# CSC 32200 Team M

# Foodtopia
# Software Requirements Specification
# For Restaurant System

**Version 1.0**

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 10/20/2020 | 1.0 | Version 1 of Software Requirements Specification | Zeal Patel, Bhavesh Shah. Yihui A Wuchen, Greg Kimatov, Xing Yang |
| | | | |
| | | | |

# Table of Contents

# Software Requirements Specification

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to present a detailed description of our online restaurant web application called Foodtopia and provide the development team with a framework to follow. The document will provide an overall description and specific requirements of our application. With use-case diagrams, and client/system diagrams, we will show the user journey and experience on the application. We will define and thoroughly explain terminology and references used in the document.

### 1.2 Scope

Foodtopia is a platform which customers can use to have food be delivered to their home, and a restaurant can use to find customers seamlessly. There are 6 types of users in our system: surfers, regular customers, VIP customers, delivery people, chefs, and managers.

Surfers can browse menu items but are unable to do anything else until they register and their account is approved. Registered customers can order food, review delivery people, chefs, and food quality, and potentially become a VIP customer. In addition to all of the perks of being a customer, VIPs can order food at a 10% discount, their reviews/ratings count twice as much as that of a regular customer, and they have access to special dishes. Meanwhile, the application also provides managers with an interface from which they can manage virtually everything on the application.

The web application will work as follows: a surfer registers to become a regular user with their personal information. Once the manager approves the user, the surfer will become a regular user and be able to make orders. After a transaction is completed, the user can review the delivery person and chef. The delivery person can also review their customers. For a variety of reasons, managers can issue warnings and deregister regular customers or remove the VIP status of VIP customers.

## 1.3 Definitions, Acronyms, and Abbreviations

All of the definitions of terms are referenced from Wikipedia.

| Terms | Definitions |
| --- | --- |
| Chef (Super User) | People who design the dish or menu and cook the dish |
| Delivery people (Super User) | People who deliver the dish to customer |
| Manager (Super User) | The administer of the system |
| Customer (Ordinary User) | People who register the account |
| VIP customer (Ordinary User) | People who register the account and spend more than $500 or place more than 50 orders |
| Surfer (Guest User) | People who doesn't register the account |
| Application Programming Interface (API) | A software intermediary that allow two software communicate with each other |
| Use Case Diagram | It's a diagram that represents the interaction between different users and systems which display the relationship between them. |
| Entity–relationship model (E/R Model) | It's the diagram of the structure of the database. |
| User Interface (UI) | An interface design that enables the interaction between people and computers. |

## 1.4 References

- TypeScript documentation, *typescriptlang.org*
- Material UI documentation, *material-ui.com*
- NodeJS documentation, *nodejs.org*
- Express documentation, *expressjs.com*
- MongooseJS documentation, *mongoosejs.com*
- Sendgrid documentation, *https://sendgrid.com/docs/API_Reference/index.html*

## 1.5 Overview

The rest of this report will be focusing on the various functionalities and requirements our system will provide and need from both a high level and a low level.

To go more in depth, Section 2 will be an analysis of our system from a high-level perspective. We have included a general use-case model which depicts how all 6 types of users (managers, chefs, delivery people, VIP customers, ordinary customers, and surfers) can interact with our system. This use-case model also has an explanation following it that allows for a more detailed understanding of the user-system interactions. Outside of our general use-case mode, Section 2 also contains requirements and assumptions that our system as a whole needs to address.

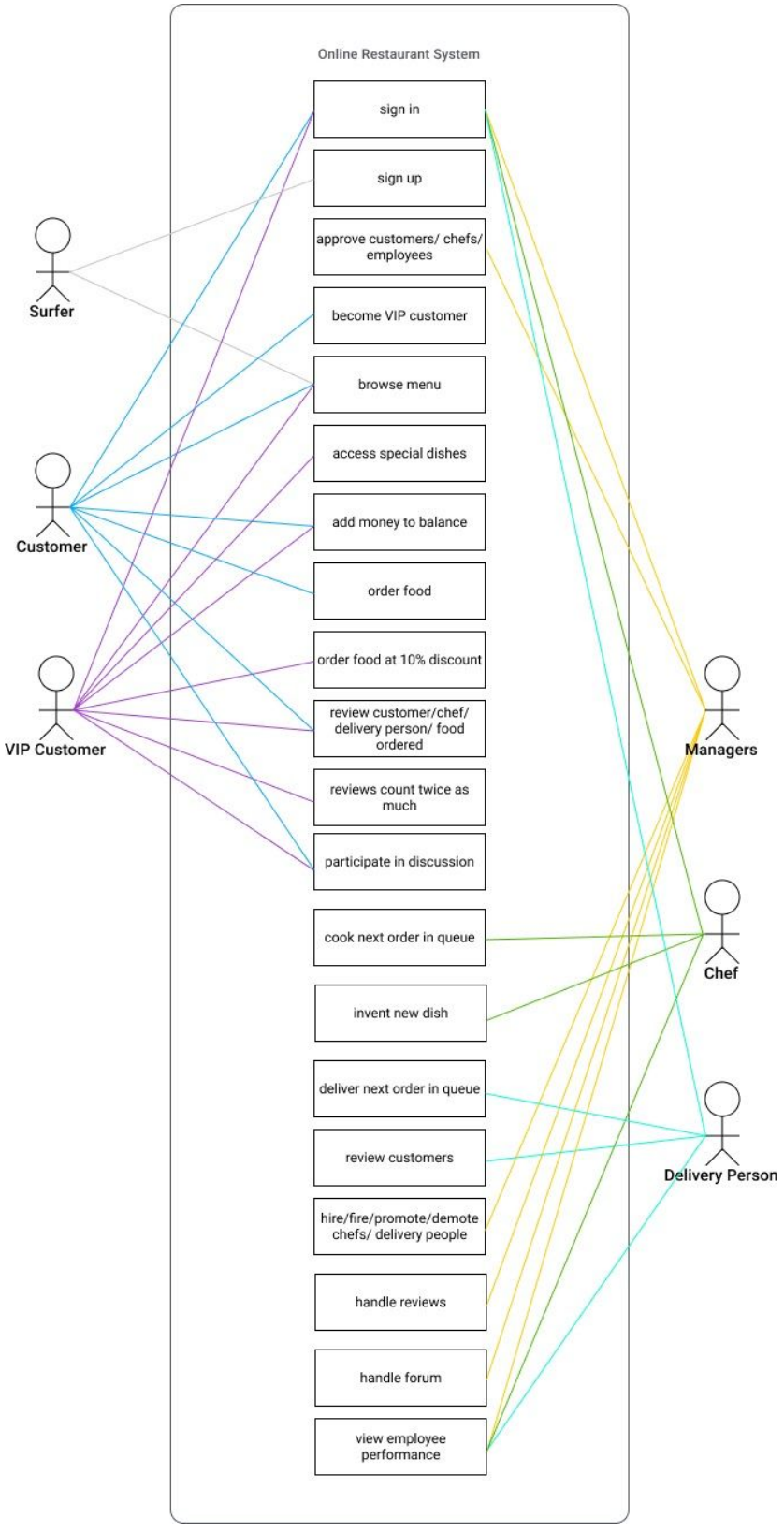
Following this section, Section 3 goes more in depth into the use-case functionalities for each user. Essentially, we have included use-case reports which contain a specific diagram of interactions as well as an explanation for each of the 6 users. Additionally, this part also touches on some of the technical background behind each user functionality. After going into the specifics of use-case functionalities per user, we have also included additional niche requirements our system needs to consider within this third section.

Lastly, Section 4 contains an Index to help make this report easier to follow. We also plan to add mockups of our user interface (UI) to this section in the next iteration of this report. All of these sections work together to make our project more comprehensive and, at the same time, more digestible for our clients.

---

# 2. Overall Description

### 2.1 Use-Case Model Survey


Online Restaurant System
sign in
sign up
approve customers/ chefs/ employees
become VIP customer
browse menu
access special dishes
add money to balance
order food
order food at 10% discount
review customer/chef/ delivery person/ food ordered
reviews count twice as much
participate in discussion
cook next order in queue
invent new dish
deliver next order in queue
review customers
hire/fire/promote/demote chefs/ delivery people
handle reviews
handle forum
view employee performance
Surfer
Customer
VIP Customer
Managers
Chef
Delivery Person

## Surfers's Role in the Application

1. Without logging in, surfers can see the top 3 most popular and top 3 highest rated menu items.
2. Surfers can browse through menu items and ratings only. Once they register with a fixed amount of deposited money and are verified by the manager, they become an ordinary customer.

## Customer's Role in the Application

1. Once a customer is logged in, they are actually able to start ordering food instead of just being able to view the menus. The first thing they see is the top 3 listing dishes, recommended to them based on their previous orders.
2. Customers can leave reviews for chefs, delivery people, and/ or the dish itself.
3. Discussion boards allow customers to participate in discussion topics (related to the restaurant) that they are interested in. If a customer sees another customer using inappropriate language, he/ she can report the other customer to the manager.
4. Customers can add money to their account at any time via the money portal in our system.
5. Additionally, customers can apply for VIP status once they spend over $500 or place more than 50 orders.
6. Like surfers, customers can see the top 3 most-popular and highest-rated dishes. Over time, customers can also view the top 3 "For-you dishes" which our system generates for each user based on their past order history.
7. Lastly customers can edit their account settings, which includes their username, password, email, and address.

## VIP Customer's Role in the Application

1. Similar to customers, once a VIP customer is logged in, their accounts possess all of the functionalities of a regular customer, such as the ability to order food, leave reviews, participate in the discussion forum, view their top 3 "For-you dishes", add money to their account, and edit their account settings.
2. When it comes to ordering dishes, VIP customers receive a 10% discount on their order.
3. As for leaving reviews, VIP customers' reviews count as twice as much as that of a regular customer.
4. Lastly, VIP customers can also get access to special, exclusive dishes that regular customers do not have access to.

## Chef's Role in the Application

1. Once a chef is logged in, their accounts possess all of the functionalities of an employee user, which includes being able to view their performance at any time.
2. In the chef dashboard, each chef is able to independently invent new dishes and decide on the menu for the restaurant.
3. As new orders are made by customers, they are added to the chef's queue of meals to prepare. This can be viewed from the chef dashboard as well.
4. Chefs can view their ratings, reviews, and overall job performance, much like the manager can.
5. Like the customers, chefs can edit their account settings, which includes their username, password, email, and address.

## Delivery Person's Role in the Application

1. Once a delivery person is logged in, their accounts possess all of the functionalities of an employee user, which includes being able to view their performance at any time.
2. As new orders are made by customers, they are added to the delivery person's queue of meals to deliver. This can be viewed from the delivery person dashboard.
3. A delivery person can complain or compliment a customer to whom they delivered dishes.
4. Delivery people can view their ratings, reviews, and overall job performance, much like the manager can.
5. Like the customers, employees like delivery people can edit their account settings, which includes their username, password, email, and address.

## Manager's Role in the Application

1. Once a chef is logged in, their accounts possess all of the functionalities of a manager user, which includes approving employees and users.
2. A manager can view employees performance and based on that they hire, fire, or demote employees.
3. As customers post their reviews which can include complaints, a manager handles them accordingly.
4. If a customer does not follow the terms of use, the manager can take away their VIP status or even deregister them.

## 2.2 Assumptions and Dependencies

Backend Dependencies

- Programming language
    - TypeScript: It is a strict syntactical superset of JavaScript. It allows us to use JavaScript features with the additional functionality of writing types & interfaces for data structures will be used.
- Main dependencies
    - Node: Run environment for building server-side applications. We installed “ts-node” to run Node with TypeScript.
    - Express: Backend framework for building APIs.
    - MongoDB: document-oriented, NoSQL database that uses JSON like documents.
    - JSONwebtoken: Allows to create access tokens for an application.
    - Axios: Promise-based HTTP client to make API requests.
    - Sendgrid: Email delivery service used to send approval emails.
    - BcryptJS: Allows to encrypt passwords before storing them in the database.
    - Body-parser: Allows to parse incoming request bodies in a middleware.
- Developer dependencies
    - types/config, types/express, types/mongoose, types/node: Allows to specify types for TypeScript
    - Jest: Allows to write tests for our code
    - Nodemon: Allows to run our files using node
    - TS-node: Allows to write TypeScript in Node

Frontend Dependencies

- Programming language
    - JavaScript: High level programming language that is generally used when developing web-based applications. It offers functionality such as DOM manipulation, client-side scripting, etc.
- Main dependencies
    - Material-UI: React components that implement Material Design
    - React: JavaScript library for building user interfaces
- Developer dependencies
    - React Testing Library: React DOM testing utilities

## 3. Specific Requirements

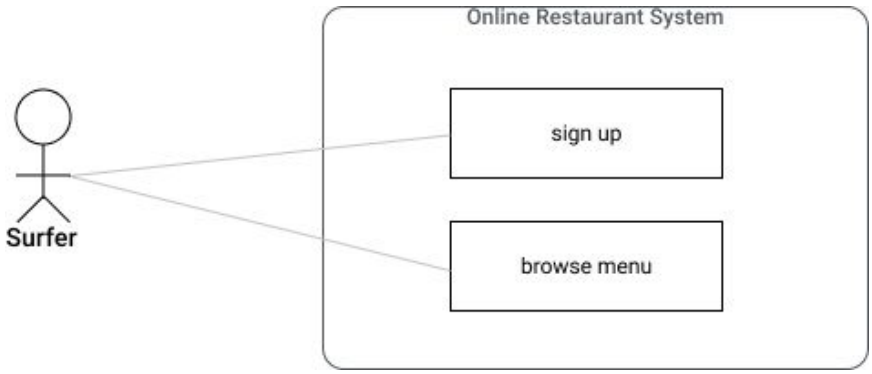
### 3.1 Use-Case Reports

Surfer Use-Cases (Guest Users)

Surfers are guest users on Foodtopia. As these surfers aren't registered, we have limited their functionality to only being able to go through the menu.

1. *Browse Items*
   Surfers can browse through the menus on Foodtopia's home page. They can search for items via the search bar and also toggle certain conditions (such as vegetarian, kosher, halal, etc.) via checkboxes underneath the search bar. Additionally, surfers will be able to see the top 3 most ordered dishes as well as the top 3 highest rated dishes. Lastly, surfers can click an item for more details.

2. *Sign Up*
   Surfers can sign up for an account and deposit some money into their account in order to become a "Customer" and start ordering items. If a surfer tries to order an item without having signed up, our application will redirect him/her to the "Sign Up" page.



## Customer Use-Cases (Ordinary Users)

Customers are ordinary users on Foodtopia. Since they have gone through registration and have been vetted by the manager, we have given them a lot more functionality than surfers such as ordering food and leaving reviews. Customers also have the ability to gain VIP status depending on how much they purchase. VIP status gives customers certain benefits such as ordering food at a discount as well as having their reviews count as double weight.

1. *Login*
   The customers log in to their account to gain access to functionalities only available to customers.

2. *Browse Items*
   Once again, this functionality is very similar to what surfers have. The key difference is that for long-term customers, our application generates top 3 “For-you dishes”, which are personalized to a customer based on their past order history.

3. *Order Items*
   The Customers browse through the item lists and add several orders into the shopping cart. Then, they choose three ways to receive the orders: stay in the restaurant, pick up the orders from the restaurant, or have the order delivered to their house. To stay in the restaurant, the customer needs to pick the available time and seat. To have the order delivered to their house, they will need to pick a delivery person. If the cost of order exceeds the amount of money in the account, the account will freeze and wait until a new deposit.

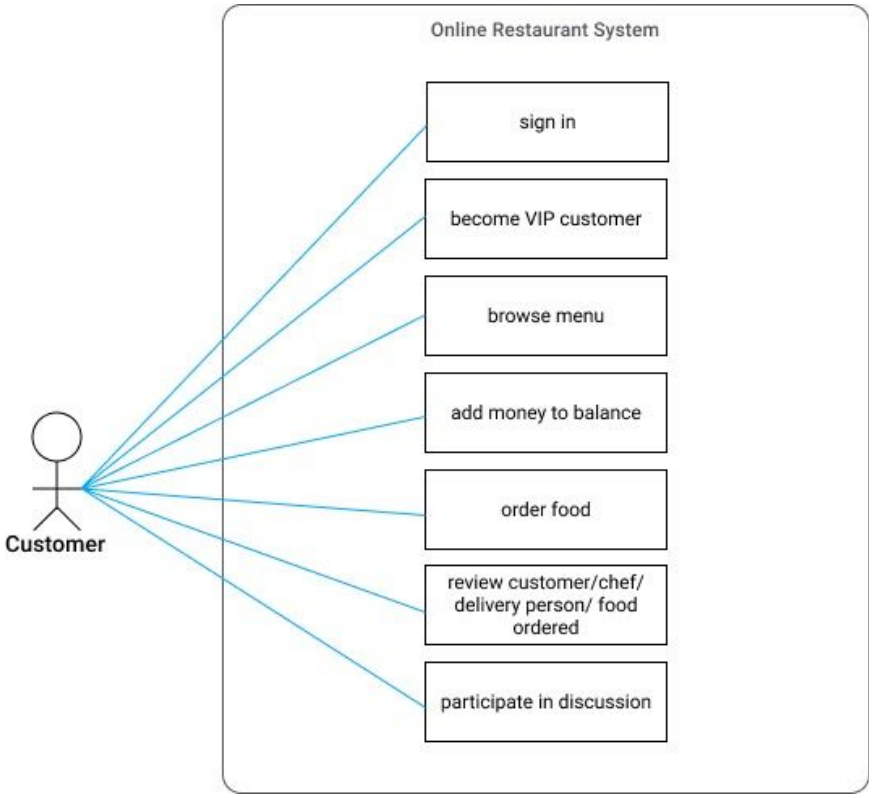
4. *Post to Discussion Forums*
   Customers can start discussion threads or participate in previous threads. Threads could be about the food items, the delivery people, as well as the chefs.

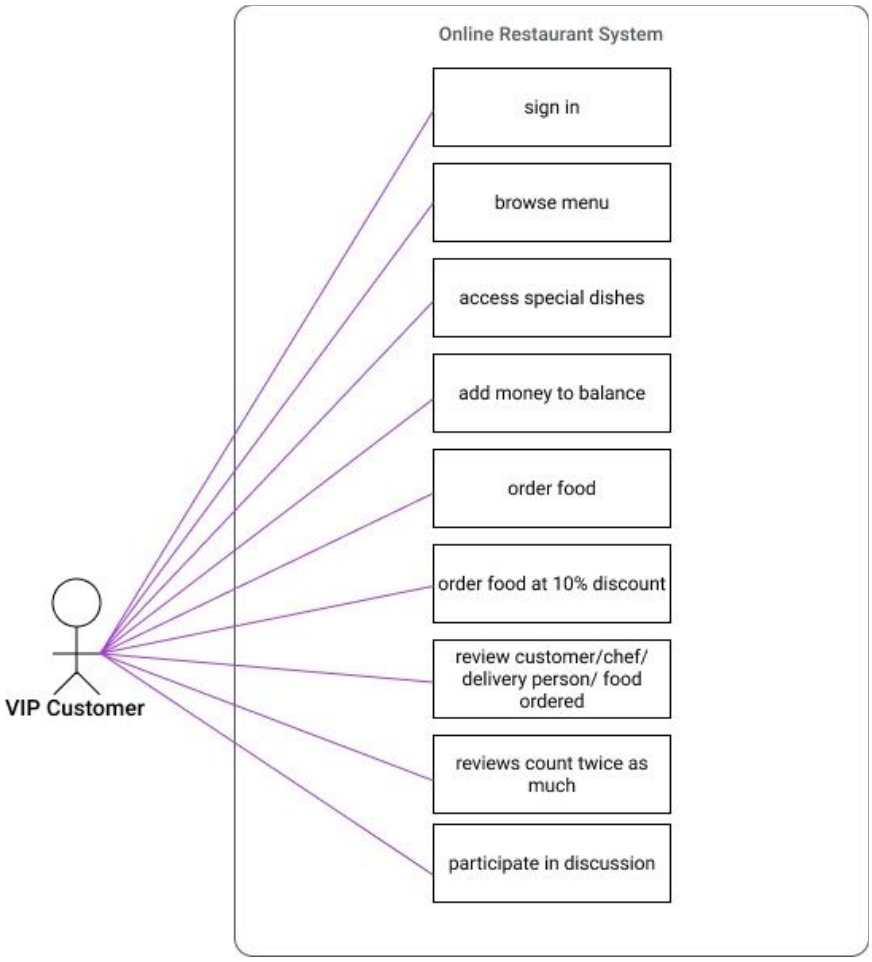
5. *Leave Reviews*
   Customers can leave reviews for food items, the delivery people, as well as the chefs. Additionally, customers can also file a review for a customer that may have posted something inappropriate in the discussion forum. If a customer has earned VIP status, then his/ her review will count as twice as much of that of a regular customer.

6. *Upgrade to VIP*
   Customers can apply for VIP status once they spend $500 or place more than 50 orders. If these conditions are met, they will be granted benefits such as a 10% discount on orders, access to exclusive dishes, as well as their reviews counting twice as much as regular customers.

7. *Deposit Money*
   Customers can visit the money portal on our application where they can deposit more money into their Foodtopia account.

8. *Edit Account Setting*s
   Customers can edit their account settings, such as username, password, email, address, etc. Additionally, users will also have the ability to delete their account on this page. In that case, we would also refund the customer's deposited money back to them.






## Employees Use-Cases (Super Users)

Employees are also superusers on Foodtopia. Unlike managers, employees are more concerned with day-to-day operations rather than managing the entire restaurant. For this system, we have 2 types of employees: chefs and delivery people. Chefs are responsible for cooking the next order in the order queue, and also have the power to invent new dishes. Delivery people, on the other hand, can deliver the next order in the order queue, as well as review the customers. Both chefs and delivery people share some common functionalities such as the ability to view their performance.

1. *Login (Both)*
   The employees log in to their account to gain access to functionalities only available to employees.

2. *Invent New Dish (Chefs)*
   Chefs have the freedom to invent new dishes, where they can upload an image of dish, list all the ingredients and list dietary restrictions.

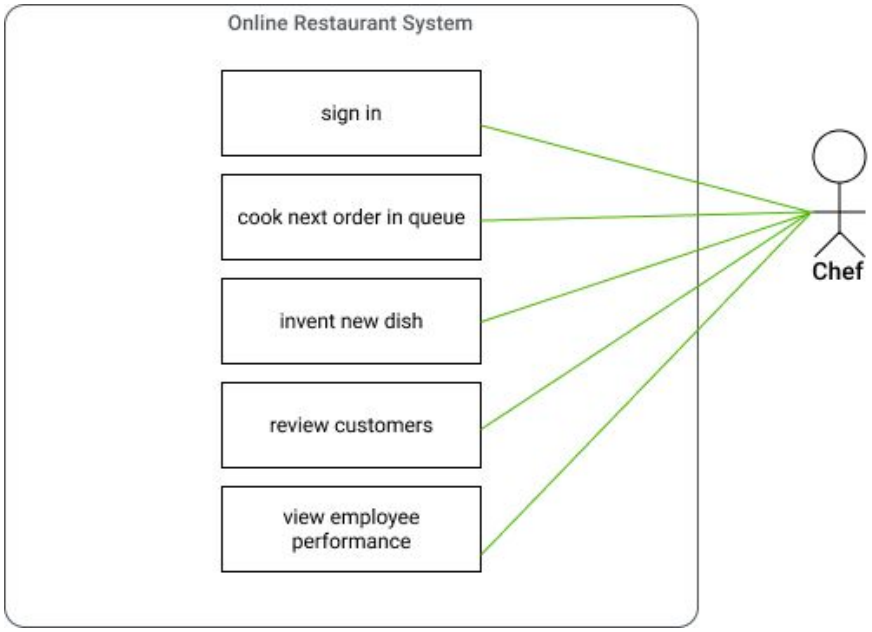
3. *Cook Next Order in Queue (Chefs)*
   Chefs can view the orders they have to cook in the order they come in and then process them accordingly.

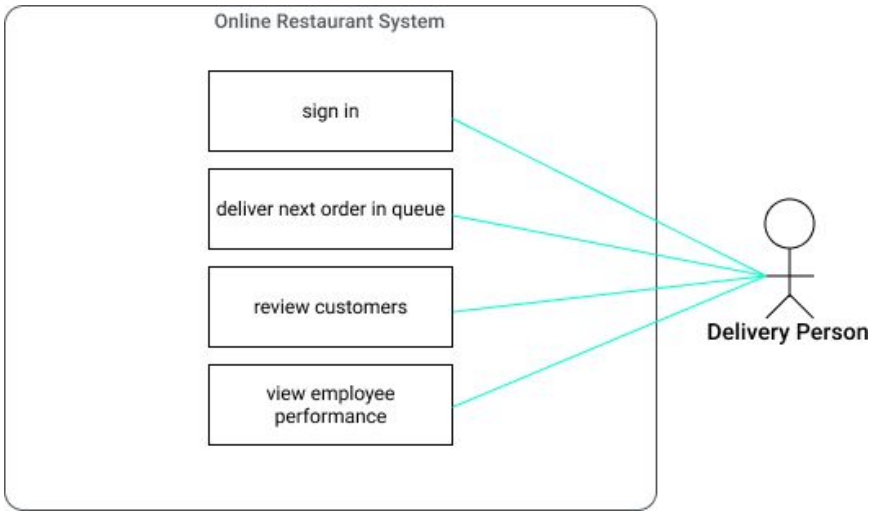
4. *Deliver Next Order in Queue (Delivery People)*
   Delivery people can view the orders they have to deliver in the order they come in and then process them accordingly.

5. *Leave Review (Delivery People)*
   Delivery people can leave a review on the customers they delivered the food to.

6. *View Performance (Both)*
   Both chefs and delivery people will be able to see how they are getting rated. Additionally, chefs can see if they have gotten any orders in the past 3 days or if they are consistently getting low reviews on their recipes.

7. *Edit Account Settings (Both)*
   Customers can edit their account settings, such as username, password, email, address, etc. Additionally, users will also have the ability to delete their account on this page.






## Manager Use-Cases (Super Users)

Managers are superusers on Foodtopia. This gives them administrative control of a large percentage of our application. For example, they have the ability to approve customers/ employees that sign up for an account, hire/fire/promote/demote employees, flag reviews/ discussion-posts, as well as suspend customer accounts.

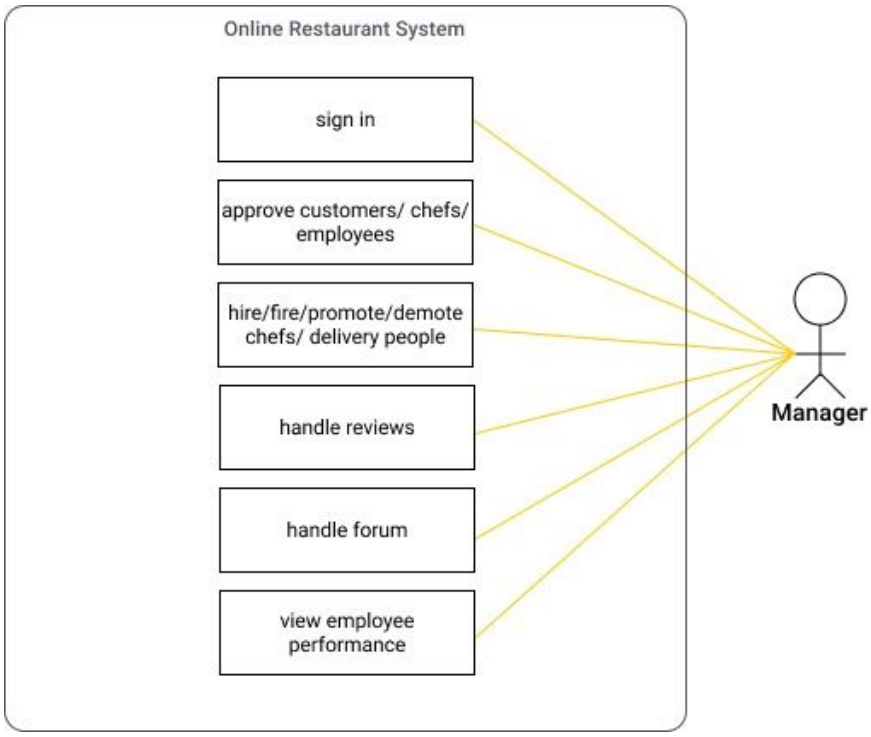
1. *Login*
   The manager logs in to their account to gain access to functionalities only available to managers.

2. *Process Customer Registrations*
   The manager has the right to approve or deny customers who sign up for an account on Foodtopia.

3. *Handle Reviews*
   The manager can handle the review from the customer or delivery people and decide to approve the review or dismiss the review (which would send a warning to the person that initiated the review) depending on whether the review is genuine and does not contain inappropriate words. If a customer receives 3 warnings, his/ her account would be closed, and their money would be returned.

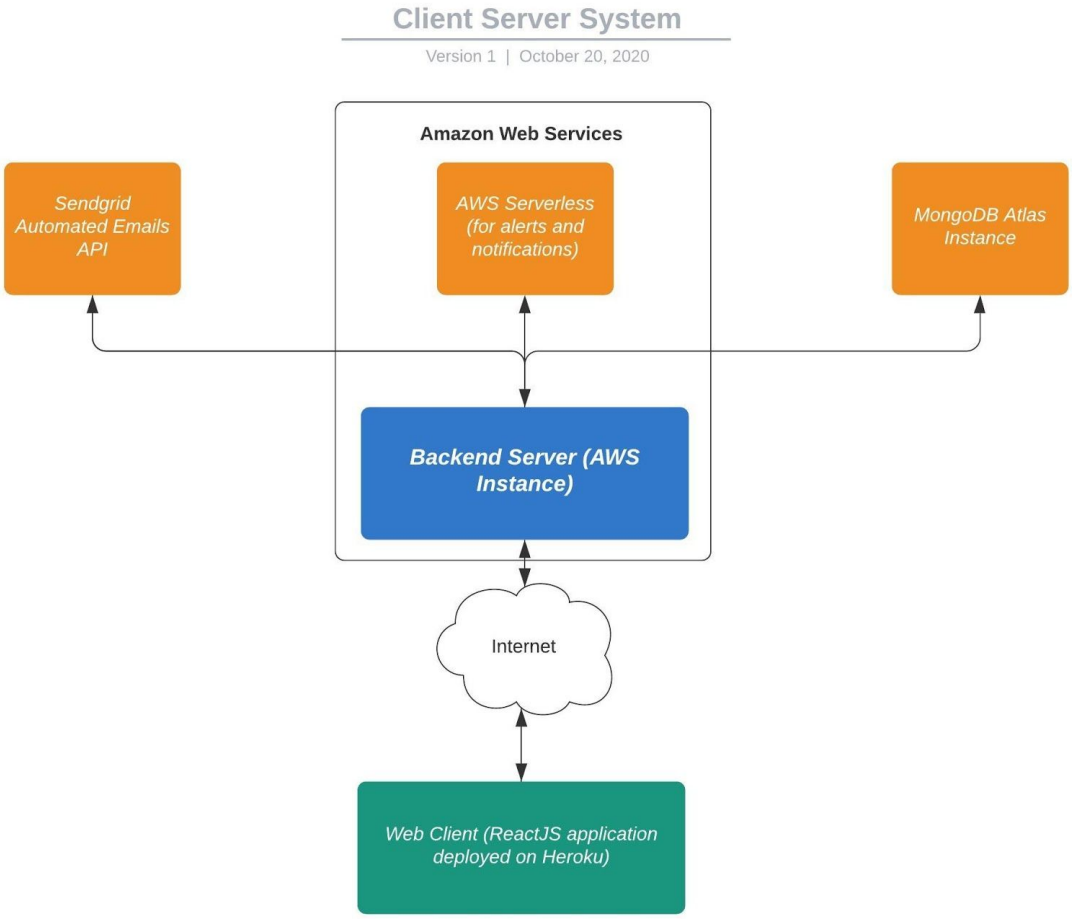
4. *Handle Discussion Forum*
   Similarly to handling reviews, the manager could remove a customer when they either receive three warnings or decide to close their account. This action will lead to clearing the deposit and closing the account.

5. *Hire/Fire/Raise/Cut pay employee*
   The manager could choose the employees to hire after employees have gone through registration. Additionally, the manager could review the profile of employees (chefs and delivery people) and take action based on their performance. If employees have poor performance, then managers can cut their salary. If their salary is cut twice, managers can fire employees. However, if employees are performing well, managers have the power to raise their salary.




### 3.2 Supplementary Requirements

For our online restaurant system to work efficiently and flawlessly, there are a few supplementary requirements that need to be taken into consideration, particularly efficiency, security, and scalability. This section will cover these requirements in detail. We have also included a snapshot of version 1 of our Client-Server system as well to better illustrate how we will be handling these 3 requirements.




1. *Efficiency*
   It is crucial for our restaurant system to be efficient. To handle this, we have planned to employ a couple of technical strategies in our system. For example, we have planned to make HTTP requests to our backend server asynchronous. This will allow customers to continue using the system while requests are being dealt with under the hood. Additionally, we are also planning on utilizing caching so that we don't have to initiate as many HTTP requests. For example, for fairly invariable data like menu items, this can be cached in the frontend client.

2. *Security*

Since our system requires users to create accounts and provide their credit card account information, we need to ensure that our system is very secure. In order to do this, we will be maintaining user sessions in the web client using JSONWebTokens (JWT). That way if a customer's device is ever hacked into or stolen, the customer's account will require the attacker/ thief to re-log to our system since the customer session token expired after an hour. Additionally, we will be encrypting user's passwords with the help of Bcrypt in our backend server.

3. *Scalable System*

   We are expecting that our restaurant application's user base will grow very quickly. Thus it is important that our system is scalable. As seen in our client-server system above, we are utilizing a microservice architecture. While we have one central backend server, we also have abstracted away multiple functionalities into other microservices that connect with our central backend. For example, for automated emailing functionality, we have connected our central backend to a 3rd party API (Sendgrid's email delivery system). For additional functionalities that we want to add, our system is thus vertically scalable, since we can essentially separate this functionality into another microservice that connects to our central backend.

## 4. Supporting Information

## 4.1 Index