Zabegalovka

Restaurant
Software Requirements Specification
For Menu & Ordering System

Version <1.1>

grade: 88
comments: the collaboration diagrams are great, only the overall one providing the entire system is not given. The petri-nets are all wrong and not in par with the col. diags--whoever developed them need to work hard to pick up how to draw an acceptable petri-nets.

Revision History

Date	Version	Description	Author
24/04/22	<1.0>	First draft of the report	Mohamed Kharma Radmir Sataev Jacob Onbreyt Yevheniya Solomyana Matheus Figueroa
26/04/22	<1.1>	Second draft of the report	Mohamed Kharma Radmir Sataev Jacob Onbreyt Yevheniya Solomyana Matheus Figueroa

Table of Contents

Phase II: Design Report

1 Introduction

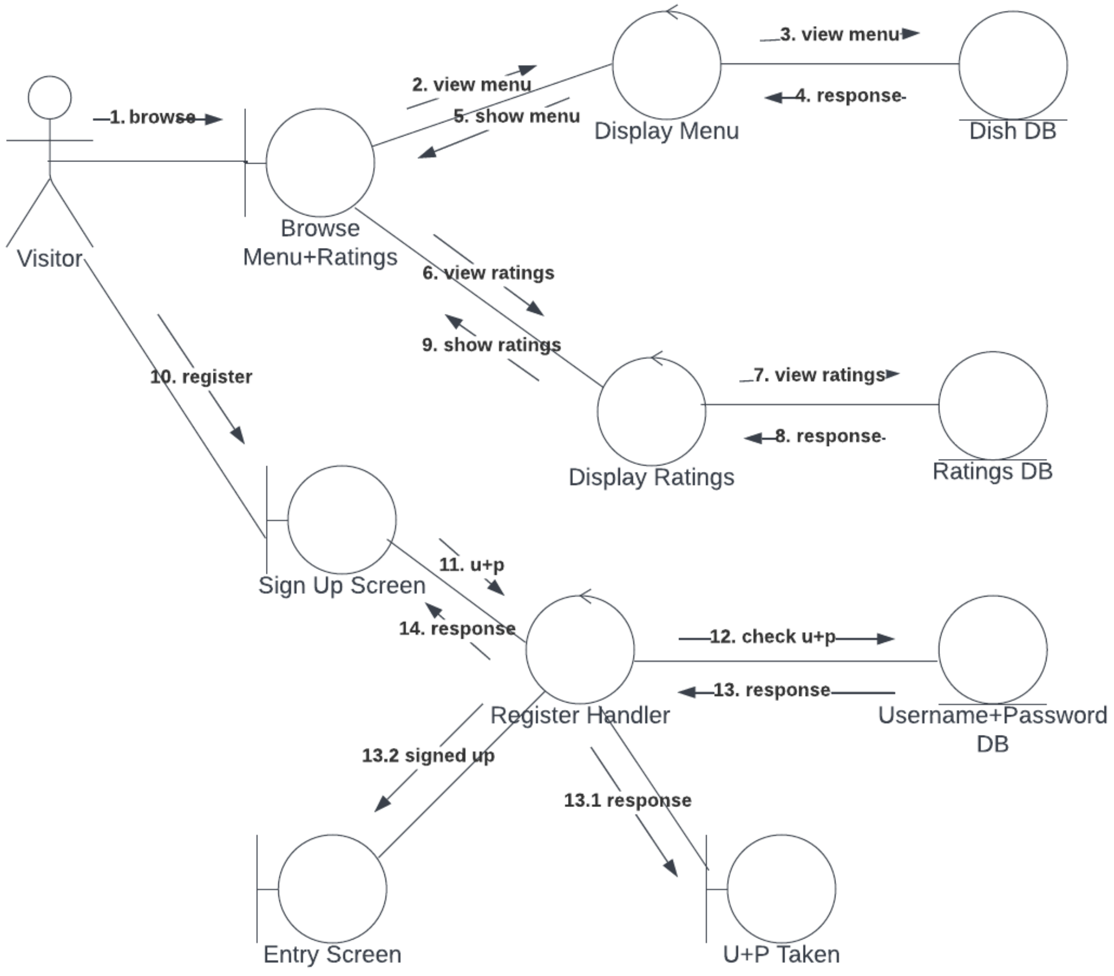
This report is meant to provide the data structure and logic to carry out the functionalities dictated by the specification. Keep in mind the implementation will be entirely based on this document, adequate details should thus be given at length.

1.1 Collaboration Class Diagram

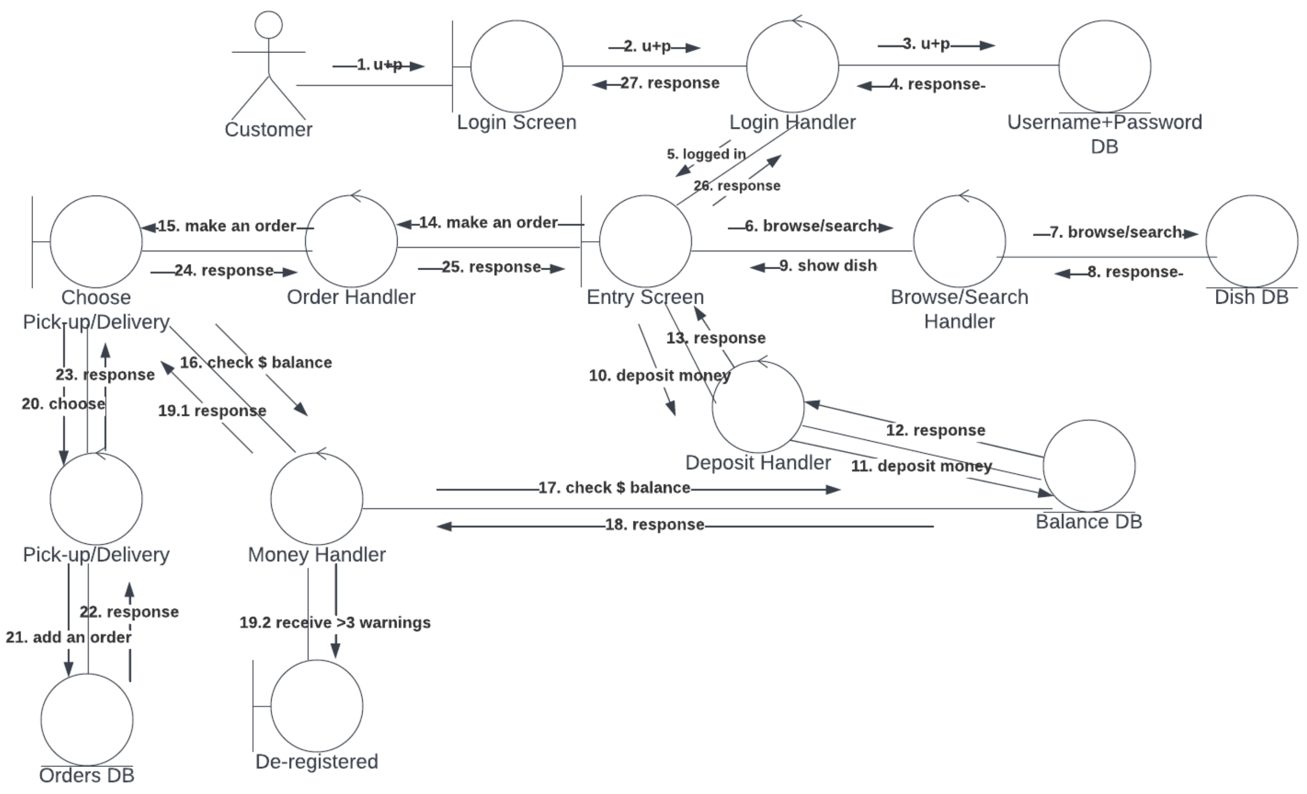
provide an abstracted high-level col. diag

An overall picture of the system is described in the collaboration class diagram. It is divided into parts based on users who interact with the online restaurant delivery system. The diagram shows how objects use the system and their according behaviors. It also provides an overall overview of the Zabegalovka system.

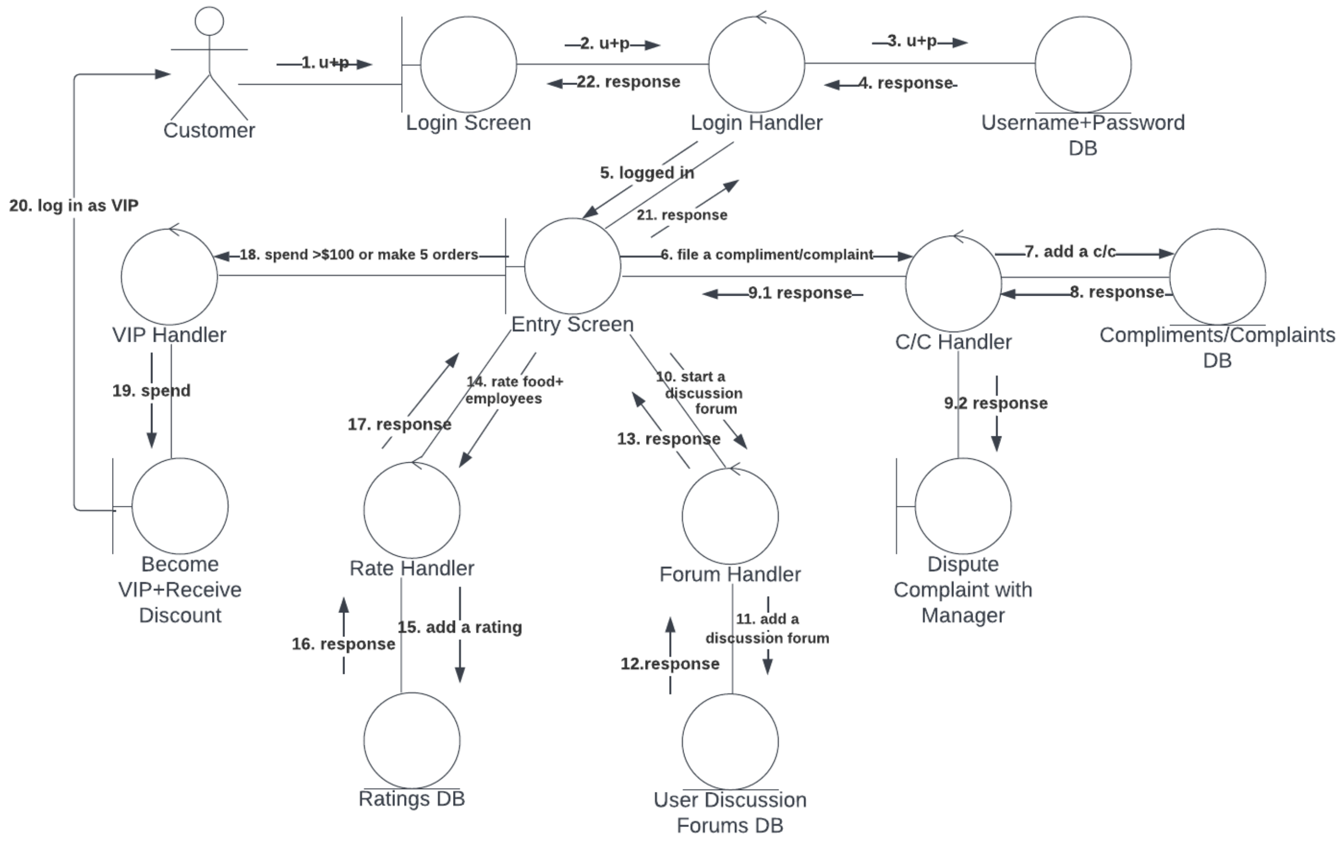
1.1.1 Visitor:



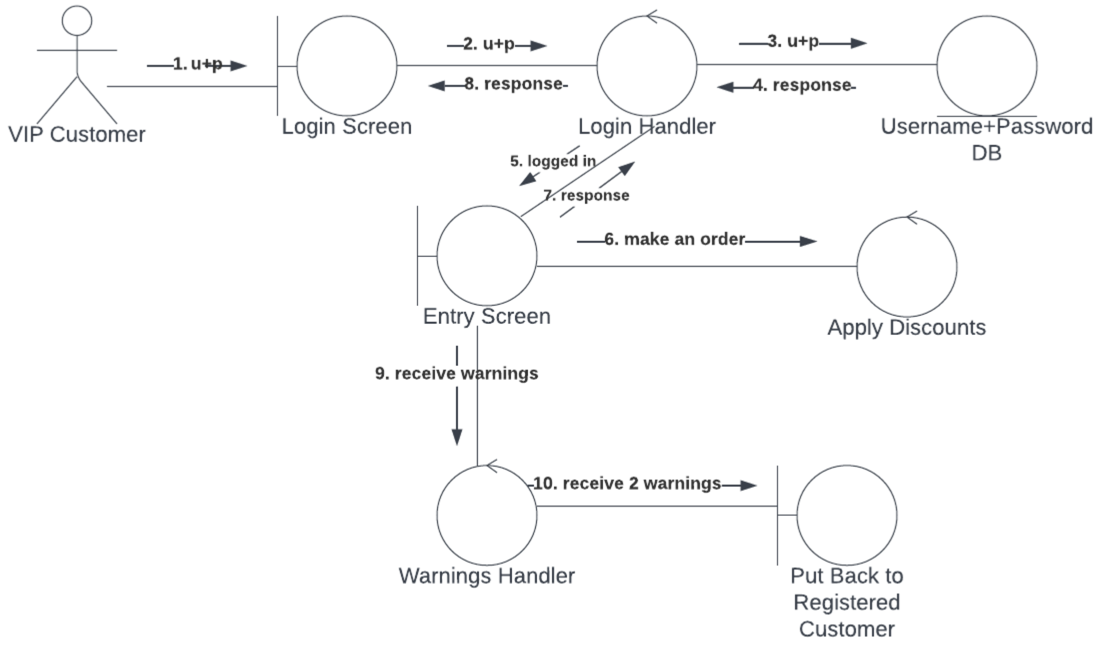
1.1.2 Registered Customer:



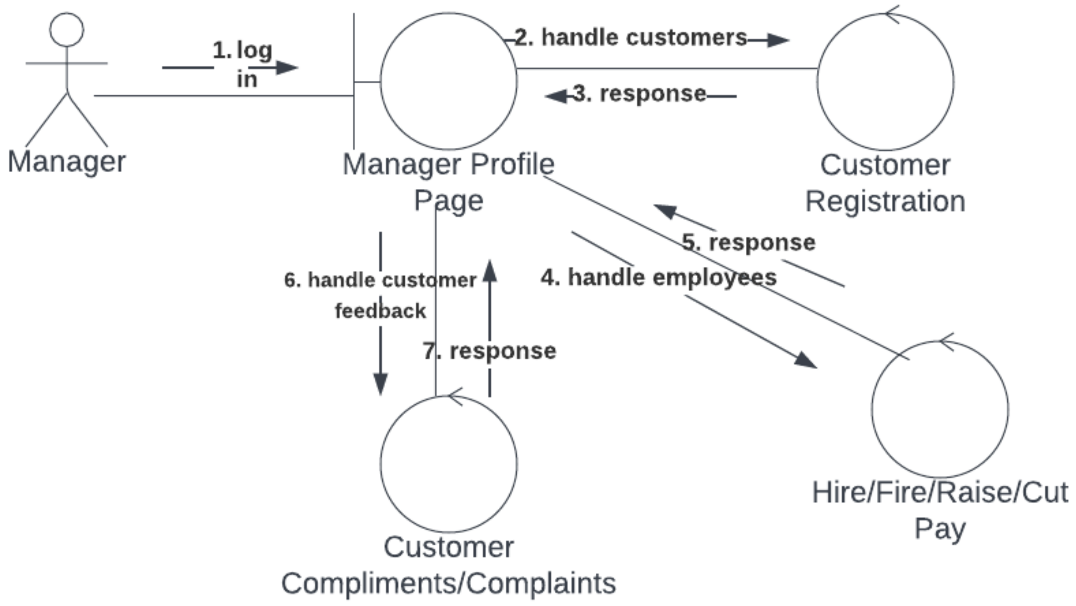
Registered Customer (cont.): divided into two parts for better viewing and understanding



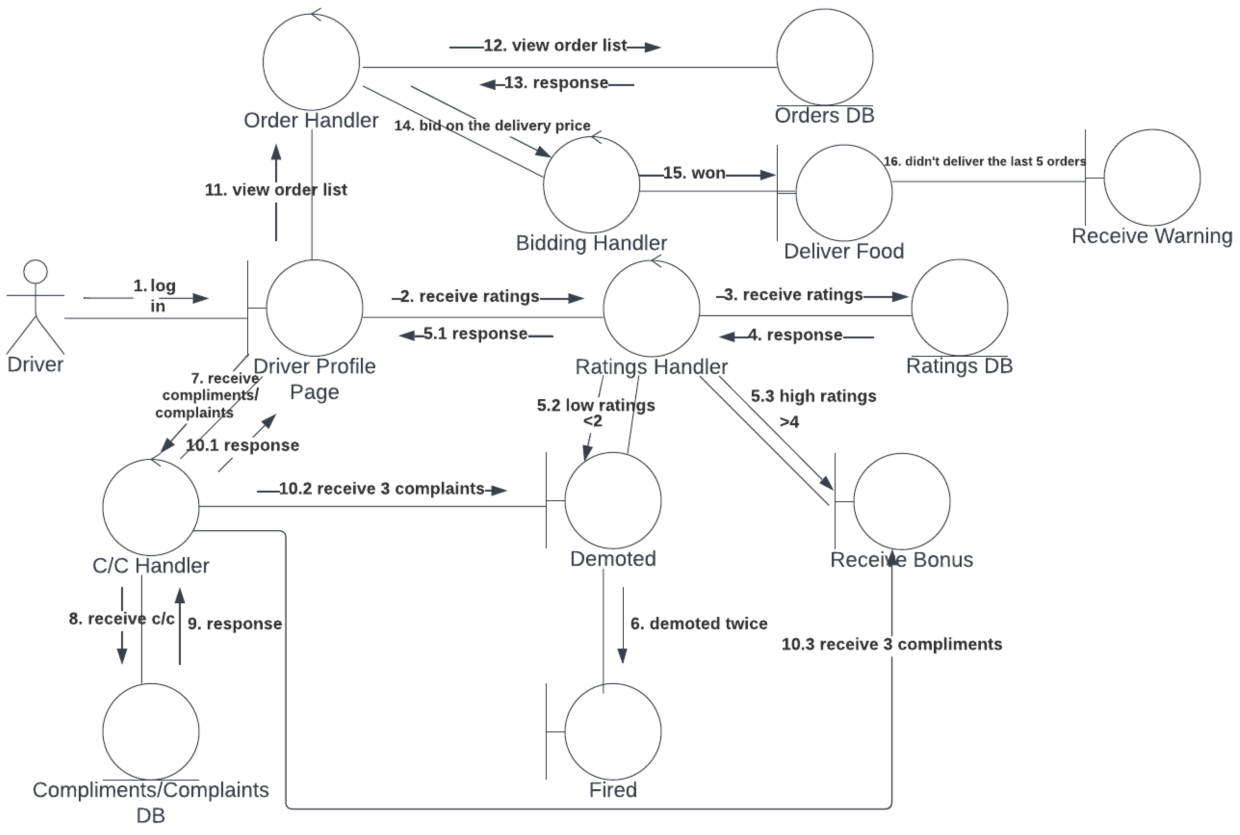
<u>VIP Customer:</u> shown additional VIP privileges; overall has same behaviors as Registered Customers



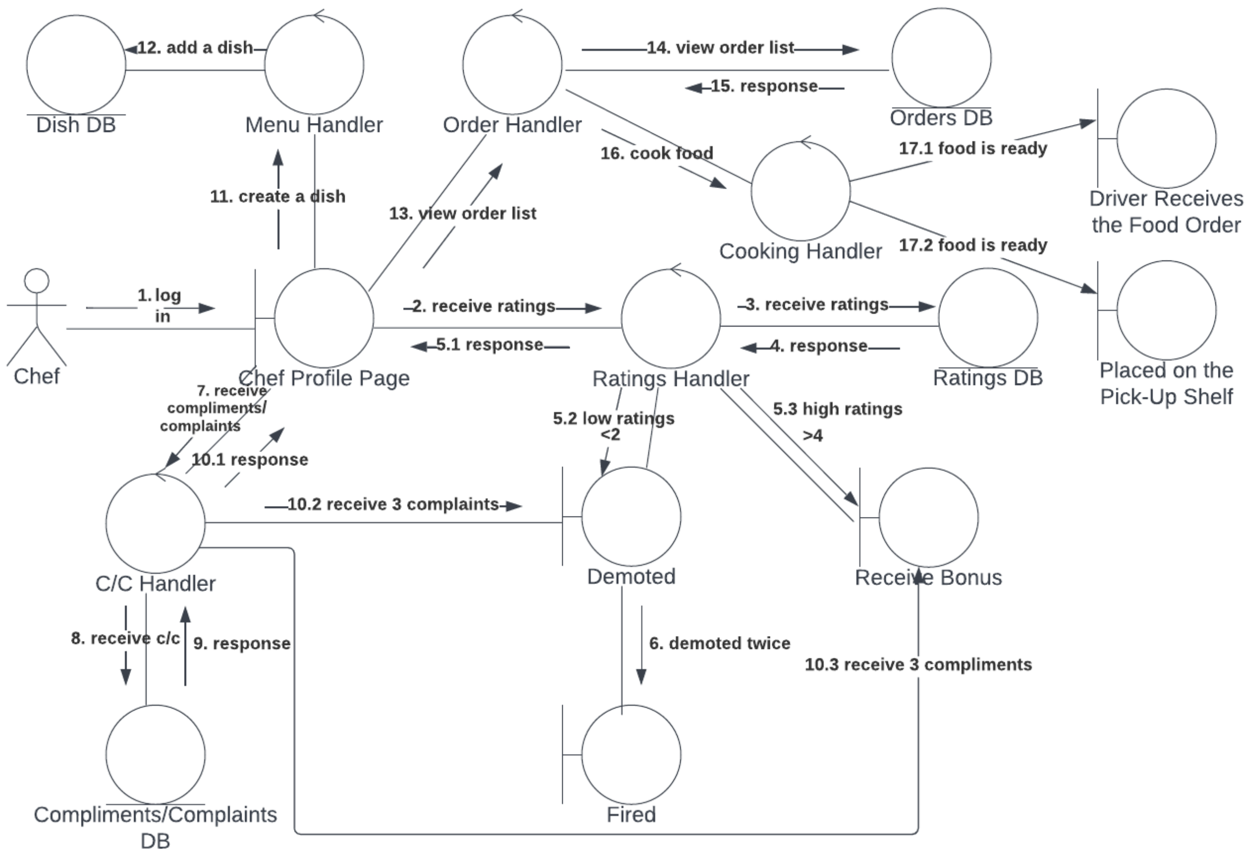
1.1.3 Manager:



1.1.4 Driver:



1.1.5 Chef:



2 Use Cases

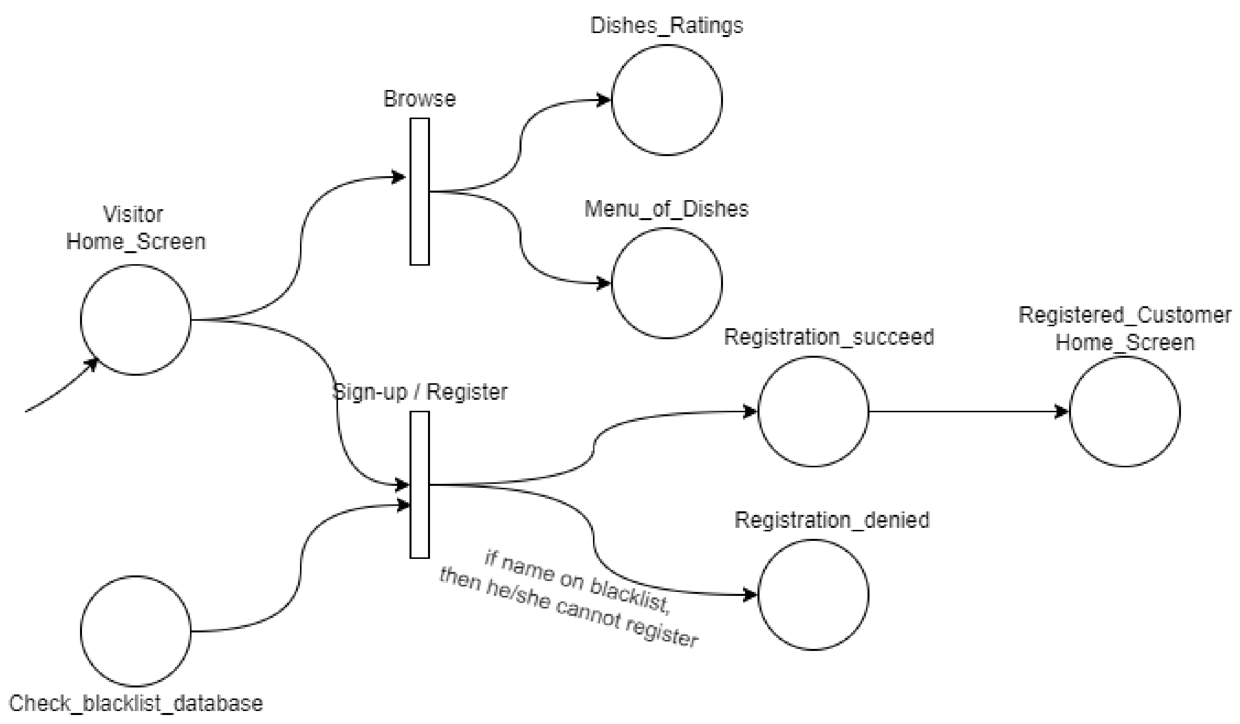
Scenarios for each use case: normal AND exceptional scenarios. Collaboration or sequence class diagram for each use case, choose 3 or more use cases: draw the Petri-nets instead of class diagrams

2.1 Customer as Visitor Use Cases

Use-Case: Browse the menu / sign up for an account

- Normal Use Case Scenario: Visitor is only able to browse restaurant dishes menu and ratings. Also, can apply and be a registered customer.

- Exceptional Scenario: Check the blacklist database of the restaurant, if the customer was previously a registered customer who got kicked-out, his/her registration will be denied (cannot register).

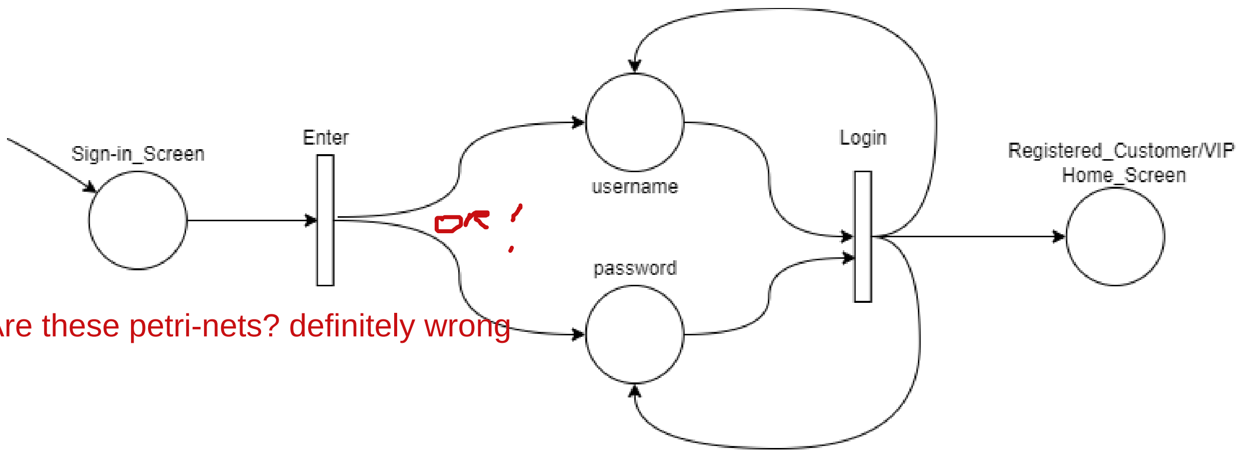


2.2 <u>Customer as Registered Use Cases</u>

Use-Case: Login

- Normal Use Case Scenario: customer provide the correct username and password for authorization

- Exceptional Scenario: Customer provided wrong username/password. Optional: Give him/her the choice to rest password if can't remember.

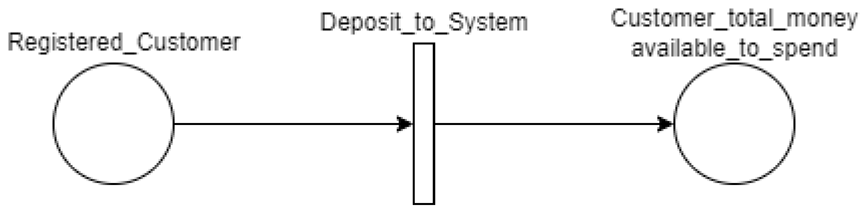


Are these petri-nets? definitely wrong

Use-Case: Deposit Money

- Normal Use Case Scenario: Every registered customer must deposit money into his/her account to use it for ordering.

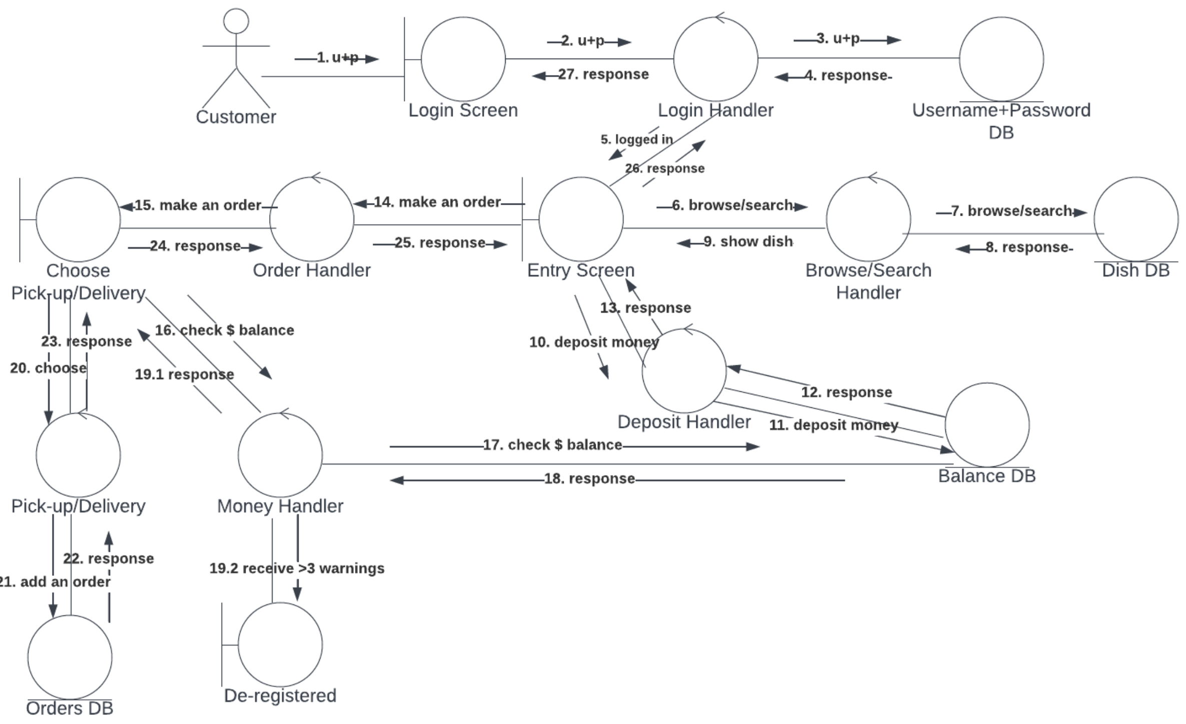
- Exceptional Scenario: None.



Use-Case: Order Dishes

- Normal Use Case Scenario: once the customer logs in, he/she can browse all available dishes and add any of the dishes to his/her order cart. After the registered customer chooses all the desired dishes and his deposited balance money is enough for the order, then he gets to choose whether he/she wants to pick up their order or have it delivered. Lastly, when the order gets submitted, it gets added on the Orders database.

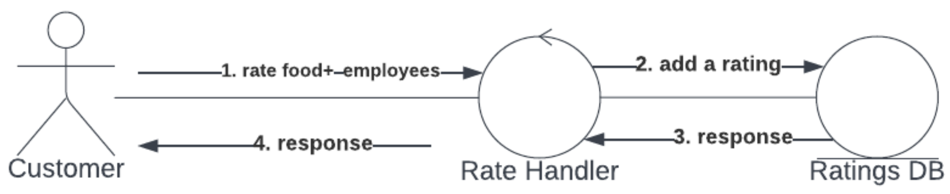
- Exceptional Scenario: Customer balance is not enough for the order. Thus, the customer is asked to add more balance before submitting the order and receive a warning for being reckless. Having 3 warnings as a registered customer will result in getting de-registered.



Use-Case: Rate dishes/delivery service

- Normal Use Case Scenario: Once the order is completed, customers can review their order through the history page and rate the dish and delivery service between 1 to 5 stars, where 1 being the lowest rating and 5 being the highest.

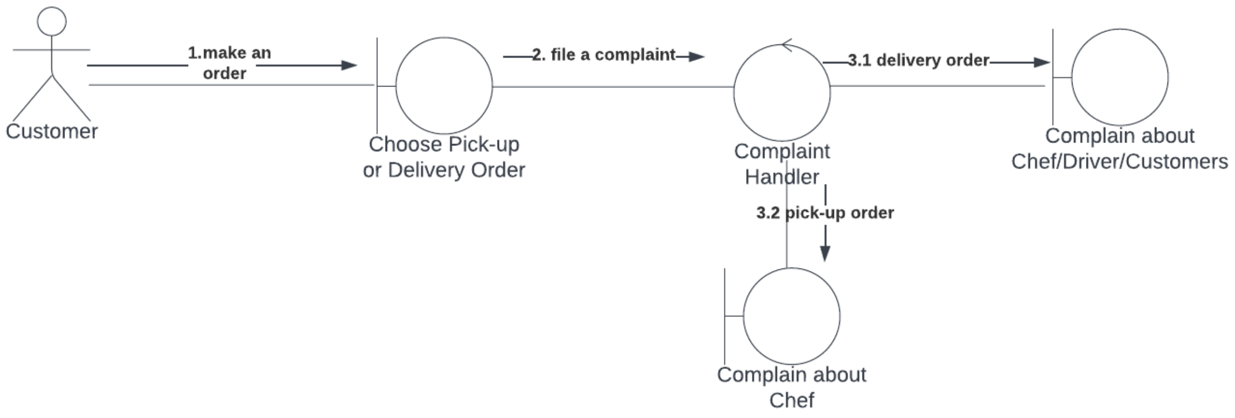
- Exceptional Scenario: None.



Use-Case: Complaints

- Normal Use Case Scenario: After receiving the order, the customer can file a complaint about the Chef cook or the Delivery person service or other customers who didn't behave in the discussion forums.

- Exceptional Scenario: If the customer chooses to pick up the order, he/she can only complain about the Chef

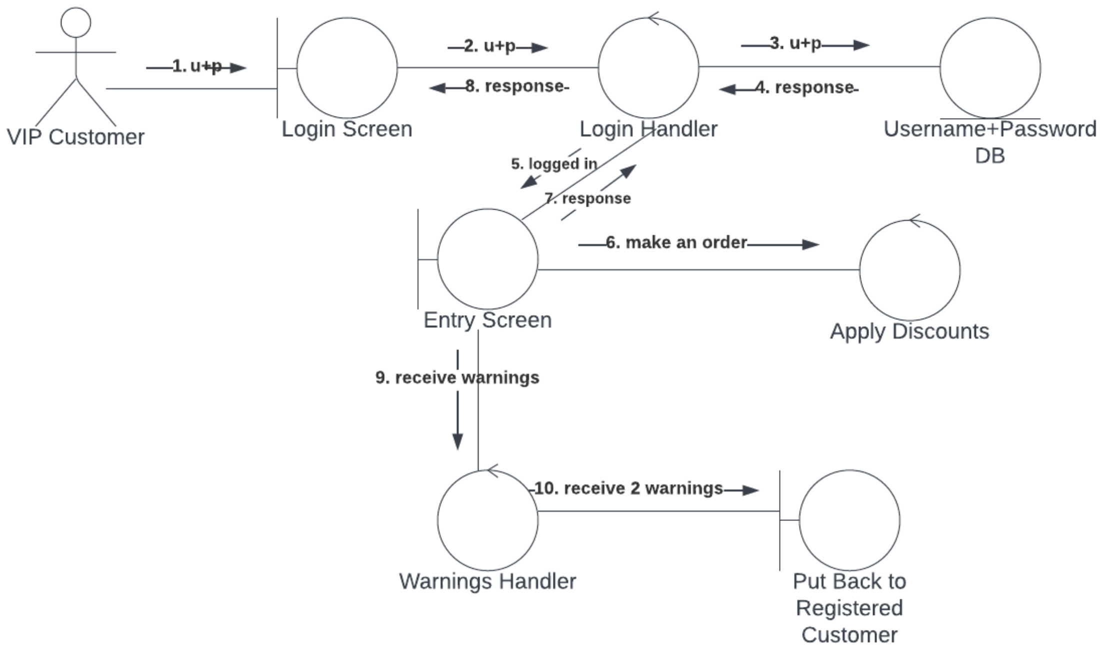


2.3 <u>Customer as VIP Use Cases</u>

Use-Case: VIP customer

- Normal Use Case Scenario: Registered customers could become VIP customers if they spent more than $100 or had 5 ordered without any outstanding complaints.

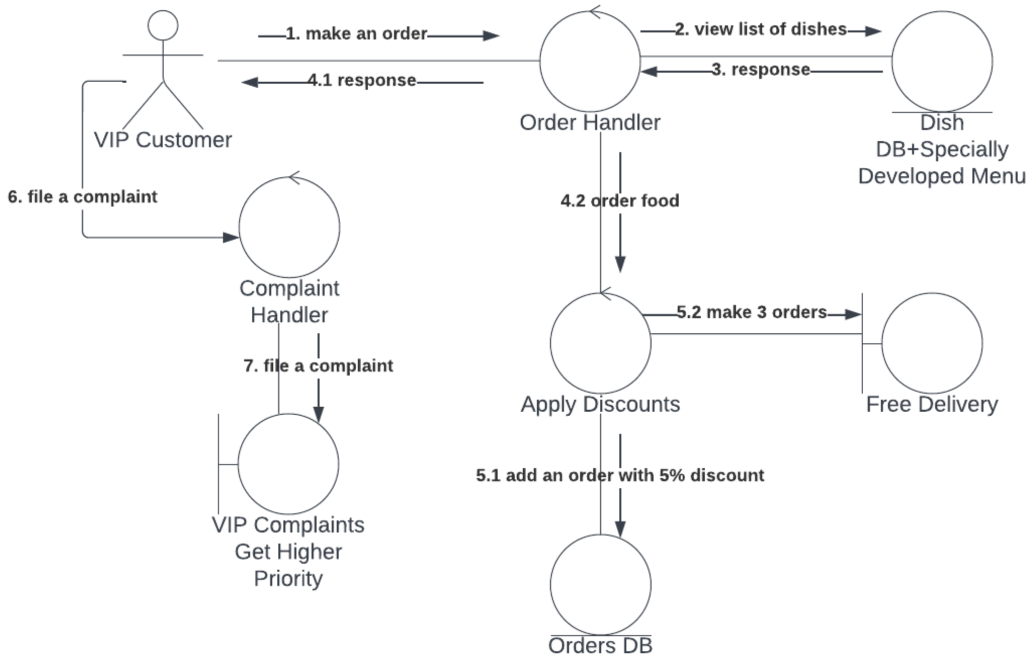
- Exceptional Scenario: VIPs having 2 warnings are put back to registered customers (with warnings cleared)



Use-Case: perks for VIP customers

- Normal Use Case Scenario: VIP customers will receive 5 % discount on their ordinary orders and 1 free delivery for every 3 orders, have access to specially developed dishes, and their complaints/compliments are counted twice as important as ordinary ones

- Exceptional Scenario: None

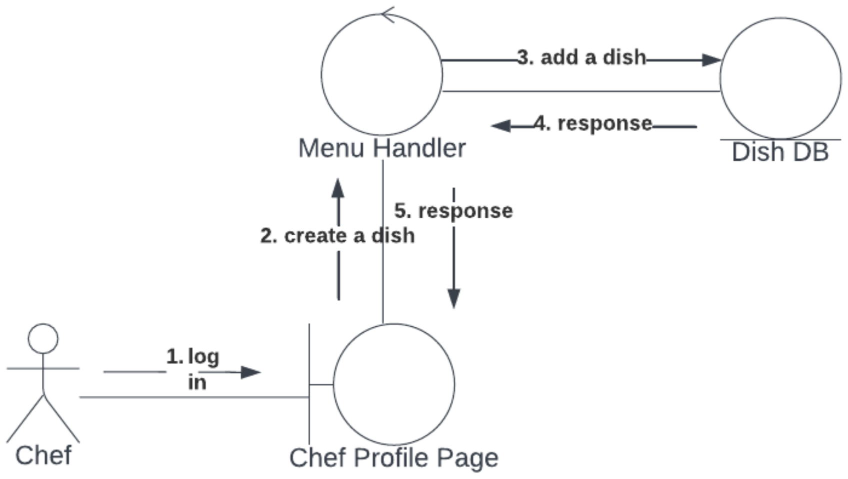


2.4 <u>Chef Use Cases</u>

Use-Case: Add item to menu

- Normal Use Case Scenario: Chef can add a dish to the menu including the name, description, price, and picture of the dish.

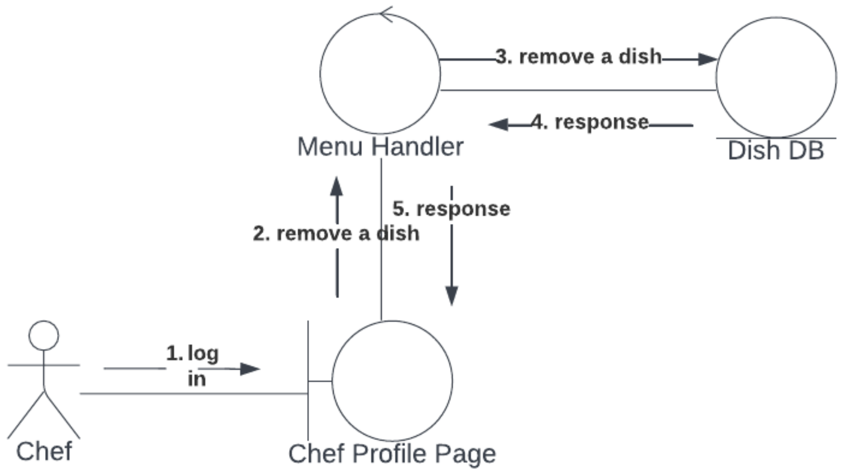
- Exceptional Scenario: None



Use-Case: Remove item from menu

- Normal Use Case Scenario: Chef can remove dish from the menu.

- Exceptional Scenario: None

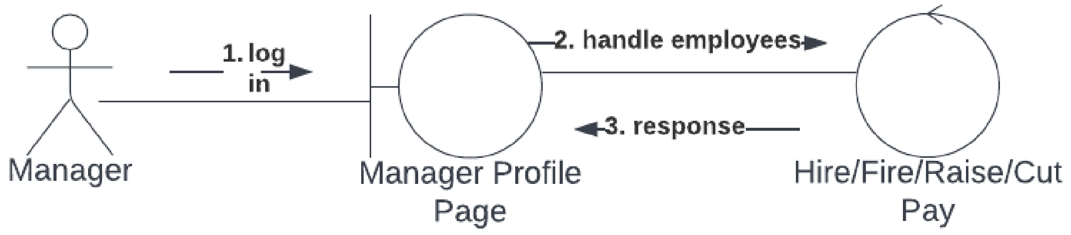


2.5 <u>Manager Use Cases</u>

Use-Case: Managing employees

- Normal Use Case Scenario: A manager hires/raises/fires/cuts pay for chefs and delivery people.

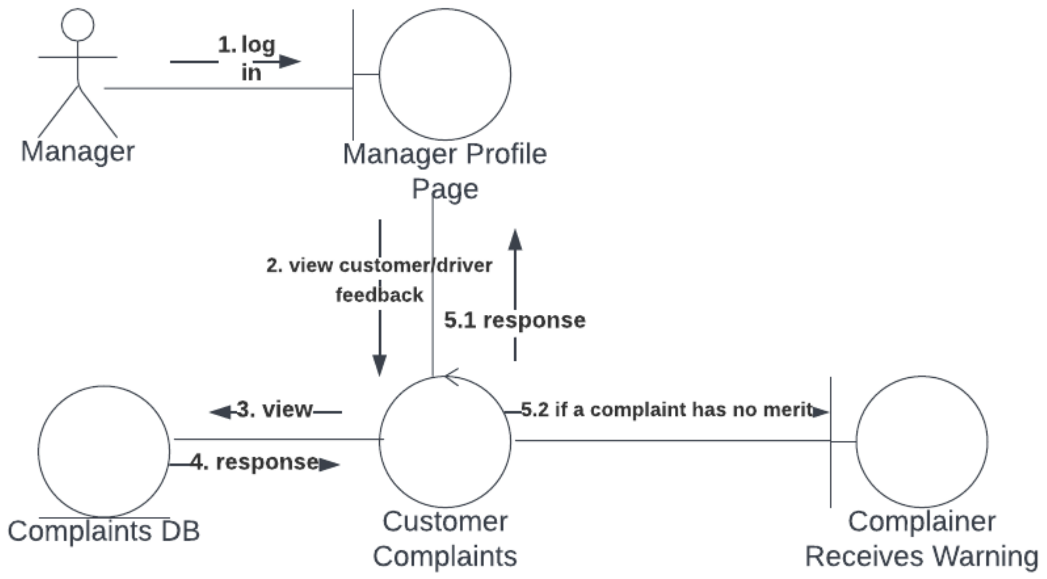
- Exceptional Scenario: None



Use-Case: Deal with complaints

- Normal Use Case Scenario: All filed complaints received from Customers or Delivery people will be reported to the manager, in which the manager decided if the complaint is valid or not.

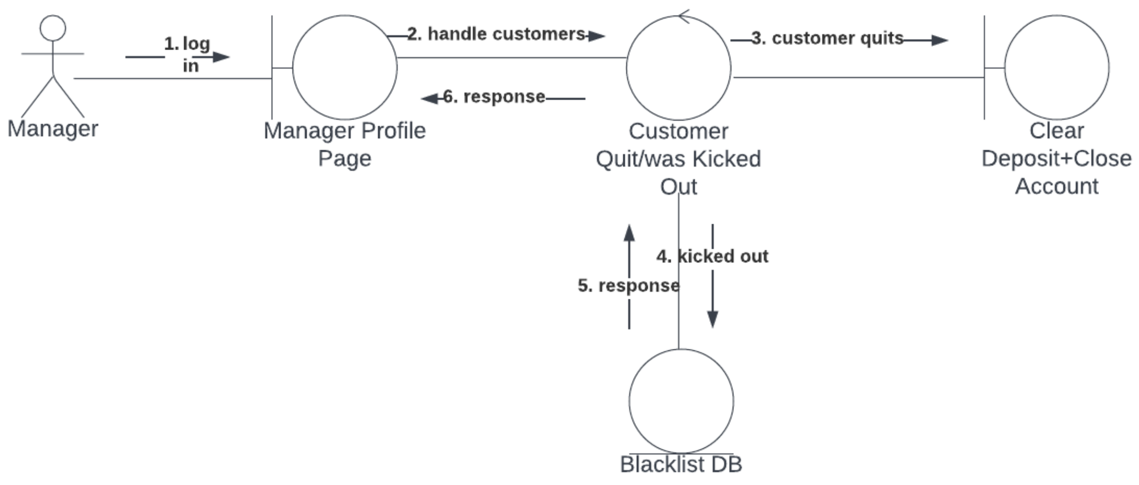
- Exceptional Scenario: If any of the complaints is found to be without merit, the person who filed it will receive the warning



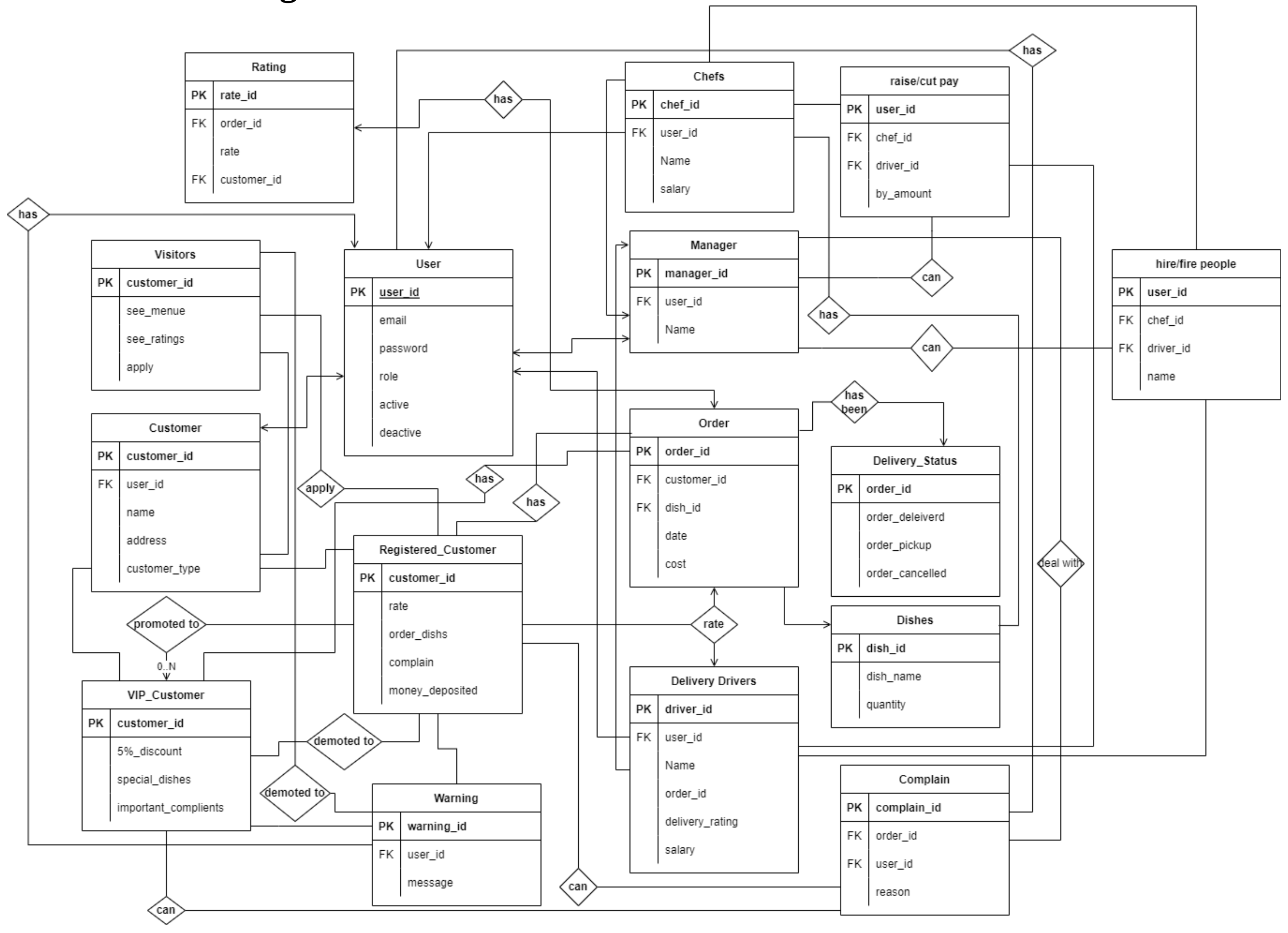
Use-Case: Managing Users accounts

- Normal Use Case Scenario: Customers who get kicked out (due to exceeding the limited warning) or choose to quit will be handled by the manager, in which their deposit will be cleared, and their account closed. All kicked out customers will be added to the restaurant blacklist database (cannot register again)

- Exceptional Scenario: None


Manager
1. log in
Manager Profile Page
2. handle customers
6. response
Customer Quit/was Kicked Out
3. customer quits
Clear Deposit+Close Account
4. kicked out
5. response
Blacklist DB

3 E/R Diagram


Rating
PK rate_id
FK order_id
rate
FK customer_id
Chefs
PK chef_id
FK user_id
Name
salary
raise/cut pay
PK user_id
FK chef_id
FK driver_id
by_amount
Visitors
PK customer_id
see_menue
see_ratings
apply
User
PK user_id
email
password
role
active
deactive
Manager
PK manager_id
FK user_id
Name
hire/fire people
PK user_id
FK chef_id
FK driver_id
name
Customer
PK customer_id
FK user_id
name
address
customer_type
Order
PK order_id
FK customer_id
FK dish_id
date
cost
Delivery_Status
PK order_id
order_deleiverd
order_pickup
order_cancelled
Registered_Customer
PK customer_id
rate
order_dishs
complain
money_deposited
Dishes
PK dish_id
dish_name
quantity
VIP_Customer
PK customer_id
5%_discount
special_dishes
important_complients
Delivery Drivers
PK driver_id
FK user_id
Name
order_id
delivery_rating
salary
Warning
PK warning_id
FK user_id
message
Complain
PK complain_id
FK order_id
FK user_id
reason
has
has
has
has
has
has
has
has been
can
can
can
can
apply
promoted to
0..N
demoted to
demoted to
rate
deal with

4 Detailed Design

4.1 Customer Methods:

Login:

```
LoginUser( user , passwd) {
        exist = checkDB (user, passwd);
        if (exist) {
        if(exist.stat = VIP)
        {
        VIPSignIn();
        }
                signIn();
        }
}
```

Sign Up:

```
SignUpUser (username, passwd) {
        User newuser = new User ();
        newuser.setUsername(username);
        newuser.setPassword(passwd);

        if (newuser.success) {
                signIn();
        }
        else {
                invokeError();
        }
}
```

Menu Viewing:

```
getmenu() {
        return List<dish>
        //return all menu items
}

getMenuByName(dishName) {

        return List<dish> of dishName;

}

getMenuByRating (rating) {

        return List<dish> of rating;

}
```

Order Item / Delete Item:

```
//create or update order
orderFood(dishname, quantity) {
        exist = findInDB(dishname);
        if (exist) {
                addToCart(dishname,quantity);
        }
}

deleteFood(dishname,quantity) {
        if (quantity == cart.dishname.size) {
                deleteAll(dishname);
        }
        else {
                for (i = 0 to quantity) {
                        deleteOne(dishname);
                }
        }
}
```

Ratings / Complaints:

```
RateResturant(name, rate, comment) {
        if (rate != null) {
                restaurant.name.rate += rate;
        }
}


ResturantComplaint(name, complaint) {
        if (complaint != null) {
                restaurant.name.complaintArray.append(complaint);
        }
}
```

4.2 Chef Methods:

Add / Delete item from menu:

```
addToMenu(item) {
        exist = checkItemInDB(item);

        if (!exist) {

                restaurant.menu.addItem(item);
        }
```

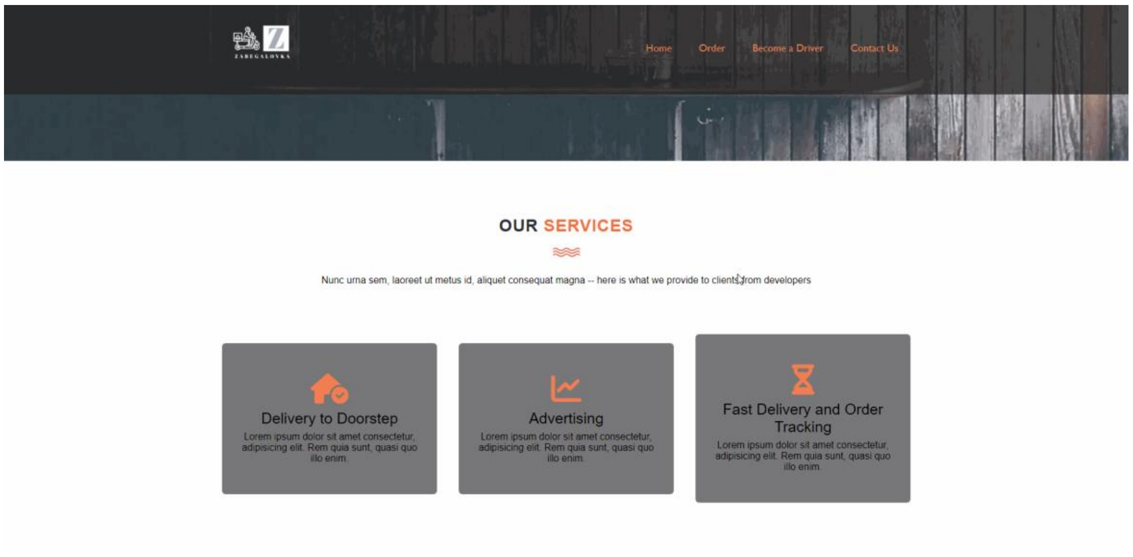
```
    }


    deleteFromMenu(item) {
            exist = checkItemInDB(item);
            if (exist) {
                    restaurant.menu.deleteItem(item);
            }
    }
```

4.3 Manager Methods:

```
  manageEmployee (employee, newRating, complaint) {
    employee.rating += newRating;
    employee.complaintArray.append(complaint);
            if ((employee.rating < RATING_LIMIT) ||
                            (employee.complaintArray.size > COMPLAINT_LIMIT) ) {
                    employeeConsequence();
                    }
                    else {
                            promoteEmployee();
                    }
    }
```

5 System Screens

5.1 Home Page



OUR SERVICES

Nunc urna sem, laoreet ut metus id, aliquet consequat magna -- here is what we provide to clients from developers

Delivery to Doorstep

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rem quia sunt, quasi quo illo enim.

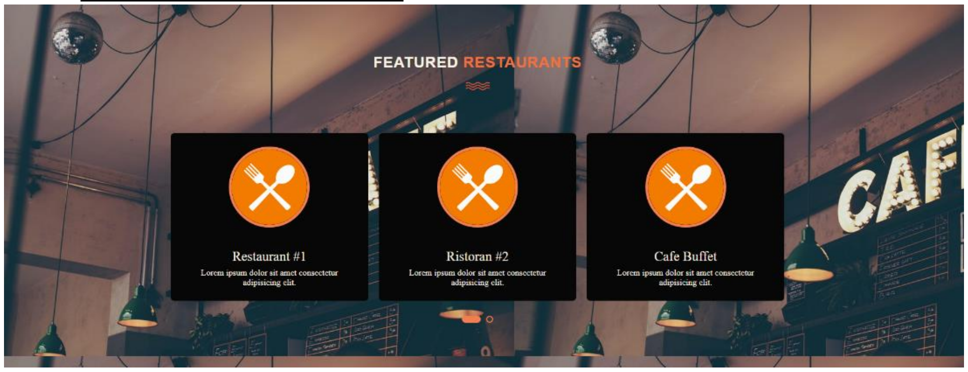
Advertising

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rem quia sunt, quasi quo illo enim.

Fast Delivery and Order Tracking

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Rem quia sunt, quasi quo illo enim.

5.2 Featured Restaurants

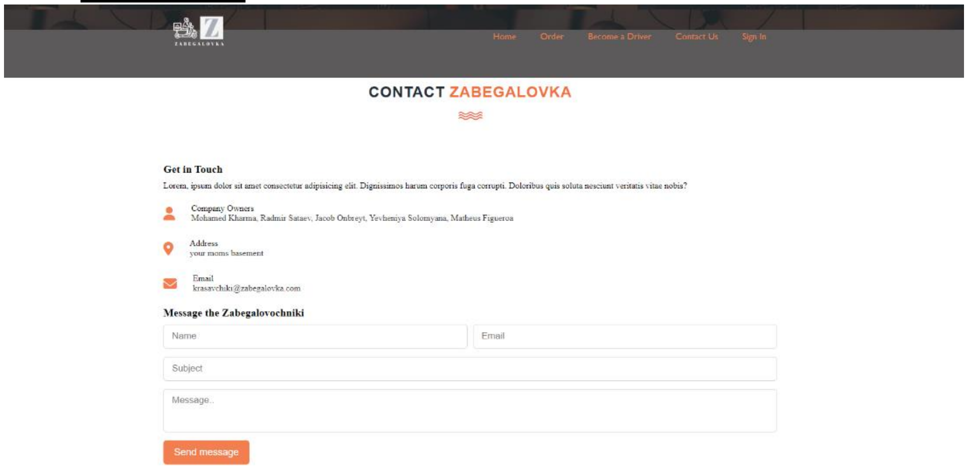

FEATURED RESTAURANTS

Restaurant #1
Lorem ipsum dolor sit amet consectetur adipisicing elit.

Ristoran #2
Lorem ipsum dolor sit amet consectetur adipisicing elit.

Cafe Buffet
Lorem ipsum dolor sit amet consectetur adipisicing elit.

5.3 Contact Us



CONTACT ZABEGALOVKA

Get in Touch

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dignissimos harum corporis fuga corrupti. Doloribus quis soluta nesciunt veritatis vitae nobis?

Company Owners
Mohamed Kharma, Radmir Sataev, Jacob Onbreyt, Yevheniya Solomyana, Matheus Figueroa

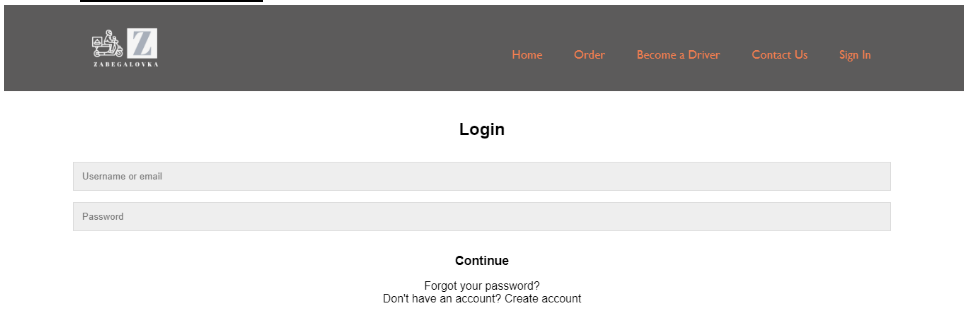
Address
your moms basement

Email
krasavchiki@zabegalovka.com

Message the Zabegalovochniki

Name
Email
Subject
Message..
Send message

5.4 Sign in Page



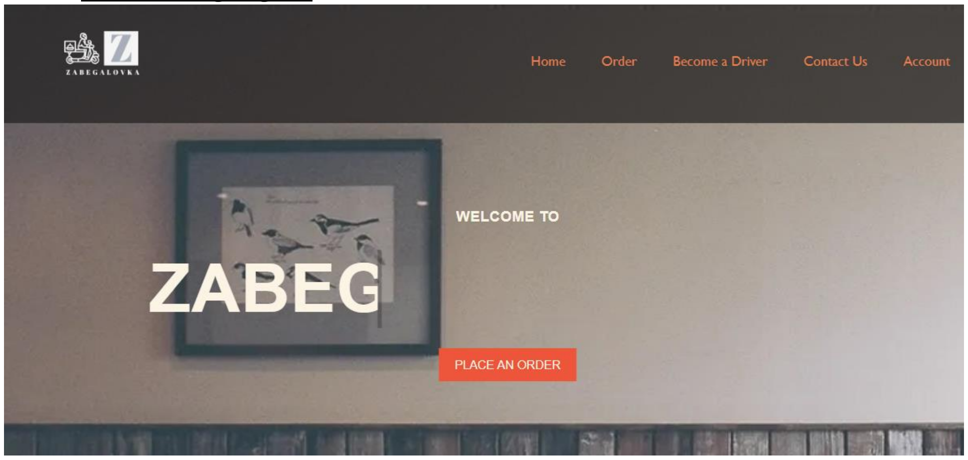
Login

Username or email
Password

Continue

Forgot your password?
Don't have an account? Create account

5.5 After singing in



6 Group Meetings

In total, we have had 5 meetings:

Meeting #	Time Spent	Date	Reason for meeting
1	1 Hour	March 22, 2022	Got to know each other's
2	2 Hours	March 25, 2022	Discussed how we are going to approach the project
3	3 Hours	March 27, 2022	Worked on the first report
4	1 Hours	April 24, 2022	Discussed and decided to make a web application
5	4 Hours	April 25, 2022	Worked on the application and second report

7 Additional Information

Address of the git repo (github, gitlab, bitbucket, etc.) of your team's work so far - put all materials including this report there

GitHub repo