

---

**Tasty Express**

---

**Tasty Express Delivery Services  
Software Requirements Specification  
For Online Restaurant Systems**

**Version 2.0**

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## Revision History

Date	Version	Description	Author
28/03/22	1.0	First Rough Draft of Tasty Express	Samuel Wolnerman Gaurav Kuwar Mohammed Fahad Usman Abbas, Andrei Skipor
26/04/22	2.0	Phase II: Design Report of Tasty Express	Samuel Wolnerman Gaurav Kuwar Mohammed Fahad Usman Abbas, Andrei Skipor

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## Table of Contents

1. Introduction	4
1.1 Class Collaboration diagram of entire system	4
2. All use cases	5
3. E-R diagram for Entire System	12
4. Detailed Design	13
5. System Screens	17
6. Memos of Group Meetings	17
7. Git repo link	17

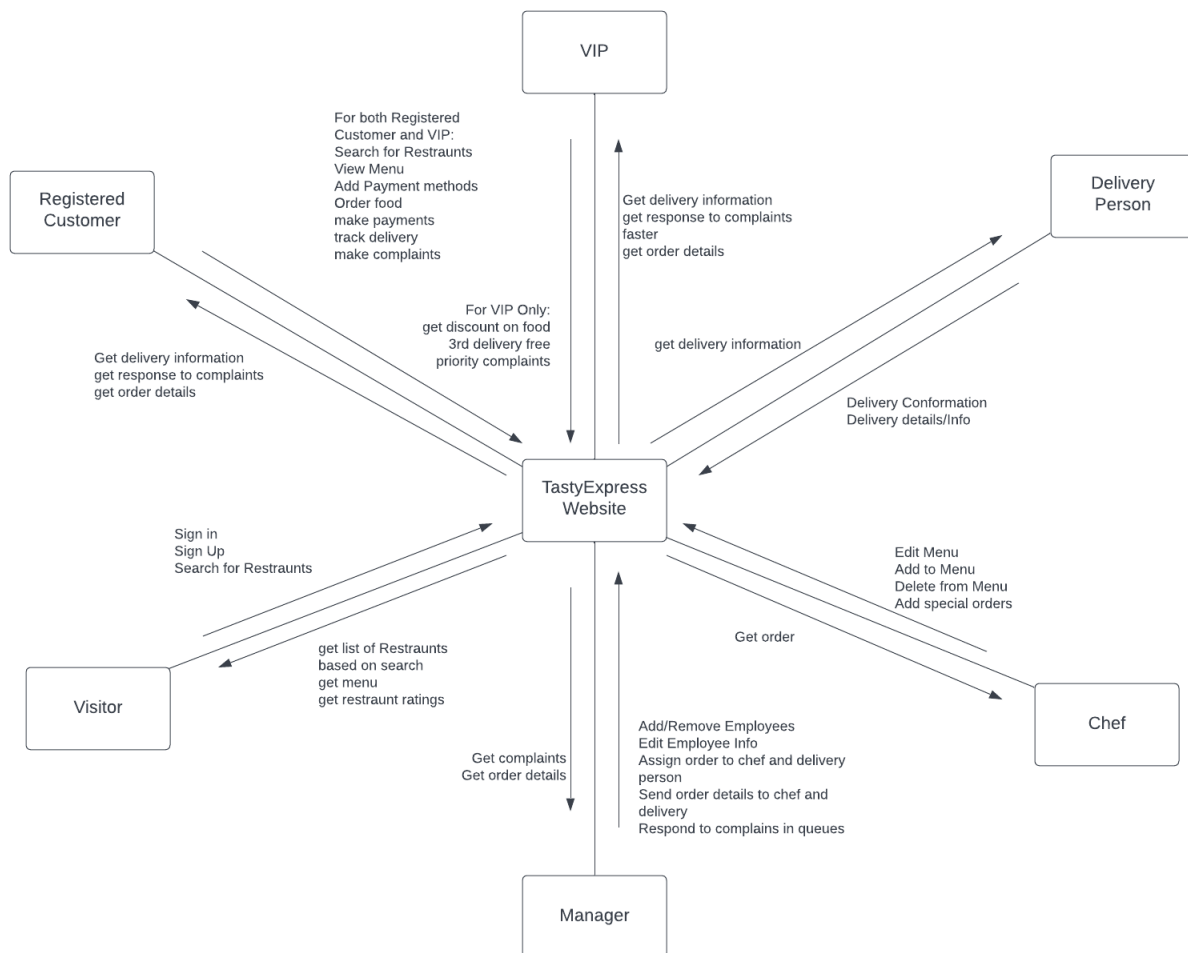
TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

# Software Requirements Specification

## 1. Introduction

This design report will give us the necessary overview of the online restaurant system. Specifically, the overview will show how the TastyExpress online restaurant system will look, operate, and function.

### 1.1. Class Collaboration Diagram of Entire System



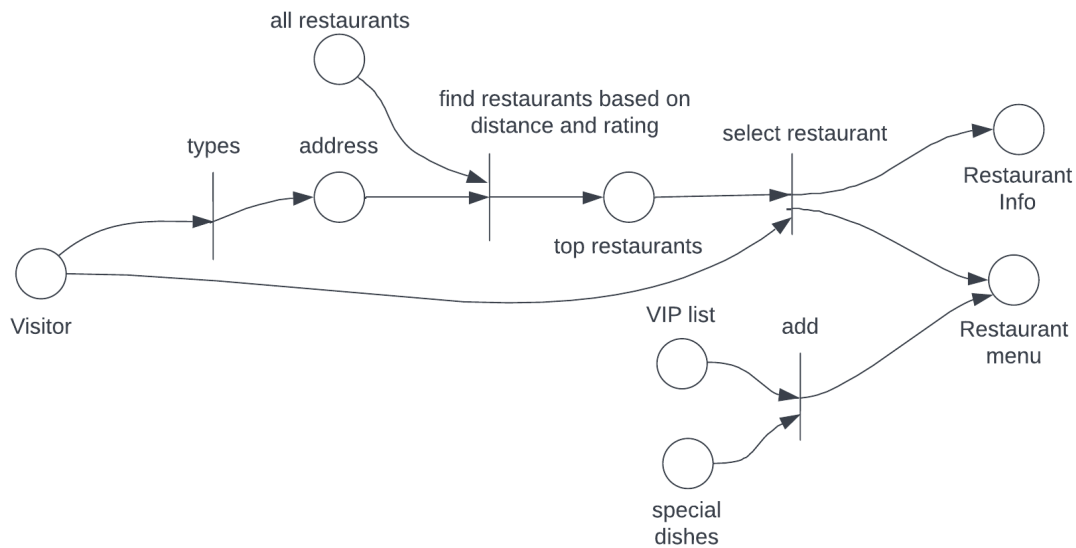
TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 2. All use cases

This section of the report will provide a more detailed look at the use case diagrams. The section will feature a collaboration class diagram or Petri-net for each use case model that would essentially give us a basic understanding of the system's functionality.

### 2.1. Search for Restaurants (Petri-net)

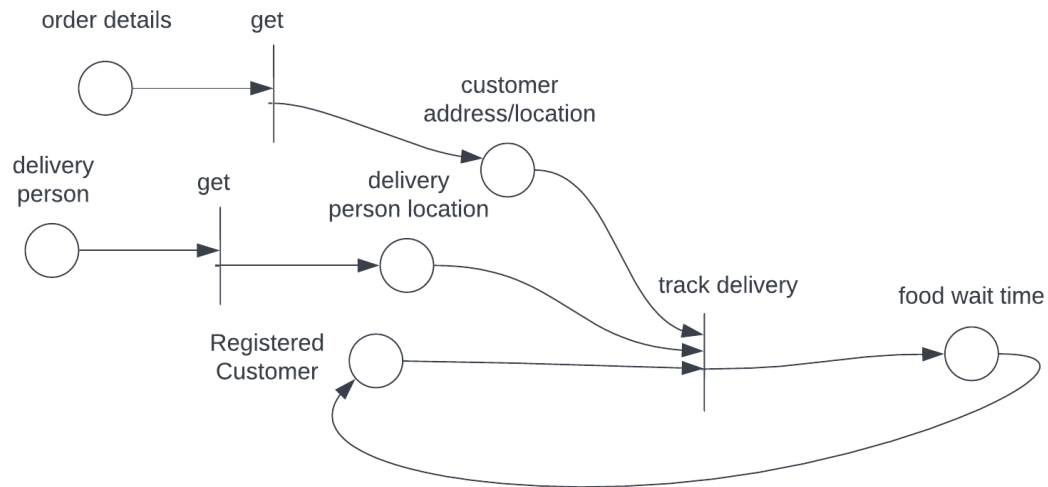
Visitor is someone without an account, everything a visitor can do is someone with an account. Not only that but this section will show what the visitor can do as well as access the online restaurant system.





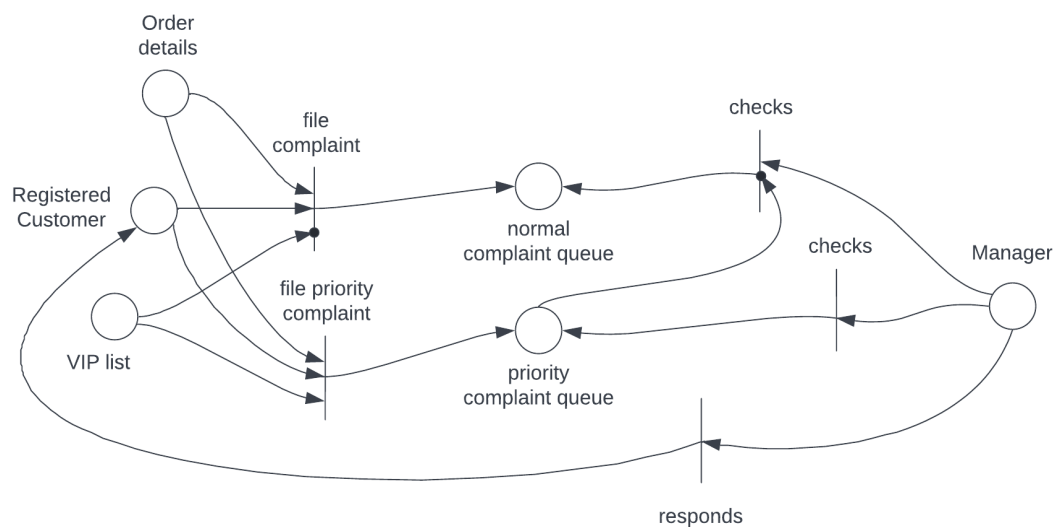
TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 2.3. Track Order (Petri-net)



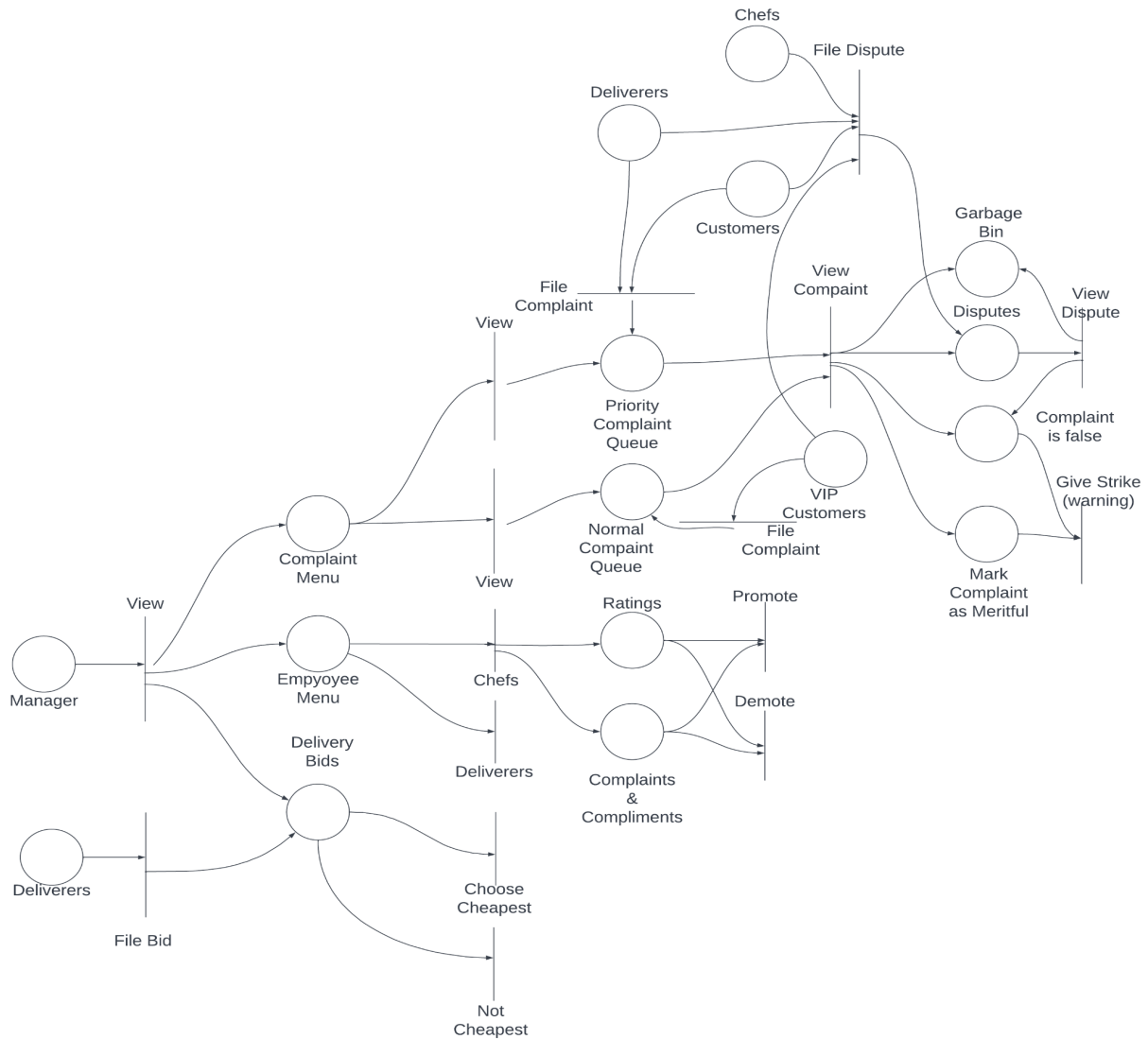
## 2.4. File complaints

This section shows the process in which complaints would be handled.



TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

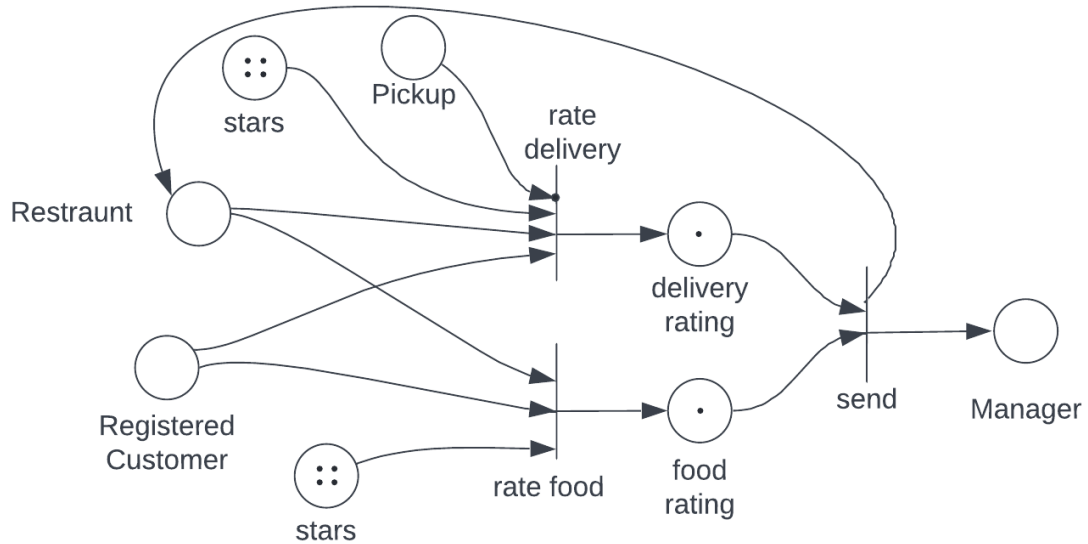
## 2.5 Manager (Petri-net)



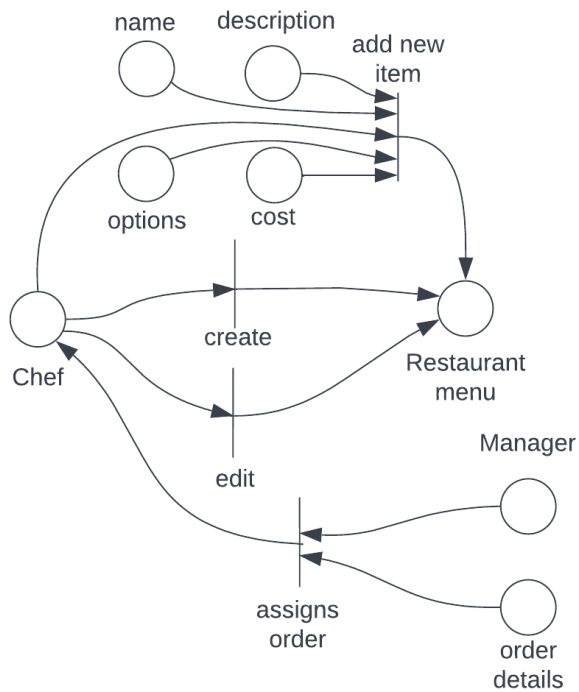


TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 2.5. Rate food and delivery (Petri-net)



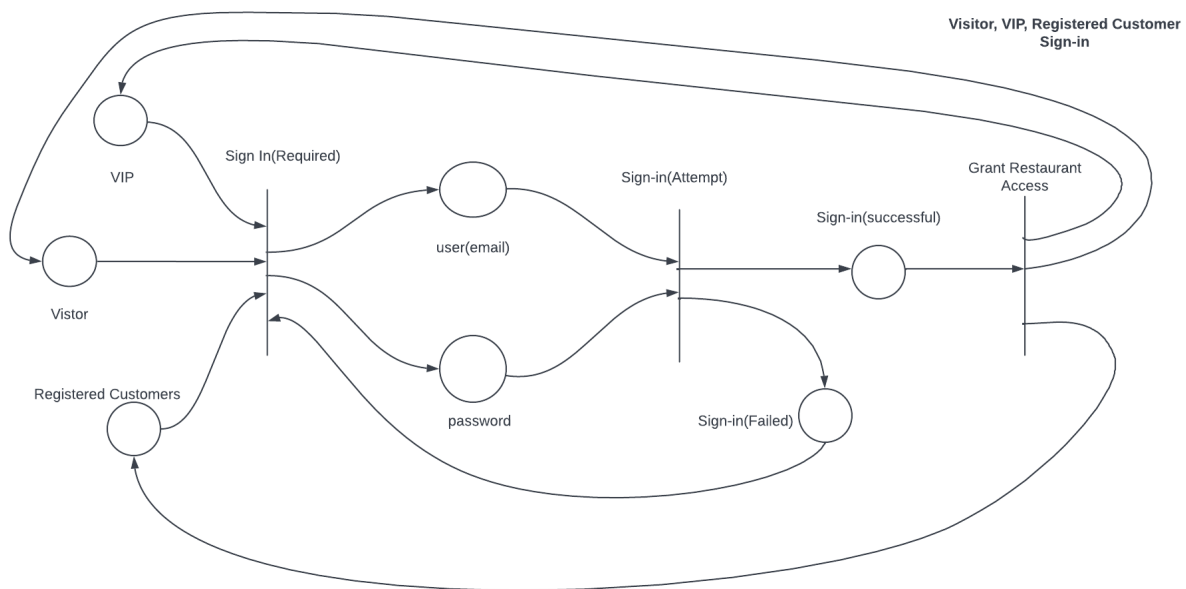
## 2.6. Chef use case (Petri-net)



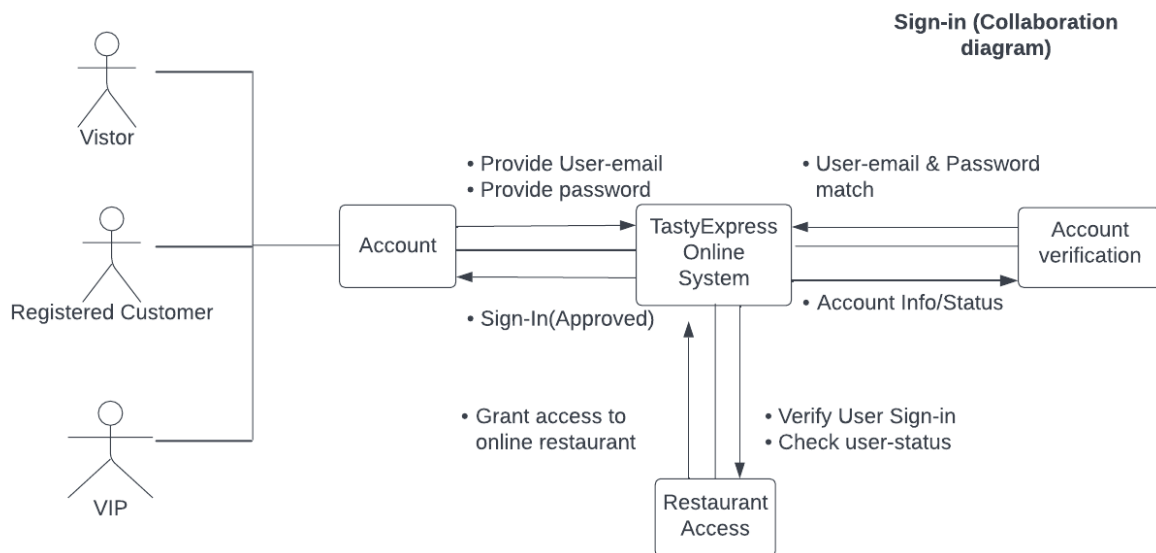
TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 2.7. Sign-In(Petri-net)

This part of the system will ask the visitor, registered customer, and VIP to sign in. The diagram shows the sign-in process for the users and the series of steps that it would take to successfully be granted access to the online system.



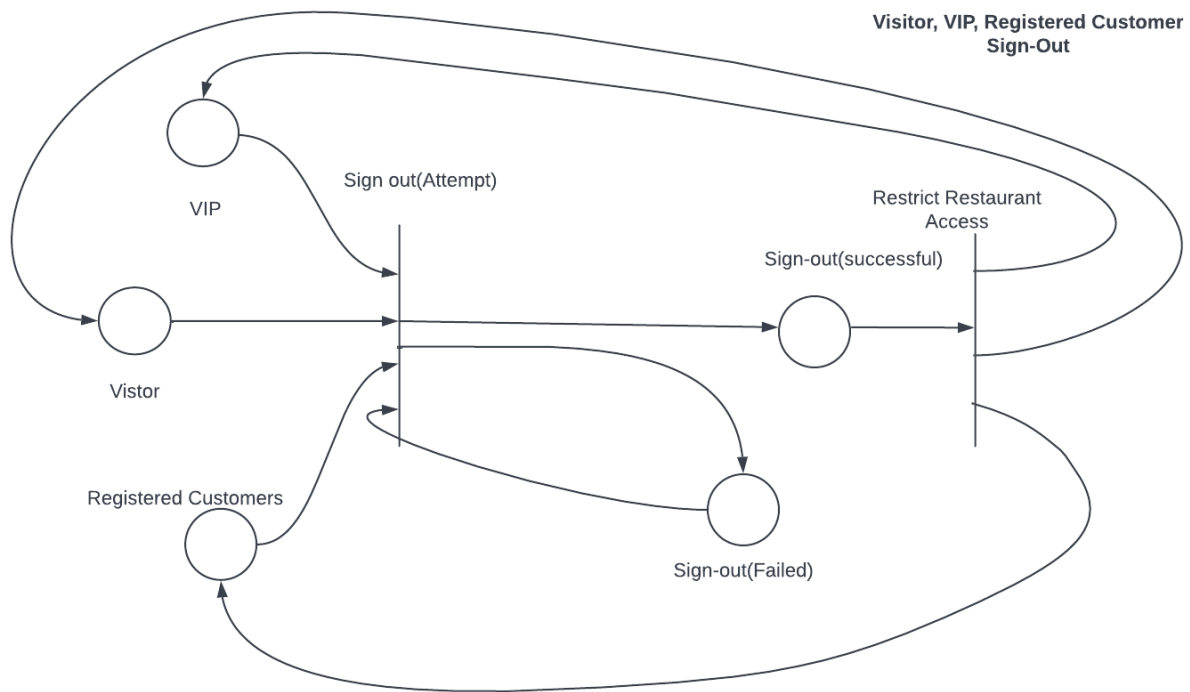
## 2.7. Sign-In(Collaboration diagram)



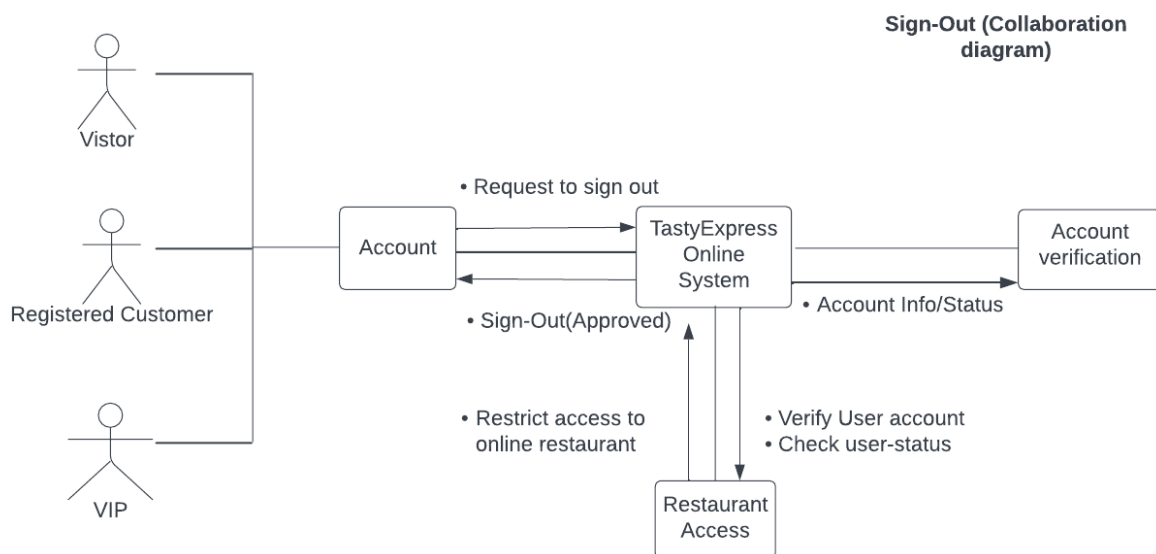
TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 2.8. Sign-Out(Petri-net)

The diagram shows the process the user will take in order to successfully sign out of the online system.

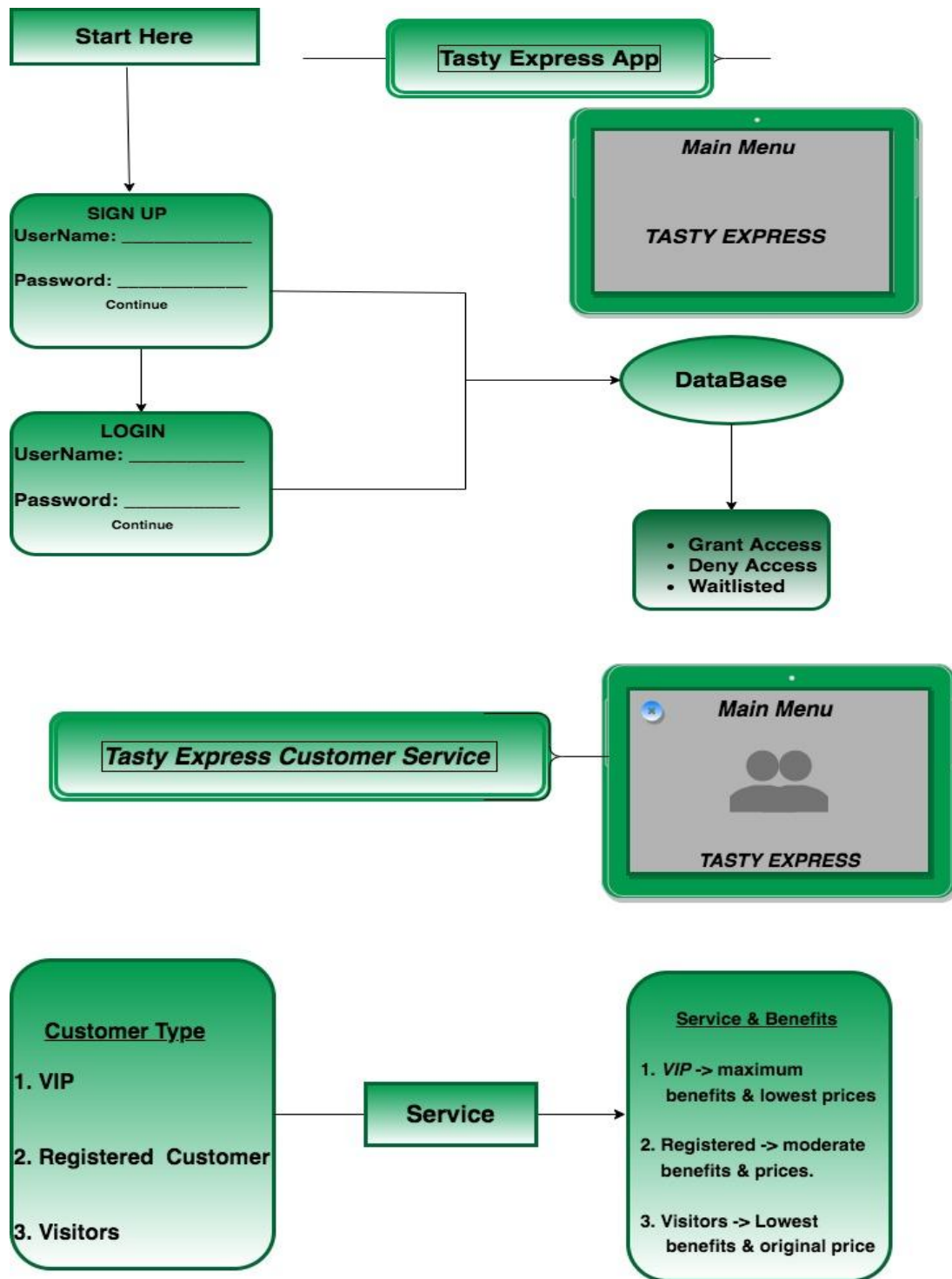


## 2.7. Sign-Out(Collaboration diagram)



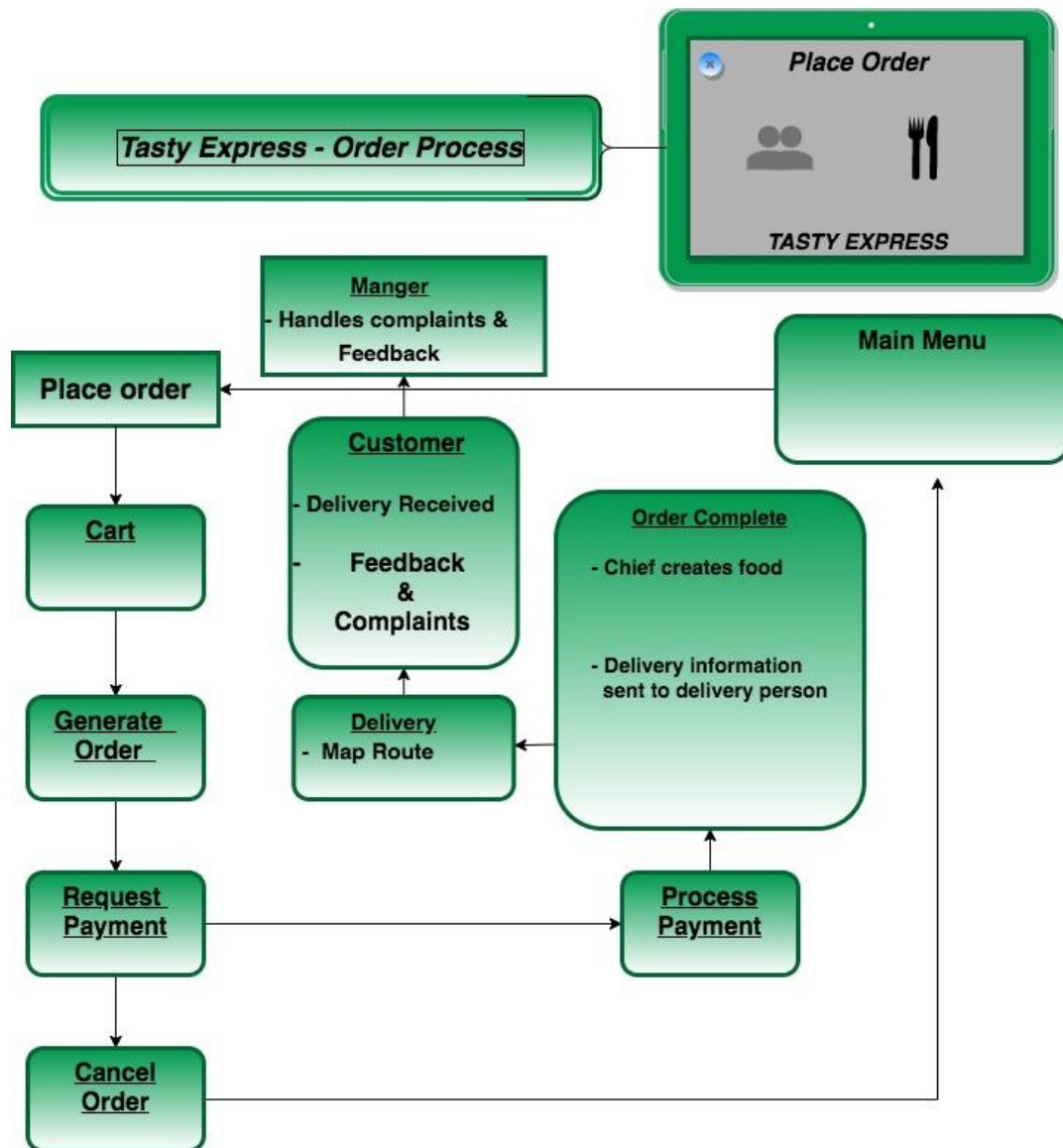
TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 2.7. State-Diagram



TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

I

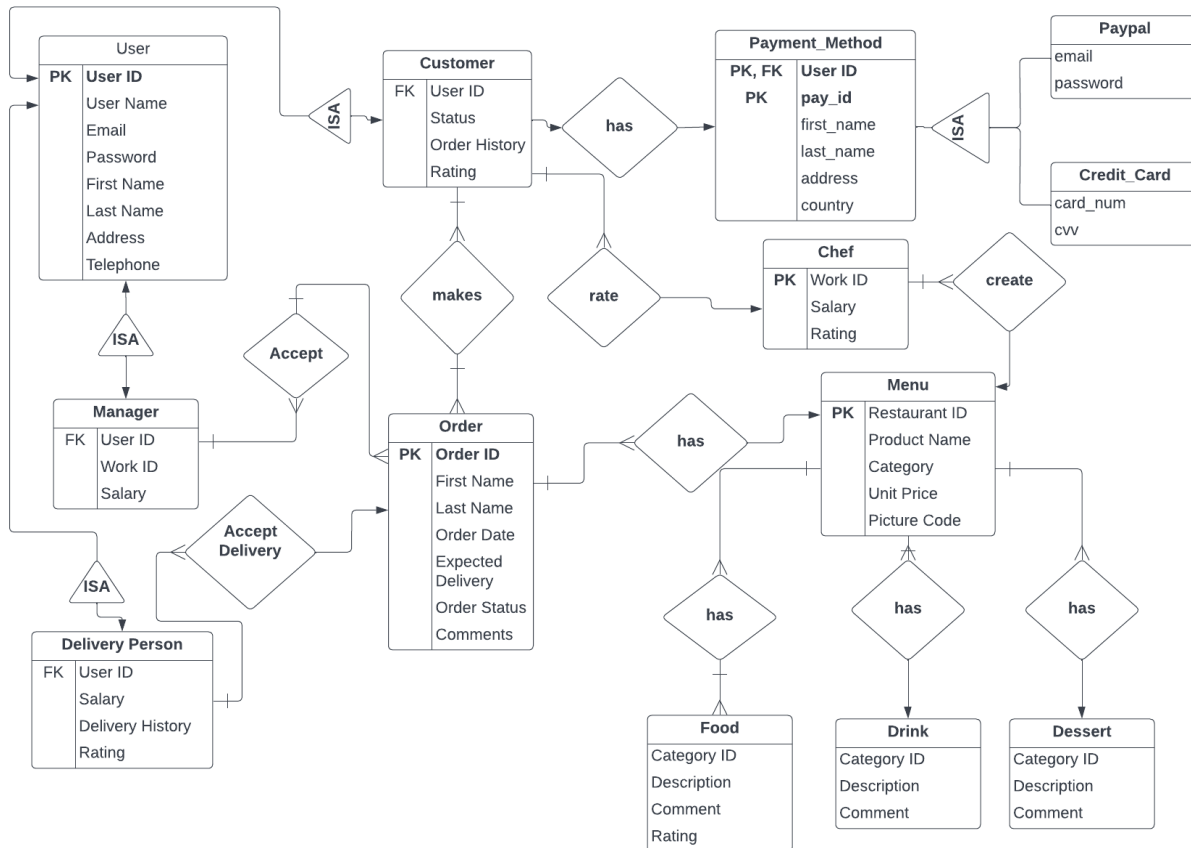


## 2.7. State-Diagram(Cont.)

The diagram shows a simple overview of the running process of the application. This will essentially give a quick run-down of the flow of events a user will take while using the online restaurant system.

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

### 3. E-R diagram of the entire system



### 4. Detailed Design

#### 4.1. SignOut

The user will have the opportunity to sign out of the online restaurant. This will be done by checking whether user.login is set to -1, or 1. If -1 then that means that the user has requested to sign out of the system. This will then clear the user's session within the system. If the user decides to cancel the sign-out request, then the user.login will be set to 1 which will emphasize that the user has canceled the request to sign out.

```
def Sign_Out(user):
```

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

```

if user.login == -1:
    print("User has requested to sign-out")
    Clear_session(user.session) # Clear the user session to allow sign out.
else if user.login == 1:
    print("User has canceled sign-out request")
else:
    print("Error processing request")

```

#### 4.2. RatingUsers

Here we will be rating users based on their performance on the job. Ratings will scale from -3 to 3, with -3 being that the user has been fired due to receiving the worst possible rating and 3 being that the user has been promoted with an increase in salary. Any ratings between -3 and 3 will output a progress report to the user.

```

def Rating(user_rating):
    user.Rating_status += user_rating
    if (User.Rating_status = 2)
        print("Promotion: User has a rating of 2"):
    else if (User.Rating_status = 3)
        print("Promotion: User has received a rating of 3, User Salary Increased"):
    else if (User.Rating_status = -2)
        print("Caution: User has a rating of -2, please make improvements"):
    else if (User.Rating_status = -3)
        print("Terminate: User has received a rating of -3, User has been fired"):

```

#### 4.3. AddFood

Desired food would be added to the menu

```

def add_food(food)
    add food to menu

```

#### 4.4. DeleteFood

Selected food would be deleted from the menu

```

def delete_food(food)
    Delete food from menu

```

#### 4.5. Edit Menu

Selected food would have the opportunity to be edited if needed.

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

```
def Edit_menu():
    print("What do you want to edit?")
    Selected = input(Enter food/drink/item) etc here: )
    Find Selected in Menu:
        Edit Selected food
```

#### 4.6. Chief(User)

The chief will have the privilege of committing various actions throughout the online system such as editing the menu, deleting food and adding food.

```
def Cheif_User():
    Print("What does Chief want to do")
    Chief_response = input("Options(Edit/add/delete)
    If Cheif_response = "add":
        add_food(Food)
    else if Chief_response = "delete":
        delete_food(Food)
    else if Chief_response = "edit":
        Edit_Menu()
```

#### 4.7 Manager functions:

These are the classified functions that will be represented in the class Manager. The functions defined in the Manager class are what the manager privileges within the online restaurant.

```
priority_queue = []
normal_queue = []

def send_reply(response, user_id):
    send email to user_id with response

def get_complain():
    if len(priority_queue) > 0:
        complain = priority_queue.pop(0)
    else:
        complain = normal_queue.pop(0)
    return complain
```



TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

```
def add_employee(user_id, salary, role):
    if role == "Delivery Person":
        insert into "Delivery Person" values (user_id, salary, role);
    elif role == "Chef":
        insert into "Chef" values (user_id, salary, role);
```

```
def edit_employee_info(user_id, role, col_to_change):
    if role == "Delivery Person":
        for col in col_to_change:
            update in "Delivery Person" value col = col_to_change
    elif role == "Chef":
        for col in col_to_change:
            update in "Chef" value col = col_to_change
```

#### 4.8 Payment Functions

Defined are the payment functions that are essential for the payment processing in the online system.

```
def add_payment_method():
    paymentType = request["paymentType"]
    uid = session["user_id"]
    first = request["first"]
    last = request["last"]
    address = request["address"]
    country = request["country"]

    if paymentType == "paypal":
        email = request["email"]
        pass = request["password"]

        insert into "Paypal" values
        (user_id, first_name, last_name, address, country, email, password);
    else: # card
        card_num = request["card_num"]
        cvv = request["cvv"]

        insert into "Credit_Card" values
        (user_id, first_name, last_name, address, country, card_num, cvv);
```

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

```
def remove_payment_method():
    pay_id = request["pay_id"]
    paymentType = request["paymentType"]
```

```
    if paymentType == "paypal":
        DELETE FROM Paypal
        WHERE pay_id=pay_id;
    else:
        DELETE FROM Credit_Card
        WHERE pay_id=pay_id;
```

#### 4.9 Order Food functions:

Functions that would be needed throughout the ordering process would be defined below.

```
def addToCart():
    # Everytime item is added to cart
    item = request["item_selected"]
    quantity = request["quantity"]

    session["cart"] = [item, quantity]

def order():
    total = 0
    for item, quantity in session["cart"]:
        cost = get item cost from database with item
        total += cost * quantity
```

```
    delivery_cost += calculate taxes
```

```
    if customer is VIP and count = 3:
        delivery_cost = 0
        total = apply discount to total
```

```
    total += calculate taxes
```

```
def checkout():
    pay_id = request["pay_id"]
    uid = session["user_id"]
    Order_details = request["Order_details"]
```

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

```
paymentInfo = "SELECT * FROM Payment_Method
              WHERE pay_id=pay_id and user_id=uid;"
```

```
send to owner payment using paymentInfo
INSERT INTO orders VALUES Order_details
```

#### 4.9 Display items on the menu for a restaurant

```
def display_menu():
    SELECT * FROM Menu
    WHERE restaurant = session["restaurant"];
```

## 5. System Screens

[Click Here](#) to see the System Screens (complete Front End) and GUI Walkthrough.

We build functionality for sign up and login for prototypes. [Click here](#) to see the demo.

## 6. Group Meeting Memos

Date : 04/16/2022

Setup github, phase II reports. Detailed plan on functionality and diagrams for initial Phase II report. Decided to switch to a web based model.

Date : 04/25/2022

Discussed diagrams of the overall system and then divided work use case diagrams.

Date : 04/26/2022

Discussed how we would do the diagrams, how many petri-net diagrams we needed and how many class collaboration, and then decided who works on what.

Date : 04/27/2022

Discussed ER diagram and possible design changes for better database design.

Date: 04/28/2022

Decided what pseudocode functions to work on for the system and split the work.

TastyExpress	Version: 2.0
Phase II: Design Report	Date: 4/10/2022
Second Draft	

## 7. Git repo Address

<https://github.com/mohammedfahadnyc/Tasty-Express/tree/fahad-04-27-22>