

# Online Food Ordering System

This Project report was presented to the National University in partial fulfillment of the requirement for the degree of B. Sc. (Hon's) in Computer Science & Engineering.

## **Submitted By**

Oabdel Hasan  
Registration No:  
16502000589  
Session: 2016-17

MD.Taijul Islam  
Registration  
No:16502000590  
Session: 2016-17

## **Supervised By**

S.M. Mahmudul Hasan  
Lecturer  
Department of Computer Science and Engineering  
New Model Degree College

Date of Submission:  
24September, 2022

National University, Bangladesh



# **APPROVAL**

The Project “Online Food Ordering System”, Oabdel Hasan Semanto Registration No: 16502000589 and Taijul islam Registration No: 16502000590 to the Department of Computer Science and Engineering, New Model Degree College, Dhaka, Bangladesh has been accepted as satisfactory for the partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science and Engineering under National University and approved as to its style and contents.

---

## **Supervisor**

S.M. Mahmudul Hasan  
Lecturer  
Department of Computer Science and Engineering  
New Model Degree College, Dhaka

# DECLARATION

We do hereby solemnly declare that we have completed partial work of this project under the supervision of S.M. Mahmudul Hasan, Lecturer of Computer Science and Engineering, New Model Degree College, affiliated with the National University of Bangladesh. We also declare that neither this project nor any part of this has been submitted elsewhere for the award of any degree.

We further undertake to indemnify the University against any loss or damage arising from breach to the forgoing obligation.

## **Submitted By**

---

Oabdel Hasan  
Registration No:16502000589  
Department of CSE  
New Model Degree College, Dhaka  
Session: 2016-2017

---

MD.Taijul Islam  
Registration No:16502000590  
Department of CSE  
New Model Degree College, Dhaka

# CERTIFICATE

This is to certify that the work report on “Online Food Ordering System” is the bonafide record of work done by Oabdel Hasan Semanto (Reg No: 16502000589) and MD.Taijul Islam (Reg No: 16502000590) for partial fulfillment of the requirement of the degree of B.Sc in Computer Science and Engineering from National University .

The work report has been carried out under my guidance and is a record of the successful work. I wish their success in future

## **Supervised By**

---

**S.M. Mahmudul Hasan**

Lecturer

Department of Computer Science and Engineering

New Model Degree College, Dhaka

# **ACKNOWLEDGMENT**

In this very special moment, first and foremost we would like to express our heartiest gratitude to the almighty God for allowing us to accomplish this B.Sc in CSE Degree successfully. In achieving the gigantic goal, we have gone through the interactions with and help from other people, and would like to extend our deepest appreciation to those who have contributed to this dissertation itself in an essential way. We would like to express our heartfelt thanks to all of you for being with us with immense support and care and to make this work successful.

# **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 trouble-shooting tips have been included for all three components to give the reader a clear idea of intended typical use cases for the system.





<b>APPROVAL</b>	<b>i</b>
<b>DECLARATION</b>	<b>ii</b>
<b>CERTIFICATE</b>	<b>iii</b>
<b>ACKNOWLEDGMENT</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>

## **Table of Contents**

### **CHAPTER-1 Introduction And Feasibility**

1.1 Problem Statement	1
1.2 Objective	1
1.3 Feasibility	2

### **CHAPTER-2 Requirements Specification**

2.1 System Model	3
2.2 Functional Requirements	3
2.3 The Web Ordering System	3
2.4 Menu System	3
2.5 Order Retrieval System	4
2.6 Web Ordering System	4
2.7 Menu Management System	5
2.8 Order Retrieval System	5
2.9 Non-functional Requirements	6
2.10 System Evolution	7

### **CHAPTER-3 System Design**

3.1 ER Diagram:	8
3.2 Schema Diagram:	9
3.3 Level 1: The Database & the 3 Components	10
3.4 Level 2: Web Ordering System Components	11
3.4.1 Level 3: The Login Form	12
3.4.2 Level 3: The Main Menu	12
3.4.3 Level 3: The Account Management Form	12
3.4.4 Level 3: The Order Form	12
3.4.5 Level 3: The Shopping Cart	12
3.4.6 Level 3: The Checkout Form	12
3.5 Level 2: Menu Management System Components	12

3.5.1 Level 3: The Navigation Tree	12
3.5.2 Level 3: The Forms	13
3.6 Level 2: Order Retrieval System Components	13
3.6.1 Level 3: Summary Panel	13
3.6.2 Level 3: Order Detail Panel	13
3.7 User Interface Design	13
3.8 Help System Design	13
<b>CHAPTER-4 Testing Design</b>	
4.1 Testing Phases	16
4.2 Database	16
4.3 Web Ordering System	16
4.4 Menu Management System	16
4.5 Order Retrieval System	16
4.6 Requirements Traceability	16
4.7 Testing Schedule	17
4.8 Recording Procedures	17
4.9 Hardware and Software Requirements	17
<b>CHAPTER-5 User Interface</b>	
5.1 Interface Of Add Admin	19
5.2 Interface Of Admin Dashboard	19
5.3 Interface Of Admin login	20
5.4 Interface Of Insert Product	20
5.5 Interface Of View Order	21
5.6 Interface Of Customer Registration	21
5.7 Interface Of View Admin	22
5.8 Interface Of View Product	
5.9 Interface Of View user	22
5.10 Interface Of Cart	23
5.11 Interface Of Contact	23
5.12 Interface Of shop	24
5.13 Interface Of Homepage	24
<b>CHAPTER-6 Conclusion</b>	
6.1 Conculation	26
<b>Reference</b>	

## **List Of Figures**

2.1 System Model	3
3.1 ER Diagram:	8
3.2 Schema Diagram:	9
3.3 Level 1: The Database & the 3 Components	10
3.4 Level 2: Web Ordering System Components	
3.5 Level 2: Menu Management System Components	12
4.1 Testing Phases	16

# **CHAPTER- 1**

## **Introduction and Feasibility**

## **1.1 Problem Statement:**

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.

## **1.2 Object**

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 on the webpage, it is placed into the database and then retrieved, in pretty much real-time, by a desktop application on the restaurant's end. Within this application, all items in the order are displayed, along with their corresponding options and delivery details, in a concise and easy to read manner. This allows restaurant 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.

## **1.3 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.

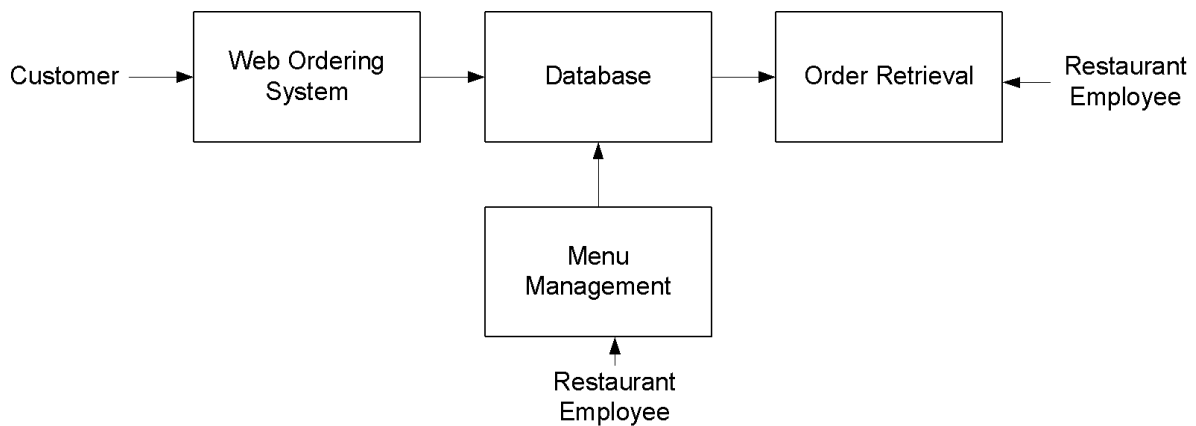
In addition to time, a second factor influencing feasibility is resources, which also should not be a concern here. The online ordering system is structured like a fairly standard web application, and as such requires no special hardware and only basic software, namely web and database servers, to function 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**

## 2.1 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.



**Fig: System Model**



## 2.2 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 provides 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 with and these functions are outlined below, grouped by component.

## 2.3 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.

## 2.4 Menu 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.

## **2.5 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.

## **2.6 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.

## **2.7 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.

## **2.8 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.

## **2.9 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.

## **2.10 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

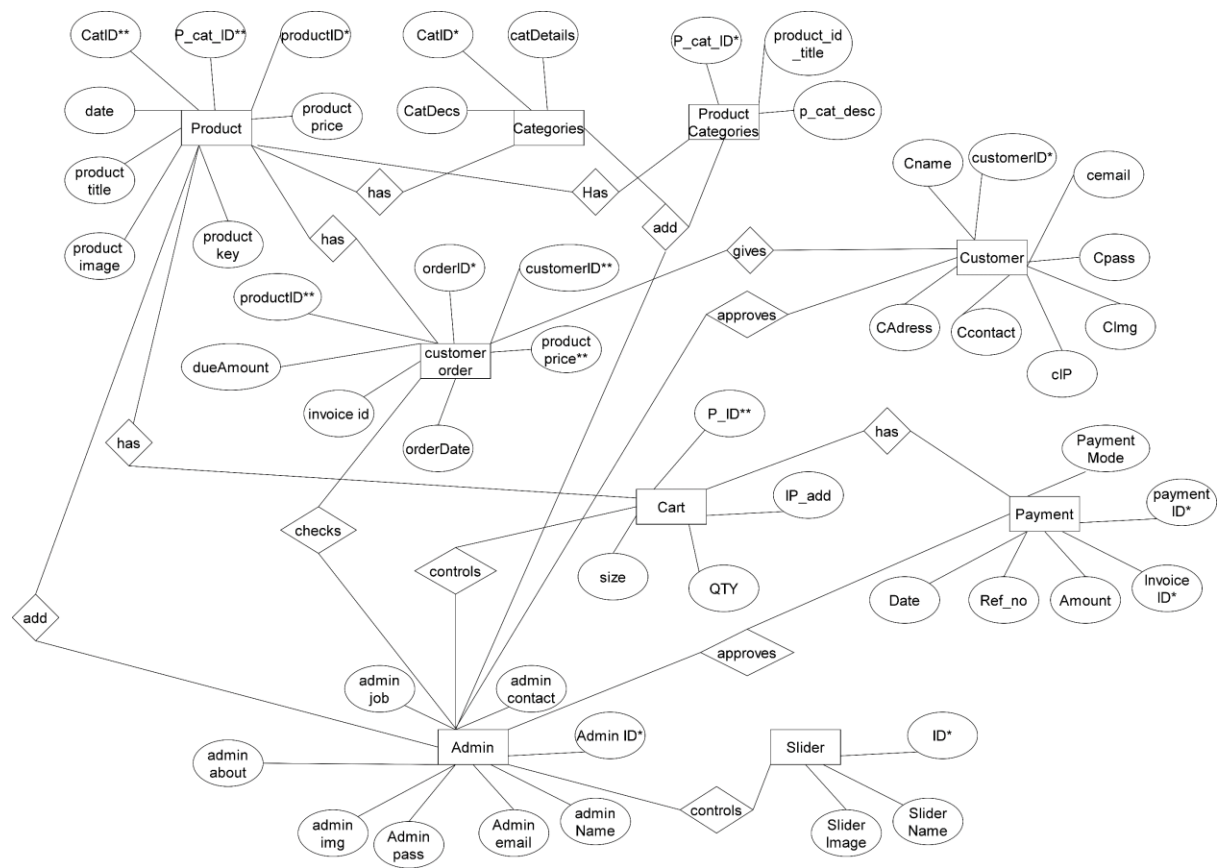
booming popularity of mobile applications, I would really like to make the web interface available as a phone application as well. Also it may make sense to at some point migrate the menu management and order retrieval systems to web, or even mobile, applications as well, as some users may prefer to 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**

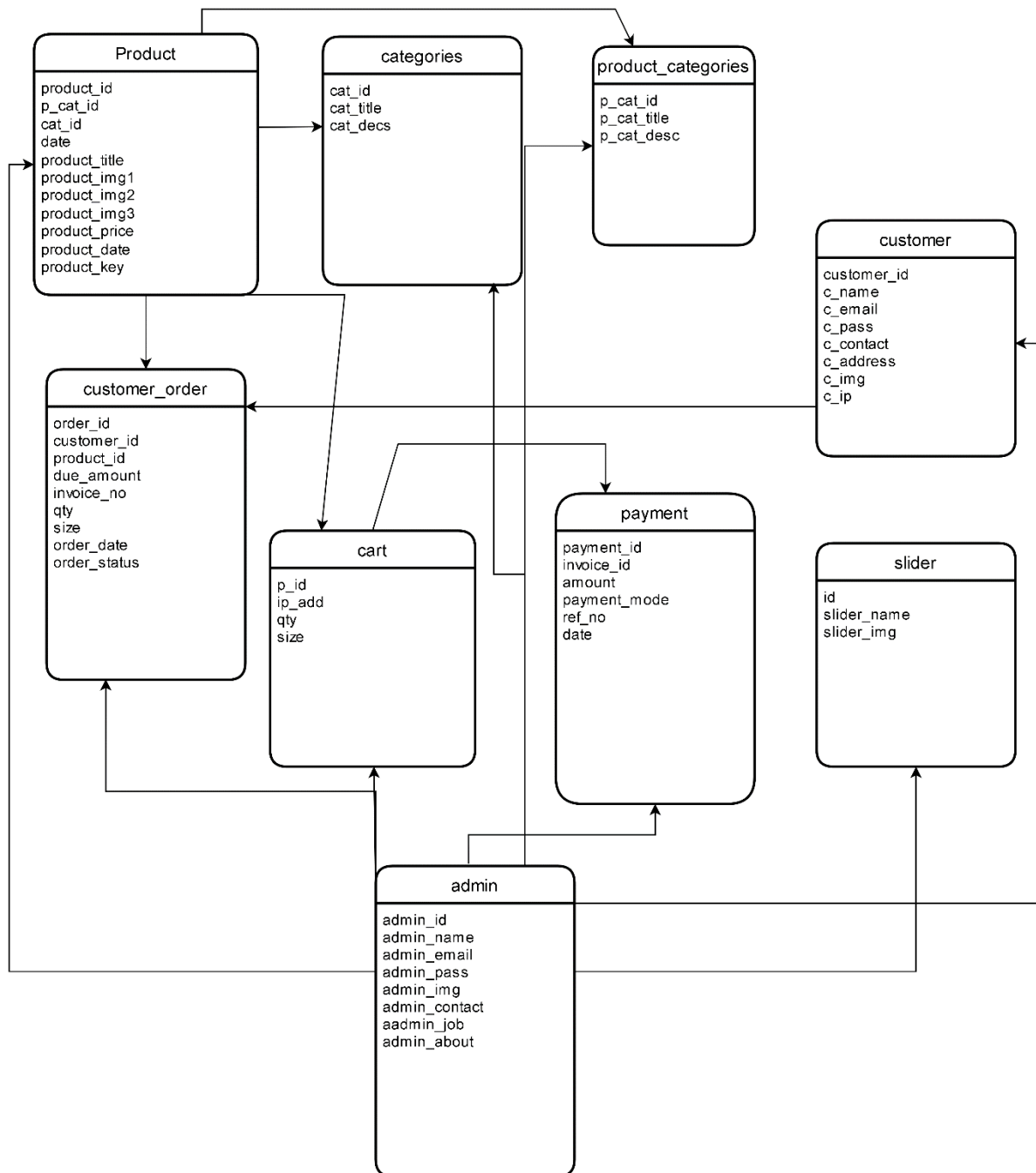
## **System Design**

### 3.1 ER Diagram:



### Fig:ER Diagram

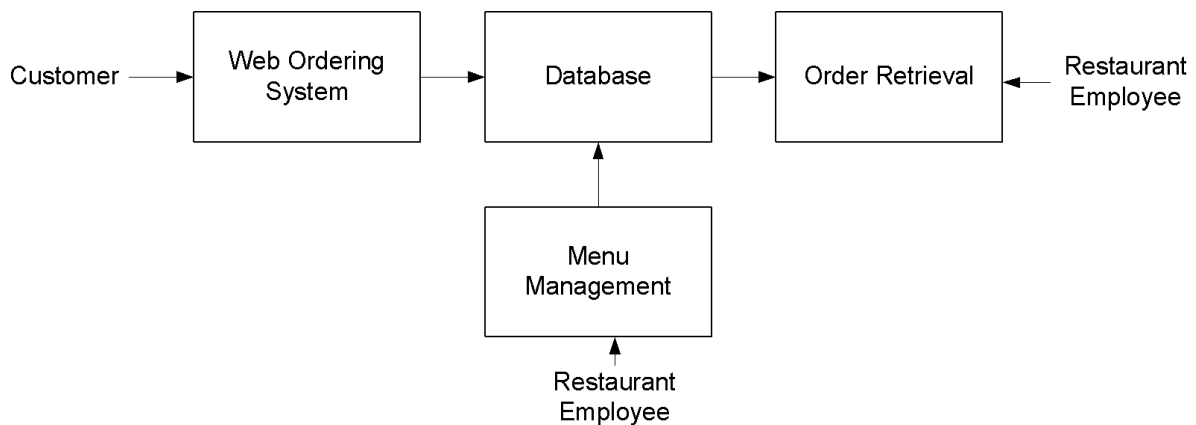
### 3.2 Schema Diagram:



**Fig:Schema Diagram**

### 3.3 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.

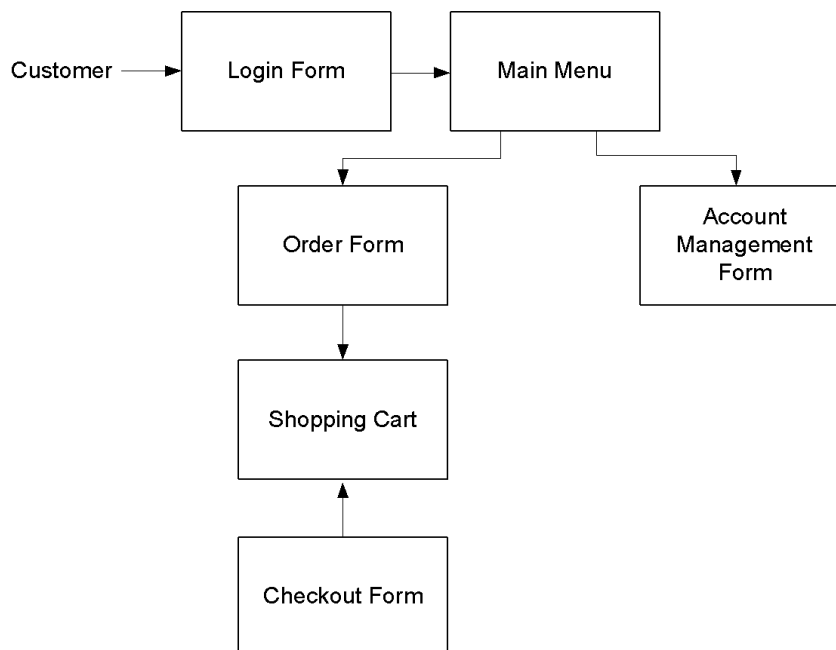


**Fig: The Database & The 3 Components**



### 3.4 Level 2: Web Ordering System Components

The web ordering system has 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.



**Fig:Web Ordering System Component**

### **3.4.1 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.

### **3.4.2 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.

### **3.4.3 Level 3: The Account Management Form**

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

### **3.4.4 Level 3: The Order Form**

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

### **3.4.5 Level 3: The Shopping Cart**

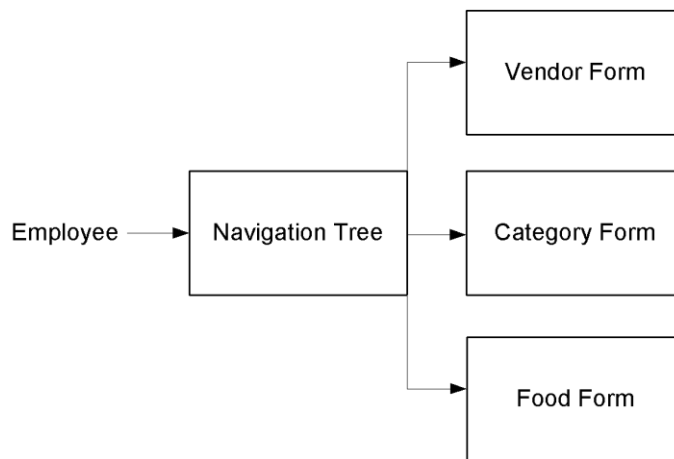
The shopping cart performs much like a shopping cart in any other application. After an item is added to the order, it is displayed, along with its price, in the shopping cart. The shopping cart also keeps a running total of the current price of the whole order. By clicking on an item in the shopping 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.

### **3.4.6 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.).

## **3.5 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.



**Fig:Menu Management System**

### 3.5.1 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.

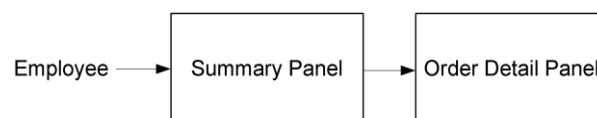
### 3.5.2 Level 3: The Forms

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.

## 3.6 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.

### 3.6.1 Level 3: Summary Panel



**Fig: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.

### **3.6.2 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.

## **3.7 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 checkboxes. 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 of text fields and corresponding labels, along with save and discard buttons for each form.

## **3.8 Help System Design**

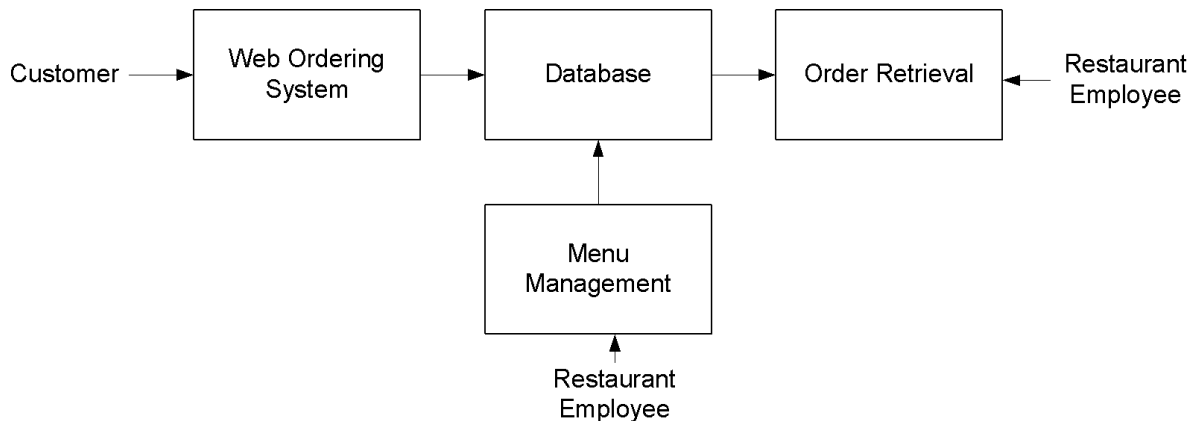
Due to the form-based nature of the applications, the design of the help systems will be minimal. In both the desktop and web applications, it will be accessed from the application's main menu and will open in a new window. Modeled after the typical help system design, it will be both searchable and 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**

## 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.



**Fig: Testing Phases**

## 4.2 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.

## 4.3 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.

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 be very important to see if there is anything I have not accounted for.

## **4.4 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.

## **4.5 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.

## **4.6 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 designed to contain only the absolutely necessary functionality, the testing of the different functions should flow smoothly into one another.

## **4.7 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 8<sup>th</sup>), 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 15<sup>th</sup>, 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.

## **4.8 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.

## **4.9 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 Interface**

## 5.1 Interface Of Add Admin

Dashboard/Add Admin

Insert Categories

user Name

Email

user Password

user Job

user Contact

User About

user Image

Choose file

No file chosen

Submit

2021 10/11

template

## 5.2 Interface Of Admin Dashboard

Admin Panel Dashboard Product Product Categories Categories Slider View Customer View Order View Payments Users

Dashboard

DASHBOARD

3 Product

View Details

2 Customer

View Details

3 Product Categories

View Details

4 Order

View Details

New Order

Order NO:	Customer Email:	Invoice No:	Total	Date	Status
1	tajulmohim@gmail.com	1103501069	1	2022-07-18 20:41:35	Complete
2	tajulmohim@gmail.com	1075650465	1	2022-07-16 20:07:38	Complete
3	simanto@gamil.com	594033752	1	2022-07-16 19:24:03	Complete
4	tajulmohim@gmail.com	958005037	3	2022-07-16 17:19:14	Complete

View All Order

Admin Name:tajul

Admin Job: worner

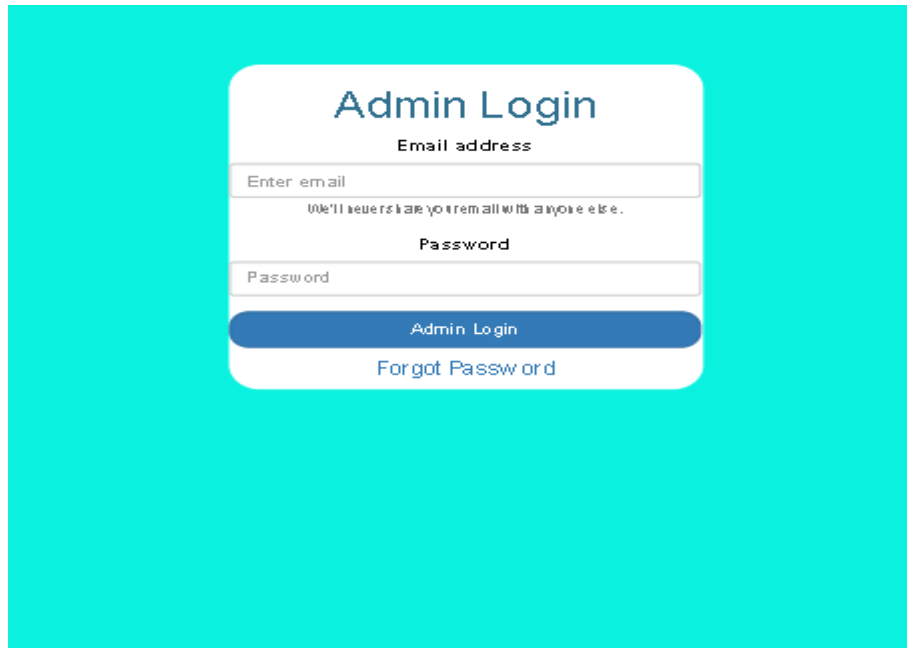
email:tajulmohim@gmail.com

contact:01684859036

About

I am a devoloper

## 5.3 Interface Of Admin login

The image shows a web interface for Admin Login. It features a white rounded rectangle centered on a solid blue background. Inside the rectangle, the title "Admin Login" is at the top in a large, dark blue font. Below it, the label "Email address" is followed by a text input field containing the placeholder "Enter email". A small note below the email field reads "We'll never share your email with anyone else.". The next section is labeled "Password" and contains a text input field with the placeholder "Password". At the bottom of the form, there is a blue button labeled "Admin Login" and a link labeled "Forgot Password" in blue text below it.

Admin Login

Email address

Enter email

We'll never share your email with anyone else.

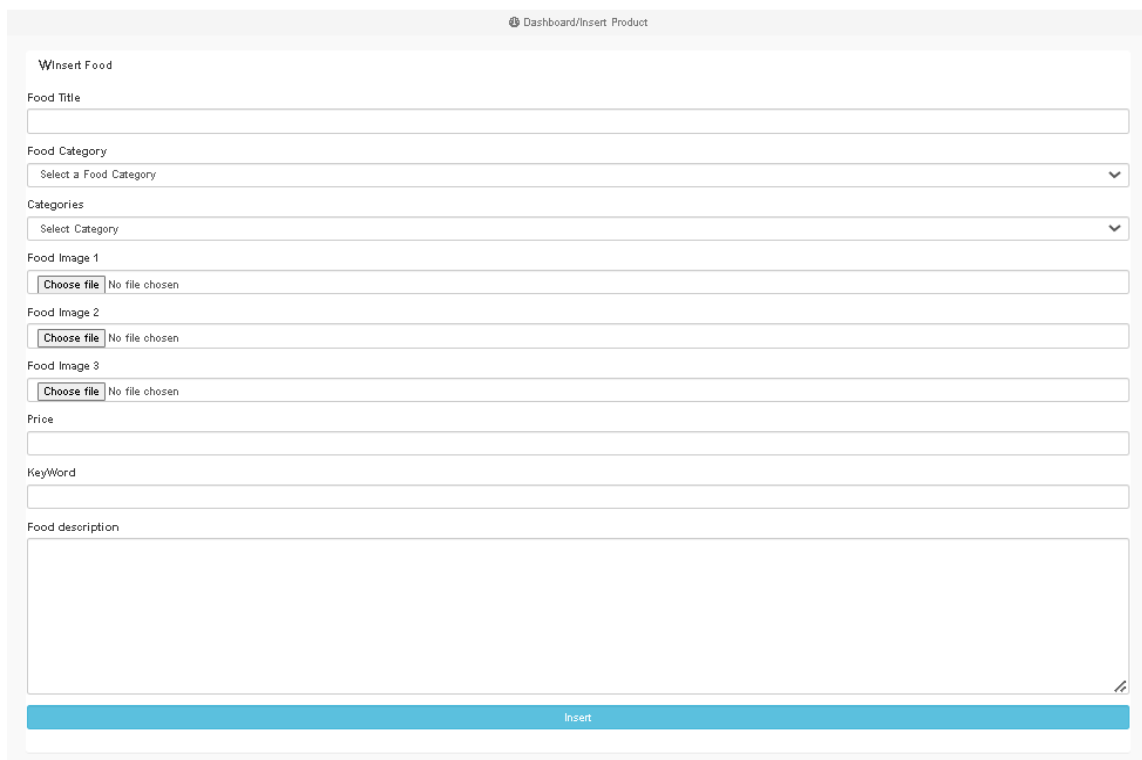
Password

Password

Admin Login

Forgot Password

## 5.4 Interface Of Insert Product

The image shows a web interface for inserting a new product. At the top, there is a breadcrumb trail "Dashboard/Insert Product". The main form is titled "Winsert Food" and contains several input fields: "Food Title" (a text box), "Food Category" (a dropdown menu with "Select a Food Category" as the selected option), "Categories" (a dropdown menu with "Select Category" as the selected option), "Food Image 1", "Food Image 2", and "Food Image 3" (each with a "Choose file" button and "No file chosen" text), "Price" (a text box), "KeyWord" (a text box), and "Food description" (a large text area). At the bottom of the form is a blue button labeled "Insert".

Dashboard/Insert Product

Winsert Food

Food Title

Food Category

Select a Food Category

Categories

Select Category

Food Image 1

Choose file No file chosen

Food Image 2

Choose file No file chosen

Food Image 3

Choose file No file chosen

Price

KeyWord

Food description

Insert

## 5.5 Interface Of View Order

Dashboard/View Order									
View Order									
Order Id	Customer Email	Invoice No:	Product Title	Product Qty	Product Size	Order Date	Total Amount	Order Status	Delete Order
1	taijulmohim@gmail.com	958005037	pizza	3	Large	2022-07-16 17:19:14	900	Complete	<a href="#">Delete</a>
2	simanto@gamil.com	594033752	Burger	1	Medium	2022-07-16 19:24:03	200	Complete	<a href="#">Delete</a>
3	taijulmohim@gmail.com	1075650465	Breakfast	1	Medium	2022-07-16 20:07:38	130	Complete	<a href="#">Delete</a>
4	taijulmohim@gmail.com	1103501069	Burger	1	Medium	2022-07-18 20:41:35	200	Complete	<a href="#">Delete</a>

## 5.6 Interface Of Customer Registration

WELCOME GUEST
Shopping Cart Total price :0TK,Items:0
Register | My account | Go to cart | Login

wowFood
Home Shop My account Shopping Cart Contact us
0 Items in cart

Home / Registration

Product Categories

pizza

souf

Hot naga

Categories

pizza

Souf

Bangoli Food

### Customer Registration

Customer Name

Customer Email

Customer Password

AREA

Contract Number

Address

Address

[Choose file](#) No file chosen

[Register](#)

ages
shopping cart
contact
top
My Account

Top Product Categories

pizza

Souf

Bangoli Food

Where to find us
friend.com
Sohawal
kddkkskdkks
+01674555554
Goto contact us page


Get the news
new
Subscribe

sers
sign
register

Stay IN touch

©2021 Bili
template










## 5.7 Interface Of View Admin

Dashboard/New user								
View Admin								
Admin Id	Admin Image	Admin Name	Admin Email	Admin Contact	Admin Job	Admin About	Delete Payment	Admin Edit
1		taijul	taijulmohim@gmail.com	01684859036	woner	I am a devoloper	<button>Delete</button>	<button>Edit</button>

© 2021 taijul

template



## 5.8 Interface Of View Product

Dashboard/View Product							
View Product							
Product Id	Product Title	Product Image	Produc Price	Product keyword	Product Date	Product Delete	Product Edit
1	pizza	1. 	300	piza	2022-06-15 19:54:50	<button>Delete</button>	<button>Edit</button>
		2. 					
		3. 					
2	Breakfast	1. 	130	known	2022-07-22 11:49:33	<button>Delete</button>	<button>Edit</button>
		2. 					
		3. 					
3	Burger	1. 	200	burger	2022-06-19 18:52:03	<button>Delete</button>	<button>Edit</button>
		2. 					
		3. 					

## 5.9 Interface Of View user

Dashboard/View Customer

View Customer

Customer Id	Customer Name	Customer Email	Customer Image	Customer Area	Customer Address	Contact	DELETE
1	taijul	taijulmohim@gmail.com		01684859036	dhanmondi 27	01684859036	<button>Delete</button>
2	simanto	simanto@gamil.com		01684859036	dhanmondi 27	01684859036	<button>Delete</button>

© 2021 taiju

template

## 5.10 Interface Of Cart

WELCOME TO BEST

Shopping Cart Total price :0Tk,Items:0

Register | My account | Go to cart | Login

wowFood

Home

Shop

My account

Shopping Cart

Contact us

0 Items in cart

Home / Shopping cart

### Shopping Cart

only you have 0 items in your cart


Product	Quantity	Unit price	Size	Delete	Sub total
total					0tk

Continue Shopping


Update cart

Proceed to checkout


You also like those Food



pizza  
300 tk



Breakfast  
130 tk



Burger  
200 tk

Order Summary

Shipping and additional costs are calculated based on the values you have entered.

Order Subtotal	0tk
Shipping and handling	0tk
Tax	0tk
Total	0tk

Pages

[Shopping cart](#)  
[Contact](#)  
[Shop](#)  
[My Account](#)

Top Product Categories

[pizza](#)  
[Soft](#)  
[Bakery Food](#)

Where to find us

[friend.com](#)  
[Sokawal](#)  
[krdkdkrkkks](#)  
[+01674595554](#) [Go to contact us page](#)

Get the news

Stay IN touch







© 2021 km

template

## 5.11 Interface Of Contact

WELCOME TO WWW.FOODS

Shopping Cart Total price :0TK, Items :0

Register | My account | Go to Cart | Login

WowFood

Home Shop My account Shopping Cart Contact us

0 Items in cart

Home / Contact Us

Product Categories

[pizza](#)
[solt](#)
[Hot tags](#)

Categories

[pizza](#)
[solt](#)
[Bangla Food](#)

Contact Us

If you have any questions, please feel free to contact us, our customer service is working for you 24/7.

Name

Email

Subject

Sub ject

Send Message

Pages

[Shopping cart](#)
[Contact](#)
[Shop](#)
[My Account](#)

Top Product Categories

[pizza](#)
[solt](#)
[Bangla Food](#)

Where to find us

[mendo.com](#)
[soltana](#)
[kashakasha](#)
[+01674555554](#)
[Go to contact us page](#)

Get the news

Subscribe

Stay IN touch

[Facebook](#)
[Twitter](#)
[Instagram](#)
[LinkedIn](#)

© 2022 Food

template

## 5.12 Interface Of shop

WELCOME TO WWW.FOODS

Shopping Cart Total price :0TK, Items :0

Register | My account | Go to Cart | Login

WowFood

Home Shop My account Shopping Cart Contact us

0 Items in cart

Home / Shop


Product Categories

[pizza](#)
[solt](#)
[Hot tags](#)

Categories

[pizza](#)
[solt](#)
[Bangla Food](#)

Shop




Burger

Price:200 tk

View Details

Add to cart




Breakfast

Price:150 tk

View Details

Add to cart



pizza

Price:300 tk

View Details

Add to cart

First Page

1

Last Page

Pages

[Shopping cart](#)
[Contact](#)
[Shop](#)
[My Account](#)

Top Product Categories

[pizza](#)
[solt](#)
[Bangla Food](#)

Where to find us

[mendo.com](#)
[soltana](#)
[kashakasha](#)
[+01674555554](#)
[Go to contact us page](#)

Get the news

Subscribe

Stay IN touch

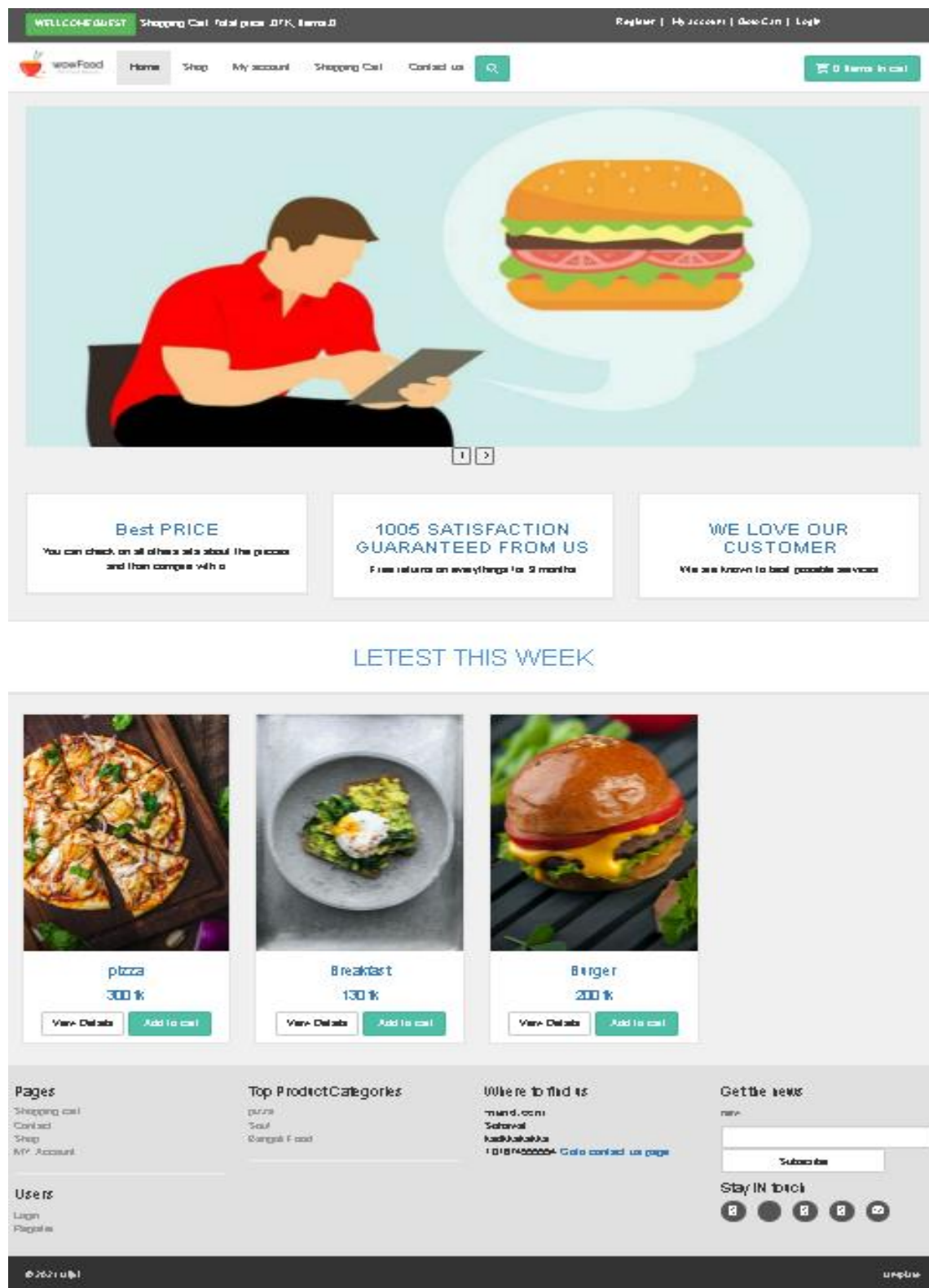
[Facebook](#)
[Twitter](#)
[Instagram](#)
[LinkedIn](#)

© 2022 Food

template



### 5.13 Interface Of Homepage



## **CHAPTER-6**

### **Conclusion**

## **6.1 Conclusion**

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 places like college cafeteria, etc. This project can later be expanded on a larger scale. It is developed for restaurants to simplify their routine managerial and operational task and to improve the dining experience of the clients. This also helps the restaurant owners develop healthy customer relationships by providing reasonably good services. The system also enables the restaurant to know the items available in real time and make changes to their food and beverage inventory based on the orders placed and the orders completed.

# Reference

1. Downloads | Oracle. 2015. Downloads | Oracle. [ONLINE] Available at:  
<https://www.oracle.com/downloads/index.html>
2. PHP Tutorials and Online Training. 2015. [ONLINE] Available at:  
<http://www.oracle.com/technetwork/python/index-jsp-135888.html>
3. SQL Tutorial. 2015. [ONLINE] Available at:  
<http://www.w3schools.com/sql/default.asp>
4. MySQL Downloads. 2015. [ONLINE] Available at:  
<http://www.mysql.com/downloads/>
5. stackoverflow. [ONLINE] Available at: <https://stackoverflow.org/download-70.cgi>
6. Github .Available at : <https://github.com/>
7. Academiaedu for patterns. Available at :  
[https://pvgiomnsk.com/wp-content/uploads/pdf/aicte/inventory\\_management\\_technique](https://pvgiomnsk.com/wp-content/uploads/pdf/aicte/inventory_management_technique)