
Online Restaurant

Online Restaurant Design Report For Web Application

**Version 1.0
Group K**

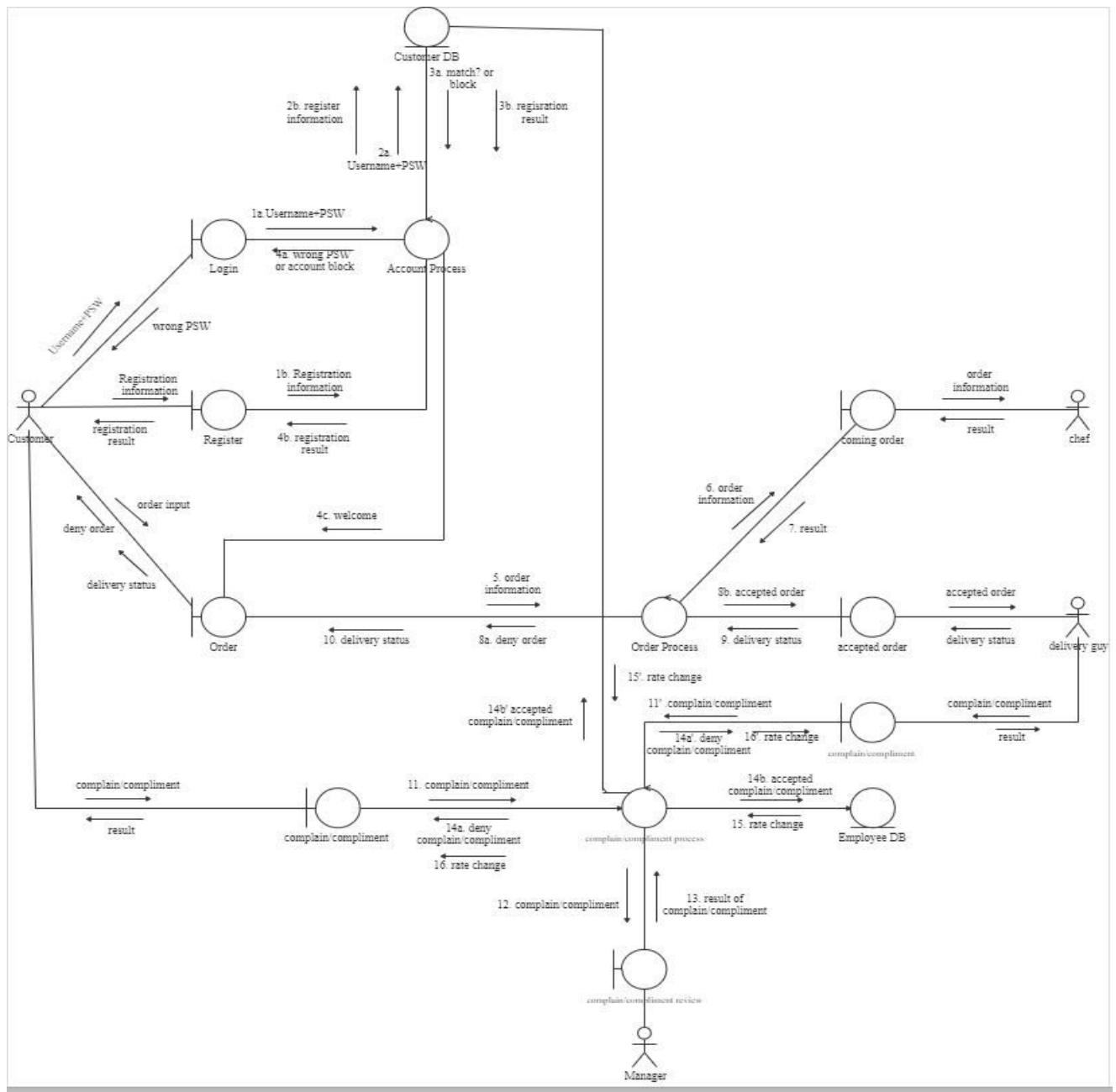
Table of Content

Introduction	2
All use case	5
Use Case Diagram	5
Use Case: Login as user, Login as guest / register	6
Users and VIP users can order and file a rating	6
Manager User case	7
Chef User Case	8
Delivery People User Case	8
Customer User Case	9
Structure of the project	10
Manage review collaboration class diagram	12
E-R Diagram	13
Detailed Design	13
System Screens	16
Group Meetings	16
Github	17

1. Introduction

The purpose of this project is to build an Online Restaurant management system which include features for food ordering and food delivering. In addition, this system will help managers to handle daily restaurant issues such as firing/hiring employees, handling customers' compliments and complaints. This system also allows the customers to browse and order their food, and allows them to file a rating(1-5) everytime they order their food. This document will also cover the hardware, software, and various other technical dependencies.

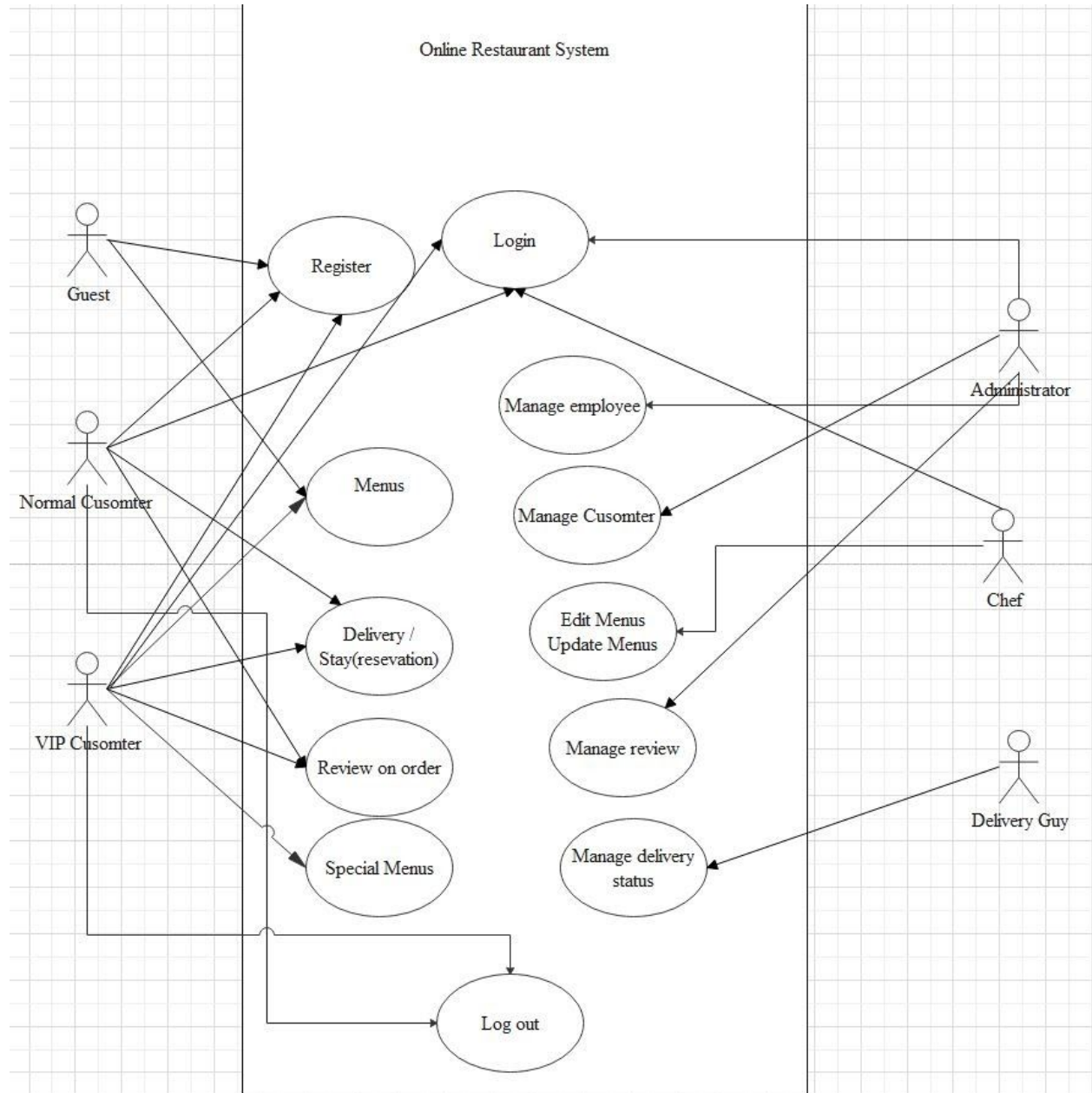
Below figure shows an overall system by using the collaboration class diagram



1. It will display a screen asking the user to login, register as a new customer.
 - 1a. If the user chooses to login, it will ask the user to insert login ID and password.
 - 1b. If the user chooses is not yet registered, users have to register with their user id, password, and personal information.
2. Check the information in Customer Database.
 - 2a. The insert Username and password will run the account process function, and go into the Customer Database to match with the data and check if the account is blocked by the manager or not.
 - 2b. For the register information, it will also go into the Customer Database to see if username exists or not and input user's information into the customer database.
3. Return the response from Customer Database to the account process
 - 3a. Wrong password or username.
 - 3b. Return the result of registration.
4. Account process will take the feedback from the customer database.
 - 4a. Tell the user their input username or password is wrong.
 - 4b. Return the result of registration.
 - 4c. Successful signed in.
5. Ask users for order information and it will be send to Order process center
6. Chef received the order.
7. Chef sent back the result whether the order can be made or not.
8. Process center will handle the result:
 - 8a. Deny of ordering due to out of resources.
 - 8b. Accepted order and will go to the delivery guy.
9. Delivery status sent to Process center.
10. Order process center will send the status of the order to the customer.
11. Customers can have complaints / compliments to the delivery guy or the chef, it will be send to review process center
- 11'. Delivery guys can insert complaints / compliments to the customer, it will be send to review process center
12. Complaint / compliment will be sent to the Manage to decide whether accept or decline.
13. Result sent back to the Review Process Center.
14. The Process Center will take the result of the decision from manager
 - 14a. Deny complaint / complaints will be informed to the customer or delivery guy.
 - 14b. Accepted complaints / compliments will be saved into the customer database or employee database.
15. Employee rating will change in the Employee Database.
16. Customer rating will change in the Customer Database.

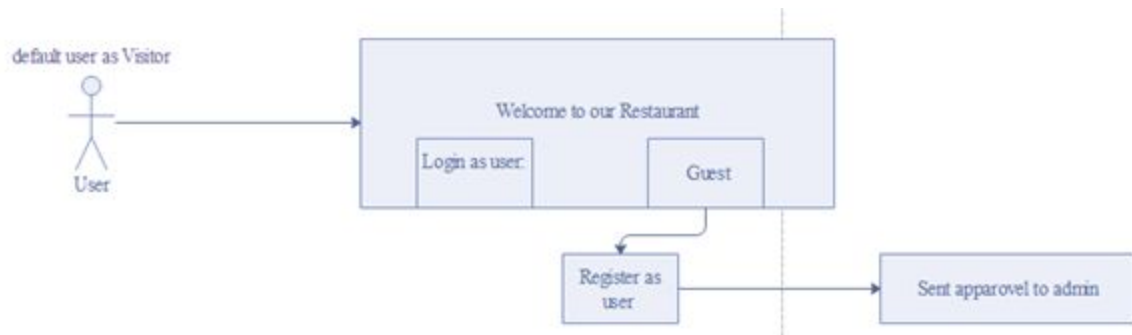
2. All use case

Use Case Diagram



Use Case: Login as user, Login as guest / register

Diagram:



Description: anyone who browse the website will set as default : Visitor / Guest
Step By Step Description:

1. Visitor / guest will get to choose to stay as guest or register as user
Visitors can only visit the web.
2. Visitors can choose to register by sending an application to super users and wait for approval.
3. Visitors can browse the menus and ratings.

Users and VIP users can order and file a rating

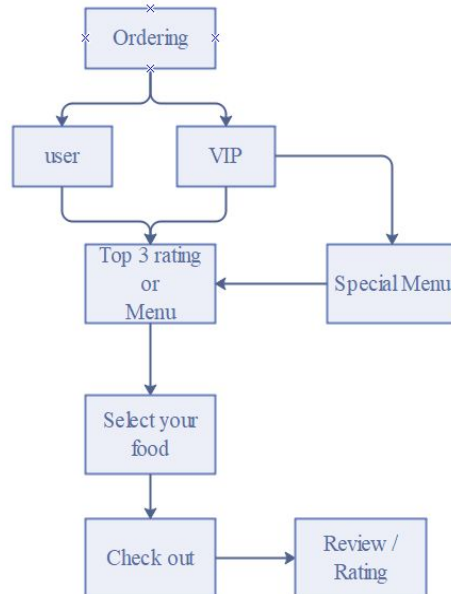


Diagram:

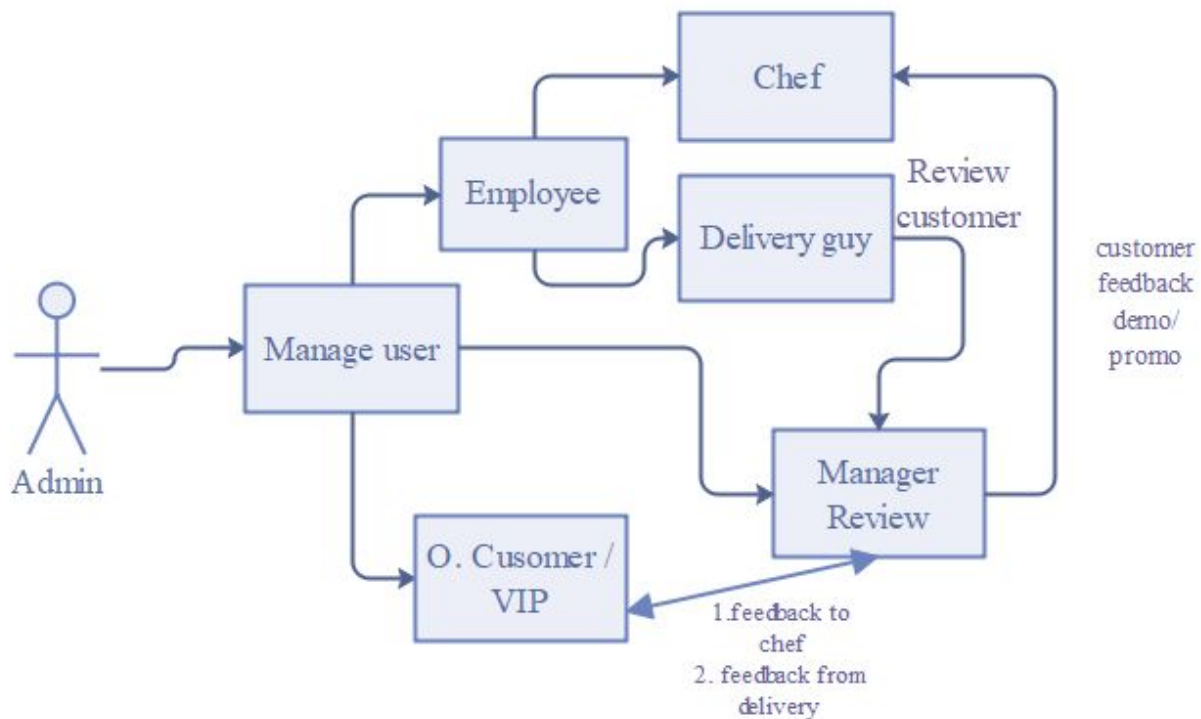
- A. Customers(Users) can choose to order a dish from the menu which is made by one of the chefs. Vip users have special dishes and they can have discounts for their dish.
- B. Once the food is ordered, the system will subtract the money from the user's account. If the user doesn't have enough deposit, the transaction will freeze

until the user deposits more money.

- C. Users can choose where they want to eat their food by either selecting delivery, eating in the restaurant or pick up.
- D. Customers can choose to file a compliment or complaint to the chef or to the delivery person after they receive the food.

Manager User case

The managers can manage all of the other users, include ordinary customers, vip customers, visitors, chefs and delivery people



- A. Complaints/Compliments from customers/delivery people
- B. Manager need to view all the complaints and compliments
 - If received a complaint :
Manager need to made the final decision to dismiss the complaint or convert to a warning and inform the impacted party
 - If received a compliment :
Manager needs to check the compliment, if the compliment is without any merits or reasons, the manager needs to send a warning to the customer/delivery people who are sending this compliment. Otherwise, the receiver will be rewarded.
- C. Manager needs to clear the deposit and close the account for those customers

who are kicked out of the system or choose to quit the system.

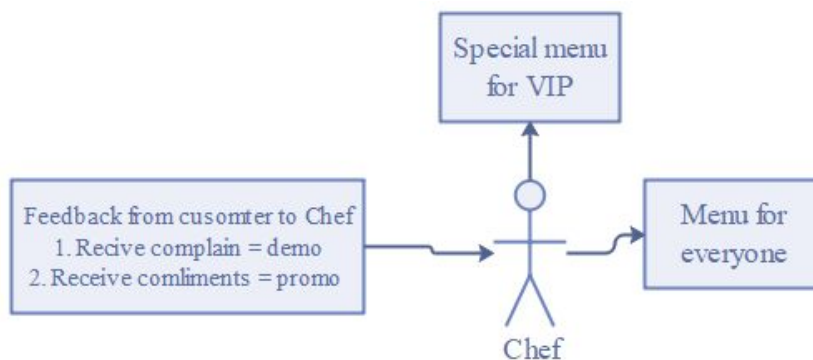
D. Manager keeps a taboo list words

- If customer use one of those word, giving a warning to the customer and replace that word by *.
- If a message appears 3 words from the taboo, block that message.

E. Demote/Promote staffs

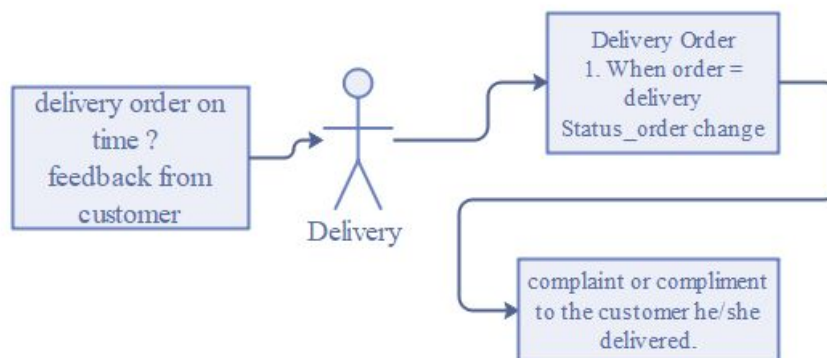
- If staff received 3 complaints or no order for 3 days, will be demoted, and a staff demoted twice is fired.
- If staff received 3 compliments will be promoted.

Chef User Case



- A. Chef independently decided the menu.
- B. Set the descriptions and keywords for the customers.
- C. Received compliments to promote.
- D. Received complaints to demote.

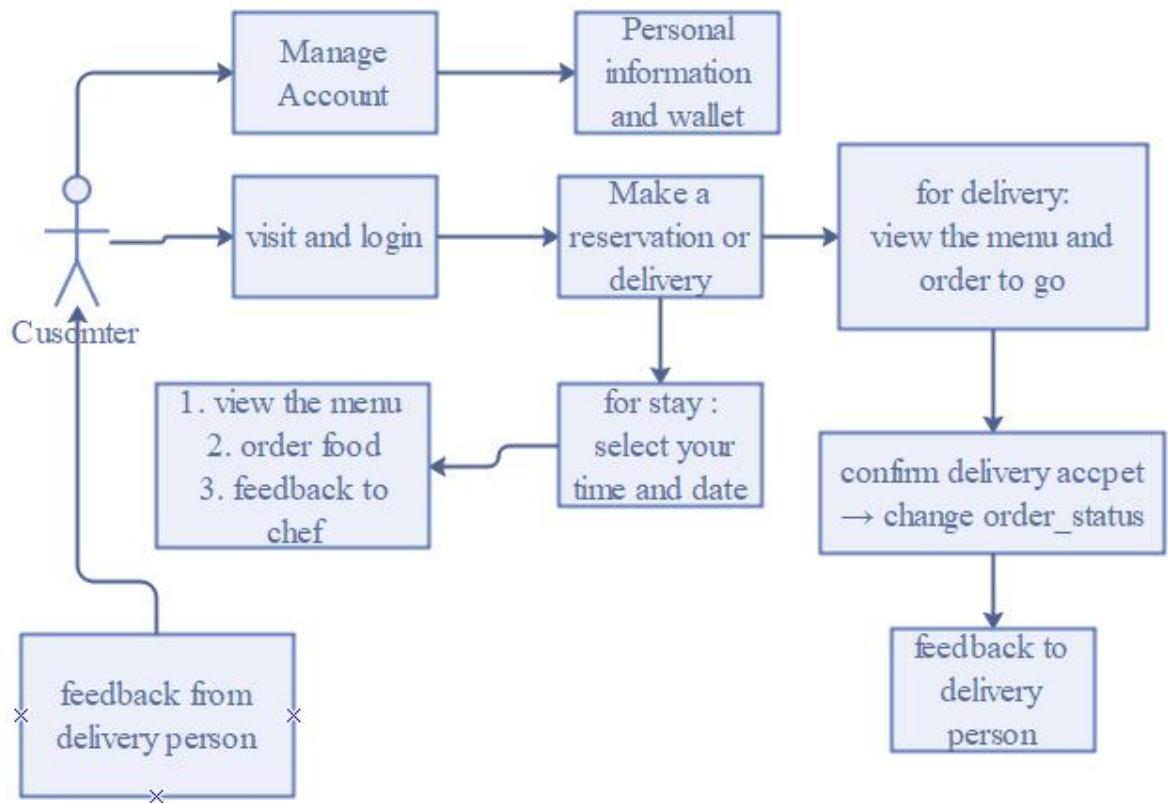
Delivery People User Case



1. File a complaint or compliment to the customer he/she delivered.
2. Delivery on time or no (feedback from customer) → change the status of order delivery

3. Received compliments to promote.
4. Received complaints to demote.

Customer User Case



Description: Customers can access their personal account.

1. The customer has the ability to access their account which they can update their personal information or deposit money into their account.
2. The customer can login to their account to start ordering the food from the chef.
 - a. They have the option to dine in or make a reservation for delivery/ take out.
3. Once the customers receive their dishes, they can give a rating to the chef and delivery person(if they choose delivery) about the quality of the services that they received.
4. The delivery person can also choose to file a compliment/complaint about the customers.
5. In times of customers using taboo words in a message, The warning is given to the customers followed by replacing the word with “*”.
6. After 3 warnings, the customer is de-registered. VIP customers will be downgraded to regular registered customers after 2 warnings given that both the warnings are cleared.

Structure of the project

Before Login

- 1.Login
- 2.Register
- 3.Administrator
- 4.Login
- 5.Contact Us

After Administrator Login :

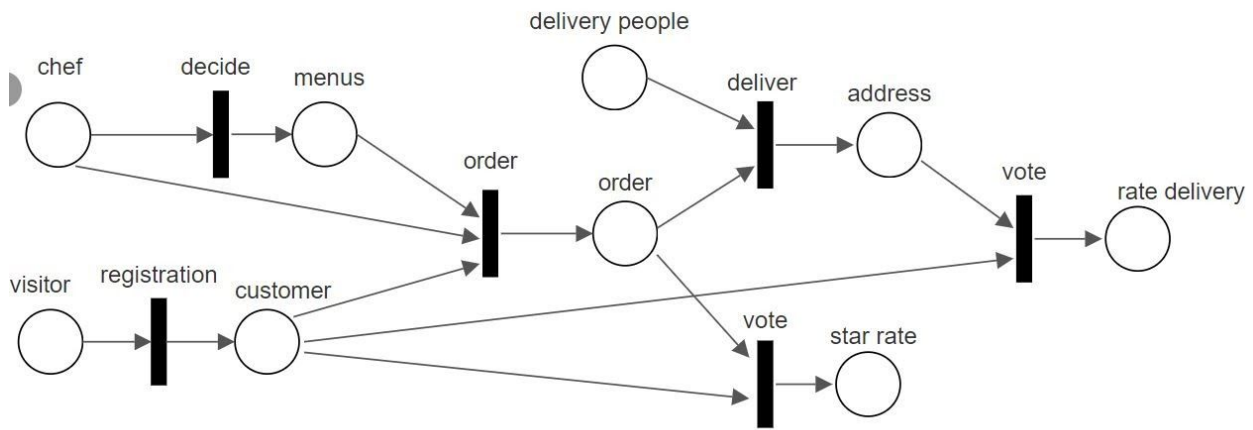
- 1.Edit Website Details
- 2.Add Food Items
- 3.Remove food Items
- 4.Manage Review
- 5.Manage Employee
- 6.Manage Customer
- 7.Logout

After User Login

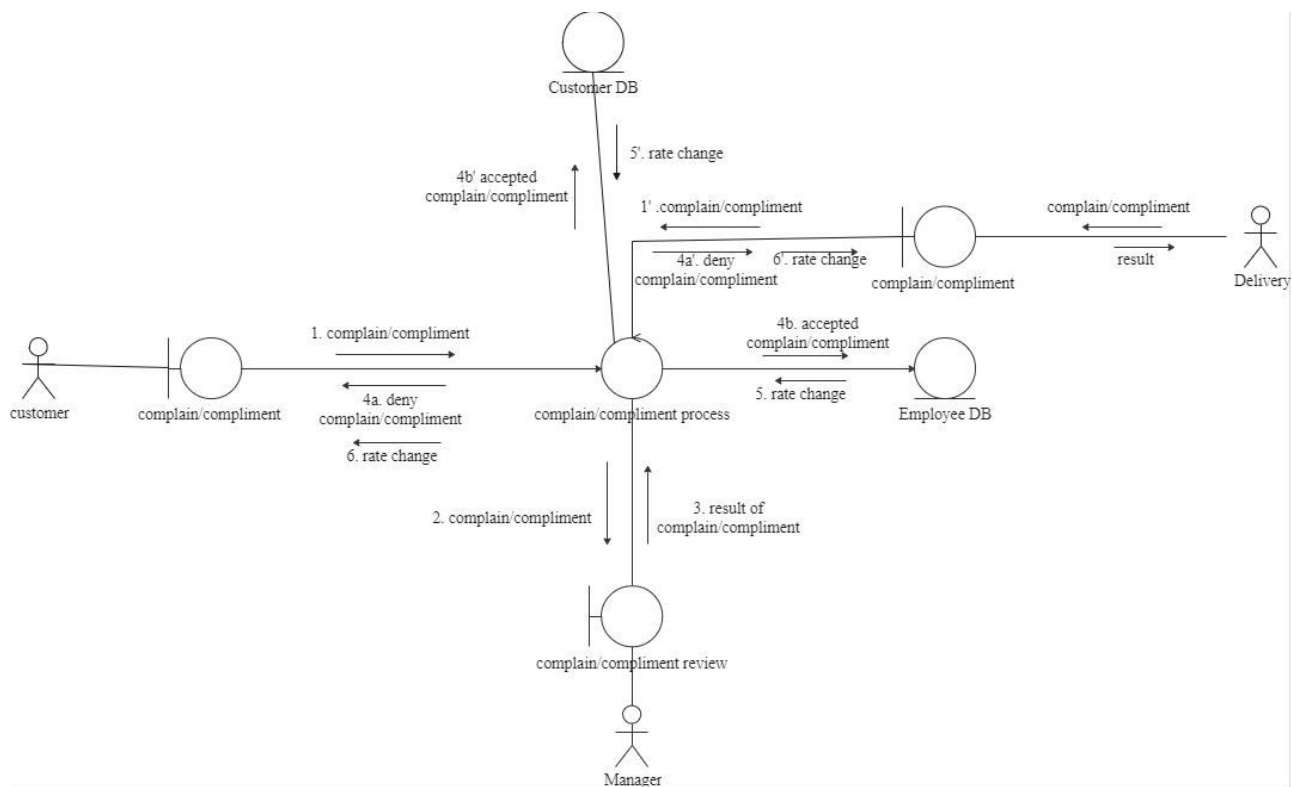
- 1.My Profile
- 2.Menu
- 3.Special Menus (VIP only)
- 4.My Cart
- 5.Check out
- 6.My review
- 7.Logout

Petri-net Diagram for few features

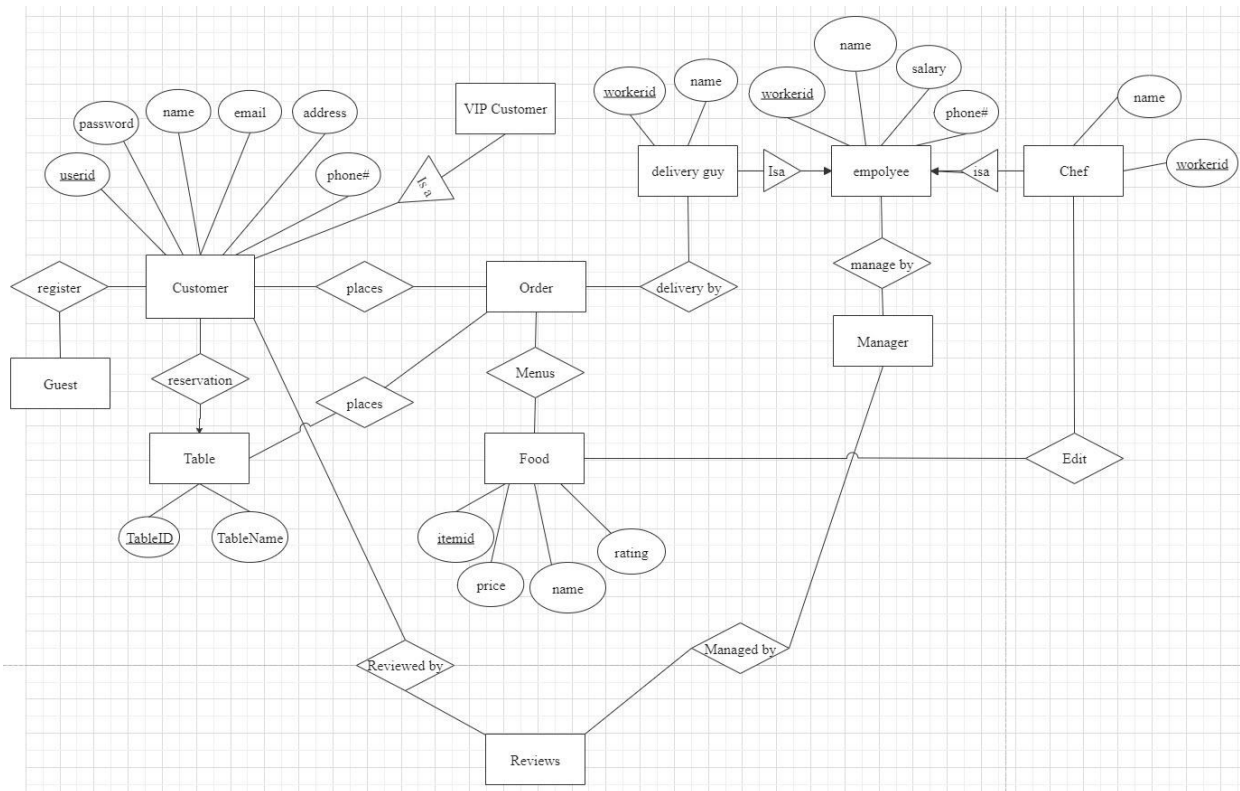
1. Visitors need to register to order food.
2. Chef can decide daily menus.
3. Customers must have enough deposited money to make the order.
4. Every order is delivered by delivery people to the customer personal address.
5. Customers can rate every order they placed.
6. Customers can rate delivery people who delivered food for them.



Manage review collaboration class diagram



3. E-R Diagram



4. Detailed Design - noted that this pseudo code is written in mostly python but we might use other programming languages. And the name of methods/structure/contents might be changed in order to fit our program. But the general purposes will be the same.

For example, we might remove a customer if warning ≥ 3 in python, however, in php, the implementation might be different, but the idea will be the same, to remove this customer.

```
def checkUser(level)
    0 = customer
    1 = vip_member
    blacklist = 9
```

```
def RegisterUser(userName, fname, lname):
    New_user = userName
    add user to registered users
    Assign user a unique id
    New_user_id = #some number
```

```
New_user.f
name = fname
New_user.lname = lname
```

```
def RateUser(user_rate):
    warning = 0
    if (avg_user_rate > 2)
        warning +=1
        print("warning: User has received an average rating of 2 or less")
    if(warning = 3):
        print("warning: User has received 3 warnings, remove user!")
        delete user
```

```
def AddMenuItem(item):
    add item to menu
```

```
def UpgradeUser(user):
    if(user.order_placed_count >= 50 && cus_spend >= 500):
        add user to VIP users
    if(avg_user_rating <= 2 && complains_count <= 3):
        login_level - 1
        if (login_level < 0)
            delete user
    // login_level 0 represent normal customer
    // login_level 1 represent VIP customer
```

```
def ComplaintGiven( complains_count, userRating):
    warning += 1
    //warning : user has received 3complaints and an average rating of 2 or less
```

```
def Addmenuitem(item):
    add a item to menu
```

```
def DropMenuItem(avg_rating):
    if(avgRating < 2)
        delete item
```

```
def pay_roll(login_level; delivery; user_id):
```

```
initialize the user_id
if ( delivery = completed):
    employee_wallet += employee_payment
```

```
def inc_Salary(empolyee_id):
    initialize New_rate
    if(login = admin)
        rate = new_Rate
```

```
def display_Menu(login_level)
    if (login_level != 9)
        display menu
```

```
def manager_view(login = admin):
    if (login = admin)
        edit_menu():
        checkordering():
        display_menu():
        erase.warning():
        manage customer():
        manage emp():
```

```
def food_rating():
    if (food_rate <2)
        food_mark += 1  warning
    if (food_mark >2)
        delete food
    else
        food_mark= 0
```

```
def demo_emp(empolyee_id):
    if (avg_rating_food < 3 && complaints >= 3)
        complaints += 1
    warning = warning + compliments - complaints
    if (warning >= 2)
        delete employee
```

```
def checkout(login_leve, total, user_id)
    balance = user.database.balance
    if (balance >= total)
        if (login_level = 0):
```

```
        balance = balance - total
    else
        balance = balance - (total *0.9)
else
    not enough money
```

```
accountExist($conn,$username, $pwd) {
```

```
    if($username != "" && $pwd != "")
    {
        $sql = "SELECT * FROM info WHERE username = '$username' and pwd = '$pwd' ";
        $result = mysqli_query($conn,$sql);
        $check = mysqli_fetch_array($result);

        if(isset($check))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```

```
}
```

```
deleteS($conn,$username)
```

```
{
    $sql = "DELETE FROM info WHERE username = '$username'";
    mysqli_query($conn,$sql);
}
```

```
vipgone($conn, $username)
```

```
{
    $sql = "UPDATE info SET level =0, warning = 0 WHERE username = '$username' and warning >=2";
    mysqli_query($conn,$sql);
}
```


5. System Screens

All the web pages right now are basic sketches, it will be improved later with advanced CSS/HTML. But the functionality is the same.

The Online Restaurant			
Home	New User	About Us	Contact

Username:	<input type="text"/>
Password:	<input type="password"/>
<input type="submit" value="Submit"/>	Admin?

This is the login page for the project. Both VIP users and regular users will enter their username and password here.

//Regular user

The Online Restaurant			
Order	Complain	Deposit	Log out

Welcome User: hi

Money left: 1021

Warnings: 0

//VIP user

The Online Restaurant

Order

Complain

Deposit

Log out

Welcome VIP User: VIPUSER

Money left: 300

Warnings: 3

//if we click deposit for both regular and VIP users

Money you want to deposit:

Deposit!

//for admin

The Online Restaurant			
Home	New User	About Us	Contact
Admin Username: <input type="text"/>			
Password: <input type="text"/>			
<input type="submit" value="Submit"/>			

The Online Restaurant

Welcome User: Admin

Demote

Fire

Compliment/complain

de-register

Compliment/complain

log out

//if we click de-register , this will show all the VIP users or regular users who has a warning of ≥ 2 and ≥ 3 respectively.

The Online Restaurant

Welcome User: Admin

List:warning ≥ 3

username	FullName	Warning
example	example	3

De-register Username:

VIP warning ≥ 2

username	FullName	Warning
VIPUSER	VIP	3

Demote to regular:

6. Group Meetings

Date: November 07, 2020

Participants: All group members

Agenda:

1. We decided to use PHP, HTML and CSS as our main structure for this project.
2. Come up with a main web page sketch.

Problem:

1. The name of our restaurant remained undecided.
2. The type of our restaurant remained undecided.
3. Undecided pictures for our cover page and menus.

Date: November 13, 2020

Participants: All group members

Agenda:

1. Connected to the database system.
2. Draw the collaboration class diagram.

3. Draw the use case diagram.

Date: November 14, 2020

Participants: All group members

Agenda:

1. Draw the Petri-net diagram.
2. Work on HTML for the web page sketch.
3. Draw the E/R diagram.
4. Work on pseudo code.

Date: November 16, 2020

Participants: All group members

Agenda:

1. Put all of the diagrams, codes and figures into the report.

7. Github

All of our works are saved in our [Github](https://github.com/qxou14/322-Software-Engineering) webpage.

<https://github.com/qxou14/322-Software-Engineering>