

ONLINE FOOD DELIVERY SYSTEM

A PROJECT REPORT

for

Project (KCA451)

Session (2023-24)

Submitted by

CHELSEY SINGH

(2200290140050)

**Submitted in fulfillment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
M/S SHRUTI AGGARWAL
Teaching Assistant**



Submitted to

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(MAY 2024)**

CERTIFICATE

Certified that **CHELSEY SINGH <2200290140050>** has/ have carried out the project work having “**ONLINE FOOD DELIVERY SYSTEM**” (**Major Project-KCA451**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

CHELSEY SINGH (2200290140050)

.....

.....

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

M/S SHRUTI AGGARWAL

Teaching assistant

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi

Head

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

ABSTRACT

The Online Food Ordering System described in this document has been designed to fill a specific niche in the market by providing small restaurants with the ability to offer their customers an online ordering option without having to invest large amounts of time and money in having custom software designed specifically for them. The system, which is highly customizable, allows the restaurant employees to easily manage the site content, most importantly the menu, themselves through a very intuitive graphical interface.

The website, which is the only component seen by the restaurant customers, is then built dynamically based on the current state of the system, so any changes made are reflected in real time. Visitors to the site, once registered, are then able to easily navigate this menu, add food items to their order, and specify delivery options with only a few clicks, greatly simplifying the ordering process. Back in the restaurant, placed orders are promptly retrieved and displayed in an easily readable format for efficient processing.

The purpose of this document is to provide in-depth descriptions of design and implementation details of the system, as well as descriptions of all available functionality and plans for evolution. In addition, user manuals and troubleshooting tips have been included for all three components to give the reader a clear idea of intended typical use cases for the system.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **M/S SHRUTI AGGARWAL** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Table of Contents

Chapter 1: Justification & Feasibility	Error! Bookmark not defined.
Justification	Error! Bookmark not defined.
Feasibility	2
Chapter 2: Requirements Specification.....	9
System Model.....	9
Functional Requirements	9
The Web Ordering System.....	9
System model DFD.....	10
Menu Management System.....	9
Order Retrieval System.....	10
User Interface Specifications	10
Web Ordering System.....	10
Menu Management System.....	11
Order Retrieval System.....	11
Non-functional Requirements	11
System Evolution.....	12
Chapter 3: System Design.....	13
System Design.....	13
Level 1: The Database & the 3 Components.....	13
Level 2: Web Ordering System Components	13
Level 2: Menu Management System Components	15
Level 2: Order Retrieval System Components	16
User Interface Design	17

Help System Design	17
Chapter 4: Testing Design.....	18
Testing	18
Phases	18
Requirements Traceability	34
Testing Schedule	35
Recording Procedures	36
coding	36
Chapter 5: User Manual.....	37
Introductory Manual	37
Using the Desktop Application	37
Using the Web Ordering System.....	43
Using the Help System	45
System Reference Manual	46
Services (Alphabetical).....	46
Error Recovery	47
Installation.....	47
Chapter 6: result and discussion	37
Chapter 7: conclusion	52

LIST OF TABLES

Table 2.1. Customer description.....	14	P
Table 2.2. Purchase bill.....	14	

LIST OF FIGURES

Figure 2.1.0 level DFD	11
Figure 2.2. Data model table design	12
Figure 2.3. entity relationship diagram	13
Figure 3.1. Navigation tree	17
Figure 3.2. Summary panel	18
Figure 4.1. Oder retrieval system	24
Figure 4.2. Payment page	26

Chapter 1

Justification & Feasibility

Justification

In today's age of fast food and take-out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until very recently, all of these delivery orders were placed over the phone, but there are many disadvantages to this system. First, the customer must have a physical copy of the restaurant's menu to look at while placing their order and this menu must be up to date. While this expectation is not unreasonable, it is certainly inconvenient.

Second, the orders are placed using strictly oral communication, which makes it far more difficult for the customer to receive immediate feedback on the order they have placed. This often leads to confusion and incorrect orders. The current system is also inconvenient for the restaurant itself, as they must either have a dedicated staff member to answer the phone and take orders, or some employees must perform double-duty, distracting them from their regular tasks.

What I propose is an online ordering system, originally designed for use in college cafeterias, but 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. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order are, in fact, what was intended.

The system also greatly lightens the load on the restaurant's end, as the entire process of taking orders is automated. Once an order is placed

employees to quickly go through the orders as they are placed and produce the necessary items with minimal delay and confusion.

While there are already a few systems like this in existence, all those I have encountered have been designed specifically for one restaurant, and thus cater only to their unique needs. Perhaps the greatest advantage of my system is its flexibility. The web order forms are built dynamically from the database, which can be managed using a graphical user interface. This allows the restaurant employees to not only set up and customize the system on their own, but also allows them to make changes to the menu in real time. For this reason, the exact same system can be used by numerous businesses with absolutely no modification to the code itself, which greatly increases its usefulness.

Feasibility

At the present moment, the system is entirely functional, save the few minor bugs which are bound to present themselves during more extensive testing. A user is currently able to register and log in to the website and place an order. That order is then displayed, correctly and completely, in the order retrieval desktop application. Much of what is left to do focuses not on improving functionality, but rather on improving user experience by creating richer graphical interfaces for the user to interact with and modifying the application's icons and color schemes to make them more pleasing to look at and use. For this reason, I feel that completing the project in the required timeframe is very feasible, particularly if I am able to adhere to the dates outlined in Figure 1, below.

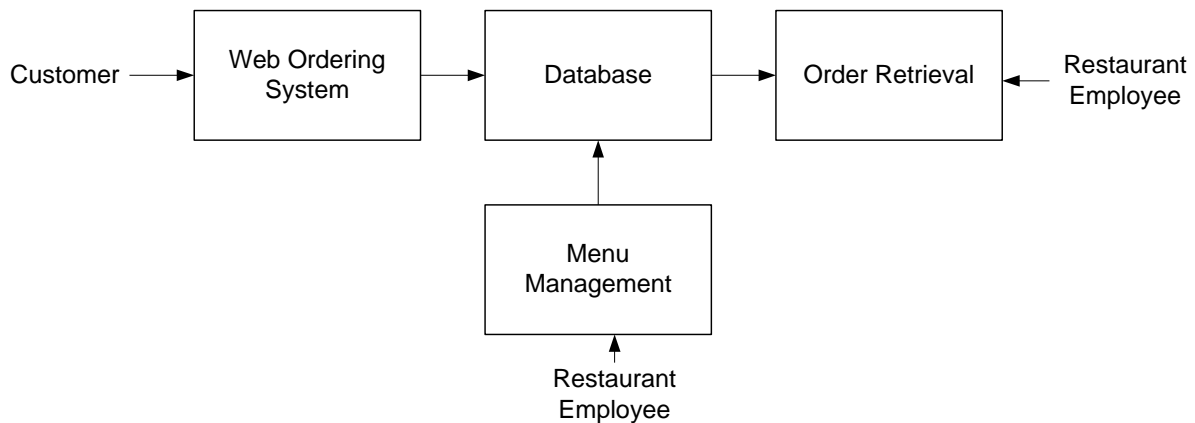
In addition to time, a second factor influencing feasibility is resources, which also should not be a concern here. The online ordering system is structured like a fairly standard web application, and as such requires no special hardware and only basic software, namely web and database servers, to function properly. Therefore, I anticipate finishing all of the required work on time or, ideally, ahead of schedule, leaving me with time to investigate a few additional features I would like to add but are not integral to the system.

Chapter 2:

Requirements Specification

System Model

The structure of the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the restaurant to control what can be ordered by customers. The second component is the web ordering system and provides the functionality for customers to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the restaurant to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.



Functional Requirements

As can be seen in the system model diagrammed above, each of the three system components essentially provides a layer of isolation between the end user and the database. The motivation behind this isolation is twofold. Firstly, allowing the end user to interact with the system through a rich interface provide a much more enjoyable user experience, particularly for the non-technical users which will account for the majority of the system's users. In addition, this isolation layer also protects the integrity of the database by preventing users from taking any action outside those which the system is designed to handle. Because of this design pattern, it is essential to enumerate exactly which functions a user will be presented and these functions are outlined below, grouped by component.

System Model DFD



Figure 2.1. 0 level DFD

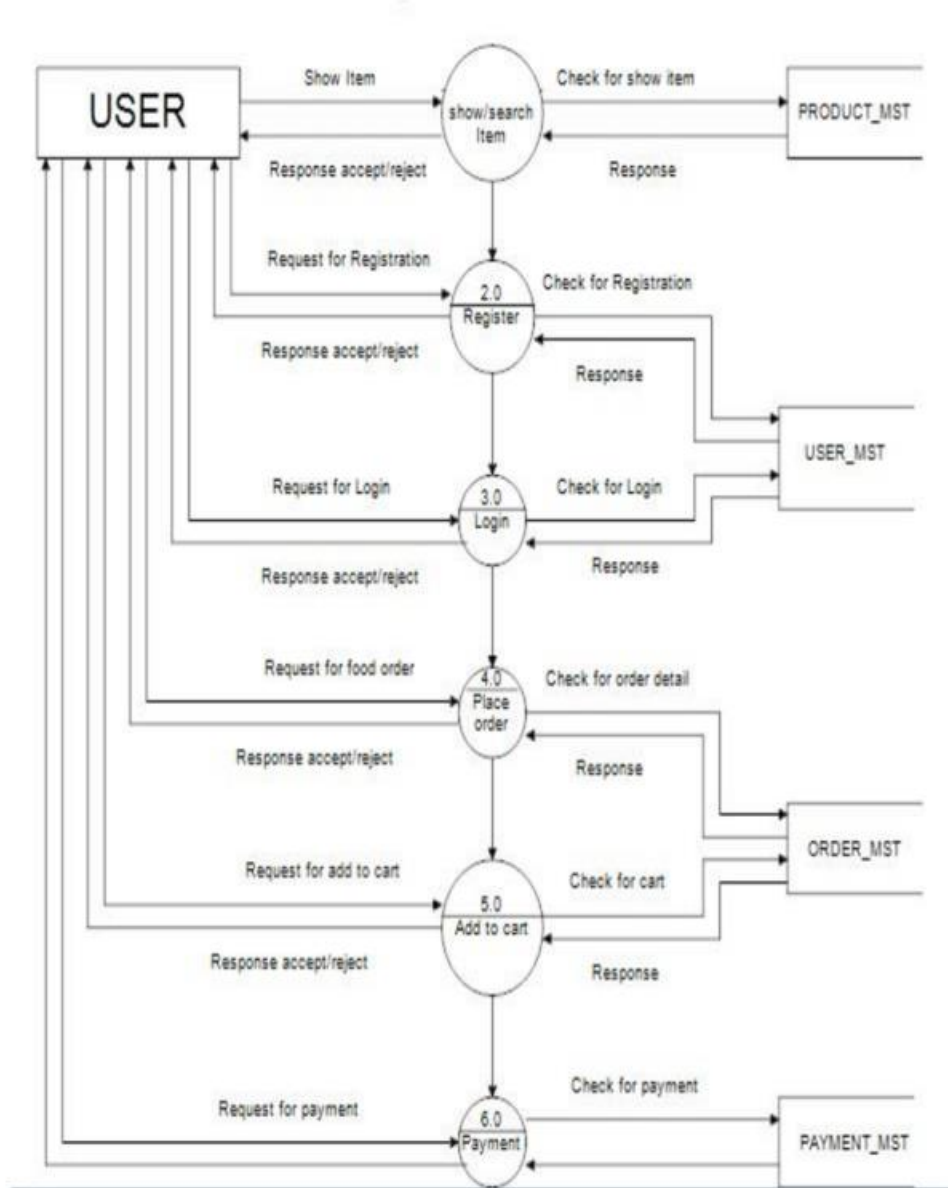


Figure 2.2.Datamodel-table design

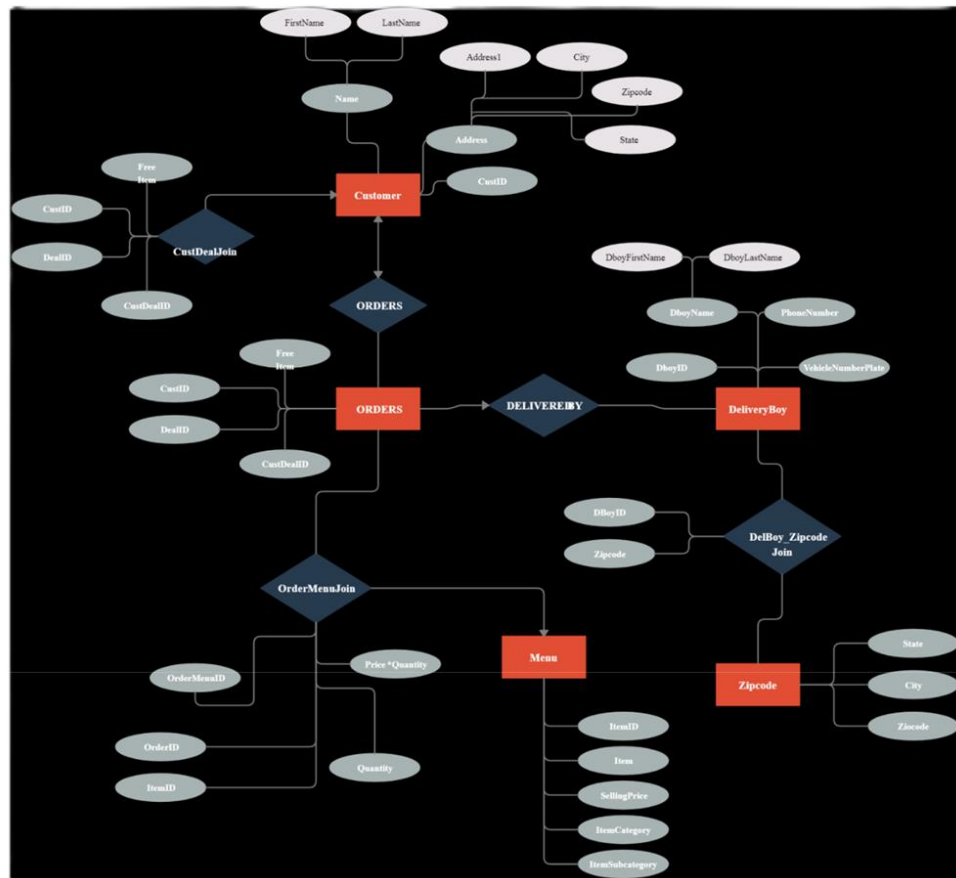


Figure 2.3.entity relationship diagram

Table 2.1.customer description

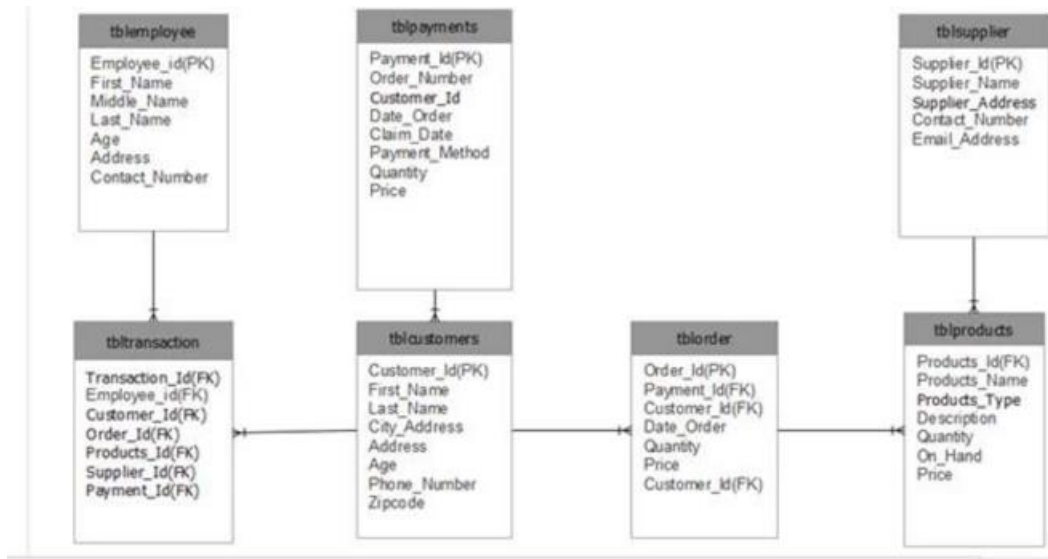
Table Name- Customer (This table contains details of customer)

Field Name	Data Type	Field Size	Description	Constraints
Cust_Id	Number	Integer	Customer ID.	Primary Key
Cust_Name	Text	255	Customer Name.	Not Null
Cust_Add	Text	255	Customer Address.	Not Null
Cust_Gender	Text	255	Customer Gender.	Not Null
Cust_PhNo	Number	Double	Customer Mobile Number.	Not Null
Cust_Date	Date/Time	-	Record Entry Date.	Not Null

Table 2.2.purchase bill

Table Name- Purchase Bill (This table contains details of Purchase Bill)

Field Name	Data Type	Field Size	Description	Constraints
PBill_Id	Number	Integer	Bill ID.	Primary Key
Sup_Id	Number	Integer	Supplier ID.	Foreign Key
Prod_Id	Number	Integer	Brand Id	Foreign Key
Prod_Price	Number	Double	Product Unit Price.	Not Null
Prod_Qty	Number	Integer	Product Quantity.	Not Null
T_Amount	Number	Double	Product Total Price.	Not Null
Pur_Date	Date/Time	-	Record Entry Date.	Not Null
Exp_Date	Number	Integer	Expiry Date	Not Null



User Interface –Refer Forms Design and Reports.it should show forms as well as reports with its descriptions

Note –refer form Design that will give on your what’s app group.Such as

Login page

This is login page for admin/manager that operates the system.

Home page

This is the home page where all the information related to website is visible such as menus, about us, cart ,payment etc.

My Cart

It is cart page it contain details about menu cart.

Payment Confirmation

The Web Ordering System

Users of the web ordering system, namely restaurant customers, must be provided the following functionality:

- Create an account.
- Manage their account.
- Log in to the system.
- Navigate the restaurant's menu.
- Select an item from the menu.
- Customize options for a selected item.
- Add an item to their current order.
- Review their current order.
- Remove an item/remove all items from their current order.
- Provide delivery and payment details.
- Place an order.
- Receive confirmation in the form of an order number.

As the goal of the system is to make the process of placing an order as simple as possible for the customer, the functionality provided through the web ordering system is restricted to that which most pertinent to accomplish the desired task. All of the functions outlined above, with the exceptions of account creation and management, will be used every time a customer places an order. By not including extraneous functions, I am moving towards my goal of simplifying the ordering process.

Menu Management System

The menu management system will be available only to restaurant employees and will, as the name suggests, allow them to manage the menu that is displayed to users of the web ordering system. The functions afforded by the menu management system provide user with the ability to, using a graphical interface:

- Add a new/update/delete vendor to/from the menu.
- Add a new/update/delete food category to/from the menu.
- Add a new/update/delete food item to/from the menu.
- Add a new/update/delete option for a given food item.
- Update price for a given food item.
- Update default options for a given food item.

- Update additional information (description, photo, etc.) for a given food item.

It is anticipated that the functionality provided by this component will be one of the first things noted by the restaurant user, as they will have to go through it to configure their menu, etc. before beginning to actually take orders. Once everything is initially configured, however, this component will likely be the least used, as menu updates generally do not occur with great frequency.

Order Retrieval System

Of the three components, the order retrieval system is functionally the simplest. Like the menu management system, it is designed to be used only by restaurant employees, and provides the following functions:

- Retrieve new orders from the database.
- Display the orders in an easily readable, graphical way.
- Mark an order as having been processed and remove it from the list of active orders.

User Interface Specifications

Each of the system components will have their own unique interface. These are described below.

Web Ordering System

Users of the web ordering system will interact with the application through a series of simple forms. Each category of food has its own form associated with it which presents a drop down menu for choosing which specific item from the category should be added to the order, and a series of check boxes and radio buttons for selecting which options are to be included. Adding an item to the order is accomplished by a single button click. Users select which category of food they would like to order, and therefore which form should be displayed, by navigating a menu bar, an approach which should be familiar to most users.

Entering delivery and payment details is done in a similar manner. The user is presented with a form and must complete the required fields, which include both drop down and text boxes, before checking out and receiving a confirmation number. One thing worth noting here is that whenever possible drop down boxes and buttons were used over freeform input in order to both simplify the ordering process and reduce the possibility of an SQL injection attempt.

Menu Management System

User interaction with the menu management system is similar to that with the web ordering system. Users navigate a tree structure to find the vendor, category, or specific food item that they would like to modify and after making their selection they are presented with a form which displays all of the current fields and values associated with that item, all of which can be modified or removed. The form also presents buttons which allow the addition of new fields and values. Unlike the web ordering system, however, most of the input here will be freeform, specifically in the form of text boxes, since there is no finite set of fields which could be added. This does not raise a major concern though, as input sanitation will be performed, and the user, who is assumed to be a restaurant employee, is less likely to be malicious than a web user.

Order Retrieval System

User interaction with the order retrieval will be very simple. The application will automatically fetch new orders from the database at regular intervals and display the order numbers, along with delivery time, in a panel on the left hand side of the application. To view the details of an order, the user must simply click on that order number, which will populate the right-hand panel with the details, displayed in an easy to read and navigate tree structure. This structure can intuitively be expanded and collapsed to display only the desired information. Finally, once an order is processed, the user clicks a single button, labeled “Processed”, to remove it from the list of active orders.

Non-functional Requirements

Because the design patterns of the Online Ordering System are pretty much the standard for a web application, the non-functional requirements of the system are very straightforward. Although written using Google Web Toolkit, the application is cross-compiled to HTML and JavaScript, along with a PHP backend, all of which are supported by any reasonably well maintained web server, although I would recommend Apache2, and particularly the free XAMPP distribution.

All of the application data is stored in a PostgreSQL database, and therefore a PostgreSQL server must also be installed on the host computer. As with Apache2, this software is freely available and can be installed and run under most operating systems.

The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a restaurant that is not expecting to see much web traffic, or possibly doing only a limited test run, an average personal computer may be appropriate. Once the site starts generating

more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cutoffs will need to be determined through a more thorough stress testing of the system.

System Evolution

As mentioned in the system model, at the heart of the entire ordering system is the database. In fact, the system could be completely operational using nothing but the database and an appropriate shell utility, assuming that all users are well-versed in SQL and enjoy using it to order food. While this would be a bit extreme, it does illustrate the point that the one part of the system which will stay relatively constant is the database. On the other hand, it is very probable that the other components will continue to evolve with time. For example, with the booming popularity of mobile applications, I would really like to make the web interface available as a phone application as well. Also it may make sense to at some point migrate the menu management and order retrieval systems to web, or even mobile, applications as well, as some users may prefer to use them as such.

I am also certain that if this system goes into actual use, many requests will arise for additional features which I had not previously considered, but would be useful to have. For this reason, I feel as though the application can be constantly evolving, which I consider a very good thing.

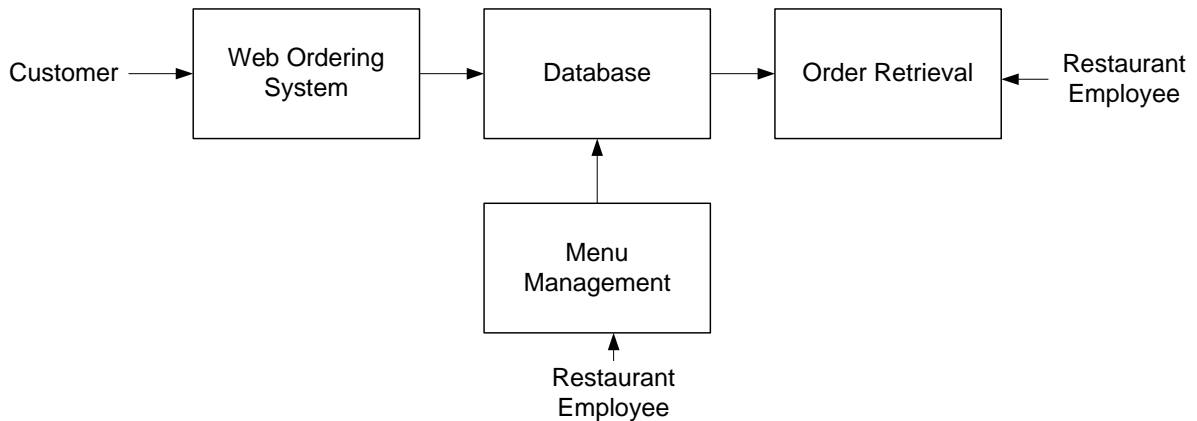
Chapter 3:

System Design

System Design

Level 1: The Database & the 3 Components

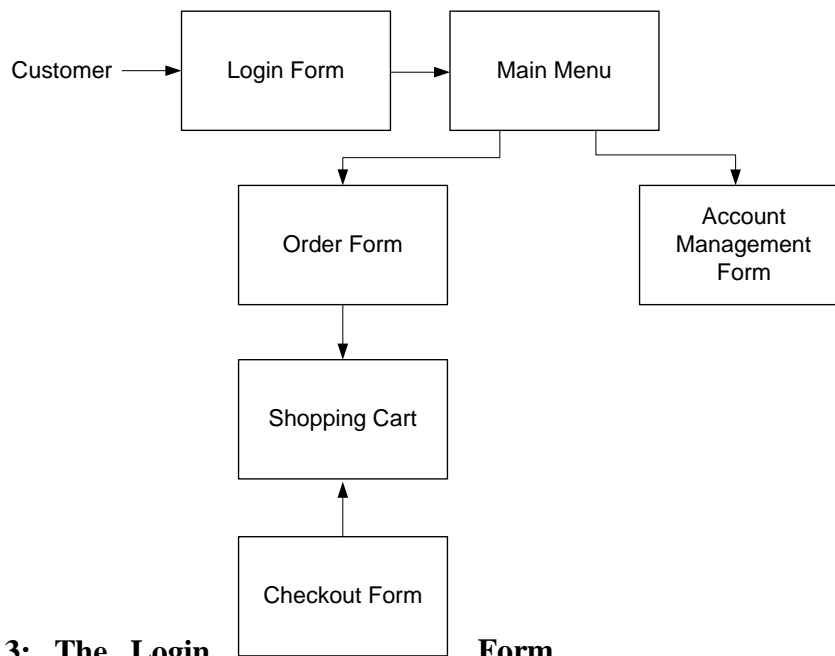
The structure of the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the restaurant to control what can be ordered by customers. The second component is the web ordering system and provides the functionality for customers to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the restaurant to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.



Level 2: Web Ordering System Components

The web ordering system is comprised of 6 major components. These are the login form, the main menu, the account management form, the order form, the shopping cart, and the checkout form. When the customer first arrives at the site, they are presented with the login form. After either signing in or, if they do not yet have an account, first registering and then signing in, the user is taken to a welcome page with the main menu. From here, they have two options – they can either change their password and other preferences through the account management form, or they can select an order form and begin adding items to their order. If they choose the second path, they can navigate the numerous order

forms using the main menu, each of which corresponds to a specific category of order items, adding items to their shopping cart along the way. At any time they can view and modify their shopping cart and when they are finally ready to place their order, they can proceed to the checkout form. The checkout form uses the contents of the shopping cart to present a summary of the order and to calculate the total cost, in addition to allowing the user to specify all of the necessary delivery details.



Level 3: The Login Form

The login form is standard for a form of this type. It provides text fields for username and password, which the user must enter before signing in. This form also gives the option for a user to register for the site if they have not yet done so.

Level 3: The Main Menu

The main menu, found at the top of the screen like in most applications, presents the user with two levels of selections. They must first choose the vendor they would like to view and then choose a category of food. Once they make these two selections, the application generates an order form specifically for that type of food, and displays this form to the user.

Level 3: The Account Management Form

Currently the account management form only offers the user the option to change their password.

Level 3: The Order Form

The order form, which is dynamically generated based on selections from the main menu,

Level 3: The Shopping Cart

The shopping cart performs much like a shopping cart in any other application. After an item is added to the order, it is displayed, along with its price, in the shopping cart. The shopping cart also keeps a running total of the current price of the whole order. By clicking on an item in the shopping cart, the user can review all of the details for that particular item. Finally, the shopping cart contains a button for the user to proceed to checkout.

Level 3: The Checkout Form

The checkout form is the user's last chance to verify that the contents of their order are correct before actually placing it. This form also provides fields for the user to supply all of the necessary checkout and delivery details (payment type, delivery address, etc.).

Level 2: Menu Management System Components

In order to make use of the menu management system, the user must interact with the navigation tree, which uses a hierarchical tree structure to display all of the vendors, categories of foods, and specific food items stored in the system. When the user selects an item from this tree, they are able to edit the item using the appropriate form – a Vendor Form if a vendor is selected, a Category Form if a category of foods is selected, and a Food Form if an individual food item is selected.

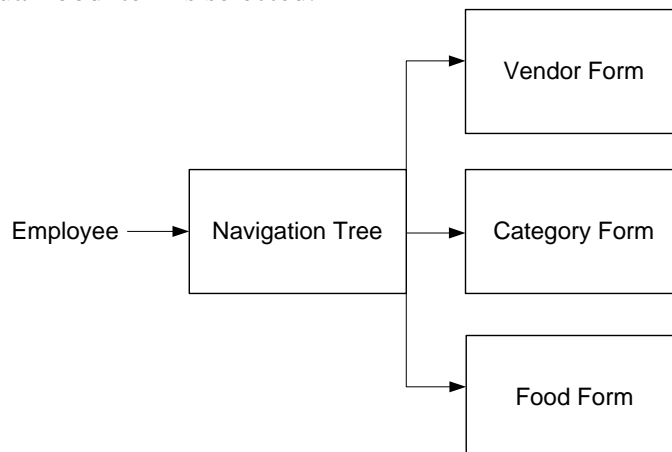


Figure 3.1 navigation tree

Level 3: The Navigation Tree

The navigation tree is a 3-level (excluding the root) hierarchical arrangement, with each leaf corresponding to a form. At the first level are vendors, at level two categories of food, and at level 3 individual food items. When a leaf is selected, it brings up a form corresponding to the item at that leaf.

Level 3: The Forms

There are three types of forms in the menu management system - Vendor Forms, Category Forms, and Food Forms. The three forms are all similar, allowing the user to add, edit, and remove information relevant to the selected item. Where they differ is in the specific fields that the user is able to edit. After changes to any of the forms are saved, the necessary records in the database are updated.

Level 2: Order Retrieval System Components

The simplest of the three components, the order retrieval system can be broken down into just two components. They are the summary panel, which displays a list of all currently active orders, and the order detail panel, which highlights just a single order. When the application first starts, the order details for the first order in the list are displayed. In order to view the details of a different order, the user must simply select it from the list in the summary panel.

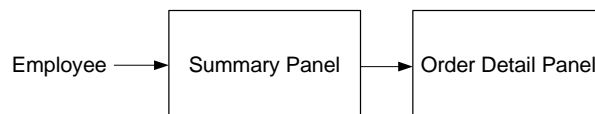


Figure 3.2. summary

panel

Level 3: Summary Panel

The summary panel, located on the left side of the screen, displays a list of all currently active orders, along with their delivery times and statuses. By changing the selected item in this list, the user is able to control the contents of the order detail panel.

Level 3: Order Detail Panel

The order detail panel which contains a hierarchical tree structure for viewing all of the details related to the order which is currently selected in the summary panel. This component also contains a button to mark the order as processed and remove it from the list of active orders.

User Interface Design

The user interface design principles can be broken into two groups. The interface in the web application is designed to limit free form user input, using mostly drop down menus, radio buttons and check boxes. This is done for two reasons – to simplify the ordering process as much as possible, and to limit SQL injection attempts. Free form input is necessary in the menu management component, however, as all of the values must be user supplied. The interface for this component contains traditional forms comprised of text fields and corresponding labels, along with save and discard buttons for each form.

Help System Design

Due to the form-based nature of the applications, the design of the help systems will be minimal. In both the desktop and web applications, it will be accessed from the application's main menu and will open in a new window. Modeled after the typical help system design, it will be both searchable and include a navigation tree highlighting common topics. There will be a help page for each form type, describing the significance of each field on the page.

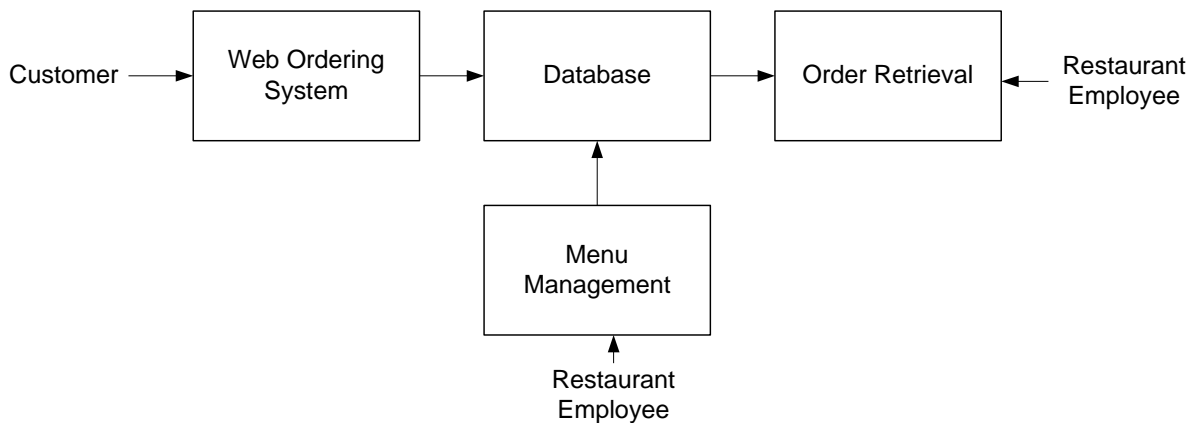
Chapter 4:

Testing Design

4.1 Testing

Phases

The structure of the system can be divided into three main logical components, plus the database, which is invisible to the end user. Each of these components must be tested individually, and the approaches which will be used for each component are 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 applications, I wrote all of the SQL statements I expected to need and executed them directly, essentially isolating the database, using the psql client. By doing this I was able to reveal, and promptly fix a large percentage of the errors within the database itself.

Web Ordering System

Testing of the web ordering system will be the most strenuous, as it is the 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 web 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, I will also be monitoring and inspecting the

JSON requests and responses (using Firebug) to make sure that everything is sent and received correctly.

4.2 TESTING APPROACH

To build up our project we use software testing process for executing a program with the intent of finding errors that is uncovering errors in a program makes it a feasible task and also trying to find the error in a program as it is destructive process.

4.2.1 TYPE OF TESTING

Type of testing we Use in Our Project Here we just mentioned that how the testing is related to this software and in which way we have test the software? In our project we have used 4 types of testing these are listed below –

- Unit testing: Unit testing where individual program unit or object classes are tested here by using this testing we have focused on testing the functionality of methods.
- Module Testing : Where this is the combination of unit is called module. Here we tested the unit program is where the module program have dependency
- Sub- system Testing : The we combined some module for the preliminary system testing in our project
- System Testing : where it is the combination of two or more sub – system and then it is tested. Here we tested and entire system as per the requirements.

4.2.2 USE CASE

A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

- The actor. The system user -- this can be a single person or a group of people interacting with the process.
- The goal. The final successful outcome that completes the process.
- The system. The process and steps taken to reach the end goal, including the necessary functional requirements and their anticipated behaviours.

- Identifying all system users and creating a profile for each one. This includes every role played by a user who interacts with the system.
- Selecting one user and defining their goal -- or what the user hopes to accomplish by interacting with the system. Each of these goals becomes a use case.
- Describing the course taken for each use case through the goal.
- Considering every alternate course of events and extending use cases -- or the different courses that can be taken to reach the goal.
- Identifying commonalities in journeys to create common course use cases and write descriptions of each

Coding

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <meta charset="UTF-8">

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

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

    <meta name="Description" content="Enter your description here" />

    <title>Restro 1839</title>

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootst
rap.min.css" rel="stylesheet">

    <link    rel="stylesheet"    href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.2/css/all.min.css">

    <linrel="stylesheet" href="https://fonts.googleapis.com/css?family=Quicksand">
    <style>

ul,
ol {
    margin: 0;
    padding: 0;
}
```

```

/===== Variable Define =====/
:root {
  --primary-color: #FFC222;
  --secondry-color: #00a149;
  --white-color: #fff;
  --text-color: #b0b0b0;
  --black-color: #1E1D23;
  --primary-font: 'Roboto', sans-serif;
  --secondry-font: 'Quicksand', sans-serif;
}
html {
  scroll-behavior: smooth;
}
body {
  font-family: var(--primary-font);
  font-size: 100%;
  font-weight: 400;
}
.testimonial-section .carousel-indicators button{
  width:15px;
  height:15px;
  outline:none;
  border-radius: 50%;
  border: none;
  margin-right:1rem;
  opacity:0.2;

```

```
testimonial-section .carousel-indicators button.active{  
  opacity:1;  
}  
  
.forms {  
  padding: 20px  
}  
  
.inputs input {  
  margin-bottom: 13px;  
  border: none;  
  border-bottom: 2px solid #eee  
}  
  
.inputs input:focus {  
  margin-bottom: 13px;  
  border: none;  
  border-bottom: 2px solid #7B1FA2;  
  box-shadow: none  
}  
  
.inputs textarea {  
  margin-bottom: 13px;  
  border: none;  
  border-bottom: 2px solid #eee;  
  width: 100%;  
  resize: none  
}  
  
.inputs textarea:focus {  
  margin-bottom: 13px;
```



```

border: none;

border-bottom: 2px solid #7B1FA2;

box-shadow: none;

resize: none

}

.form-control {

padding: .375rem .25rem;

}

.color{

color:orange;

}

.col-md-9{

color:whitesmoke;

}

.get{

color:darkorange;

}

.mes{

color:black;

}

/footer/

.footer-dark {

padding:50px 0;

color:#f0f9ff;

background-color:black;

```

```

}

<!-- header design -->

<header>

<nav class="navbar navbar-expand-lg navigation-wrap">

<div class="container">

<a class="navbar-brand" href="#">



</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<!-- <span class="navbar-toggler-icon"></span> -->

<i class="fas fa-stream navbar-toggler-icon"></i>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ms-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="home.php">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="about.php">About Us</a>

</li>

<li class="nav-item">

<a class="nav-link" href="foods.php">Our Menu</a>

</li>

```

```

<li class="nav-item">

<a class="nav-link" href="contact.php">Contact Us</a>

</li>

<li class="nav-item">

<a class="nav-link" href="login.php" class="main-btn">Login</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<!-- section-1 top-banner -->

<section id="home" class="banner full-screen-mode parallax">

<div class="container-fluid px-0 top-banner">

<div class="container">

<div class="row">

<div class="col-lg-5 col-md-6">

<!--<h1>WELCOME TO <br> <span>RESTRO 1839</span></h1>-->

<div class="banner-cell">

<h1>Welcome To Res<span class="typer" id="some-id" data-delay="200"

data-delim=":" data-words="tro 1839" data-colors="red"></span><span

class="cursor" datacursorDisplay="_" data-owner="some-id"></span></h1>

</div>

<!-- <h2>Medium 2-topping* pizza</h2>

<p>*Additional charge for premium toppings. Minimum of 2 required.</p>--
>

```

```

<div class="mt-4">

  <button class="main-btn">Order now <i class="fas fa-shopping-basket
ps3"></i></button>

</div>

</div>

</div>

</div>

</div>

</section>

<section id="about">

  <div class="row d-flex justify-content-center align-items-center">

    <div class="col-md-12">

      <div class="About-us text-center">


        <h2 class="write">About Us</h2>

        <p class="mb-5 text-white">We celebrate people, revel processes, share
receipt, and
most of all, We have fun.</p>


        <h2 class="choose">Choose Us</h2>

        <h4 class="why text-white">Why choose Us ?</h4>

        <br/>


        <div class="container d-flex align-items-center justify-content-center position-
relative
flex-wrap">

```

```
<div class="card d-flex position-relative flex-column">
```

```
<div class='imgContainer'>
```

```
 </div>
```

```
<div class="content">
```

```
<h2>1. Delicious Food</h2>
```

```
<p class="content-text">we provide delicious & healthy fusion food by  
drawing on
```

```
the simple traditions of cuisines from all over the globe. We believe delicious  
good food offer
```

```
great smile .</p>
```

```
</div>
```

```
</div>
```

```
<div class="card d-flex position-relative flex-column">
```

```
<div class='imgContainer'>
```

```
 </div>
```

```
<div class="content">
```

```
<h2>2. Expert Chefs</h2>
```

```
<p class="content-text">Cooks will usually occupy entry-level positions in a  
kitchen
```

```
and experience training from the specialized chefs. </p>
```

```
</div>
```

```
</div>
```

```
<div class="card d-flex position-relative flex-column">
```

```
<div class='imgContainer'>
```

```
 </div>
```

```
<div class="content">
```

```
<h2>3. Friendly People</h2>
```

```
<p class="content-text">We treat employees fairly and conduct business in a  
way
```

```
that's friendly to the environment and compassionate towards animals.</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- section-3 about
```

```
<section id="about">
```

```
<div class="about-wrapper">
```

```
<div class="overlay">
```

```
<div class="row d-flex justify-content-center align-items-center">
```

```
<div class="col-md-9">
```

```
<div class="About-us text-center">
```

```
<br />
```

<h2>Our Restaurant</h2>

<h4 class="write">About Us</h4>

<p class="mb-5 text-white">We celebrate people, revel processes, share receipt, and

most of all, We have fun.</p>

<h2 class="choose">Choose Us</h2>

<h4 class="why text-white">Why choose Us ?</h4>

<div class="row row-cols-1 bg-transaprent row-cols-md-3 g-4">

<div class="col">

<div class="card h-100 w-75 bg-black text-white">

ONLINE FOOD ORDERING SYSTEM Page No. 37

<h2 class="let">01</h2>

<div class="card-body">

<h4 class="auto-style1">Delicious Food. </h4>

<p class="card-text">we provide delicious &healthy fusion food by drawing on the

simple traditions of cuisines from all over the globe.We beleive delicious good food offer

great smile .</p>

</div>

<div class="card-footer">

<small class="text-muted"></small>

</div>

</div>

</div>

```

<div class="col">

<div class="card h-100 w-75 bg-black text-white ">

<h2 class="alert">02</h2>

<div class="card-body">

<h4 class="auto-style1">Expert chefs.</h4>

<p class="card-text">Cooks will usually occupy entry-level positions in a
kitchen and

experience training from the specialized chefs. </p>

</div>

<div class="card-footer">

<small class="text-muted"></small>

</div>

</div>

</div>

<div class="col">

<div class="card h-100 w-75 bg-black text-white">

<h2 class="active">03</h2>

<div class="card-body">

<h4 class="auto-style1">Friendly people.</h4>

<p class="card-text">We treat employees fairly and conduct business in a way
that's

friendly to the environment and compassionate towards animals.</p>

ONLINE FOOD ORDERING SYSTEM Page No. 38

<div class="card-footer">

<small class="text-muted"></small>

</div>

</div>

```


</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>-->

<!-- menu section -->

<section id="menu" class="menu_wrapper">

<div class="container">

<div class="row ">

<div class="col-12 text-content text-center">

<h2>OUR MENU</h2>

<p>Good Food Offers Great Smile.</p>

</div>

</div>

In phase two I will perform exceptional use case testing, where I will artificially generate cases that shouldn't arise, but possibly could, and monitor how the system handles these cases. These cases fall into one of 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 the 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.

Menu Management System

Testing of the menu management system will be very similar to that of the web ordering system, as I will first run test cases where the user supplies acceptable values, and afterwards test how the system responds to unexpected input.

Order Retrieval System

Of all the components, testing of the order retrieval system will be the simplest. Since it is simply an interface to display the results of database queries and has no potential to change the state of the system, the only thing that really needs to be tested is how the system responds when a result set is not in the form it is expecting. This will be done by intentionally corrupting the database and analyzing the response of the order retrieval system.

Requirements Traceability

In the requirements document, I specified the following functional requirements:

For the Web Ordering System:

- Create an account.
- Manage their account.
- Log in to the system.
- Navigate the restaurant's menu.
- Select an item from the menu.
- Customize options for a selected item.
- Add an item to their current order.
- Review their current order.
- Remove an item/remove all items from their current order.
- Provide delivery and payment details.
- Place an order.
- Receive confirmation in the form of an order number.

For the Menu Management System:

- Add a new/update/delete vendor to/from the menu.
- Add a new/update/delete food category to/from the menu.
- Add a new/update/delete food item to/from the menu.
- Add a new/update/delete option for a given food item.
- Update price for a given food item.
- Update default options for a given food item.
- Update additional information (description, photo, etc.) for a given food item.

For the Order Retrieval System:

- Retrieve new orders from the database.
- Display the orders in an easily readable, graphical way.
- Mark an order as having been processed and remove it from the list of active orders.

In order to assure thorough testing, I will have to generate cases in which each of these functions is performed, not just with a single input value, but an example of each possible class of input. This may seem tedious, but is absolutely necessary since the system is so heavily dependent on user input and must respond appropriately to anything the user may do. The good news is that because the system was design to contain only the absolutely necessary functionality, the testing of the different functions should flow smoothly into one another.

Testing Schedule

Throughout all of the design and development phases, I have been performing unit tests on each component, assuring that it works properly before introducing it into the rest of the system, but I plan on beginning rigorous testing of the system starting in the first week of December. I will begin by putting the system through its normal paces as a normal user would be expected to. I will test the entire functionality of the system, but will do so following the normal logical flow, and only providing reasonable values for user input.

Once I am happy with how the system performs for normal use cases (which hopefully will be completed by December 8th), I will move on to testing the exceptional use cases. These are the use cases that would never be encountered by the average user, but, whether through confusion or malice, may come up.

Two examples would be adding hundreds of items to an order or attempting to supply an SQL statement as a login credential.

In the final phase of testing, which I will begin on December 15th, I will carry out stress and performance testing. In this phase, I will bombard the system with an increasing number of HTTP requests and then measure response times and see at what point things begin to break down.

Recording Procedures

The recording procedures I will use can be divided into two categories. For the use case testing, the tests will be designed in a “Pass/Fail” manner, making recording the results very simple. Each time a series of tests is run, the results will be recorded in a spreadsheet, where each outcome can be identified by the name of the test case along with the execution date and time.

For the stress and performance testing, the procedure will be similar. However, since these tests involve numerical results rather than simply “Pass/Fail”, I will be able to not only record the results, but to also perform analysis (both numerically and graphically) on them to get a better idea exactly what the numbers mean.

Hardware and Software Requirements

The testing to be performed will require no special hardware, but some specially designed software may be used, particularly when performing stress testing. I plan on writing and utilizing a multi-threaded Java program to bombard the system with an increasing number of HTTP requests and then measure response times and see at what point things begin to break down. I also will be using the Firefox plug-in Firebug to monitor the JSON requests and responses.

Chapter 5: User Manual

Introductory Manual

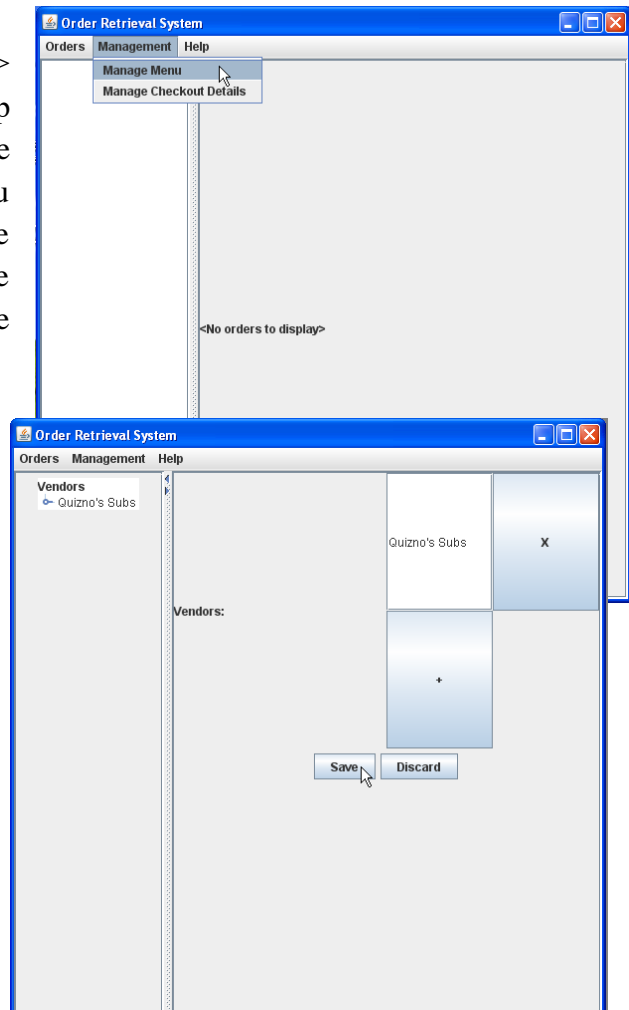
Using the Desktop Application

When the system is first installed, it contains no menu information. Therefore, the first step in using the system is to add vendor information. Start by launching the desktop client.

Managing Vendors

Navigate to “Management” > “Manage Menu” in the toolbar at the top of the screen. This will bring up the menu management view. The menu management view provides a simple graphical interface for editing the content of the menu displayed to the customer.

Click the “+” button to add a new vendor, then supply the name of the vendor and click “Save”. This adds the vendor to the navigation tree on the left. If there will be multiple vendors using the system, repeat this process for each of them. If at any point you would like to modify the vendor names, do so from this screen.



Managing Categories

The screenshot shows the 'Order Retrieval System' window with the 'Management' tab selected. On the left, a 'Vendors' tree shows 'Quizno's Subs' selected. The main area displays the 'Quizno's Subs' category. The 'Name' field is 'Quizno's Subs'. The 'Categories' field contains a list of categories: Subs, Bullets & Torpedos, Flatbread Sammies, Chopped Salads, Soups, and a '+' button. Below the list are 'Save' and 'Discard' buttons.

Category	Action
Subs	X
Bullets & Torpedos	X
Flatbread Sammies	X
Chopped Salads	X
Soups	X
+	

Figure 4.1 order retrieval system

Categories represent a related group of foods, usually what would be found as a menu heading. Categories must be associated with a vendor, so to add a new category select the appropriate vendor from the navigation tree on the left. In the category field, click the “+” button to add a new category then supply the category name and click “Save”.

The screenshot shows the 'Order Retrieval System' window with the 'Management' tab selected. On the left, the 'Vendors' tree shows 'Quizno's Subs' expanded, with 'Soups' selected. The main area displays the 'Soups' category. The 'Name' field is 'Soups'. The 'Sizes' field contains a list of sizes: Cup, Bowl, and a '+' button. The 'Options' field contains a list of options: Crackers, Broccoli Cheese Soup, Chicken Noodle Soup, Chili, and a '+' button. Below the list are 'Save' and 'Discard' buttons.

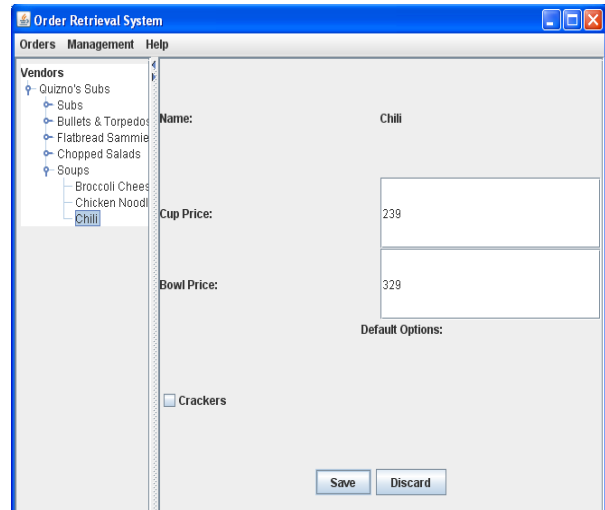
Size	Action
Cup	X
Bowl	X
+	

Option	Action
Crackers	X
+	
Broccoli Cheese Soup	X
Chicken Noodle Soup	X
Chili	X
+	

Now select the category you just created from the navigation tree. This will bring up a new form. If items from the category are available in multiple sizes, add these sizes to the “Sizes” field. Some categories allow customers to specify additional options. These options are specified in the “Options” field. Finally, the food items within this category must be added. This is done by clicking the “+” button in the “Food Items” field and supplying the name of the item. When you are finished, click “Save”.

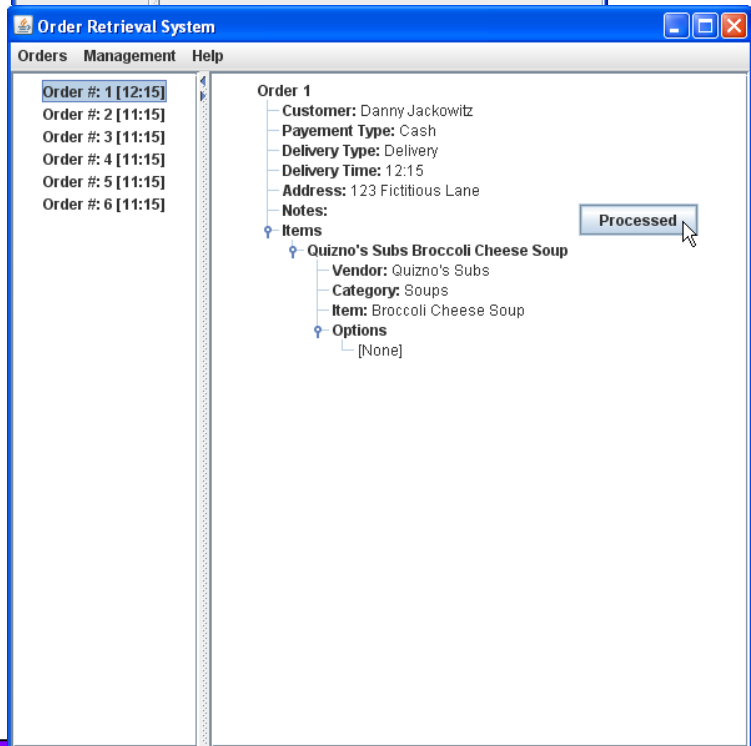
Managing Foods

To edit food details, select the food you wish to modify from the navigation tree. From this screen you can now modify the prices and default options for the selected food item. Repeat this process for all items you have added. The ordering system is now ready for use. If at any point you need to edit menu information you can return to these screens through the “Management” tab on the application toolbar.



Managing Orders

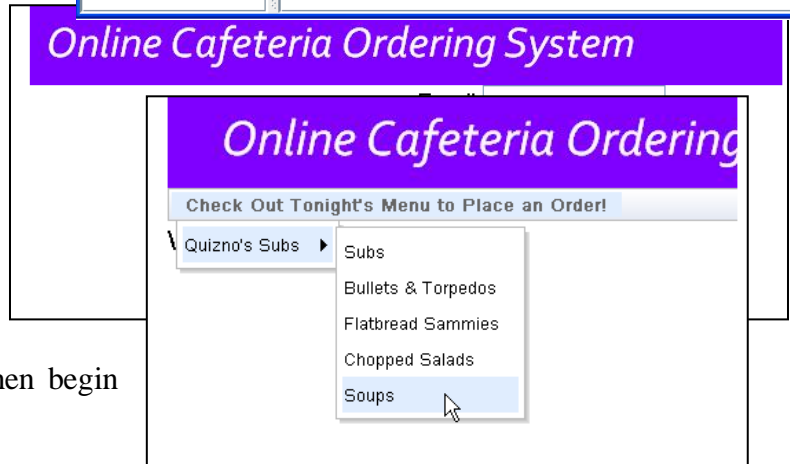
To manage placed orders, navigate to “Orders” > “View Orders” on the application toolbar. You will now see a list of all currently active orders on the left side of the application. To view the details of a specific order, select that order from the list. Once an order has been prepared, mark it as processed by clicking the “Processed” button. As new orders are placed, they will be added to the list of active orders.



Using the Web Ordering System

Signing Up & Signing In

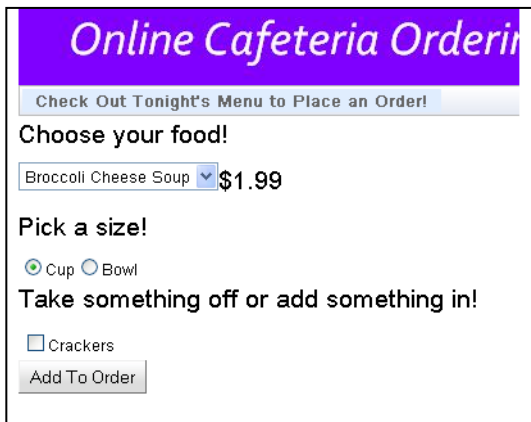
When a customer first visits the website, they must register before they can begin using the system. To do this, click on the “Sign Up” button on the main page. Then fill out the required fields and click “Register”. An e-mail containing your password will be sent to the specified account and you can then begin using the system.



Selecting Food Items

After signing in on the main page, click “Check Out Tonight’s Menu...” and select the category of food you would like to order. This will bring up an order form specific to that category.

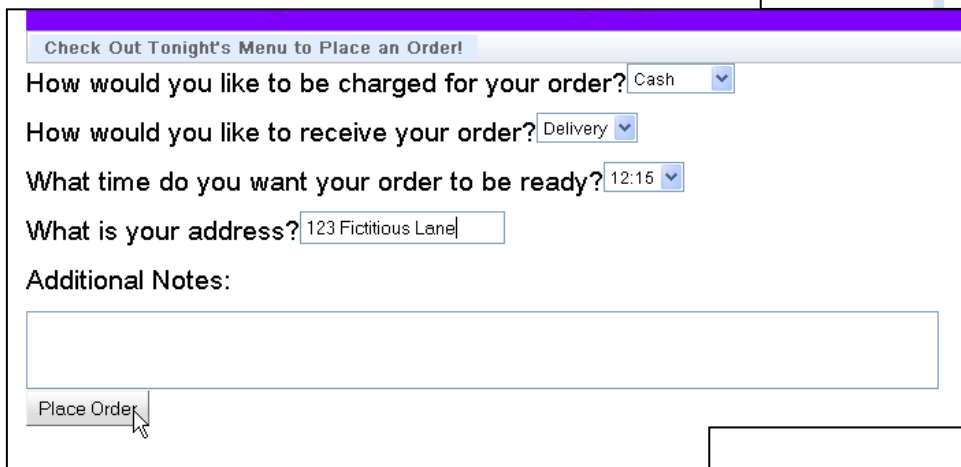
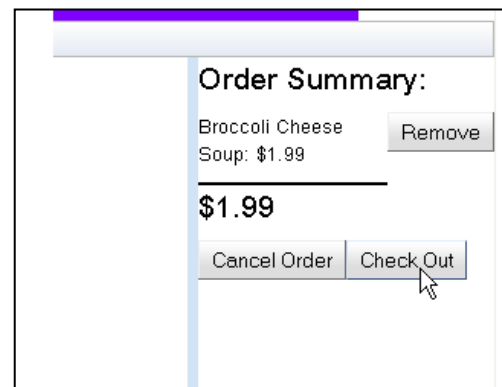
Figure 4.2 payment page



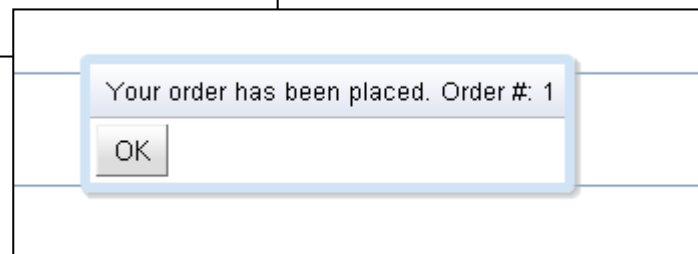
On this form, select the food you would like to order, specifying the desired size and selecting any appropriate options, then click add to order. This will add the item to your “shopping cart”, located on the far right of the screen. Repeat this process for as many items as you would like to order.

Placing an Order

When you are finished adding items to your order, click “Check Out” from within the shopping cart. You will then be presented with one final form to complete. After supplying the required payment and delivery details, click “Place Order”.



You should then receive confirmation that your order was placed successfully, along with your order number, completing the ordering process.



Using the Help System

Figure 4.3order summary

The help systems in both the desktop application and online ordering system can be accessed through the “Help” tab on the application toolbar. In both cases, accessing this tab will present the user with a tutorial very similar to those presented above, instructing them, step-by-step, on how to complete the desired action. In addition to this in-application help, the help document for the desktop application will also be distributed with the system as a .pdf to make employee training easier.

System Reference Manual

Services (Alphabetical)

Desktop Application

Adding Categories

Adding Foods

Adding Vendors

Deleting Categories

Deleting Foods

Deleting Vendors

Marking Orders as Processed

Modifying Categories

Modifying Foods

Modifying Vendors

Retrieving Orders

Web Ordering System

Adding an Item to Your Order

Checking Out

Modifying an Item within Your Order

Placing an Order

Removing an Item from Your Order

Selecting a Food Item

Signing In

Signing Up

Viewing the Menu

Error Recovery

Because the all of the information relevant to the functionality of the system is stored in the database, which is designed to maintain consistent state even when an application error occurs, the most effective way to recover from an error is to restart the application, either by closing and restarting the application (desktop application) or by refreshing the browser (web ordering system). In either case, a restart should have no negative effects on the database and the application should function normally.

If, however, the same error continually occurs, this likely indicates that the database has somehow been corrupted. Although measures have been taken to protect the consistency of the database, there may still be a slight chance of this corruption occurring through either user error or intentional tampering. If the database has become corrupted, a clean (destructive) reinstallation of the system must be performed. This can be done following the same process used to initially install the system.

Installation

The installation process consists mainly of the creation of the PostgreSQL database that coordinates the entire system. At the present moment, this creation is handled by a script which must be run from within a PostgreSQL client (such as psql). This is not ideal, however, and when the time comes for the system to actually be distributed to a restaurant for testing, this script will be wrapped in an executable (most likely platform dependent), which the user will simply be able to double-click to complete the installation process.

Chapter 6:

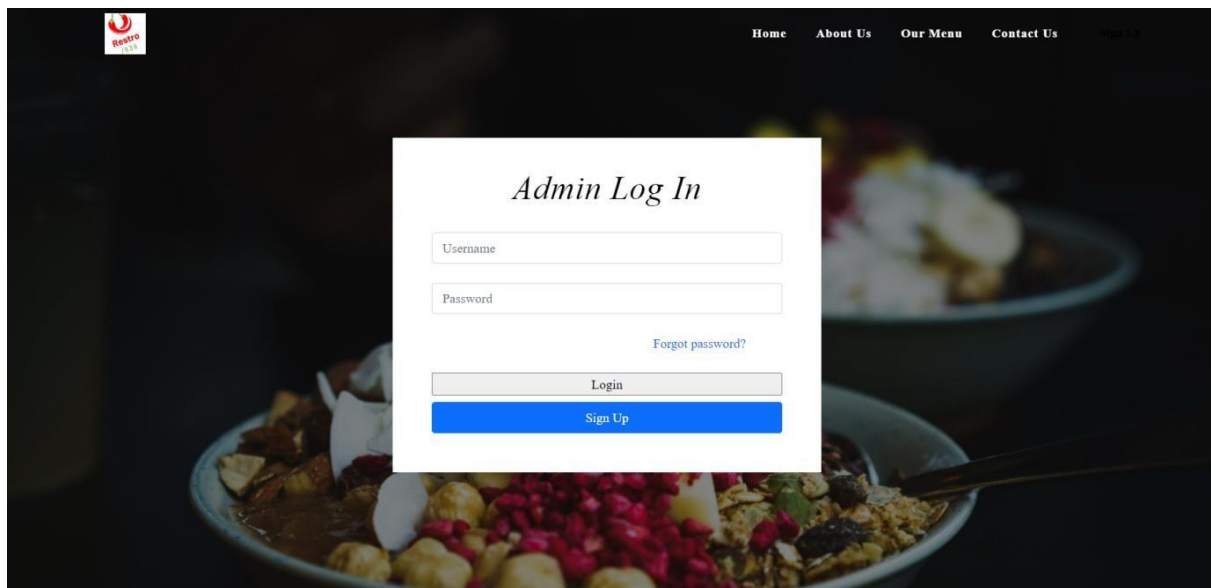
Results and discussion

5.1 OUTPUT SCREEN

Home page



Admin login





GET IN TOUCH

Contact Us



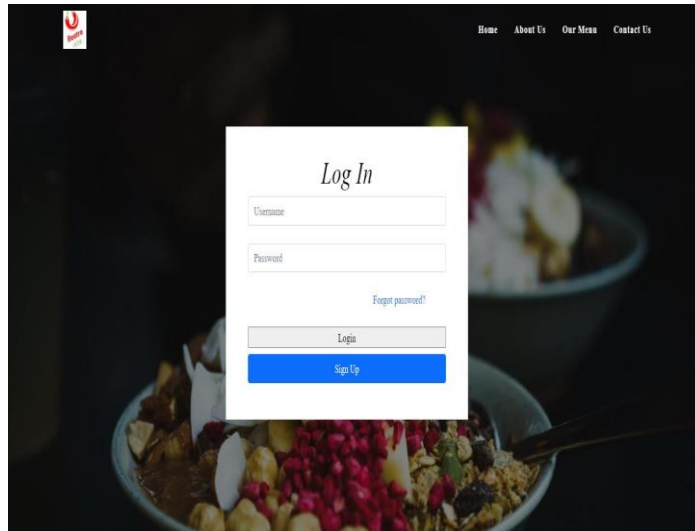
Address
Bistupur, Jamshedpur




Phone
9304494874 , 8797018830

email address
shubhamsingh93044@gmail.com


Send Message




Food Menu




Smoky BBQ Pizza
Rs.9.00
Best Firewood Pizza in Town.
[Order Now](#)



Chicken Wrap
Rs.5.00
Crispy flour tortilla loaded with juicy chicken, bacon, lettuce, avocado and cheese drizzled with a delicious spicy Ranch dressing.
[Order Now](#)



Cheeseburger
Rs.4.00
A cheeseburger is a hamburger topped with cheese. Traditionally, the slice of cheese is placed on top of the meat patty.
[Order Now](#)



Grilled Cheese Sandwich
Rs.3.00
Assembled by creating a cheese filling, often cheddar or American between two slices of bread and is then heated until the bread browns and cheese melts.
[Order Now](#)

45



About Us

We celebrate people, revel processes, share receipt, and most of all, We have fun.

Choose Us

Why Choose Us ?



Services

Good Food
Best Delivery
Hooding

About

Company
Team
Careers


Restro /839

Manage Admin

[Add Admin](#)

S.N.	Full Name	Username	Actions		
1.	Administrator	admin	Change Password	Update Admin	Delete Admin
2.	Shubham	shubham	Change Password	Update Admin	Delete Admin
3.	Ankita Kumari	Ankita	Change Password	Update Admin	Delete Admin

Selected Food



Smoky BBQ Pizza

Rs.9.00

Quantity

Delivery Details


Full Name

Phone Number

Email

Address

Pay and Confirm Order



Add Admin

Full Name:	<input type="text" value="Enter Your Name"/>
Username:	<input type="text" value="Your Username"/>
Password:	<input type="password" value="Your Password"/>
<input type="button" value="Add Admin"/>	

Chapter 7:

Conclusion

RESTRO 1839 (Food ordering System) has been computed successfully and was also tested successfully by taking "Test Cases". It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

Food ordering System is developed using HTML, CSS, JS as front end and PHP, MySQL as back end on windows environment.

The goals that are achieved by the software are:

- User friendly
- Simplification of the operations
- Portable and flexible for further enhancement
- Less processing time and getting required information

With the help of this system, people can easily order the food. It can also ensure that the people do not waste their precious time and use their time productively in the other works. In long run, this will ensure that it helps to reduce labour cost. This system proves to be more cost effective and reliable over other systems. This system is difficult to forge or cheat when compared to other systems in terms of payment for the food. It is very easy to use and has least maintenance. It does not require any human intervention and thus can be called fully automated. There isn't any limitations as such for this system, however one needs to take care of the smaller parameters like server breakdown while this system is implemented.

An online food ordering system is developed where the customers can make an order for the food and avoid the hassles of waiting for the order to be taken by the waiter. Using the application, the end users register online, read the E-menu card and select the food from the e-menu card to order food online. Once the customer selects the required food item the chef will be able to see the results on the screen and start processing the food. This application nullifies the need of a waiter or reduces the workload of the waiter. The advantage is that in a crowded restaurant there will be chances that the waiters are overloaded with orders and they are unable to meet the requirements of the customer in a satisfactory manner. Therefore by using this application, the users can directly place the order for food to the chef online. In conclusion an online food ordering system is proposed which is useful in small family run restaurants as well as in

Management System is a web-based technology that aids the restaurant industry in carrying out tasks effectively and efficiently. It aids in managing cash flow for managers. Managers can view analytics data to assess company growth. The manager can control orders and employee schedules by using this system. The full complement is a restaurant management system. It provides access to the Online Order platform, third-party connectors software, and comprehensive CRM solution, which together cover a sizable portion of your restaurant's requirements. They are not the outdated hardware and software sets for restaurants that were previously offered. They are the hottest things around, smooth, manageable, inexpensive, and quick. In the "Online Food Ordering Project," we made every effort to meet all the demands of the restaurant. Because it is straightforward and adaptable, the project is successful. The biggest benefit of my project is that it draws plenty of users because of its simplicity. A novice user may operate it with ease. Any type of restaurant can utilize our software. By automating meal ordering, billing, and inventory control, the restaurant management system assists the restaurant manager in managing the restaurant more successfully and efficiently. The system handles the transaction and stores the data produced. These data will be used to create reports that assist the restaurant manager in making wise business decisions. For example, the manager can decide whether more waiters, delivery men, delivery carts, and cooks are needed based on how many clients will be present during a specific time period. When this project is finished, all security concerns will be resolved. Additionally, a quick and secure authentication process will be used for record maintenance. Because it automatically pulls information about a consumer from the database on subsequent visits, data entry is quick and easy. As a result, our program will undoubtedly succeed in replacing the antiquated manual way of storing secure information. The work plan also specifies the specific

29 front end and back end characteristics of the technology being used in the project. Future project goals and its scope have been elaborated. 5.2 Future Work Each project should pay close attention to future development because it contains the system's most recent features. It lessens software issues and defects. It develops a close relationship with customers based on their comments or preferences. Developer will incorporate certain dynamic elements that are briefly described below into my restaurant management system. Reporting module with real time mechanism. • Modern architecture with smooth transitions. • System for email and mobile confirmation. • Selling Point

Glossary

Web application – an application, often comparable in functionality to a desktop application, which is accessed over a network, often via a web browser, rather than hosted on the user's machine

Navigation tree – an expandable and collapsible structure used to display large amounts of information in an easily navigable manner

Application toolbar – usually located at the very top of the window, the application toolbar provides the user with quick access to many common functions

In-application help – most commonly accessed through the “Help” tab on the application toolbar, in-application help provides the user with a quick reference from directly within the application. This is contrasted with more in-depth offline documentation.

Navigation tree – an expandable and collapsible structure used to display large amounts of information in an easily navigable manner

Destructive reinstallation – this reverts the system back to its initial state, destroying all of the user's data. Generally this is a “when all else fails” scenario.

REFERENCES

- [1] J. BLOCH, *Effective Java—Programming Language Guide*, Addison-Wesley, Boston, Massachusetts, 2001.
- [2] P. A. BONATTI, C. DUMA, D. OLMEDILLA, AND N. SHAHMEHRI, *An integration of reputation-based and policy-based trust management*, in Proceedings of the SemanticWeb Policy Workshop, Galway, Ireland, 2005.
- [3] CARNEGIE MELLON UNIVERSITY, *Online auction fraud: Data mining software fingers both perpetrators and accomplices*. Science Daily, <http://www.sciencedaily.com/releases/2006/12/061205143326.htm>, accessed May 2011, Dec. 5, 2006.
- [4] E. FREEMAN AND E. FREEMAN, *Head First HTML with CSS & XHTML*, O'Reilly Media, Sebastopol, California, 2005.
- [5] GAMPASOFT, *Quality assurance*, Gampasoft.com, <http://www.gampasoft.com/quality.html>, accessed May 2011, n.d.
- [6] C. S. HORTSMANN AND G. CORNEL, *Core Java 2, Volume I – Fundamentals*, PrenticeHall, Upper Saddle River, New Jersey, 2004.
- [7] C. S. HORTSMANN AND G. CORNELL, *Core Java Volume II—Advanced Features*, Prentice Hall, Upper Saddle River, New Jersey, 2008.
- [8] F. MARCHIONI, *JBoss AS 5 Development*, Packt Publishing, Birmingham, England, 2009.
- [9] H. MARTY, *More Servlets and JavaServer Pages*, Pearson Education, Upper Saddle River, New Jersey, 2001.
- [10] N. RICHARDS AND S. GRIFFITH, *JBoss—A Developer's Notebook*, O'Reilly Media, Sebastopol, California, 2005.
- [11] D. THAU, *The Book of JavaScript 2nd Edition: A Practical Guide to Interactive Web Pages*, No Starch Press, San Francisco, California, 2006.
- [12] J. TREVATHAN AND W. READ, *Detecting collusive shill bidding*, in Proceedings of the Fourth International Conference on Information Technology: New Generations, Las Vegas, Nevada, 2007.
- [13] W. WANG, H. ZOLTÁN, AND A. B. WHINSTON, *Shill bidding in multi-round online*

Index

database, 8, 11, 12, 13, 25

Error Recovery, 34

Help System, 32

Installation, 34

Managing Categories, 28

Managing Foods, 30

Managing Orders, 30

Managing Vendors, 28

menu management, 8, 10, 11,
12, 13, 15

mobile applications, 13

order retrieval system, 8, 11, 15

Placing an Order, 31

requirements, 1, 12

Selecting Food Items, 31

Signing In, 31, 33

Signing Up, 31

system model, 8, 13

System Model, 8, 15

User interaction, 12

web ordering system, 8, 9, 10,
11, 12, 15