

# Software Requirements Specification

Version 1.0

September 15, 2021

Restaurant Ordering System

Harsh Saglani

[hsaglani@umd.edu](mailto:hsaglani@umd.edu)

Submitted in partial fulfillment  
of the requirements of  
ENPM809W - Project Phase 1

## Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>4</b>
<b>1. Introduction</b>	<b>5</b>
1.1. Purpose	5
1.2. Scope of the Project	5
1.3. Intended audience	6
1.4. References	6
1.5. Glossary	6
1.6. Overview	7
<b>2. Overall Description</b>	<b>8</b>
2.1. System Environment	8
Figure 1 - System Environment	8
2.2. Functional Requirements Specification	8
2.2.1 User's use case	8
2.2.2 Chef's use case	13
2.2.3 Admin's use case	15
2.3. User Characteristics	18
2.4. Non-Functional Requirements	18
<b>3. Requirements Specification</b>	<b>19</b>
3.1. External Interface Requirements	19
3.2. Functional Requirements	19
3.2.1 Search Items	19
3.2.2 Food Display	19
3.2.3 View Cart	20
3.2.4 View Order	20
3.2.5 Rate Food	21
3.2.6 Login	21
3.2.7 Add chef	22
3.2.8 Change Password	22
3.3. Detailed Non-Functional Requirements	23
3.3.1 Logical Structure of the Data	23
Figure 2 - Logical Structure of the Data	23
3.3.2 Security	25
3.4. Misuse Cases	26

3.4.1 Misuse case: Delete Reviews	26
3.4.2 Misuse case: Add Malicious Chef Account	26
3.4.3 Misuse case: Delete actual chef's account	27
3.4.4 Misuse case: Change order status of a customer's order	27
3.4.5 Misuse case: Delete available food items	28
3.4.6 Misuse case: Anonymous Order	29
<b>4. Threat Identification &amp; Modeling</b>	<b>29</b>
4.1. Threat: Weak intranet password	29
4.2. Threat: Weak admin / chef login password	30
4.3. Threat: Physical access to devices that should be used inside the restaurant.	30
4.4. Threat: Man in the middle	30
4.5. Threat: Loss of Data	31

## List of Figures

1. Figure 1 - System Environment	8
2. Figure 2 - Logical Structure of the Data	23

# **1. Introduction**

## **1.1. Purpose**

The purpose of this document is to present a detailed description of the Restaurant Ordering System to allow for software design to proceed with a perceptive of the design that is to be structured and how the process of it develops. The topics of, general description of design elements and their interactions, how the system will be structured, data & functional structure are to be further discussed in order to help producing test cases, and help in maintenance services, and also satisfy requirements, design details indicated in the SRS document.

## **1.2. Scope of the Project**

This project will be a web application software for ordering and reviewing food items of a restaurant. The system is designed to be a quick paced interface and help the customer decide and order the food for their meal. The ordering system will be completely automatic with almost zero interaction between the restaurant staff and the customers. The web application will help other customers in ordering food items from the restaurant by viewing previous customer ratings and reviews.

More specifically, the system is designed to allow the customer to see the restaurant's menu on a device (such as a tablet), look at previous customer ratings and reviews, order and review food items from the menu. The customer will be asked for some identification such as a phone number or an email address after which they'll be prompted with the screen where they can rate & review the food items they've ordered. On the other hand, the restaurant can add food items to be ordered and also manage

the ratings and reviews. The chefs in the restaurants will be able to see the ratings, reviews and suggestions if any.

### **1.3. Intended audience**

Intended audience of software design description is all stakeholders which includes people / customers who will visit the restaurant using the software, the restaurant staff, development team and testers.

### **1.4. References**

[1] IEEE. IEEE Std 1016-2009 IEEE Standard for Information Technology – System Design – Software Design Descriptions. IEEE Computer Society, 2009

### **1.5. Glossary**

SRS	Software Requirement Specification
User/Customer	Person who wants to or will use the system
DB	Database
Admin	Administrator - A restaurant staff who will be in charge of adding and managing food items as well as managing ratings and reviews given by users.
Info	Information
Chef	An individual working in the restaurant who will make food and/or update the order status
Attacker	An individual or organization who wants to harm the system in any shape or form
Item / Food Item	A food item which will be part of the menu of the restaurant
Cart	A cart is a virtual shopping cart which will hold items which the user would like to order.

## **1.6. Overview**

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

In the fourth chapter, Threat Identification and Modeling section, of this document is written primarily for the developers and the testers. It describes the possible threats to the system, how they are dangerous and possible steps to mitigate them.

## 2. Overall Description

### 2.1. System Environment

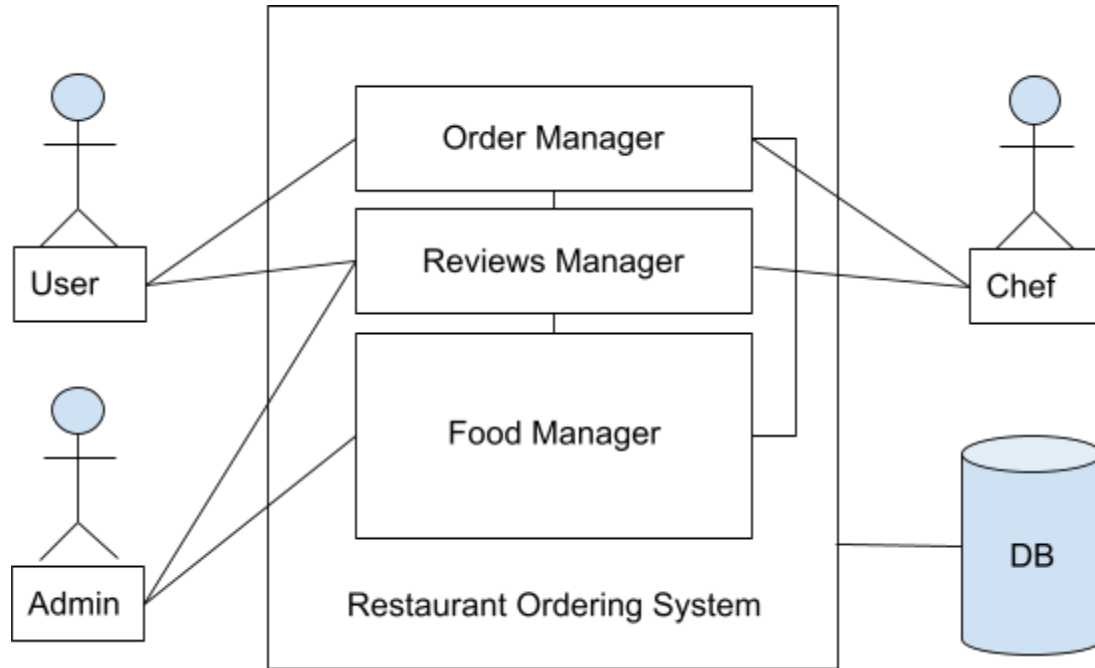


Figure 1 - System Environment

The Restaurant Ordering System has three actors - the user, the chef and the admin. The user will be able to place orders and the chef will be able to see all the orders which are placed. The user will be able to rate & review a food item it ordered and the chef will be able to see the rating as well as review the food item received. The users can also leave a suggestion if they want for the chef to see. The Admin will be able to manage the reviews as well as add food items to the menu.

### 2.2. Functional Requirements Specification

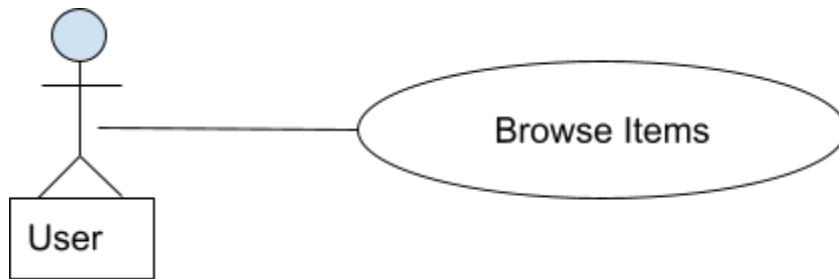
This section outlines the use cases for each of the actors separately.

#### 2.2.1 User's use case

Use case: **Browse Items**



**Diagram:**



**Brief Description:**

The user can see a list of food items with their pictures and some other information.

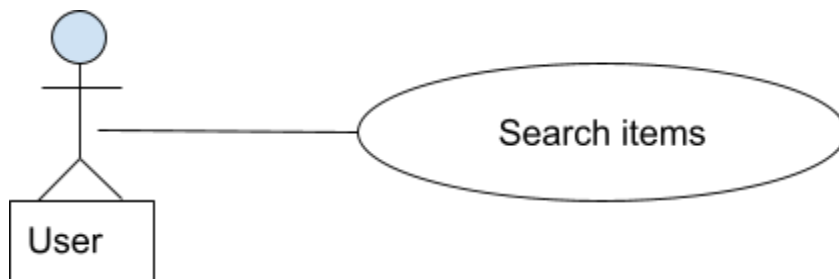
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's home page.

1. The system displays a list of food items with their pictures, rating and price
2. The user browses through that list
3. The user can choose to add any item to cart from that list

**Use case: Search items**

**Diagram:**



**Brief Description:**

The user can search for a particular item.

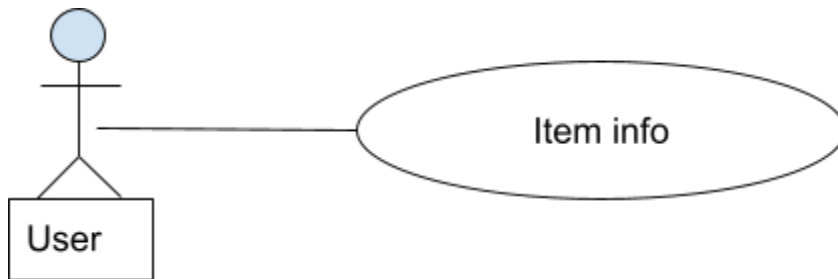
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's home page.

1. The user chooses to search for a particular item
2. The system searches for the item in its database
3. The system displays results of items according to user's search criteria

Use case: **Item information**

**Diagram:**



**Brief Description:**

The user can view a food item's information.

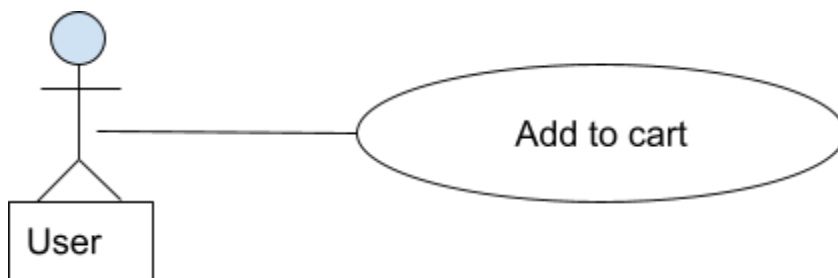
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's home page.

1. The user clicks on an item from the list shown on the home page
2. The system fetches information about the item from the DB
3. The system displays the item info page which shows the information on the selected food item. The information includes pictures, user ratings, reviews, price of the item, name of the item, etc.
4. The user can choose to add this item to the cart if they'd like to

Use case: **Adding to Cart**

**Diagram:**



**Brief Description:**

The user can add a food item to the cart.

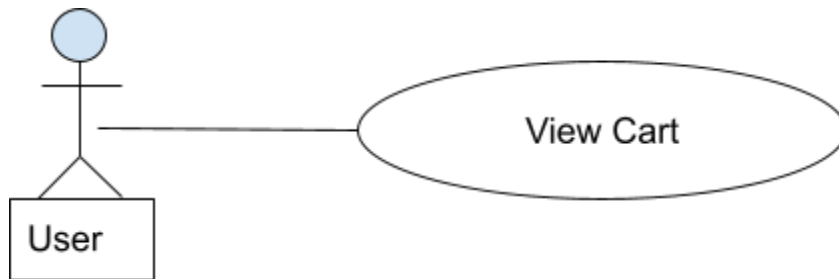
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's home page or the Item information page.

1. The user chooses the quantity of the item they want
2. The user clicks on add to cart for a particular item
3. The system updates the cart information

Use case: **View Cart**

**Diagram:**



**Brief Description:**

The user can view the selected items and quantity.

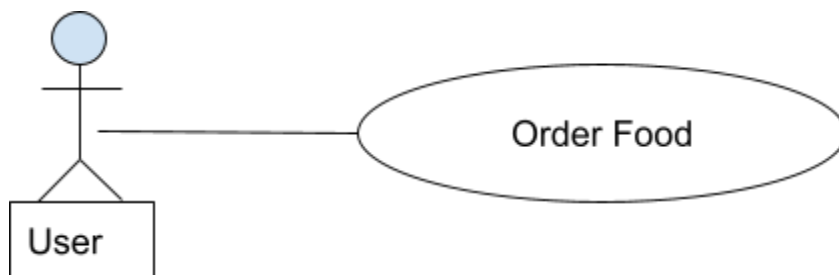
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's home page or the Item information page.

1. The user clicks on the cart icon/button
2. The system redirects user to the cart page
3. The system displays information about the food items that the user has selected (if any). The information may include the item name, picture, price, quantity and total quantity
4. The user has an option to checkout / order the food items that are selected

Use case: **Order Food**

**Diagram:**



**Brief Description:**

The user can order selected items in his/her cart.

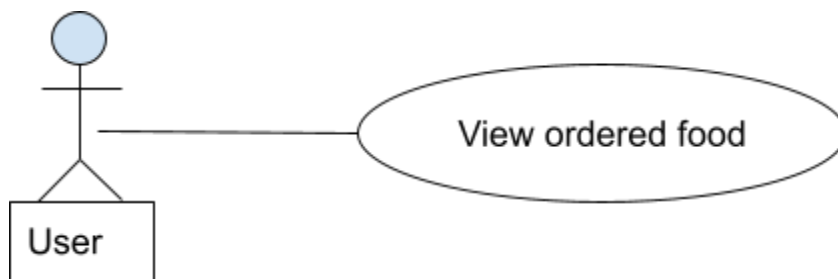
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's cart page.

1. The user clicks on the checkout / order button
2. The systems asks for a phone number and their table number
3. The user provides the required information and clicks submit.
4. The system updates details about the food order in the database
5. The system shows the user that the order has been placed / not placed

Use case: **View ordered food**

**Diagram:**

**Brief Description:**

The user can view the status and info of food items ordered.

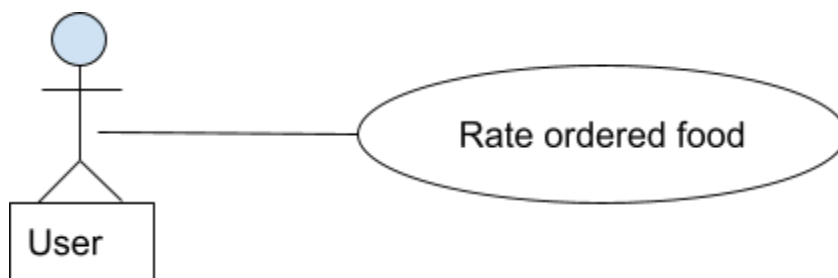
**Initial Step-By-Step Description:**

Before this use case can be initiated, the user has already accessed the System's orders page.

1. The system displays the current status of the order

Use case: **Rate ordered food**

**Diagram:**

**Brief Description:**

The user can rate the food he/she ordered.

**Initial Step-By-Step Description:**

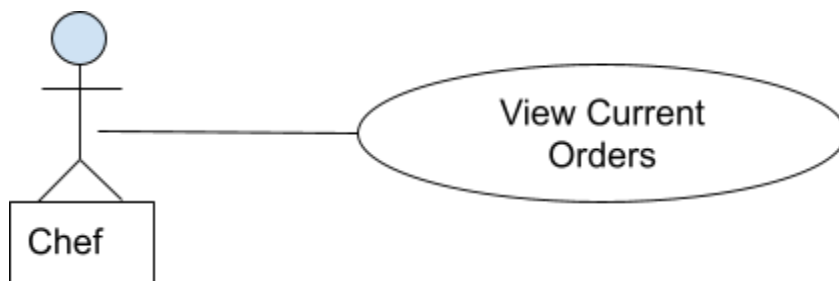
Before this use case can be initiated, the user has already accessed the System's orders page and has placed an order which is now in ready state.

1. The user can choose to select a rating for each individual food item
2. The user can choose to write a review for each individual food item
3. If any of the above steps are done, the user can click on submit to submit the rating and review
4. The system updates the information provided by the user in the DB

**2.2.2 Chef's use case**

Use case: **View Current Orders**

**Diagram:**

**Brief Description:**

The chef can view all the uncompleted orders placed by users.

**Initial Step-By-Step Description:**

Before this use case can be initiated, the chef has already accessed the System's home page and is logged in as a chef.

1. The system displays a list of current (not marked as ready) orders placed by the user
2. The chef can choose to update the state of an order from placed to accepted, being made, ready

Use case: **Update order status**

**Diagram:**



**Brief Description:**

The chef can update the status of an order placed by the user.

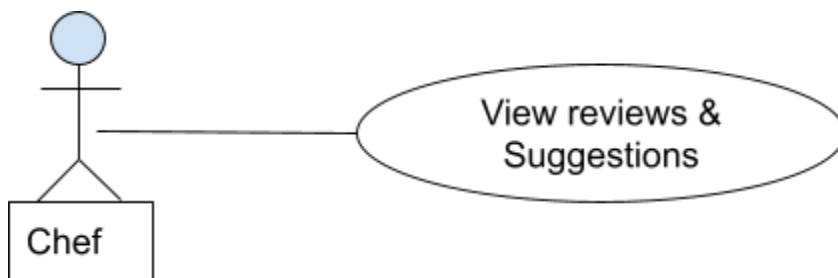
**Initial Step-By-Step Description:**

Before this use case can be initiated, the chef has already accessed the System's home page and is logged in as a chef.

1. The system displays a list of current orders placed by the user
2. The chef chooses to update the state of an order from placed to accepted, or accepted to being made, or being made to ready
3. Upon selecting each state, the system will confirm the chef's choice and update the state
4. The system updates the state in the DB

**Use case: View Order Review & Suggestion**

**Diagram:**



**Brief Description:**

The chef can view order reviews and suggestions given by the users.

**Initial Step-By-Step Description:**

Before this use case can be initiated, the chef has already accessed the System's home page and is logged in as a chef.

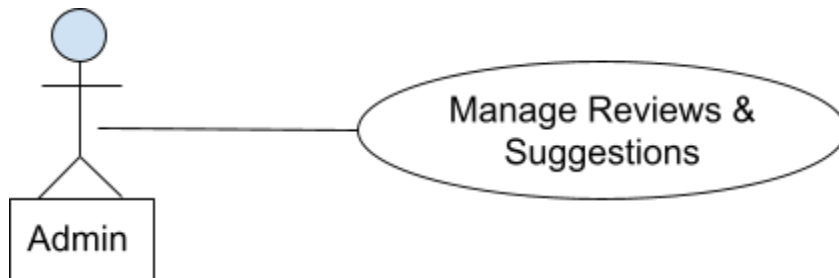
1. The system will display a list of orders along with their suggestions sorted by date.

2. The system will display a list of recent reviews along with the food item for which the review was.

### 2.2.3 Admin's use case

Use case: **Manage Reviews & Suggestions**

**Diagram:**



**Brief Description:**

The admin can manage reviews & suggestions given by users.

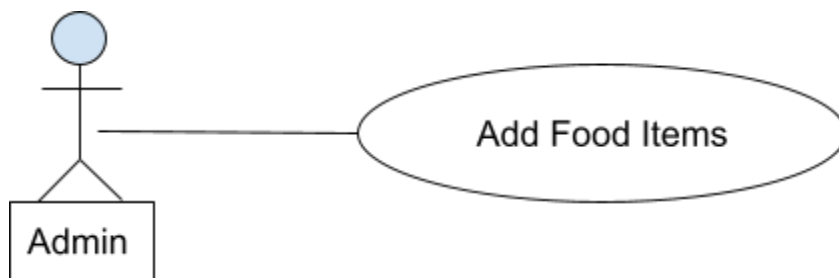
**Initial Step-By-Step Description:**

Before this use case can be initiated, the admin has already accessed the System's home page and is logged in as an admin.

1. The system shows the list of items along with their review
2. The system shows the list of suggestions on the user's order
3. The admin can choose to delete a review
4. The admin can choose to delete a suggestion

Use case: **Add Food Items**

**Diagram:**



**Brief Description:**

The admin can add food items.

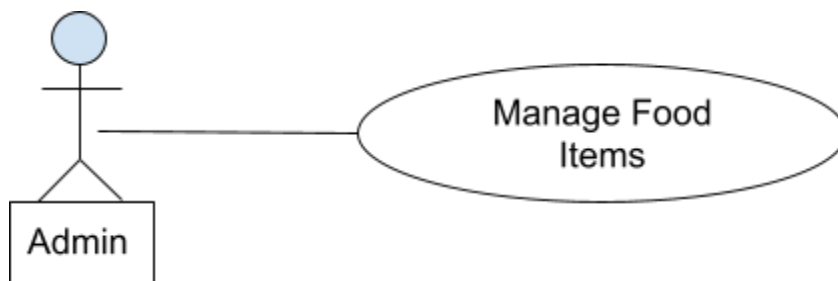
**Initial Step-By-Step Description:**

Before this use case can be initiated, the admin has already accessed the System's home page and is logged in as an admin.

1. The system displays a form which contains information such as the food item's name, picture, price and description.
2. The admin fills all the mandatory fields
3. The admin clicks on submit button
4. The system adds the provided details in the DB

Use case: **Manage Food Items**

**Diagram:**



**Brief Description:**

The admin can edit and delete food items.

**Initial Step-By-Step Description:**

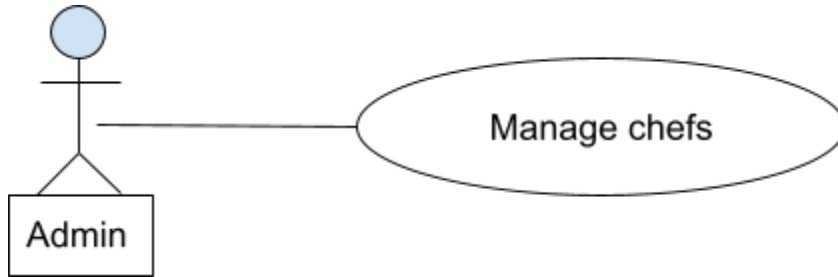
Before this use case can be initiated, the admin has already accessed the System's home page and is logged in as an admin.

1. The system lists out all the available food items in the DB
2. The admin can choose to edit or delete the food item
  - a. If the admin chooses to edit the item
    - i. The system opens a dialog box / web page for updation of information
    - ii. Admin updates information and clicks on submit
    - iii. The system updates the information and brings admin back to the list of items
  - b. If the admin chooses to delete the item
    - i. The system prompts for a confirmation
    - ii. The Admin can choose to confirm
    - iii. The system deletes if confirmed or brings admin back to the list of items

Use case: **Manage Chefs**

**Diagram:**





**Brief Description:**

The admin can manage chefs.

**Initial Step-By-Step Description:**

Before this use case can be initiated, the admin has already accessed the System's manage chefs page and is logged in as an admin.

1. The system prompts the admin to choose from the chefs to manage
2. The admin can choose to add a new chef account
  - a. The system will open a dialog box asking for additional information about the chef such as the login id
3. The admin can choose to delete a chef account
  - a. The system will ask for confirmation

### ***2.3. User Characteristics***

The user and chef is expected to be Internet Literate and be able to use a search engine, navigate the website, use buttons, drop downs, menus and similar tools. The main screen of the web application will have a search function to search for a food item.

The admin is expected to be Internet Literate and be able to manage the restaurant's menu, navigate the website and also manage the ratings and reviews. The admin is also expected to be able to use buttons, drop downs, menus and similar tools.

### ***2.4. Non-Functional Requirements***

The restaurant ordering system will be on a server which will be the part of an Intranet network. A tablet or similar machine will be used, which will be connected to the intranet. The software developed here assumes the use of a tool such as Apache and PHP server for connection between the Web pages and the database. The speed of the User's connection will depend on the hardware used rather than characteristics of this system.

The server will also contain a database management system - MySql and run Windows Operating System.

### 3. Requirements Specification

#### 3.1. External Interface Requirements

The only external interface required is a Database which will be running on the same server on which the application will be running. The database will hold every detail of the application such as user's phone numbers or email addresses, food items in the restaurant, user reviews and ratings, etc.

#### 3.2. Functional Requirements

##### 3.2.1 Search Items

<b>Use Case Name</b>	Search items
<b>Trigger</b>	The user accesses the system's home page
<b>Precondition</b>	N/a
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. The user types some name of the food they want to search for in the search bar</li><li>2. The system searches for related food items and shows a list of those items</li></ol>
<b>Alternative Paths</b>	N/a
<b>Postcondition</b>	<ol style="list-style-type: none"><li>1. The user can choose to add the item to cart</li><li>2. The user can choose to view info about that item by clicking on the item</li></ol>
<b>Exception Paths</b>	The User may abandon the search at any time.
<b>Other</b>	The items come from the DB

##### 3.2.2 Food Display

<b>Use Case Name</b>	Food Display
<b>Trigger</b>	The user has accessed the system's home page
<b>Precondition</b>	N/a
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. The user accesses the home page</li><li>2. The system will display a list or grid of food items along with their pictures, price and a button for add to cart.</li></ol>
<b>Alternative Paths</b>	<ol style="list-style-type: none"><li>1. The user clicks on a particular food item</li><li>2. The system shows info about that food item</li></ol>

<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. The user can choose to add the item to cart</li> <li>2. The user can choose to view info about that item by clicking on the item (already here if followed alternative path)</li> </ol>
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. The user clicks back. (alternative path)</li> <li>2. The system brings the user back to the home page.</li> </ol>
<b>Other</b>	The items come from the DB

### 3.2.3 View Cart

<b>Use Case Name</b>	View cart
<b>Trigger</b>	The user has accessed the system's home page
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The user might have added items to cart</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The user clicks on view cart button</li> <li>2. The system displays a list of items already in the user's cart (if available) or displays an empty cart</li> <li>3. If there are items in the cart, the system will show a similar interface to how the items are listed on the web page.</li> <li>4. The system will also show the total price of all the items.</li> <li>5. The system will show a textbox asking for the user's phone number and a button to order the items.</li> </ol>
<b>Alternative Paths</b>	<ol style="list-style-type: none"> <li>1. The user can choose to access the cart any time from the navigation bar</li> </ol>
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. The user can choose to order the items in the cart</li> <li>2. The system will ask some identification information</li> <li>3. The user will fill it up and click on order</li> <li>4. The system will update the order</li> </ol>
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>3. The user clicks back. (alternative path)</li> <li>4. The system brings the user back to the home page.</li> </ol>
<b>Other</b>	The items come from the DB

### 3.2.4 View Order

<b>Use Case Name</b>	View order
<b>Trigger</b>	The chef has accessed the system's home page
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The chef has logged in</li> </ol>

<b>Basic Path</b>	1. The system shows the chef available orders as a card. The card will contain the list of food items in that order and a
<b>Alternative Paths</b>	N/a
<b>Postcondition</b>	1. The chef can choose to update status of an order 2. The system will update upon confirmation
<b>Exception Paths</b>	1. The chef cancels to update the status when asked to confirm
<b>Other</b>	N/a

### 3.2.5 Rate Food

<b>Use Case Name</b>	Rate Food
<b>Trigger</b>	The user has clicked to view ordered food
<b>Precondition</b>	1. The user has ordered some food 2. The chef has updated the status of the food as ready
<b>Basic Path</b>	1. The system will show rating bars / stars along with a review textbox along each individual food item that had been ordered. 2. The system also shows a single textbox for any suggestions to the chef 3. The user selects the ratings and writes the review or suggestions if he/she wants to. 4. The user clicks on submit 5. The system updates the rating, review and suggestion if provided.
<b>Alternative Paths</b>	N/a
<b>Postcondition</b>	1. The user is brought back to the home page.
<b>Exception Paths</b>	1. The user can choose to leave reviews and suggestions blank. 2. The user can choose not to rate or review.
<b>Other</b>	N/a

### 3.2.6 Login

<b>Use Case Name</b>	Login
<b>Trigger</b>	The chef / admin clicks on the login page
<b>Precondition</b>	1. The chef / admin has access to the system's home page

<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The system prompts the chef / admin with a login id and password</li> <li>2. The chef / admin will enter the required details</li> <li>3. The system will check if the info is correct</li> </ol>
<b>Alternative Paths</b>	N/a
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Redirected to the home page and their session will be active</li> </ol>
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. Upon wrong / missing information, the system will prompt to enter again</li> </ol>
<b>Other</b>	The information will be checked with the information already present in the database

### 3.2.7 Add chef

<b>Use Case Name</b>	Add chef
<b>Trigger</b>	The admin clicks on manage chefs page
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The admin is logged in</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The admin goes to manage chefs page</li> <li>2. The system shows a list of accounts in the system</li> <li>3. The admin can click on add chef button to add a chef</li> <li>4. The system will ask for additional information about the chef</li> <li>5. The admin will submit the information</li> <li>6. The system will create a new chef account</li> </ol>
<b>Alternative Paths</b>	N/a
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Prompt the admin if the operation was successful</li> <li>2. Redirected to the manage chefs page</li> </ol>
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. The admin can choose to cancel the process at any time.</li> </ol>
<b>Other</b>	The information will be updated in the DB

### 3.2.8 Change Password

<b>Use Case Name</b>	Change password
<b>Trigger</b>	When a chef logs in with default password (for the first time) or admin/chef clicks on change password button.
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The chef knows the login id and default password</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The chef enters the login id and default password</li> <li>2. The system prompts the chef to enter a new password and repeat the new password</li> </ol>

	<ol style="list-style-type: none"> <li>3. The chef enters a new password and repeats the new password</li> <li>4. The chef clicks on submit</li> <li>5. The system updates the new password</li> </ol>
<b>Alternative Paths</b>	<ol style="list-style-type: none"> <li>1. The admin/chef clicks on change password</li> <li>2. The system asks for new password and repeat password</li> <li>3. The admin/chef enters a new password and repeat password.</li> <li>4. The admin/chef clicks submit</li> <li>5. The system updates the new password in the DB.</li> </ol>
<b>Postcondition</b>	<ol style="list-style-type: none"> <li>1. Redirected to the home page after successful password change.</li> </ol>
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. The chef / admin can choose not to change their passwords and hit back and go to the previous page.</li> <li>2. In the case of a first time chef login, the chef won't be logged in and access other pages which require login until the password has been changed.</li> </ol>
<b>Other</b>	N/a

### 3.3. Detailed Non-Functional Requirements

#### 3.3.1 Logical Structure of the Data

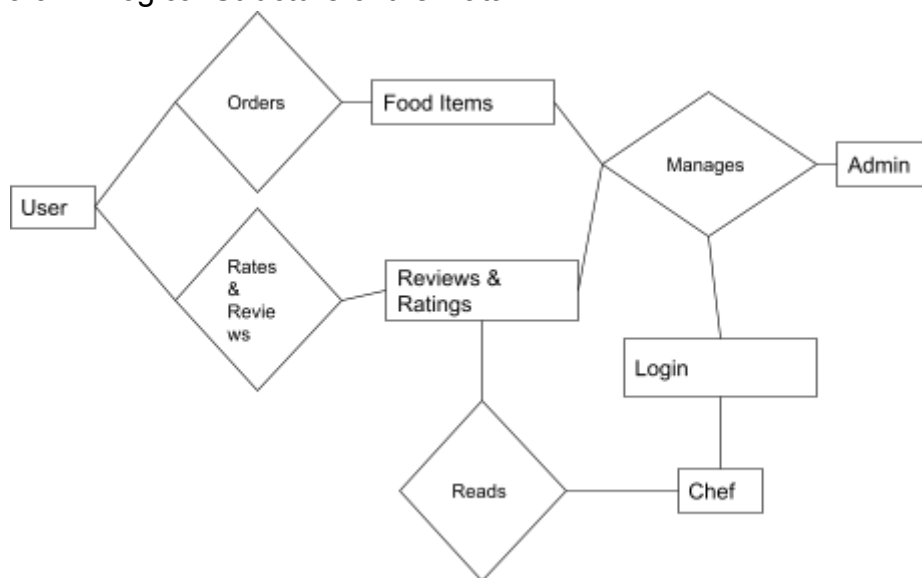


Figure 2 - Logical Structure of the Data

#### User Entity

Data Item	Type	Description	Comment
Email Address	Text	Internet address	
Password	Text	Password of the chef/admin	Hash of the password will be stored.

### Food Entity

Data Item	Type	Description	Comment
Id	Number	Identifier of the food item	Primary Key
Name	Text	Name of the food item	
Price	Number	Price of a single piece of item	
Description	Text	Details / Info about the food item	
Photo	Text	Image of the food item	

### Rating & Reviews Entity

Data Item	Type	Description	Comment
Id	Number	Identifier of the review	Primary key
Food Id	Number	Identifier of food it's related to	Foreign key
Rating	Number	Between 1 - 5	
Review	Text	Review about the food item	

### Orders Entity

Data Item	Type	Description	Comment
Order Id	Number	Identifier of the order	Primary Key
Order Status	Text	Status of the order	
User Phone number	Text	User's phone number for identification	
Suggestion	Text	A suggestion about the food to the chef	

### Order Details Entity

Data Item	Type	Description	Comment
Id	Number	Identifier of Row	Primary Key
Order Id	Number	Identifier of order	Foreign Key
Food Id	Number	Identifier of Food	Foreign Key



### 3.3.2 Security

The whole application is on the restaurant's intranet. Only devices connected to the wireless network will be able to access the web pages. A password will be required to connect to the restaurant's intranet.

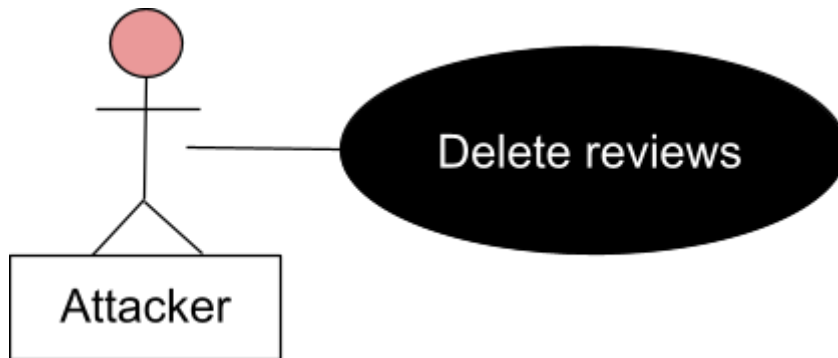
The tablet on which the system will be accessed will have only login based security. Without login, the system will allow the customer to access the food menu and order items. One thing to note is that the customer will have to enter their phone number before the order can be placed.

For preventing the user from accessing the admin or chef's pages, a login system will be implemented which requires a login id as well as password. Only The Admin has 'modify' access to the ratings and reviews. The user has read / write access and the chef has read only access to the ratings, reviews and suggestions.

### 3.4. Misuse Cases

#### 3.4.1 Misuse case: Delete Reviews

Diagram:



#### Brief Description:

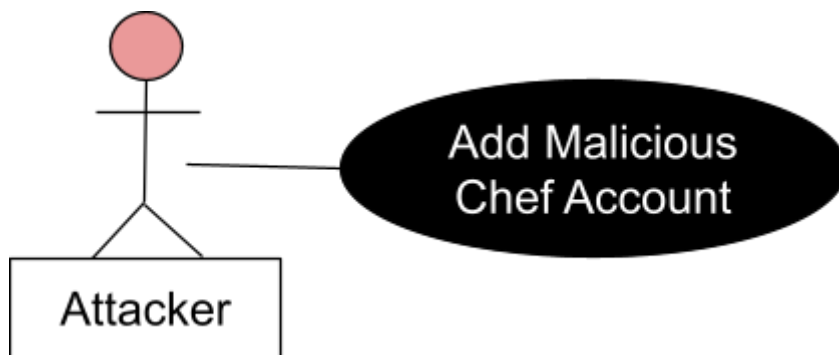
If the attacker got access to the admin's account might be able to change or delete the reviews of legitimate users. This will have an impact on the restaurant's services. For example, imagine an attacker deleting every 3,4 and 5 rating from every food item. This will lead to food items only having lower reviews and thus customers will be discouraged to eat in the restaurant because almost every food item has low rating.

#### Attacker Motivation / Advantage:

The attacker can be a malicious or disgruntled employee, person or organization who wants to damage the reputation of the restaurant.

#### 3.4.2 Misuse case: Add Malicious Chef Account

Diagram:



#### Brief Description:

If the attacker is able to get access to an admin account, he/she can easily add a malicious chef account. This might lead to unwanted access to the restaurant's ordering system.

**Attacker Motivation / Advantage:**

The attacker can be a malicious or disgruntled employee, person or organization who wants to damage the reputation of the restaurant.

**3.4.3 Misuse case: Delete actual chef's account**

**Diagram:**



**Brief Description:**

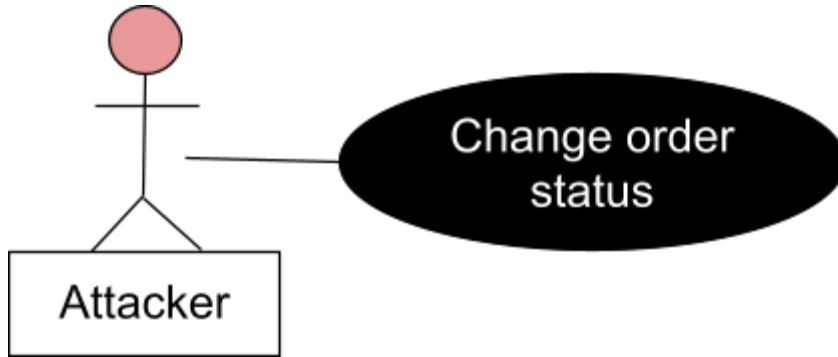
If the attacker is able to gain access to the admin's account, the attacker can delete the account of an actual chef working at the restaurant. This might lead to denial of service for the chef.

**Attacker Motivation / Advantage:**

The attacker can be any malicious user, person or organization who wants to create havoc in the restaurant, create delay or confusion and waste time. The restaurant might have to shut down in order to figure out why the chefs can't log in to their accounts.

**3.4.4 Misuse case: Change order status of a customer's order**

**Diagram:**



**Brief Description:**

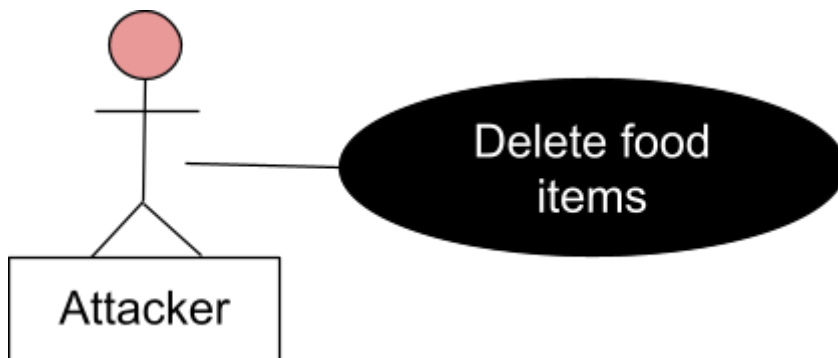
If the attacker is able to gain access to one of the chef accounts, the attacker can change the order status of any order. For example, the attacker can change the order status of an order as soon as they arrive. This might lead to the order being marked as ready and not shown to the chef at all. This will lead to loss of business and the customer never getting the food they ordered.

**Attacker Motivation / Advantage:**

The attacker can be any malicious user, person or organization who wants to create havoc in the restaurant, create delay or confusion and waste time. The restaurant might have to shut down in order to figure out why the chefs can't see orders placed by the customers.

**3.4.5 Misuse case: Delete available food items**

**Diagram:**



**Brief Description:**

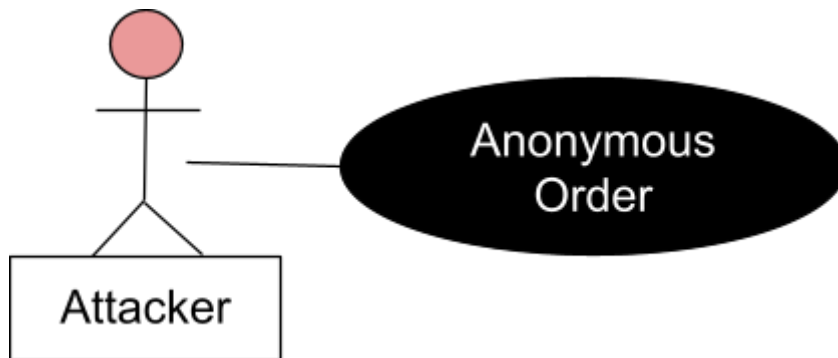
If the attacker is able to gain access to an admin account, the attacker can delete any food item they want. For example, imagine if the attacker deleted every food item that has ever been added to the system's menu. In this case, the restaurant will have to go through the process of doing it again.

**Attacker Motivation / Advantage:**

The attacker can be any malicious user, person or organization who wants to create havoc in the restaurant, create delay or confusion and waste time. The restaurant might have to shut down in order to figure out why the menu is empty and there is no food item to be ordered.

#### 3.4.6 Misuse case: Anonymous Order

**Diagram:**



##### **Brief Description:**

An attacker who has access to the restaurant's intranet network can be able to create orders which actually don't belong to anyone. This will lead to a lot of food wastage, loss of business time as well as resources.

The attacker will then also be able to add new orders to the customer's session and this will lead to charging the customer more than what they ordered.

##### **Attacker Motivation / Advantage:**

The attacker can be any malicious user, person or organization who wants to create havoc in the restaurant, create delay or confusion and waste time. The restaurant might have to shut down in order to figure out why there are anonymous orders.

## **4. Threat Identification & Modeling**

### **4.1. Threat: Weak intranet password**

##### **Brief Description:**

A weak password to get access to the intranet will directly give the attacker access level equal to a customer of the restaurant. This can lead to exploitation of misuse case ordering food (3.4.3)

**Stride Category:** Spoofing Identity, Elevation of Privilege

**Mitigation:** Strong password practices; Not writing, accidentally leaking passwords anywhere. Change passwords regularly.

#### ***4.2. Threat: Weak admin / chef login password***

**Brief Description:**

For getting access to the login page, the attacker must either have access to one of the physical devices already connected to the intranet network or have access to the intranet network in some way or form. Once the attacker is inside the network, it can try to attack the login page for chefs and admins.

**Stride Category:** Elevation of Privilege, Spoofing Identity

**Mitigation:** Strong password practices; not writing, accidentally leaking passwords.

#### ***4.3. Threat: Physical access to devices that should be used inside the restaurant.***

**Brief Description:**

Devices that are being used inside the restaurants have a lot of network related information on them. If an attacker gets access to these devices which are in use without being inside the restaurant or any old device which was previously used, the attacker can gain access to the intranet network (4.1.)

**Stride Category:** Elevation of Privilege, Spoofing Identity

**Mitigation:** Clean device memory before disposal. Fix devices inside the restaurant so that they cannot be carried out without an authority being notified.

#### ***4.4. Threat: Man in the middle***

**Brief Description:**

Assuming an attacker is already inside the intranet network, it can launch a man in the middle attack. The attacker can sniff, capture and change information in the network packets and cause misuse case 3.4.3. Or can manage to steal information about login credentials.

**Stride Category:** Information Disclosure

#### **4.5. Threat: Loss of Data**

**Brief Description:**

Assuming an attacker has access to the admin's account. The attacker can try to delete everything that he/she possibly can. This might lead to a huge loss of data including food items, genuine customer reviews and ratings. This will also mean that the restaurant will have to set up and add the food items again.

**Stride Category:** Denial of Service, Tampering of Data

**Mitigation:** Keeping regular backups of the database.