



KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

**KIIT SCHOOL OF COMPUTER APPLICATIONS
2023**

Online Food Ordering System(OFOS)

Project submitted in partial fulfillment for the award of the degree of

Bachelor's Of Computer Science (BSC CS)

Submitted By:

Abhishek Kumar(2047001)

Under Guidance of

Prof. Kamakhya Narain Singh

CERTIFICATE OF ORIGINALITY

This is to certify that the project report entitled “**Online Food Ordering System (OFOS)**” Submitted to School of Computer Application, KIIT Deemed to be University in partial fulfillment of then requirement for the award of the degree of Bachelor’s Of Computer Science(BSC CS), is an authentic and original work carried out by Mr. Abhishek Kumar under my guidance.

The matter embodied in this project is genuine work done by the student and has not been submitted whether to this University or to any other University/Institute for the fulfillment of the requirements of any course of study.

Signature of the Student:

Signature of the Guide:

Date:

Date:

Name:

Designation:

School of Computer ApplicationsKIIT

Deemed to be University Certificate

This is to certify that the project work entitled **Online Food Ordering System (OFOS)** submitted by **Abhishek Kumar** bearing roll no. **2047001** is an authentic and original work.

Signature

(Internal Examiner)

Date:

Signature

(External Examiner)

Date:

DECLARATION

I, **Abhishek Kumar**, with roll no. **2047001** do hereby declare that the project report entitled “**Online Food Ordering System (OFOS)**” submitted to School of Computer Applications, KIIT Deemed to be University, Bhubaneswar for the award of the degree of Bachelor’s Of Computer Science (BSC CS), is an authentic and original work carried out by me from 1st March 2023 to 29th April 2023 at KIIT Deemed to be University under the guidance of **Prof. Kamakhya Narain Singh**.

Signature of the student

Date:

Acknowledgement

This satisfaction which accompanies the successful completion of any task is incomplete without the mention of those persons whose hands are behind the success. Because the success is the epitome of hard work, prevention, zeal, determination and the most encouraging guidance and advice serving as beacon light and crowing our effort with success.

I am grateful to Dr. Veena Goswami, the Dean of SCA, KIIT Deemed to be University for her endless support and kind cooperation for completion of this project. I am also thankful to Prof. Kamakhya Narain Singh, Prof. Rabindra Kumar Barik , Prof. Sumit Tatarve and other faculty members of KIIT Deemed to be University who have constantly helped and guided us despite being so far away from us.

Table of Contents

Chapter 1: Justification & Feasibility.....	9
Introduction.....	9-10
Justification.....	11-12
Feasibility.....	13
Chapter2: Requirements Specification.....	14
System Model.....	14
Functional Requirements.....	15
The Web Ordering System.....	15-16
Menu Management System.....	17
Order Retrieval Systems.....	18
User Interface Specifications.....	18
Web Ordering System.....	18
Menu Management System.....	19
Order Retrieval System.....	19-20
Non Functional Requirements.....	20-21
System Evolution.....	21
Chapter :3.....	22
Level 1: The Database and the three Components.....	22
Level 2: Web Ordering System Components.....	22-23
Level 3: The Login Form.....	23
Level 4: The Account Management Form.....	23
Level 5: The order Form.....	23
Level 6:The shopping Cart.....	24
Level 7: The checkout form.....	24

Level 8: Menu Management system components.....	24
Level 9: The Navigation tree.....	24
Level 10: The Forms.....	25
Level 11: Order retrieval system components.....	25
Level 12: Summary Panel.....	25
Level 13: Order Detail Panel.....	26
User Interface Design.....	26
Help System Design.....	27
Chapter 4: Testing Design.....	28
Testing.....	28
Phases.....	28
DFD.....	28-30
ER Diagram.....	31
Level 1: Database.....	31
Level 2: Web Ordering System.....	31
Level 3: Menu Management System.....	32
Requirements Traceability.....	32-33
Testing Schedule.....	34-35
Recording Procedures.....	35
Hardware and Software Requirements.....	36
System Requirements.....	36
Chapter 5: User Manual.....	37
Using the Desktop Application.....	37-40
Using the Web Ordering System.....	40-41
Using the Help System.....	42
Services	42

Error Recovery.....	43
Installation.....	43-44

Chapter 1: Justification & Feasibility

Introduction

Nowadays, digital business platforms are very popular and save us much effort and time in our daily life. E-commerce companies such as Amazon and Ebay could deliver goods to customers very efficiently. On the one hand, customers could select goods and place orders online without visiting the shop, which is usually time-consuming. Besides, they do not need to carry the goods to home. Instead, the shop would deliver the goods and save customers' efforts. On the other hand, using digital business platforms could make it more convenient for shop owners to manage orders, collect and analyse data and provide better service. In the catering industry, the demand of combining the convenience of digital business with their traditional delivery service is increasingly growing. Unlike common e-commerce companies, the restaurants usually could deliver food in less than half an hour and actually saves customers' time when compared to visiting the restaurants. In order to stand out in the digital business trend of catering industry and provide more satisfying service, we designed this Pizza Hot project. Customers usually expect fast delivery and food in good condition to eat. So in order to adapt to customers' expectation and earn more profits for restaurant owners, we improved the traditional digital business platform. In addition to the general functionalities of e-commerce platform, we developed a hardware system which is intended to be put in the delivery box. After the restaurant receives customers' orders and the dishes are ready to deliver, the delivery man put the portable system in the delivery box. The system could record the temperature in the box and the time of delivery. After the dishes arrives, it will calculate a suggested tip for customers based on temperature, time and distance the delivery man covered. If the food's temperature is suitable, the tips could be higher and otherwise the system gives a discount. In this way, we could realize a win-win relationship between customer and restaurants. Customers can expect fast delivery and hot food. Restaurants can make more money by providing good service, which also helps them stands out among competitors. Delivery staffs will be provided information for more efficient operations such as turn by turn navigation. The online food ordering system provides convenience for the customers. It

overcomes the disadvantages of the traditional queuing system. This system increases the takeaway of foods than visitors. Therefore, this system enhances the speed and standardization of taking the order from the customer. It provides a better communication platform. The user's details are noted electronically. The online food ordering system set up menu online and the customers easily places the order with a simple mouse click. Also with a food menu online you can easily track the orders, maintain customer's database and improve your food delivery service. This system allows the user to select the desired food items from the displayed menu. The user orders the food items. The payment can be made online or pay-on-delivery system. The user's details are maintained confidential because it maintains a separate account for each user. An id and password is provided for each user. Therefore it provides a more secured ordering. It is known globally that, in today's market, it is extremely difficult to start a new small-scale business and live-through the competition from the well-established and settled owners. In fast paced time of today, when everyone is squeezed for time, the majority of people are finicky when it comes to placing a food order. The customers of today are not only attracted because placing an order online is very convenient but also because they have visibility into the items offered, price and extremely simplified navigation for the order. Online food ordering system that I am proposing here, greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

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 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.

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 diagramed 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.

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 is 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

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 straight forward. Although written using Google Web , 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 Apache, and particularly the free XAMPP distribution.

All of the application data is stored in a MySQL database, and therefore a MySQL server must also be installed on the host computer. As with Apache, 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

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. ↑

Level2: 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 4: 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 5: The Account Management Form

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

Level 6: The Order Form

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

Level 7: 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 8: 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 9: 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.

Level 10: 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 11: 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 12: 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.

Level 13: 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 14: 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

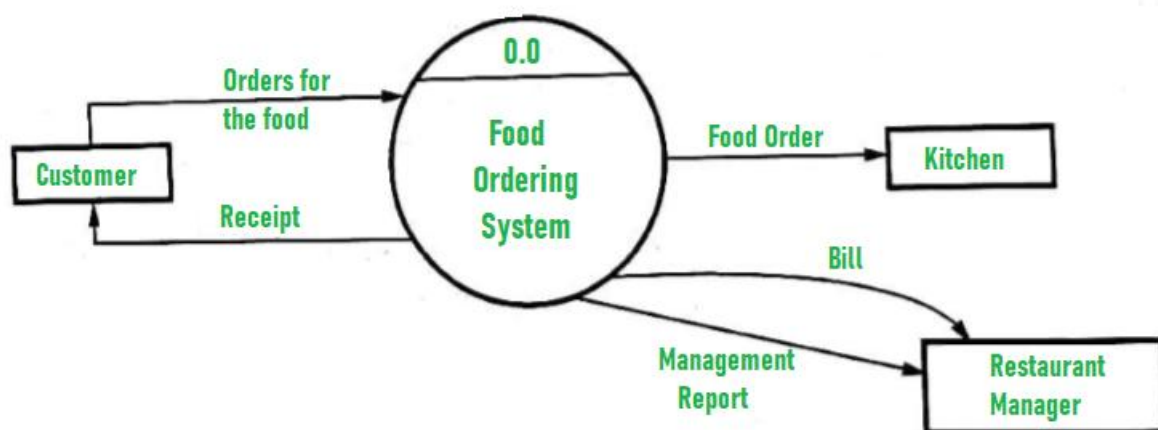
Testing Phases

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.

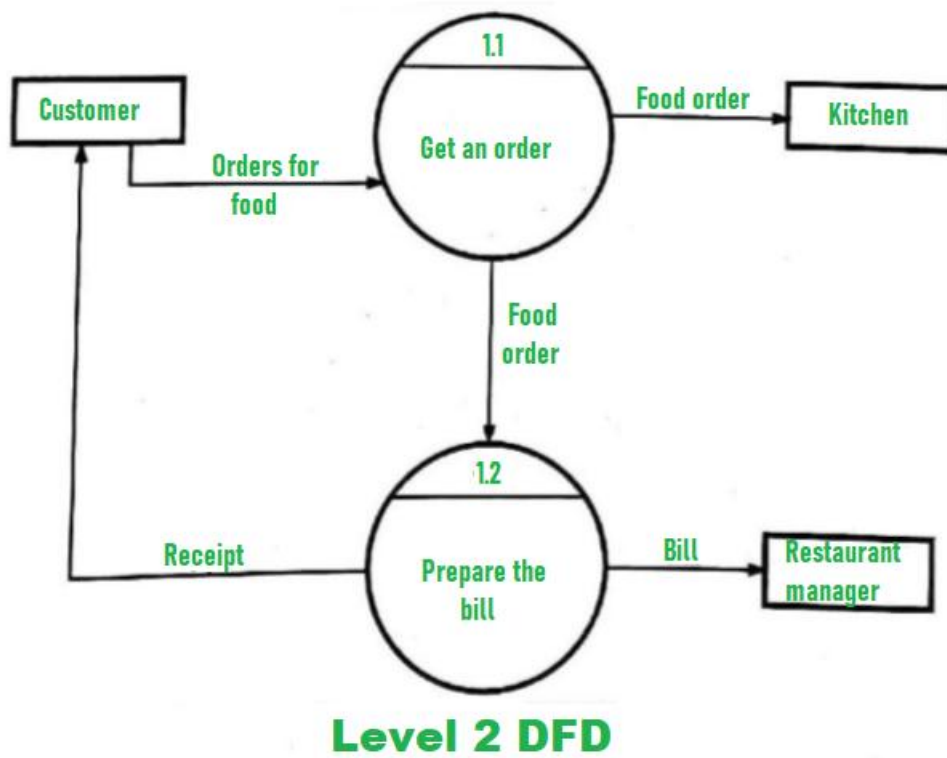
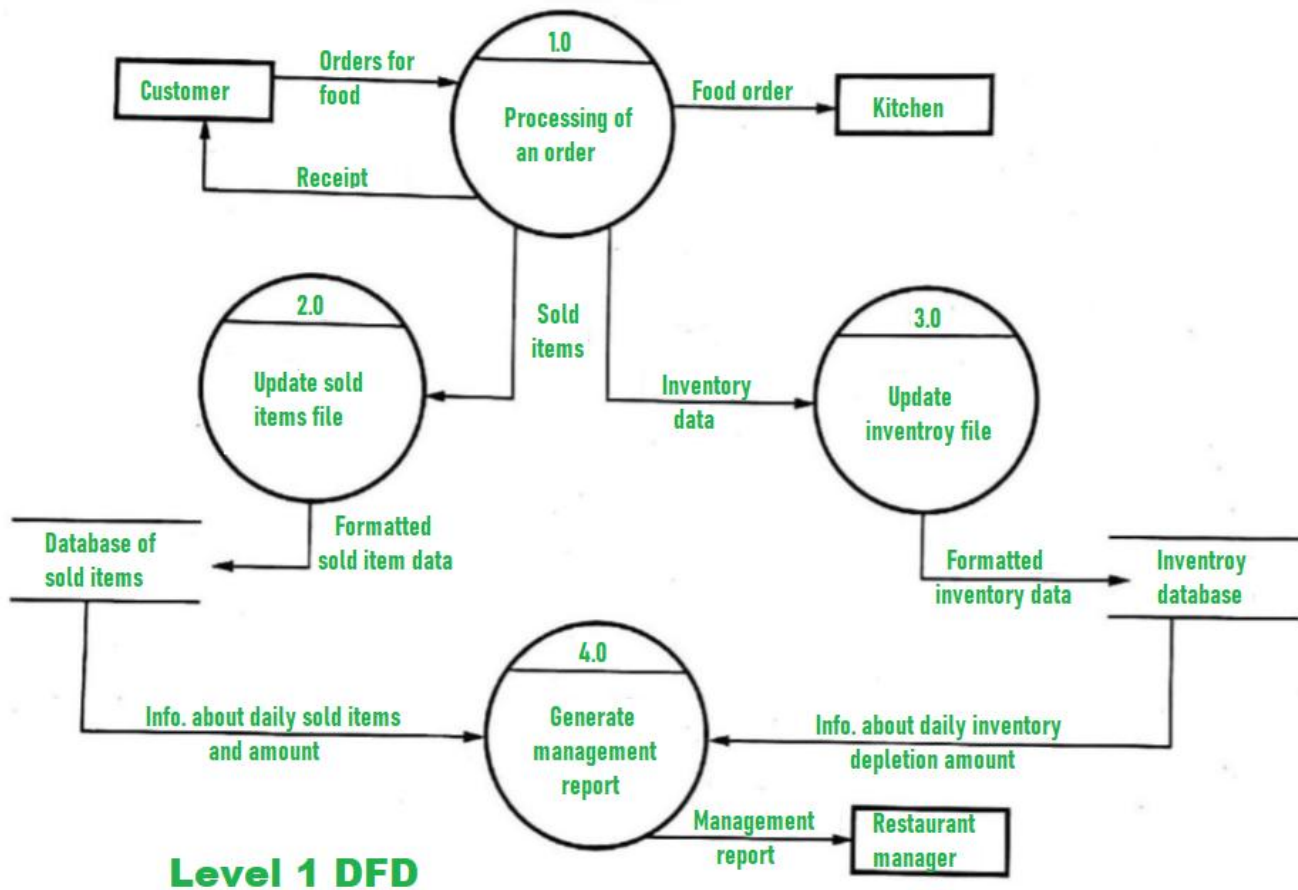
Database

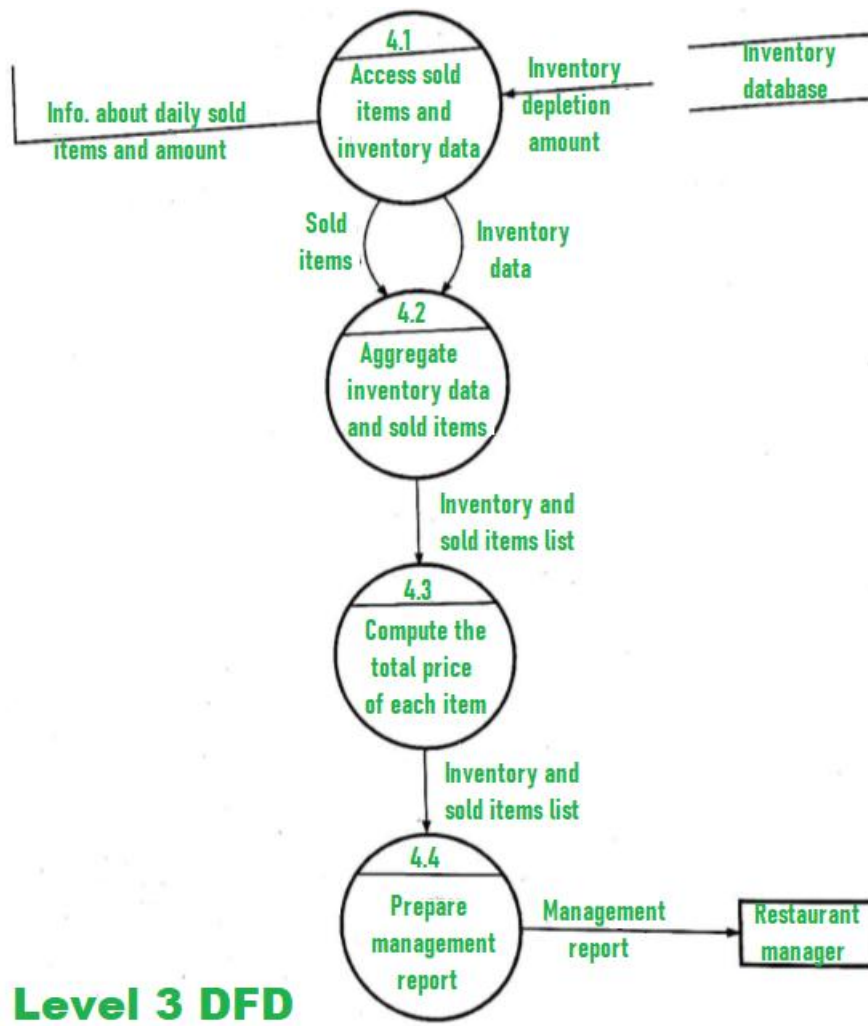
Testing of the database component is very straightforward, and has actually already been mostly completed. The database was the first component designed and before beginning work on any of the applications, I wrote all of the SQL statements I expected to need and executed them directly, essentially isolating 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.

DFD

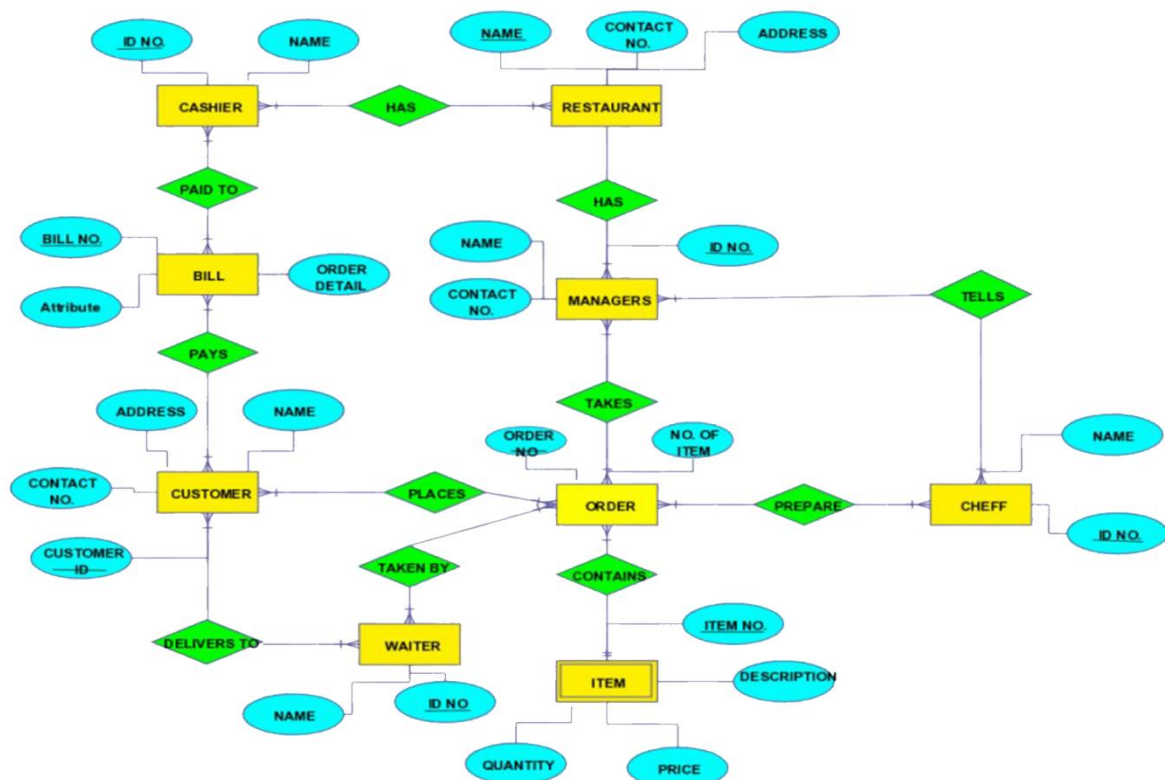


Level 0 DFD (Context Level)





ER Diagram



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.

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.

System Requirement:

Operating System:-Windows 10,Windows 11...

Ram:4GB

Server:-XAMPP

Software:-PHP,MySql

Chapter 5: User Manual

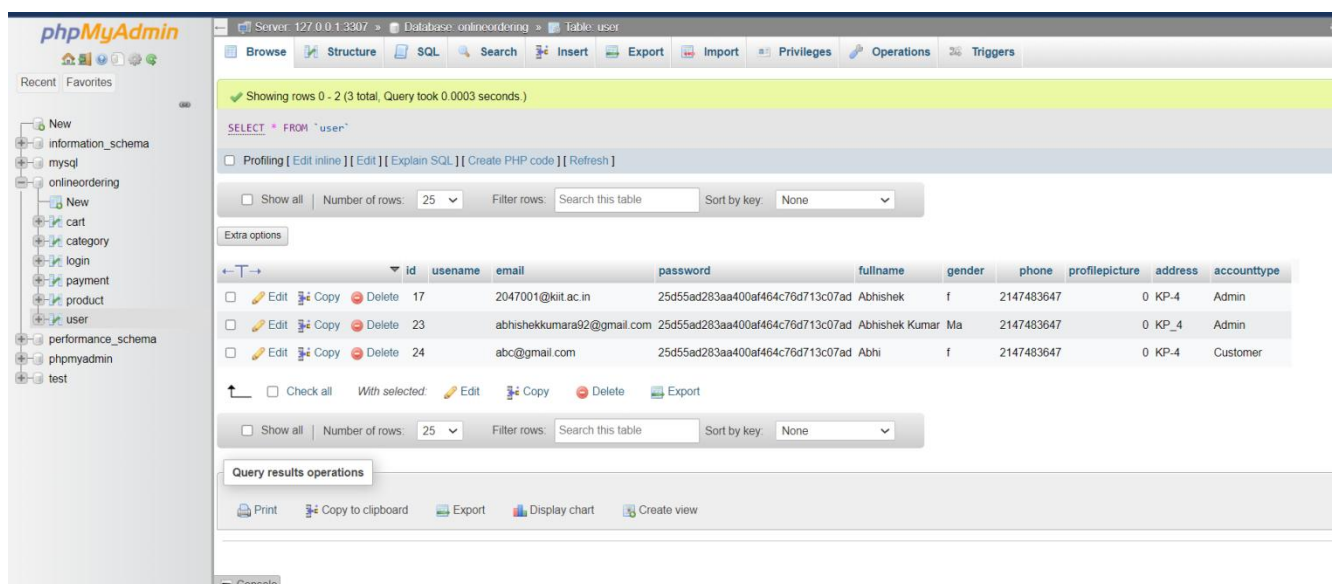
Using the Desktop Application

When the system is first installed, it contains no menu information.

Therefore, the firststep in using the system is to add vendor information. Start by launching the desktop client.

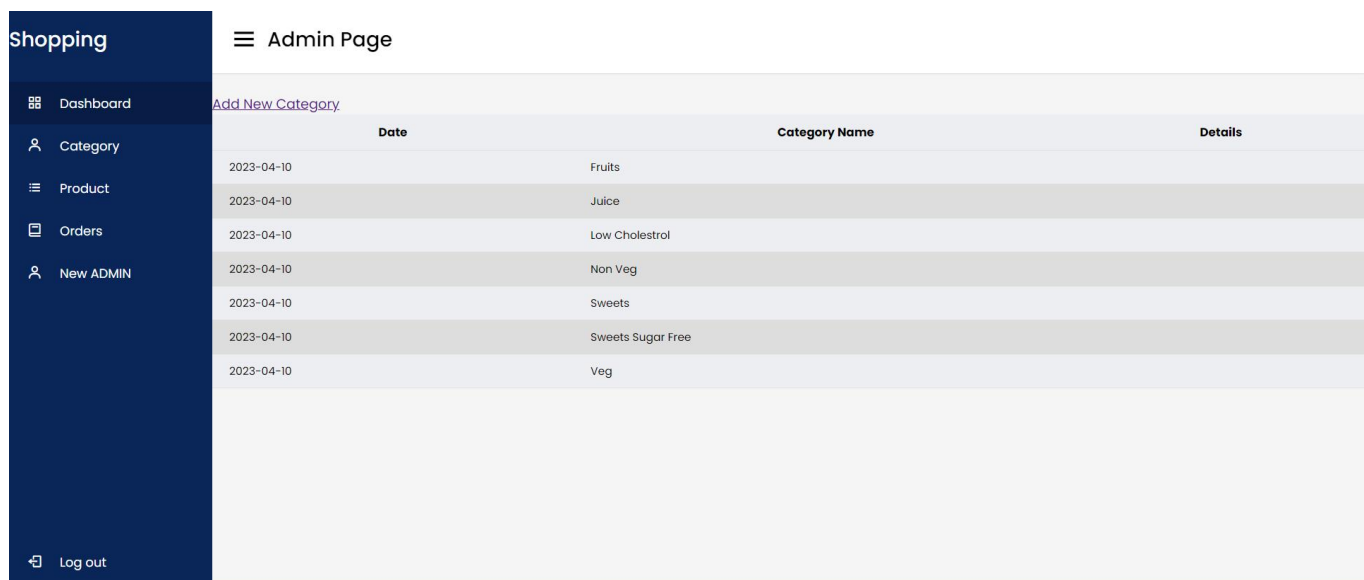
1.1Managing Customers

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 customers.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 bemultiple 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.



1.2Managing Categories

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 'Shopping Admin Page' with a dark blue sidebar on the left containing navigation links: Dashboard, Category, Product, Orders, and New ADMIN. The main content area has a header 'Admin Page' and a link 'Add New Category'. Below this is a table with columns 'Date', 'Category Name', and 'Details'. The table contains seven rows of category data.

Date	Category Name	Details
2023-04-10	Fruits	
2023-04-10	Juice	
2023-04-10	Low Cholesterol	
2023-04-10	Non Veg	
2023-04-10	Sweets	
2023-04-10	Sweets Sugar Free	
2023-04-10	Veg	

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”.

1.3Managing 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 items. 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.

phpMyAdmin

Server: 127.0.0.1:3307 > Database: onlineordering > Table: product

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Triggers

Recent Favorites

New

information_schema

mysql

onlineordering

- New
- cart
- category
- login
- payment
- product
- user

performance_schema

phpmyadmin

test

Showing rows 0 - 24 (74 total, Query took 0.0003 seconds)

SELECT * FROM `product`

Profiling

Edit inline

Edit

Explain SQL

Create PHP code

Refresh

1

>

>>

☐ Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

←

→

▼

id

productname

price

image

categoryname

☐

Edit

Copy

Delete

18

Apple

210

Apple.jpg

Fruits

☐

Edit

Copy

Delete

19

Orange

140

Oranges.png

Fruits

☐

Edit

Copy

Delete

20

Grape

90

Grape.png

Fruits

☐

Edit

Copy

Delete

21

Lychee

390

Lychee.png

Fruits

☐

Edit

Copy

Delete

22

Mango

230

Mango.jpg

Fruits

☐

Edit

Copy

Delete

23

Papaya

240

Papaya.png

Fruits

☐

Edit

Copy

Delete

24

Pineapple

180

Pineapple.png

Fruits

☐

Edit

Copy

Delete

25

Sitaphal

260

Sitaphal.png

Fruits

☐

Edit

Copy

Delete

26

Apple Juice

80

Apple Juice.png

Juice

☐

Edit

Copy

Delete

27

Lychee Juice

100

Lychee Juice.jpg

Juice

☐

Edit

Copy

Delete

28

Mango Juice

80

Mango Juice.png

Juice

1.4Managing Orders

To manage the placed order, navigate “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.

Shopping		Orders Page					
Dashboard							
Category							
Products							
Orders							
New Admin							
Log out							

Id	Customer	Location	Order Date	Status	Amount	Action	
19	Abhishek kumar	kp4	2023-04-28	Pending	140	Approve	Reject
18	Abhishek kumar	kp4	2023-04-28	Accepted	210	Approve	Reject
18	yu	kp4	2023-04-28	Rejected	210	Approve	Reject
18	yu	kp4	2023-04-28	Pending	210	Approve	Reject
84	Abhishek kumar	kp4	2023-04-28	Pending	150	Approve	Reject
22	Abhishek kumar	kp4	2023-04-28	Pending	230	Approve	Reject
20	Abhishek kumar	kp4	2023-04-28	Pending	90	Approve	Reject
41	Abhishek kumar	kp4	2023-04-28	Pending	30	Approve	Reject

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 email containing your password will be sent to specified account and you can then begin using the system.

Login

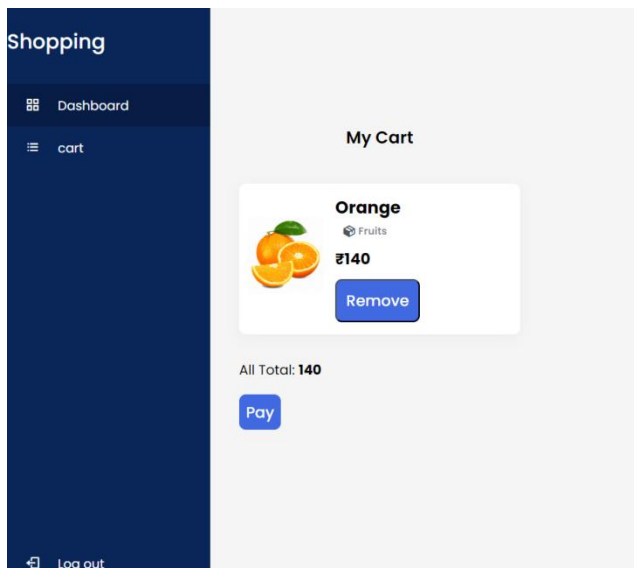
Login

Not a member? [Signup Now](#)

1.1 Selecting Food Items

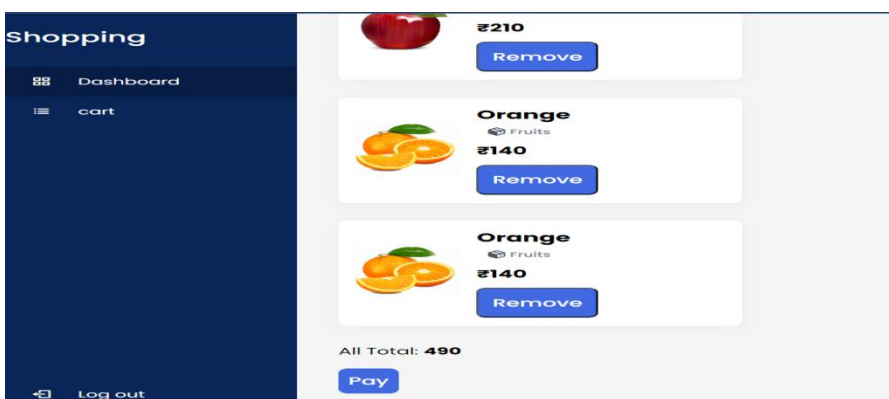
After signing in on the main page, click “Check Out Tonight’s Menu...” and select the category of food you would like to order. This will bring up an order form specific to that category. On this form,

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 .



1.2Placing 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 You should then receive confirmation that your order was placed successfully, along with your order number, completing the ordering process.



Using the Help System

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 helpdocument for the desktop application will also be distributed with the system as a .pdf to makeemployee training easier.

1.1Services Desktop Application

Adding Categories

Adding Foods

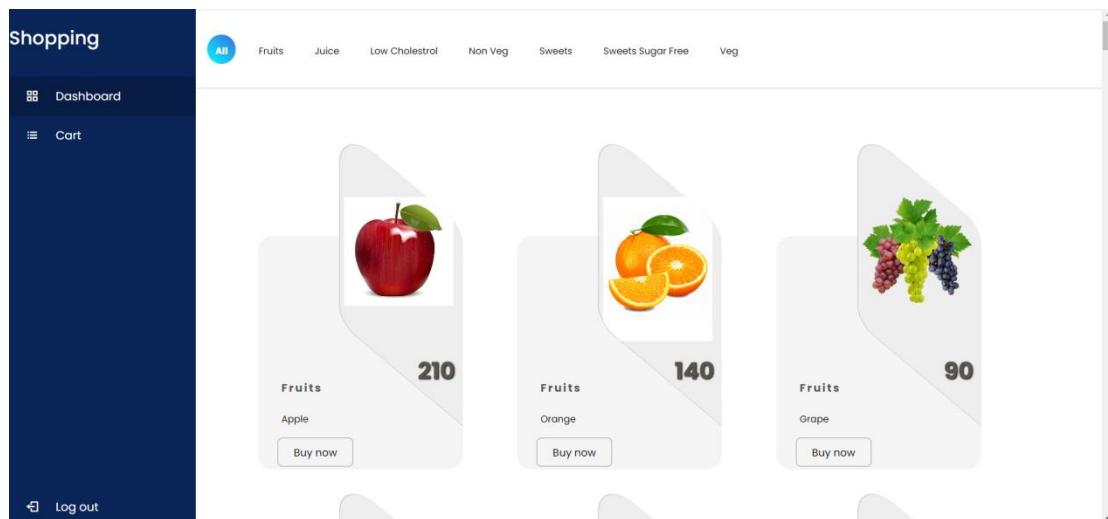
Adding Admin

Deleting Categories

Deleting Foods...etc

1.2Web 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



1.3Error 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 MySQL database that coordinates the entire system. At the present moment, this

creation is handled by a script which must be run from within a MySQL client . 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.