

**Functional Specification**

**Food Delivery System**

Release: Final

Date: 12-11-2017

Authors: Li Chen, Jianhui Lu, Weiran Yuan

# Report History

## Document Location

This document is only valid on the day it was printed.

The source of the document will be found at RPI

## Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision date** | **Author** | **Version** | **Summary of Changes** | **Changes marked** |
| 10.28.2017 | Li, Jianhui, Weiran | 0.1 | Initial Draft |  |
| 12.10.2017 | Li | 1.1 | Update | 9. 10. |
| 12.11.2017 | Li | 2.1 | Final | 7. |

## Approvals

This document requires the following approvals:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date of Issue** | **Version** |
| Ingrid Liu | PhD, Professor |  | 0.1 |
|  |  |  |  |

## Distribution

This document has additionally been distributed to:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date of Issue** | **Status** |
|  |  |  |  |
|  |  |  |  |

**Table of Contents** Page

[1 Report History 2](#_Toc500955597)

[1.1 Document Location 2](#_Toc500955598)

[1.2 Revision History 2](#_Toc500955599)

[1.3 Approvals 2](#_Toc500955600)

[1.4 Distribution 2](#_Toc500955601)

[2 Purpose 4](#_Toc500955602)

[3 Background 4](#_Toc500955603)

[4 Metrics 5](#_Toc500955604)

[5 Scope 5](#_Toc500955605)

[5.1 In Scope 5](#_Toc500955606)

[5.2 Out of Scope 5](#_Toc500955607)

[6 Timescales and Priorities 6](#_Toc500955608)

[7 Summary of Business Requirements 6](#_Toc500955609)

[8 Summary of Functional Areas 7](#_Toc500955610)

[9 Functional Requirements by Module / Use Case Descriptions 7](#_Toc500955611)

[9.1 Module 1 (Customer End): Web Ordering System 7](#_Toc500955612)

[9.2 Module 2 (Restaurant End): Menu Management System 8](#_Toc500955613)

[9.3 Module 3: Order Retrieval System 9](#_Toc500955614)

[10 Screens and Workflows 9](#_Toc500955615)

[10.1 Ordering System 9](#_Toc500955616)

[10.2 Menu Management System & Order Retrieval System 10](#_Toc500955617)

[11 User Interface Description 10](#_Toc500955618)

[11.1 Ordering Interface 10](#_Toc500955619)

[11.2 Menu Management Interface 11](#_Toc500955620)

[11.3 Ordering Retrieval Interface 11](#_Toc500955621)

[12 Users and Security 11](#_Toc500955622)

[13 System administration and maintenance 12](#_Toc500955623)

[14 Non-Functional Requirements 12](#_Toc500955624)

# Purpose

The purpose of this document is to summarise the functional requirements of food delivery system (web application). It is not a system solution, but a guideline of the required system functionality.

The document sets out detail of

* Metrics
* User types
* The modules
* User tasks
* Functional requirements of each module.

# Background

In this day and age, food delivery is becoming more and more popular among customers around the world. There is a great demand of getting food from restaurants without stepping out of home. Although almost every one is using a computer or a smart phone, most restaurants in the states are still accepting delivery orders via phone calls. We think that an online ordering system would have a lot of advantages over phone ordering system. Firstly, customers don’t need to have a physical copy of the menu any more. Secondly, customers can have a visual confirmation to make sure that their orders are placed correctly. What’s more, it will not be necessary for the employees of restaurants to answer the phone and write down the orders.

What we are trying to offer is a web application of online ordering, which is designed for use in food delivery industry. The main advantage of our system is that it will greatly simplifies the ordering process, not only for customers but the restaurant employees. When a customer visits our ordering webpage, it will display an up-to-date and interactive menu that contains all the available options. And it will dynamically adjust the price when the customer checks different options. After making the selection, the item will be added to their order list, and the customer can view the details at any time before he/she checks out. This provides the instantaneous visual confirmation of the selected and ensures that the items in the order list are actually expected.

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.

This system also greatly reduces the burden on the restaurant, because the entire order process is automated. Once orders are placed by customers on the page, they will be input into the database. Then almost in the same time, the order will be retrieved from the database to the end of the restaurant. In this application, all items of an order, all the corresponding options and delivery details are displayed in a straightforward and humanistic manner. This allows the restaurant staff to quickly take the order, and produce the ordered items with minimal confusion and delay.

# Metrics

The system is required to cater for the following approximate current volumes, (based on figures from YEAR)

|  |  |
| --- | --- |
| New students registered per annum | 4,400 |
| Historic registered students | 140,000 |
| New enquirers per year | 400 |
| Course managers and designers accessing the system | 22 |

# Scope

This functional specification will cover both current deliverables as well as some future functionalities. The food delivery system basically consists of five main components: menu management system, web ordering system, order retrieval system, delivery tracking system and customer feedback system. The first two components are the critical sections of the system. Given that our time is limited and the size of the group is small, we put delivery tracking system and customer feedback system as future functionality which is currently out of scope.

## In Scope

* First, the system should have a subsystem of menu management for restaurants. It should provide restaurant users a simple and humanistic way to control what could be ordered by their customers.
* Second, the system must allow customer users to place their orders. It should also provide the functionality for users to view the detailed information of restaurants and foods.
* Third, the system should provide restaurant users the functionality to keep track of orders that have been placed and orders that have been processed. In specific, the system should retrieve and display order information for restaurant users once an order is placed by a customer. And it should update the status of the order once it is processed by the restaurant.

## Out of Scope

* The web application might be able to provide expected delivery time and allow customers to track the status of their orders in real time. The status could be order accepted by the restaurant, out for delivery, food delivered, etc.
* The web application could provide the functionality for customers to provide feedbacks of the food, the restaurant, the delivery process and the deliveryman.

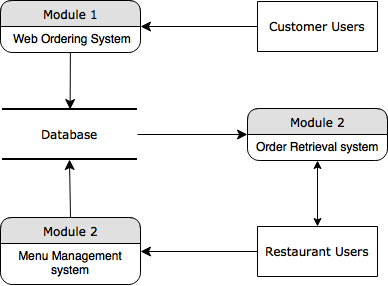
# Timescales and Priorities

|  |  |  |
| --- | --- | --- |
| **Project milestone** | **Tools** | **Due date** |
| Project start: environment settings, IDE installations | XAMPP / Eclipse / TomCat | 10-30 |
| Website overall design: web style | Mockflow | 11-6 |
| Customer account page: create account, login, account management | Brackets | 11-10 |
| Customer view page: view lists of food | Brackets | 11-10 |
| Place order page: shopping cart, order information, calculate prices and tips, payment | Brackets | 11-14 |
| Customer order page: current order & history order | Brackets | 11-17 |
| Restaurant update menu page | Brackets | 11-17 |
| Restaurant order retrieval page: receive new order notification, modify order state | Brackets | 11-20 |
| Create customer table in database | PostgreSQL database | 12-9 |
| Create restaurant table in database | PostgreSQL database | 12-9 |
| Create order table in database | PostgreSQL database | 12-9 |

# Summary of Business Requirements

* Customers can view restaurant details and menus without logging in. However, orders can only be placed after the customer logs in.
* The website is a food ordering platform for different restaurants, its not just for one specific restaurant. Therefore, signup approach should also be provided to restaurant users. Restaurants should be displayed on the home page for customers to click into the corresponding order page.
* Customers should be able to view their order cart in the order page. The order cart should react to any order activity (add, remove, change quantity) dynamically.
* Orders should be allowed to be placed before payments, because customers might choose to pick their food on-site and make payments in the restaurants.
* Each order should have three states: New (Order Sent), Processing (Accepted) and Completed (Completed)
* Users don’t have to login again in a same equipment within an hour, thus cookies should be stored when logged in.

# Summary of Functional Areas



**Customer End**

**Restaurant End**

**Restaurant End**

# Functional Requirements by Module / Use Case Descriptions

This section sets out the functional requirements of the system by module. It would include details of key functions that the system must perform:

* Key processes (including bulk processes, workflows etc)
* Creation/Amendment/Deletion of records

The requirements set out here are ranked in **M**o**SC**o**W** order:

M – Must Have  
S – Should Have  
C – Could Have  
W – Would like to have

## Module 1 (Customer End): Web Ordering System

To provide restaurant users a simple and humanistic way to control what could be ordered by their customers

| **Function Id** | **PROGRAMME Functional description**  **& User Cases** | **Priority Level**  **(MoSCoW)** | **Key Analysis Points/Notes** |
| --- | --- | --- | --- |
| 1.1 | Allow customer users to manage identified accounts   * Create an account * Log in to the system * Edit the profile of their accounts | M  M  S |  |
| 1.2 | Allow customer users to create orders when viewing a menu   * Navigate the restaurant’s menu * Select an item from the menu * Customize options for a selected item * Add an item to their current order, create a new order when it is the first item | S  M  S  M |  |
| 1.3 | Allow customers to review their current order | M |  |
| 1.4 | Allow customers to modify their current order   * Remove an item from current order * Remove all items from current order * Go back to the menu page | M  S |  |
| 1.5 | Provide delivery and payment details  Save default delivery and payment details | M  C |  |
| 1.6 | Place an order to the restaurant | M |  |
| 1.7 | Receive confirmation in the form of an order number | S |  |

## Module 2 (Restaurant End): Menu Management System

To allow customer users to place their orders and view the detailed information of restaurants and foods

| **Function Id** | **PROGRAMME Functional description**  **& User Cases** | **Priority Level**  **(MoSCoW)** | **Key Analysis Points/Notes** |
| --- | --- | --- | --- |
| 2.1 | Add a new/update/delete vendor to/from the menu | S |  |
| 2.2 | Add a new/update/delete food category to/from the menu | S |  |
| 2.3 | Add a new/update/delete food item to/from the menu | M |  |
| 2.4 | Add a new/update/delete option for a given food item | C |  |
| 2.5 | Update price for a given food item | S |  |
| 2.6 | Update default options for a given food item | C |  |
| 2.7 | Update additional information (description, photo, etc.) for a given food item | S |  |

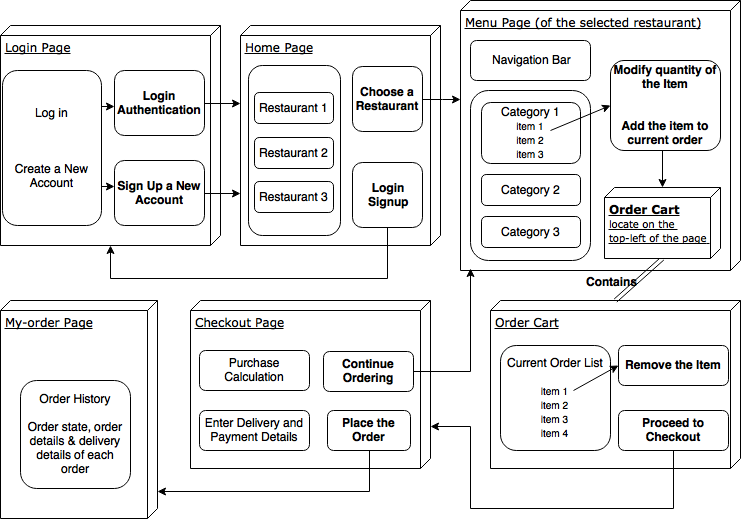
## Module 3 (Restaurant End): Order Retrieval System

To provide restaurant users the functionality to keep track of orders that have been placed and orders that have been processed

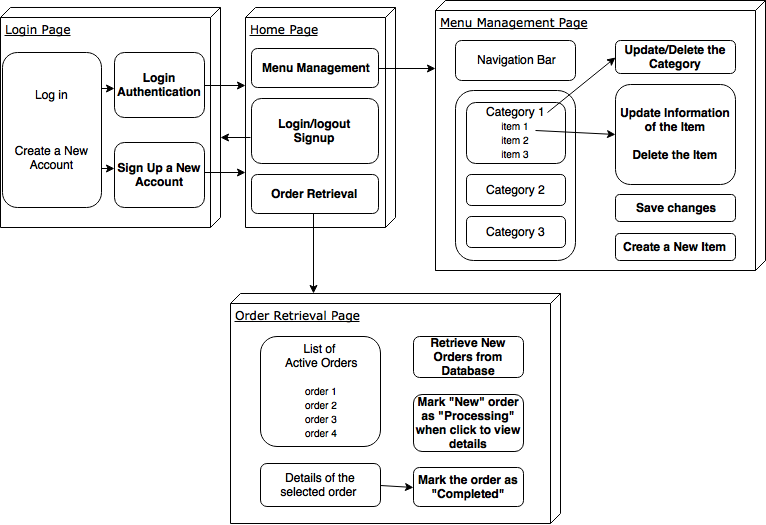
| **Function Id** | **PROGRAMME Functional description**  **& User Cases** | **Priority Level**  **(MoSCoW)** | **Key Analysis Points/Notes** |
| --- | --- | --- | --- |
| 3.1 | Retrieve new orders from the database | M |  |
| 3.2 | Display the orders in an easily readable, graphical way | S |  |
| 3.3 | Mark an order as having been processed and remove it from the list of active orders | S |  |

# Screens and Workflows

## Ordering System



## Menu Management System & Order Retrieval System



# User Interface Description

## Ordering Interface

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.

Users of the web application interact with it via a series of simple forms. Each category of food will have its own form associated with it. It will provide drop-down menus for users to select which particular item of the category they want to add to the order list. A series of checkboxes and radio buttons will be presented for users to select which options to include. You can add an item to the order list by simply clicking the button. Users can choose which category of food they want to order by navigating the menu bar.

Delivering and paying transactions is processed in a common way. Users are presented with a form of the drop-down lists and the text boxes where the required fields must be filled before checking out. One thing to note is that, in order to simplify the process and reduce the possibility of SQL injection attempts, drop down boxes or buttons are used rather than text boxes whenever possible.

## Menu Management Interface

Interactions with the menu management interface is quite similar to those with the web ordering interface. Users browse a tree structure to find the category or some specific food items that they would like to make a change. Then after making their selection, they will be presented with all the current fields and values associated with the item. It will also provide buttons that allow users to add new fields or values. However, unlike the web ordering interface, most inputs here will be in a free format, typically in the form of a text box. This will not cause significant concern because the input hygiene will be implemented and users who are considered to be restaurant employees should be less likely to be malicious.

## Ordering Retrieval Interface

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.

The application will automatically refresh and get the new order from the database and display the order number and delivery time in a panel located on some part of the application. The user can simply click on an order number to view the details. The details of the order will be displayed on another panel in an easy-to-read and navigable manner. The structure can be expanded or collapsed intuitively, displaying only the required information. Finally, once the order is processed, the restaurant user can click on the "Processed" button to remove the order number from the list and change the status of the order from active to processed.

# Users and Security

|  |  |  |
| --- | --- | --- |
| User Types | User Interfaces | User permissions |
| Restaurant user | Menu management interface  Order retrieval interface | Restaurant user account management  Modify the restaurant’s menu  Modify the restaurant’s working hours / status  Modify the restaurant’s information  Accept orders  Change the status of accepted orders |
| Customer user | Ordering interface | Customer user account management  Place order(s)  Cancel order(s)  Mark an order as received |

# System administration and maintenance

The base data in the system (e.g. users, parameters) will be maintained by the maintenance team (same as the development team) on a weekly basis? Members of the team are the owners of the data.

# Non-Functional Requirements

The application should be cross-compiled to HTML and JavaScript, along with a PostgreSQL database and a JSP backend. Particularly, we will use the XAMPP bundle as it is freely available online and it is compatible will different operation systems. For the implementation of Java codes for the server end, we use the Eclipse IDE with Apache TomCat installed.

All of the application data will be stored in a PostgreSQL database. PostgreSQL is the default database on macOS Server as of OS X Server version 10.7. Therefore, we can use PostgreSQL as our object-relational database management system.

The requirement for hardware is pretty low since it is an experimental project. It could be any computer that is capable to run both web and database servers and handle the expected traffic. Therefore, a normal personal computer would be appropriate for the system.