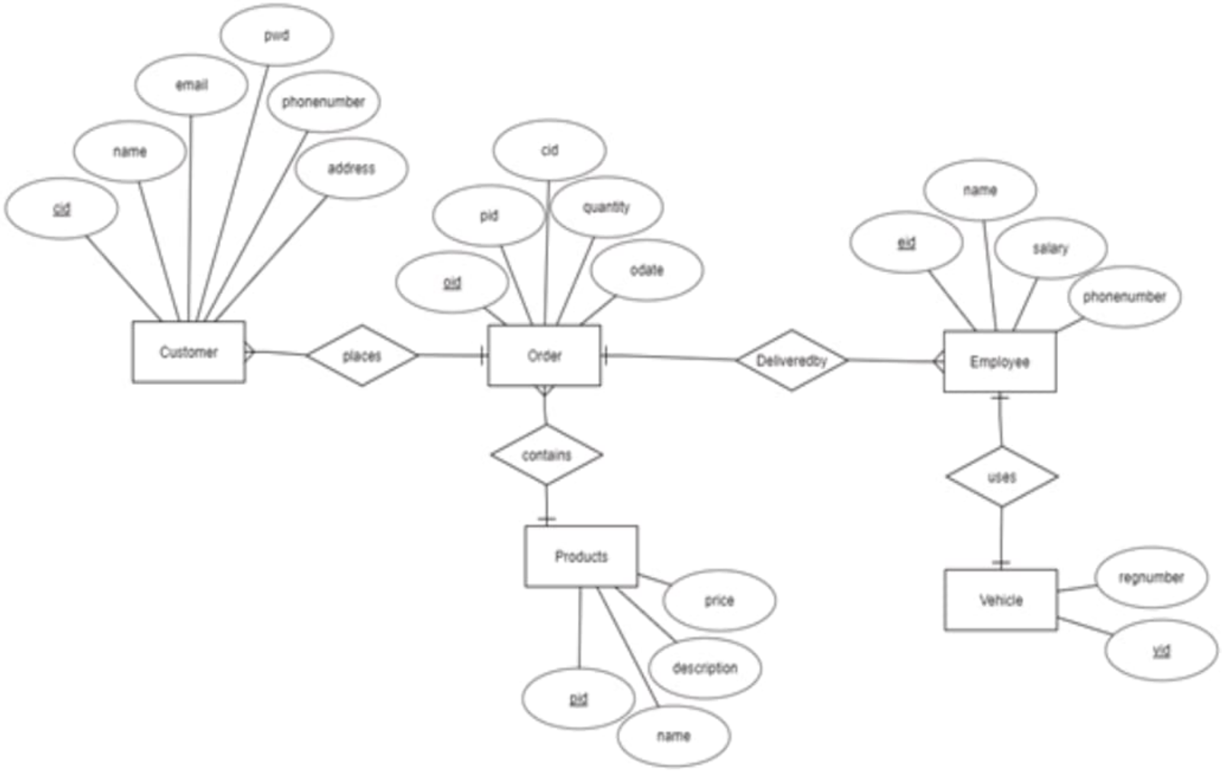


Figure:ER diagram

Chapter 4
Implementation

This activity includes programming, testing and integration of modules into a progressively more complete system. Implementation is the process of collect all the required parts and assembles them into a major product.

Login_u.php

```
<?php
session_start();
$error='';

if (isset($_POST['submit'])) {
if (empty($_POST['username']) || empty($_POST['password'])) {
$error = "Username or Password is invalid";
}
else
{
// Define $username and $password
$username=$_POST['username'];
$password=$_POST['password'];
// Establishing Connection with Server by passing server_name, user_id and password as a parameter
require 'connection.php';
$conn = Connect();

// SQL query to fetch information of registerd users and finds user match.
$query = "SELECT username, password FROM CUSTOMER WHERE username=? AND password=? LIMIT 1";

// To protect MySQL injection for Security purpose
$stmt = $conn->prepare($query);
$stmt -> bind_param("ss", $username, $password);
$stmt -> execute();
$stmt -> bind_result($username, $password);
$stmt -> store_result();
if ($stmt->fetch())
{
            $_SESSION['login_user2']=$username; // Initializing Session
            header("location: foodlist.php"); // Redirecting To Other Page
} else {
$error = "Username or Password is invalid";
}
mysqli_close($conn); // Closing Connection
}
}
?>
```

CustomerLogin.php

Below is a part of customerlogin.php

```
<script type="text/javascript">
      window.onscroll = function()
      {
        scrollFunction()
      };

      function scrollFunction(){
        if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
          document.getElementById("myBtn").style.display = "block";
        } else {
          document.getElementById("myBtn").style.display = "none";
        }
      }

      function topFunction() {
        document.body.scrollTop = 0;
        document.documentElement.scrollTop = 0;
      }
    </script>
```

Contact Us.php

Below is a form Used to Contact Us.

```
<form method="post" action="">
        <br style="clear: both">
        <h3 style="margin-bottom: 25px; text-align: center; font-size: 30px;"> Contact Form</h3>

        <div class="form-group">
          <input type="text" class="form-control" id="name" name="name" placeholder="Name"
required autofocus="">
        </div>

        <div class="form-group">
          <input type="email" class="form-control" id="email" name="email" placeholder="Email"
required>
        </div>

        <div class="form-group">
          <input type="Number" class="form-control" id="mobile" name="mobile"
placeholder="Mobile Number" required>
        </div>

        <div class="form-group">
          <input type="text" class="form-control" id="subject" name="subject" placeholder="Subject"
required>
        </div>

        <div class="form-group">
         <textarea class="form-control" type="textarea" id="message" name="message"
placeholder="Message" maxlength="140" rows="7"></textarea>
         <span class="help-block"><p id="characterLeft" class="help-block">Max Character length :
140 </p></span>
        </div>
        <input type="submit" name="submit" type="button" id="submit" name="submit" class="btn
btn-primary pull-right"/>
       </form>


     </div>
   </div>

   </div>

   <?php
if (isset($_POST['submit'])){
require 'connection.php';
$conn = Connect();
$Name = $conn->real_escape_string($_POST['name']);
$Email_Id = $conn->real_escape_string($_POST['email']);
$Mobile_No = $conn->real_escape_string($_POST['mobile']);
$Subject = $conn->real_escape_string($_POST['subject']);
$Message = $conn->real_escape_string($_POST['message']);

$query = "INSERT into contact(Name,Email,Mobile,Subject,Message)
VALUES('$Name','$Email_Id','$Mobile_No','$Subject','$Message')";
$success = $conn->query($query);

if (!$success){
  die("Couldn't enter data: ".$conn->error);
}

$conn->close();
}
?>
```

Logout_m.php

```
<?php
session_start();
if(session_destroy()) // Destroying All Sessions
{
header("Location: managerlogin.php"); // Redirecting To Home Page
}
?>
```

Connection.php

```
<?php

function Connect()
{
	$dbhost = "localhost";
	$dbuser = "root";
	$dbpass = "";
	$dbname = "foodorder";
	//Create Connection
	$conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname) or die($conn->connect_error);

	return $conn;
}
?>
```

Chapter 5
Software Testing

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.
There are four main types of software testing:-

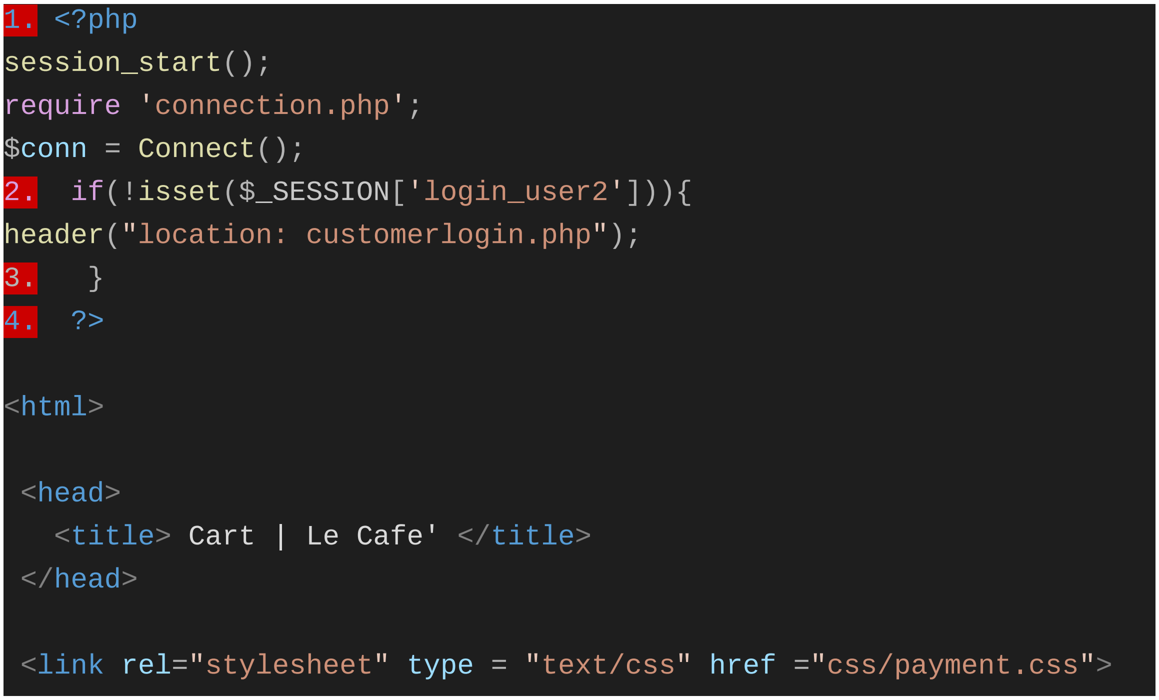
1. **Unit testing:-**Testing individual components or modules of the software to ensure they function as intended.

2. **Integration testing:-**Testing the interactions between different components or modules to ensure they work together as intended.
3. **System testing:-** Testing the entire system as a whole to ensure it meets the requirements and specifications.

4. **Regression Testing:-** Testing to ensure that changes or updates to the software have not introduced new defects or issues.

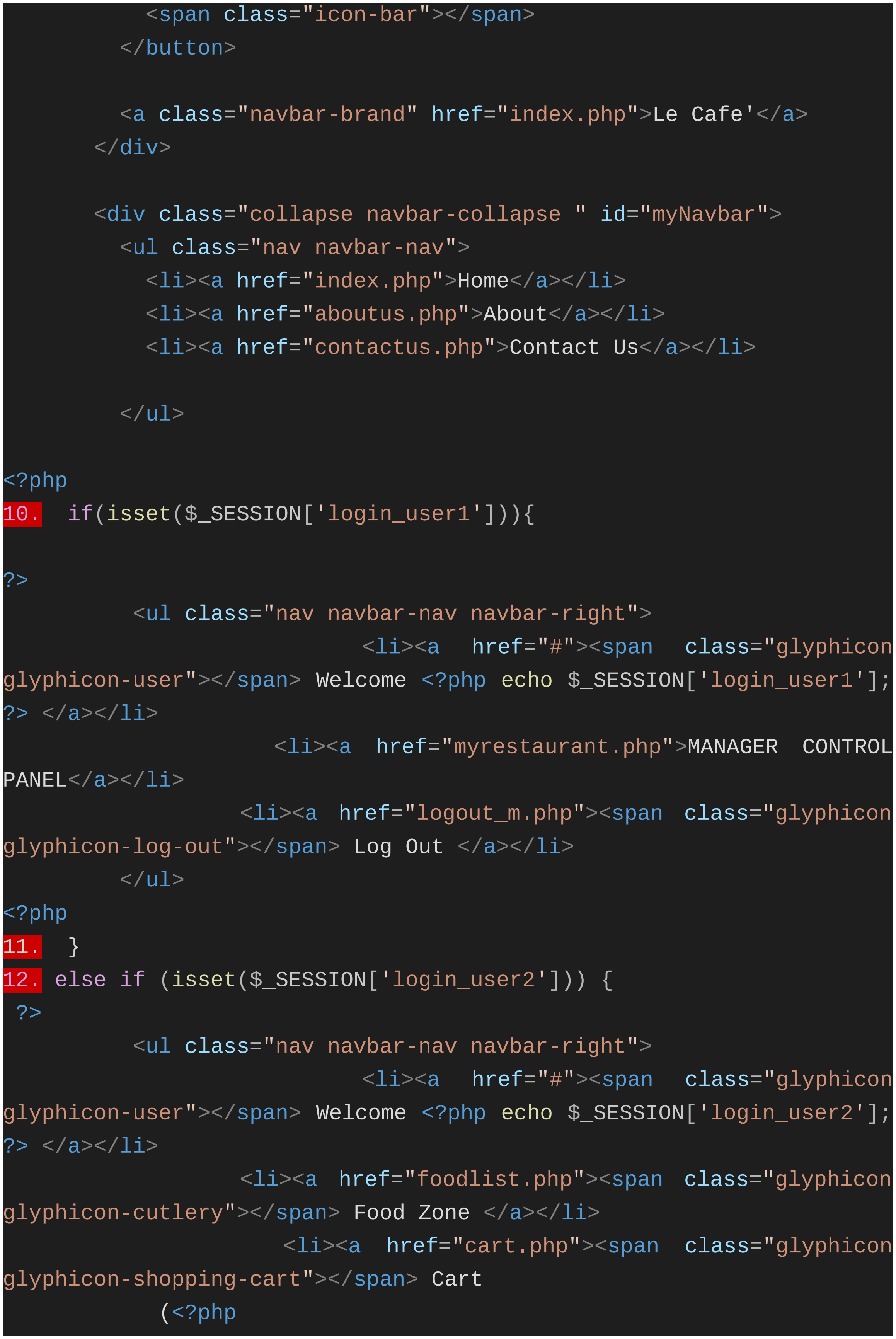
First we are going to discuss whitebox testing which is also a part of unit testing.

White Box Testing:- It focuses on testing the internal structure and design of the software. It involves testing the software code, architecture, and other internal components to ensure they are functioning as intended and to identify any defects or issues that may arise.
In white box testing we calculate the cyclomatic complexity of a code using control flow diagram. So here is a piece of php code of a payment module for which we are calculating the cyclomatic complexity:

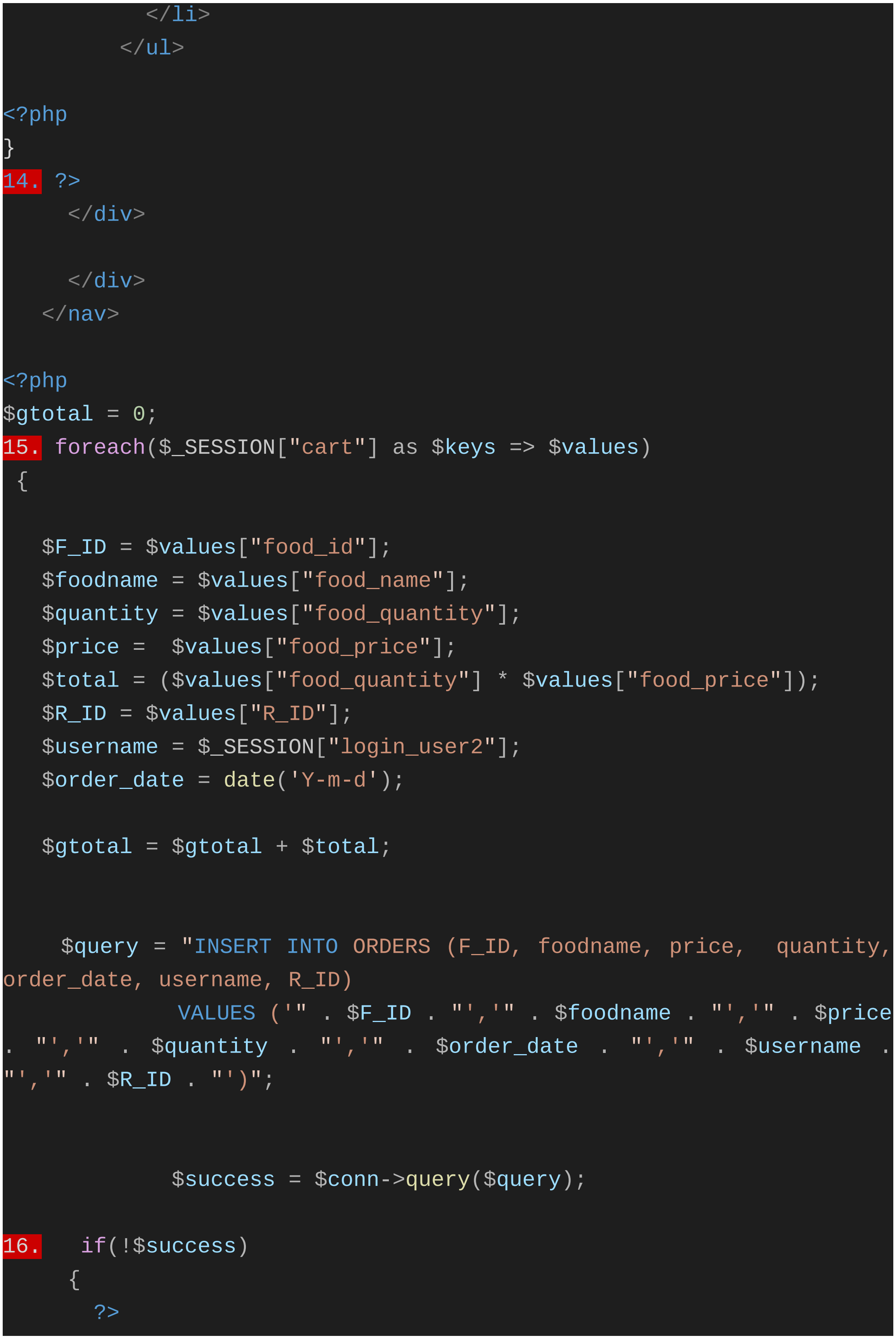


```
1. <?php
session_start();
require 'connection.php';
$conn = Connect();
2.  if(!isset($_SESSION['login_user2'])){
header("location: customerlogin.php");
3.    }
4.  ?>

<html>

 <head>
   <title> Cart | Le Cafe' </title>
 </head>

 <link rel="stylesheet" type = "text/css" href ="css/payment.css">
```

```
    <link     rel="stylesheet"     type     =     "text/css"     href
="css/bootstrap.min.css">
 <script type="text/javascript" src="js/jquery.min.js"></script>
 <script type="text/javascript" src="js/bootstrap.min.js"></script>

 <body>
   <button onclick="topFunction()" id="myBtn" title="Go to top">
     <span class="glyphicon glyphicon-chevron-up"></span>
   </button>
    <script type="text/javascript">
     window.onscroll = function()
     {
       scrollFunction()
     };

5.     function scrollFunction(){
6.                 if   (document.body.scrollTop   >   20   ||
document.documentElement.scrollTop > 20) {
         document.getElementById("myBtn").style.display = "block";
7.       }
8.   else {
         document.getElementById("myBtn").style.display = "none";
       }
9.     }

     function topFunction() {
       document.body.scrollTop = 0;
       document.documentElement.scrollTop = 0;
     }
   </script>

           <nav    class="navbar    navbar-inverse    navbar-fixed-top
navigation-clean-search" role="navigation">
     <div class="container">
       <div class="navbar-header">
              <button  type="button"  class="navbar-toggle  collapsed"
data-toggle="collapse" data-target="#myNavbar">
           <span class="sr-only">Toggle navigation</span>
           <span class="icon-bar"></span>
           <span class="icon-bar"></span>
```

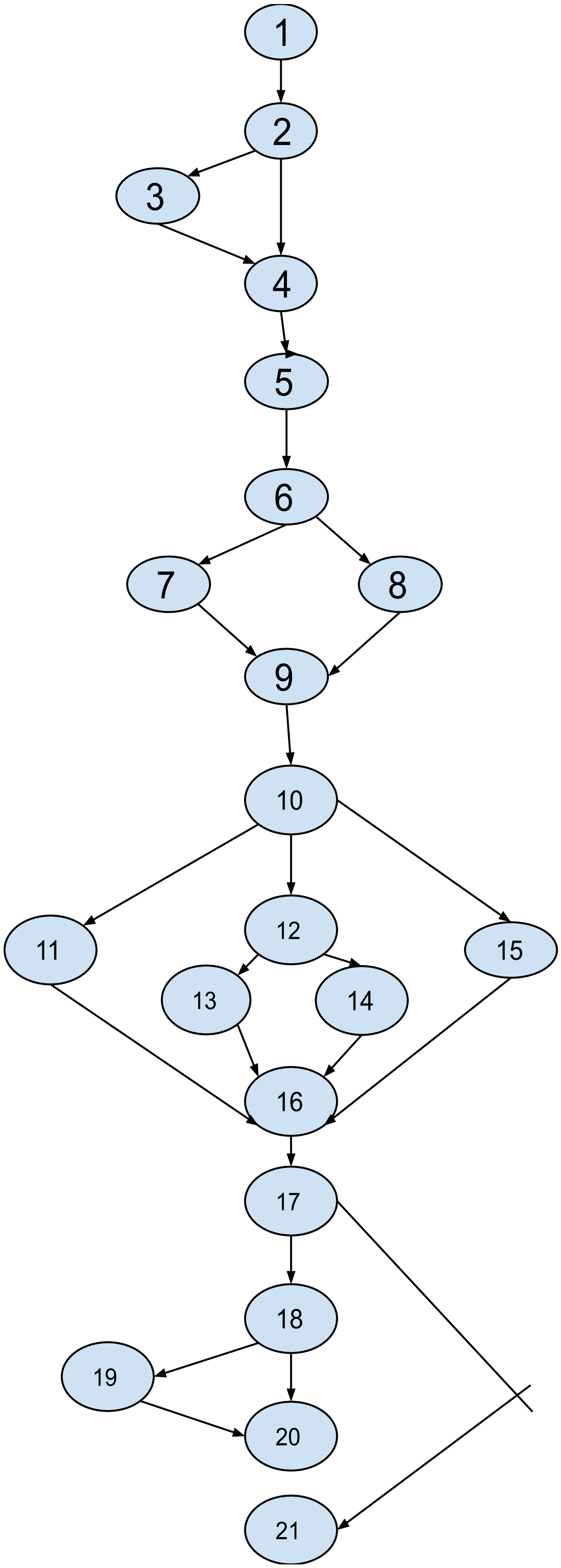


```
           <span class="icon-bar"></span>
         </button>

         <a class="navbar-brand" href="index.php">Le Cafe'</a>
       </div>

       <div class="collapse navbar-collapse " id="myNavbar">
         <ul class="nav navbar-nav">
           <li><a href="index.php">Home</a></li>
           <li><a href="aboutus.php">About</a></li>
           <li><a href="contactus.php">Contact Us</a></li>

         </ul>

<?php
10.  if(isset($_SESSION['login_user1'])){

?>
          <ul class="nav navbar-nav navbar-right">
                            <li><a   href="#"><span   class="glyphicon
glyphicon-user"></span> Welcome <?php echo $_SESSION['login_user1'];
?> </a></li>
                     <li><a  href="myrestaurant.php">MANAGER  CONTROL
PANEL</a></li>
                  <li><a  href="logout_m.php"><span  class="glyphicon
glyphicon-log-out"></span> Log Out </a></li>
         </ul>
<?php
11.  }
12. else if (isset($_SESSION['login_user2'])) {
 ?>
          <ul class="nav navbar-nav navbar-right">
                            <li><a   href="#"><span   class="glyphicon
glyphicon-user"></span> Welcome <?php echo $_SESSION['login_user2'];
?> </a></li>
                  <li><a  href="foodlist.php"><span  class="glyphicon
glyphicon-cutlery"></span> Food Zone </a></li>
                      <li><a  href="cart.php"><span  class="glyphicon
glyphicon-shopping-cart"></span> Cart
            (<?php
             if(isset($_SESSION["cart"])){
             $count = count($_SESSION["cart"]);
             echo "$count";
           }
             else
               echo "0";
             ?>)
             </a></li>
                  <li><a href="logout_u.php"><span class="glyphicon
glyphicon-log-out"></span> Log Out </a></li>
         </ul>
 <?php
}
13. else {

 ?>

<ul class="nav navbar-nav navbar-right">
                    <li><a href="#" class="dropdown-toggle active"
data-toggle="dropdown" role="button" aria-haspopup="true"
aria-expanded="false"><span class="glyphicon glyphicon-user"></span>
Sign Up <span class="caret"></span> </a>
               <ul class="dropdown-menu">
                           <li> <a href="customersignup.php"> User
Sign-up</a></li>
                         <li> <a href="managersignup.php"> Manager
Sign-up</a></li>
             <li> <a href="#"> Admin Sign-up</a></li>
           </ul>
           </li>

                    <li><a href="#" class="dropdown-toggle active"
data-toggle="dropdown" role="button" aria-haspopup="true"
aria-expanded="false"><span class="glyphicon
glyphicon-log-in"></span> Login <span class="caret"></span></a>
             <ul class="dropdown-menu">
             <li> <a href="customerlogin.php"> User Login</a></li>
             <li> <a href="managerlogin.php"> Manager Login</a></li>
             <li> <a href="#"> Admin Login</a></li>
           </ul>
```



```
           </li>
         </ul>

<?php
}
14. ?>
     </div>

     </div>
   </nav>

<?php
$gtotal = 0;
15. foreach($_SESSION["cart"] as $keys => $values)
 {

   $F_ID = $values["food_id"];
   $foodname = $values["food_name"];
   $quantity = $values["food_quantity"];
   $price =  $values["food_price"];
   $total = ($values["food_quantity"] * $values["food_price"]);
   $R_ID = $values["R_ID"];
   $username = $_SESSION["login_user2"];
   $order_date = date('Y-m-d');

   $gtotal = $gtotal + $total;


    $query = "INSERT INTO ORDERS (F_ID, foodname, price,  quantity,
order_date, username, R_ID)
             VALUES ('" . $F_ID . "','" . $foodname . "','" . $price
. "','" . $quantity . "','" . $order_date . "','" . $username .
"','" . $R_ID . "')";


             $success = $conn->query($query);

16.   if(!$success)
     {
       ?>
```

```
      <div class="container">
         <div class="jumbotron">
           <h1>Something went wrong!</h1>
           <p>Try again later.</p>
         </div>
       </div>

       <?php
17.     }

18. }

19.
       ?>
       <div class="container">
         <div class="jumbotron">
           <h1>Choose your payment option</h1>
         </div>
       </div>
       <br>
<h1 class="text-center">Grand Total: &#8377;<?php echo "$gtotal";
?>/-</h1>
<h5 class="text-center">including all service charges. (no delivery
charges applied)</h5>
<br>
<h1 class="text-center">
    <a href="cart.php"><button class="btn btn-warning"><span
class="glyphicon glyphicon-circle-arrow-left"></span> Go back to
cart</button></a>
   <a href="onlinepay.php"><button class="btn btn-success"><span
class="glyphicon glyphicon-credit-card"></span> Pay
Online</button></a>
    <a href="COD.php"><button class="btn btn-success"><span
class="glyphicon glyphicon-"></span> Cash On Delivery</button></a>
</h1>



<br><br><br><br><br><br>
       </body></html>
```



Control Flow Diagram

As cyclomatic complexity (CC) can be calculated using 3 methods:

1. By the number of edges and nodes in CFD.
   CC = E-N+2 = 24-19+2 = 7

2. By counting non overlapping bounded areas:
   CC = no. of non overlapping area +1
   = 6+1 = 7
3. By Counting no. of decision and loop statement:
   CC = no. of decision statement +1
   = 6+ 1 = 7

Hence the cyclomatic complexity of the above module is 7.

Use of Cyclomatic Complexity:

● Determining the independent path executions thus proven to be very helpful for Developers and Testers.
● It can make sure that every path has been tested at least once.
● Thus help to focus more on uncovered paths.
● Code coverage can be improved.
● Risk associated with the program can be evaluated.33
● These metrics being used earlier in the program helps in reducing the risks.

Advantages of Cyclomatic Complexity:.

● It can be used as a quality metric, giving relative complexity of various designs.
● It is able to compute faster than Halstead's metrics.

- It is used to measure the minimum effort and best areas of concentration for testing.
- It is able to guide the testing process.
- It is easy to apply .

Disadvantages of Cyclomatic Complexity:

- It is the measure of the program's control complexity and not the data complexity.
- In this, nested conditional structures are harder to understand than non-nested structures.
- In case of simple comparisons and decision structures, it may give a misleading figure.

Chapter 6
User Manual

6.1 Introduction

The Food ordering system application provides users a simple interface to order food from their favorite restaurant in just a click from the device.This document provides instructions for using the application.

6.2 Getting Started

Go to google and search for our web application.

6.3 Quick Start

Step 1: Open the website on your device.

Step 2: The login screen will appear ,if you already have a account then login using your registered mobile number, otherwise sign up for a new account. Signup for a new account using your mobile number.

Step 4: After you have successfully logged in,Your profile will be opened.

Step 5: Select your favorite food and order it by making payment online.

Step 6: Monitor your delivery and enjoy your food.

6.4 Troubleshooting

Missing or Incorrect Password or EMail. A message will be displayed in the event incorrect login information is entered. Try again with proper credentials to access your account or signup using a different phone number.

If your payment is done by you and it is not showing on the payment screen then make the payment again and if we get the money twice it will be automatically refunded within 5 days.

Chapter 7
Conclusion

The entire project has been developed and deployed as per the requirements stated by the user. It is found to be bug free as per the testing standards that are implemented.This project fulfill the requirements of owner and helps them to expand their business in the way he wants.This project provides an offer to the customer to login through the login portal using the login id and password.It is very helpful for the customers to look for the food only from their favorite restaurant.Food can be ordered directly from the app and the live monitoring of delivery is also shown to customer.Finally, we like to conclude that we put all our efforts throughout the development of our project and tried to fulfill most of the requirements of the user

Chapter 8
Bibliography

1. Elmasri, R., & Navathe, S.B. (2015). Fundamentals of Database Systems. 7th edition. Pearson Education pdf.
2. IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
3. The principle source of text book material is “Software Engineering A PRACTITIONER'S APPROACH” by Roger S. Pressman.
4. The principle source of text book material is “A Concise Introduction to Software Engineering” by Pankaj Jalote.
5.Javatpoint(https://www.javatpoint.com/software-engineering)
6. Youtube