MBJ Co.

ChefNODE
Software Requirements Specification
For Web Application

Version 1.1

Revision History

Date	Version	Description	Author
24/3/2017	1.0	Preliminary stages	Mohamed Sondo Jake Biller Benjamin Yi
24/4/2017	1.1	Revised use-case diagrams	Mohamed Sondo Jake Biller Benjamin Yi

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

This documentation provides a detailed description of the ChefNODE restaurant ordering system and analyzes the different subsystems within this web application. As it is a web application, this document explains the features of the interface available to the users, how the users will interact with the application, and how the application will respond to these interactions. This document will also describe the reliability of its dependencies such as APIs and libraries published by third parties.

1.2 Scope

ChefNODE is a food delivery system which allows individuals to order food from a chosen restaurant. Customers, and potential customers, are given a window view to restaurants, allowing them to view menus of a restaurant as well as the chef(s) preparing the menus. Upon receiving the meal, customers can review their experience and rate their food.

Moreover, ChefNODE gives certain responsibilities to users besides customers – the employees of the restaurant: the manager, chefs, and delivery men. These responsibilities include such things as changing the items on the menu (chefs), the hiring and firing of employees (managers), and choosing a delivery route (delivery).

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
Manager (MG)	A restaurant personnel – has access to a control panel which handles customer registrations and work status of its employees (hire/fire/promote).
Chef (CH)	A restaurant personnel – can set up a menu.
Delivery Personnel (DP)	A user who can access and view orders placed, and plan a route of delivery of the order.
Registered Customer (RC)	A registered customer who can also browse/search for restaurants and view the restaurant's details. Furthermore, RC can place an order and provide feedback for the restaurant.
Non-registered Customer (NC)	A visiting user who can only browse/search for restaurants and view the restaurant's details and menu.
System (SY)	The system provides checkpoints for specific actions of the system's users.
Manager Control Panel (MG-CP)	A control panel specific to a manager user.
Chef Control Panel (CH-CP)	A control panel specific to a chef user.
Node.js	An asynchronous, event-driven JavaScript runtime built to build scalable network applications.
Express.js	A web application framework for Node.js designed to build web applications and APIs.
React.js	A JavaScript framework for building and manipulating user interfaces.
MongoDB	A NoSQL database program designed to use a JSON-like format (or a hash format) to store data.
Passport	An authentication middleware for Node.js which allows applications to have easy registration through social media applications such as Facebook and Google+.

Materialize	A CSS framework utilizing Material Design.
MERN	A scaffolding tool allowing for easy-to-build applications using Mongo, Express, React, and Node.
Google Maps API	An API that allows developers to embed maps and locate specific places such as restaurants.
Big Oven API	An API that allows calls to a very large database of food items and recipes.

1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. *IEEE Computer Society, 1998*.
"Node.Js V7.7.4 Documentation". Nodejs.org. N.p., 2017.

1.5 Overview

The remainder of this specification will detail the functionality of this web application

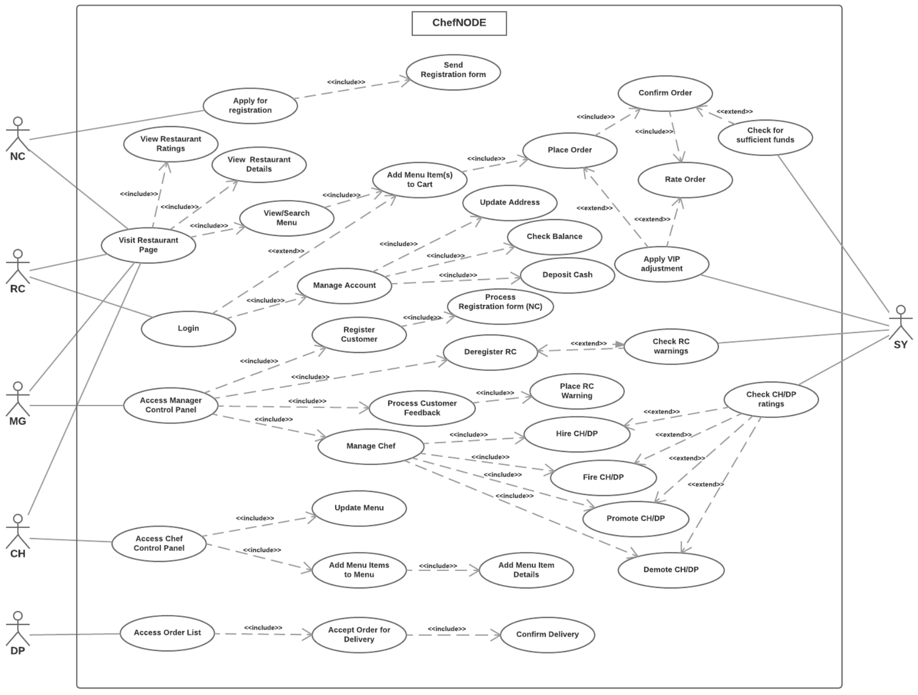
The second, following section will provide a brief description of the application by reviewing use-cases and dependencies.

The third section provides some better details of the applications not included in it's prior sections and uses more specific definitions to describe different use-cases and dependencies of this application.

2. Overall Description

2.1 Use-Case Model Survey

The following diagram provides a visual to available features of the application unique to each user.



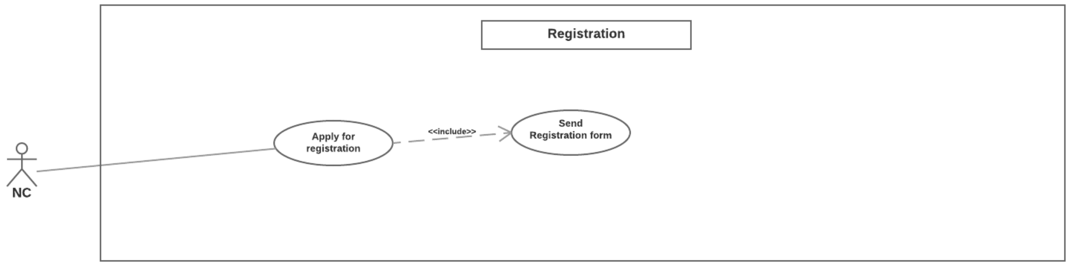
2.2 Assumptions and Dependencies

Assuming each user is able to use a web browser and to navigate a web application, this application relies heavily on the upkeep of its dependencies, mainly a working web and database host, a working map API, and a compatible browser. It is assumed that the hosts are always available and that the application is kept updated to stay compatible with most recent libraries, APIs, and browsers.

3. Specific Requirements

3.1 Use-Case Reports

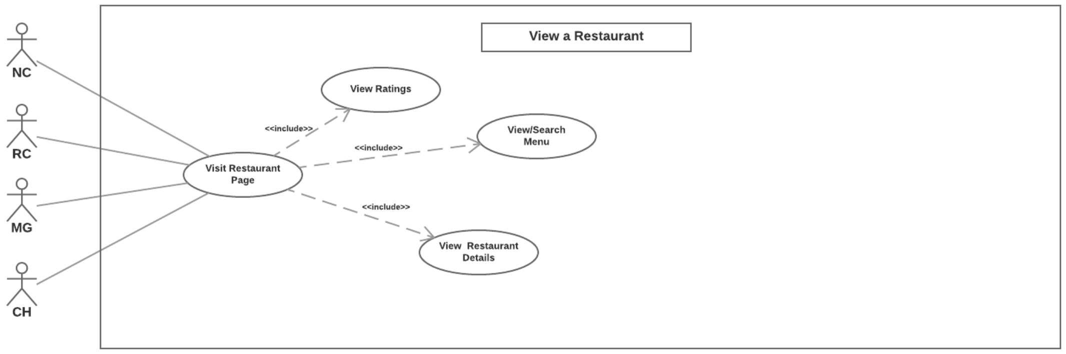
Use-Case: Registration



Description: Assuming the system is displayed properly to a visiting user, a non-registered customer (NC) can view a restaurant and its details. But, to place an order, the NC must first submit a registration form to become a registered customer (RC). Once an RC, the user gains multiple access controls for the system.
Details:

1. NC enters the “sign-up” page
2. NC enters valid information on the sign-up form
3. NC submits/sends the form and waits for approval
4. MG views the form and approves, thus registering the customer
5. SY creates an account for NC, which then becomes RC

Use-Case: View a restaurant page

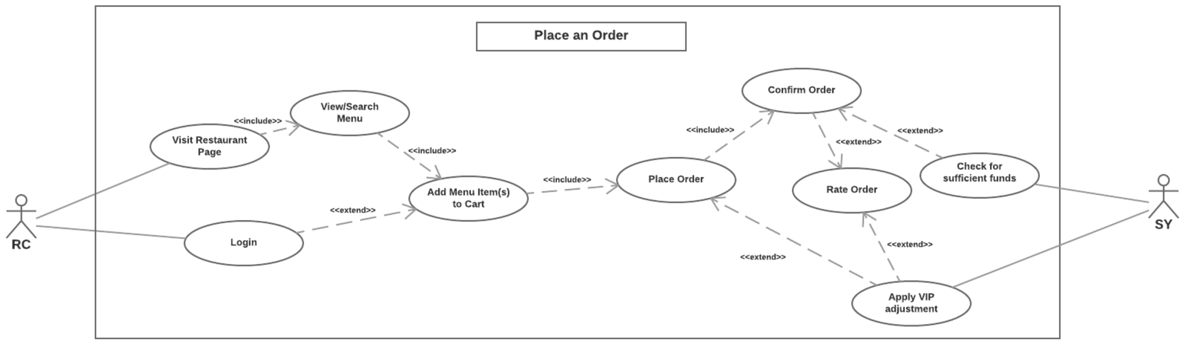


Description: All users will be allowed to view a restaurant page, which includes the ratings, menus, and additional details of the restaurant.
Details:

1. User connects to the website
2. User browses restaurants and chooses a restaurant
3. User views the restaurant’s page
4. User views ratings

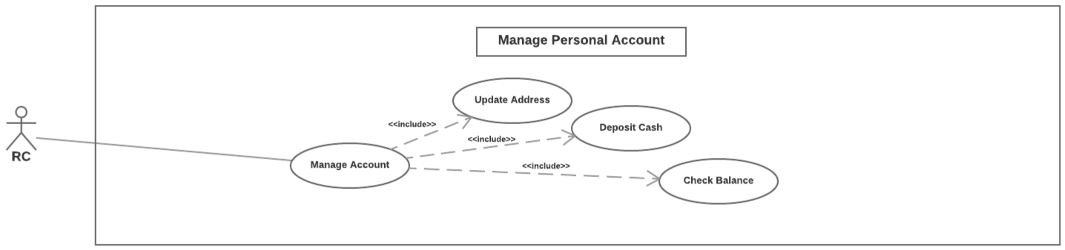
   a. Ratings will be an average of feedback received by RCs who ordered from current restaurant.
5. User views menu
   a. The menu will present the top five most popular dishes at the top of the menu
6. User views details
   a. View vicinity and location of current restaurant
   b. View hours of operation of current restaurant
   c. View CH operating at the current restaurant

Use-Case: Place an order



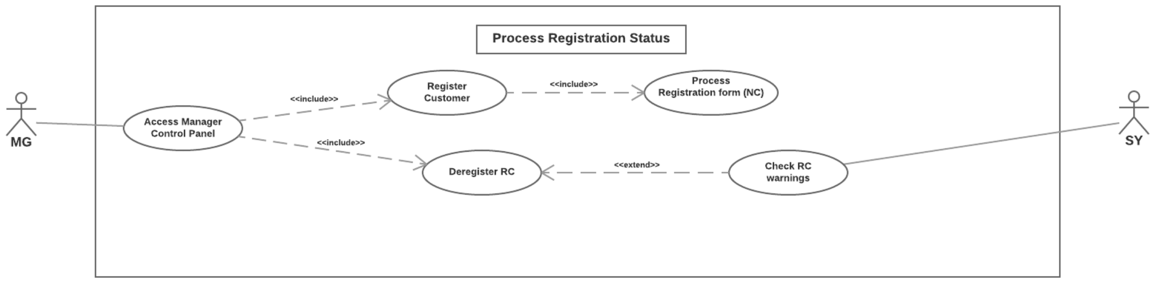
Description: Once registered, a user gains an online cart to add menu items from a menu of a restaurant. Once the user has desirable items in the cart, the user can place the order.
Details:
1. User finds a restaurant and entered the restaurant's page
2. User views and searches the menu of the restaurant
3. User cannot proceed further unless the user is logged in as RC
4. RC adds menu items to the cart
5. RC places the order
6. SY checks the RC for sufficient funds, and adjusts total
7. Order is confirmed
8. RC options to rate the order, and SY applies an adjustment to the rating

Use-case: Manage personal account



Description: A RC must manage the RC account to have a valid address and enough balance to place an order.
Details:
1. RC visits account settings page
2. RC edits and saves address details
3. RC transfers or deposits cash into the account
4. RC can check the current balance

Use-case: Process registration status



Description: MG is granted special access through the MG-CP (manager control panel). This control panel allows MG to manage the system's users.

Details:

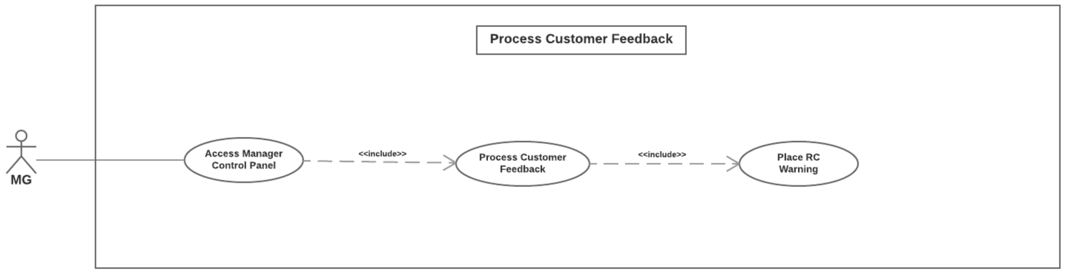
Registration process:

1. User logs in as MG
2. MG views submitted forms
3. MG reviews a specific form
4. MG accepts form and a RC account is created

De-registration process:

1. User logs in as MG
2. MG views RC and prepared to de-register
3. SY checks RC account for a specific number of warnings
4. SY allows de-registration, and RC account is destroyed

Use-case: Process customer feedback


Process Customer Feedback
MG
Access Manager Control Panel
Process Customer Feedback
Place RC Warning
<<include>>
<<include>>

Description: MG is granted special access through the MG-CP (manager control panel). This control panel allows MG to process feedback given by RC.

Details:

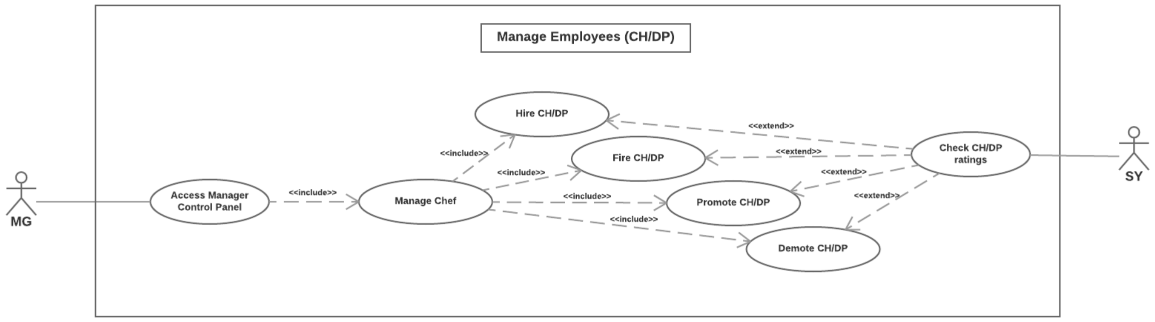
Accepting a comment

1. User logs in as MG
2. MG accesses MG-CP
3. MG views unprocessed feedback provided by RC
4. MG confirms and accepts feedback
5. SY updates the average rating of the restaurant or menu item

Placing a warning

1. User logs in as MG
2. MG accesses MG-CP
3. MG views unprocessed feedback provided by RC
4. MG flags RC for inappropriate feedback

Use-case: Manage employees (chefs and delivery personnel)



Description: MG is granted special access through the MG-CP (manager control panel). This control panel allows MG to manage the employees of the restaurant (CH and DP).
Details:

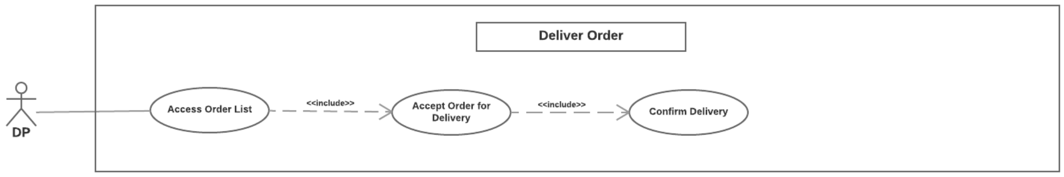
Hiring
1. User logs in as MG
2. MG accesses MG-CP
3. MG views CH or DP operating at current restaurant
4. MG hires CH or DP

Firing
1. Users logs in as MG
2. MG accesses MG-CP
3. MG views a CH or a DP
4. MG prepares to fire CH or DP
5. SY checks for ratings of CH or DP and approves of fire
6. CH or DP is de-registered from current restaurant

Promoting
1. User logs in as MG
2. MG accesses MG-CP
3. MG views a CH or a DP
4. MG prepares to promote CH or DP
5. SY checks for ratings of CH or DP and approves the promotion
6. CH or DP is promoted
7. SY adjusts salary of selected CH or DP

Promoting
1. User logs in as MG
2. MG accesses MG-CP
3. MG views a CH or a DP
4. MG prepares to promote CH or DP
5. SY checks for ratings of CH or DP and approves the demotion
6. CH or DP is demoted
7. SY adjusts salary of selected CH or DP

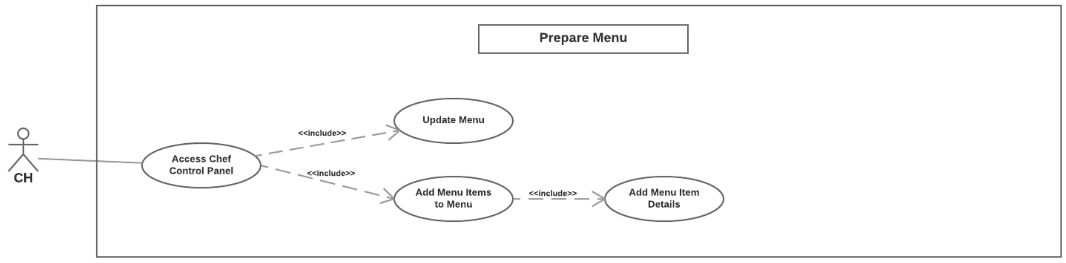
Use-case: Deliver order



Description: Delivery personnel (DP) has access to a list of orders placed by RC and proceeds to delivery to the RC location.
Details:

1. User logs in as DP
2. DP views list of orders for current restaurant
3. DP accepts an order to deliver
4. DP delivers and confirms that the order has been delivered

Use-case: Prepare menu



Description: Chefs (CH) have access to a special chef control panel (CH-CP), which allows the CH to alter the menu.
Details:

1. User logs in as CH
2. CH accesses CH-CP
3. CH views menu of current restaurant
4. CH adds or removes menu items
5. CH updates or saves the new menu

3.2 Supplementary Requirements

In order to deploy our system, we need to provide a reliable, safe and efficient system for the operations within the ChefNODE system by fulfilling the following requirements:

- works properly as required and as described.
- each use case should be implemented following the user title and relation to other users.
- handles ChefNODE system should handle login and registration process as well as order processing by customers. Specific requests should be handled accordingly without risk of system crashes.
- is well-secured and provides protection for user credentials and authentications and blocks users from passing by the protocol required in the description.
- provides an easy to use platform so that any user with rspect to their title can use the system with their full functionality. In addition, the system should not only be user-friendly but also be scalable and support extra features.
- can maintain a unique style of communication methods and allow users to get easily familiar with the application's usage
- can sustain usability through time

4. Supporting Information