Online Restaurant Order and Delivery System

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/Oct/20 | 1.0 | Document Initialization | Leji Li, Weiye Kuang, Xian Chen, Bingjing Dong |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 5

2. Overall Description 5

2.1 Use-Case Model Survey 5

2.1.1 Actors 5

2.1.2 Use Cases 5

2.1.3 Use-Case Diagram 8

2.2 Assumptions and Dependencies 8

3. Specific Requirements 9

3.1 Use-Case Reports 9

3.2 Supplementary Requirements 17

4. Supporting Information 18

4.1 Appendices 18

4.1.1 User Interface 18

Software Requirements Specification

# Introduction

## Purpose

The purpose of this Systems Requirements Specifications (SRS) document is to build an online restaurant order and delivery system. The purpose of the online restaurant order and delivery system is to ease restaurant management and to create a convenient and easy-to-use application for customer orders. The system is based on a relational database with its restaurant provides menu and customer order functions. We will have a database server collecting all the feedback from customers from their ordering using a rating system. Based on the feedback for different dishes, we will decide how the chef salary. We hope to provide a comfortable user experience along with the best dishes available.

## Scope

This SRS document specifies requirements for an online restaurant order and delivery system, including functional, performance, admit and other system requirements.

The online restaurant order and delivery system will

* The system will provide an interface for a menu with pictures and 3 top dishes suggestions to help customers choose their food. After customers ordered food, they can choose if they want to eat in a restaurant, take out, or delivery. Based on the way they choose; they can do their right to rate the like for food and delivery people.
* The system will provide a statistical history of ordered food, ratings, and reviews record. Based on the recording, the system will change the 3 listing dishes for different customers.
* Allow customers/ chef/ delivery to dispute the complaint they received, change to warning without merit by manager.
* Allow manager to finalize the complaint for customers/ chef/ delivery people.
* Provide a taboo list and keep by the manager, replace by star sign automatically.
* Allow customers to de-registered from the system, handle by the manager.

## Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specification, this document outlines the requirements that the software must fulfill.

GUI – Graphical User Interface.

User – any person who use the system.

Customer – registered customer who can order and vote dishes for their order.

VIP – special customer in the system who get priority and discount for their order.

Surfer – people who can only view menus and ratings only.

## References

The Python3 reference is available at

https://docs.python.org/3/reference/index.html

Graphical User Interfaces with Tk is available at

https://docs.python.org/3/library/tk.html

## Overview

This section is an overall look for the client to be familiar with this software application. Including base functions, the general idea for the whole application.

Section 2 contains more detail of the requirements of the online restaurant order and delivery system, including a Use-Case model survey and assumptions and dependencies.

Section 3 contains all the technical requirements for the software, including Use-Case reports and supplementary requirements.

# Overall Description

## Use-Case Model Survey

### Actors

#### Chef

The person who makes the dishes and sets up the menus.

#### Delivery People

The person who delivers customers’ order.

#### Manager

The person who controls and admins the system.

#### Customer

The person who orders dishes and gives feedback via the system.

#### VIP

The customer who spent more than 500 dollars and/or ordered more than 50 orders. VIPs’ vote is twice as much as the regular Customers.

#### Surfer

The person who can view the menus and ratings and discussions on the dishes on the menus.

### Use Cases

#### Login

Let all actors above (except surfer) access to the system

#### Edit Menus

Chefs can add/modify/delete the detail of a dish on menus with this use case, which includes the dish itself, the description of the dish and keywords.

#### Handle Customer Registration

This use case allows a Manager to review the customer registration request from a surfer. The manager can either approve or deny the request.

#### View Menus

This use case allows all actors to browser the menus in the system.

#### Search Menus

This use case allows all actors to search specific dishes by typing in keywords.

#### Order

This use case allows Customers and VIP to order dishes.

#### Vote

This use case allows Customers and VIPs to give rating to the dishes they ordered. They can also rate for delivery quality if the order is a delivery order.

#### Customer Upgrade

This use case allows Customers upgrade to VIPs if they have spent over 500 USD or placed more than 50 orders.

#### Start Discussion

This use case allows Customers and VIPs to create a discussion thread on a dish/the cook of the dish/delivery people.

#### Participate Discussion

This use case allows Customers and VIPs reply on the existing discussion thread.

#### View Special Menus

This use case is for VIPs only. VIPs can browser the system’s special menus.

#### Customer Registration

This use case is for Surfers only. Surfers can apply for becoming Customers by deposit a fixed amount of balance and review by a Manager.

#### Hire

This use case allows a Manager to hire the staff of the company, including Chefs and Delivery People.

#### Demoted

This use case allows a Manager to reduce the payment of a Chef or a Delivery People; Or downgrade a VIP to a regular Customer.

#### Promoted

This use case allows a Manager to raise the payment of a Chef or a Delivery People.

File Complaint or Compliment

This use case allows a Customer/VIP to file a complaint/compliment to a dish/Chef/Delivery People. This use case also allows a Delivery People to file a complaint/compliment to the Customer/VIP who made the order.

#### Handle Complaint

This use case allows a Manager to review the received complaints. This Manager can either turn a complaint to the formal warning or dismiss it.

#### Dispute Complaint

This use case allows the person who received a complaint, Customer/VIP/Delivery People, to argue against the complaint.

#### De-register

This use case allows all actors, except Surfers, to send a request to a Manager for reviewing. If a Customer of a VIP brings up a de-register request, his/her account will be temporarily frozen.

Handle De-register

This use case is for Managers only. A manager reviews the de-register request and either approves or dismiss.

#### Review Order

This use case will be triggered when a Customer/VIP submit an order. If the payment is not enough for the order, that order will be frozen until additional payment is received.

#### Taboo List

This use case allows a Manager to maintain a list of taboo words, which should not be shown in discussions.

#### Review Comment

This use case will be triggered when a comment is submitted. If the comment contains any taboo words, the one who submit the comment will receive a warning, and the taboo words will be replaced by \*\*\*; if there are 3 or more taboo words, the comment will be blocked.

### Diagram Description automatically generatedUse-Case Diagram

Figure 1 Use-Case Diagram for Online Restaurant

## ****Assumptions and Dependencies****

Team members should help each other in order to keep the development in progress. This application will be developed using the Python programming language and evolutionary model will be applied. There will be weekly stand-up meetings about the parts that each member is working on. Tasks will be assigned and finish weekly. Target hardwired will be capable for application deployment and have the compute power to run the application.

# Specific Requirements

Team members should help each other in order to keep the development in progress. This application will be developed using the Python programming language and evolutionary model will be applied. There will be weekly stand-up meetings about the parts that each member is working on. Tasks will be assigned and finish weekly. Target hardwired will be capable for application deployment and have the compute power to run the application.

## Use-Case Reports

Table 3.1.1 presents the Customer Registration use case description to show the interaction between a surfer and the system, and the functional requirements when the surfer apply to be a registered customer

|  |  |
| --- | --- |
| **Use Case** | Customer Registration |
| **Primary Actor** | Surfer |
| **Goal In Context** | surfer can apply to be a registered customer |
| **Preconditions** | Not a registered customer |
| **Trigger** | None |
| **Scenario** | 1)The surfer click the button “Register”  2)The surfer deposit fixed amount of money  3)The system sent an application to the manager |
| **Exceptions** | None |

**Table 3.1.1**

Table 3.1.2 presents the Login use case description to show the interaction between the actors and the system, and requirements when the actor login the system

|  |  |
| --- | --- |
| **Use Case** | Login |
| **Primary Actor** | All actors |
| **Goal In Context** | Enable the actor access to their subsystems. |
| **Preconditions** | All actors and they are not already logged in |
| **Trigger** | Try to use the system |
| **Scenario** | 1)The actor click the “Login” button  2)The actor enter the username and password  3)The actor click “Login”  4)The system checks the username and password |
| **Exceptions** | Invalid username or password |

**Table 3.1.2**

Table 3.1.3 presents the Edit Menu use case description to show the interaction between the chef and the system, and requirements when the chef modifies the menu.

|  |  |
| --- | --- |
| **Use Case** | Edit Menu |
| **Primary Actor** | Chef |
| **Goal In Context** | Modify the menu, add a dishes, delete a dishes or change the price |
| **Preconditions** | Chef, already logged in |
| **Trigger** | Chef want to change the menu |
| **Scenario** | Add a dishes:  1)The chef click the button “Add”  2)The chef enter the name of the dishes and description  3)The chef enter the price  4)The chef upload the image  5)The chef click “Submit”  6)The system check whether the name already existed  7)The system check the price valid or invalid  8)The system add the dishes to the menu  Delete a dishes:  1)The chef click the dishes  2)The chef select “delete”  3)The chef click “Confirm”  4)The system delete the dishes from the menu  Change price:  1)The chef click the dishes  2)The chef select “Change price”  3)The chef enter the new price  4)Click “Submit” |
| **Exceptions** | Add an dishes:  1. Name of the dishes already existed  2. Invalid price  Change price:  1.Invalid price |

**Table 3.1.3**

Table 3.1.4 presents the Order use case description to show the interaction between the customers and the system, and requirements when the customer places an order.

|  |  |
| --- | --- |
| **Use Case** | Order |
| **Primary Actor** | Customer, VIP |
| **Goal In Context** | Customer add dishes to cart , delete dishes from cart and check out |
| **Preconditions** | Logged in customer |
| **Trigger** | Customer want to place an order |
| **Scenario** | 1)The customer click the dishes he wants  2)The customer select the quantity  3)The customer add the dishes to the cart  4)The customer click “cart” go to the cart  5)The system shows the dishes in the cart  6)The customer can click “delete” to delete the dishes from the cart  7)The customer can choose eat-in, pick-up, delivery  8)The customer click “check out” to pay the bill  9)If VIP, the system will give 10% discount |
| **Exceptions** | Money in the account isn’t enough |

**Table 3.1.4**

Table 3.1.5 presents the hire use case description to show the interaction between the manager and the system, and requirements when the manager hires a new chef or delivery people

|  |  |
| --- | --- |
| **Use Case** | hire |
| **Primary Actor** | Manager |
| **Goal In Context** | Manager hire chef or delivery people |
| **Preconditions** | None |
| **Trigger** | None |
| **Scenario** | 1)Manager create a new account for the chef or the delivery people  2)Manager select the identity for the account, chef or delivery people  3)Manager enter the username and password  4)The system check the username |
| **Exceptions** | Username already existed |

**Table 3.1.5**

Table 3.1.6 presents the Cut Pay use case description to show the interaction between the manager and the system, and requirements when the manager cuts pay

|  |  |
| --- | --- |
| **Use Case** | Cut Pay |
| **Primary Actor** | Manager |
| **Goal In Context** | Manager cuts the salary of a chef or a delivery person |
| **Preconditions** | None |
| **Trigger** | A chef or a delivery person is demoted |
| **Scenario** | 1)Manager selects the chef or the delivery people who he wants to cut pay for  2)Manager enters the amount of money he wants to cut |
| **Exceptions** | None |

**Table 3.1.6**

Table 3.1.7 presents the Raise Pay use case description to show the interaction between the manager and the system, and requirements when the manager raises pay

|  |  |
| --- | --- |
| **Use Case** | Raise Pay |
| **Primary Actor** | Manager |
| **Goal In Context** | Manager raise the salary of a chef or a delivery person |
| **Preconditions** | None |
| **Trigger** | A chef or a delivery person is promoted |
| **Scenario** | 1)Manager selects the chef or the delivery people who he wants to raise pay for  2)Manager enters the amount of money he wants to raise |
| **Exceptions** | None |

**Table 3.1.7**

Table 3.1.8 presents the Fire use case description to show the interaction between the manager and the system, and requirements when the manager fires a chef or a delivery person.

|  |  |
| --- | --- |
| **Use Case** | Fire |
| **Primary Actor** | Manager |
| **Goal In Context** | Manager fires a chef or a delivery person |
| **Preconditions** | None |
| **Trigger** | A chef or a delivery person is demoted twice. |
| **Scenario** | 1)Manager delete the account of the chef or the delivery people |
| **Exceptions** | None |

**Table 3.1.8**

Table 3.1.9 presents the Vote use case description to show the interaction between the customers and the system, and requirements when the customer rates the food.

|  |  |
| --- | --- |
| **Use Case** | Vote |
| **Primary Actor** | Customer, VIP |
| **Goal In Context** | Customers grade the food. |
| **Preconditions** | The customer purchased food before |
| **Trigger** | The customer receives the food delivered |
| **Scenario** | 1)The voting window display  2)The customer vote |
| **Exceptions** | None |

**Table 3.1.9**

Table 3.1.10 presents the Vote use case description to show the interaction between the customers and the system, and requirements when the customer searches the dishes.

|  |  |
| --- | --- |
| **Use Case** | Search |
| **Primary Actor** | Customer |
| **Goal In Context** | The customer can find a dishes quickly |
| **Preconditions** | Logged in customer |
| **Trigger** | The customer want to find a dishes |
| **Scenario** | 1)The customer enter the name of dishes  2)The system display the searching results |
| **Exceptions** | The dishes doesn’t exist |

**Table 3.1.10**

Table 3.1.11 presents the Discussion use case description to show the interaction between the customers and the system, and requirements when the customer starts a new discussion or joins in a discussion.

|  |  |
| --- | --- |
| **Use Case** | Discussion |
| **Primary Actor** | Customers, VIPs |
| **Goal In Context** | Customers can share their comments and discuss with other customers |
| **Preconditions** | Logged in customers |
| **Trigger** | None |
| **Scenario** | 1)The customer clicks the “Discussion” button  2)The system goes to the discussion window and lists the hottest discussion topic.  3)The customer can click the topic to view the comments from other comments and join in the discussion  3)The customer can click “Create” to start a new discussion |
| **Exceptions** | Taboo words are replaced by \*\*\*\*  Comments with more than 3 taboo words would be blocked |

**Table 3.1.11**

Table 3.1.12 presents the Fire Compliments/Complaints use case description to show the interaction between the customers, the delivery people and the system, and requirements when they fire a complaint or a compliment

|  |  |
| --- | --- |
| **Use Case** | Fire Compliments/Complaints |
| **Primary Actor** | Customer, VIP, deliver people |
| **Goal In Context** | Customers can file complaints or compliments to the chef or the delivery people  Delivery people can file complaints or compliments to the customers he delivered to. |
| **Preconditions** | The customer purchased food before (for eat-in and pick up, the customer can only complaint or compliment to the chef)  The delivery person delivered dishes to the customer before |
| **Trigger** | The customer is dissatisfied or very satisfied with the dishes and delivery.  The customer takes an abominable attitude toward the delivery person |
| **Scenario** | For customers:  1)The customer can find the chef and the delivery person from his order history  2)The customer can go to the description page of the chef or the delivery person by clicking the name  3)The customer can click “Complaint” or “Compliment” and write down the reason  4)The system sends the complaint or compliment and the reason to the manager.  For delivery people:  1)The delivery person can find the customer from his delivered history  2)The delivery person clicks the username of the customer and selects “Complaint” and then writes down the reason.  3)The system sends the complaint and the reason to the manager. |
| **Exceptions** | None |

**Table 3.1.12**

Table 3.1.13 presents the Dispute Complaints use case description to show the interaction between the customers, the chef, the delivery people and the system, and requirements when they want to dispute a complaint.

|  |  |
| --- | --- |
| **Use Case** | Dispute Compliant |
| **Primary Actor** | Customer, VIP, Delivery people, Chef |
| **Goal In Context** | When the customer, the chef or the delivery person disagrees with the complaint, they can dispute the complaint. |
| **Preconditions** | The customer, the chef or the delivery person receives a submitted complaint |
| **Trigger** | They are dissatisfied with the result |
| **Scenario** | 1)The system displays a window to inform that he was complained and the reason  2)If the customer, the chef, or the delivery person disputes the complaint, he can write down the reason and click “Dispute”  3）The system sends this compliant to the manager |
| **Exceptions** | None |

**Table 3.1.13**

Table 3.1.14 presents the Handle Complaints use case description to show the interaction between the manager and the system, and requirements when someone receives or disputes a complaint.

|  |  |
| --- | --- |
| **Use Case** | Handle Compliant |
| **Primary Actor** | Manager |
| **Goal In Context** | Manager handles the complaints |
| **Preconditions** | None |
| **Trigger** | Someone files a complaint to the other or disputes the complaint he received. |
| **Scenario** | 1)The system lists the complaints  2)The manager can click the complaint to see the details  3)The manager handles the complaint  4)If the manager thinks the complaint is without foundation, he can dismiss the complaint, also he could send a warning to the person who files the complaint.  5)If the manager agrees with the complaint, then the system would inform the person to whom the complaint is. |
| **Exceptions** | None |

**Table 3.1.14**

Table 3.1.15 presents the Kick use case description to show the interaction between the manager and the system, and requirements when a customer receives 3 warnings and de-registered or wants to quit the system.

|  |  |
| --- | --- |
| **Use Case** | Kick |
| **Primary Actor** | Manager |
| **Goal In Context** | Manager kicks the customer out. |
| **Preconditions** | None |
| **Trigger** | A customer receives 3 warnings and de-registered.  A customer wants to quit the systems |
| **Scenario** | 1)The system displays the customer who is de-registered or wants to quit the system  2)The manager clicks “Confirm”  3)The system clears the deposit and delete the account and file of the customer |
| **Exceptions** | None |

**Table 3.1.15**

## Supplementary Requirements

|  |  |
| --- | --- |
| **Requirements** | **Description** |
| sales\_rank | The system ranks all the dishes, basing on the sales volumes of the dishes. |
| rating\_rank | The system ranks all the dishes, basing on the average of customers’ rating points for the dishes. |
| history\_rank | Each customer has his own history\_ranking.  The system ranks the dishes, basing on the times of the dishes purchased by the customer. |
| VIP | A register would become a VIP when he spent 500 dollars or placed 50 orders totally. |
| Demote | The chef whose dishes received consistently low ratings, 3 complaints or no order for 3 days will be demoted.  The delivery people who received 3 complaints will be demoted.  The salary of the demoted chef and delivery people will be less, and the system will record the demoting in their files. |
| Promote | The chef whose dishes received consistently high ratings; 3 compliments will be promoted.  The delivery people who received 3 compliments will be promoted.  The salary of the promoted chef and delivery people will be higher, and the system will record the promotion in their files. |
| check\_taboo | The system will review the comments from the customers. The customer whose comment contains taboo words would receive a warning and the taboo words would be replaced by \*\*\*. The comments with more than 3 taboo words would be blocked. |
| put\_back | The VIPs who receive 2 warnings would be put back to the registered customer and then the system would clear the warnings. |
| de-registered | When a customer receives 3 warnings, the system would inform the manager and the manager would kick the customer out. |

# Supporting Information

## Appendices

### User Interface

The UI for customers provides the menus lists and the login button.

After they log in, they can see their account information, such as Order history, account balance, or log out account.

For logged users, they can select the item from the menus by clicking on the item. There is also an order list button for customers to check what they have chosen. They can choose to delete the item or place an order. When a delivery person delivers an order, they can see the details from the Order history to decide the complaint or compliment the delivery person or chef. And left the evaluation for the order.

The UI for chef provides the order lists with the time of order placed. Display the note adding by the customers for their order. For each order, there is a button to notify the delivery person that the order is ready. Also, the UI provides chef's own menus, so they can edit their menus by themselves.

The UI for the delivery person provides the list order that is confirmed ready by the chef. Each order displays the address of the order and the note from the customer. For each order, provide a button to notify the customers when they delivered the address. Delivery person Then can select the order finish button to notify system order finished.

The UI for manager provides all order information in order history. For each order, it provides the price and the evaluation from customers. Also, a display a button that can connect to the complaint/compliment interface. If it is Complaint the text of title will be color of red, if it is compliment title color will be green. For each case, managers have right to merit the case.

The UI for the main page will display the menus button to connect with the menus interface which shows the menus list and the price. Main page also provides the recommend dish and the discount poster if there is any.