

# CPTS 583 Software Quality Deliverable 2-2

## Restaurant Automation System

Team name: Code Cougar

Team member: Jianqiao Liu, Wen-Chih Li, Parikshit Panwar

## 1 Introduction

The software product proposed is “Restaurant Automation”, which focuses on providing convenience monitoring and handling of various tasks of waiter, busboy, manager, and chef. Each user category has their own privileges for handling their respective tasks.

## 2 Installation and Configuration:

**Link to repository:**

<https://github.com/Jimmy-WSU/Restaurant-Automation-CodeCougar>

The software product is made as a web application and therefore shall require and supports Google Chrome. However, due to time constraints, the project is not deployed to the cloud server. User has to do some installation to run the software.

PS: Development environment is based on Windows.

### 2.1 Database

The database used in the project is MySQL Community Server 8.0.23.0.

User needs to install MySQL to use this system. During the installation, the user needs to set the username to "root" and password to "1234" when establishing the connection.

After the installation, user need to go the Windows Command Prompt to execute these command line:

Use “`mysql -u root -p`” and then input the password “1234” to login the mysql.

Execute these two commands to reset the password and privileges:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '1234';
```

```
FLUSH PRIVILEGES;
```

After that, user need to create a database called restaurant\_automation. Execute this command to create the database:

```
CREATE DATABASE restaurant_automation;
```

And use this command to use the database:

```
USE restaurant_automation;
```

Then, user needs to execute the SQL file to create tables, the location of the SQL file is "Restaurant-Automation-CodeCougar\server\sql\restaurant\_automation.sql". The command is (do not forget to add the name of the SQL file):

**SOURCE** [location of restaurant\_automation.sql on your computer];

## 2.2 Back-end

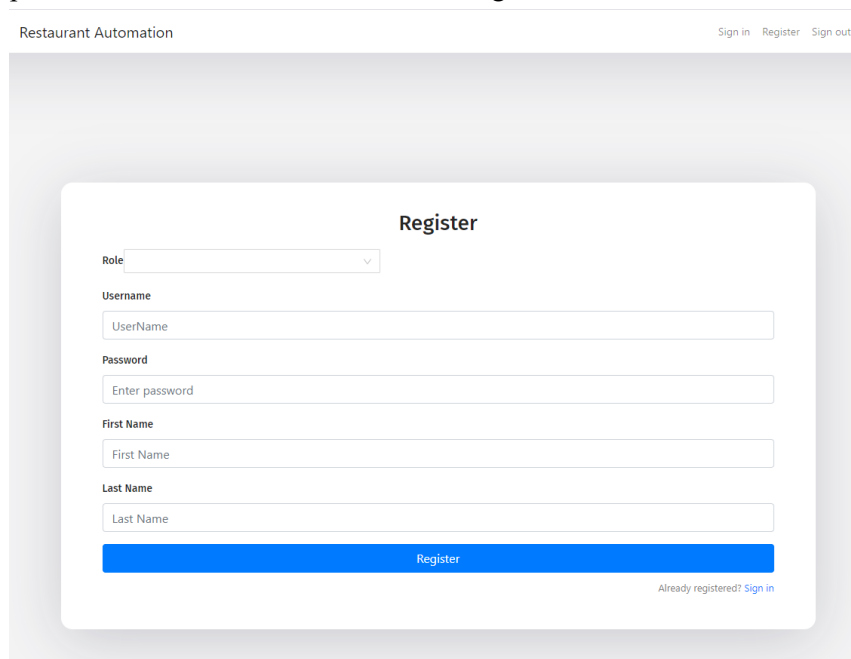
The back-end server and front-end server should be running at the same time. The server we used for back-end is Express. First, user needs to install Node.js, and then goes to the "Restaurant-Automation-CodeCougar\server" in the PowerShell and executes the order "npm install" to install the node modules. After that, user can use "npm start" command to run the back-end server.

## 2.3 Front-end

First, user needs to go to the "Restaurant-Automation-CodeCougar\client" in the PowerShell and executes the order "npm install" to install the node modules. After that, user can use "npm start" command to run the front-end server.

# 3 Usage for different types of users

All new users need to register to use the system. User can click the "Register" button to go to the register page. On the register page, user can select his/her role in the restaurant, input username, password, first and last name to do the registration.



The screenshot shows the 'Register' page of the 'Restaurant Automation' system. At the top, there is a navigation bar with 'Sign in', 'Register', and 'Sign out' links. The main content area features a white card titled 'Register' with the following fields: a 'Role' dropdown menu, a 'Username' text input, a 'Password' text input with a placeholder 'Enter password', a 'First Name' text input, and a 'Last Name' text input. A prominent blue 'Register' button is located at the bottom of the card. Below the button, there is a link that says 'Already registered? Sign in'.

Fig. 1, Register

Restaurant Automation

Sign in Register Sign out

**Sign In**

UserName

Enter username

Password

Enter password

Submit

Fig. 2, Sign In

The user has to first sign in by providing the registered username and password to be able to use the software.

Since this software will be governed by different categories of users, therefore, each category of user will have the following privileges:

### 3.1 Waiter

The waiter could go to the food menu page to select food and choose the table which is free to take orders for customers.

**Foodmenu**

[Back](#)

Welcome: SteveWaiter

[Get Food Menu](#)

<input checked="" type="checkbox"/>	Food Name	Food Price	Sales
<input type="checkbox"/>	Hamburger	1	1000
<input checked="" type="checkbox"/>	Steak	9.99	100
<input checked="" type="checkbox"/>	French fries	1	500
<input type="checkbox"/>	Salad	5	100
<input type="checkbox"/>	Dumplings	2.99	40

Total Price: 10.99

Free Table Number:

[Submit](#) [Cancel](#)

< 1 >

Fig. 3, Take Order

After creating the order, it will go to the order detail page to show the details of the order:

**Order Details**

[Back](#)

[Get Order Details](#) [Serve The Foods](#)

Order ID: 56

Table ID: 4

Waiter Name: SteveWaiter

Chef Name: Unassigned

Food List: Steak, French fries

Total Price: 10.99

Order Status: Started

Create Time: 2021-03-12T15:04:49.000Z

Fig. 4, Order Details

On order details page, waiters should pay attention to the order status, waiters could click the "Get Order Details" button to get the latest status of the order. When "Order Status" shows "Ready", waiters could go to kitchen to bring the food to customers. Then, waiters could click the "Serve the Foods" button to finish the order.

Also, waiter can go to the order list page to get the list of the "Ready" (which means the chef have finished the foods) order and serve the food.

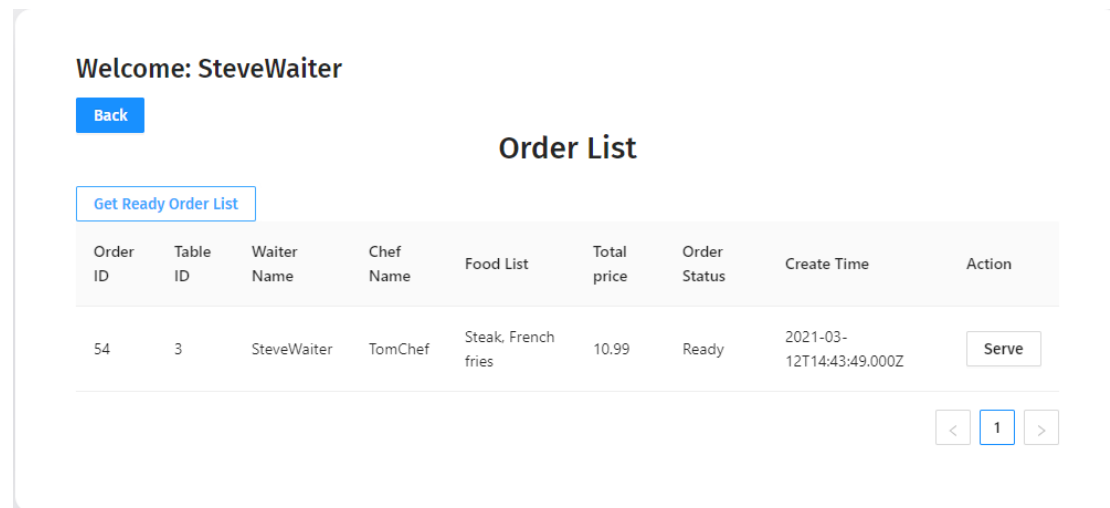


Fig. 5, Waiter Order List

### 3.2 Chef

The chef will be able to view the list of orders which are not assigned by other chefs.

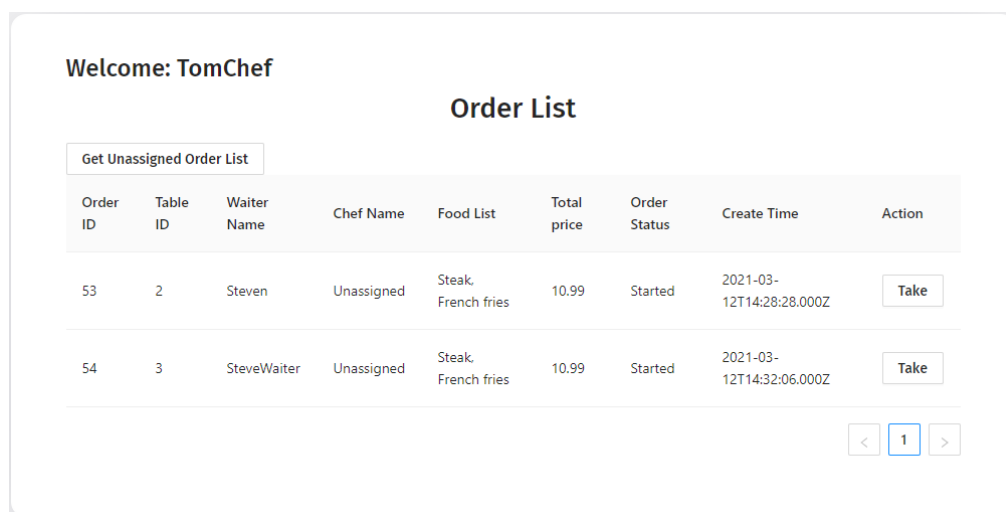
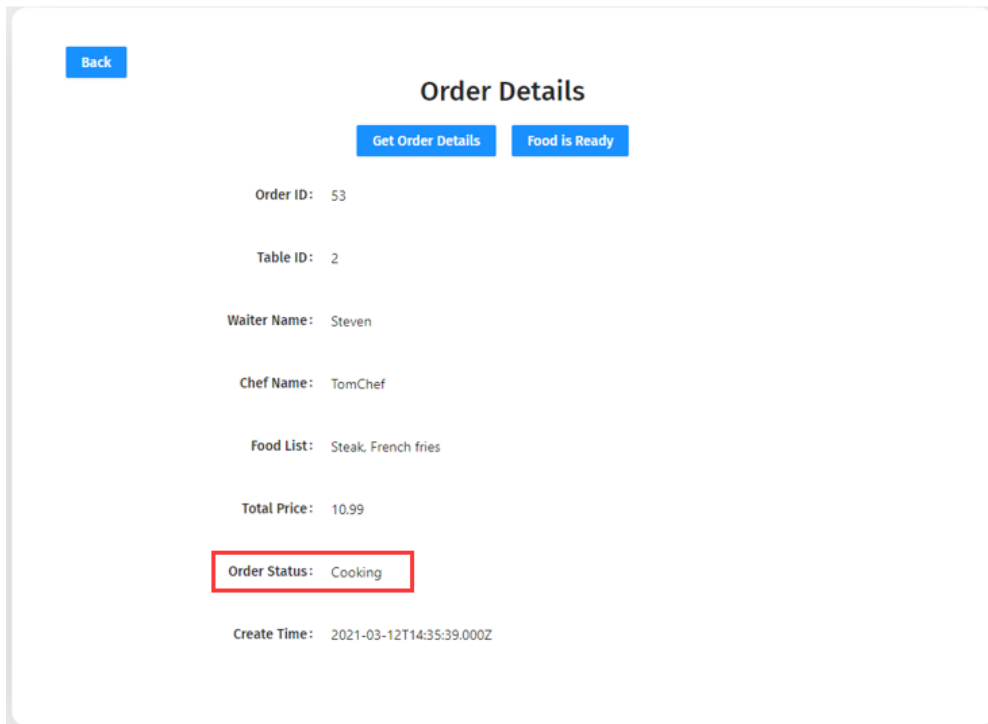


Fig. 5, Chef Order List

Chef can click the "Take" button to take the order and start to cook the foods. After clicking the "Take" button, the status of the order will change to "Cooking".

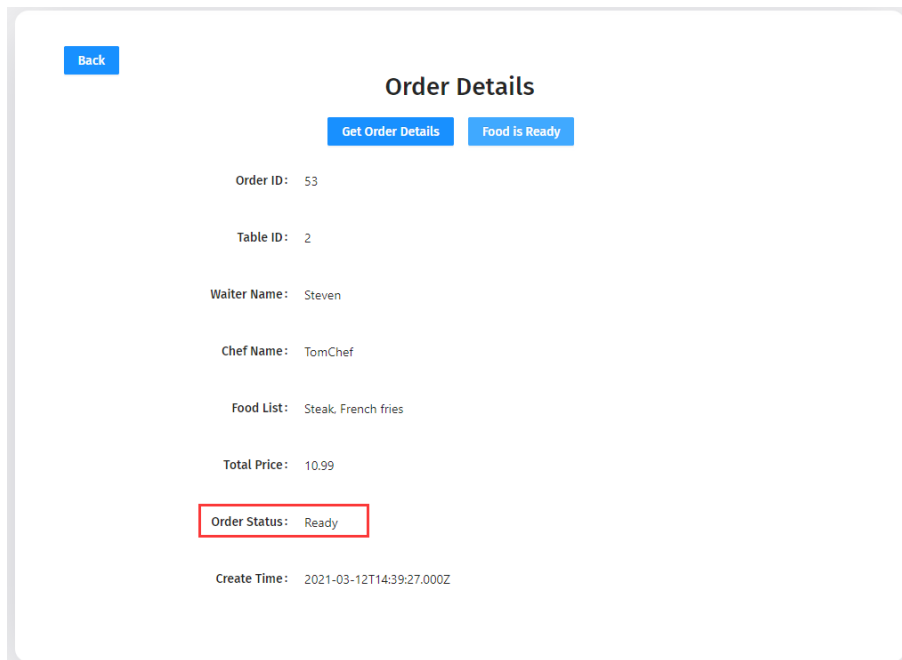


The screenshot shows a web form titled "Order Details". At the top left is a blue "Back" button. Below the title are two blue buttons: "Get Order Details" and "Food is Ready". The form contains the following fields:

- Order ID: 53
- Table ID: 2
- Waiter Name: Steven
- Chef Name: TomChef
- Food List: Steak, French fries
- Total Price: 10.99
- Order Status: Cooking (highlighted with a red box)
- Create Time: 2021-03-12T14:35:39.000Z

Fig. 6, Order Details

Once the chef finishes the cooking, he/she can click the "Food is Ready" button to finish the foods, and the status will change to "Ready":



This screenshot shows the same "Order Details" form as Figure 6, but with the "Order Status" field set to "Ready" (highlighted with a red box). The "Create Time" has updated to 2021-03-12T14:39:27.000Z. All other fields remain the same.

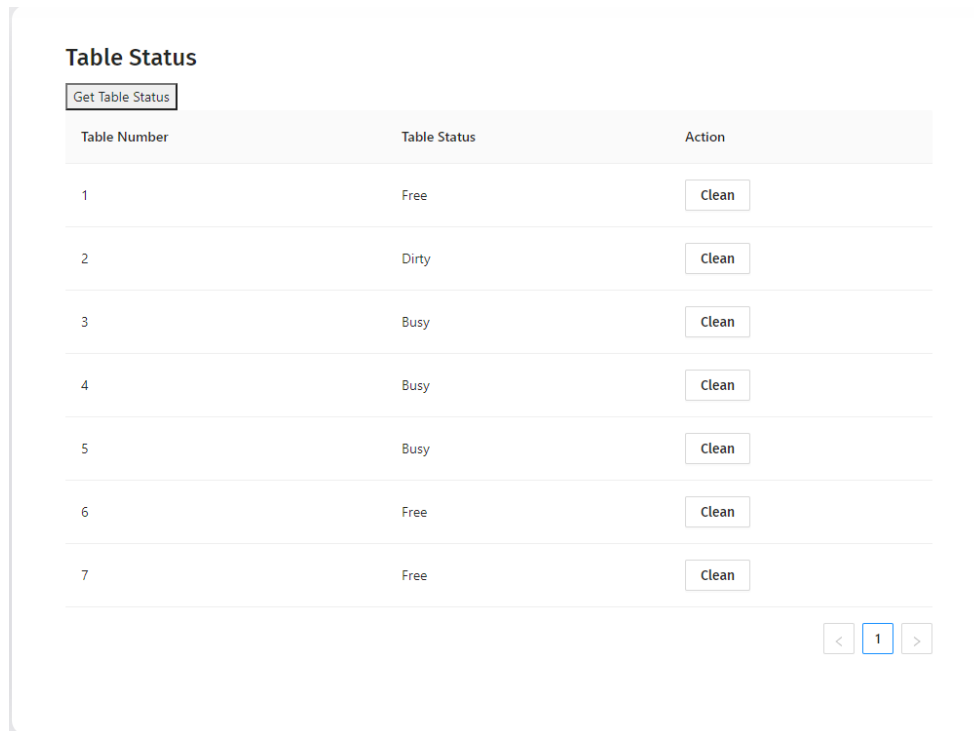
Fig. 7, Order Details

Then, when the waiters notice the foods are ready, they can serve the foods to customers.

### 3.3 Busboy

The busboy will be able to monitor the table clean status using the "table status" portal. The busboy can change the status of the dirty table, after cleaning it, from "dirty" to "Free" using the portal.

"Free" means the table is clean with no customers, "Busy" means the table is occupied by the customers, and "Dirty" means the table is dirty after customers having foods.



<button>Get Table Status</button>		
Table Number	Table Status	Action
1	Free	<button>Clean</button>
2	Dirty	<button>Clean</button>
3	Busy	<button>Clean</button>
4	Busy	<button>Clean</button>
5	Busy	<button>Clean</button>
6	Free	<button>Clean</button>
7	Free	<button>Clean</button>
		<button>&lt;</button> <b>1</b> <button>&gt;</button>

Fig. 7, Table status page

### 3.4 Manager

The manager has the privilege to monitor and edit food menu and track employee information, it can be done by selecting the respective tasks from the menu provided on the manager homepage.

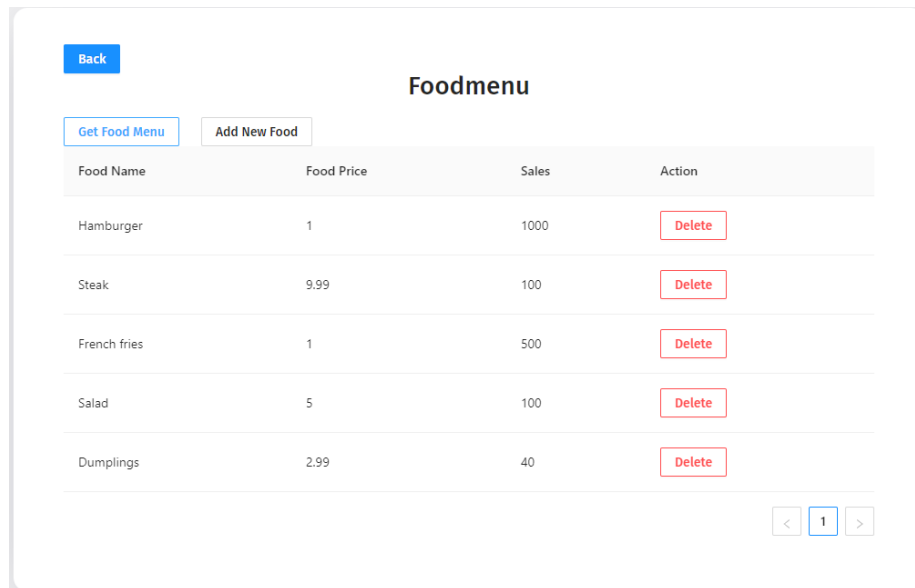


Fig. 8, Food Menu Management page  
Manager can add or delete food on the Food menu page.

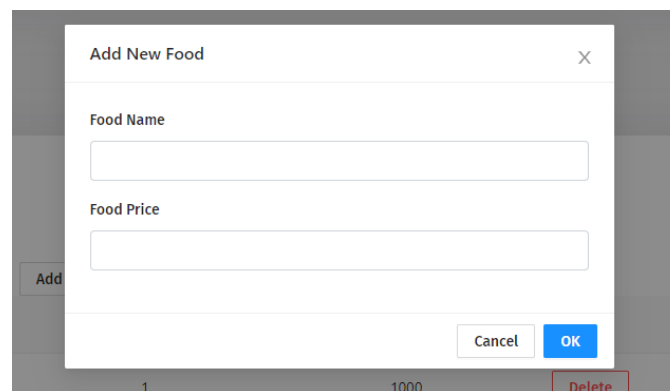


Fig. 9, Add New Food

Manager can see the information of the employees on this page.

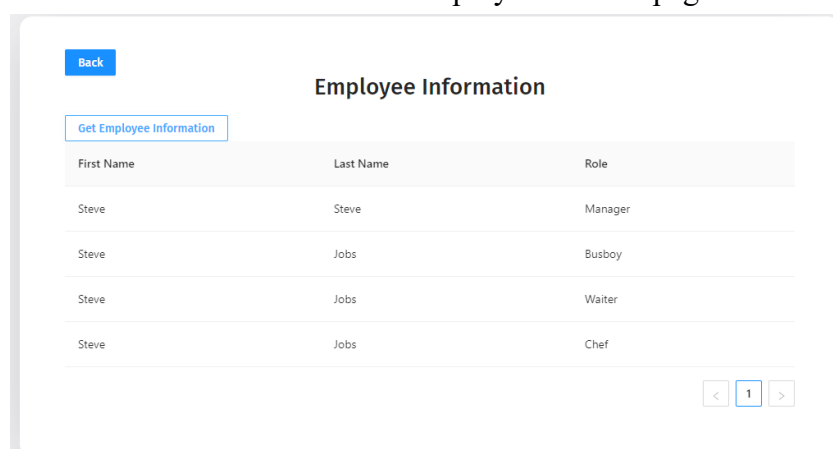


Fig. 10, Employee information



## 4 Notice

Due to time constraints and discoveries in the implementation process, we delete some functions proposed in the requirement document. For instance, we delete the role of host, since we think this role is superfluous. And we delete the authorized employee activities, sales analysis and modify restaurant information functions of the manager.

In order to test the functions, we provide some different roles of accounts:

For waiter functions:

**Username:** SteveWaiters

**Password:** 123456

For chef functions:

**Username:** TomChef

**Password:** 123456

For busboy functions:

**Username:** JamesBusboy

**Password:** 123456

For manager functions:

**Username:** BobManager

**Password:** 123456