**Alimama**

**Online Pizza Ordering System**

**Design Report**

**For Computer Application**

**Version <1.0>**

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
| <21/04/18> | <1.0> | First version of Design Report for Online Pizza Ordering System | Hongjie Huang,  Xingcheng Zhang,  Youwen Zeng,  Alan Lau |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Table of Contents** | |  |  |
| 1. | **Introduction** | | |  | 4 |
|  | 1.1 | | Purpose |  | 4 |
|  | 1.2 | | Collaboration Class Diagram of the System |  | 4 |
| 2. | **Use Case Analysis** | | |  | 5 |
|  | 2.1 | | Register Account |  | 5 |
|  | 2.2 | | Login/ Visitor Login |  | 6 |
|  | 2.3 | | Pick Location |  | 7 |
|  | 2.4 | | Place Order |  | 8 |
|  | 2.5 | | Rate |  | 9 |
|  | 2.6 | | Chef |  | 10 |
|  | 2.7 | | Registered customer |  | 11 |
|  | 2.8 | | Delivery person |  | 12 |
|  | 2.9 | | Manager |  | 13 |
| 3. | **Entity-Relation Diagram** | | |  | 14 |
| 4. | **Detailed Design** | | |  | 14 |
|  | 4.1 | | User’s function |  | 14 |
|  | 4.2 | | Manager function |  | 16 |
|  | 4.3 | | Delivery person function |  | 17 |
|  | 4.4 | | Chef function |  | 18 |
| 5. | **System Screens** | | |  | 20 |
|  | 5.1 | | Login and Customer GUI |  | 20 |
|  | 5.2 | | Delivery person GUI |  | 23 |
|  | 5.3 | | Chef GUI |  | 26 |
|  | 5.4 | | Manager GUI |  | 27 |
| 6. | **Minutes of group meetings** | | |  | 30 |
| 7. | **The graded 1st phase report (attached)** | | |  | 30 |

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**Design Report**

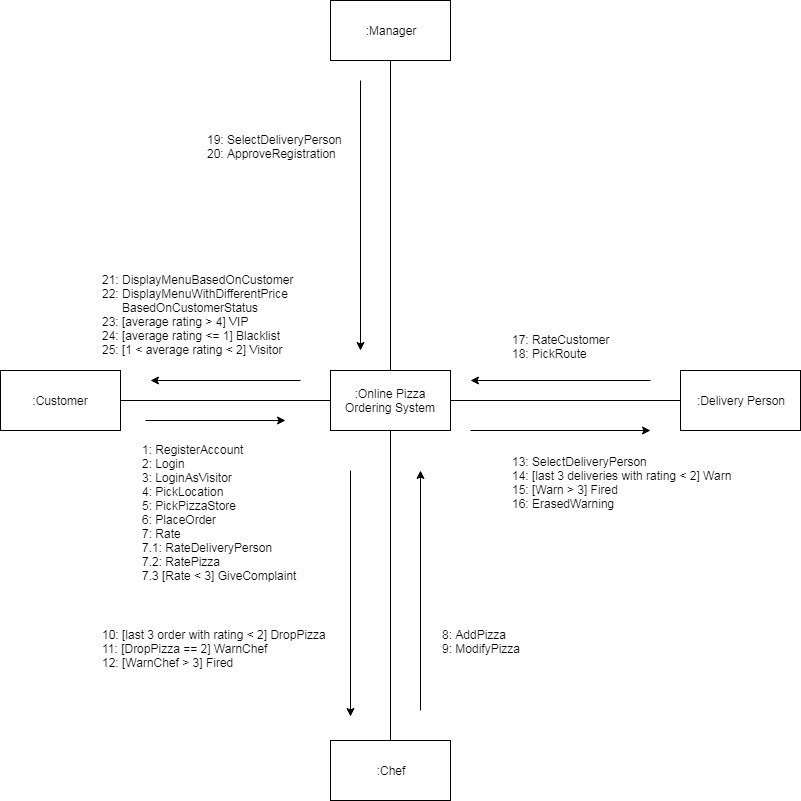
**1.Introduction**

* 1. **Purpose**

The purpose of this document is to present the design description of the Online Pizza Order System using several diagrams including collaboration class diagram, Petri-net diagram, E/R diagram as well as some codes to explain the ideas. This document will cover the overview of our software, the functions of each type of user and various other technical dependencies of it. We will present the general picture of the system by showing all use cases, the interface of the entire system, and the detail design using pseudo-code to delineate the input/output and main functionalities. We will also demonstrate the major GUI screens of the system and a prototype of certain functionality of our own choice. What’s more, we will include the minutes of our group meetings and attach the modified first specification report, and hopefully this document will present our system to the readers in a more visual way.

**1.2 Collaboration Class Diagram of the System**

Overall pizza ordering system Collaboration Diagram

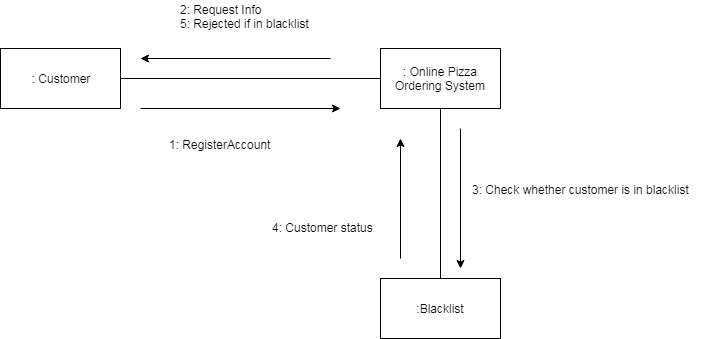
****

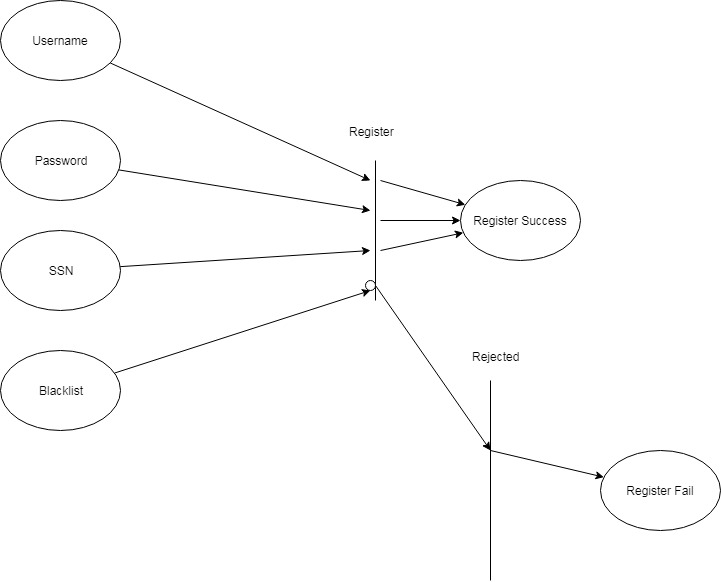
|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**2.Use Case Analysis**

**2.1 Register Account**

A customer can register to become a registered customer to receive special offer. If the username exists in the system, the customer has to re-chose the username. If the ssn provided by customer is in the blacklist, the customer will be rejected. If the customer successfully register then he will have to wait for approval from manager of the store that he register to.

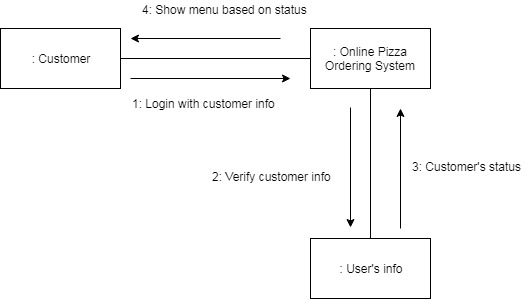
****

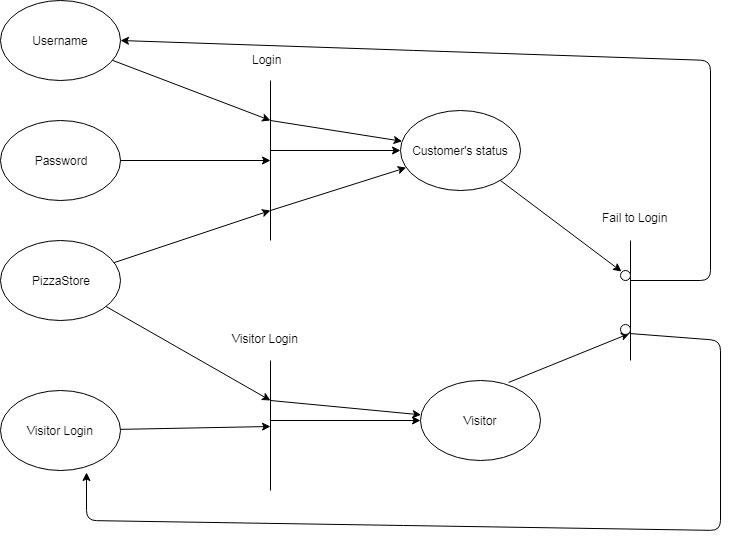
****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**2.2 Login/ Visitor Login**

A returning customer can log in and proceed as registered customer or VIP. Otherwise a customer can click visitor login to proceed as visitor which will have to pay higher price. If the password or username provided by the customer is not correct, the customer will be asked to retry.

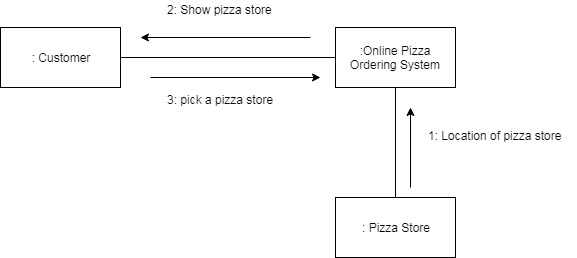
****

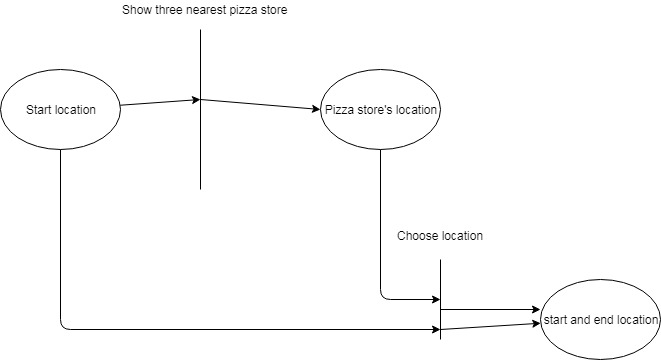
****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**2.3 Pick Location**

A customer must pick a location(where he is at) and a pizza store where he wants to order from.

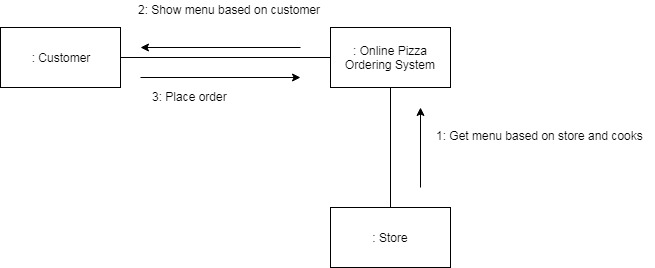
****

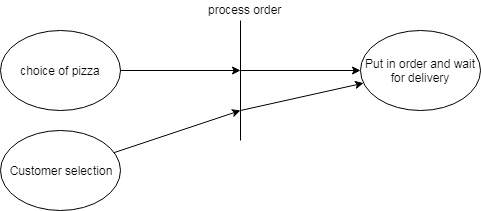
****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**2.4 Place Order**

After the customer log in or use the visitor login, he/she is then provided a list of pizza that he can choose. The listed pizzas are based on the customer’s order history or popularity of the pizzas if he is a visitor.

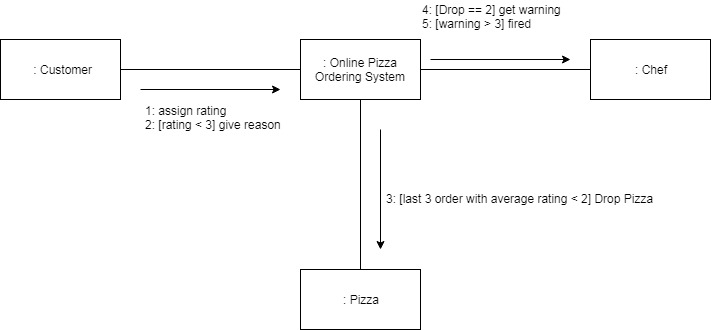
****

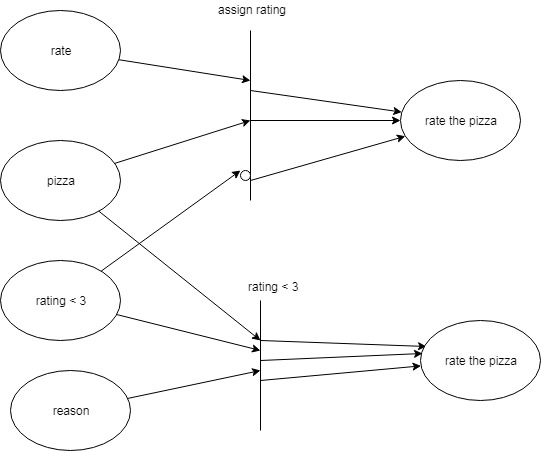
****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**2.5 Rate**

After the customer finished his/her order, he will have to rate all the pizza that he ordered. Any pizza that have rating less than 3 must be given a reason, which will be handled by the manager.

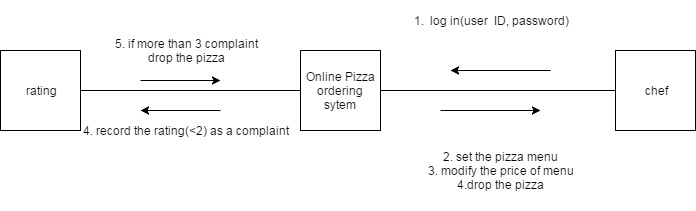
****

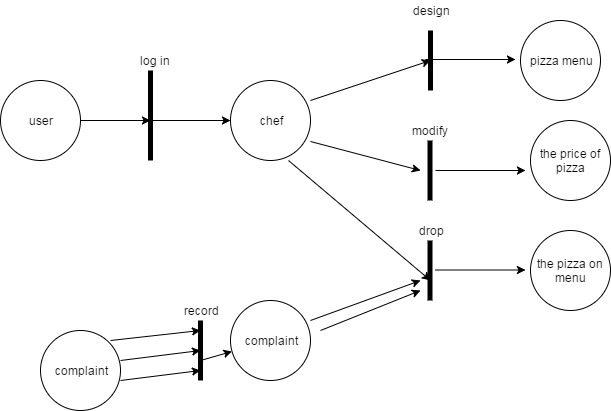
****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**2.6 Chef**

The chef can design the pizza menu, modify the price of pizza and drop the pizza on the menu.The customer will rate the pizza after finishing the orders.A pizza receiving average rating less than 2 in the last 3 orders will be dropped. The cook whose pizza was dropped twice will be warned. A cook warned more than 3 times is laid off.

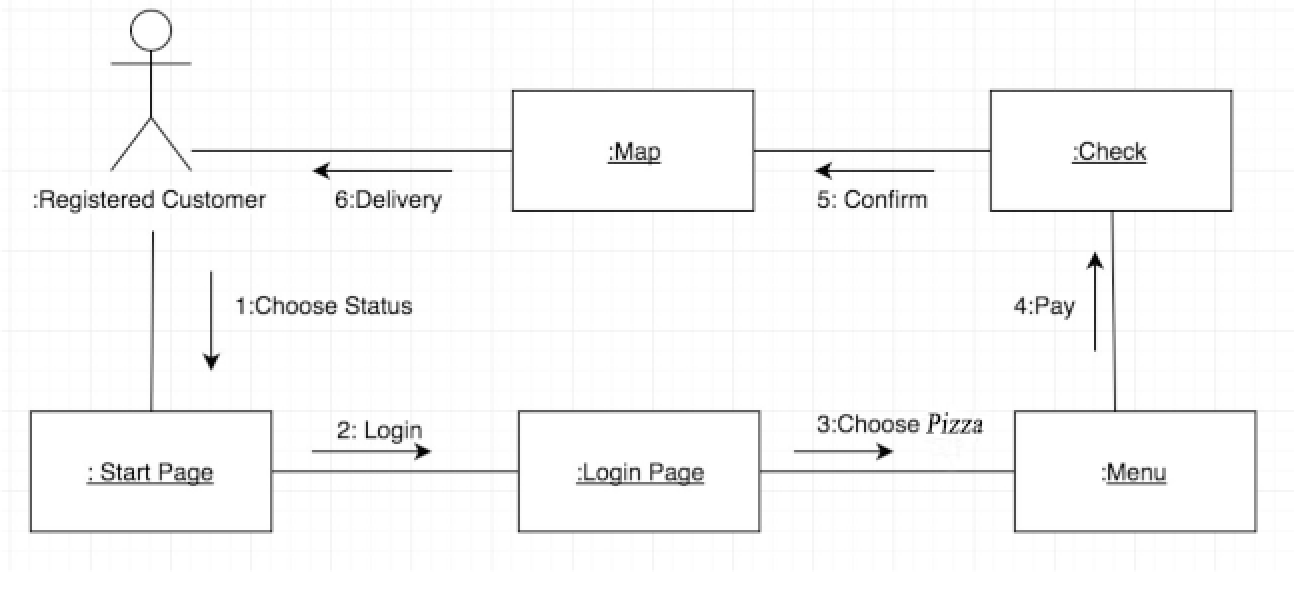
****

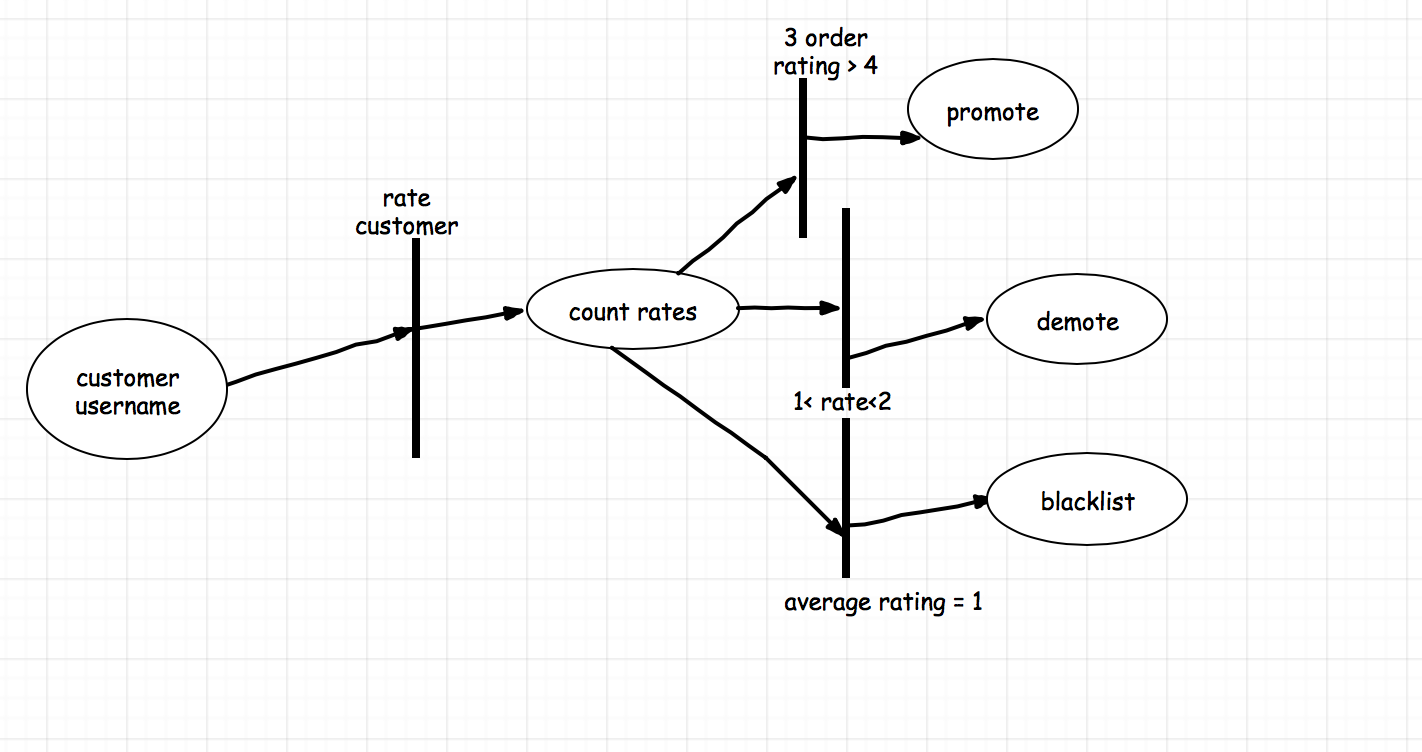
****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

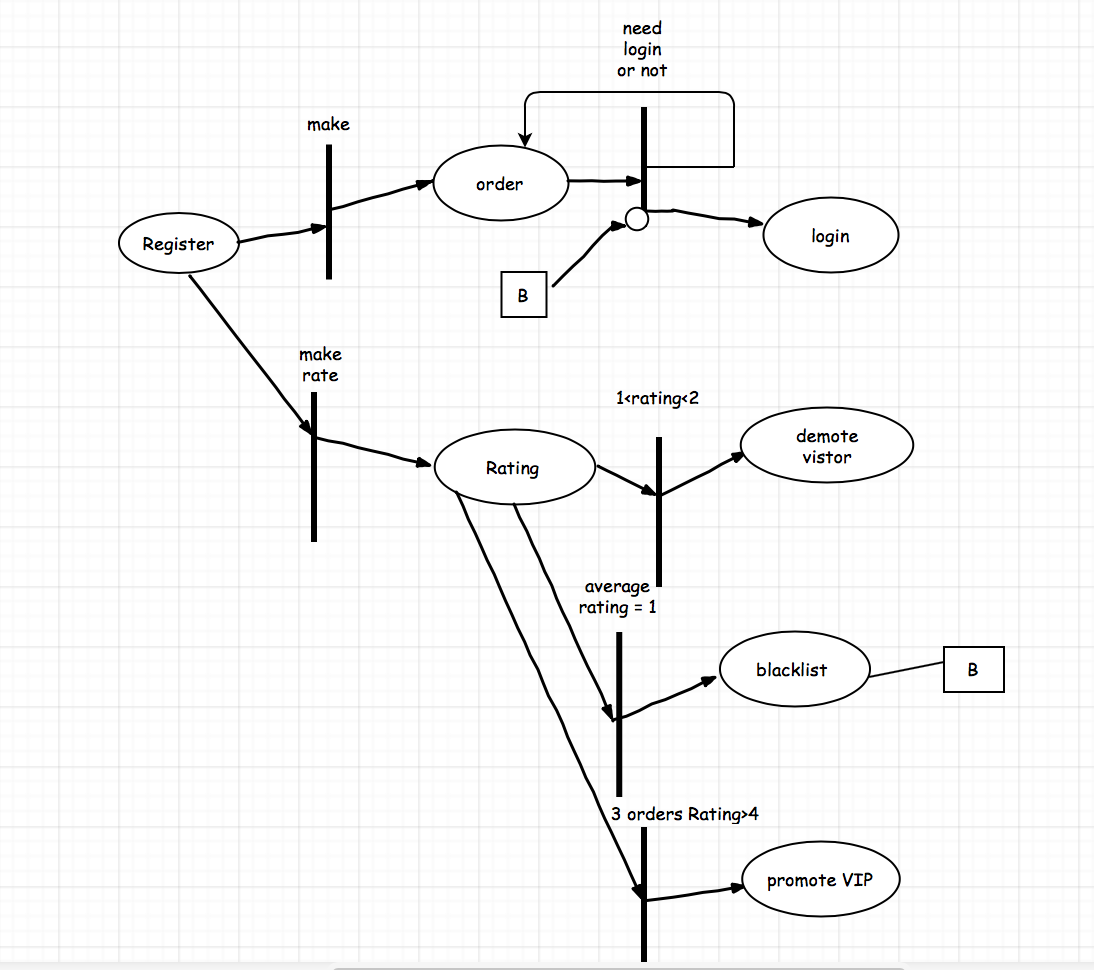
**2.7 Registered customer**

The registered customer at the start page will first chose their direction on the map, and the system will show three closest pizza stores and the customer can pick one, and then customer can choose to whether login in or not, and then pick their pizzas and make payments based on their customer type, then manager will pick a delivery person to deliver the order. The customers might be promoted or demoted based on their ratings.

****

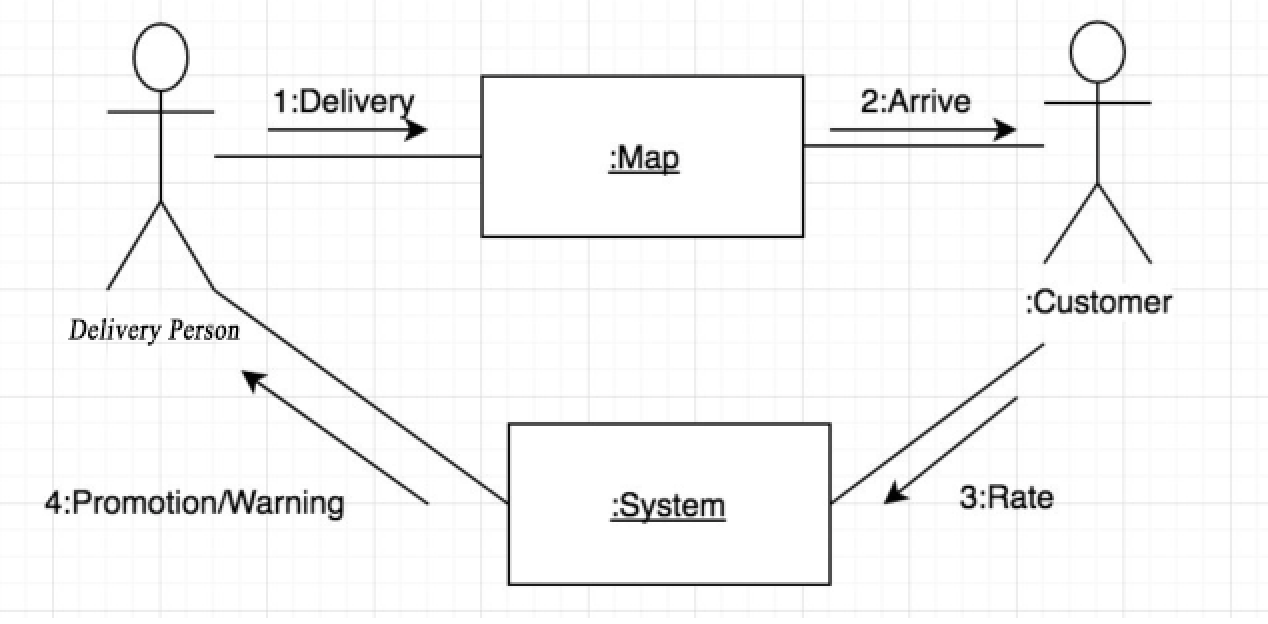


|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

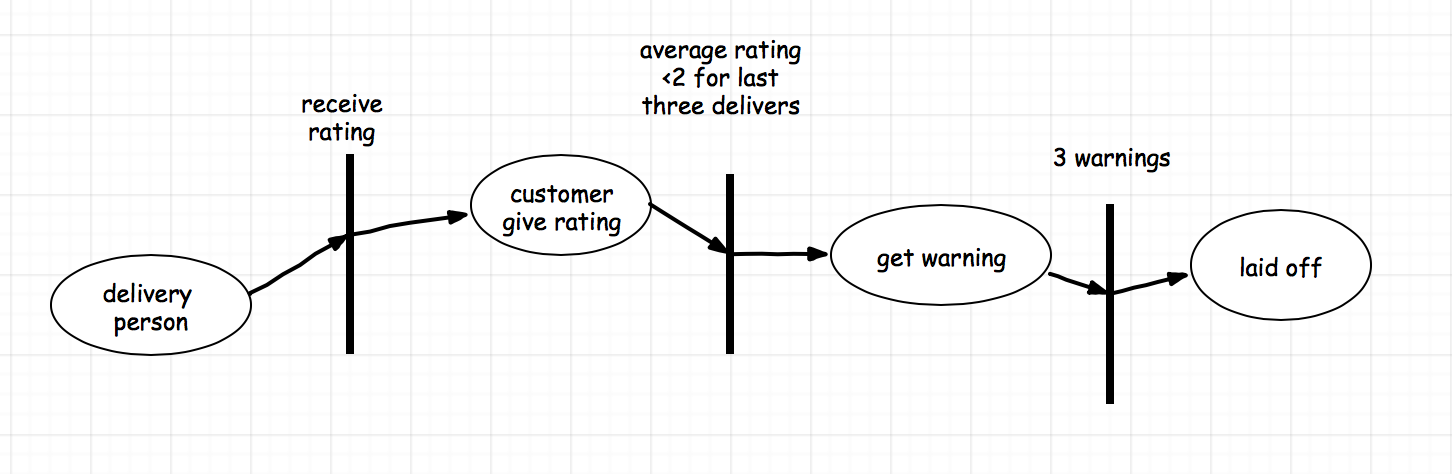
****

**2.8 Delivery person**

The delivery person can use the system map to find the shortest path to the customer, and he will rate the customer as well being rated by the customer. And their ratings will be viewed by the manager to promote/demote the customer, warned/fired the delivery person.

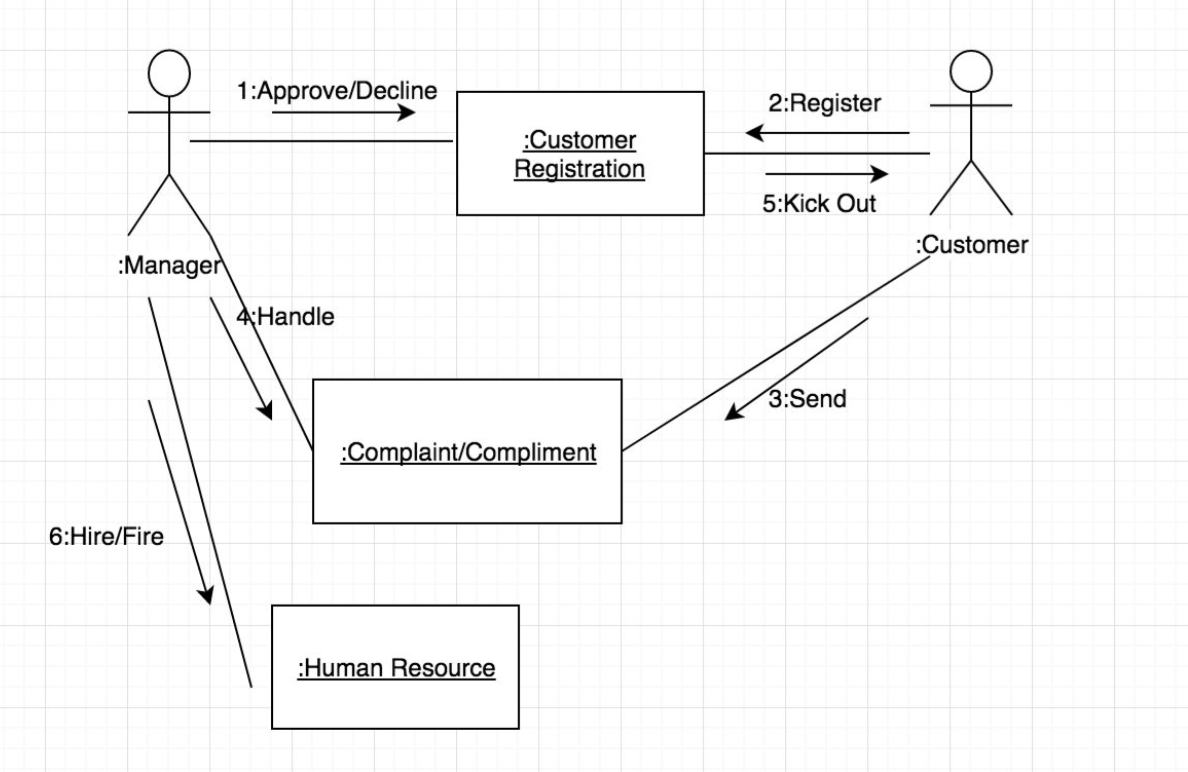
****

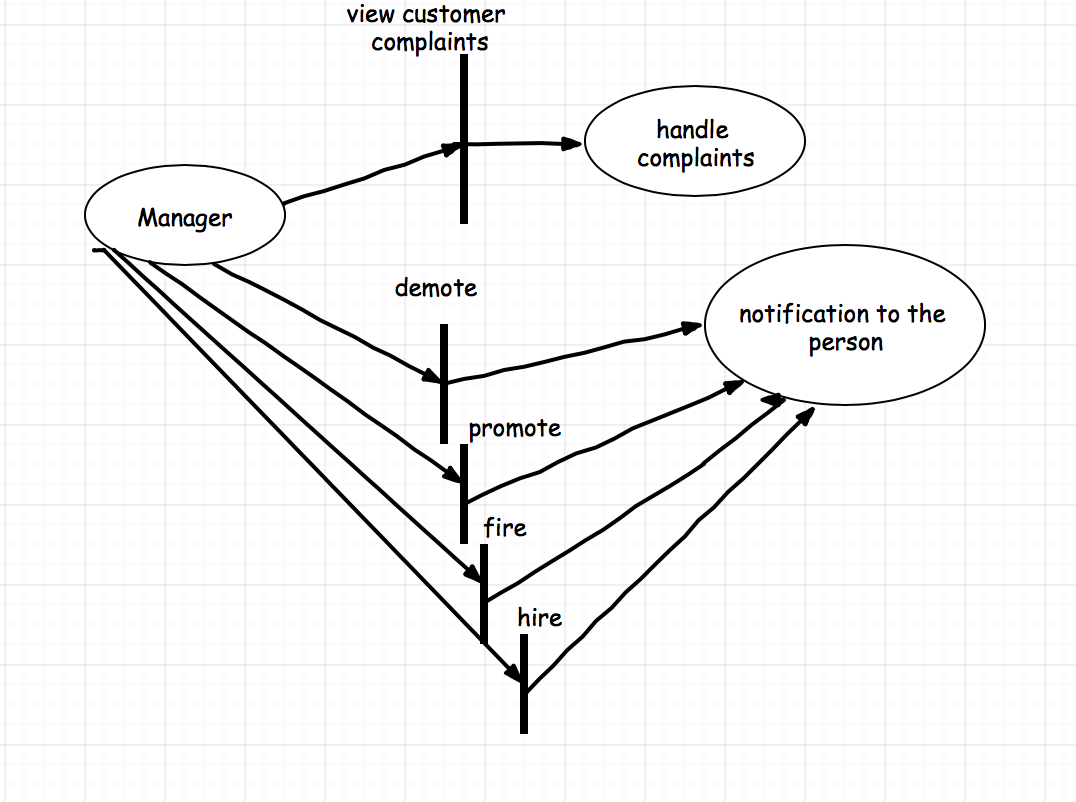
|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

****

**2.9 Manager**

Managers/superusers for each store can decide pays of cooks and delivery people, and handles complaints and managements of customers. They also have the ability to promote/demote a customer, and hire/fire a employee.

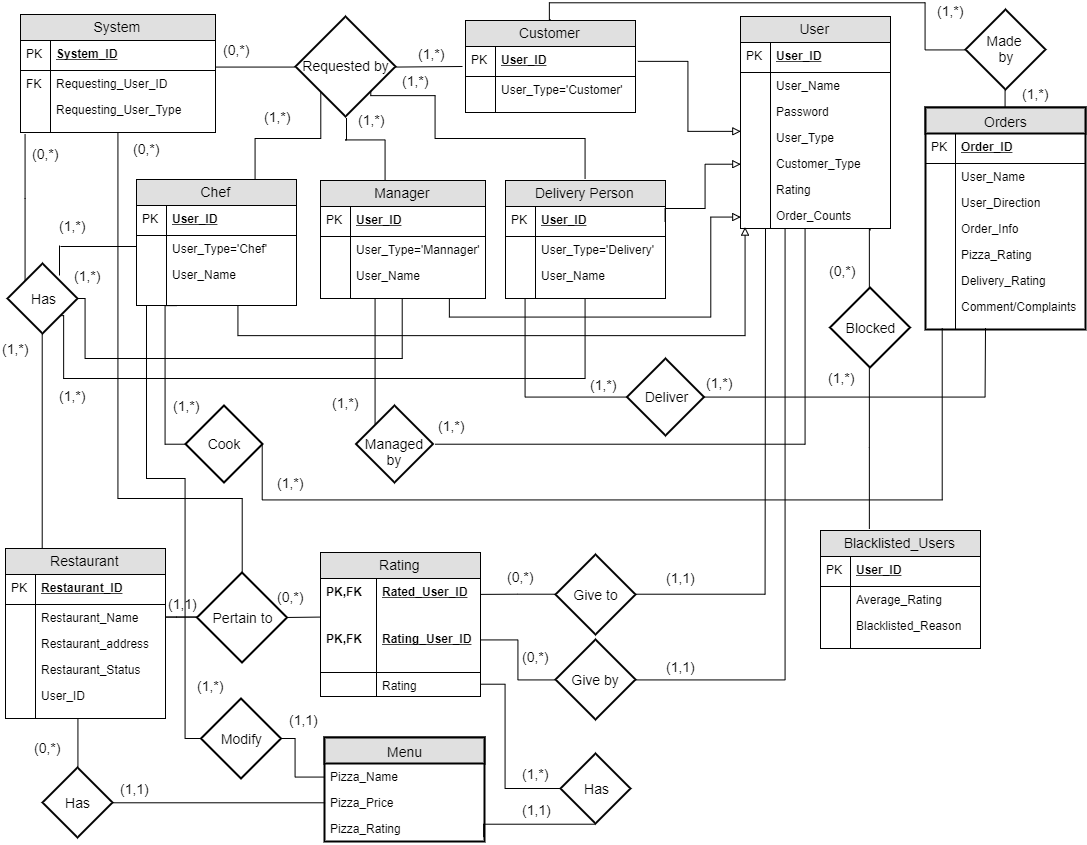
****



|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**3.Entity-Relation Diagram**

Below is the E/R diagram describing the overall system of our Online Pizza Ordering System. We used Pandas to simulate the online database, so some attributes are being presented in the format of the E/R tables.



**4. Detailed Design**

**4.1 User’s function**

def register():  
 # input: username, password, ssn  
 # output: write the user's information into the database and wait for manager to approve  
 if ssn in blacklist:

prompt"you are blacklisted by 'pizzastore' please contact them if you believe"

"this is a mistake"  
 return false

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

elif username is used:  
 prompt"please use another username"  
 return false

else:  
 write the user's information into the database and wait for manager to approve  
 return  
  
def login():

# input: username, password

# output: return username and status  
 if username doesn't match:  
 prompt "the username is incorrect"  
 return false

elif password doesn't match:

prompt "the password is incorrect"  
 return false  
 elif both match:  
 return username, status  
 # the username will be used so we know who made the order  
 # the status is used so we know what price should be shown  
  
def visitor\_login():  
 # input: None  
 # output: return identity = visitor  
 # this function will be triggered by a button, once the button is clicked,  
 identity = visitor  
  
def employee\_login():  
 # input: username, password  
 # output: username, position  
 employeeinfo = pandas.read\_csv('employeeinfo.csv')  
 if username not in employeeinfo:  
 return false  
 elif password not in employeeinfo:  
 return false  
 else:  
 return employee, position   
  
def sort(): # this will take all the location of the pizza store and find the nearest three of them  
 # input: location of customer  
 # output: list containing three nearest pizza store's location  
 all\_store\_location = pandas.read\_csv("store")  
 x, y = customer\_location  
 answer = []  
 while i < 3:  
 closest = all\_store\_location[0]  
 for store in all\_store\_location:

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

if store closer than closest:  
 closest = store  
 answer.append(closest)  
 all\_store\_location.remove(closest)  
 return answer  
  
def place\_order():  
 # input: customer's location  
 # output: write the order info into the database and wait for manager to assign deliver person  
 orderinfo = Dataframe(pizzastorelocation, customerlocation, None)  
 open pendingorder.csv as pdo  
 insert orderinfo into pdo  
 close pdo   
  
defshowmenu():  
 # input: menu, customeridentity  
 # output: display menu  
 if customeridentity == visitor:  
 display most popular pizza  
 elif customeridentity == registered customer:  
 display his three last order pizza with price = price \*= 0.9  
 elif customeridentity == VIP:  
 display his three last order pizza with price = price \*= 0.8   
  
def ratepizza():  
 # input: list of pizza that were ordered

# output: write the rating into the database

For eachpizza in listofpizza:

display to the screen

get rating for each of them

write it to the database

**4.2 Manager function**

def Manager:

if register in blacklist:

return deniedRequest

else:

print (Congratulations!)

return userlist

def handle\_warning:

if identity = customer && 1< rating for visitor < 2:

return demotelist

if identity = customer && average rating >4:

return promotelist

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

if identity = (delivery person or cooker) average rating < 2 and count = 3:

return laidOff

def deniedRequest:

if manager want to erase warning:

return register in blacklist to userlist

else:

print(“Registration denied !”)

return false

**4.3 Delivery person function**

#Delivery person will give rating to customers after the deliver

Delivery Person->Online Pizza Ordering System

- Input: Username,Password, Type(Customer or employee)

- Output: List of orders that currently hold to screen, the edited rating of customers

- Function: Allow Delivery Person to give rating to a customer

def Give\_Rating(Customer, Order\_Counts,New\_Rating):

If Customer is on the order list and not rated:

New\_Rating=Give\_rating.get();

With Open Users.csv:

for User.Customers=Customer:

If Order\_counts>0:

Ave\_Rating=((Users.Rating\*Order\_Counts)+New\_Rating)/(Order\_Counts+1))

else:

Ave\_Rating=New\_Rating

Users.Customer.Rating=New\_Rating

Save changes to Users.csv.

# Delivery will use the system to find the shortest path from store to the customer’s position

Delivery Person-> Online Pizza Ordering System

- Input: Username,Password, Type(Customer or employee)

- Output: List of orders that currently hold to screen, the shortest path to the selected order

- Function: Allow Delivery Person to find the shortest path for a order

def Find\_Path(Customer, Start, Store,Rating,Map):

Order\_info=Orders.read()

Start=Store.direction

End =Order\_info.customer\_direction

Map=Random\_Map\_Based\_On\_Traffic

Final\_Path=Dijkstra's algorithm(Start,End, Map).return()

for nodes in Final\_Path:

connect each node

Print Final\_Path to screen

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**4.4 Chef function**

Chef user ->online pizza ordering system

Chef interface: 1. pizza menu

2. modify the price of pizza

3. design the pizza menu

Each choice is a function

1. Pizza menu

The name and price of pizza will beforehand be stored in the csv file

if the user click the bottom ‘pizza menu’

open the csv file show the data that store in the file

else: close bottom (exit current window)

2. Modify the price of pizza

In this function, it requires the chef to enter two inputs:

input1: the pizza that chef want to modify

input2: the new price of pizza

if (check the target pizza whether or not exist in the menu/file)

modify the price of target pizza

else: show error, no such pizza

3. Design the pizza

input1: enter the new pizza name

input2: enter the price of pizza

and then open the csv file and save the data of both input to the csv file.

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

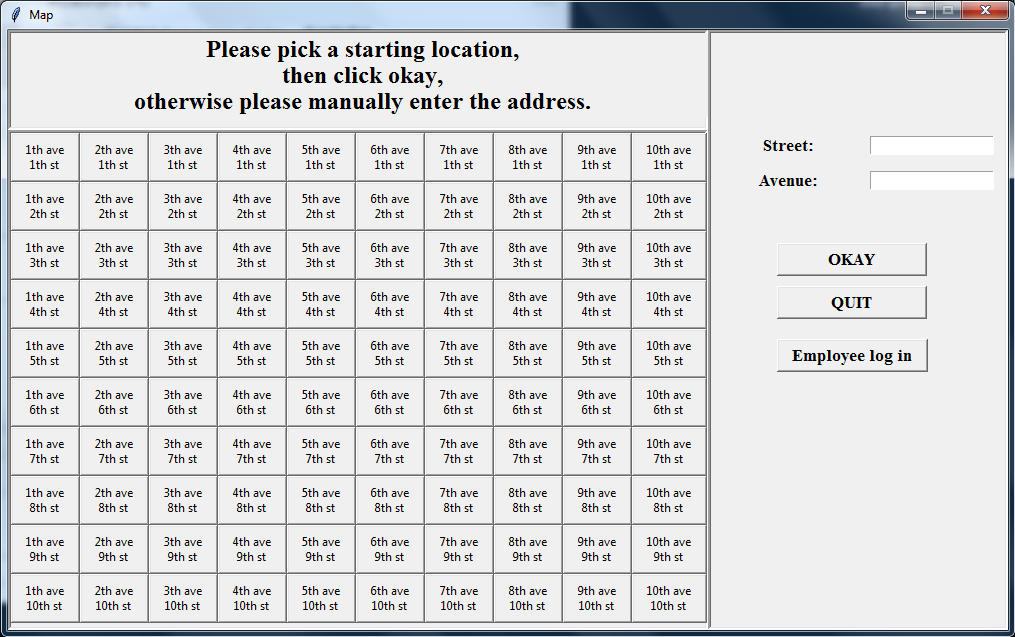
**5. System Screens**

Note that modification and improvement of the system will be performed, and the

the pictures here are only for references, please make the actual application as the standard.

**5.1 Login and Customer GUI**

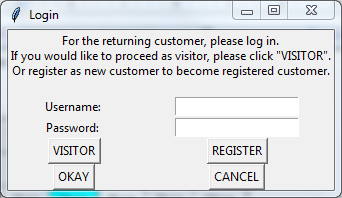
This the first screen that the user will see. This will ask for user’s location. The user can either pick a point on the map or manually enter his address. After the button “OKAY” was pressed, three pizza store from different location will show up. This can be seen at the second picture

****

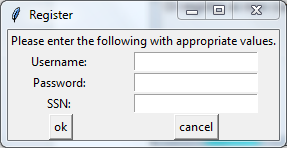
The second picture show multiple pizza store location for customer to choose from. The customer will pick one of them by clicking the location. Then the customer will be prompted to log in or to proceed as visitor. This can be seen from the third diagram.

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

****

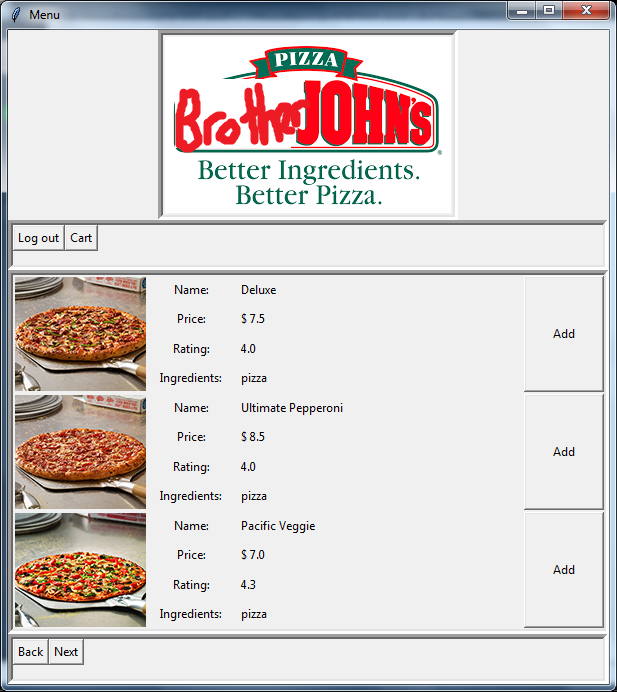
****

Below is the register interface, the customer will have to fill in all the fields to proceed. The ssn is needed to check whether or not the customer is in blacklist or not since no one can have multiple ssn.

****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

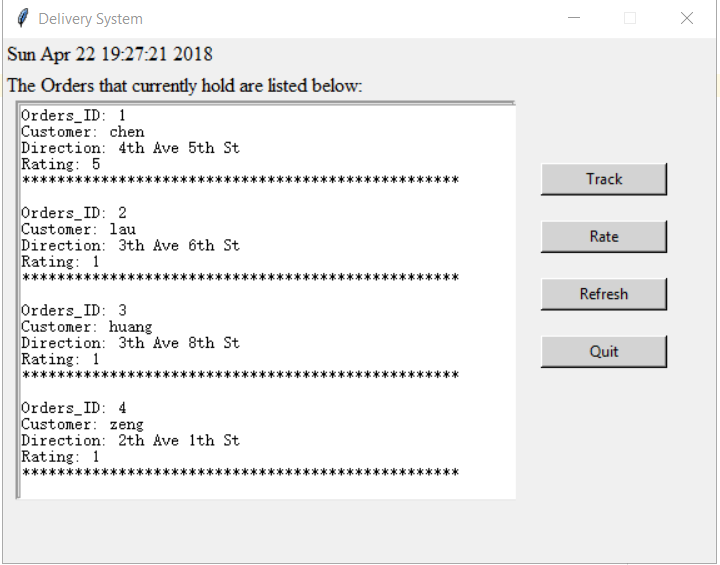
Below is the screen that the customer will see after he login. If the customer is a registered customer, the first three pizza is the last three pizzas from his past orders. If the customer is login as visitor, the pizza will be shown according to the rating, from most popular to least popular.



|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

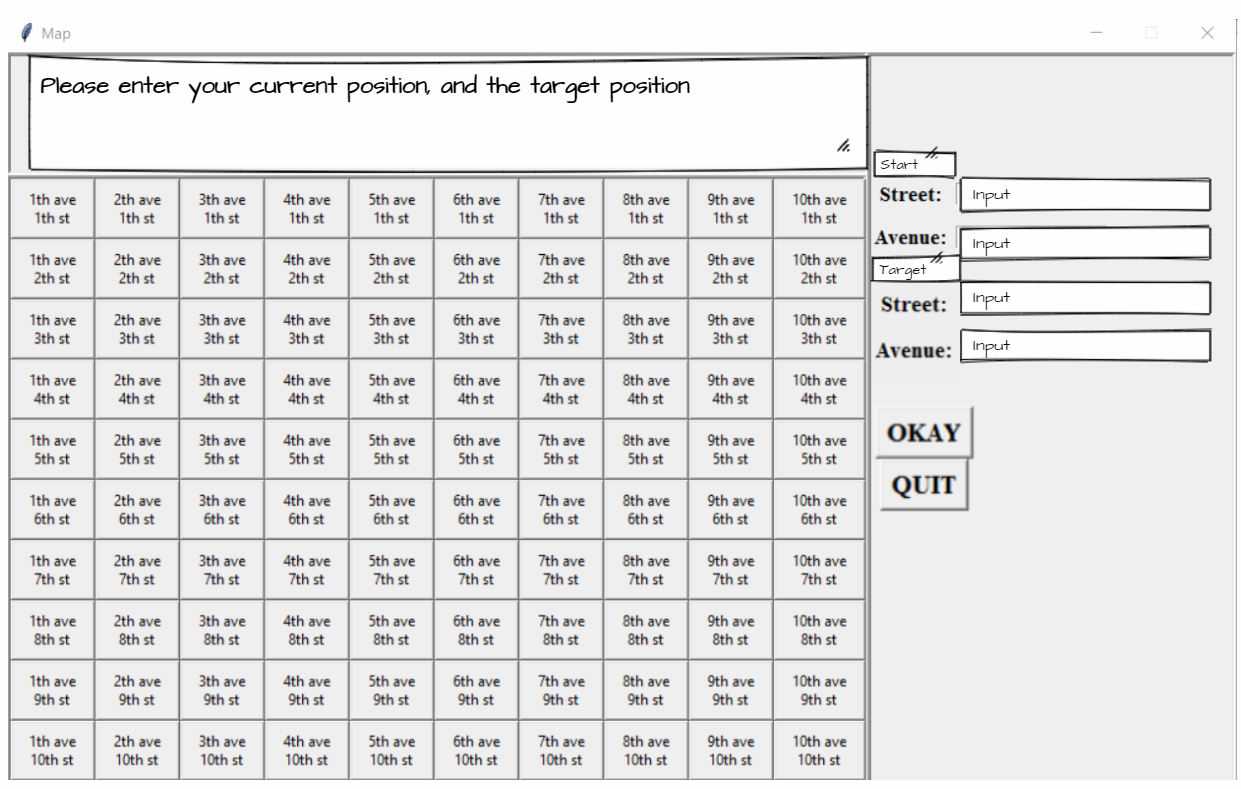
**5.2 Delivery person GUI**

The user will login in to our Online Pizza Ordering System by entering their username and password, after that our system will determine the user type of the user. If the user type is “Delivery Person”, our system will show up the following screen. The left side will display the orders that currently hold by the system.The right side are buttons of functions that user can do.

****

The delivery person can click the track button, which will show up the same screen as we have described as the first screen, where delivery person will enter the current position of the store, and the target position, then click the okay button to confirm.

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

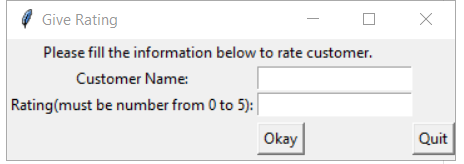


After the delivery person enters the information, for example, if the start position is 5th Ave 3th St, and the target position is 7th Ave 5th St, then the system will display the shortest path from the start position to the target position, which is the green line as shown in the picture.

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |



Once the rate button is clicked, the window below will be showed up, and the delivery person will enter the name of the customer he want to rate, and the rating that he will give to the customer, which will be a positive number from 0 to 5, and then this rating is used to calculate the average rating of the customer based on his/her number of orders that already made.

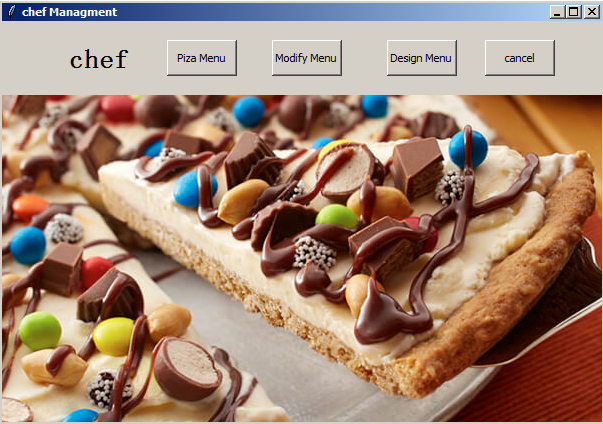


|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

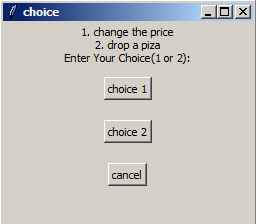
**5.3 Chef GUI**

Below is the main interface of chef.

There are four choices: pizza menu, modify menu, design menu, cancel bottom

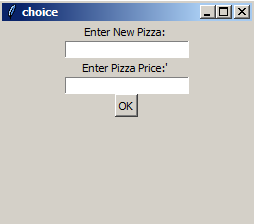
****

below is new window after the chef click the “Modify Menu”. There are 3 choice here. choice 1 go to the change the price of pizza, choice 2 is to drop the price

****

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

Below picture is the window after the chef click “design menu”. The chef can add new pizza to the menu and decide its price.

****

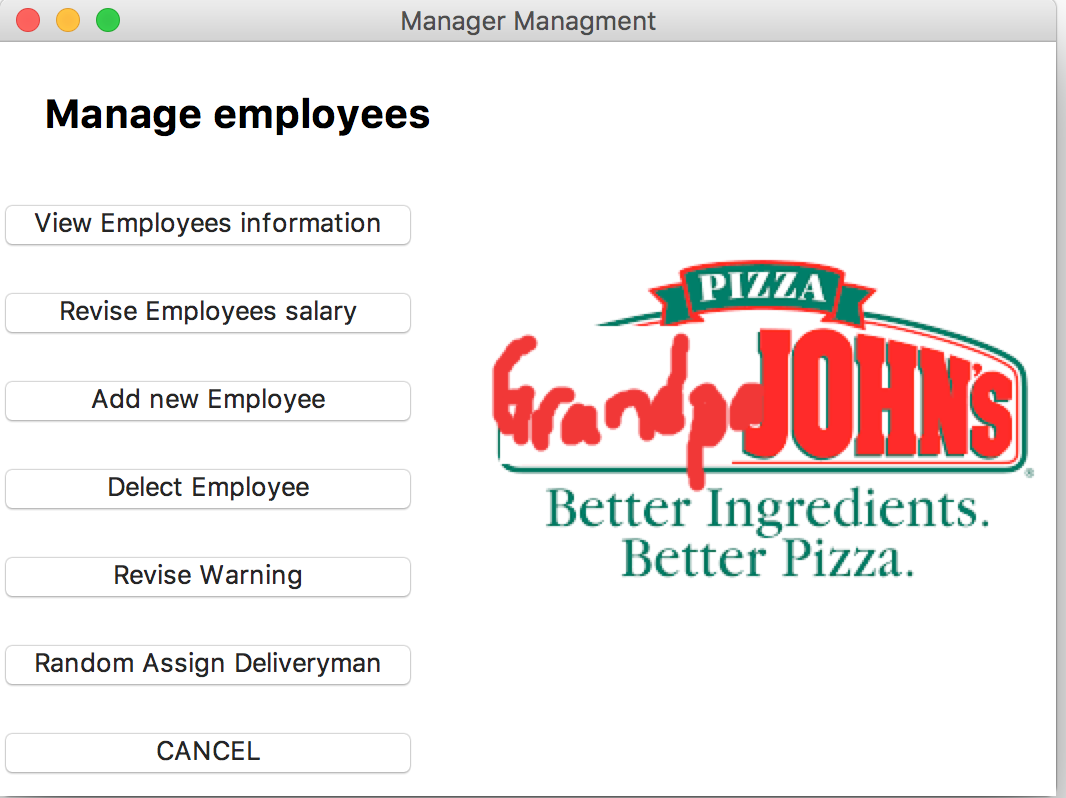
**5.4 Manager GUI**

Below is the main interface of Manager.

There are 6 buttons in this interface GUI, which are View Employees Information,

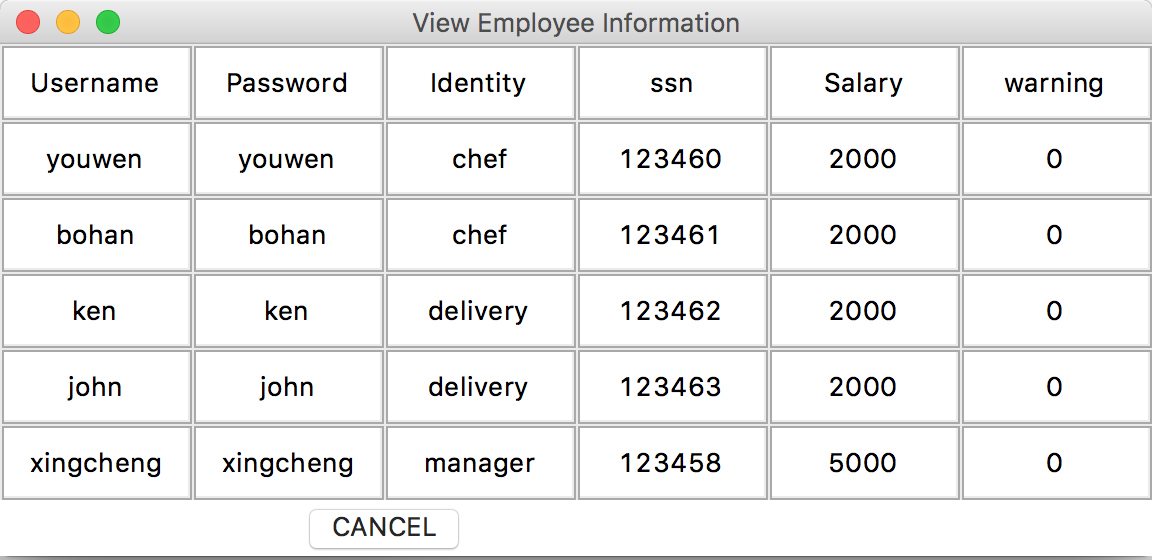
Revise Employees salary,Add new Employee, Delect Employee, Revise Warning and

Random Assign Delivery Man. Different store will show different Manager interface GUI.

****

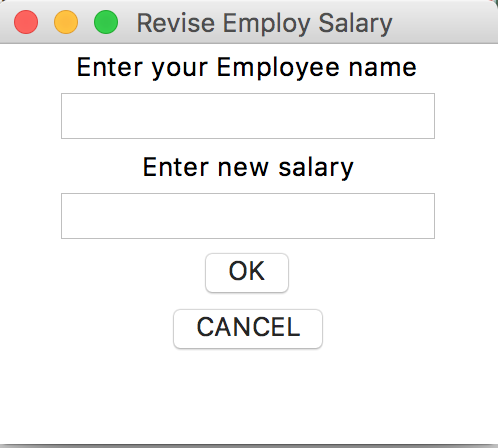
|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**View Employee Information**

****

This interface will show all Employee information, we can change information by using these informations.

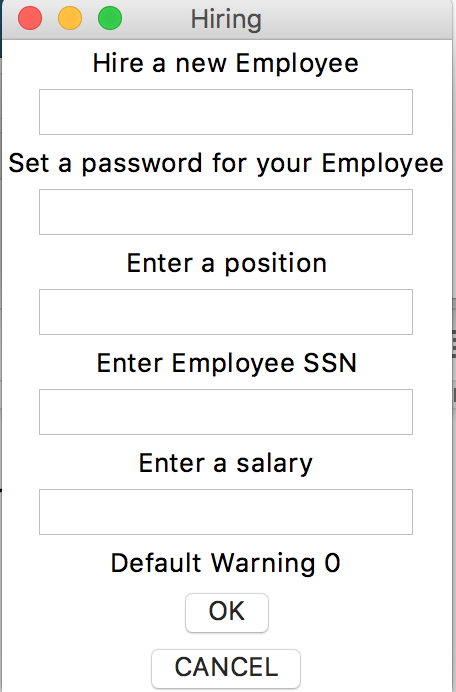
**Revise Employ Salary Interface**

****

We can change Employee’s salary base on his contribution in the store.

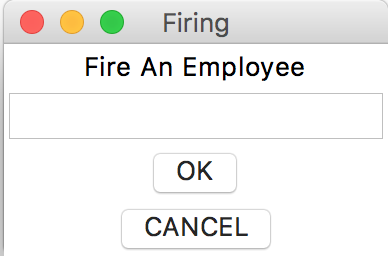
|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**Add new Employee**



Manager can add new Employees to specific store.

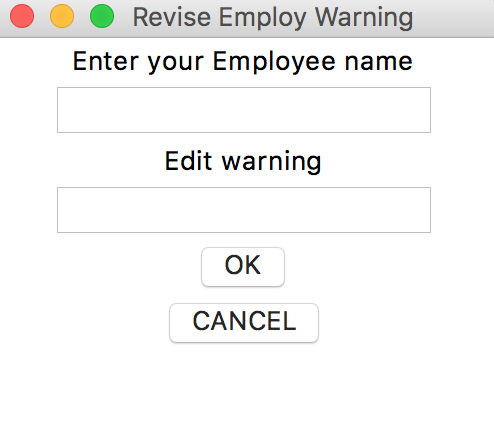
**Delete Employee**

****

Manager can delete Employees.

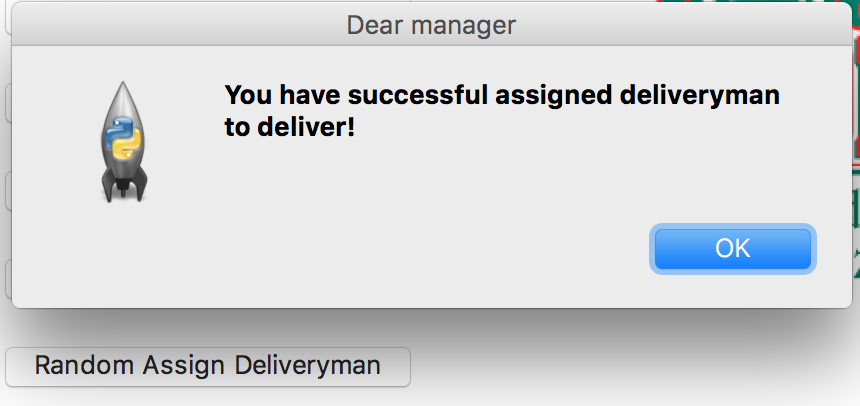
|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**Revise Warning**

****

Manager can revise warning.

**Random assign delivery man**

****

Manager can assign delivery man to accomplish the order, in our case, we decide to use random function to make each order fast.

|  |  |  |
| --- | --- | --- |
| <Online Pizza Ordering System> | Version: | <1.0> |
| Design Report | Date: <21/04/18> | |
| <First Draft> |  |  |

**6. Minutes of group meetings**

|  |  |  |
| --- | --- | --- |
| **Date** | **Topic** | **Time** |
| 04/09/2018 | Basic Database | 0.5 hour |
| 04/13/2018 | Customer Login | 1 hour |
| 04/14/2018 | Manager GUI | 2 hours |
| 04/19/2018 | Map with shortest distance | 4 hours |
| 04/21/2018 | Balance Check | 3 hours |

**7. The graded 1st phase report with responses to the comments and  
 possible updates(attached)**

The grade of 1st phase report was 88/100

Received Comments:

“The arrows should be point towards the use case that is being included! Extended. You don’t have to include all the little details line what happens when a user has a specific rating, you can just have one use-case for ratings. It’s better to include the little details in the sequence or collaboration diagrams!”

Some corrections have been made for the graphs to the first phase report, and the corrected version has been attached.