
TEAM T

<CSC 322: Software Engineering Group Project>

Software Requirements Specification For

<Joe's Grab and Go>

Version <2.0>

Front end:

Hyemin Shin
Pablo Lara
Anthony Zhu

Back end:

Anson Oommen
Ali Mohamed

Github: <https://github.com/azhu000/CSC322-SWE-Group-Project>

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Revision History

Date	Version	Description	Author
3/22/22	1.0	Added information to sections 1 and 2	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu
3/25/22	1.0	Added information to sections 3 and 4	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu
3/29/22	1.0	Added information to section 2,3 and 4	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu
4/26/22	2.0	Edited the Report Added more Diagrams and pseudocodes	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu
4/27/22	2.0	Edited User-Interface Prototypes	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions,Acronyms, Abbreviations	5
1.4 Reference	5
1.5 Overview	6
2. Overall Description	6
2.1 Use-Case Model Survey	6
2.2 Assumptions and Dependencies	9
3. Specific Requirement	10
3.1 Use -Case Report	10
3.2 Supplementary Requirement	12
4. Group Meeting Memo	13
5. Pseudocodes	15
6. Supporting Information	21
6.1 ER Diagram	21
6.2 User-Interface Prototypes	22
6.3 Petri-Nets of Use Case	27
6.4 Sequence & Collaboration Class Diagrams	30
6.5 Use Case Diagram	34

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification document is to describe the behavior of the subsystem which is a Restaurant Ordering Website. The report will also go over the functionalities and requirements for the separate components of the system, as well as its interaction with the end user.

1.2 Scope

The software application described is an Online Restaurant Ordering Website. The Website allows the customer to login to their account, order a set of meals, have it delivered or picked up (their choice). Customers can review and comment on dishes, chefs and delivery drivers if their order was delivered. Chefs and delivery drivers can be promoted/demoted and fired based on customer reviews. Based on how much a customer has spent, they may acquire VIP status or have some benefits applied to their account. However, a customer can also be blacklisted, or have penalties applied to the account based on feedback. The Online system includes features such as menu selection, user feedback, VIP selection, as well as various security measures in place. Further functionalities will be specified in the report.

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

1.3 Definitions, Acronyms, and Abbreviations

- **GUI** - Graphical User Interface. Essentially an interface where the end user has a visually interactive environment; to order, leave reviews, etc.
- **SQL** - Structured Query Language. A Programming Language which is used to manage databases and perform various manipulations with data.
- **HTML** - HyperText Markup Language.
- **CSS** - Cascading Style Sheets; is a style sheet language used for describing the presentation of a document in a markup language such as HTML.
- **Bootstrap** - a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- **Python** - Python is an interpreted, object-oriented, high-level programming language with dynamic semantics (do not need to declare variable types)
- **Flask** - is a micro web framework written in Python.
- **Framework** - is a tool that provides ready-made components that are customized in order to speed up development.
- **PHP** - is a server side scripting language that is embedded in HTML. It is used to manage dynamic content and databases.
- **Apache** - is a free and open-source software that allows users to deploy their websites on the internet

1.4 References

Git Reference Document <https://git-scm.com/docs>

Flask Reference Document <https://flask.palletsprojects.com/en/2.0.x/>

PHP Reference Document <https://www.php.net/manual/en/langref.php>

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

1.5 Overview

In the following sections, the report will describe how the online Restaurant Ordering Website will work. The report will go over how the customers can interact with the website, as well as the chefs and the manager of the restaurant. The report will overview the functionality of the backend SQL servers, the interaction with the framework and the frontend website. Furthermore, the report will showcase the requirements needed for certain conditions and features for the customer/chefs.

2. Overall Description

2.1 Use-Case Model Survey

User Types:

- **Visitor:**
 - An individual that can view the website
 - They have the option of registering as well as viewing the menus and ratings
- **Registered Customers:**
 - Can sign in on the website and order meals
 - Are able to browse/search, order and vote (from 1 star to 5 stars) about the quality ranging from food to delivery service
 - Are able to start or participate in a discussion topic on the chefs, dishes or delivery people.
 - Are able to choose whether to pick up the food or have it delivered.
 - **Vip Customers:**

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

- Customers that spend at least \$100 or place 5 orders will receive discounts of 5% on all orders and a free delivery once every 3 orders.
- Will also have special access to specially developed dishes
- Their compliments/complaints are counted twice as important as ordinary ones.
- **Employees:**
 - The employees working at the restaurant and making the meals
 - **Chefs:**
 - Are able to independently decide the menus and make the dishes for the restaurant.
 - Can dispute complaints.
 - **Delivery Driver:**
 - Deliver food to customers
 - Can dispute complaints.
- **Manager:**
 - The person which controls the Restaurant, processes customer registration, handles customer compliments, hire/fire/raise/cut/pay.

Use Cases:

- **Homepage**
 - Will display the general information of the restaurant that will be accessible to all users.
 - This includes the popular dishes and access to the menus.
- **Register Page**
 - Will allow visitors to apply for registration to become customers.
 - **Employee Registration**
 - However, there are certain conditions on registration as an employee
 - For employee registration, the manager will add an account into the database for the employee
 - The employee cannot make an employee account
 - The employee will use the credentials given by the manager to sign into their account
- **Login Page**
 - Will allow registered users (registered customers) to login with a username and password to enter the website as a logged in user.

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

- Certain conditions need to be met. For instance, their account is not blacklisted.

- **Employee Login Page**

- Will be based on the normal login method but will be routed to another page for employee specific site
- There will be a drop down button on the main login page, where they can choose to login as a customer or an employee.
- Will allow registered employees to login with a username and password to enter the website as a logged in employee.

- **Menu Page**

- Will list the currently available dishes that the restaurant has.
- Will list the popular dishes in the main menu page
 - Will list the top 3 most popular dishes and top 3 highest rated dishes for visitor and new customer
 - Will list the top 3 dishes based on the order history for VIP and registered customer

- **Menu/Dishes**

- Each menu can be further expanded upon to see a more detailed description of each dish, reviews, etc.
- Will list special menu for VIP only

- **Users can filter search based on:**

- Popular
- Healthy
- Kids
- Dessert

- **Order Page**

- Registered customers can make an order
- If a user has items in their cart, it will be shown as current items in-cart.
- Users can choose options either to pick up orders in person or have their order delivered to them.

- **Review/Order History Page**

- Will show logged in users their previous orders from the restaurant.
 - This will be referred to when recommending dishes.

- **Ratings/Reviews**

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

- Registered customers can leave ratings between 1 to 5 stars, based on their experience.
- The manager will know what ratings/reviews each chef received as well as the deliverer.
 - Based on the reviews, the manager can administer necessary “punishments” or promotion to the chef/delivery
- Anyone who has received a complaint is allowed dispute the complaint
- The delivery is allowed to compliment or file a complaint on the customer
- **Order Bidding Page**
 - Delivery drivers compete in a bidding battle to be able to deliver orders. The cheapest option is generally chosen for delivery.
 - The manager will be able to assign the order to the deliverer
 - Manager can choose the price for the delivery
 - If he chooses the more expensive option over the cheaper, a justification is required. So, some kind of a form submission might be required.
- **Manager Page**
 - Will allow managers to manage the restaurant like handling registration of users, terminating user accounts, and handling salary of employees.
 - Will have a warning system for customers and employees.
- **Delivery/Receipt Page**
 - Will display information about the delivery, time, the cost, etc.
 - If the deliverer has not delivered an order in that past 5 orders, they will receive a warning
- **Contact Page / About Us**
 - Will provide contact information about the restaurant and will provide information about the staff members. Will also contain the history of this restaurant.

2.2 Assumptions and Dependencies

The main assumptions that we have are the following:

- The framework Flask will work as intended.

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

- Flask will be able to correctly handle the communications from backend and frontend simultaneously.
- Flask will be able to connect to the database and retrieve information from mySQL
- mySQL database is correctly designed
 - The tables within the database are correctly designed, without any corrupted data.
 - Foreign keys are correctly set up to work with the Flask framework.
 - No repetition of tables and or user information
 - Data can be correctly inserted into the tables, as well as correctly displayed from the tables.
- Web-based application
 - Users have access to the internet and can navigate through the UI.
 - Users can sign up to order dishes on the site

3. Specific Requirements

3.1 Use-Case Reports

- All Users
 - **Homepage**
 - Any user can view the homepage and interact with the visible UI. All visitors will be shown a selection of popular menu dishes, and they can also browse through the menus, but are not allowed to order if they aren't registered.
 - The popular dishes are dependent on user type.
 - **Menu Page**
 - All users are allowed to browse through the menu pages
 - Popular

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

- Healthy
- Kids
- Dessert
- **Register Page**
 - All users are allowed to access the register page
- **Login Page**
 - All users are allowed to access the login page
 - Condition being they are not blacklisted, so they are not allowed to login using that account.
- **Contact Page/About Us**
 - Displays contact information about the restaurant. Will contain “fun facts/history” about the restaurant as well as information on all the staff members.
- **Registered Users**
 - **Employee Login Page**
 - This is the login page for employees only
 - This will be a different route to the normal login page for all users.
 - Only employees will be able to login on this page
 - **Menu Page**
 - Only registered users are allowed to order from the menu pages
 - Only registered users are allowed to leave reviews/ratings
 - Only VIP users are allowed to access and to order special menu
 - **Order Page**
 - All registered users and VIPs are allowed to access the order page to make orders
 - Shows the number of orders that are in the cart
 - Shows the price of all the orders
 - **Review/Order History Page**
 - Shows the previous orders they made at the restaurant
 - Show the reviews they made on food, delivery for the previous orders
 - **Ratings/Reviews**
 - Registered users can leave ratings on dishes as well as employees. The manager will have the final decision with the review (he will decide if any actions need to be taken). The person receiving the complaint is also allowed to defend themselves and the manager will decide if any action needs to be taken.

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

- **Order Bidding Page**
 - The Delivery employee can bid on the order. The manager will get the final decision on which bid will go through (justification is required)
- **Manager Page**
 - The manager essentially controls everything. Manager will have control over confirming orders, complaints, pays, cuts. Manager will also have access to account control and can create accounts for the employees.
- **Delivery/Receipt Page**
 - Displays information about the order in the receipt page, the cost on the delivery, time, etc.

3.2 Supplementary Requirements

The website has a series of software requirements that are free as well as some non-free hardware and system requirements that are needed to run a web-server. Some of these non-free requirements will include a stable and reliable internet connection, and server hardware to host the webserver on. For this project's use, a personal computer that uses a x86, 64-bit CPU. If the user fulfills the previously mentioned requirements, then the following free requirements must be installed on the server: operating systems Windows 7 or Windows 10, Mac OS X 10.11 or higher, 64-bit, or Linux: RHEL 6/7, 64-bit.

- Flask requires that the system use Python 3.6 or newer with async support only working on Python versions 3.7 or higher.
- MySQL version 8.0 or higher
- Bootstrap 5.0 or higher

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

4. Group Meeting Memo

Date	Attendees	Meeting Agenda	Action
2/15/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Chose Programming Language - Shared opinions regarding web development environment - Shared a possible meeting schedule 	<ul style="list-style-type: none"> - Performed research on options for web development and set up and shared the information in next meeting
2/22/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Decided on project set up - Assigned a role for each group member on project <u>Front-end</u>: Pablo/Hyemin <u>Back-end</u>: Anson/Ali/Anthony - Decided regular team meeting schedule → Every Tuesday 3:30 pm - Created Git-hub repository for project → Team T repository 	<ul style="list-style-type: none"> - Studied/Performed research on different development environments - Set up the development environment by next meeting → Flask/python3/sqlite3/Git
3/1/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Confirmed that the development environment is set up correctly and assist others if there were any issues - Planned a development schedule for Front-end and Back-end 	<ul style="list-style-type: none"> - Studied languages and programs needed for development → HTML/CSS/python/Flask/Git/Javascript/PHP
3/10/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Updated each other on everyone's progress - Shared feedback on development - Shared opinions during meeting 	<ul style="list-style-type: none"> - Built a mockup for User-Interface for clearer communication - Created personal branches on git-hub repository - Worked on development

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

3/15/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Shared mockup for UI and gave feedback → Mockup 1.0 - Reviewed project specifications and clarification on the use-cases - Reassigned roles for a member regarding workload → Anthony: BE → FE 	<ul style="list-style-type: none"> - Worked on development
3/22/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Assigned roles for project report 1.0 - Checked the progress on development - Reviewed if front-end part aligned with back-end database 	<ul style="list-style-type: none"> - Worked on project report accordingly - Worked on development
3/29/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Reviewed project report - Completed project report 1.0 → Team T Project report 	<ul style="list-style-type: none"> - Worked on development
4/5/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Suggested ideas to improve workflow and communication - Checked work progress - Revised a project schedule according to work progress 	<ul style="list-style-type: none"> - Worked on development
4/24/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Checked work progress - Merged branches on Git-hub - Brainstormed on how to connect front-end and back-end 	<ul style="list-style-type: none"> - Worked on development
4/26/22	Anson Oommen Ali Mohamed Hyemin Shin Pablo Lara Anthony Zhu	<ul style="list-style-type: none"> - Assigned a role for project report 2.0 - Checked work progress on development 	<ul style="list-style-type: none"> - Worked on project report accordingly - Worked on development
4/28/22	Anson Oommen Ali Mohamed	<ul style="list-style-type: none"> - Reviewed project report and edited parts that needed to be 	<ul style="list-style-type: none"> - Worked on development

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Hyemin Shin Pablo Lara Anthony Zhu	- improved Completed report	- Reviewed functionality of front-end part
--	--------------------------------	--

5. Pseudo Codes

Checking VIP

```
checkVIP(user){
    amntSpent = retrieve data from db
    Orders = retrieve from db
    If (amntSpent > 100 || number_of_orders >= 5)
        userStatus VIP = true
        Update db
        Commit db
    }
```

Checking User Credentials/Login

```
Def login() {
    Post request
    Data = requestform.data(username)
    DB.query.filterby(data).first()
    If true
        checkBlackList(data)
        Blacklisted
        exit
        Query.data.password
        Check password
        If true: login
        Else: wrong password
    Else: no username exist, try again
}
```

Employee Login (modified user login)

```
Def login()
```

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

```

Post request
Data = requestform.data(username)
DB.query.filterby(data).first()
If true
    checkBlackList(data)
        Blacklisted
        Exit
    Check(isEmployee?)
        Query.data.password
        Check password
        If true: login
        Else: wrong password
    Else: no username exist, try again
}

```

Registering Users

```

Def register() {
    Post request
    Data = requestform.data(username)
    DB.query.filterby(data).first()
    If true: Name exists, try again
    Else
        Newuser = DB(username, password)
        DB.add(newuser)
        DB.commit
}

```

Dish Reviews

```

Def addReview(){
    Data1 = requestform.data(rating)
    Data2 = requestform.data(comment)
    NewRating = DB(customer, dishID, data1, data2)
    DB.add(NewRating)
    DB.commit
}

```

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Dish Order

```

Def PlaceOrder() {
    On post request
    If(moneyInWall < Total)
        Print insufficient fund
        warnCust()
    Else
        CheckVIP(customer)
        Data1,2,3 = requestform.data(DishID, amount, total)
        NewOrder = DB(customer, data1, data2, data3, discount)
        DB.add(NewOrder)
        DB.commit
}

```

Adding Money to Wallet

```

Def addMoney(){
    Post request
    Amnt = requestform.data(amount)
    NewAmnt = DB(customer, Amnt)
    DB.add(NewAmnt)
    DB.commit
}

```

Customer/Employee Complaints

```

Def addComplaint(){
    Data1 = requestform.data(complaint)
    NewComplaint= DB(customer/employee, complaint)
    DB.add(NewComplaint)
    DB.commit
}

```

Manage Complaints and Compliments

Warn Employee

```

Def WarnEmpl() {
    DB.query.filterby(employee.ratings < 2 || employee.complaints >3)
        payCut(employee)
        demote(employee)
}

```

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

```
If (DB.query.filterby(demote >= 2)
    fireEmpl(employee)
}
```

Pay Raise

```
Def PromoteEmpl() {
    DB.query.filterby(ratings > 4)
        payRaise(employee)
}
```

Warn Customers

```
Def WarnCust() {
    DB.query.filterby(customer.complaints)
        warningList(customerID)
    If(to many warning)
        BlackList(customerID)
}
```

Cancel Complaint

```
Def CancelComplaints(){
    DB.query.filterby(employee.complaints>=1 && employee.complement>=1)
        employee.complaints = employee.complaints - 1
        employee.complement = employee.complement - 1
}
```

Delivery

Customer chooses pickup

```
Def delivery() {
    If (choose pickup?) {
        DB.query.filterby(VIP?)
            Apply discount
            Return destination}
    Orders += 1
}
```

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Customer chooses delivery

```
Def delivery() {
    If (choose delivery?) {
        DB.query.filterby(VIP?)
        Apply Discount
        Return Bidding()
    Orders += 1
}
```

Bidding

```
Def bidding() {
    Del1 = randNum(1,10)
    Del2 = randNum(1,10)
    Del3 = randNum(1,10)
    If (del1 <= del2 && del3)
        Win = del1
        FinManager(win, del2, del3)
    If (del2 <= del1 && del3)
        Win = del2
        FinManager(win, del1, del3)
    Else
        Win = del3
        FinManager(win, del1, del2)
}
```

Check Delivery

```
checkDel() {
    If(order Made) {
        Update employees
        Set OrdMiss = Replace (OrdMiss, int, 0)
    Where
        Employee = delivery
    DB.commit
}
```

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

```

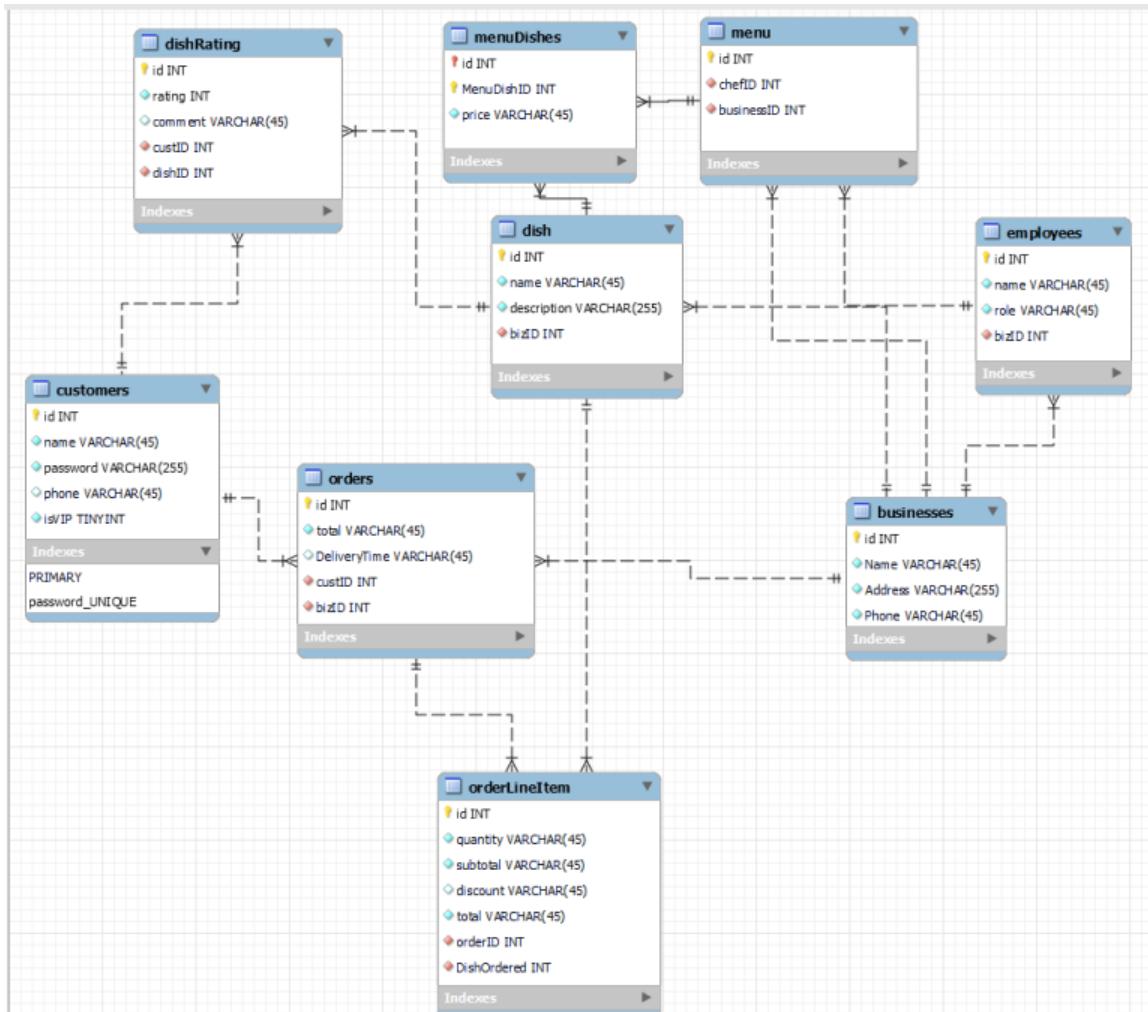
Else If (missed Order) {
    Update employees
        Set OrdMiss = Replace (OrdMiss, int, int + 1)
    Where
        Employee = delivery
    DB.commit
    }
DB.query.filterby(ordMiss >= 5)
WarnEmployee(self)
    Update employees
        Set OrdMiss = Replace(OrdMiss, int, 0)
    Where
        Employee = delivery
    DB.commit
}

```

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

6. Supporting Information

6.1 ER Diagram



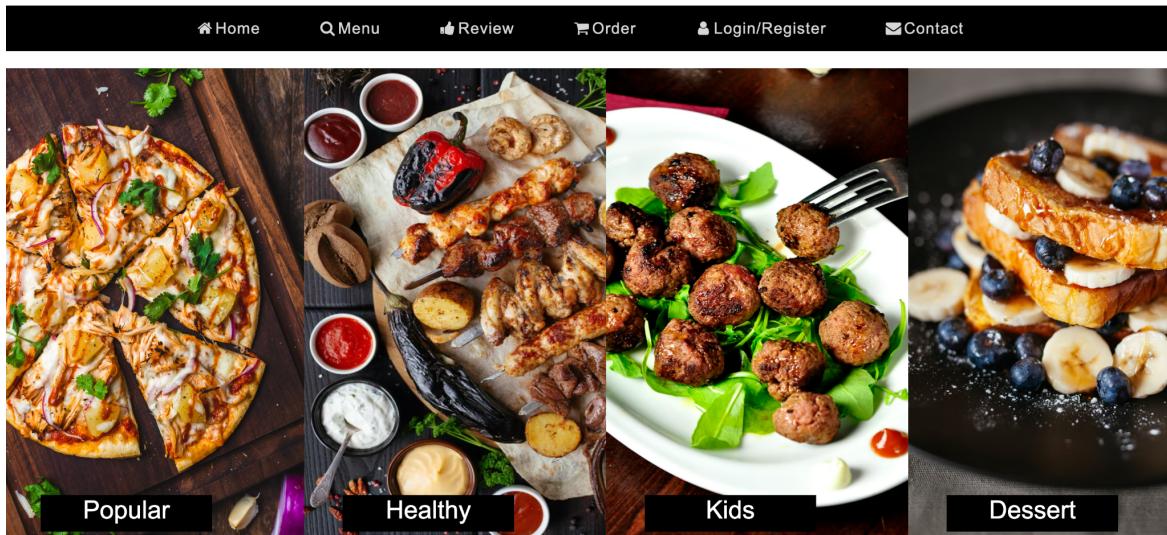
Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

6.2 User-Interface Prototypes

1) HomePage

Joe's Grab and Go

Order Food to Your Door :D



2) menu

Joe's Grab and Go

Order Food to Your Door :D

The screenshot shows a menu page for "Appetizers". It features four cards, each displaying a bowl of food and some descriptive text. Each card includes a blue "Order" button at the bottom.

Dish Name	Dish Name	Dish Name	Dish Name
FoodDetail: FoodDetail Price: \$\$\$\$\$ Review: ★★★★			
Order	Order	Order	Order

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

3) Order_cart

Joe's Grab and Go

Order Food to Your Door :D

The screenshot shows the 'My Cart' section of the website. At the top, there is a navigation bar with links: Home, Menu, Review, Order, Login/Register, and Contact. Below the navigation bar, the title 'My Cart' is displayed next to a shopping cart icon. A message indicates 'You have 3 items in your cart'. A table lists the items in the cart:

Dish name 1 Brief description	\$12
Dish name 2 Brief description	\$8
Dish name 3 Brief description	\$5
Discount VIP Discount	-\$5
Total (USD)	\$20

At the bottom of the cart area is a blue button labeled 'Continue to Check Out'.

4) Order_checkout

Joe's Grab and Go

Order Food to Your Door :D

The screenshot shows the 'Check Out' page. At the top, there is a navigation bar with links: Home, Menu, Review, Order, Login/Register, and Contact. Below the navigation bar, the title 'Check Out' is centered. There are two checkboxes for delivery options: 'I will pick up my order' and 'Please deliver my order to door'. A large input field is labeled 'Delivery address'. Below this, there are two side-by-side input fields for 'First name' and 'Last name'. Underneath these are two input fields: 'Address' containing '1234 Main St' and 'Address 2 (Optional)' containing 'Apartment or suite'.

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

State Zip

Save this address for later

Payment

Total Amount <input type="text"/>	Available Balance <input type="text"/>
Check Out	

5) Order_history

Joe's Grab and Go

Order Food to Your Door :D



Order History

Order Number #####	mm/dd/yyyy	Delivered	\$25
- Ordered item 1 - Ordered item 2 - Ordered item 3			
Order Number #####	mm/dd/yyyy	Picked up	\$35
- Ordered item 1 - Ordered item 2 - Ordered item 3			
Total (USD)		\$60	

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

6) Login

Joe's Grab and Go

Order Food to Your Door :D



Login

Enter your Email address and Password

Remember me

Login

Don't you have account? [Register](#)

7) Register

Joe's Grab and Go

Order Food to Your Door :D



Create Your Account

Don't have an account?
Get your account in a minute.

Remember me

Register

Do you have an account? [Login](#)

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

8) Contact Page

Joe's Grab and Go

Order Food to Your Door :D

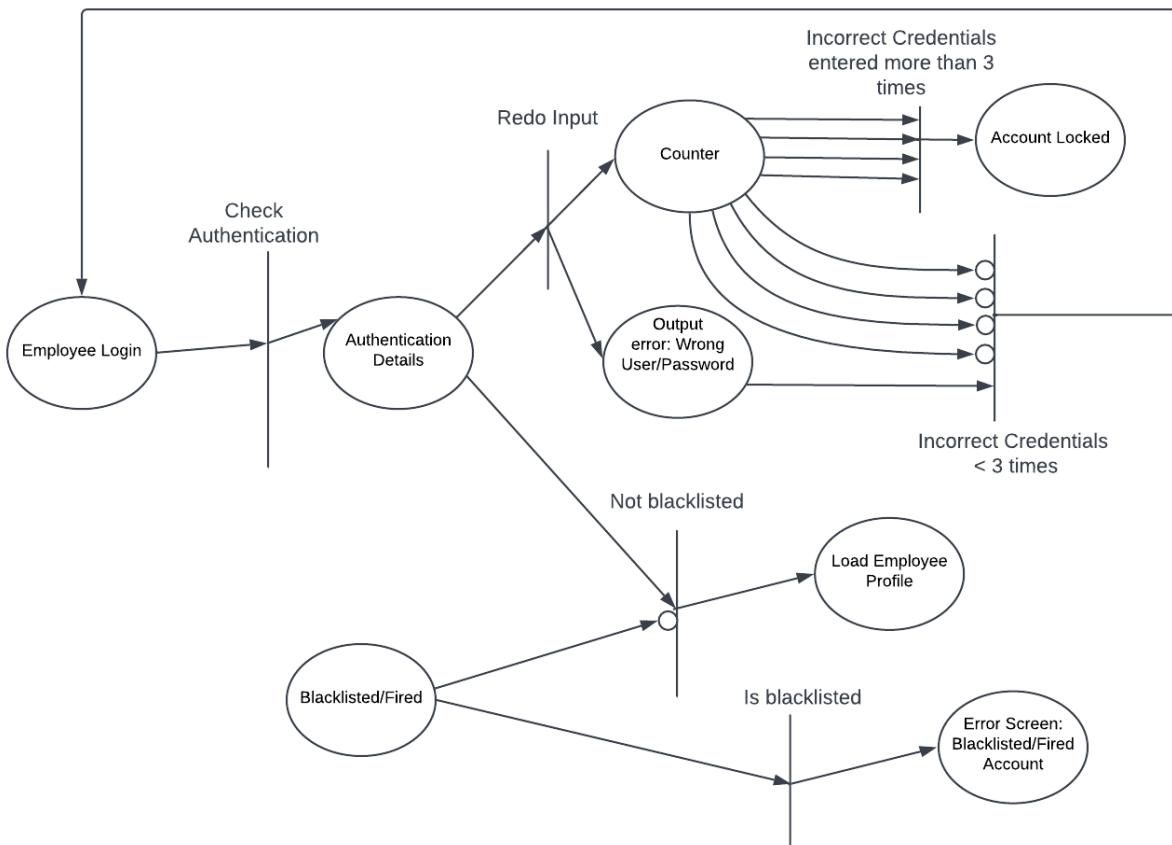
Contact Information down here

Address: 1234 Main Street New York, NY, 10044
Telephone: 1-(212)-345-1234
Email Address: thebestrestaurant@restaurants.com

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

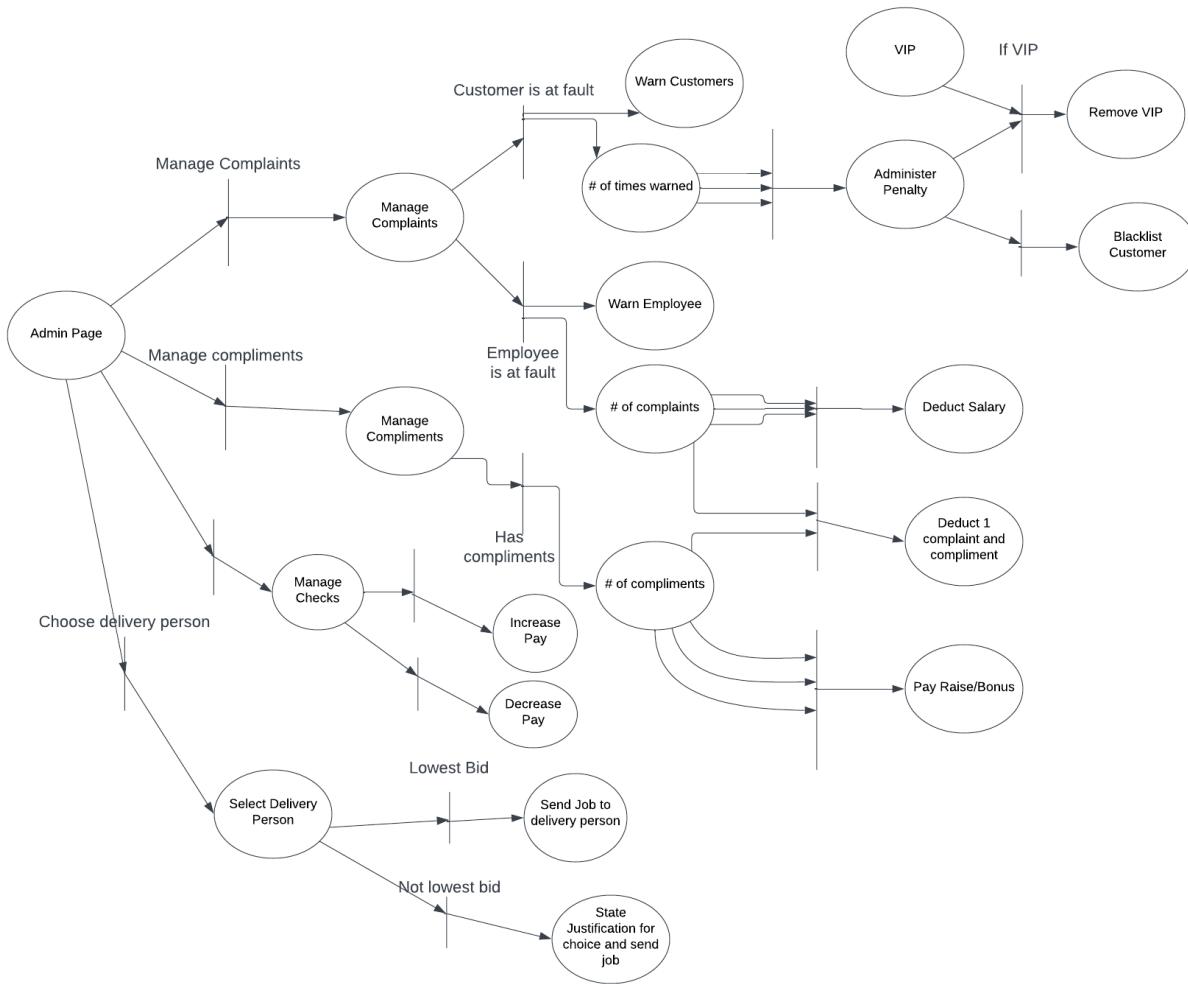
6.3 Petri-Nets of Use Case

Login Petri Net



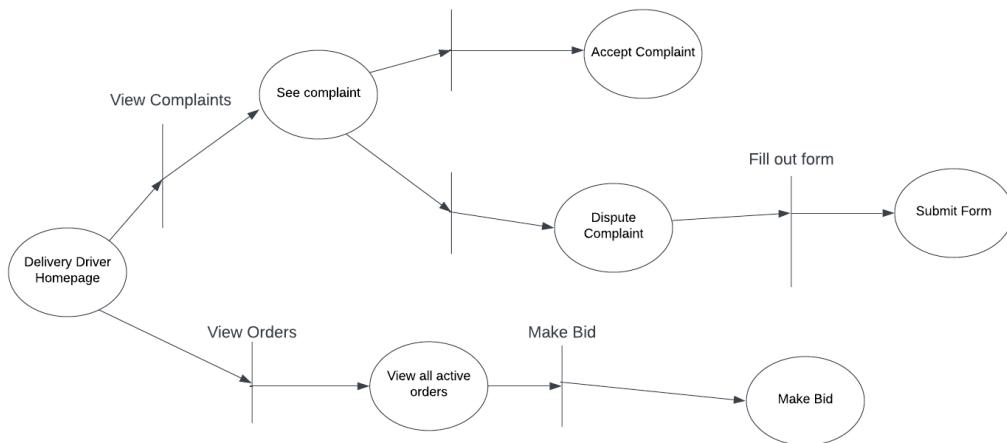
Manager Page Petri Net

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	



Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

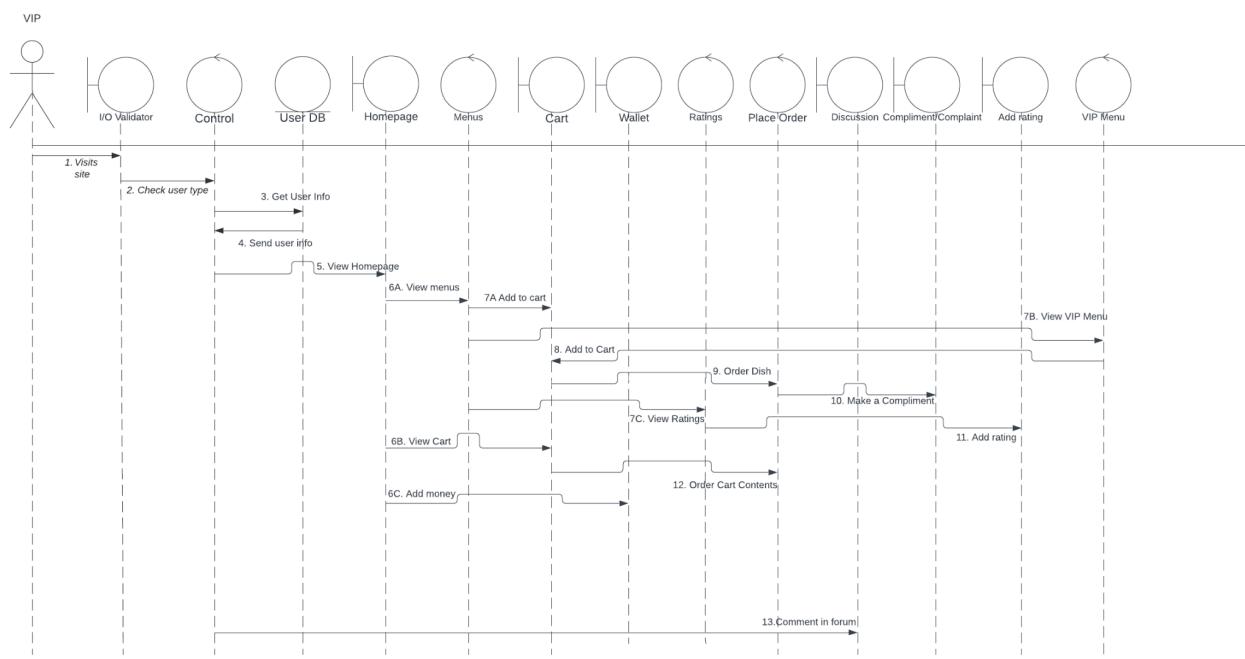
Delivery Driver Homepage Petri Net



Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

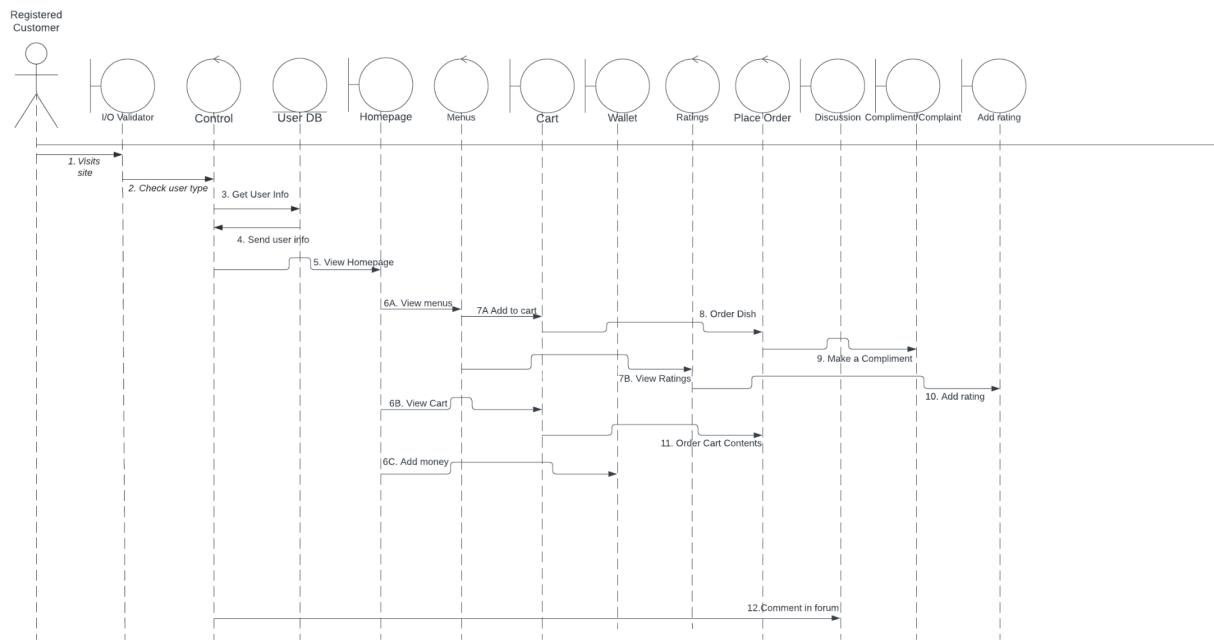
6.4 Sequence & Collaboration Class Diagrams

Vip Sequence Class Diagram:



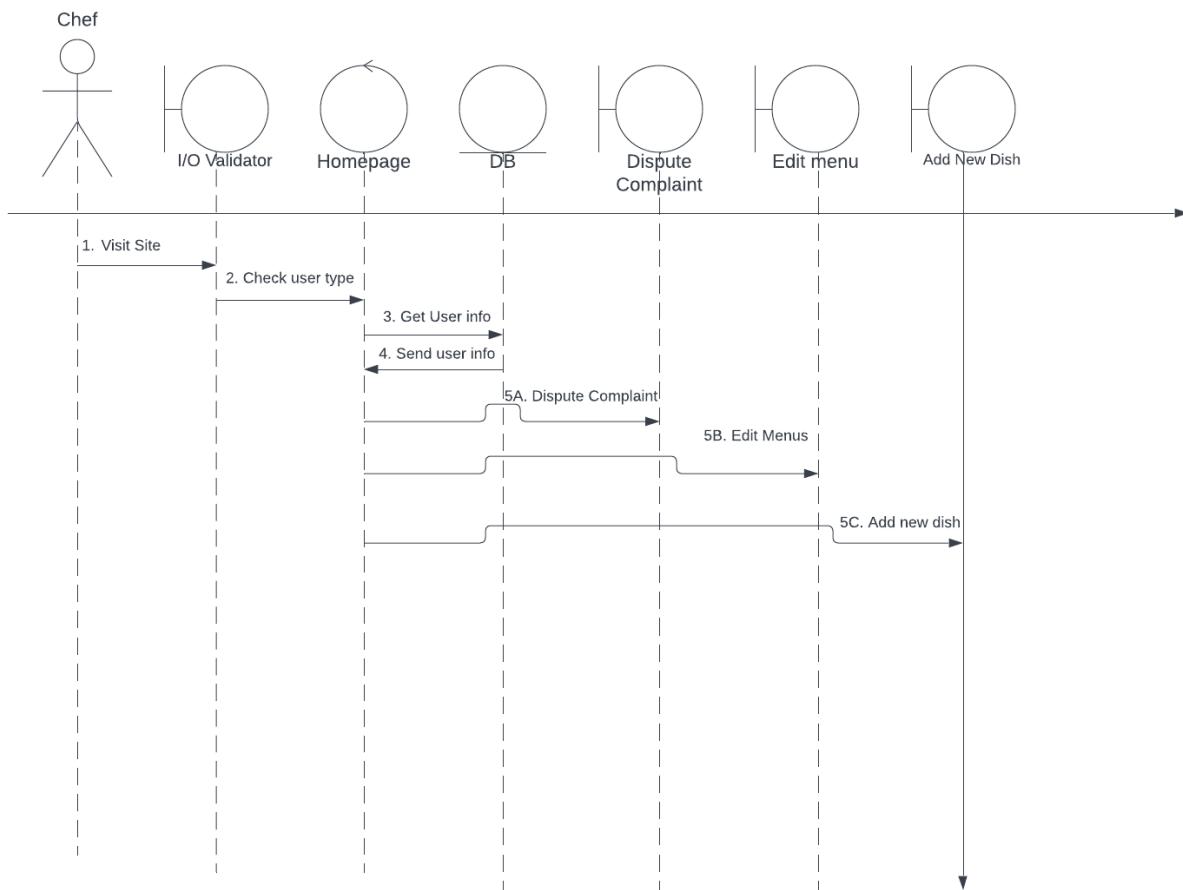
Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Registered Sequence Class Diagram



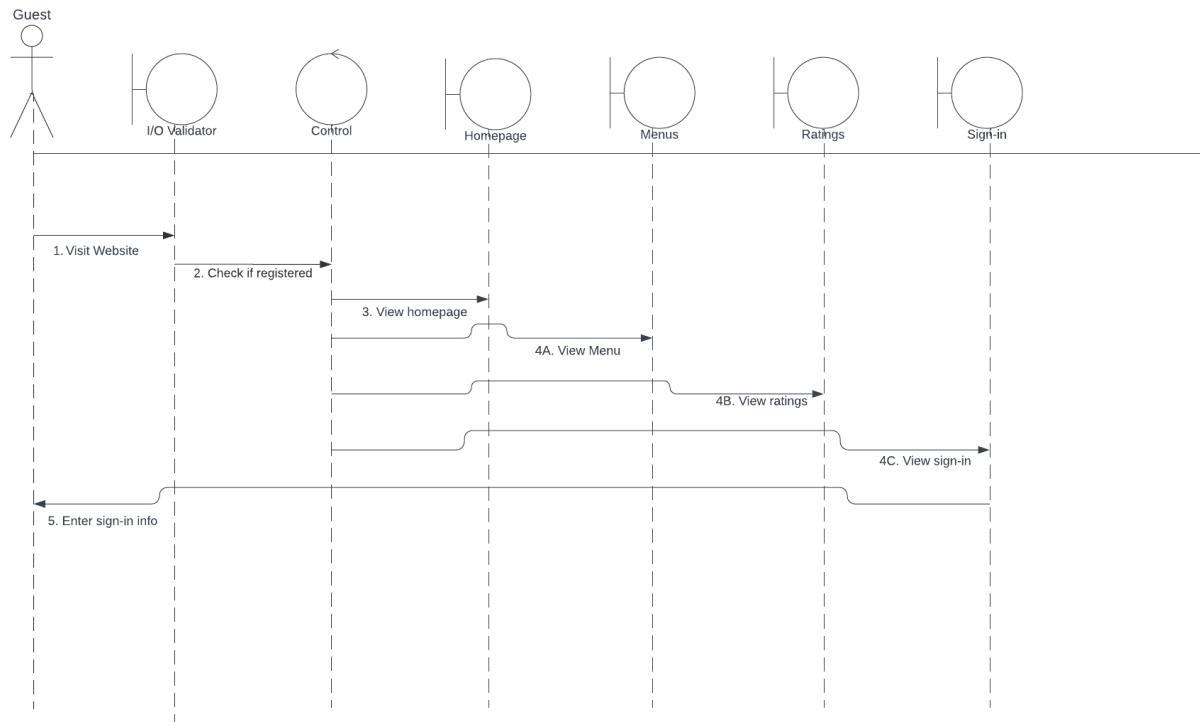
Chef Sequence Class Diagram:

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	



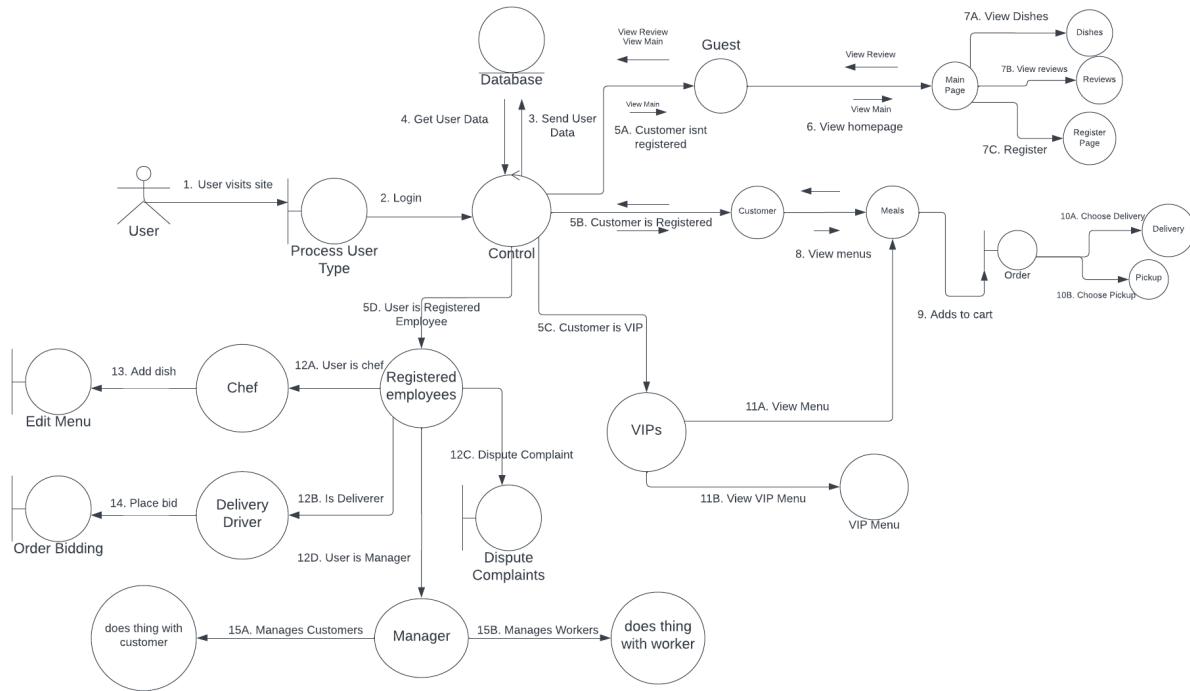
Guest Sequence Diagram:

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	



Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

Collaboration Class Diagram



Use Case Diagram

Online Restaurant & Delivery	Version: <2.0>
Software Requirements Specification	Date: 4/26/2022
Team T	

