INTERNATION INSTITUTE OF PROFESSIONAL STUDIES, DAVV

SYSTEM ANALYSIS AD DESIGN

PROJECT REPORT-

**System Requirement Specification [SRS]**

**Online Food Ordering System**

**SUBMITTED BY-**

DRAKSHI CHOPRA

**IT-2K19-16**

MTECH V Sem

**SUBMITTED TO-**

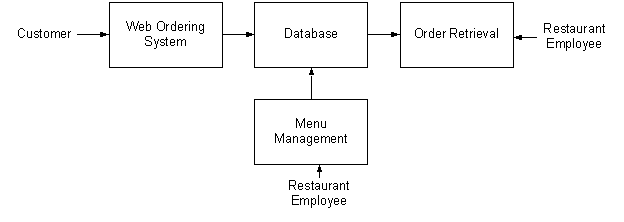
Dr. SHALIGRAM PRAJAPAT

**TABLE OF CONTENTS**

1. SYSTEM MODEL
2. FUNCTIONAL REQUIREMENTS
3. USER INTERFACE SPECIFICATION
4. NON FUNCTIONAL REQUIREMENTS
5. SYSTEM EVOLUTION

**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 most pertinent to accomplish the desired task. All of the functions outlined above, with the exceptions of account creation and management, will be used every time a customer places an order. By not including extraneous functions, I am moving towards my goal of simplifying the ordering process.

### Menu Management System

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

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

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

### Order Retrieval System

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

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

**USER INTERFACE SPECIFICATIONS**

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

### Web Ordering System

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

Entering delivery and payment deals 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 and 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 and order is processed, the user clicks a single button, labeled “Processed”, to remove it from the list of active orders.

## Non-functional Requirements

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

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

The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a restaurant that is not expecting to see much web traffic, or possibly doing only a limited test run, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cutoffs will need to be determined through a more thorough stress testing of the system.

**SYSTEM EVOLUTION**

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

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