



**FASTLaneFood**  
**Design Report**  
**For Online Restaurant System**

**Version**  
**2.0**

FASTLaneFood	Version: 2.0
Phase II: Design Report	Date: Nov.21.2019
Design Report Final	

### Revision History

Date	Version	Description	Author
Oct.21.19	1.0	First rough draft of FastLaneFood	Hasibul Islam Eftekher Husain Shahan Rahman Jung Tae Lee
Nov.21.2019	2.0	Phase II: Design Report of FastLaneFood	Hasibul Islam Eftekher Husain Shahan Rahman Jung Tae Lee

FASTLaneFood	Version: 2.0
Phase II: Design Report	Date: Nov.21.2019
Design Report Final	

## Table of Contents

1.	Introduction	3
1.1	Collaboration Class Diagram	4
2.	All use cases	4
2.1	Scenarios for each case	4
2.2	Class Diagrams for Cases	4
2.3	State Diagram/Petri-net for Cases	17
3.	E-R diagrams	21
4.	Detailed Design	22
4.1	Methods used and pseudo-code	22
5.	Systems Screens	29
5.1	GUI Screens	29
6.	Meeting Dates	36
7.	Github Repo	36



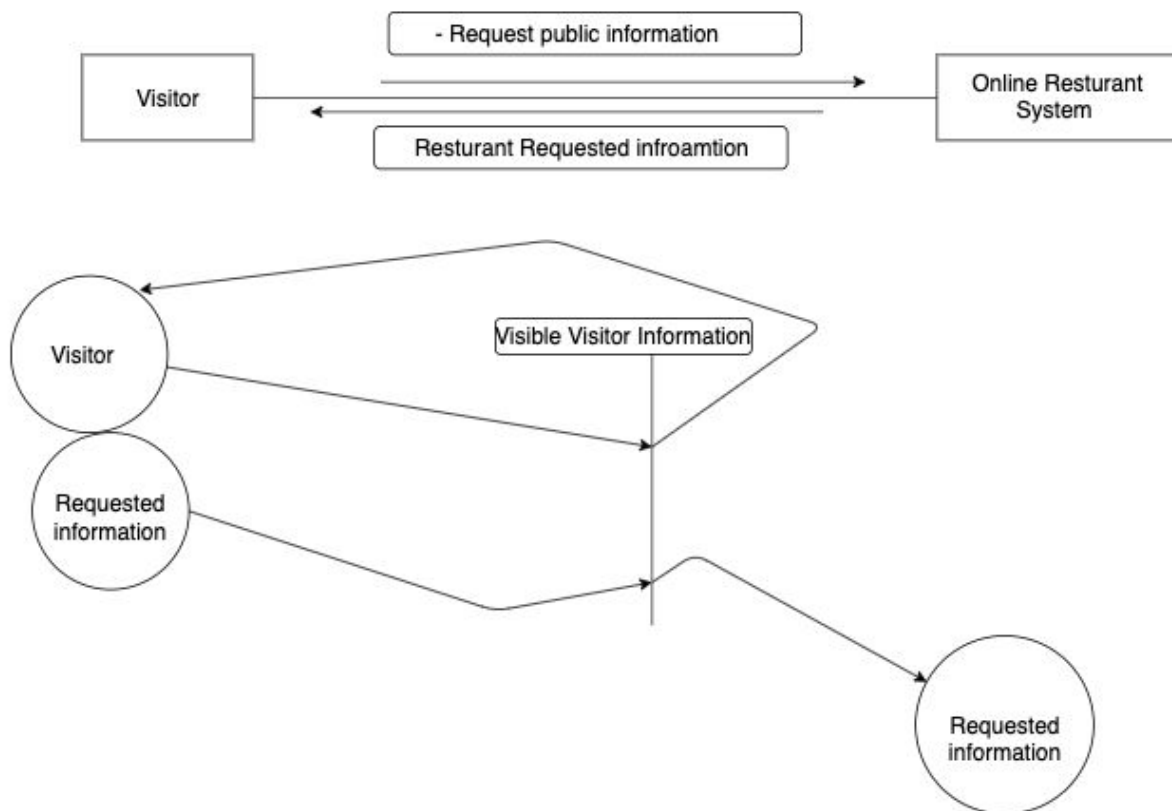
## 2. All Use Cases

In this section, we provide a more detailed overview of all the use cases. There are collaboration class diagrams and a State diagram provided for each use case, which will give a better understanding of how the system works.

### 2.1/ 2.2 Scenarios for each case / Class Diagrams for Cases

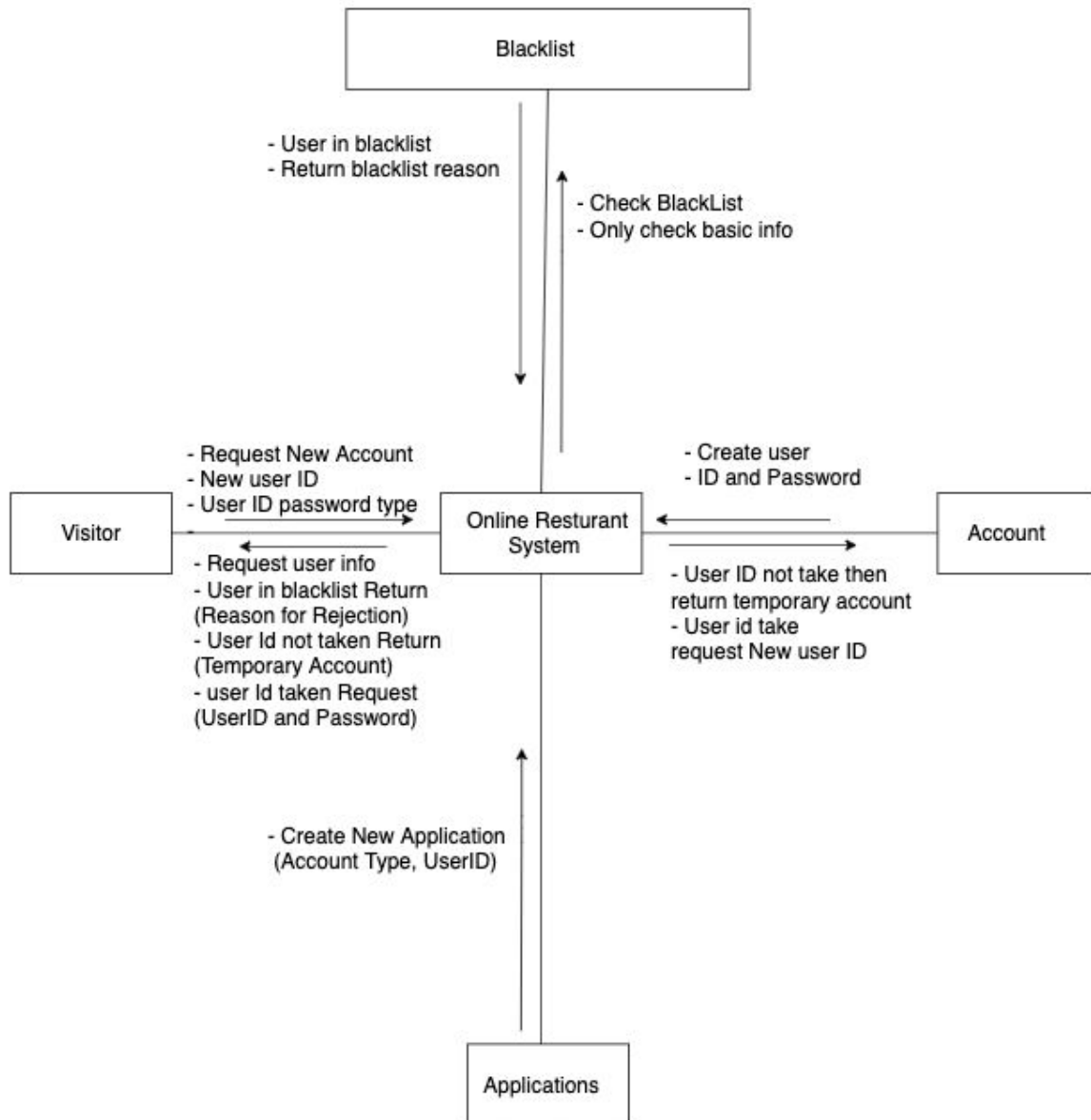
#### 2.1.1 Visitor Information

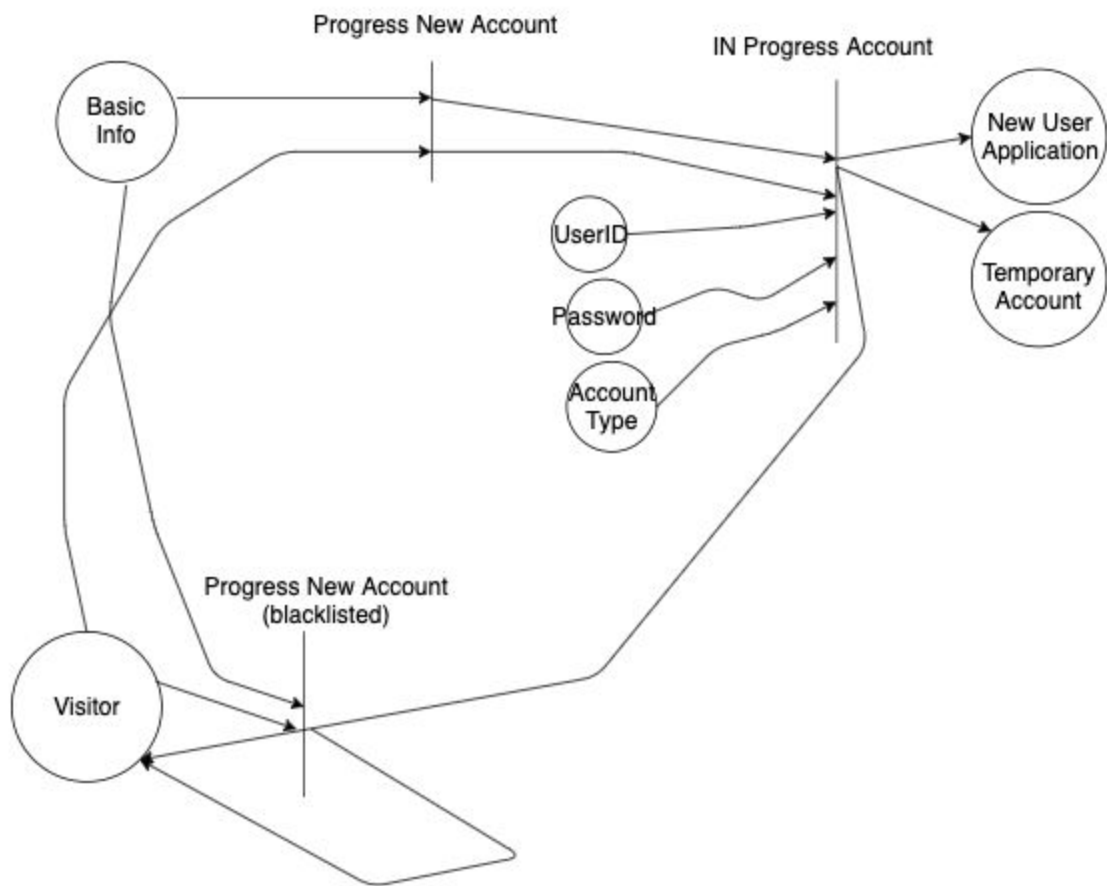
This is the information that the visitor can see and access but the system will always ask the visitor to register. This is for the Online Restaurant System, to request information for a registered customer.



### 2.1.2 Registered Customer

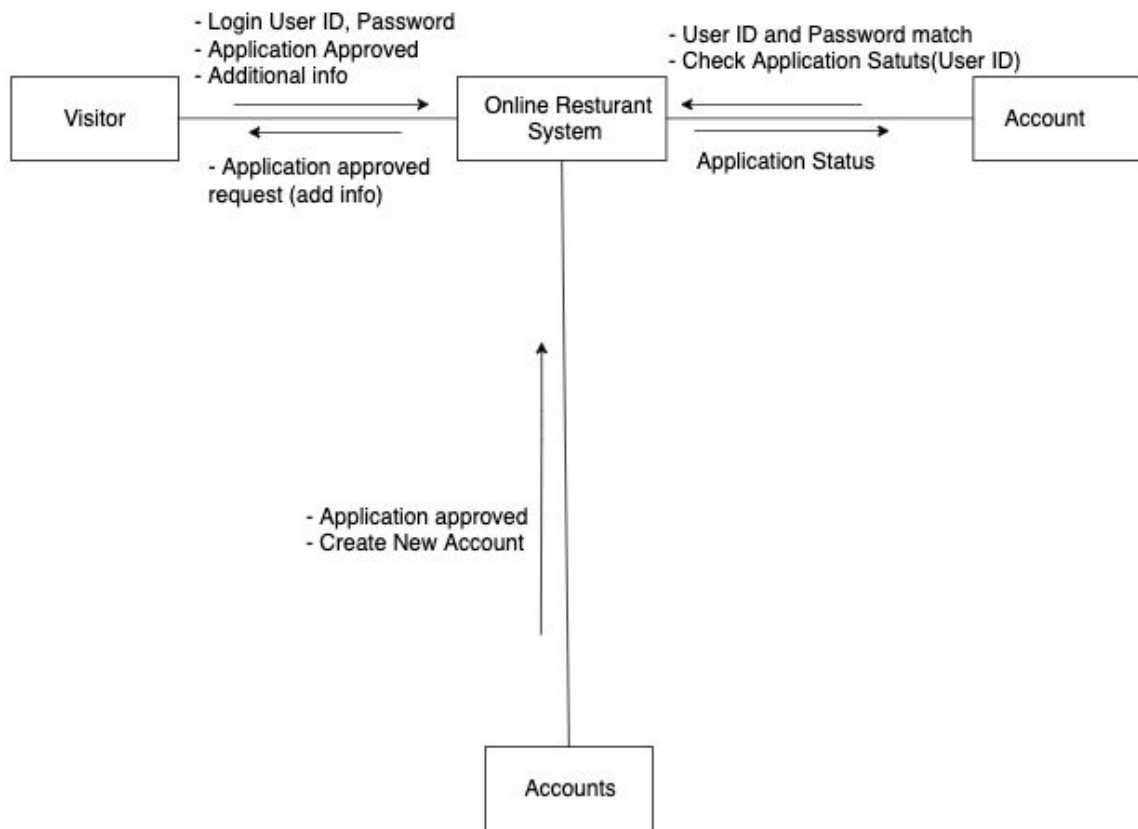
This is the part of the system where the system will ask the visitor to sign up and apply to the client. The Online Restaurant System determines if the user is on the black list before allowing them to be verified. A userID and password will be required to have the customer sign up successfully.



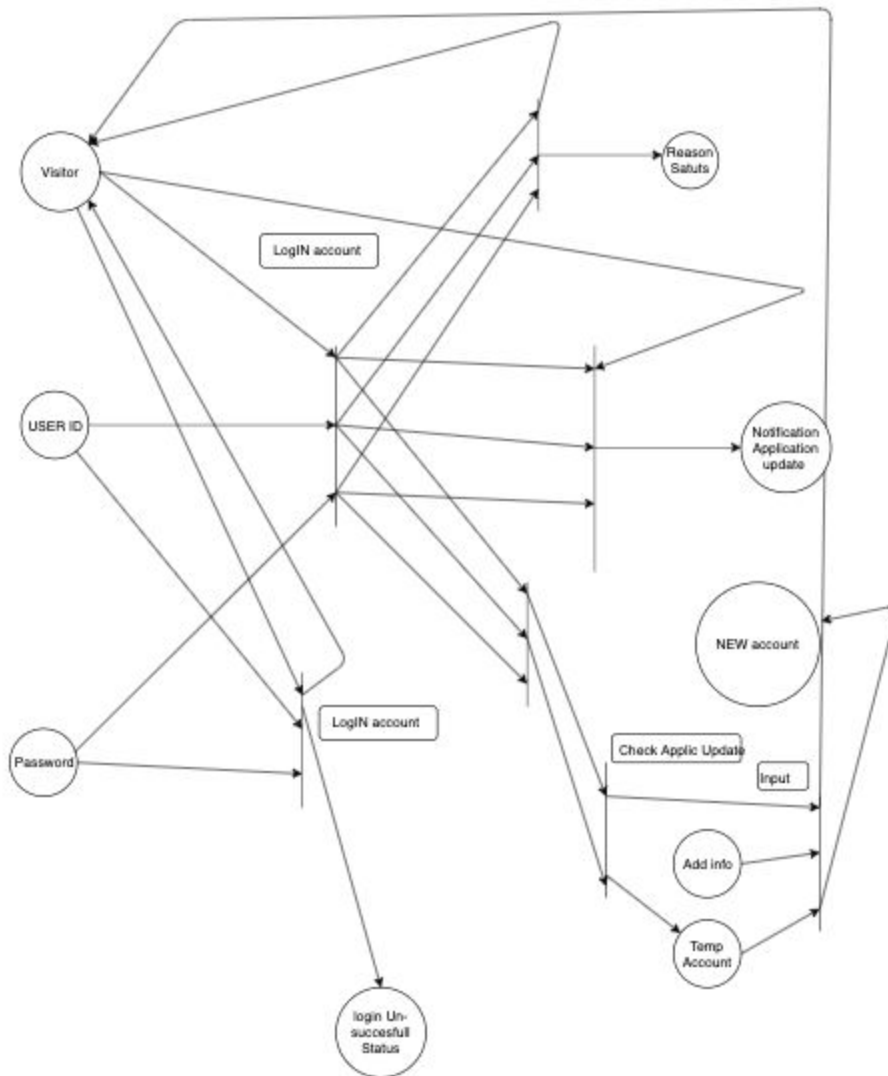


### 2.1.3 Customer Account Status

This is the part of the system where the system will allow the user to be approved or denied by the manager. Based on what the manager decides to do the user will have the effects applied on them such as visitor prices or registered prices.



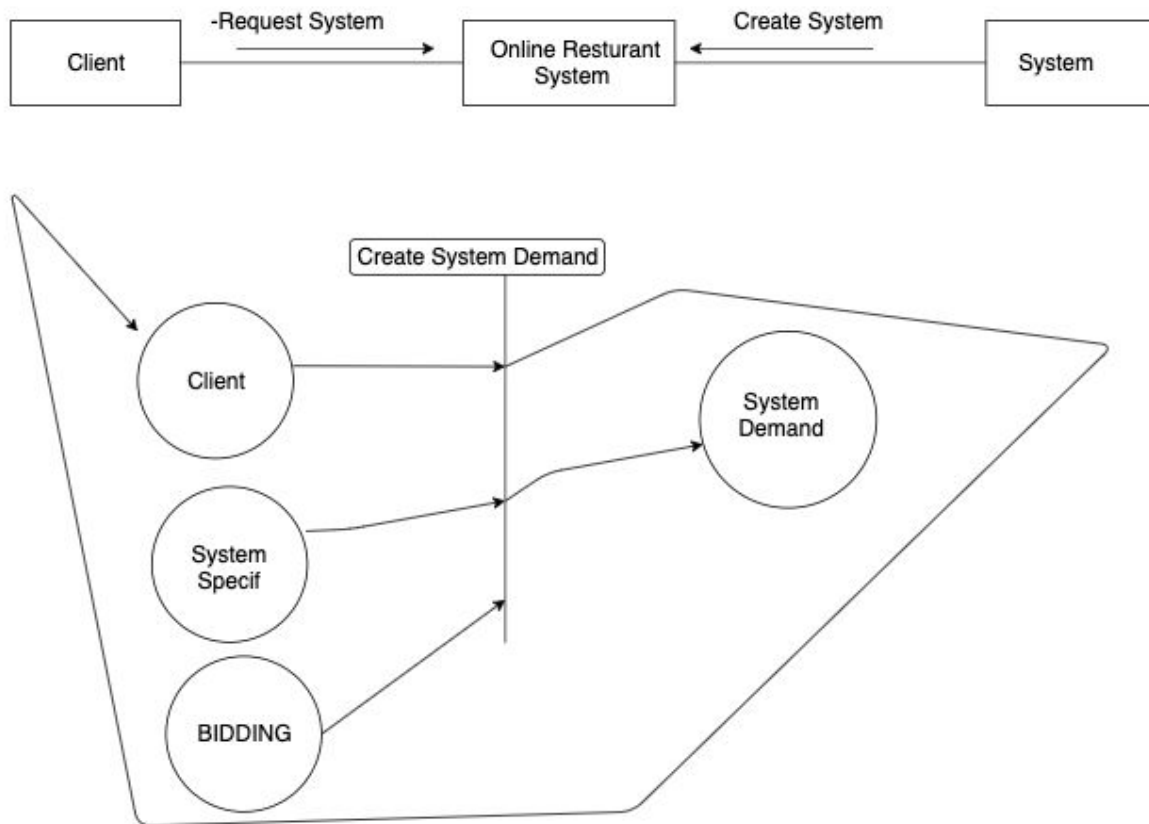




#### 2.1.4 Bidding System and Demand

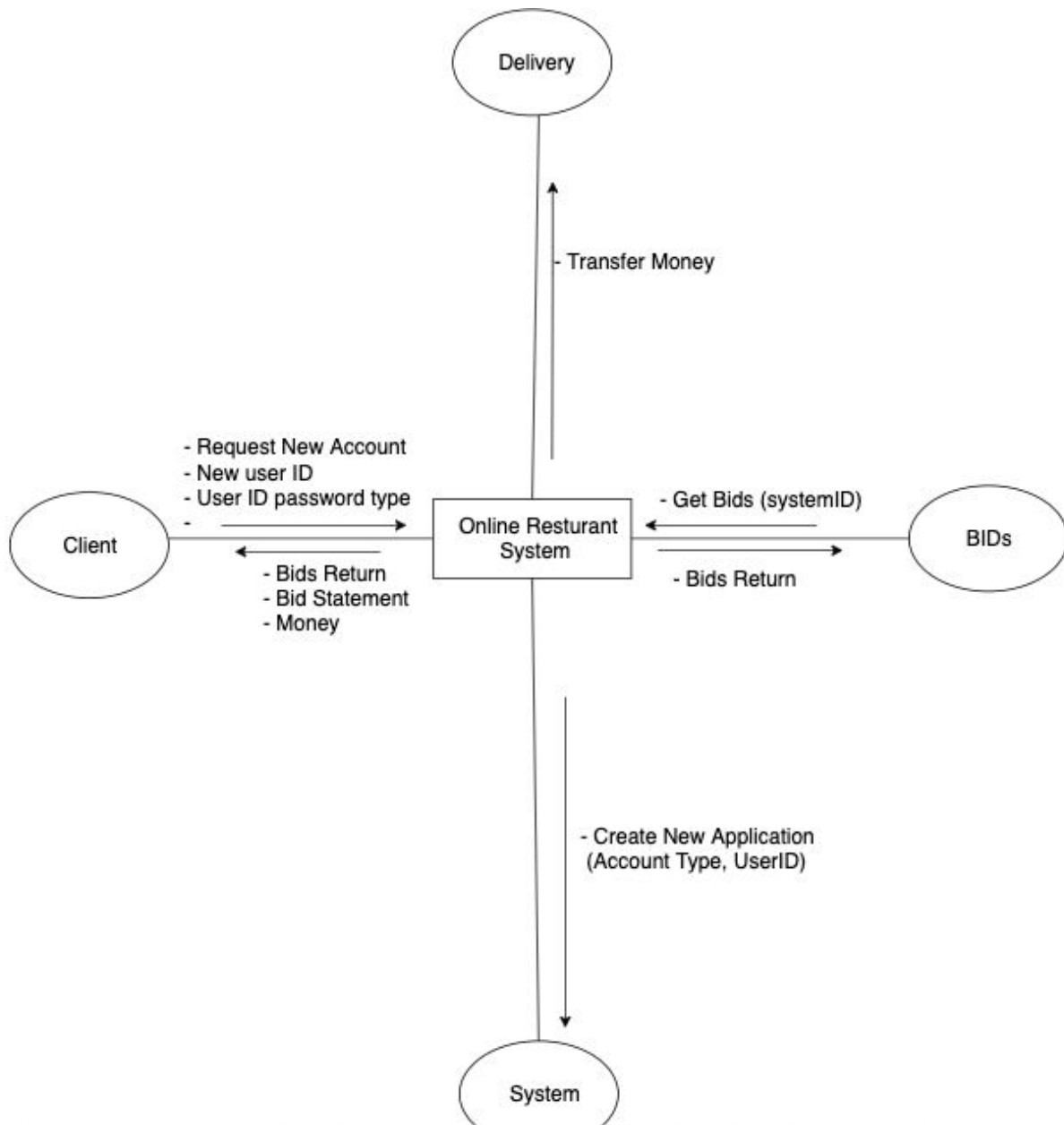
This is the part of the system where the system will find delivery boys and have them bid for the trip. The Online Restaurant System checks if the bidding timeline is the best for the

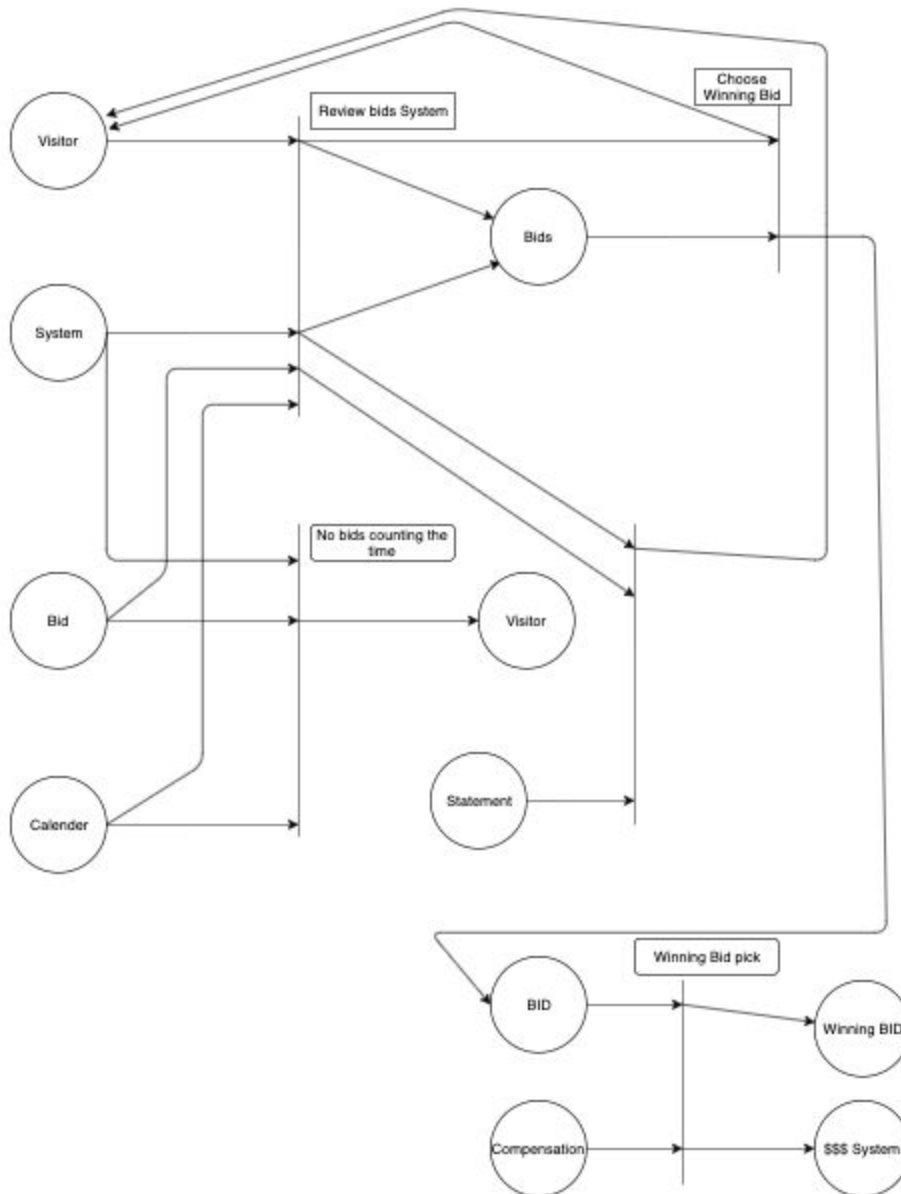
restaurant, such as time and route.



### 2.1.5 Best Choice By System

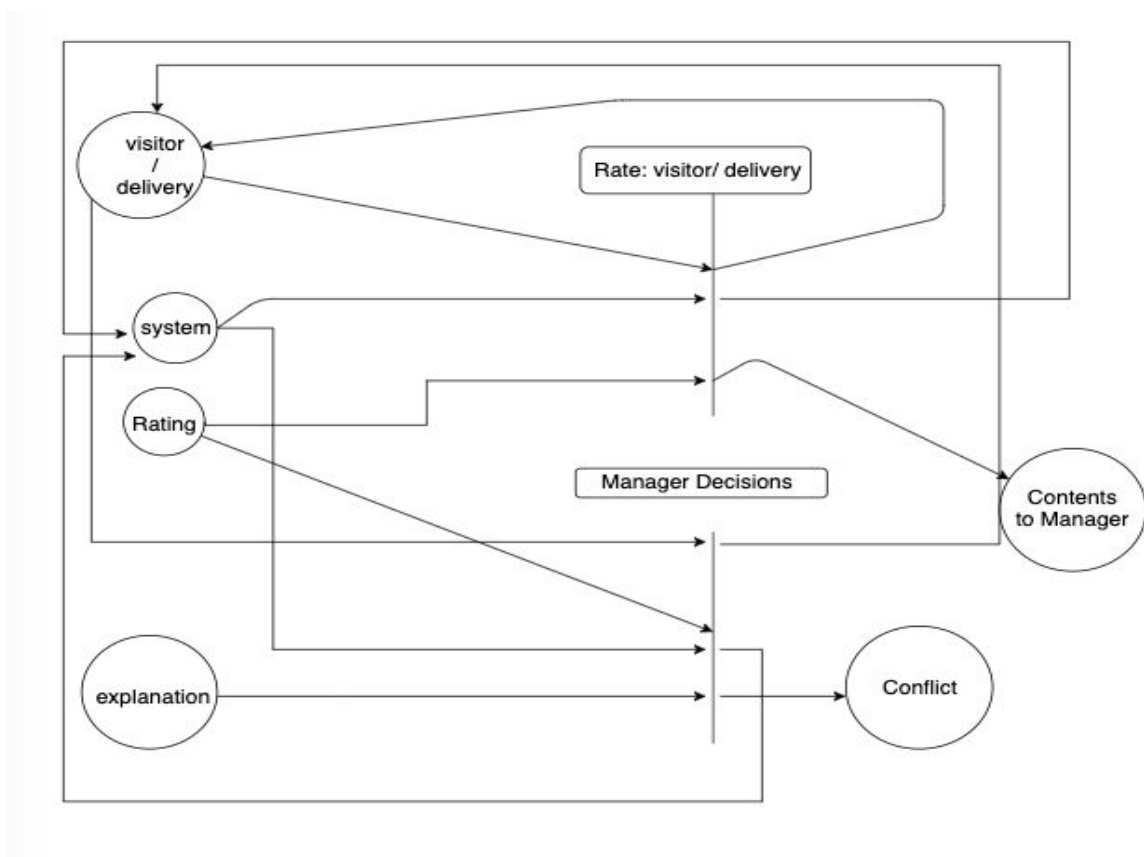
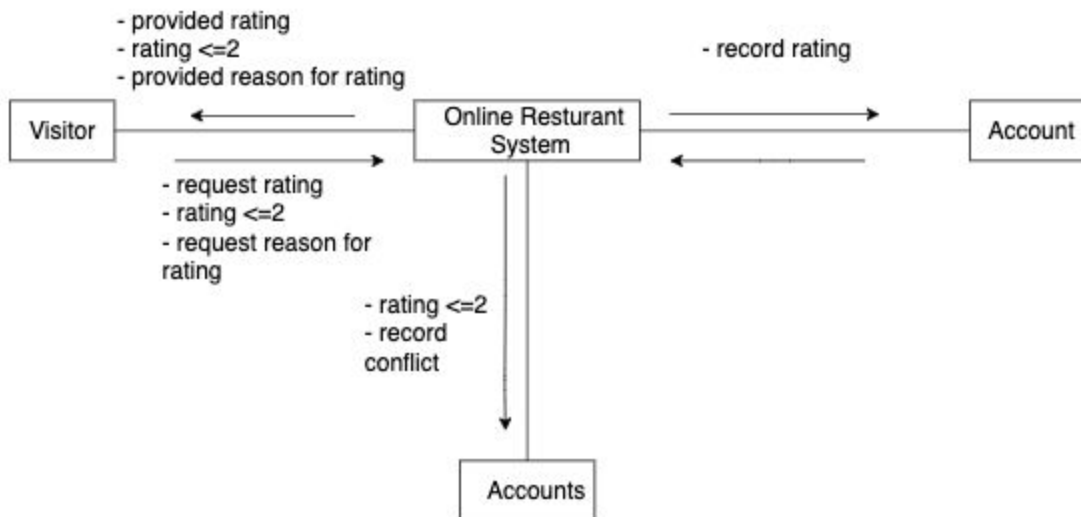
Since the system decides what are the best delivery boys to send the food, the lowest price and best route is picked for the delivery to be made. This means the system will have different demand for each order.





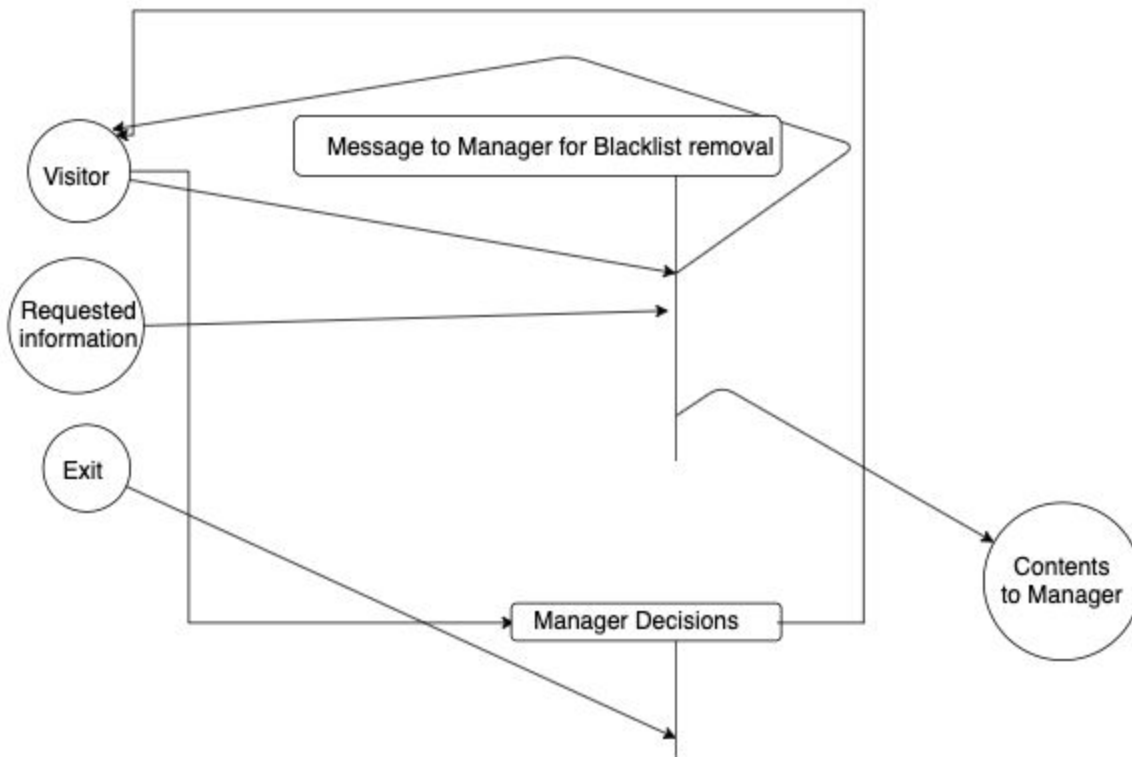
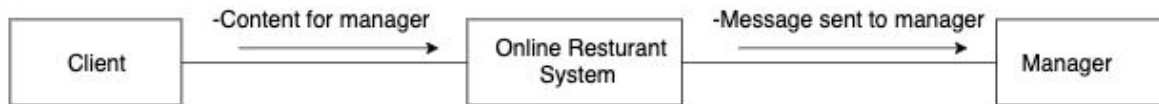
## 2.1.6 Rating System

The system will have a rating for every user there for the rate clients are based on other users. Therefore, the system needs to have a filter that sees conflicts and what user is sending this message and the rating value.



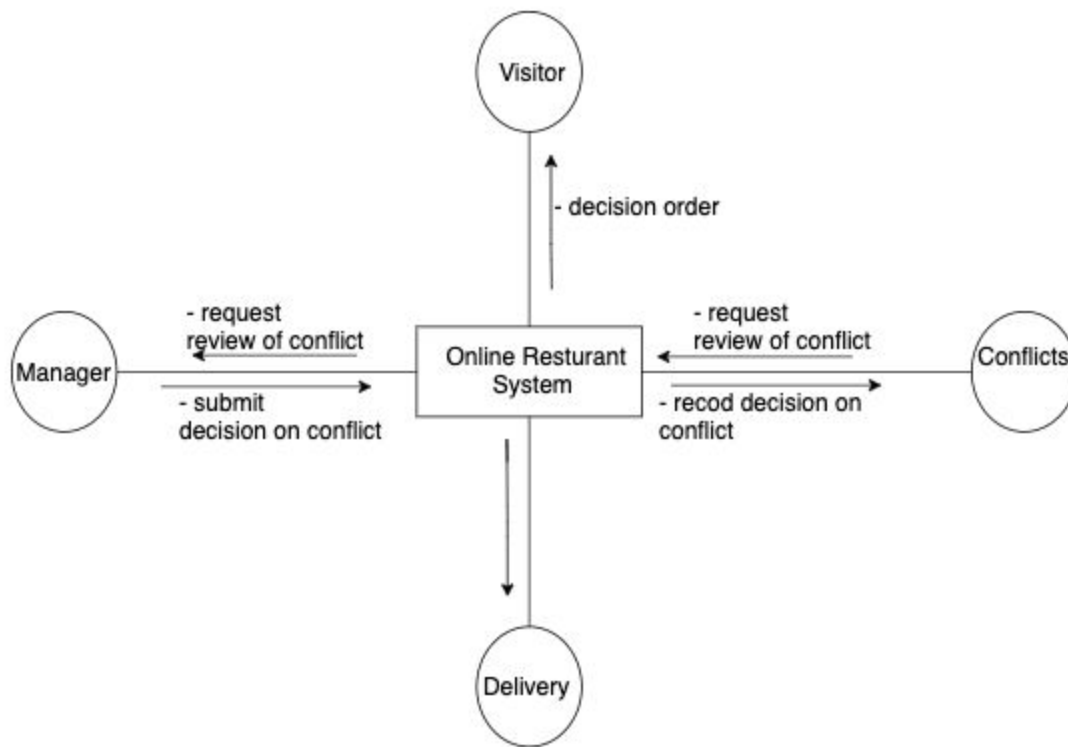
### 2.1.7 Contact Manager for Poor Rating

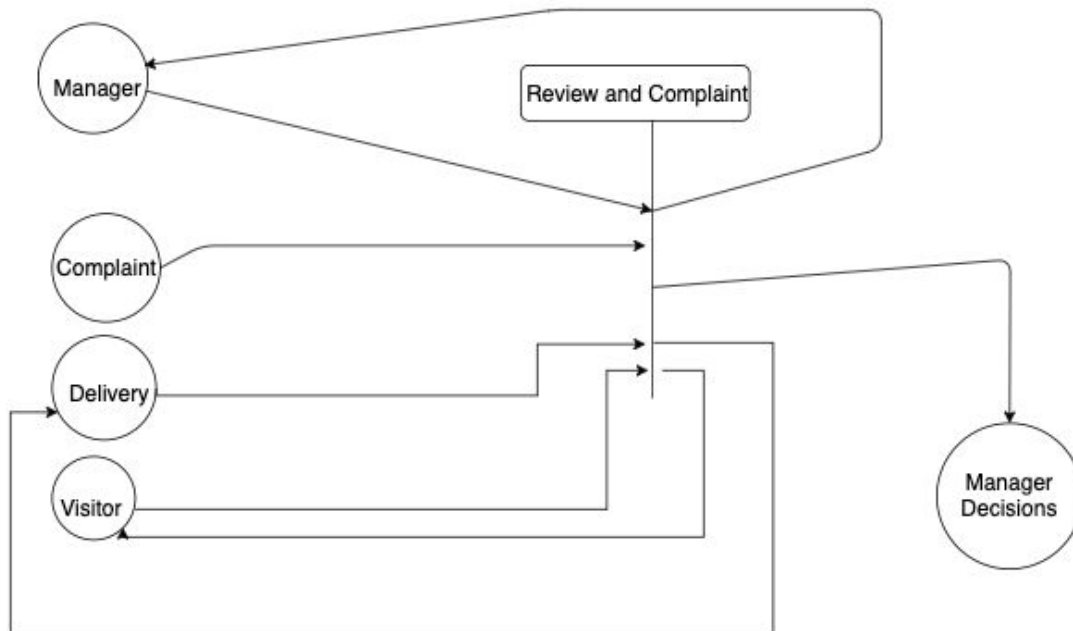
This is for the system to know when bad employees are doing their job wrong such as a chef having too many meals dropped or the delivery boy having too many bad ratings. All information is sent to the manager and has access to everything.



## 2.1.8 Evaluation of Complaints/Payment

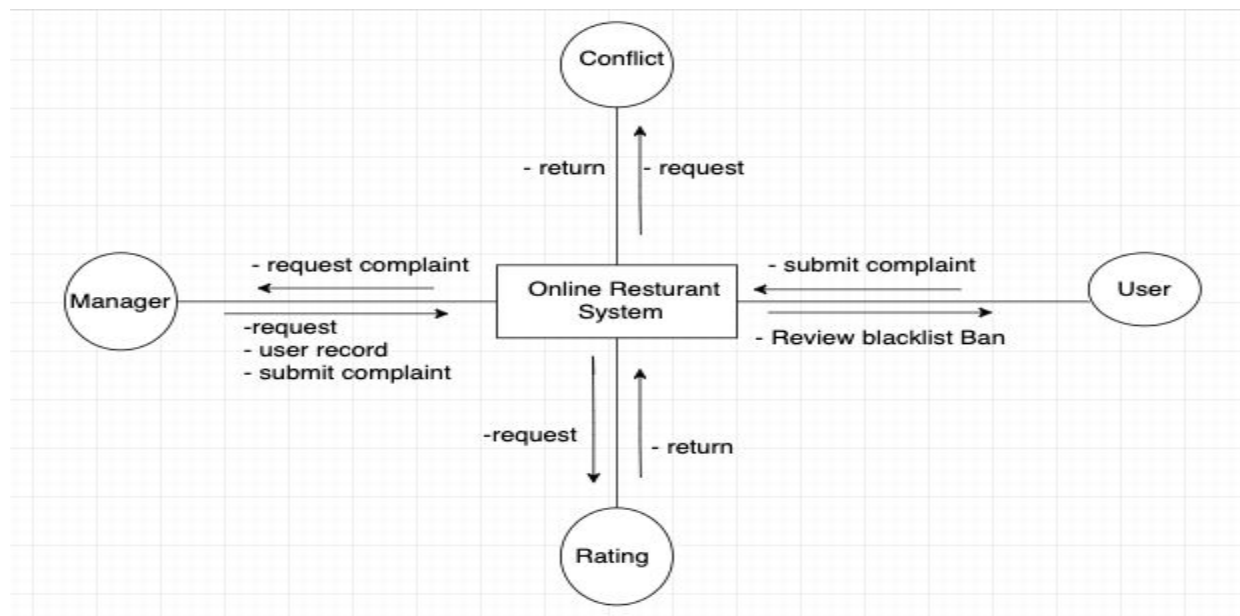
This is for the manager to resolve the submitted complaints and evaluate to do action or erase the complaint. This is also meant for giving paychecks based on the employees contracts.



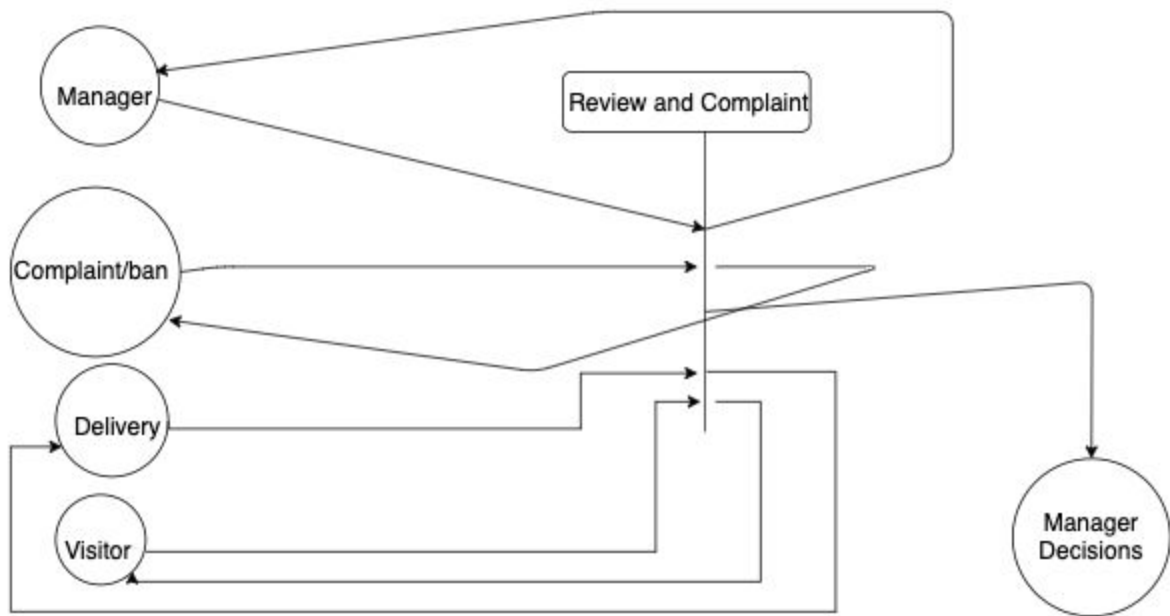


### 2.1.9 Punishments and Warning System

These are the different ways of actions that the manager can do to an employee. This is to show if the user deserves to be fired, ban, or just left on a warning.



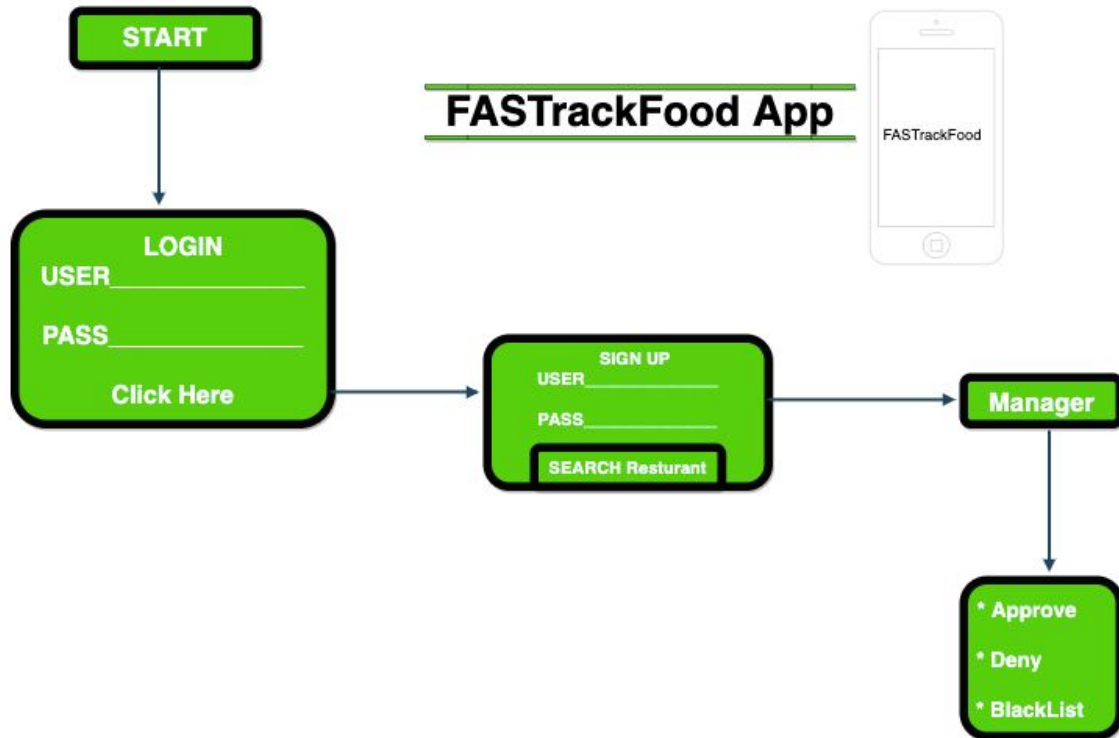


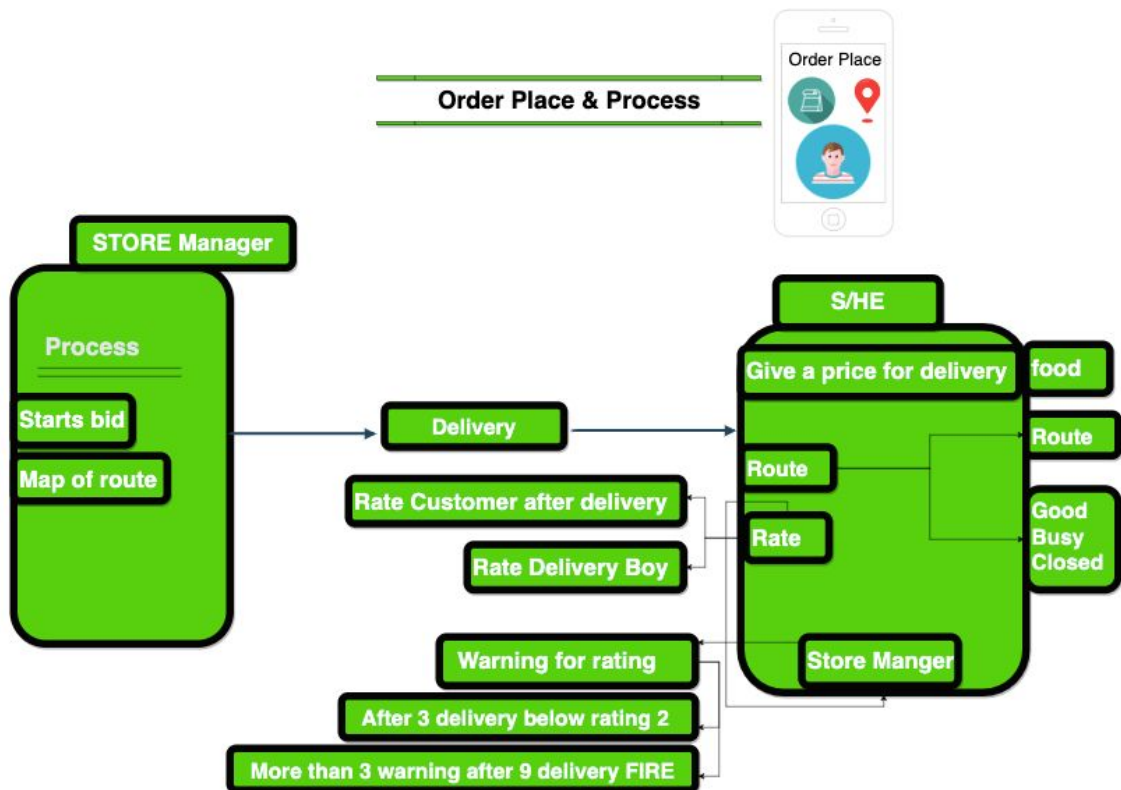
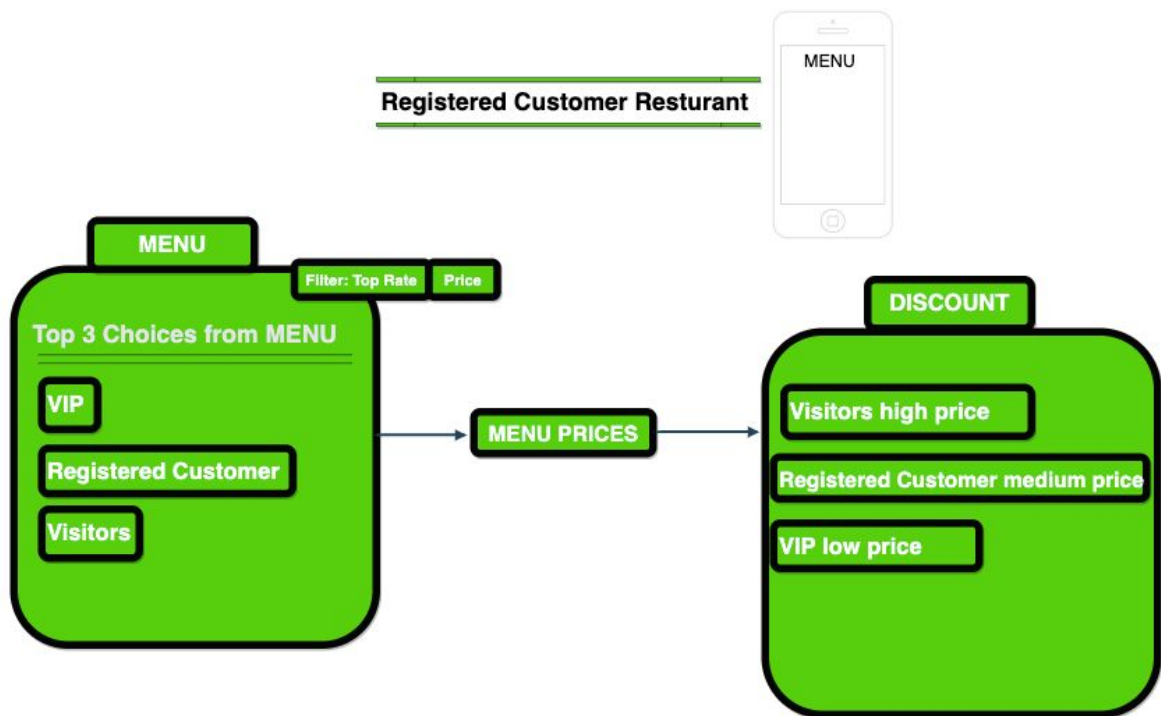


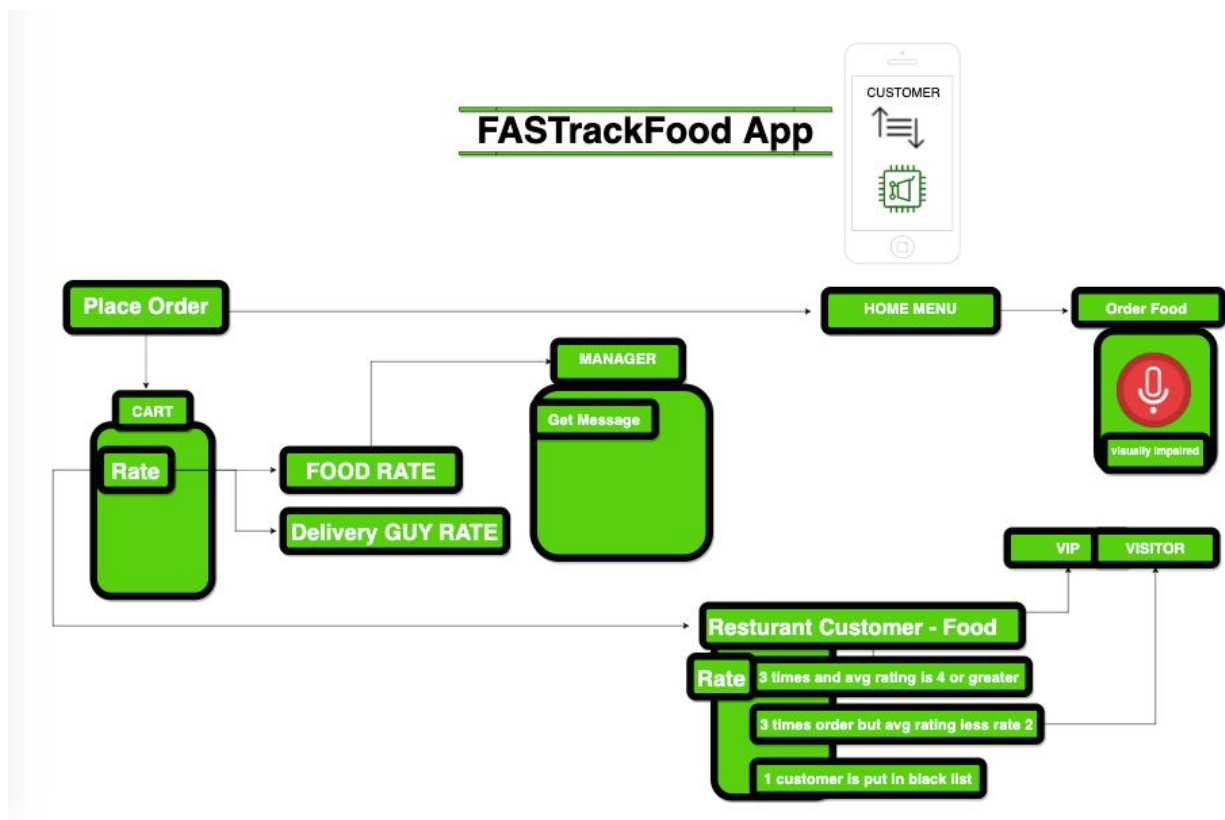
FASTLaneFood	Version: 2.0
Phase II: Design Report	Date: Nov.21.2019
Design Report Final	

## 2.3 State Diagram for Cases

*The Diagrams below show the rough logical process of how the application will run and flow.*



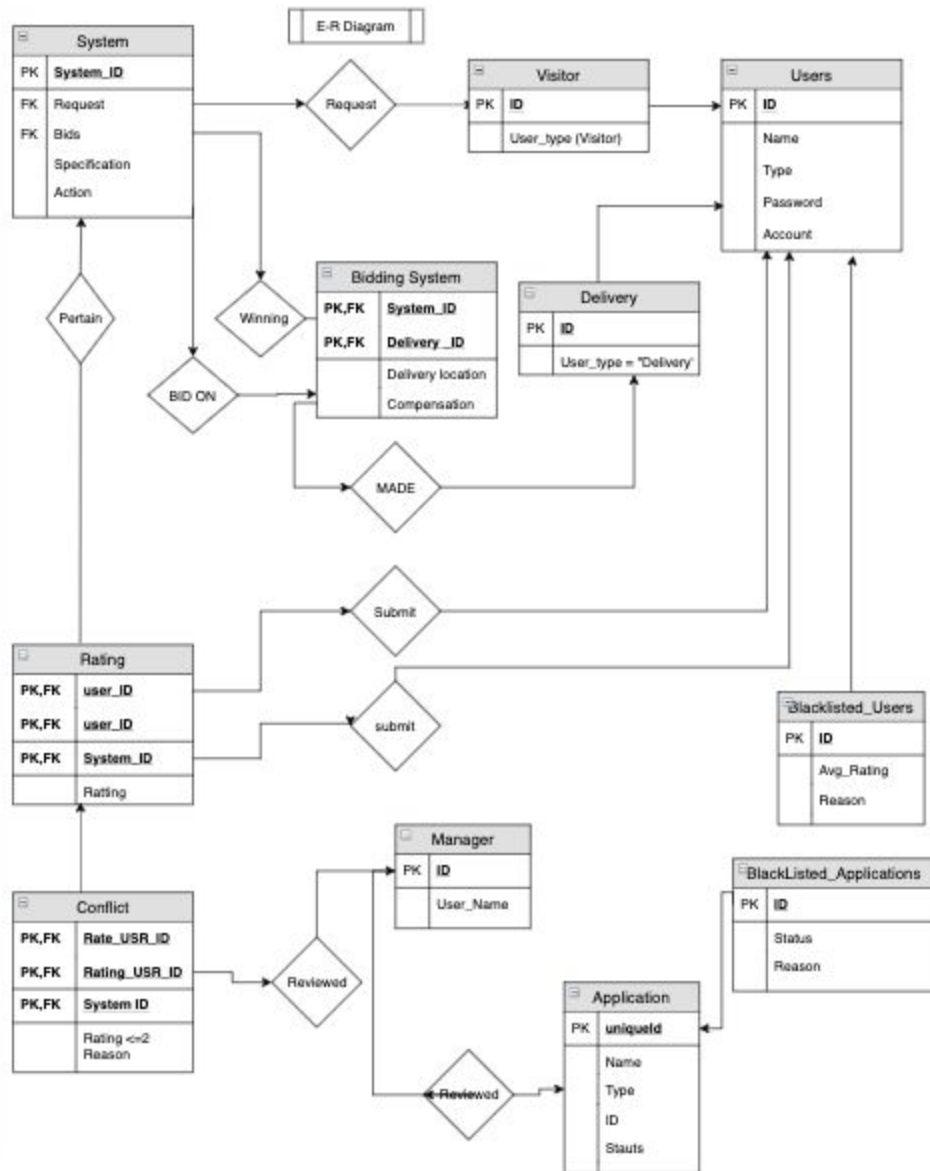




## FASTrackFood App



### 3.0 E-R diagrams



## 4.0 Detailed Design

### 4.1 Methods used and pseudo-code

#### RatingForUser

Every user can receive a rating. If the user receives an average rating of 2 or less during first 3 delivery, user gets warning and store manager is made aware of it. After 3 warnings, the user is fired or removed.

```
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!")
```

#### BlackListUser

If the average rating for a customer is 1 or less. Then customer is flagged and put in blackList.

```
def BlackListUser(avg_user_rate):  
    if(avg_user_rate < 1):  
        Add user to black list
```

#### DropMenuItem

If 3 customers has voted and the item has an average of 2 or less, the item is dropped from the menu.

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

#### AddMenuItem

Allows any new items to be added to the menu.

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

#### RegisterUser

The User is added to user list. The user is prompted to give some simple information to be registered.

```
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.fname = fname
    New_user.lname = lname
    Any more details about the user can be added here...
```

### **Upgrade User**

A user based on his rating gets promoted or demoted. If the user has an average rating of 4 then user is upgraded to be a VIP user. If user has an average rating of 4 then user is upgraded to be a VIP user.

```
def UpgradeUser(user):
```

```
    if(user.order_placed_count >= 3 && avg_user_rating >= 4):
        add user to VIP users
    if(user.order_placed_count >= 3 && avg_user_rating <= 2):
        add user to visitor users
    if(user.order_placed_count >= 3 && avg_user_rating <1):
        add user to blackList
```

### **ComplaintGiven**

If user gives 3 complaints with avg rating of 2 or less, sales person loses 10% commission and receives a warning.

```
def ComplaintGiven(complains_count, userRating):
```

```
    warning = 0
    if (complains_count >= 3 && userName.rate > 2)
        warning +=1
        print("warning: User has received 3 complaints and an average rating of 2 or less")
    userName.commission = userName.currentCommission - (userName.currentCommission *
0.1)
```



### **PlaceBid**

User places a bid. Bid starts with the highest price for the best delivery option. The lowest price to deliver for the best delivery option wins the bidding.

```
def PlaceBid(price):  
    if(price < lowest_bidded_price && best_delivery_option() ):  
        Bidding_value = price
```

### **Best Delivery Option**

The highest rating delivery system gets to be the best delivery option

```
def best_delivery_option():  
    initialize_iterator, go through all options  
    best_option = Option with the highest rating  
    Return best_option
```

### **IsOnBlackList**

This function checks if the user is blacklisted or not. A black listed user has an average rating of 1 or less.

```
def is_on_blacklist(user):  
    #true or false  
    if(user.blackListValue == True):  
        print("BlackListed")
```

### **DisplayRestaurants**

A restaurant must be registered to be displayed. Once registered, the restaurant can now be added to the list of registered restaurant. The list contains the Name of All the Restaurants. To Display the restaurant, one can print the elements of this list.

```
def display_restaurants(resList):  
    print(resList)
```

### **Pay the Employee**

This function pays the employee after the trip is complete. You need to check the user id, which identifies the employee with the id. The machine will pay the employee to the employee's wallet associated to their id.

```
def pay_Employee(login; delivery; user_id) :  
    Initialize the user_id  
    if (delivery = completed) :  
        employee_Wallet += employee_Payment
```

### **Increase the Salary**

This function increases the salary of an employee. You need to check the user's id and permission to authorize it from the manager.

```
def increase_Salary(employee_id) :  
    # make sure they're an employee by checking ID  
    initialize new_Rate  
    if (login = 3) #3 means manager  
        rate = new_Rate
```

### **Display the Menu**

This function will display the menu only for users, or just anyone with an account. Their "login" value should be greater than 0.

```
def display_Menu(login) : #displays menus for users AND admins only NO visitors  
    if (login > 0) : #only users/ admins can view the menu  
        display menu
```

### **Adding a New Meal**

This function will add a new meal to the menu. You need to check user\_id and check whether this account is a manager.

```
def make_New_Meal(user_id) :  
    menu.append(menu item)
```

### **Checking Status of Employee**

This function will check the status of an employee. You need to check whether they are busy on a trip, not busy or offline.

```
def status_Of_Employee(user_id) :  
    if (busy):  
        display that Employee is busy on a trip  
    elseif (not_busy):  
        display that Employee is not busy  
    else :  
        display that Employee is not online
```

### **Adding a New Meal**

This function will add a new meal to the menu. You need to check user\_id and check whether this account is a manager.

```
def complaint_to_Manager(employee_id, login) :  
    If (login > 0)  
        get input from user's complaint from User  
        sendComplaint() : send the complaint to Manager and System
```

### **Visitor's View**

This function will show what the visitor can see on the website. The visitor will be keep reminded to sign up for the website.

```
def visitor_View(login) :  
    if (login = 0) set setting for Login as 0 #0 = visitor  
    SignUp(): #cannot view the site... sends them to sign up / login
```

### **User's View**

This function will show what the user can see on the website. If the user tries to do anything they are not allowed to do, they will not be given access to it. The user will be allowed to rate the delivery and order a delivery.

```
def user_View(login) :  
    if (login = 1) : # check login_view value = 1 (1 means USER)
```

```

CheckDeliveryStatus: #can check users' pick up and drop off
display_Menu():
Order():
if(delivery = 1) : #1 means delivery is finished
    Rate_Delivery(): rate food and delivery guy (rating from 1-5)

```

### **Customer's Promotion**

This function will allow customers to either get promoted or demoted. It will also check whether the customer will become blacklisted from the website.

```

def user_Status() :
    if (order_Placed > 3 && avg_rating >= 4) #become promoted to VIP
        login = 4 (4 is VIP)
    elseif (order_Placed > 3 && avg_rating < 2) #become demoted to visitor
        login = 2 (2 is visitor)
    else (avg_rating < 1) #BLACKLISTED user
        login = 9999 #9999 is BLACKLISTED

```

### **Manager's View**

This function will show what the manager can view on the website. The manager will be allowed to edit and view the menu, check delivery status, order, and erase warnings for employees.

```

def manager_View(login = 3) :
    if (login = 3) : # check login = 3 (1 means manager)
        edit_Menu(): #can edit the menu
        CheckDeliveryStatus() :can check customers' pick up and drop off
        Order():
        display_Menu():
        erase.Warning(): # can erase warning (after 9 deliveries = fired)

```

### **Driver's View**

This function show what the driver can see from the website. "Login" value need to be 2, which will identify the account with the driver.

```

def driver_View(login) :
    if (login = 2) : # check login value = 2 which means DRIVER

```

```
if (warning = 9): receive upto 9 warnings (if exceed, you get fired and becomes a user)
    def fire_Employee(employee_id)
isThereAnyDeliveries();
```

### **Automatically Deleting Food**

This function will delete food if the food rating is less than 2 in 3 orders.

```
def food_Rating() :
    if (food_rate < 2)
        food_mark += 1 #marks a warning
        if (food_mark > 2)
            delete food
        else
            food_mark = 0
```

## 5.0 Systems Screens

### 5.1 GUI Screens

The images below show the mockup of our application.

**FASTFoodLane** Log In Sign Up

Username  
Password  
Log In Sign Up

**Hungry? Log In & Start Ordering!**

**Start Ordering with 3 Simple Steps!**

- Register**  
All we need is your name, address and phone number to know where your food will go to!
- Pick Your Food!**  
Pick which food you want from the thousands of options we have available to you.
- Get It Delivered!**  
After you order and when your food is all ready to go, your food will be on its way!

**Over 1000+ Selections to Choose From**

**Testimonials**

- I ordered my food and it came a lot more quickly than I expected.  
Samuel Smith
- The food I ordered was still hot when it was delivered to me. The delivery time and the service is impeccable.  
Shahan Rahman
- This delivery app is much faster than others in my area. Definitely recommend!  
Pat Zhou

**Get In Touch**

Ask us any questions or concerns you may have. From technical difficulties to an issue with your order, we're here to help! Our customer service is rated as one of the best in U.S. and many other countries.

Message  
Email Address Full Name Send


**Sign Up Today**

Sign Up

Features | About | Testimonials | Contact | Log In | Careers

180 Convent Avenue New York, NY 10081  
© FASTTrack Inc.

(This is the landing page of FastFoodLane with the login)



Log In Sign Up

Username  
Password  
Log In Sign Up

**Hungry? Log In  
& Start Ordering!**

## REGISTER

Name  First Name  Last Name

Username  Username

Email  Email Address

Password  Password


Date of Birth  4/22/2012

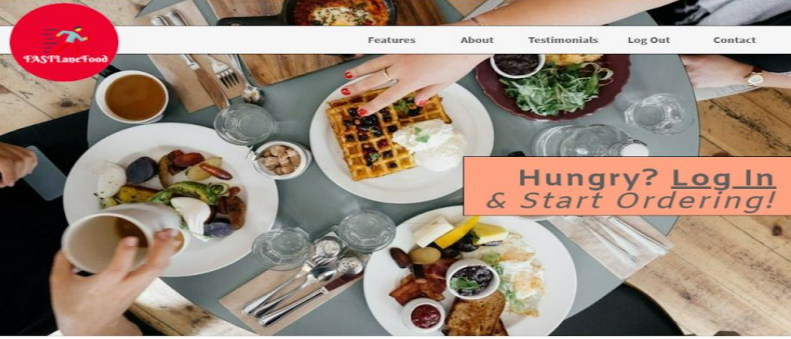
Sign Up

Features | About | Testimonials | Contact | Log In | Careers

160 Convent Avenue New York, NY 10031  
FASTTrack Inc.

(This is registering page)


[Features](#)
[About](#)
[Testimonials](#)
[Log Out](#)
[Contact](#)




**Hungry? Log In & Start Ordering!**

## Delicias Mexicanas

**HOURS:**  
MONDAY  
TUESDAY  
WEDNESDAY  
THURSDAY  
FRIDAY  
SATURDAY  
SUNDAY

24 HOURS  
24 HOURS  
24 HOURS  
24 HOURS  
24 HOURS  
24 HOURS  
24 HOURS



### Popular Dishes

Taco (Choice of Chicken or Steak)	\$2.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>
Quesadilla (Choice of Chicken or Steak)	\$11.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>
Nachos (Choice of Chicken or Steak)	\$4.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>

### Entrees

Taco (Choice of Chicken or Steak)	\$2.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>
Quesadillas (Choice of Chicken or Steak)	\$11.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>
Nachos (Choice of Chicken or Steak)	\$4.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>
Enchiladas (Choice of Chicken or Steak)	\$14.99	<input type="text" value="Select One"/>	<input type="text" value="Qty. 1"/>

### Dessert


Cheesecake	\$4.99	<input type="text" value="Qty. 1"/>
Flan	\$4.99	<input type="text" value="Qty. 1"/>
Jello	\$4.99	<input type="text" value="Qty. 1"/>

### Drinks

Sprite	\$1.99	<input type="text" value="Qty. 1"/>
Coke	\$1.99	<input type="text" value="Qty. 1"/>
Hibiscus	\$1.99	<input type="text" value="Qty. 1"/>

[Back To Restaurants](#)

[Features](#) | [About](#) | [Testimonials](#) | [Contact](#) | [Log In](#) | [Careers](#)

160 Convent Avenue New York, NY 10031  
 FASTrac Inc.

(This is when the customer is ordering on the restaurant's menu)



Home | About | Shop | Help

Your Cart

1. Shopping Cart

2. Delivery Details

3. Payment Options

Shopping Cart

LAS DELICAS MEXICANAS

TACOS

Chicken

\$2.99

3

NACHOS

Chicken

\$4.99

1

Next

Cancel

Summary

ENTER COUPON CODE

SUBTOTAL

\$14.98

DELIVERY SURCHARGE

\$3

TAXES

\$1.50

TOTAL

\$19.48

Home | About | Shop | Help

FASTTrack Inc.

(This is the ordering page)

Home | About | Shop | Help

Your Cart

1. Shopping Cart

2. Delivery Details

3. Payment Options

Delivery Details

First Name

Last Name

Address

Address 2

Country

City

Zip/Postal Code

Phone Number

☒ Delivery

Ordered as soon as possible

☐ Take Out

Next

Cancel

Summary

TACOS

\$9.99

NACHOS

\$4.99

HAVE A VOUCHER?

SUBTOTAL

\$14.98

DELIVERY SURCHARGE

\$3

TAXES

\$1.50

TOTAL

\$19.48

Home | About | Shop | Help

FASTTrack Inc.

(This is the delivery information when the customer enters their location)

Home | About | Shop | Help

Your Cart

1. Shopping Cart

2. Delivery Details

3. Payment Options

Payment method

☒ Credit Card

0000 0000 0000 0000

MM/YY

CW

Card Holder Name

☐ Paypal

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore

PayPal

Pay Now

Cancel

Summary

TACOS

\$9.99

NACHOS

\$4.99

HAVE A VOUCHER?

SUBTOTAL

\$14.98

DELIVERY SURCHARGE

\$3

TAXES

\$1.50

TOTAL

\$19.48

Home | About | Shop | Help

FASTrack Inc.

(This is the payment information)

## YOUR ORDER HAS NOW BEEN COMPLETED



Voldemort is on his way!

(123) 456-7890



### Summary



TACOS

\$9.99



NACHOS

\$4.99

SUBTOTAL \$14.98

DELIVERY SURCHARGE \$3

TAXES \$1.50

**TOTAL \$19.48**

(This is the last process when the food is delivered and ratings are complete)

## 6.0 Meeting Dates

Meeting Dates	Reason of Meeting	Team Work
<i>Mon. Sep 9th</i>	<i>Group Members Meet/Greet</i>	<i>All members were present</i>
<i>Sun. Sep 22</i>	<i>Discussion of Program Lang.</i>	<i>All members were present</i>
<i>Mon. Oct 21</i>	<i>Final Specs discussion/Programing</i>	<i>All members were present</i>
<i>Tue. Nov 12</i>	<i>Progress Update and Software Specs Report Collection</i>	<i>All members were present</i>
<i>Thur. Nov 21</i>	<i>Finalizing Design Report/Programming</i>	<i>All members were present</i>

## 7.0 Github Repo

**GITHUB LINK:** <https://github.com/CcnyUndergraduateCsDegree/FoodProject>