FOODINIT

Online Restaurant

Software Requirements Specification

Version 2.0

Revision History

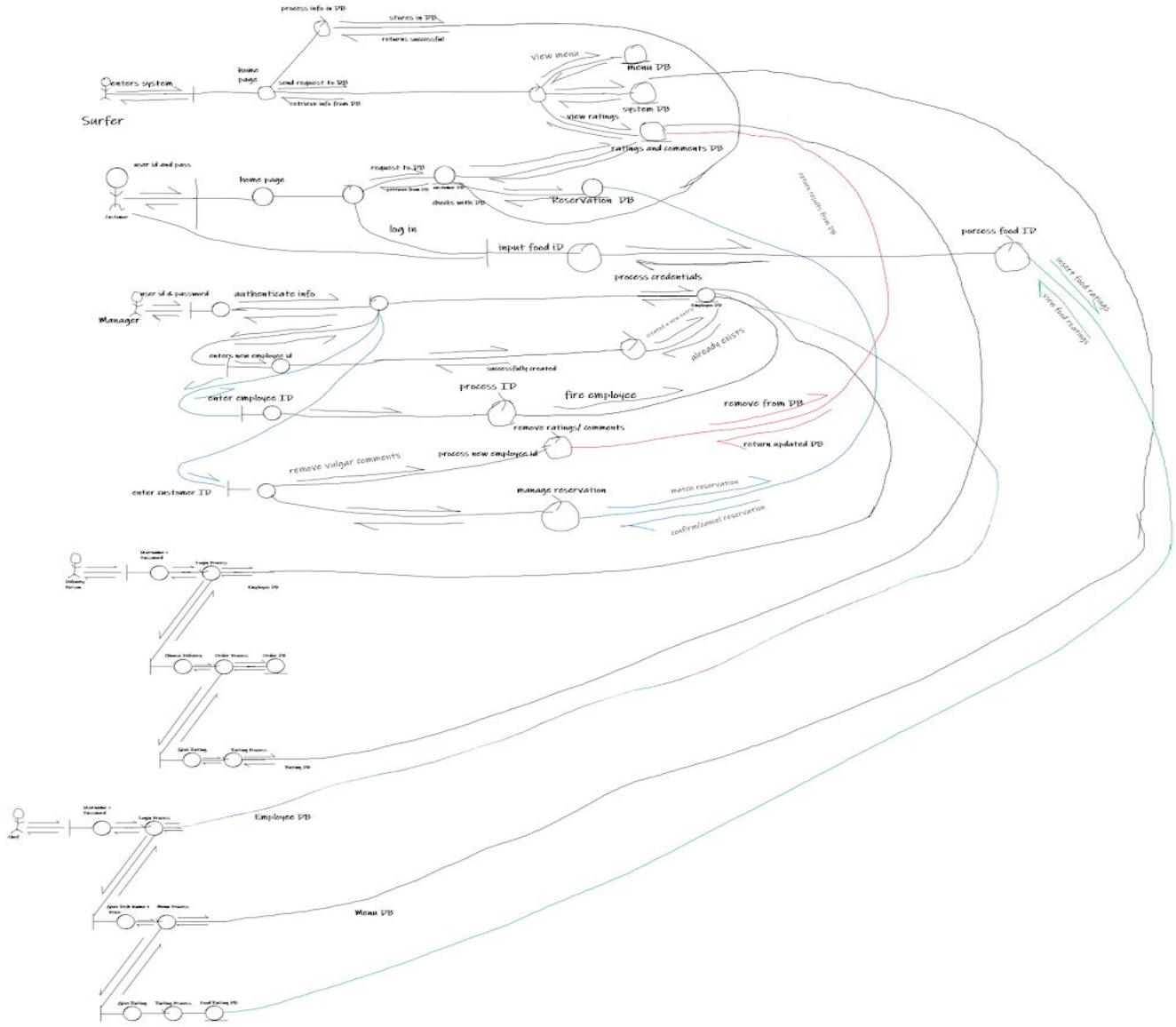
Date	Version	Description	Author
20/10/2020	1.0	Documented the Software Requirements Specification for the first time with a use case model diagram and a layout of the system to be produced	Mumtahid Akash Anvinh Truong MD Hossain Myriam Yumbla Sajid Mahmud
17/11/2020	2.0	Documented the Software Requirements Specification for the second time with a logic design approach. This logic design includes class collaboration diagram, sequence class diagram, and a formal approach of petri net diagram	Mumtahid Akash Anvinh Truong MD Hossain Myriam Yumbla Sajid Mahmud

Table of Contents

1) Introduction

1.1 Overall Collaboration Diagram

The (higher quality) collaboration diagram can also be found here under the following link and in the GitHub repository.



2) Use Case Diagrams

2.1 Create Account

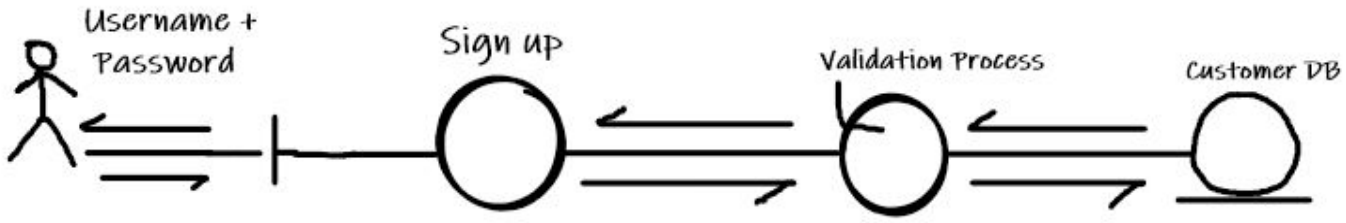
1. **Description:**

Normal Scenario:
Users are able to create an account to become a registered user and be added to the customer database. A unique userID must be provided and a password to help secure the account.

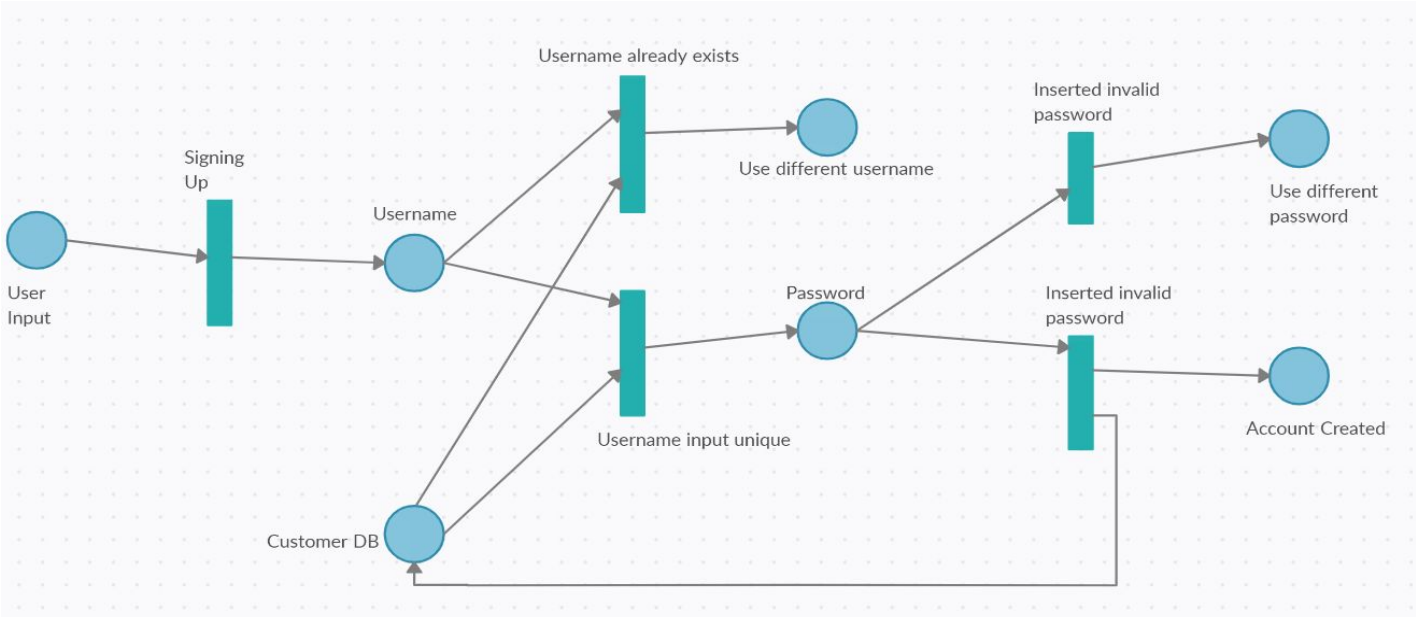
Exceptional Scenario:
An exception could be when the database is not connected to the server while the customer tries to sign up. This could result in:

a) Letting the customer access the website without the database having information on the customer, which would create problems while ordering items.
b) The website not allowed in the customer because the database connection is not confirmed.

2. **Collaboration Class Diagram:**



3. Petri Net:



2.2 Log In

1. Description

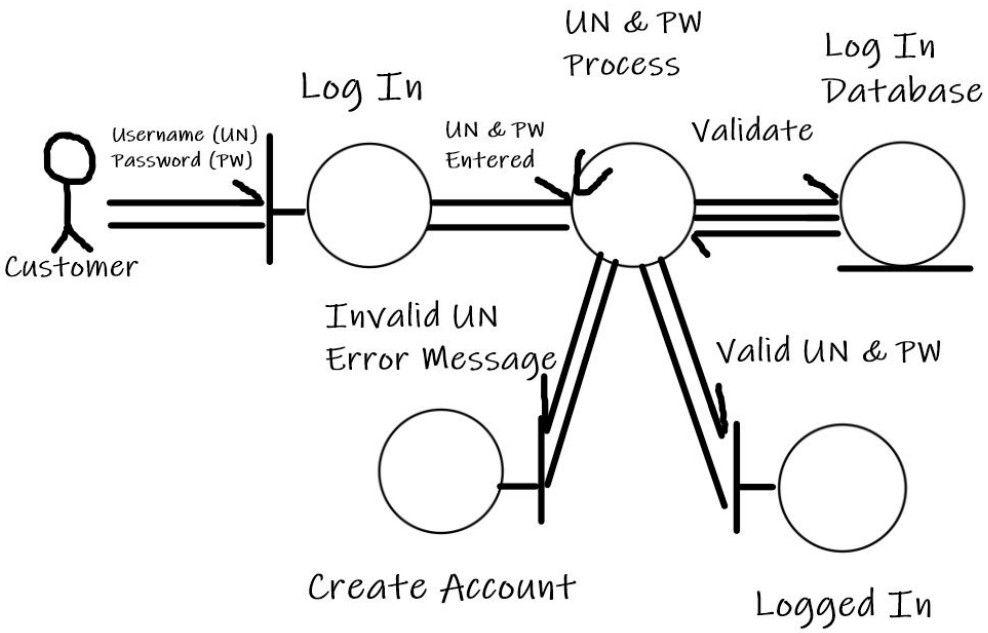
Normal Scenario:

A user logs in to their account and the system checks to see if the userID and password check out and are in the database. If it is not, then the user gets an error message indicating the account not to be found and takes them to create an account. However, if the userID and password are valid, then the user successfully logged in to their account.

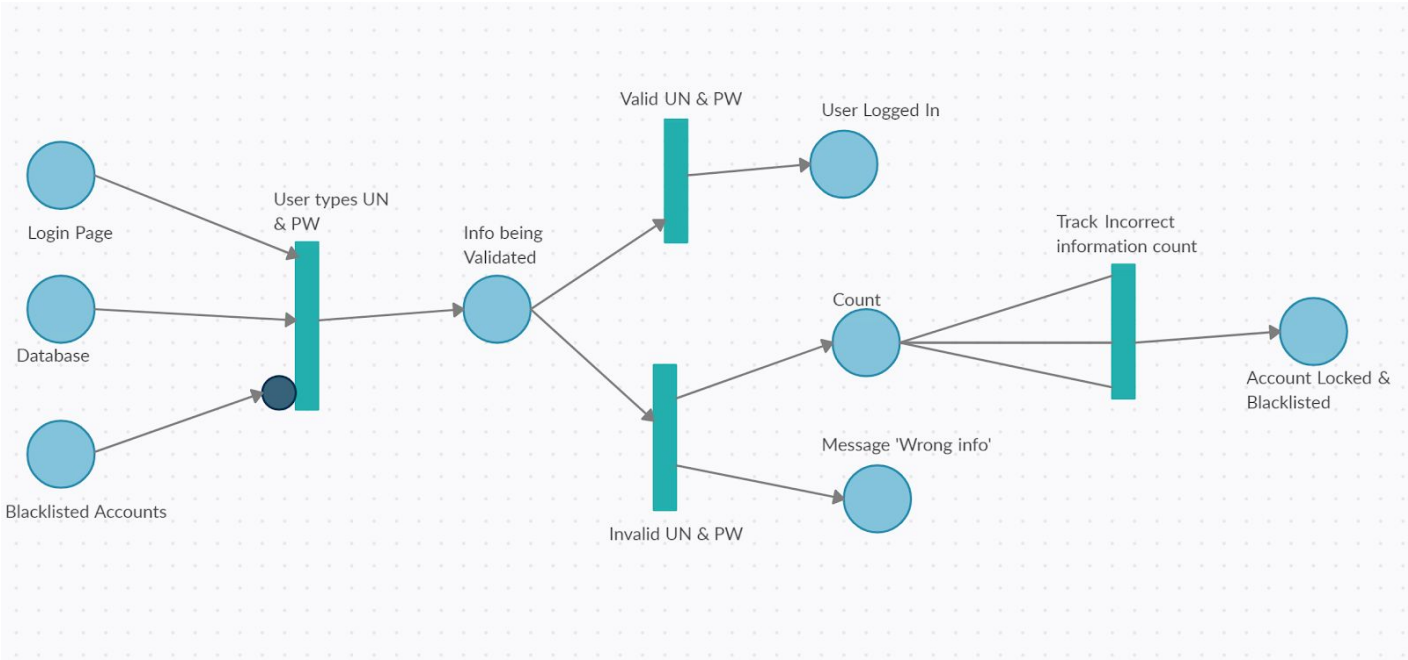
Exceptional Scenario:

A mass amount of users logging onto the website at the same exact time will overload the system. A massive amount of users having their login credentials put in incorrectly at the same time.

2. Collaboration Diagram



3. Petri Net



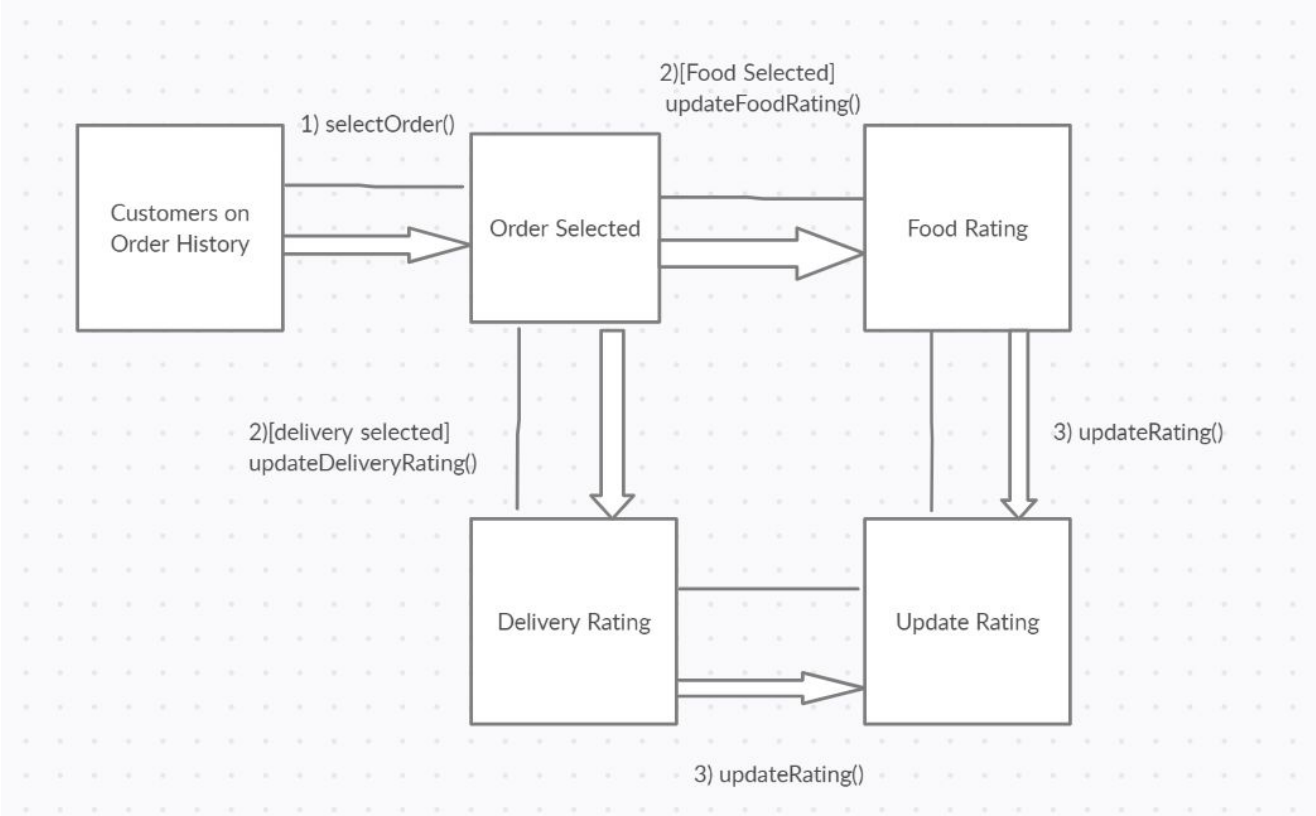
2.3 Ratings

1. Description

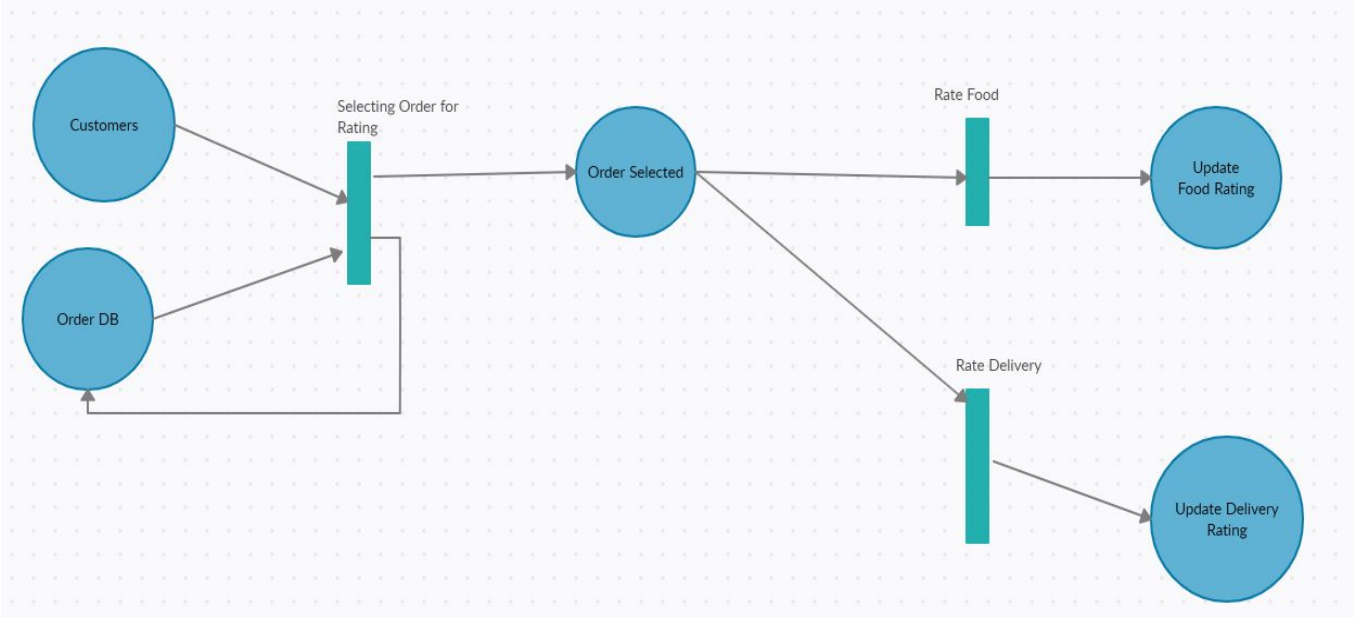
Normal Scenario:
Customers can provide ratings on both the food and the delivery service. The ratings are in the form of stars ranging from 1 to 5. The ratings are updated in the database.

Exceptional Scenario:
There can be the attempt to manipulate the ratings to be in favor of the product when in actuality, it was never ordered, which isn't allowed. Manipulation of the ratings is not permitted.

2. Sequence Class Diagram:



3. Petri Net:



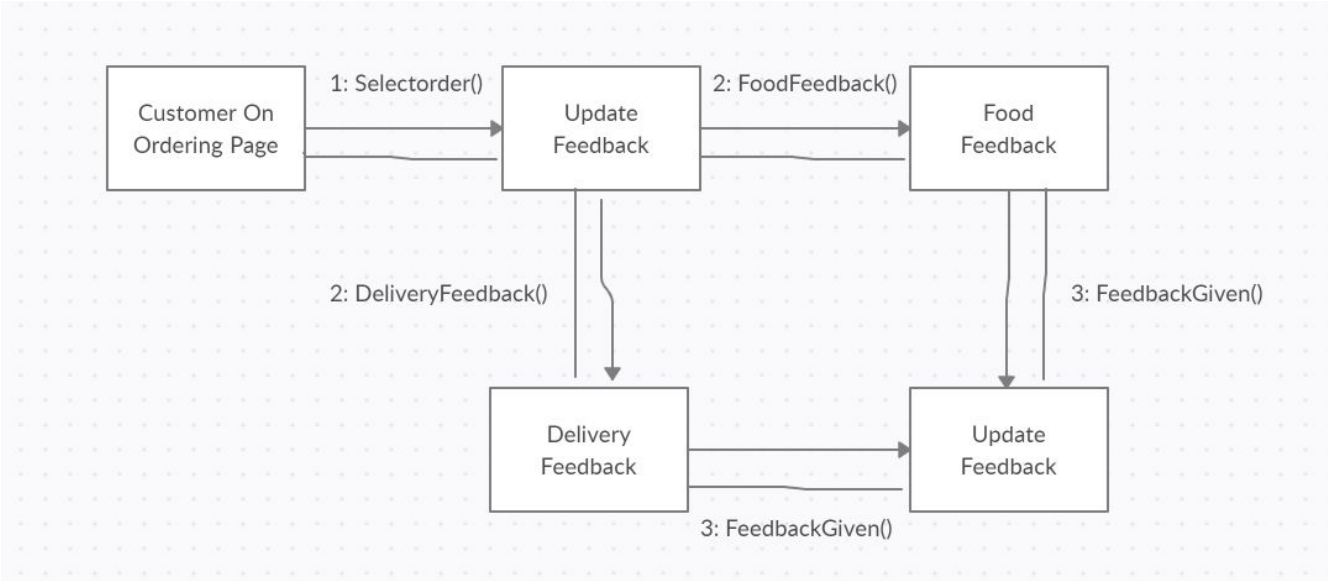
2.4 Feedback

1. Description

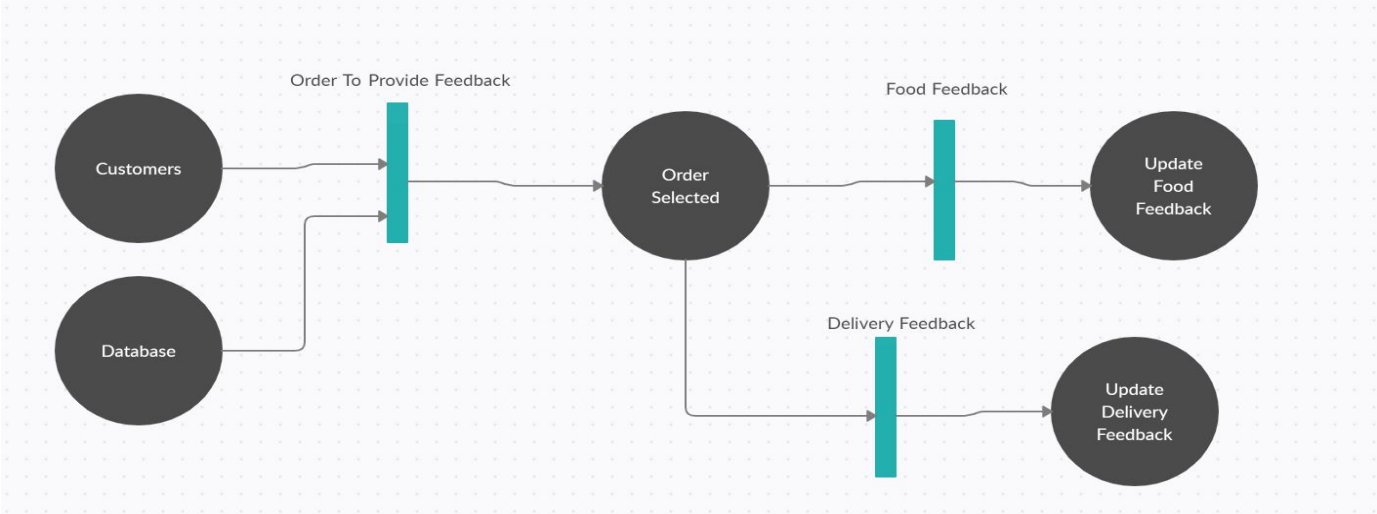
<u>Normal Scenario</u>:
Customers can provide feedback on both the food and the delivery service. The feedback is in the form of compliments and complaints.

<u>Exceptional Scenario:</u>
There can be the attempt to manipulate the feedback to be in favor of the product when in reality, it was never ordered, which isn't allowed. Manipulation of the feedback is not permitted.

2. Sequence Class Diagram



3. Petri Net



2.5 Place Order

1. **Description**

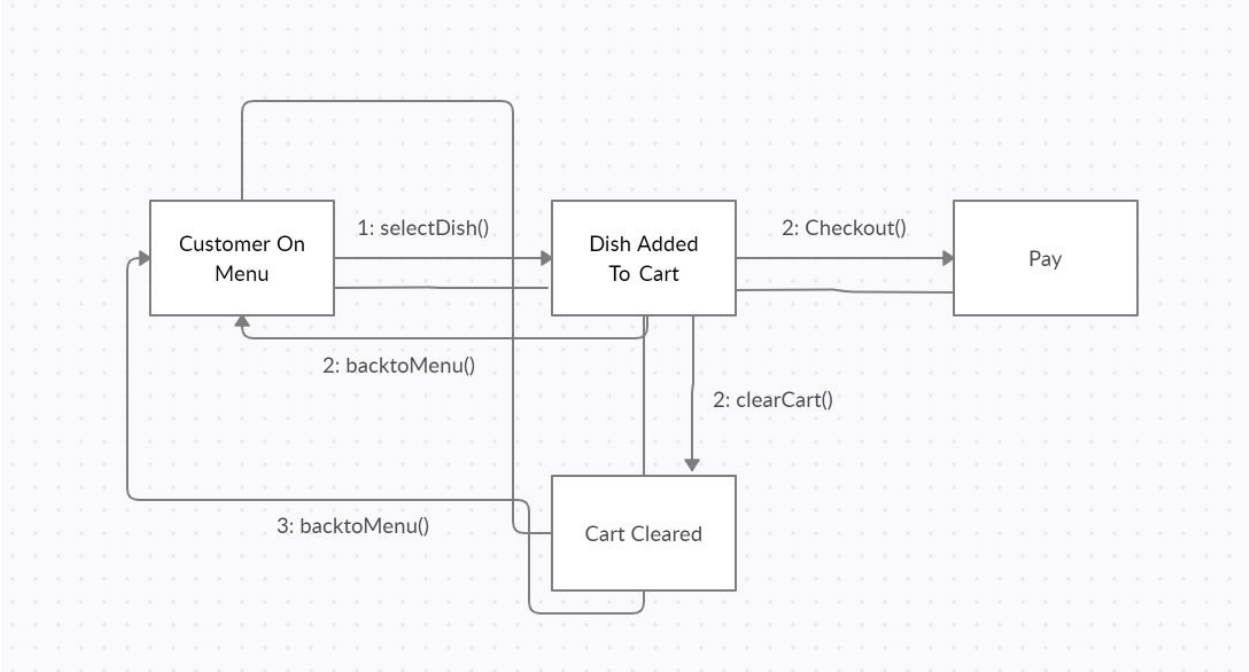
Normal Scenario:

Customers can click on a dish and add it to their cart from the menu. Additional dishes can be added to the cart if they so please. After all the dishes have been added, the customer proceeds to checkout and pays for the dishes within the cart.

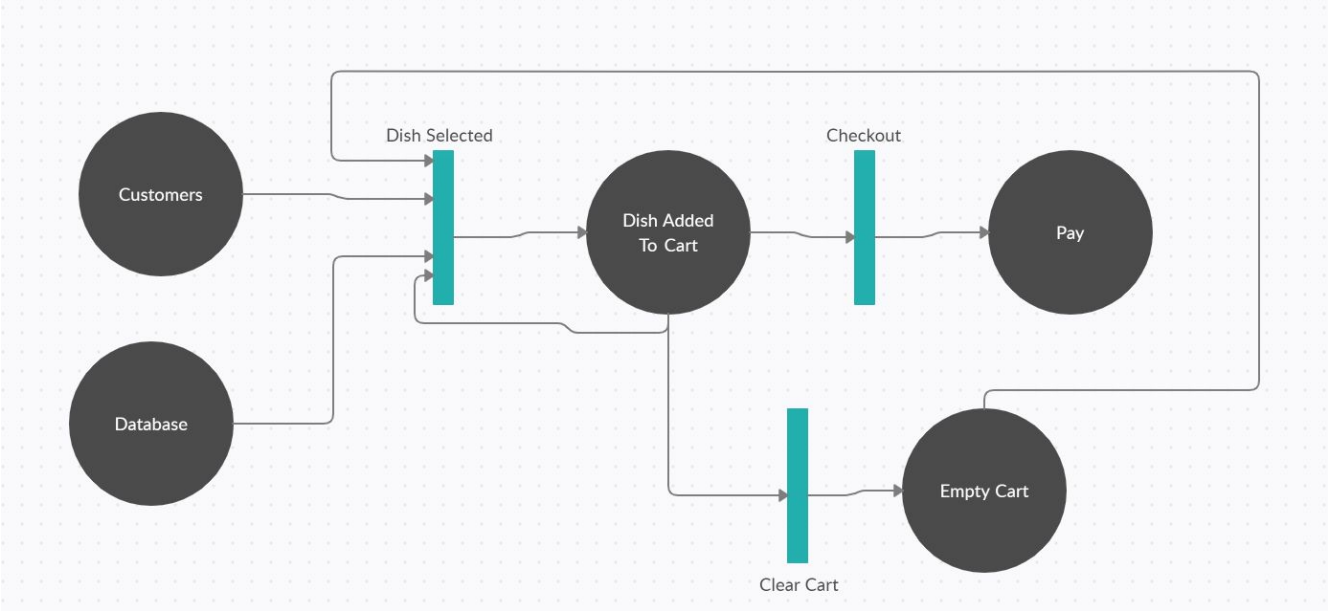
Exceptional Scenario:

A massive amount of customers checkout and pay all at the same time, overloading the system and database.

2. **Sequence Class Diagram**



3. **Petri Net**



2.6 Manage Account

1. **Description:**

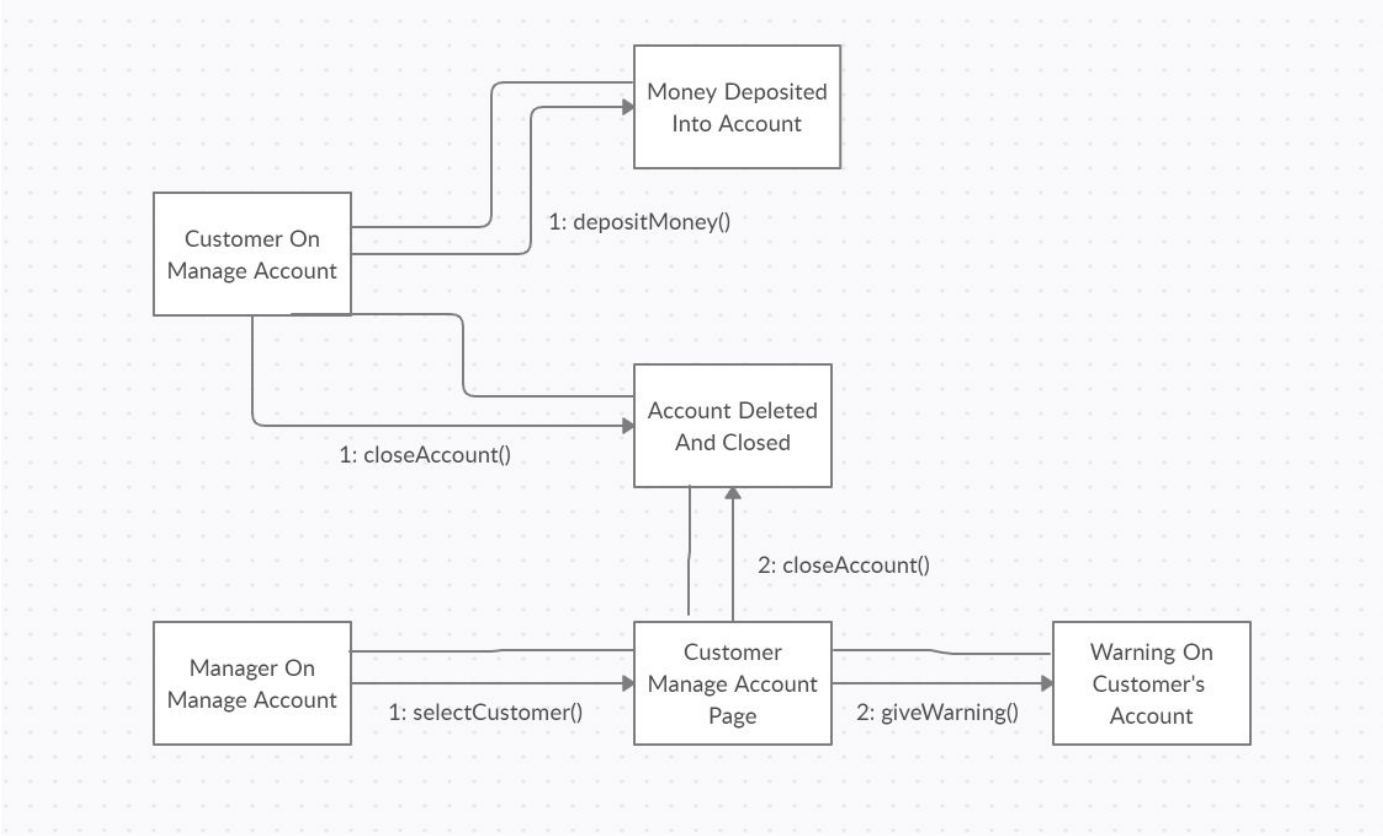
Normal Scenario:

Customers and the managers are able to manage their accounts. Customers view their own account, while the managers can view all the customer's accounts. Managers can issue warnings to the customers if their complaint is without merit.

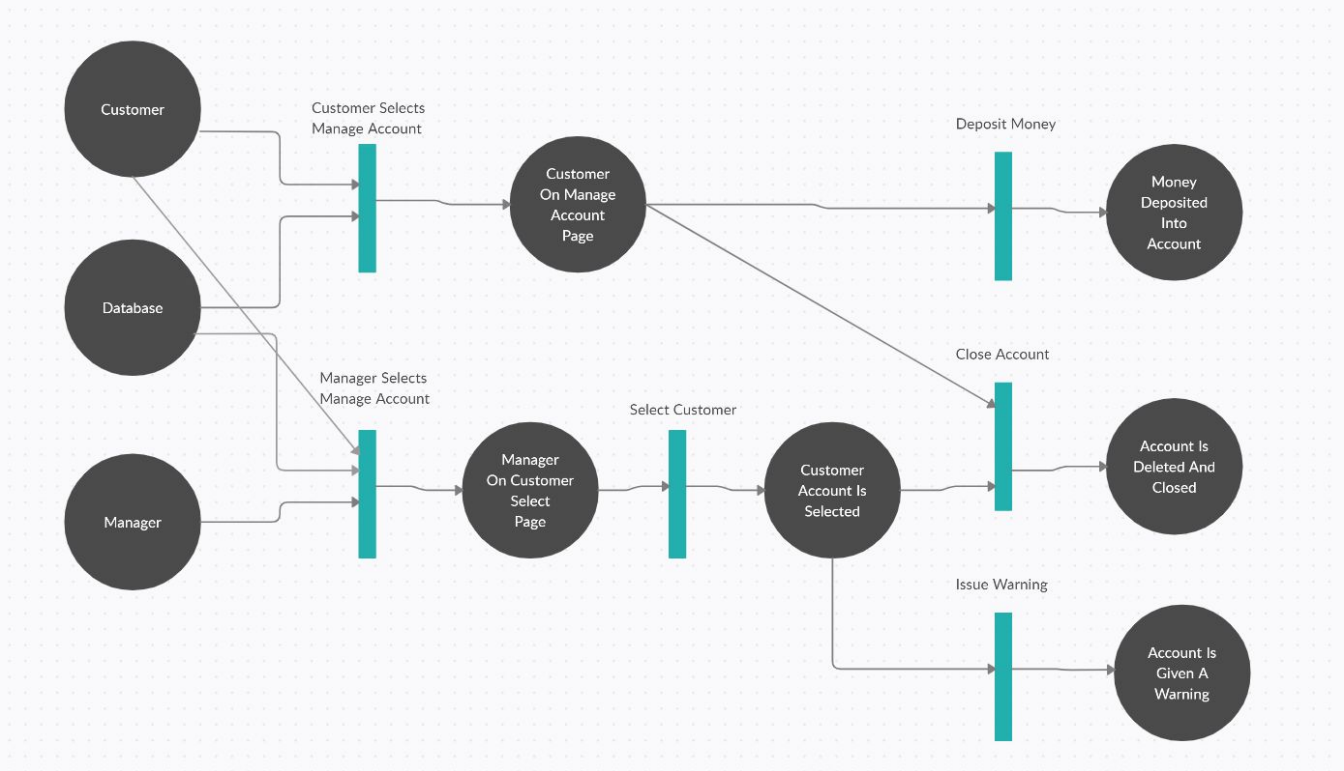
Exceptional Scenario:

Every customer deposits money into their account all at the same time, overloading the server and database. Customers somehow having access to other customer accounts.

2. Sequence Class Diagram



3. Petri Net



2.7 Manage Employees

1. Description:

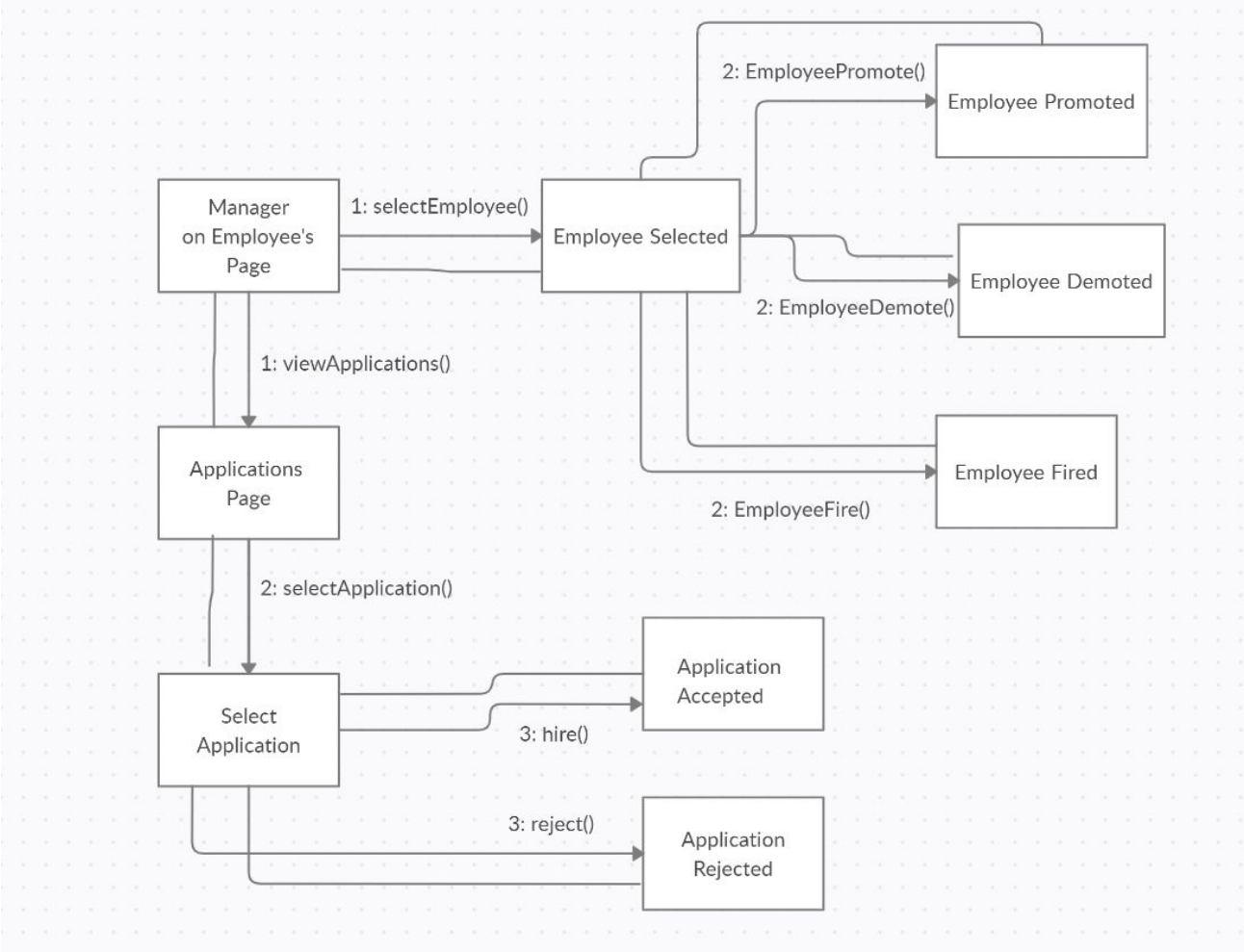
Normal Scenario:

The manager has the ability to select an employee, fire, or demote them based on their performance. They also review the applications and determine if a person is hired or rejected. A promoted employee receives a higher salary, while a demoted employee receives a reduced salary. If an employee gets demoted twice, then they are fired.

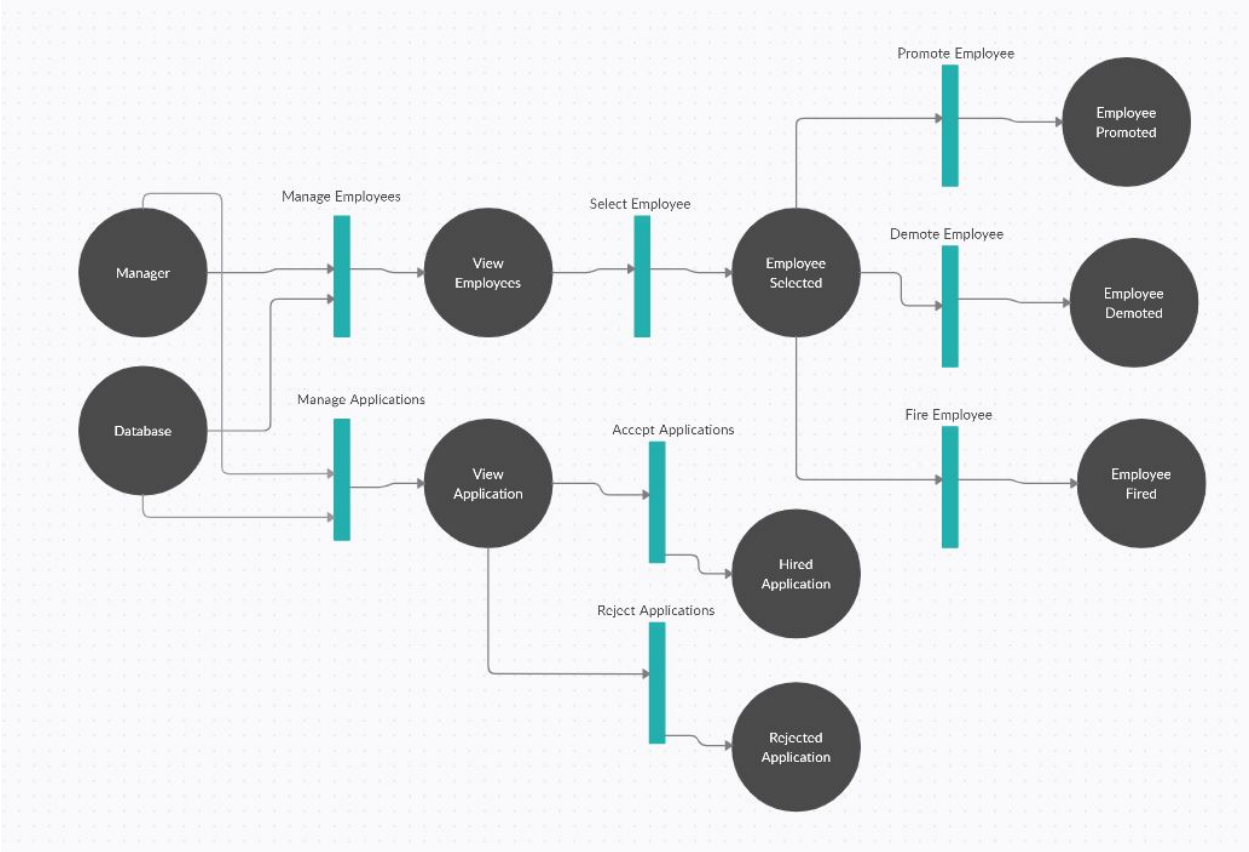
Exceptional Scenario:

Manager accidentally demotes, promotes, or fires an employee based on nothing substantial.

2. **Sequence Class Diagram**



3. Petri Net



2.8 Manage Customers

1. Description:

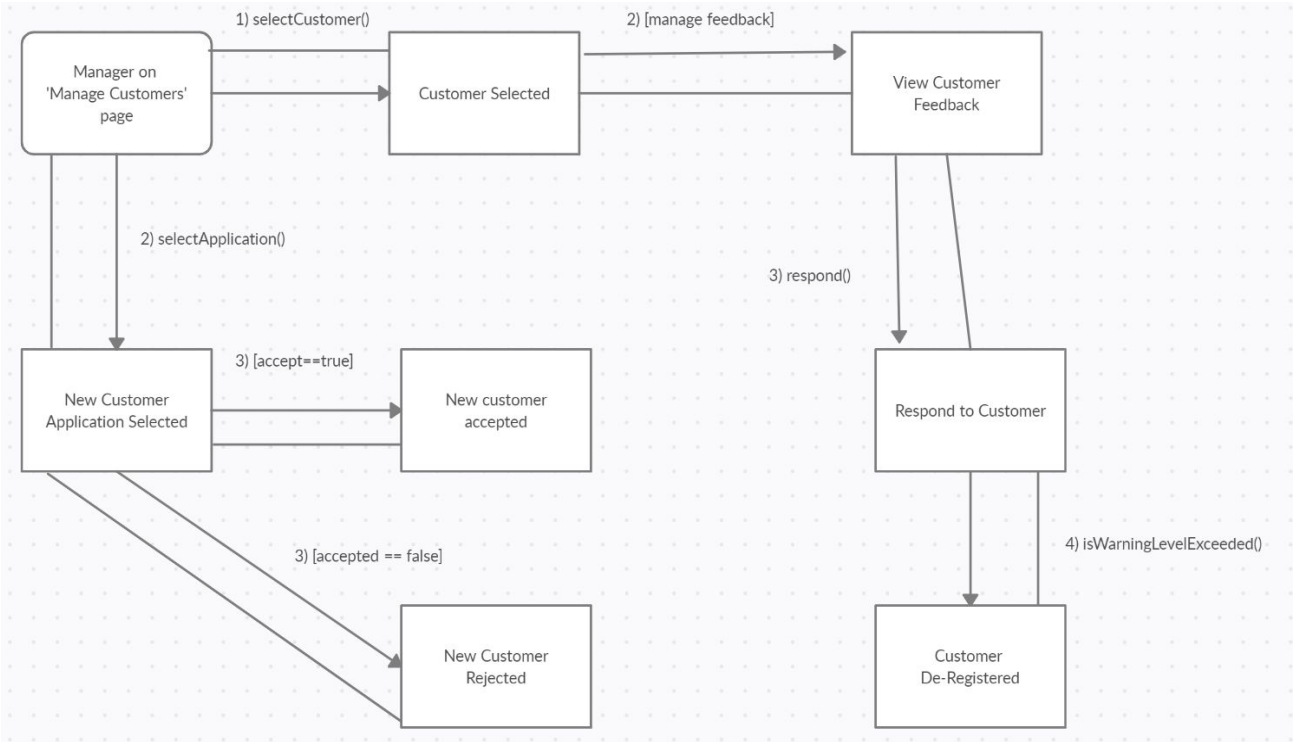
Normal Scenario:

The manager can process new and existing customers; he/she can accept or reject newcomers and de-register existing customers if they violate community guidelines.

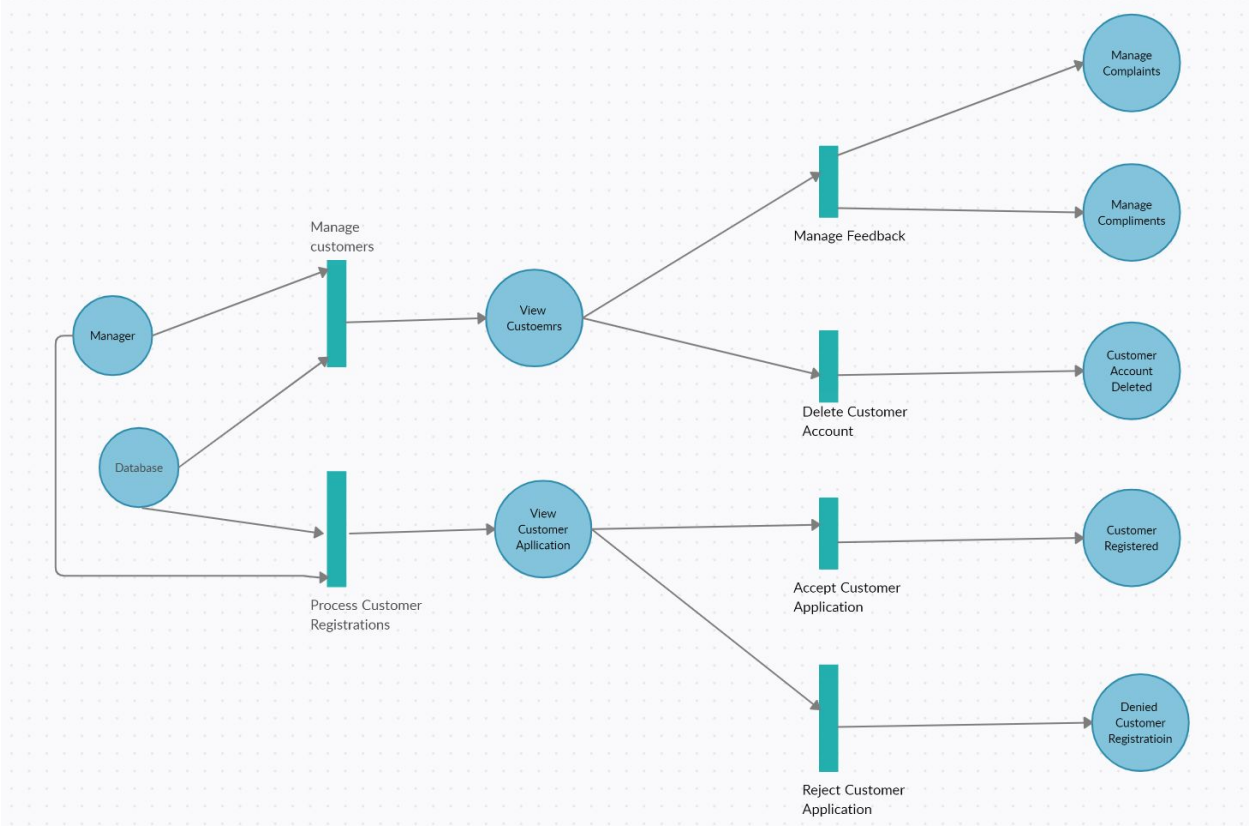
Exceptional Scenario:

An exception case would be giving a registered customer the privileges of a VIP customer. This type of modification without authorization is not allowed.

2. Sequence Class Diagram



3. Petri Net



2.9 Browse Menus

1. Description:

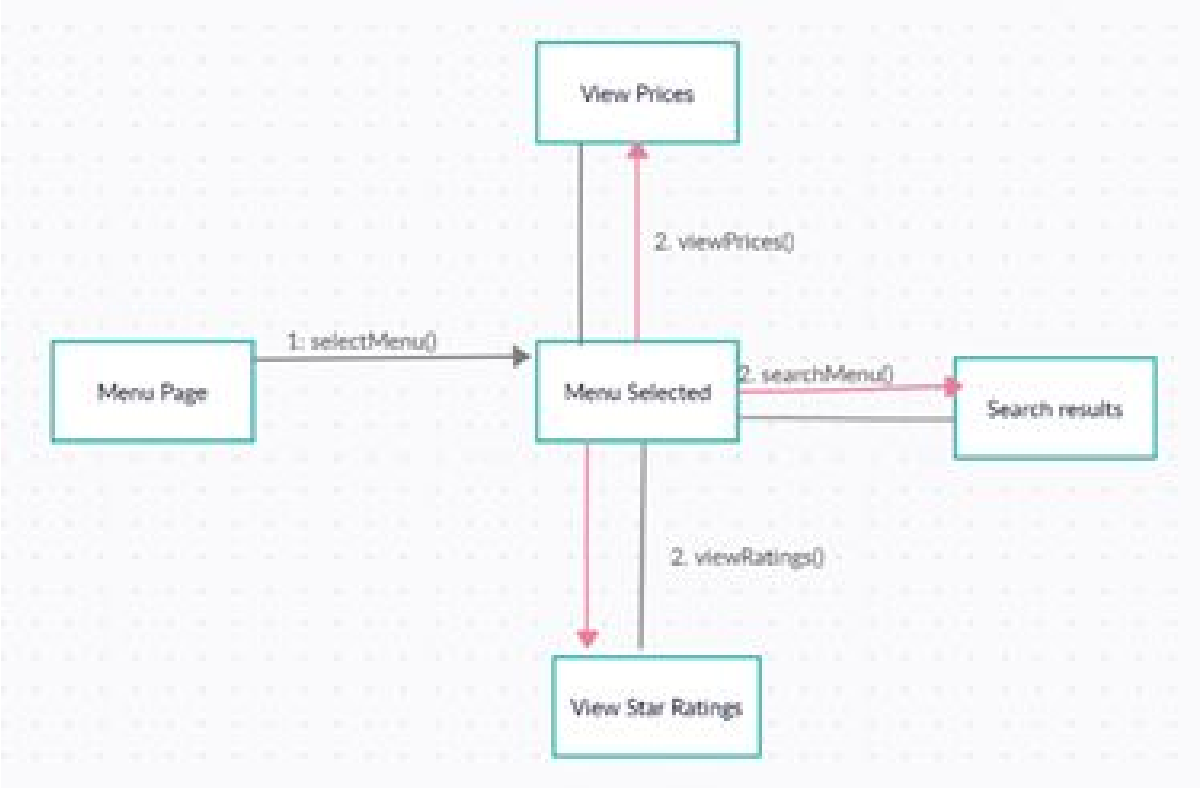
Normal Scenario:

Both the customers and the chefs will be able to view the menus. They also have the option to search for a specific item from the menu. Once an item is selected within the menu, both customers and chefs can view its star ratings or prices. Star ratings signify how much it is liked by other customers 5 being the highest and 1 being the lowest.

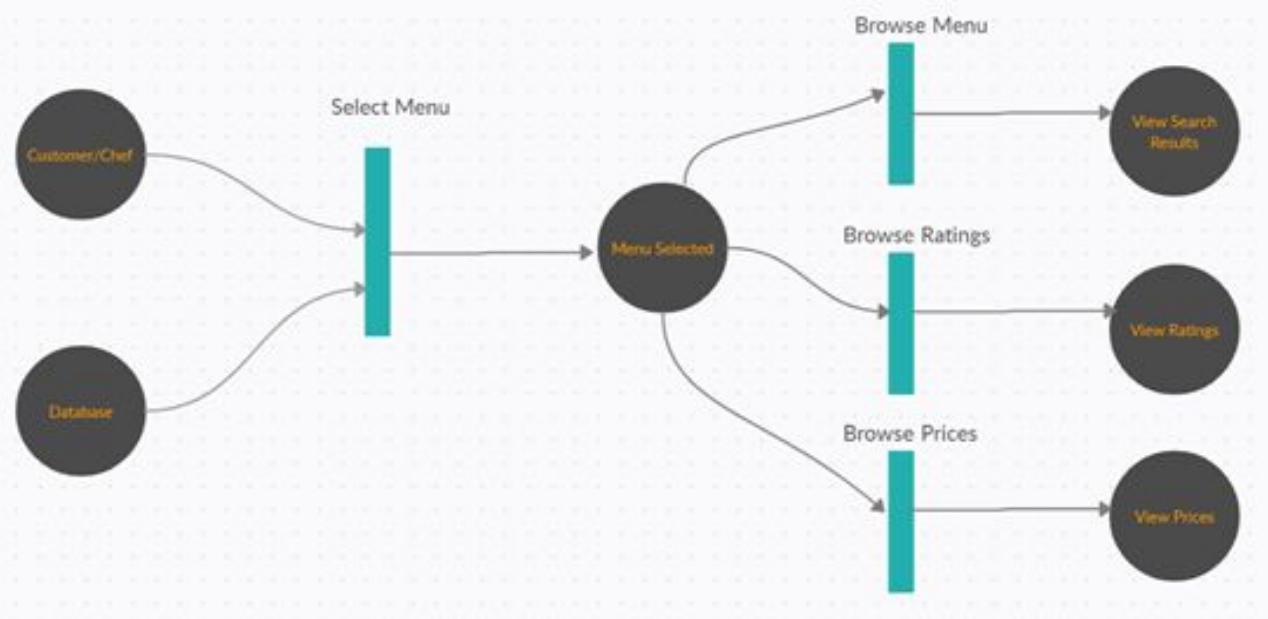
Exceptional Scenario:

In an exceptional case a customer might try to modify the menus either by changing the menu item name or the prices or the star ratings. Under no circumstances a customer could be allowed to do that.

2. **Sequence Class Diagram**



3. **Petri net diagram**



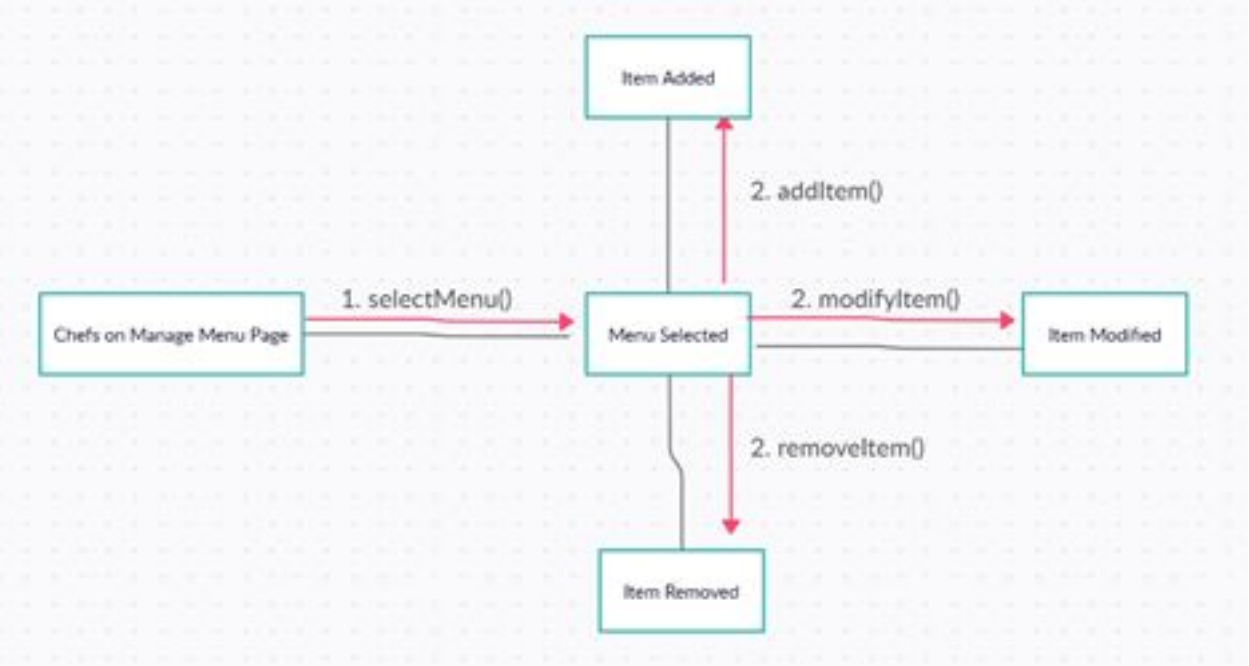
2.10 Manage Menus

1. **Description**

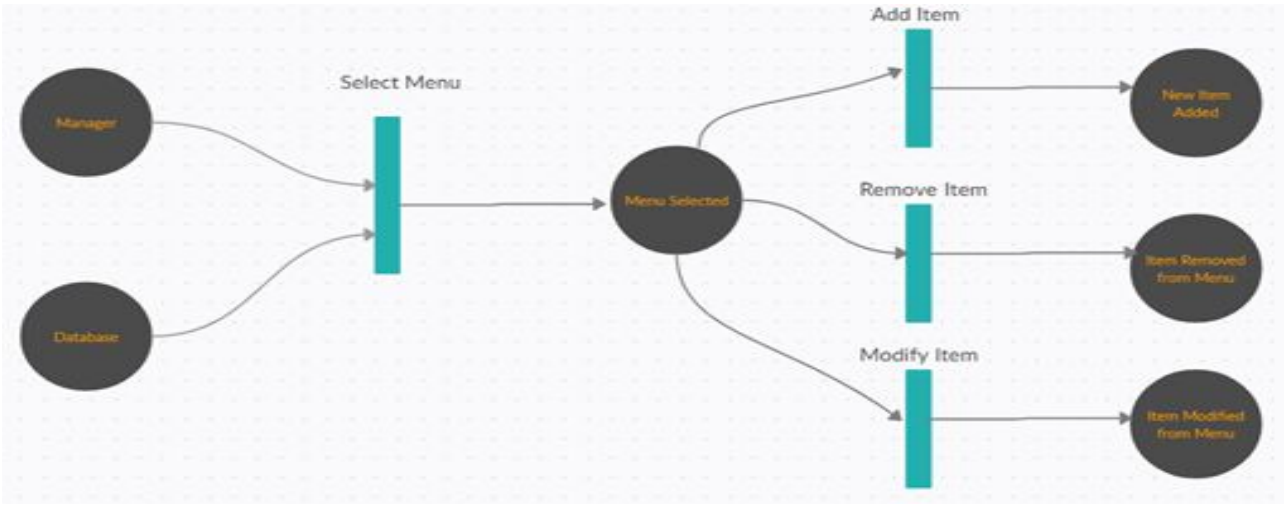
Normal Scenario:
The right to manage menus is given only to the chef and no other person should be able to access it. A chef can either add, remove, or modify items from the menu.

Exceptional Scenario:
In an exceptional scenario, a customer may try to change the menu which should not be a valid operation for customers. In other cases, a chef may try to wrongfully try to modify the ratings which the chef shouldn't have access to.

2. Sequence Class Diagram



3. Petri net diagram



2.11 Pick Up

1. *Description*

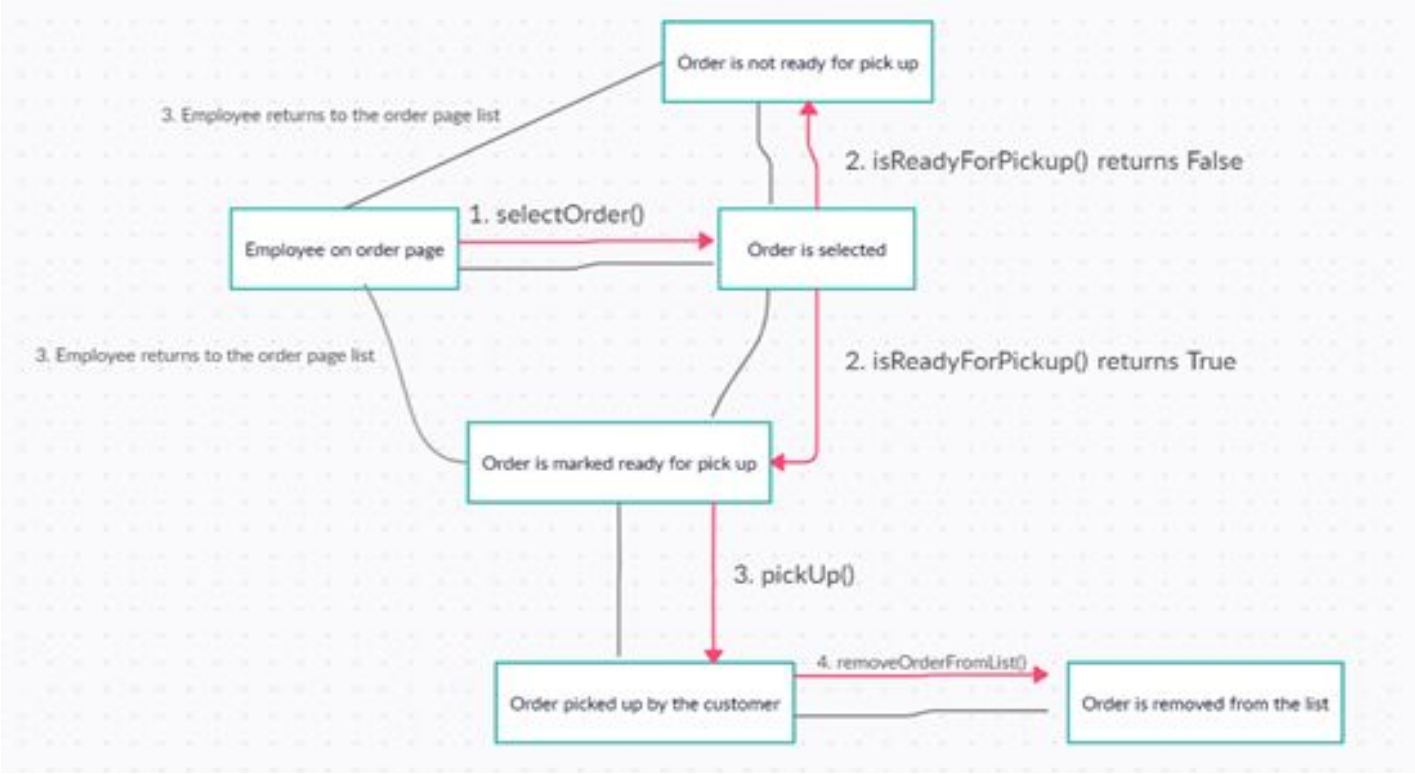
Normal scenario:

Customers and employees have access to the pickup section. Once an order is ready to pick up, customers can go to the restaurant to pick up their order. Employees on the other hand will make sure that all the items requested by the customer are there and ready for pick up.

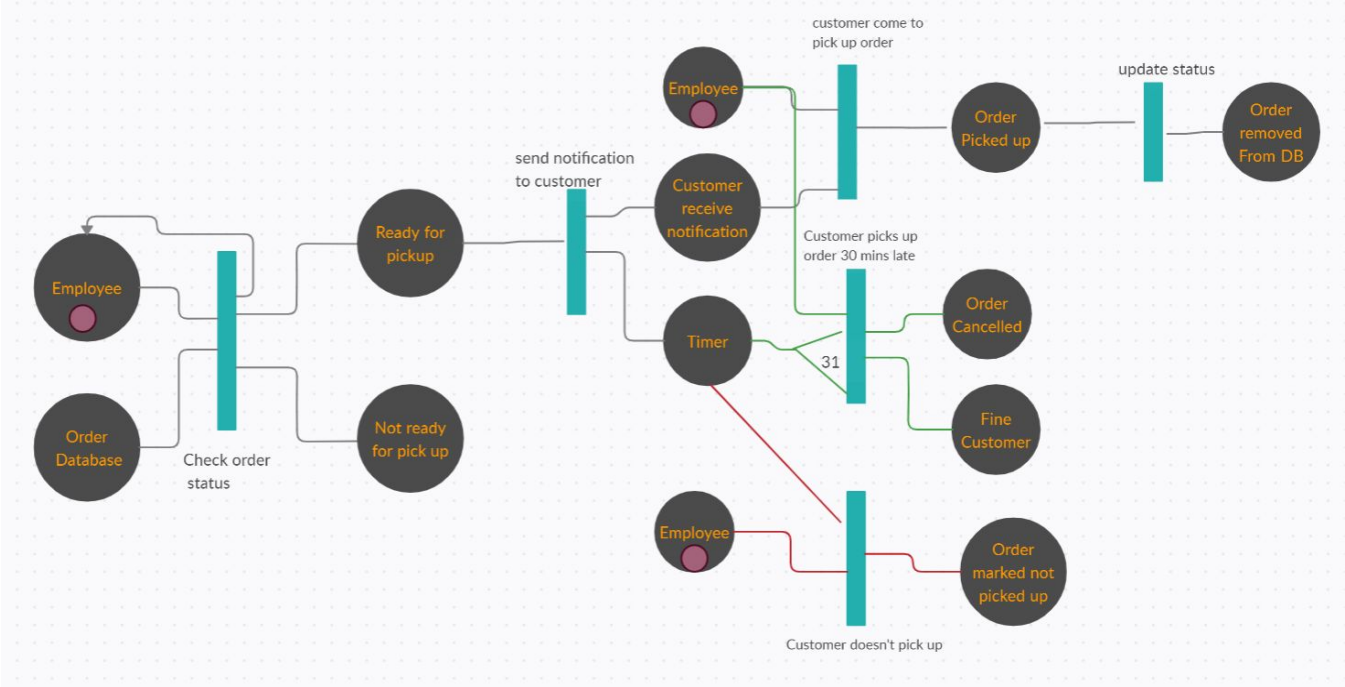
Exceptional Scenario:

Employees can wrongfully mark an order for pick up even though the order isn't ready for pick up**.**

2. Sequence Diagram



3. Petri net diagram



2.12 Reservation

1. Description

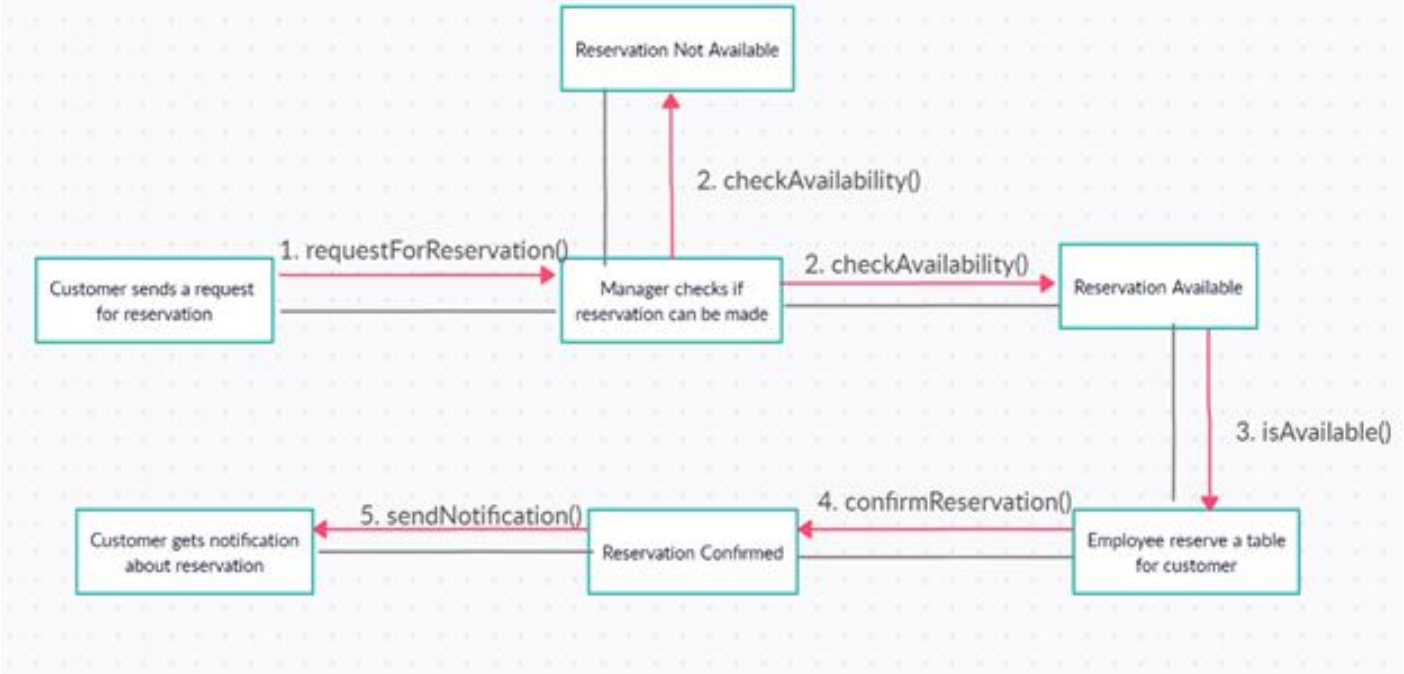
Normal Scenario:

Customers will make a reservation through the system. Once they make the request for the reservation, the manager will see if the request for reservation can be accommodated. If the manager can accommodate the customer request, he/she will accept the reservation and tell an employee to reserve a table for that customer. Once the reservation is complete the customer will get a notification.

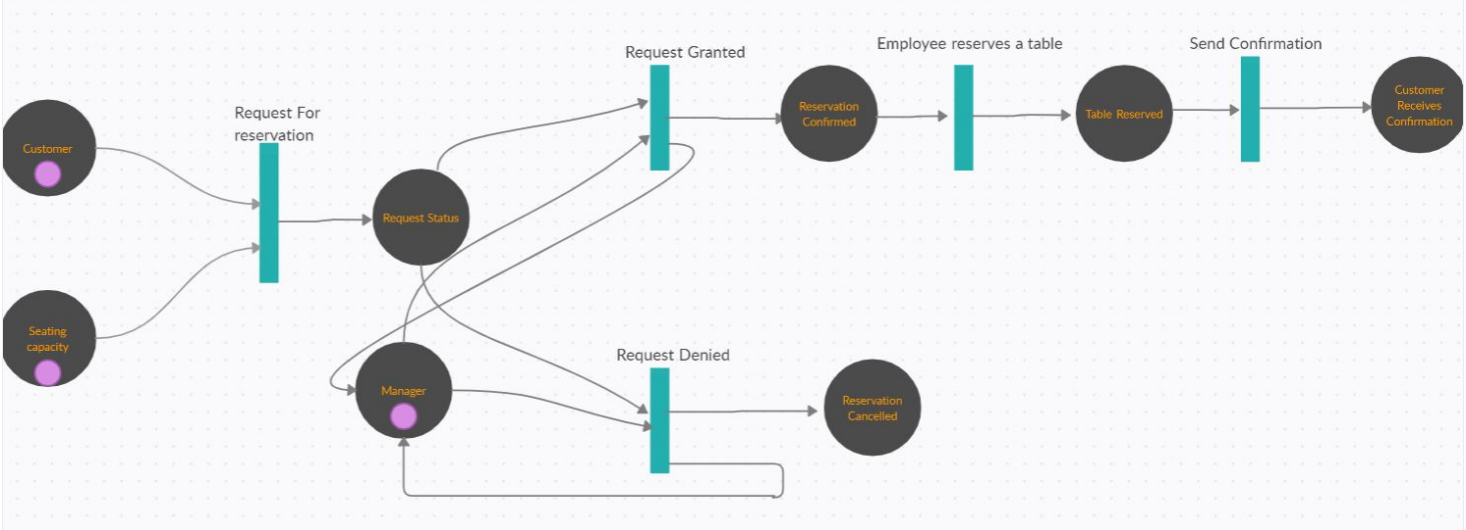
Exceptional Scenario:

Customer made a request for a reservation but even though the request cannot be accommodated by the restaurant, the manager confirmed the reservation. In that case the manager will have to cancel that reservation.

2. Sequence Class Diagram


Reservation Not Available
2. checkAvailability()
1. requestForReservation()
Customer sends a request for reservation
Manager checks if reservation can be made
2. checkAvailability()
Reservation Available
3. isAvailable()
4. confirmReservation()
5. sendNotification()
Customer gets notification about reservation
Reservation Confirmed
Employee reserve a table for customer

3. Petri net diagram



2.13 Delivery

1. Description

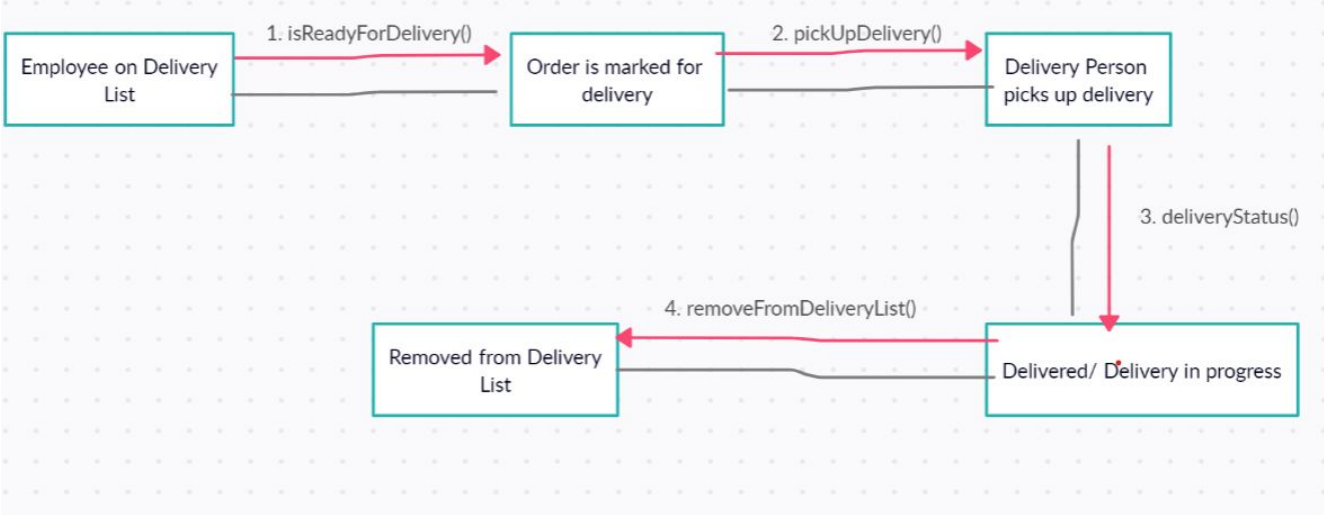
Normal Scenario

Once the order is marked ready for delivery by an employee, a delivery person will go to the address mentioned by the customer to drop off the food. Once the customer receives the food, the order will be marked with a delivered message.

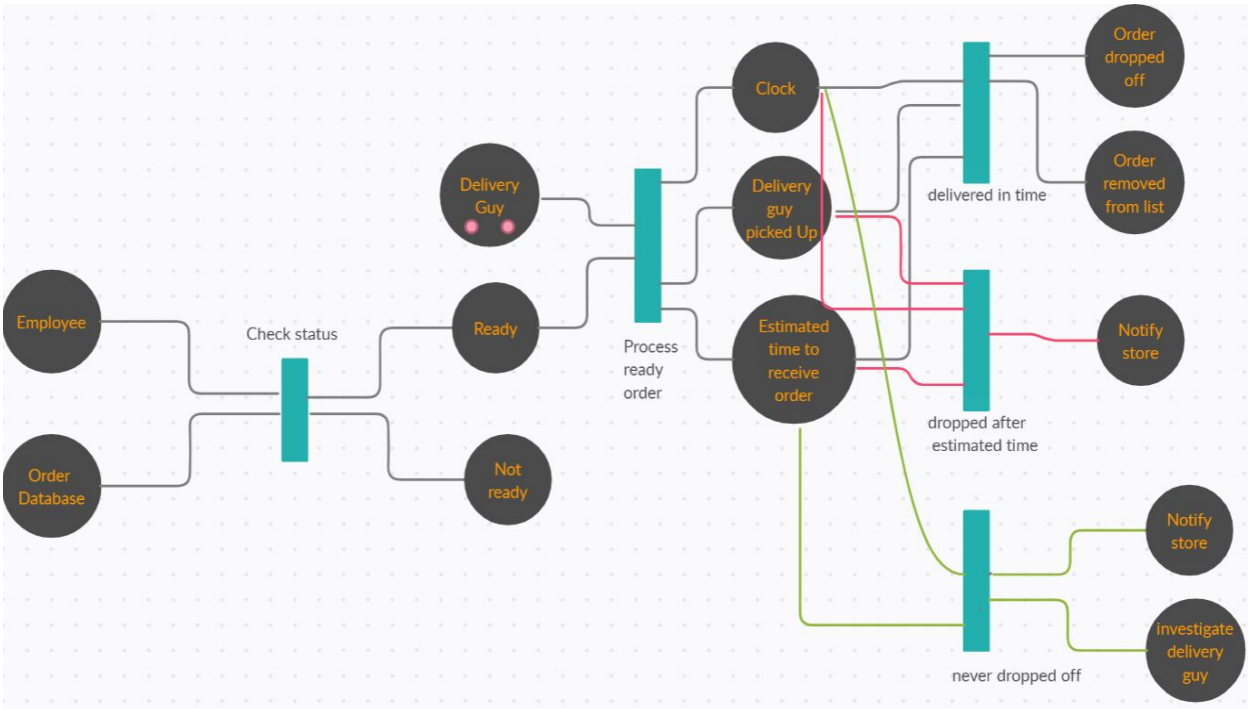
Exceptional Scenario

A delivery person may go to the wrong address and mark the order as delivered. In other cases, the customer may put the wrong address for delivery.

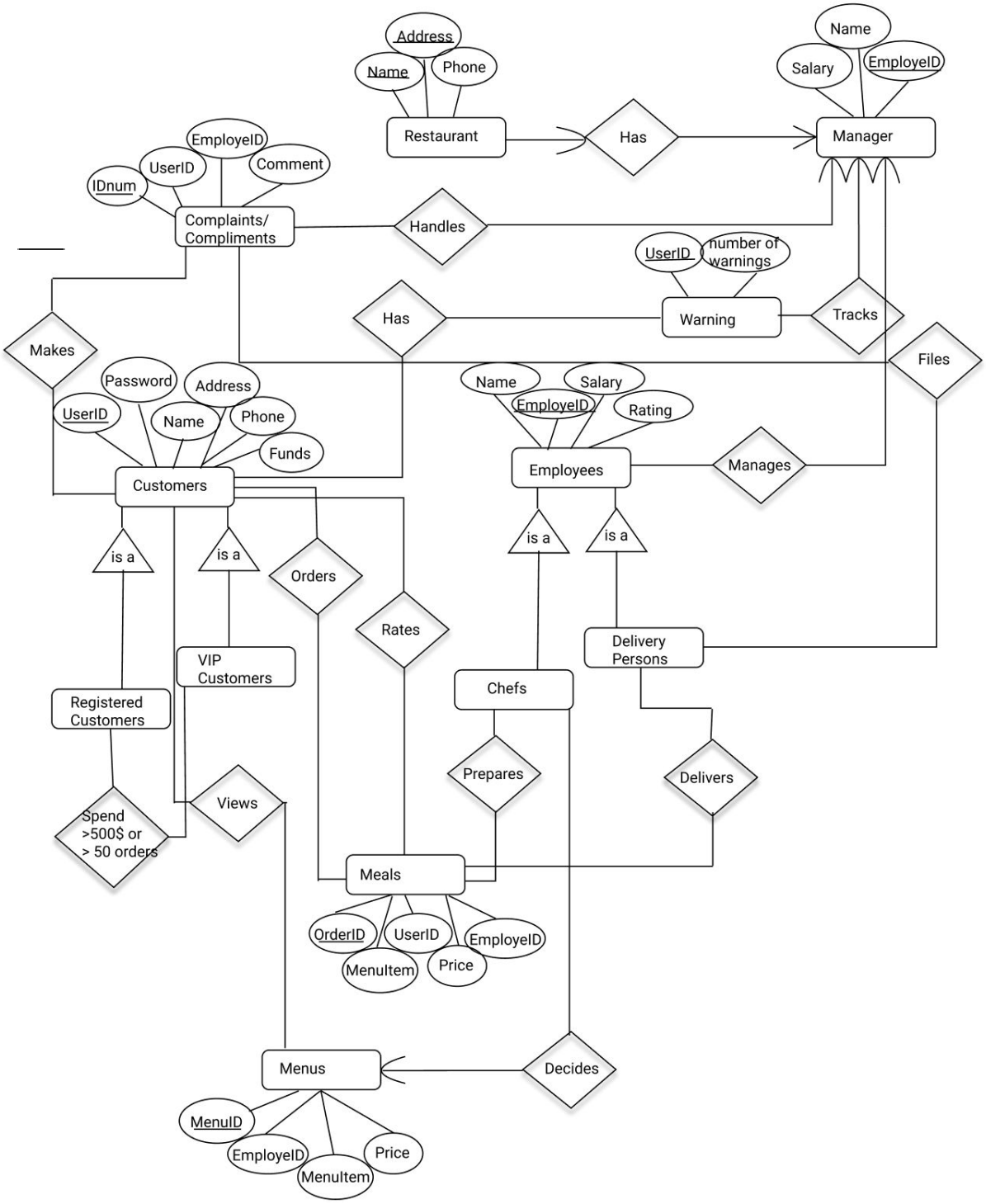
2. Sequence Class Diagram



3. Petri net diagram



3) ER Diagram


Address
Name
Phone
Restaurant
Has
Name
Salary
EmployeID
Manager
EmployeID
UserID
Comment
IDnum
Complaints/
Compliments
Handles
UserID
number of
warnings
Warning
Has
Tracks
Makes
Files
Password
Address
UserID
Name
Phone
Funds
Customers
Name
Salary
EmployeID
Rating
Employees
Manages
is a
is a
Orders
Rates
is a
is a
VIP
Customers
Registered
Customers
Chefs
Delivery
Persons
Prepares
Delivers
Spend
>500$ or
> 50 orders
Views
Meals
OrderID
UserID
EmployeID
MenuItem
Price
Menus
Decides
MenuID
EmployeID
Price
MenuItem

4) Detailed Design

4.1 Methods for Surfer

a) Create Account (userID, password) {
Take userID input from the user
If (userID in Customer Database):
Print ('Choose different userID')

Else:
Take password input from user
If (password character requirements not met):
Print('Retype the password')
Else:
Insert userID & password into database
Print('You have successfully signed up') }

b) Browse Menus (MenuID) {
Return the corresponding menu based on the menuID, sorted by ratings}

c) Search Menu (MenuItem) {
Search the menuitem in the database
If (found):
Return the item
Else:
Return message 'Not Found' }

4.2 Customer Methods

1. Manage Account (userID):
Give user options: Deposit money, Close account
If (deposit money is selected):
Money deposits into customer account
If (close account is selected):
Customer loses access to the account

Remaining funds sent back to customer's through his/her payment method.

2. Post Compliment (userID):

   Customers can post compliments about the chef or the delivery person.

   If (compliments doesn't have any taboo word)
   If (Customer == VIP)
   Increment compliment count by 2

   Else:
   Increment compliment count by 1

   Compliments will be posted.

   Else:
   If (compliments' taboo count < 3)
   The taboo words will be changed to *** and the customer will receive 1 warning.

   Else:
   The compliments won't be posted.

3. Post Complaint (userID):

   Customer can post a complaint about a chef or delivery person

   If (customer post complaint without merit)
   Customer will receive 1 warning from the manager.

   Else:
   If (complaint's doesn't have any taboo word)

   If (customer is a VIP)
   Increment complaint by 2

Else:

Increment complaint by 1

Complaints will be posted.

Else

If (taboo count < 3)

Complaint's with taboo words will be changed to *** and the customer will receive 1 warning.

Else

The complaints won't be posted.

4. Post Rating (userID, rating, menuId):

   Customers can post a star based rating where 0 being the lowest and 5 being the best.

   The ratings will be added to the database.

5. Place Order (userID, menuID):

   Customers can add items to their cart and order for a delivery.

   If (customer is a VIP):

   For every ordinary dish:

   Ordinary dish price = ordinary dish price – (ordinary dish price * 0.10)

   tell the customer that the order was processed.

   If (customer fund < food cart total price):

   Warn the customer of insufficient funds and ask to add more funds

   Else:

   Tell the customer that the order was processed.

6. Reservation (userID, time, seat)
   Customers can select a date and time to reserve a table.

   If (time or seat is unavailable):
      Customers will be asked to enter a new time slot or seats.

   Else:
      If (fund < reservation price):
         Warn the customer of insufficient price and tell them to add more funds.

      Else:
         Process the reservation

7. Deposit fund (userID, amount)
   Customers will login with their userID and add an amount to their existing account's amount.
   The amount will be added to the database.

8. Browse Menu ()
   Customers can view the menu item and its star ratings.

9. Login (userID, password):
   If (userID and password exists in database):
      Customer will be logged in.
      Customer can visit their personalized homepage.

   Else:
      Customer will get an error for incorrect credentials and will be asked to enter credentials again.

      If (login tries > 5):
         The account will be locked out and the customer will need to file a report to get it fixed.

10. Logout (userID):
    Once a customer logs out the running session will close.
    Customers will return to the main menu.

4.3 Methods for Delivery Person

a) Browse Delivery(Orders database)
    Get the list of orders to be delivered

b) Deliver Food (Order Number, userID):
    Select the order based on distance from Delivery person's location to the restaurant's address
    Get the order information from user's database
    Set the order status as 'Out for Delivery'

c) Log In (userID, password) {
    Check if the userID & corresponding password exists in database
    If userID & password exists:
        Take user to logged in homepage

    Else:
        Return an error message & to try again

d) Search Menu (MenuItem) {
    Search the menuitem in the database
    If found:
        Return the item
    Else:
        Return message 'Not Found' }

e) Log Out() {
    Change login session = false
    Change session user= null
    Take screen to homepage }

4.4 Methods for Manager

a) Log In (userID, password) {
    Check if the userID & corresponding password exists in database
    If userID & password exists:
        Take user to logged in homepage

    Else:
        Return an error message & to try again }

b) Close Account (userID){
    Send user's in-store funds to user's payment method
    Delete user's record from the database }

c) Log Out() {
    Change login session = false
    Change session user= null
    Take screen to homepage }

d) Hire Employee(name, salary) {
    Set rating = null
    If (employee = Chef):
        Set employeeId beginning with CH…..

    Else if (employee= Delivery Person):
        Set employeeId beginning with DP…..

    Else:
        Set employeeId beginning with ST…...
    Add name,employeeId,salary,rating to the employee database}

e) Fire Employees(employeeId) {

Remove the record of corresponding employeeId from the employee database }

f) Demote Employee(employeeId){
Find employeeId record in database
Set salary to 5% less
Increase demotion count by 1
If demotion count >= 2:
Fire Employee(employeeId) }

g) Promote Employee(employeeId){
Find employee record in database
Set salary to 5% more}

h) Give Warning (userID) {
Increase user's warning count by 1 }

4.5 Methods for Chef

a) Log In (userID, password){
Check if the userID & corresponding password exists in database
If userID & password exists:
Take user to logged in homepage

Else:
Return an error message & to try again

b) Log Out(){
Change login session = false
Change session user= null
Take screen to homepage }

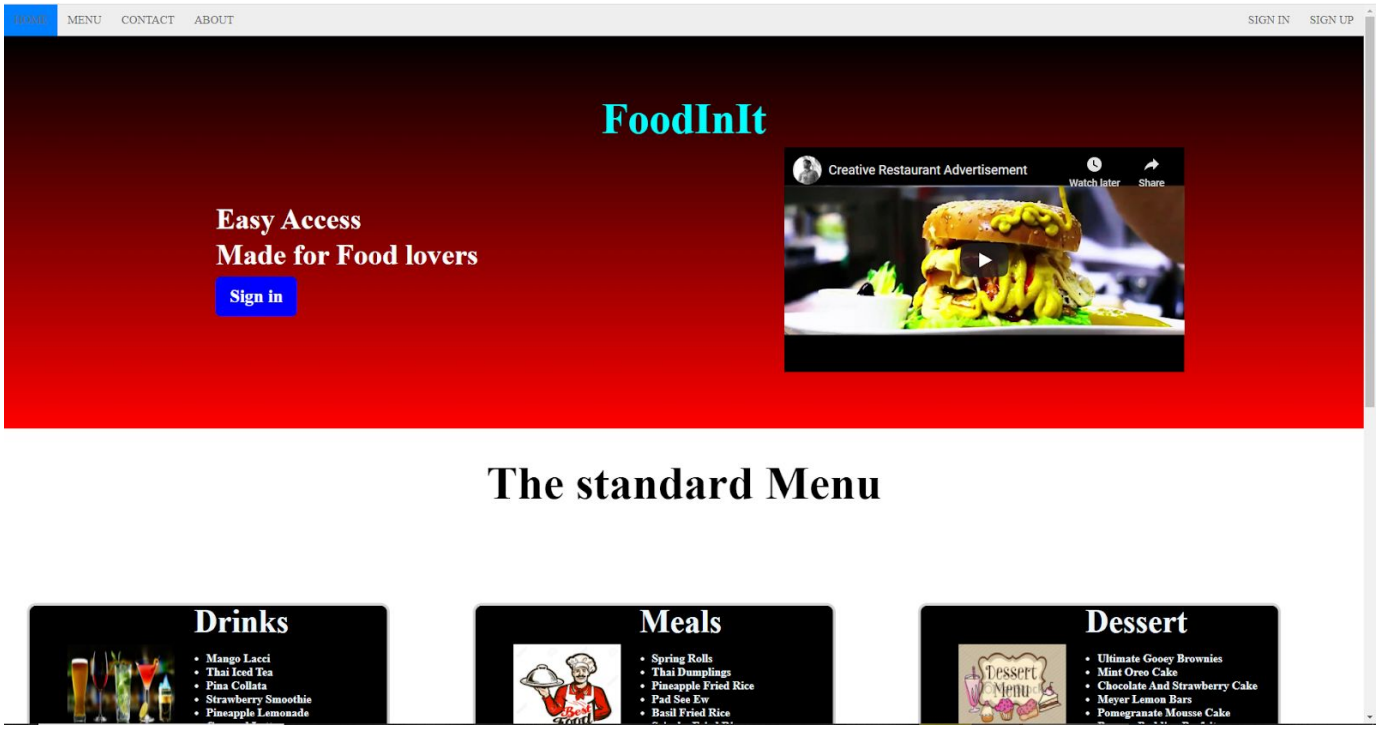
c) Add Menu Item (chefID, MenuID, itemName, price) {

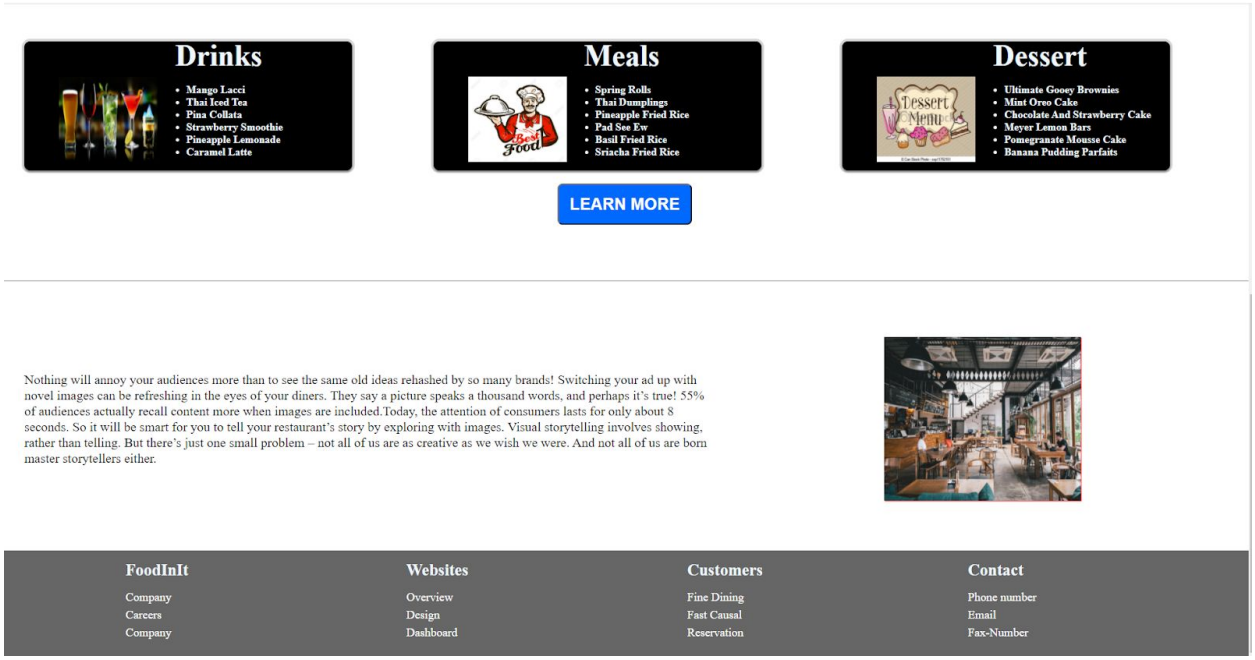
Input item name and price onto the menu.
}

d) Remove Menu Item (chefID, MenuID) {
Remove item from the menu
}

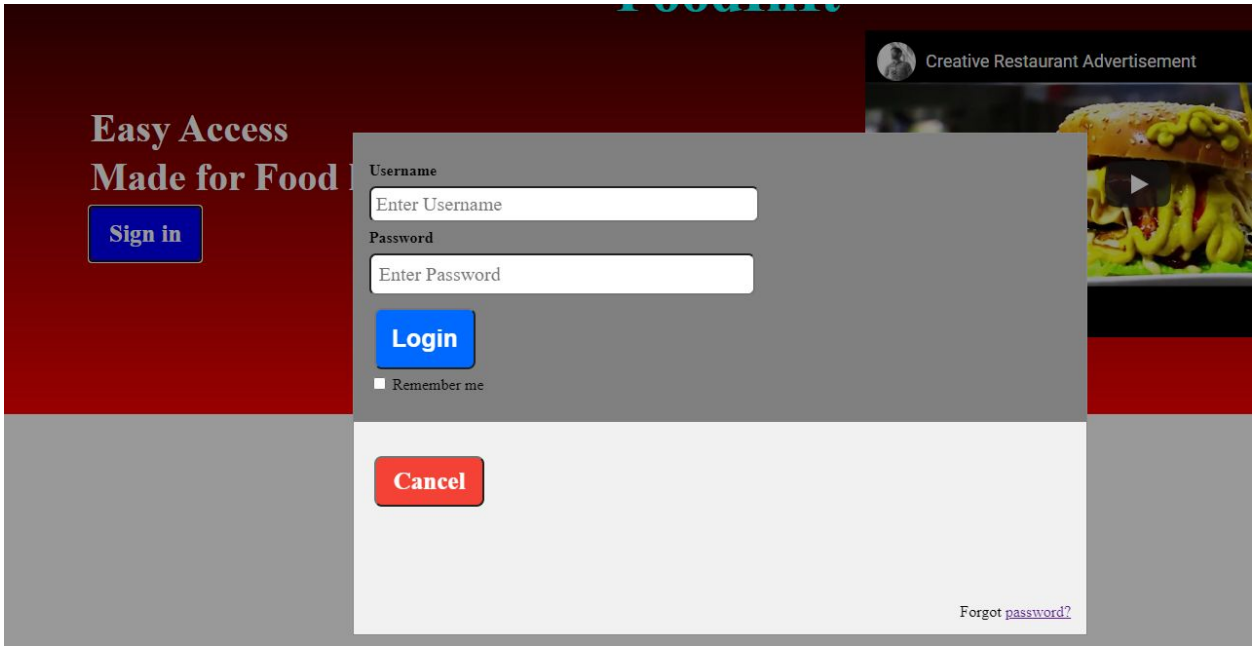
5) System Screen:

5.1 Homepage




Drinks
Mango Lacci
Thai Iced Tea
Pina Collata
Strawberry Smoothie
Pineapple Lemonade
Caramel Latte
Meals
Spring Rolls
Thai Dumplings
Pineapple Fried Rice
Pad See Ew
Basil Fried Rice
Sriacha Fried Rice
Dessert
Ultimate Gooey Brownies
Mint Oreo Cake
Chocolate And Strawberry Cake
Meyer Lemon Bars
Pomegranate Mousse Cake
Banana Pudding Parfaits
LEARN MORE
Nothing will annoy your audiences more than to see the same old ideas rehashed by so many brands! Switching your ad up with novel images can be refreshing in the eyes of your diners. They say a picture speaks a thousand words, and perhaps it's true! 55% of audiences actually recall content more when images are included.Today, the attention of consumers lasts for only about 8 seconds. So it will be smart for you to tell your restaurant's story by exploring with images. Visual storytelling involves showing, rather than telling. But there's just one small problem – not all of us are as creative as we wish we were. And not all of us are born master storytellers either.
FoodInIt
Company
Careers
Company
Websites
Overview
Design
Dashboard
Customers
Fine Dining
Fast Causal
Reservation
Contact
Phone number
Email
Fax-Number

5.2 Login


Easy Access
Made for Food
Sign in
Creative Restaurant Advertisement
Username
Enter Username
Password
Enter Password
Login
Remember me
Cancel
Forgot password?

5.3 Menu


HOME
MENU
CONTACT
ABOUT
MENU
APPETIZER
1. Shrimp Tempura $9.95
2. Fish Cake $8.95
3. Spring Roll $7.95
4. Thai Satay $8.95
DRINKS
1. Mango Lacci $8.95
2. Pina Colada $5.95
3. Strawberry Shake $7.95
4. Thai Iced Tea $8.95
ENTREE
1. Shrimp Fried Rice $19.95
2. Chow Mein Noodle $20.95
3. Pad Ma Kueryao $19.95
4. Salmon Fillet $35.95
5. Beef Stake $33.95
6. Smoke Salmon $39.95
7. Pad Gra Prow $23.95
8. Sichuan Fried Rice $19.95
DESSERT
1. Banana Pudding $8.95
2. Custard Lart $5.95
3. Key Lime Pie $7.95
4. Boston Cream Pie $8.95
5. Apple Pie $9.95
6. Kissel $4.95
7. Norman Tart $9.95
8. Tarte Tatin $8.95

6) Minutes of group meetings

1. Virtual Meetings and regular group chats were held
    a. In the group chat, we discussed what we had to get done and when all of us could get together to further discuss the project
        i. We would have an ongoing conversation of what we still had left to do
        ii. In the chat we would talk about the little tasks we set for ourselves during the last meeting
        iii. We usually used our facebook group chat to let each other know when we could start working on the google doc
    b. Our meetings were done through discord where we had a chat and a voice channel
        i. We would spend 10 minutes in the beginning by saying what we would like to get done during the time being
        ii. We would share our screen to show our progress. For example, how the system screen looked and what features we could add and where it could be added was shared.
        iii. Through the voice channel we would ask how to do certain parts of the diagram or if we should add something else to the diagrams (since we were doing the diagrams on other websites where we can easily draw them)
        iv. We would all be working together at the same time while being on discords voice channel to voice any concerns or issues we came across while we worked on the project
        v. These meetings usually were at least 3 hours and happened twice a week
        vi. At the end of our meetings we would set a day that we could continue working and some things we could get to do before the next meeting like:
            1. Like looking into how to do a diagram from class notes
            2. What websites would work best for each diagram
            3. How we would like our logo to look
2. Our only concern is how do we deal with disagreements if all of us have different thoughts on the same problem?

7) Github repo:
https://github.com/myumbla3/restaurant