Jose Miranda, Christian Ramirez

CS-319

December 4th,2020 Document Version 2.0

M&R Software

Table of Contents

Vision and Scope

Background: Business Opportunity and Customer Needs

A local business named "XYZ Solutions" with 1000+ employees on-site has asked your team to develop an online application named "Cafeteria Ordering System" that will provide its employees with the opportunity to order and purchase their meals from the company cafeteria. The intent is that by placing the order online several time-saving and cost-saving measures will be put into place, for example lines for selecting, ordering and paying for food will be reduced or ideally eliminated. From the employee standpoint, this will result in employees having more time to enjoy their lunch. From the business standpoint, this will provide the opportunity for the cafeteria to have sufficient product on hand as well as avoid the expense of waste with regard to unused product.

An added consideration is that not only will meals be ready for employees who choose to eat in the cafeteria, this system will provide the opportunity to have meals delivered to select locations so that employees can have their meals at locations outside of the cafeteria (for example in an office space, or for a working lunch at a location away from the cafeteria). Additionally, in the future, employees will also have the opportunity to order their meals not only from the XYZ Solutions cafeteria, but also from nearby restaurants for delivery to their specified locations.

Vision Statement

For employees wishing to order meals from the company cafeteria the Cafeteria Ordering System is an Internet Based application that will accept individual or group meal orders, process payments, and trigger delivery of the prepared meals to a designated location on the XYZ Solutions campus. Unlike the current telephone and manual ordering processes, employees who use the Cafeteria Ordering System will not have to go to the cafeteria to get their meals, which will save them time and will increase the food choices available to them.

Major Features

1. Order meals from the cafeteria menu to be picked up or delivered.
2. (in the future)Order meals from local restaurants to be delivered.
3. Create, view, modify and delete meal service subscriptions.
4. Register for meal payment options.
5. Request meal delivery.
6. Create, view, modify, and delete cafeteria menus.

7. Order Custom meals that aren't on the cafeteria menu.
8. Produce recipes and ingredients lists from custom meals from the cafeteria.
9. Provide system access through outside internet access by authorized employees.

Project Selection:

The development of a web-based cafeteria ordering system, an overall enhancement of the current cafeteria meal ordering system and a new web application. The current cafeteria ordering system requires employees to physically go to the cafeteria and select food for purchasing. As a result, employee time is wasted waiting in line and company products go unused. In order to help keep systems modern and alleviate the issues previously outlined, our project will address these issues directly by building from the ground up an all encompassing Cafeteria Ordering System. Thus, the implementation of this project will address the current and future needs of XYZ Solutions campus.

Problem Definition:

To address the issue of long lines and outdated systems in conventional cafeterias, we need to offer all services provided in the cafeteria in an intuitive and convenient way. We need easy access for employees to place their orders on virtually any device with internet connection, and we need to do it in a way that employees can easily understand. The User Interface needs to be visualing pleasing all while providing its users with familiar UI conventions that allows for them to quickly pick up and use.the ordering process must be quick and effortless so employees want to keep using it.

During lunch time the cafeteria staff must take in food orders as well as prepare them in a timely manner. Our team will work on alleviating some of the stress from cafeteria staff. Staff members need a way to get ahead of the work all while doing it in a manner that flows easily into their current work flow. The lack of a cohesive system can result in a hindrance in the performance of staff, Therefore training and an intuitive system is essential for employees to adjust to the change.

Problem Solution:

To alleviate employees from long wait times that take time away from their break and give our cafeteria staff an opportunity to get ahead of the orders we will create a web based application, this application will be easily accessible on any browser whether it be mobile a laptop or a desktop pc. The UI will be easy to use by following well established conventions of online ordering systems that employees can easily pick up and use. Convenience is key to creating an application that employees want to come back to so our team will create a login system that can save payment methods and preferences for meals.

On the cafeterias staff's side of things they need a system that is simple to understand and provide staff with the ability to add, remove, and edit menu items with minimal effort.Food Allergens are a major concern for the wellbeing of our customers. That's why food ingredients will be added as well.

Scope and Objectives of the System:

➢ Scope of the System:

The Cafeteria Ordering System will contain the following main components:

- User interface that provides direct access to the cafeteria menu, custom menus, user account information, and service management
- An SQL database that saves and updates user information: username, password, contact information, payment information, meal service subscription info
- Meal Service Management that allows user to track past and future meal orders, meal payments, delivery locations, custom meals, and service subscriptions

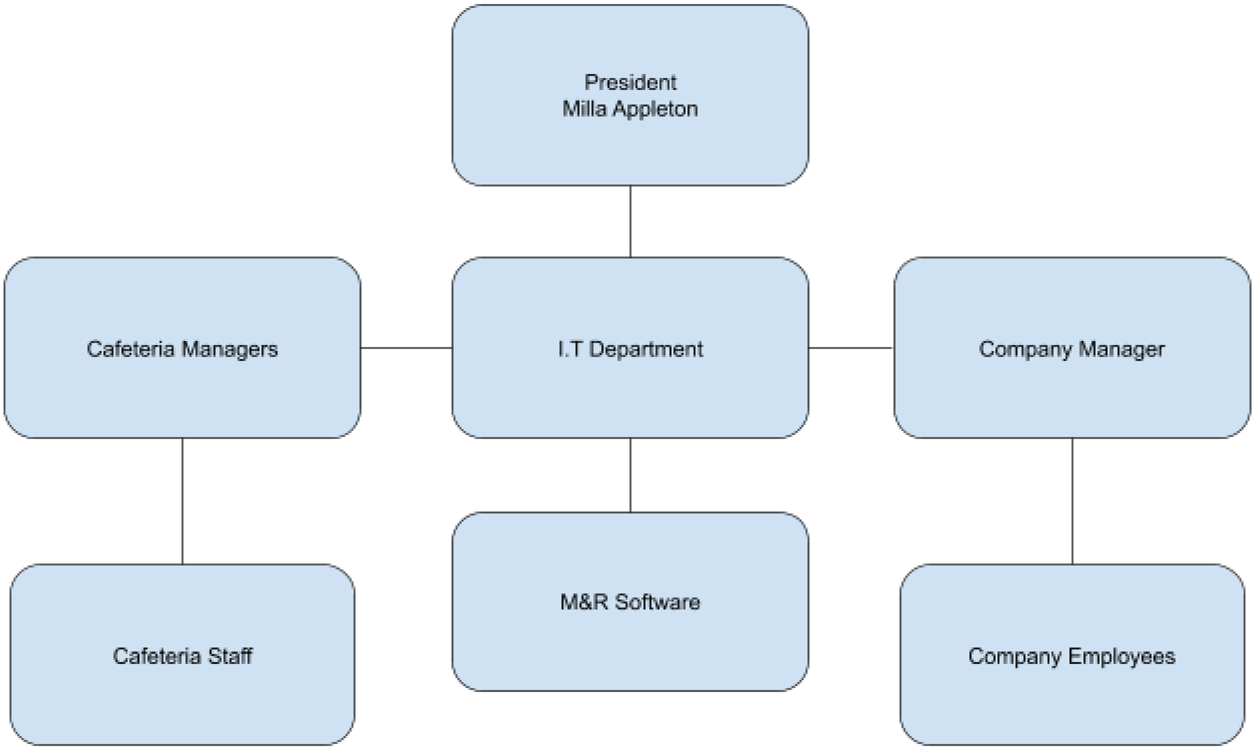
➢ Objectives of the System:

The Cafeteria Ordering Systems' objectives are

- Maintaining employee information updated
- Providing access to the online Cafeteria Ordering System
- Tracking of payment status and history
- Providing access to subscription information: username, password, contact information, payment information, meal service subscription info, etc.
- Track current orders (whether these were purchased at the campus's cafeteria or via the online ordering website)
- Access to custom meal options

- Access to local restaurants from which to order meals from
- Access to subscription options:
    - Includes:
        - Creating a subscription
        - View the users current subscriptions
        - Modifying meal service subscriptions
        - Deleting meal subscriptions
- Access to cafeteria menus:
    - Includes:
        - Creating a menu
        - Viewing the current menus
        - Modifying menus
        - Deleting menus

Organization Chart:



Use Cases

Summary

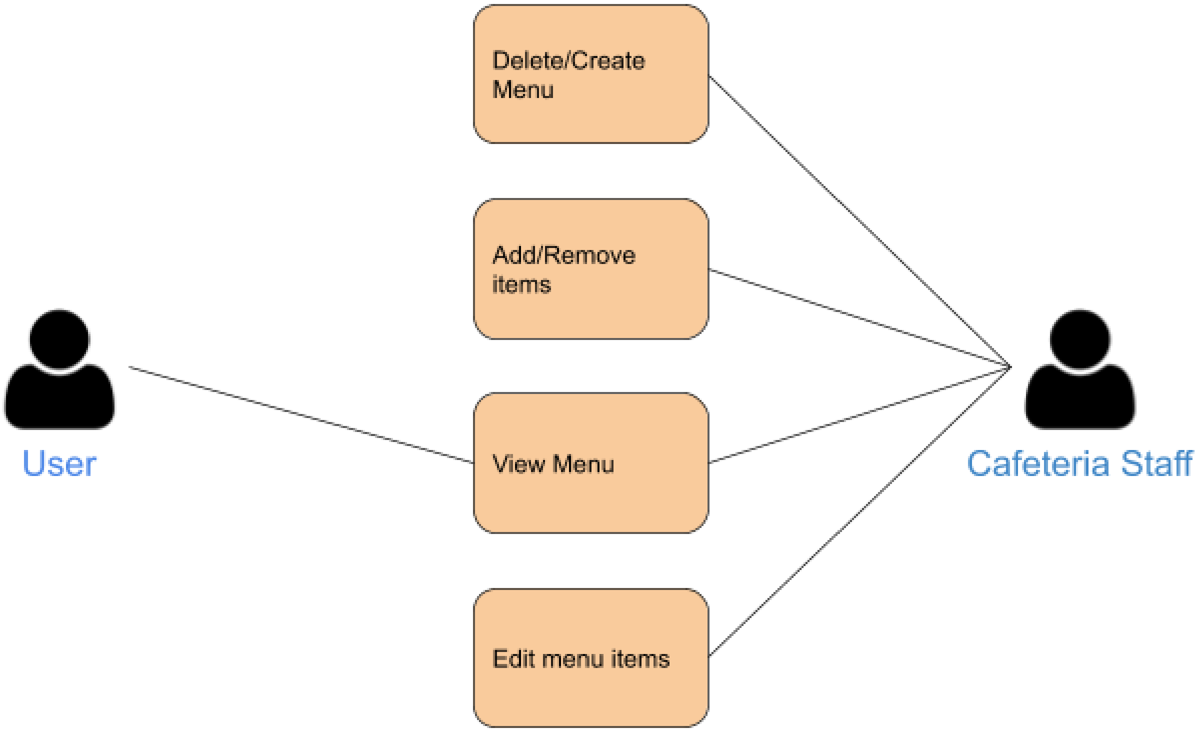
This section highlights seven of the modules presented by Cafeteria Ordering System. The application's modules are: Meal Ordering (M1), Local Restaurant Delivery (M2), Meal Subscription Services (M3), Payment Options (M4), Delivery Services (M5), Cafeteria Menu (M6), Custom Meal Options (M7), Recipe/Ingredient List Builder (M8), and Remote System Access (M9). Each module provides a detailed description of the use cases including triggers, preconditions, postconditions, flow, exceptions, priority and frequency of use. This section also provides different diagrams for each module: Activity Diagram, Extended Systems View, Data Flow, Sequence Diagram and Data Models. **For summary of the Modules and use cases, please refer to Figure 1 .**

Figure 1: Module Summary

Module	Primary Actor	Use Cases
1.0 Meal Ordering	Employee	
2.0 Local Restaurant Delivery	Cafeteria Staff	
3.0 Meal Subscription Services	Employee	
4.0 Payment Options	Employee	
5.0 Delivery Services	Cafeteria Staff	
6.0 Cafeteria Menu Services	Cafeteria Staff	6.1 Delete/Create Menu 6.2 View Menu 6.3 Add/Delete items 6.4 Edit Menu Items
7.0 Custom Meal Services	Employee	7.1 Select Delivery Options 7.2 Add/Remove Items 7.3 Add/Remove Directional Steps 7.4 Cancel Custom Order 7.5 View Order Details
8.0 Recipe/Ingredient list	Cafeteria	8.1 Search Custom Meal

Builder		Database 8.2 Generate Recipe & Ingredient List
9.0 Remote System Access		9.1 Remote Portal 9.2 Initialize Remote Access to COS

Cafeteria Menu Services



6.1 Delete/Create Menu

Cafeteria staff is allowed to delete or create entire menus. After verifying credentials on their device of choice staff members will be given the ability to access the Delete/Create Menu page.Here Staff will be able to add completely new menus where they can fill with menu items. They also have the ability to delete entire menus when they are no longer needed.

6.2 View Menu

Provided users and cafeteria staff members with the ability to view the menu. After signing in users will be able to browse through the entire menu selections

6.3 Add/Remove items

Cafeteria staff members will be able to add unique items to available menus. Entries will include price, name and ingredients. Authorized users will also be able to delete items no longer available or sold.

6.4 Edit Menu Items

Authorized cafeteria members will be able to edit items in menus. Price changes, name changes, and changes to recipes and ingredients will be able to be updated.

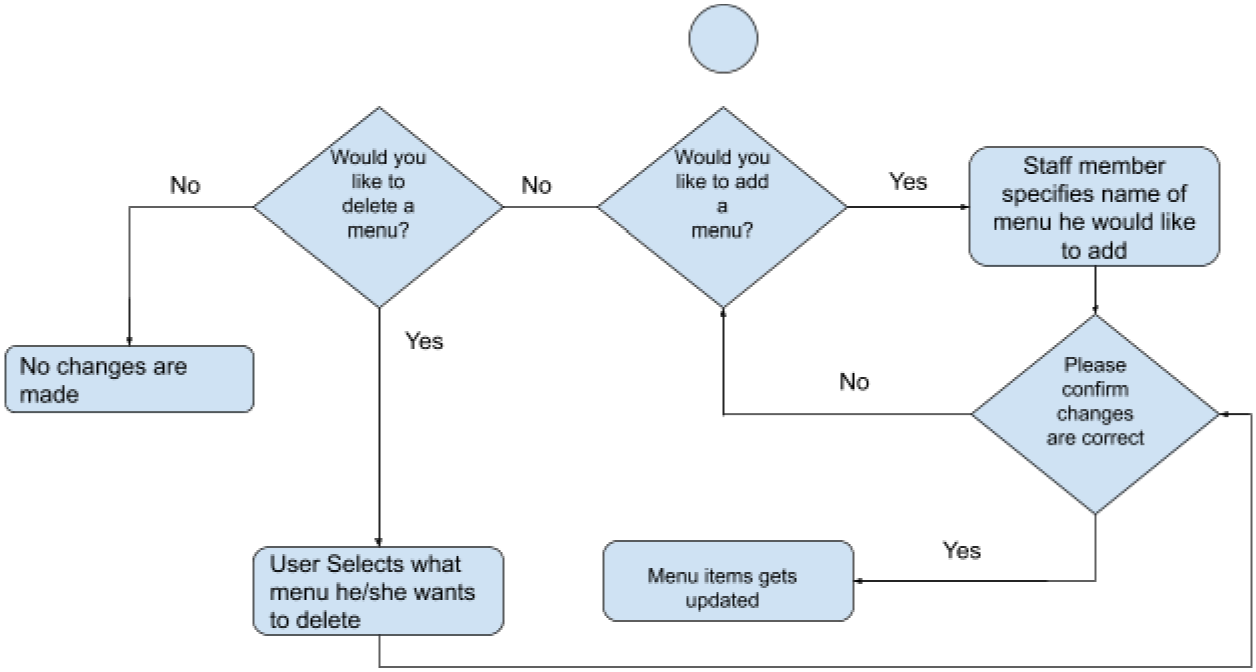
Use Case: Delete/Create Menu

The following table showcases the details for the Cafeteria Menu Services:Delete/Create Menu Use Case.

I.D and Name: Delete/Create Menu	
Created by:	Jose Miranda 11/15/2020
Primary Actor:	Cafeteria staff
Description:	Select members of the cafeteria staff will have access to modify the cafeteria menu. This includes removing an entire menu, and creating new menus as well.
Trigger:	After verifying credentials, staff will have access to delete or add menus
Preconditions:	Pre-1 Must be part of cafeteria staff (verified) P-2 Users credentials are authorized to make changes
Postconditions:	Post-1 Menu additions must be in proper format Post-2 Menu is updated in user view
Normal flow:	**Add/Create Menu** 1. User signs in with credentials 2. User selects **Modify Menu** option 3. User selects from following selections ***A.) View/Add/remove/ modify items from menu*** ***B.) Create/ delete entire menu*** 4.) After selecting **B**, all current menus will be displayed and numbered followed by the following options: ***A.) Create a new menu*** ***B.) Delete menus*** 5. If ***a.Create a new menu,*** is selected the user will enter the name of the menu he/she wants to create. If ***b.Delete menus*** is

	selected the user will enter the name of the menu he/she wants to delete 6. The user will finally be asked to confirm changes made otherwise changes will be discarded.
Exceptions:	Item not in correct format 1.) Message will display information user if incorrect format 2.) User will be prompted to reenter entry 3.) Normal flow will continue
Priority:	High
Frequency of use:	Approximately once a week to update new additional restaurant menus
Assumptions:	menus will be updated prior to the start of cafeteria opening.

Delete/Create Menu Activity Flow Diagram


Would you like to delete a menu?
No
No changes are made
Yes
User Selects what menu he/she wants to delete
No
Would you like to add a menu?
Yes
Staff member specifies name of menu he would like to add
Please confirm changes are correct
No
Yes
Menu items gets updated

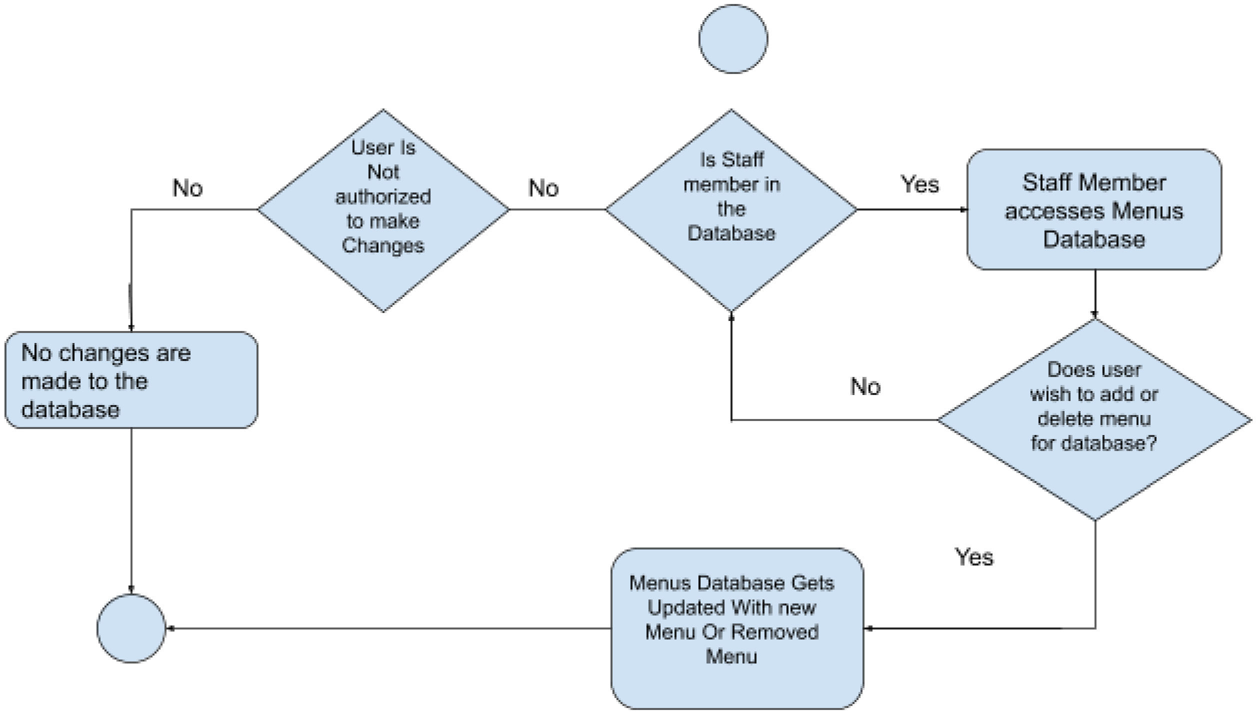
The Diagram above describes the process of deleting or creating a menu in the Cafeteria Menu Services Module.

After Signing in with authorized employee login, the user will be able to add or delete entire menus. This will be done by prompting the user if he would like to delete or add a menu. From there the user can either select a menu to delete or add to the menu.

If the user wants to delete a menu. The user must specify what menu he wants to delete and then confirm the changes made or cancel them to proceed.

If the user wishes to add a new menu he must specify the name of the menu he wants to add. The user then must confirm that his/her changes are correct before the menu can be updated.

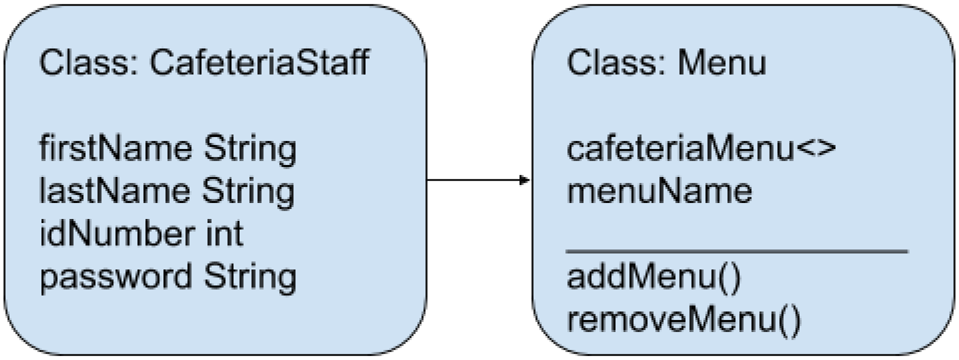
Delete/Create Menu Data Flow Diagram


No
User Is Not authorized to make Changes
No
Is Staff member in the Database
Yes
Staff Member accesses Menus Database
No changes are made to the database
No
Does user wish to add or delete menu for database?
Yes
Menus Database Gets Updated With new Menu Or Removed Menu

The diagram above gives an example of a data process of creating a new menu. If the user is in the database and authorized to make changes he will have access to adding new menus or deleting menus from the cafeteria database

At the end of every entry the user will have to confirm the changes made before it is reflected in the database.

Delete/Create Menu Data UML



Cafeteria Staff- the cafeteria class object carries cafeteria staff members information. This includes their first and last name. There employee number and there account password

Menu- this class carries a list database containing the menu items for every menu category/name. This provides the functionality to create a menu and remove an entire menu from the cafeterias database along with its items

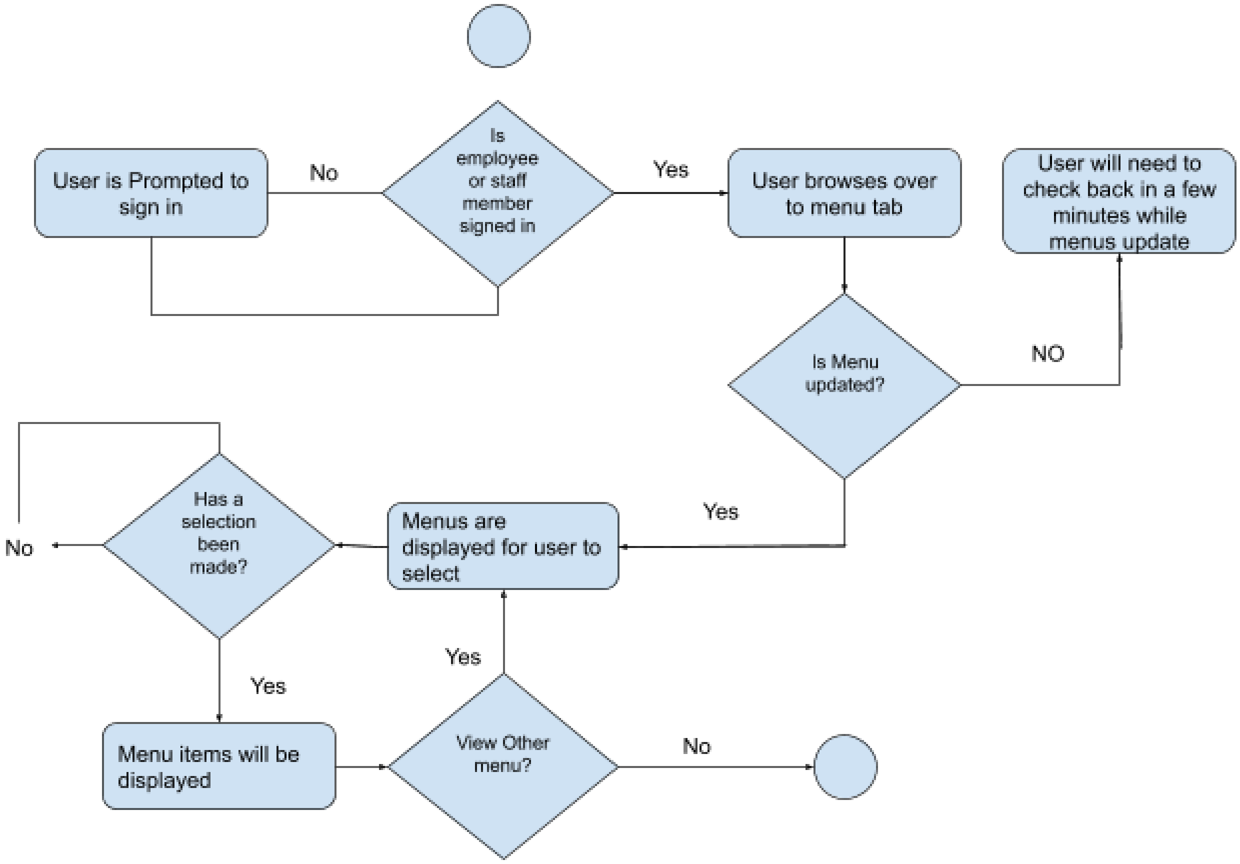
Use Case: View

The following table showcases the details for the Cafeteria Menu Services:View Menu Use Case.

I.D and Name: View Menu	
Created by:	Jose Miranda 11/15/2020
Primary Actor:	Cafeteria staff
Description:	All Cafeteria members and customers will have access to view the menu. While they can't modify anything they can view all elements of the menu including ingredients.
Trigger:	After verifying credentials, staff and employees will have access to view the menu
Preconditions:	Pre-1 Must be part of cafeteria staff or employee (verified) P-2 Users credentials are authorized to view menu
Postconditions:	Post-1 Menu is updated in user view
Normal flow:	**View menu items** 1. User signs in with credentials 2. User selects View menu 3. User selects from Menus List ***A.) Beverages*** ***B.) Meals*** ***C.)Sides*** ***D.).....*** 4.) Menu specified will be displayed and items numbered
Exceptions:	Menu Items have not been updated yet 1.) Check back later message will appear 2.) User will be taken back to select menus list

Priority:	High
Frequency of use:	Multiple times per day

View Menu Activity Flow Diagram

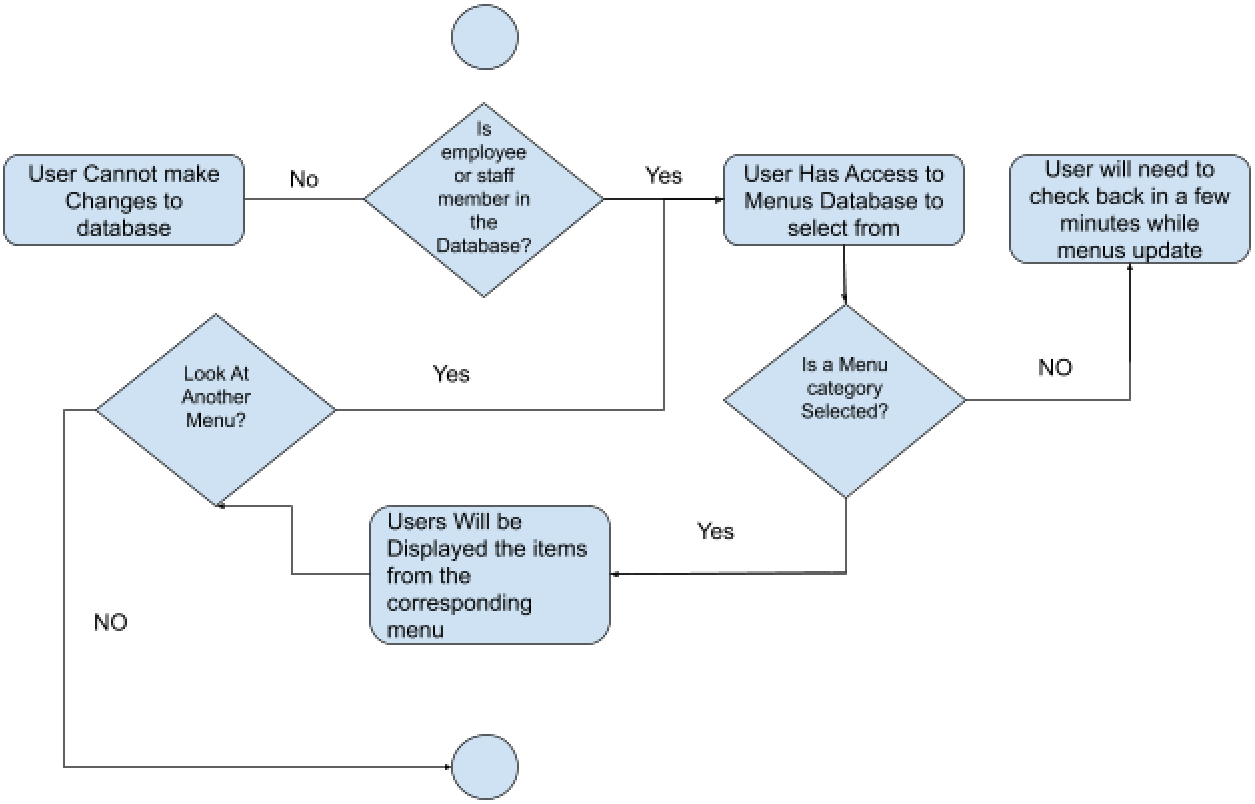

User is Prompted to sign in
No
Is employee or staff member signed in
Yes
User browses over to menu tab
User will need to check back in a few minutes while menus update
NO
Is Menu updated?
Yes
No
Has a selection been made?
Menus are displayed for user to select
Yes
Yes
Menu items will be displayed
View Other menu?
No

The diagram above walks through the process of viewing a menu on the website. This is both accessible for the employees and the cafeteria staff.

After signing in the user can hover over to the menu tab. Here the menu will be available if it has been updated for the day. Otherwise they will be prompted to check back in a few minutes. If the menus have been uploaded for the day the users will be able to select from the menus available and expand the page to show the items.

Finally the user will have the option to return to the Menus Screen to select from other menus.

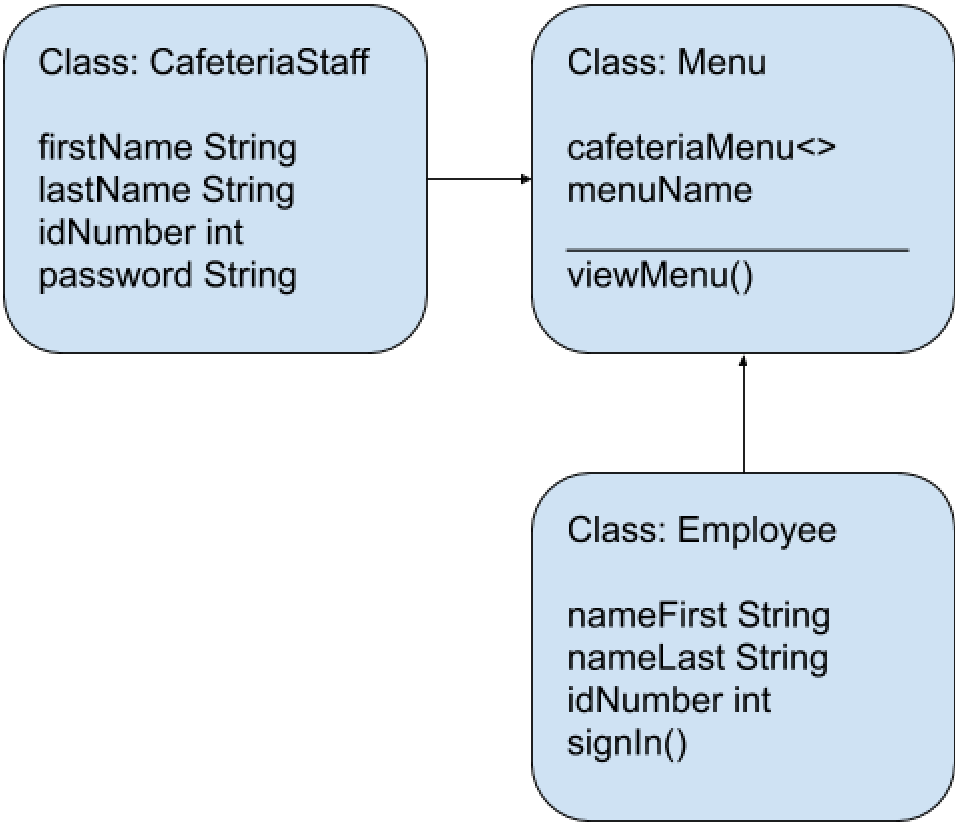
View Menu Data Flow Diagram


User Cannot make Changes to database
No
Is employee or staff member in the Database?
Yes
User Has Access to Menus Database to select from
User will need to check back in a few minutes while menus update
Look At Another Menu?
Yes
Is a Menu category Selected?
NO
Users Will be Displayed the items from the corresponding menu
Yes
NO

The diagram above walks you through the process that employees or cafeteria staff members go through to access the menus. After a user is in the database he/she will have access to the menu database.

Here users can simply browse through the menus and submenus that contain the items that they can purchase.

View Menu Data UML



Cafeteria Staff- the cafeteria class object carries cafeteria staff members information. This includes their first and last name. There employee number and there account password

Menu- this class carries a list database containing the menu items for every menu category/name. Along with the delete and add menus functionality. The viewMenu() function allows all users the view the menu including employees

Employees- this object carries employee sign-in information including their names,and id numbers for tracking orders,

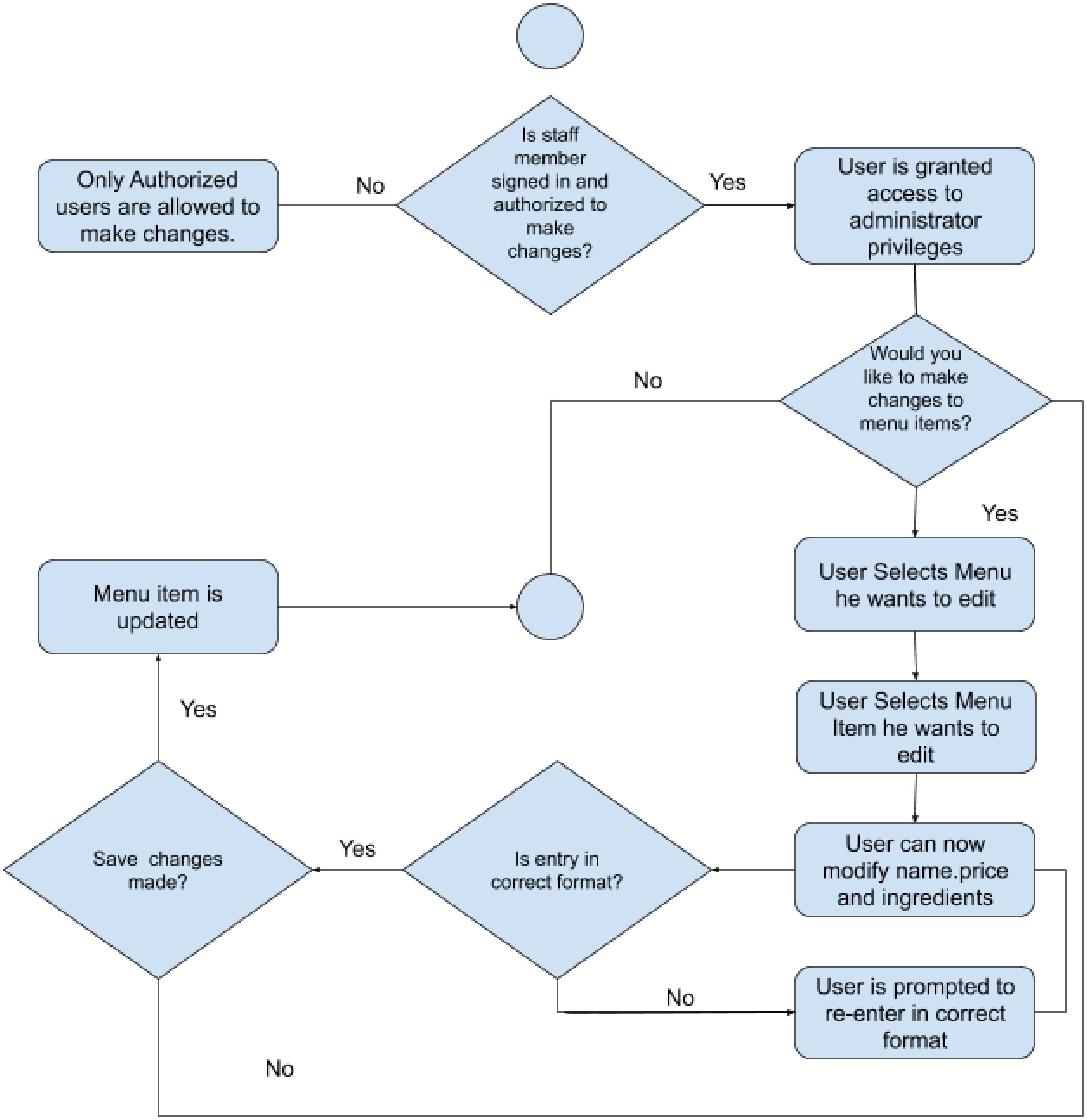
Use Case: Edit Menu Items

The following table showcases the details for the Cafeteria Menu Services:Edit Menu Use Case.

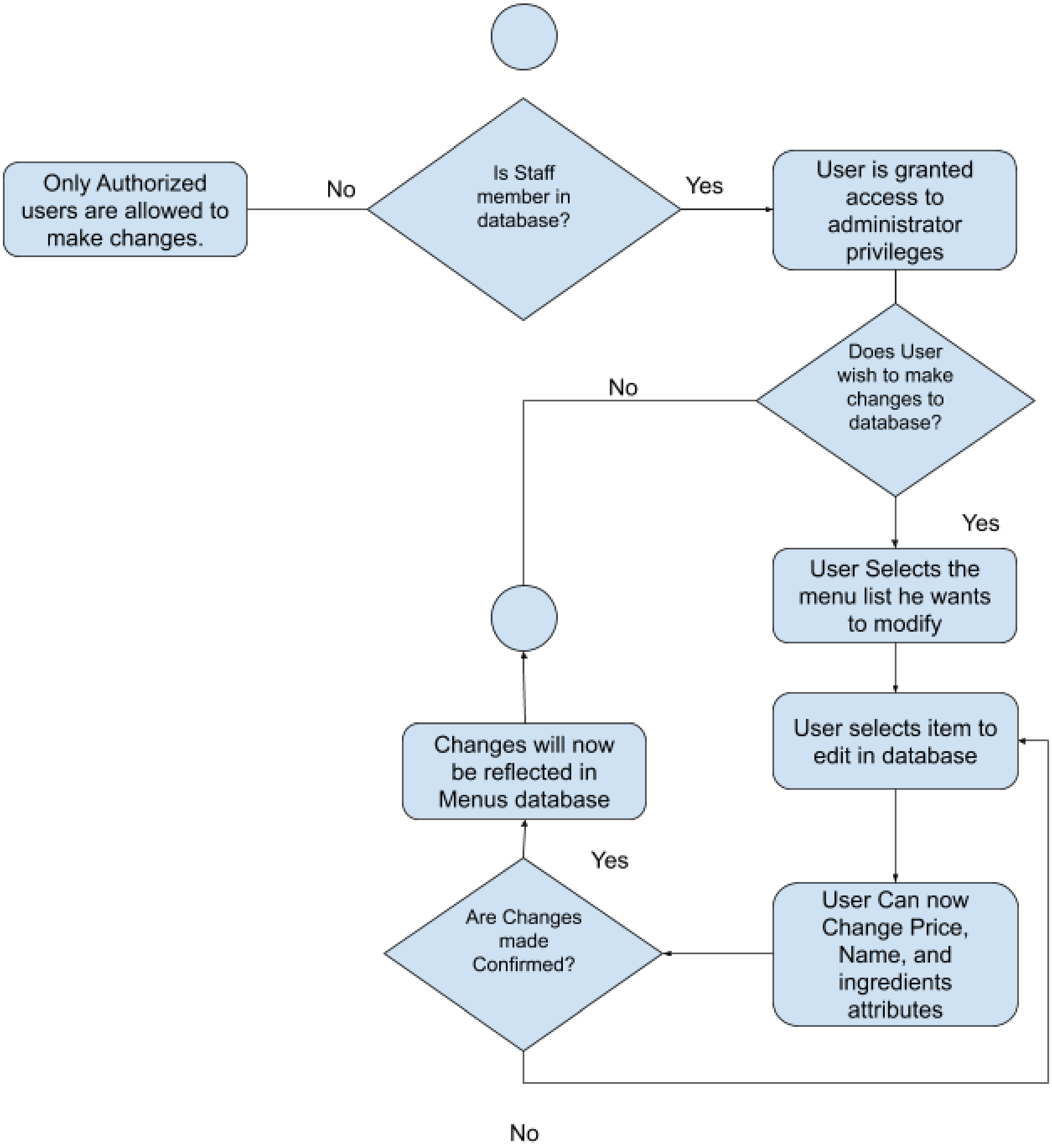
I.D and Name: Edit Menu Items	
Created by:	Jose Miranda 11/15/2020
Primary Actor:	Cafeteria staff
Description:	Select members of the cafeteria staff will have access to modify the cafeteria menu. This includes renaming, changing prices or updating item ingredients
Trigger:	After verifying credentials, staff will have access to modify menu selections.
Preconditions:	Pre-1 Must be part of cafeteria staff (verified) P-2 Users credentials are authorized to make changes
Postconditions:	Post-1 Menu modifications must be in proper format Post-2 Menu is updated in user view
Normal flow:	**Modify menu items** 1. User signs in with credentials 2. User selects Modify Menu 3. User specifies in what menu is the item he wants to edit ***A.) Beverages*** ***B.) Meals*** ***C.)Sides*** ***D.).....*** 5. Menu specified will be displayed and numbered 6.) if user wishes to modify menu he can hit the modify option that will prompt user to enter the number of item he wishes to modify 7.) user will be prompted to choose from

	selection ***A.) Edit Name*** ***B.) Change Price*** ***D.) Update Ingredients*** 8 menu will be presented one final time where users will be asked to confirm or cancel changes.
Exceptions:	Item not in correct format 4.) Message will display information user if incorrect format 5.) User will be prompted to reenter entry 6.) Normal flow will continue
Priority:	High
Frequency of use:	Approximately once a day to update prices
Other Information:	The system must be able to set items not available when sold out.
Assumptions:	Items will be updated prior to start of cafeteria

Edit Menu Items Activity Flow Diagram


Only Authorized users are allowed to make changes.
No
Is staff member signed in and authorized to make changes?
Yes
User is granted access to administrator privileges
No
Would you like to make changes to menu items?
Yes
Menu item is updated
User Selects Menu he wants to edit
User Selects Menu Item he wants to edit
Yes
Yes
Save changes made?
Is entry in correct format?
User can now modify name.price and ingredients
No
User is prompted to re-enter in correct format
No

Edit Menu Items Data Flow Diagram

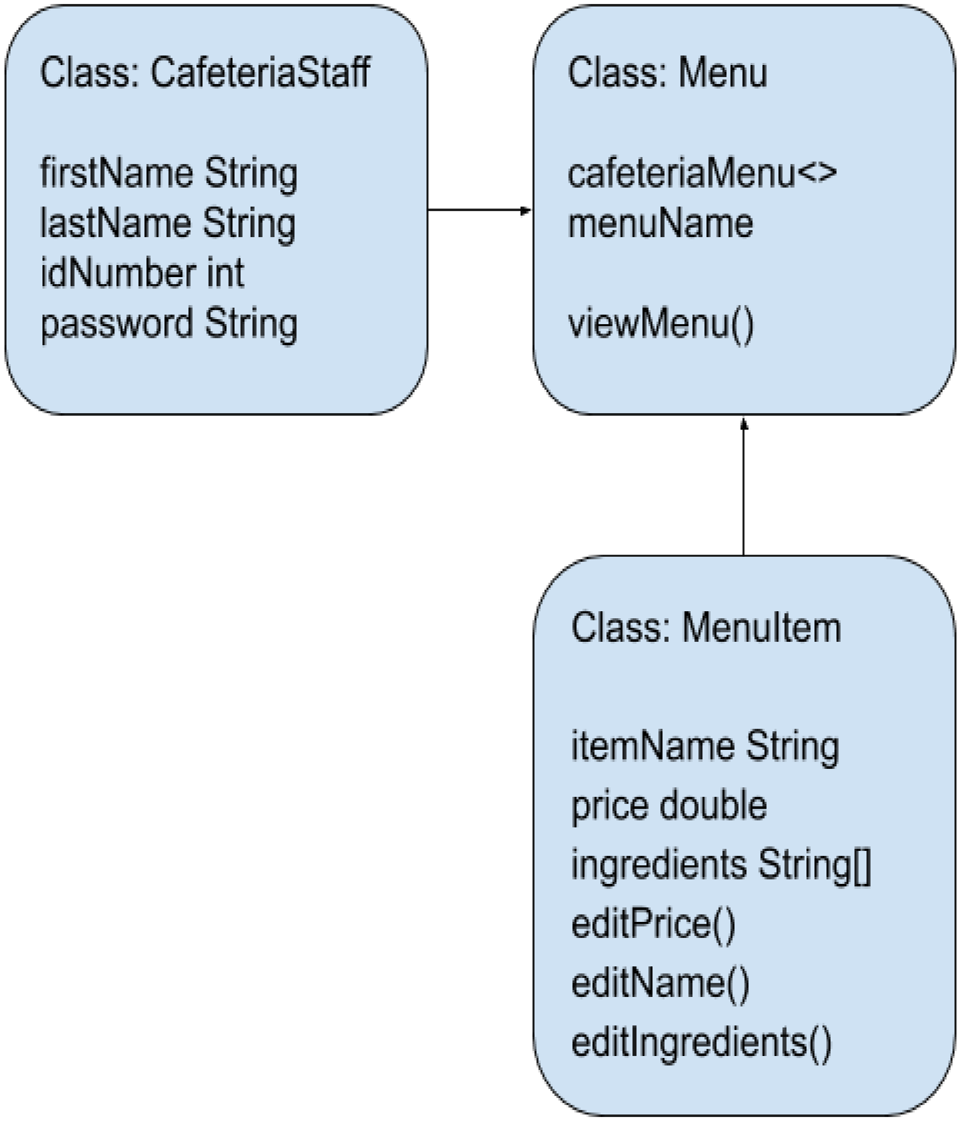

Only Authorized users are allowed to make changes.
No
Is Staff member in database?
Yes
User is granted access to administrator privileges
No
Does User wish to make changes to database?
Yes
User Selects the menu list he wants to modify
User selects item to edit in database
Changes will now be reflected in Menus database
Yes
Are Changes made Confirmed?
User Can now Change Price, Name, and ingredients attributes
No

The diagrams above showcase the process of editing a menu item. This feature is reserved for select members of the Cafeteria staff.

After verifying that the user is authorized to make changes to the menu, the user is granted administrative privileges. then the user can navigate to the edit menu tab where they can make changes to the menu.

The user will select the menu he/she wants to edit then specifies the item that will be changed. The user will have the ability to change prices, names, and ingredients. After the changes have been made the application will check for format errors. If none are present the user will then need to confirm the changes made before the menu is updated . Otherwise the user will need to reenter the item entry to ensure the proper format is used.

Edit Menu Items UML



Cafeteria Staff- the cafeteria class object carries cafeteria staff members information. This includes their first and last name. There employee number and there account password

Menu- this class carries a list containing the menu items for every menu category/name. This provides the functionality to create a menu and remove an entire menu from the cafeterias database along with its items

MenuItem- this object contains the information for every menu item including the attributes name,price and ingredients. This gives the user functionality to edit the attributes of all items.

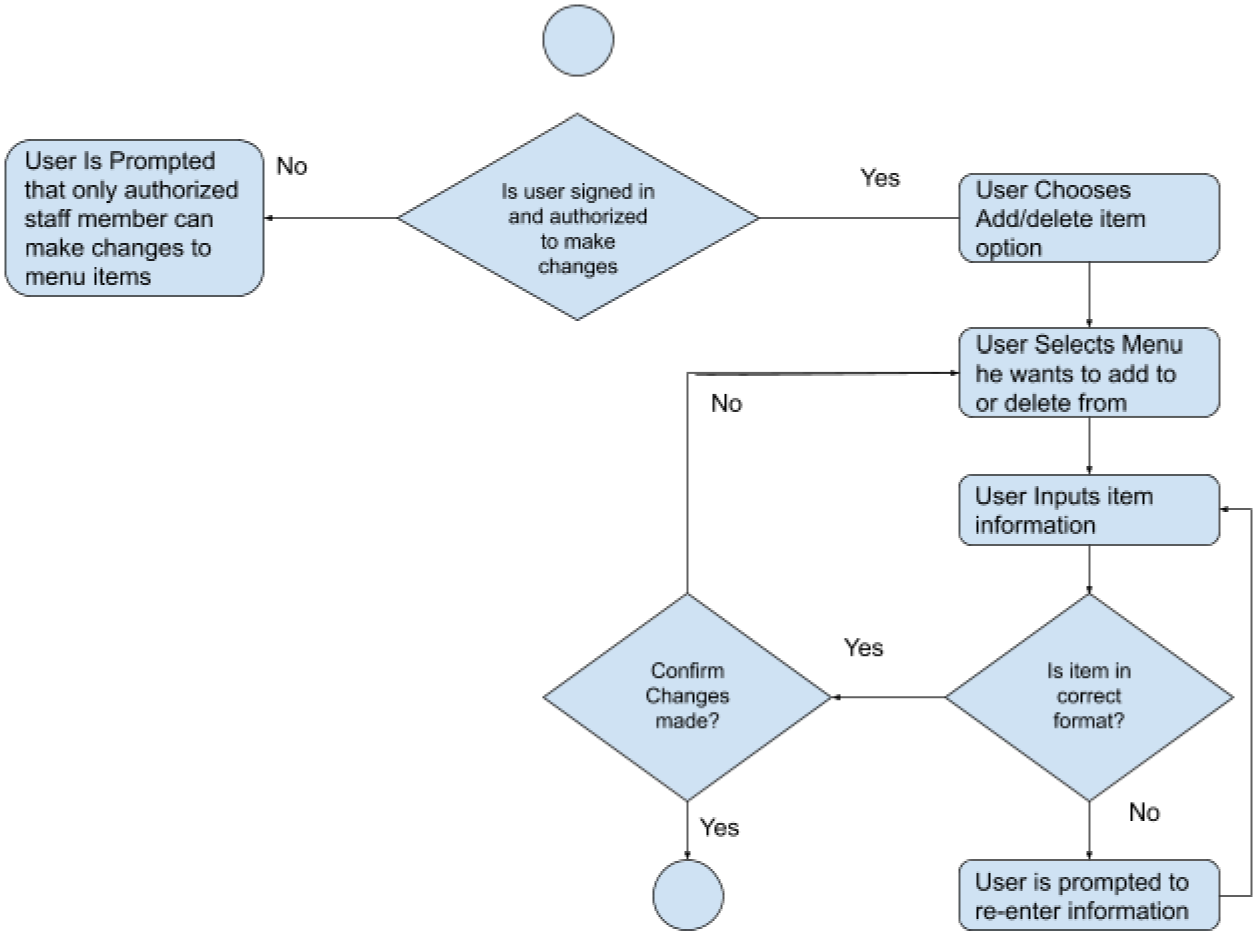
Use Case: Add/Delete Menu Items

The following table showcases the details for the Cafeteria Menu Services:Add/Delete menu items Use Cases.

I.D and Name: Add/Delete menu items	
Created by:	Jose Miranda 11/15/2020
Primary Actor:	Cafeteria staff
Description:	Select members of the cafeteria staff will have access to add and delete items to menus
Trigger:	After verifying credentials, staff will have access to modify menu selections.
Preconditions:	Pre-1 Must be part of cafeteria staff (verified) P-2 Users credentials are authorized to make changes
Postconditions:	Post-1 Menu additions must be in proper format Post-2 Menu is updated in user view
Normal flow:	**Add /remove menu items** 1. User signs in with credentials 2. User selects Modify menu 3. User selects from following selections ***A.) View/Add/remove/ modify items from menu*** ***B.) Create/ delete entire menu*** 4. select what menu user wants to add, remove or modify items from. 5. Menu specified will be displayed and numbered 6.) if user wishes to modify menu he can hit the modify option that will prompt user to enter the number of item he wishes to modify 7.) user will be prompted to choose from selection

	A.) Add ***B.) Remove*** ***D.)Edit item*** ***D.)Cancel*** 8.) menu will be presented one final time where users will be asked to confirm or cancel changes.
Exceptions:	Item not in correct format 1.) Message will display information user if incorrect format 2.) User will be prompted to reenter entry 3.) Normal flow will continue
Priority:	High
Frequency of use:	Approximately once a day to update specials and menu items.
Other Information:	The system must be able to set items not available when sold out.
Assumptions:	Items will be updated prior to start of cafeteria

Add/Delete Menu Items Activity Flow Diagram

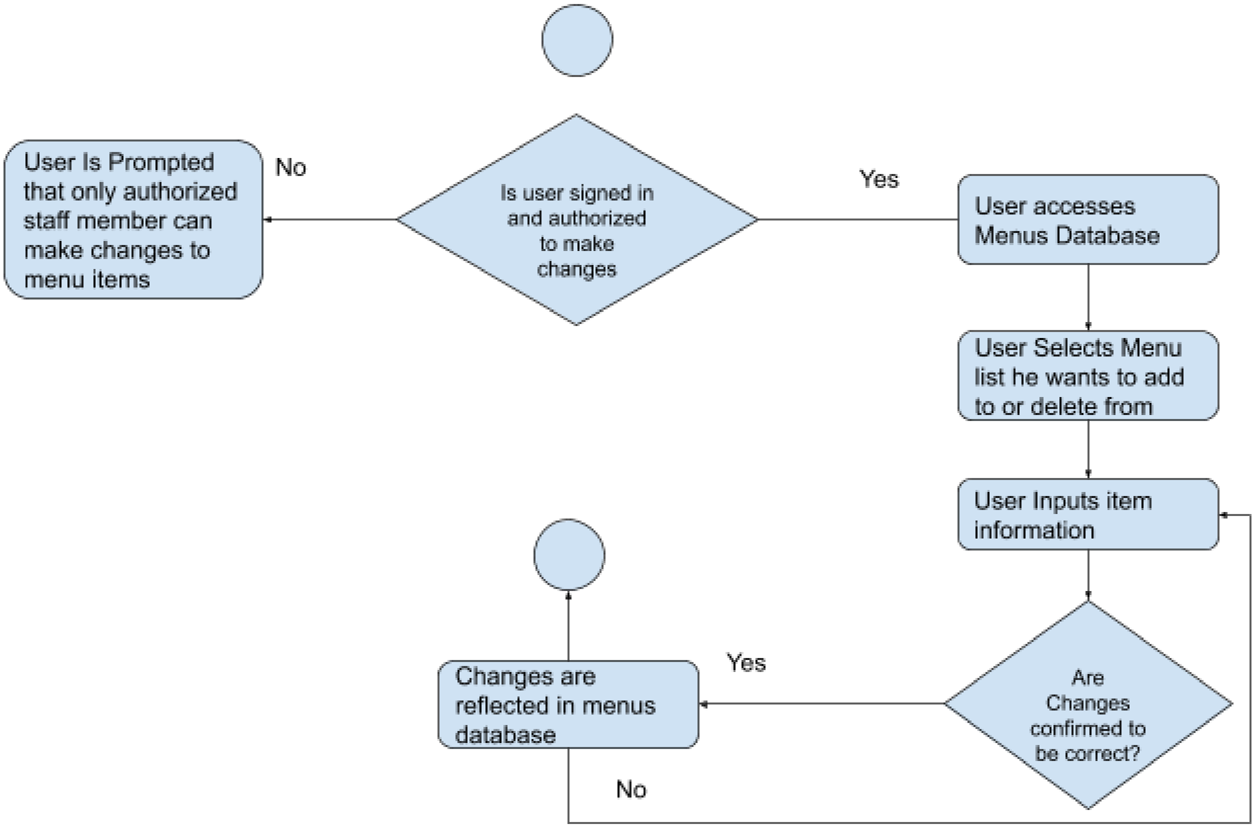

User Is Prompted that only authorized staff member can make changes to menu items
No
Is user signed in and authorized to make changes
Yes
User Chooses Add/delete item option
User Selects Menu he wants to add to or delete from
No
User Inputs item information
Yes
Confirm Changes made?
Is item in correct format?
Yes
No
User is prompted to re-enter information

The diagram above walks through the process of adding or deleting an item from a menu.

After verifying that the user is authorized to make changes they will have the ability to access the add/ delete menu tab. Here the user will select a menu he/she wants to delete or add to the user will then type in the information where then it will be automatically checked for correct format.

Finally the user will need to confirm the changes made before the items will be reflected on the updated menu. If a user cancels changes, then no changes will be made to the menu.

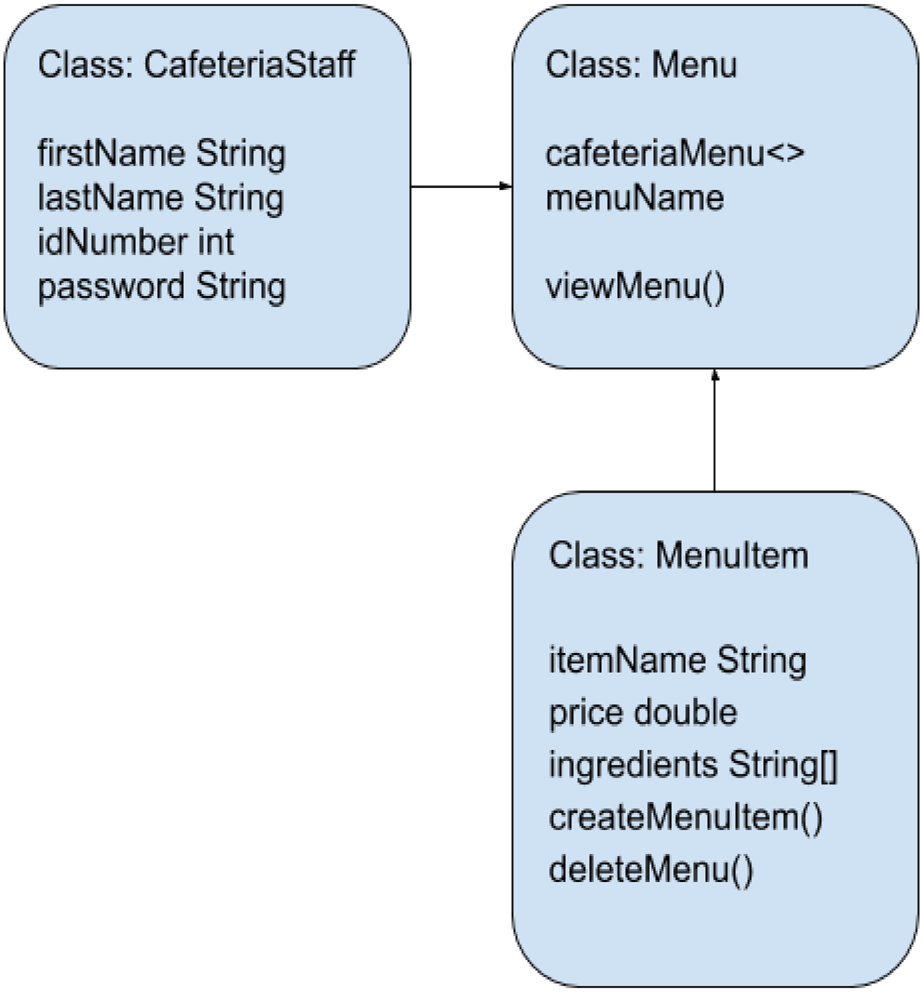
Add/Delete Menu Items Data Flow Diagram


User Is Prompted that only authorized staff member can make changes to menu items
No
Is user signed in and authorized to make changes
Yes
User accesses Menus Database
User Selects Menu list he wants to add to or delete from
User Inputs item information
Yes
Changes are reflected in menus database
Are Changes confirmed to be correct?
No

The diagram above walks you through the data flow of adding or deleting menu items. If an employee is signed in and is also authorized to make changes to the menu he will have access to modifying the items database.

The user will select the menu in which he wishes to place the item and enter the information afterwards the user will need to confirm that the changes made are correct before it is reflected in the database, otherwise the changes will be discarded

Add/delete Menu Items UML

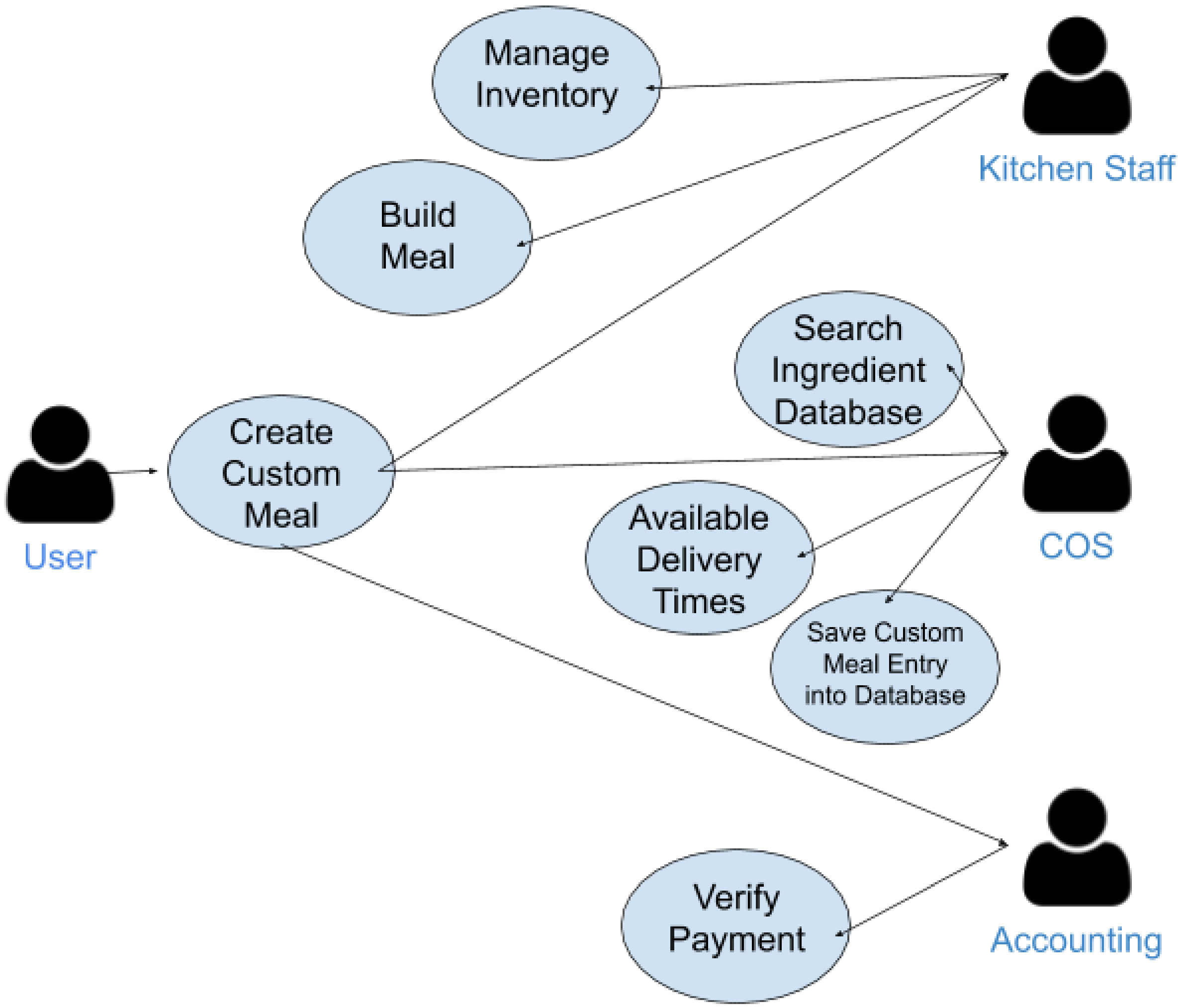


Cafeteria Staff- the cafeteria class object carries cafeteria staff members information. This includes their first and last name. There employee number and there account password

Menu- this class carries a list containing the menu items for every menu category/name. This provides the functionality to create a menu and remove an entire menu from the cafeterias database along with its items

MenuItem- this object contains the information for every menu item as well as providing a way for the user to create a new menu item or delete one.

Custom Meal Services



7.1 Select Delivery Options

The user is able to select where they would like their meal to be delivered to, if at all. They are presented with this option as a radio button which is defaulted to the cafeteria, however when switched from pickup to delivery, a form is displayed asking for the user to input the delivery destination.

7.2 Add/Remove Item to Custom Order

The user is able to add or remove items to their order by selecting either the green button with the plus sign on it for adding an item, or the red button with the minus sign on it for removing a particular item. If the user chooses to add an item to the custom order, then they are presented with a search bar which queries the input, after the input has been validated.

7.3 Add/Remove Directions/Steps to Custom Order

The user is able to add or remove steps to their order by selecting either the green button with the plus sign on it for adding a step, or the red button with the minus sign on it for removing a particular step. If the user chooses to add a step to the custom order, then they are presented with a text box to record the user's input.

7.4 Cancel Custom Order

Should the employee want to cancel their Custom Meal order before submitting, they have the ability to do so by navigating to the Cancel Order button on the main Custom Order screen. The employee will be displayed a page stating the successful cancellation of the current order, and then taken back to the main COS page.

7.5 View Order Details

The employee is able to view their current order details, prices, items, delivery destination, and order status by selecting the Order Details tab.

Use Case: Order custom meals that aren’t on the cafeteria menu

The following table showcases the details of the use case for ordering custom meals that are not currently on the cafeteria menu.

ID and Name:	Order custom meals that aren’t on the cafeteria menu.		
Created By:	Jose Miranda, Christian Ramirez	Date Created:	11/15/2020
Primary Actor:	User	Secondary Actors:	Cafeteria Inventory System, Kitchen Staff, Accounting
Description:	A user accesses the Cafeteria Ordering System from the corporate intranet or from home, chooses a specific date and time for delivery or pickup, adds food items to the list of ingredients, adds instructional steps for the meals preparation, and places an order for a meal to be delivered to a specified location.		
Trigger:	A user indicates that they want to create a custom meal		
Preconditions:	1. The user has successfully logged into the COS.		

Basic Steps:
Normal Flow:

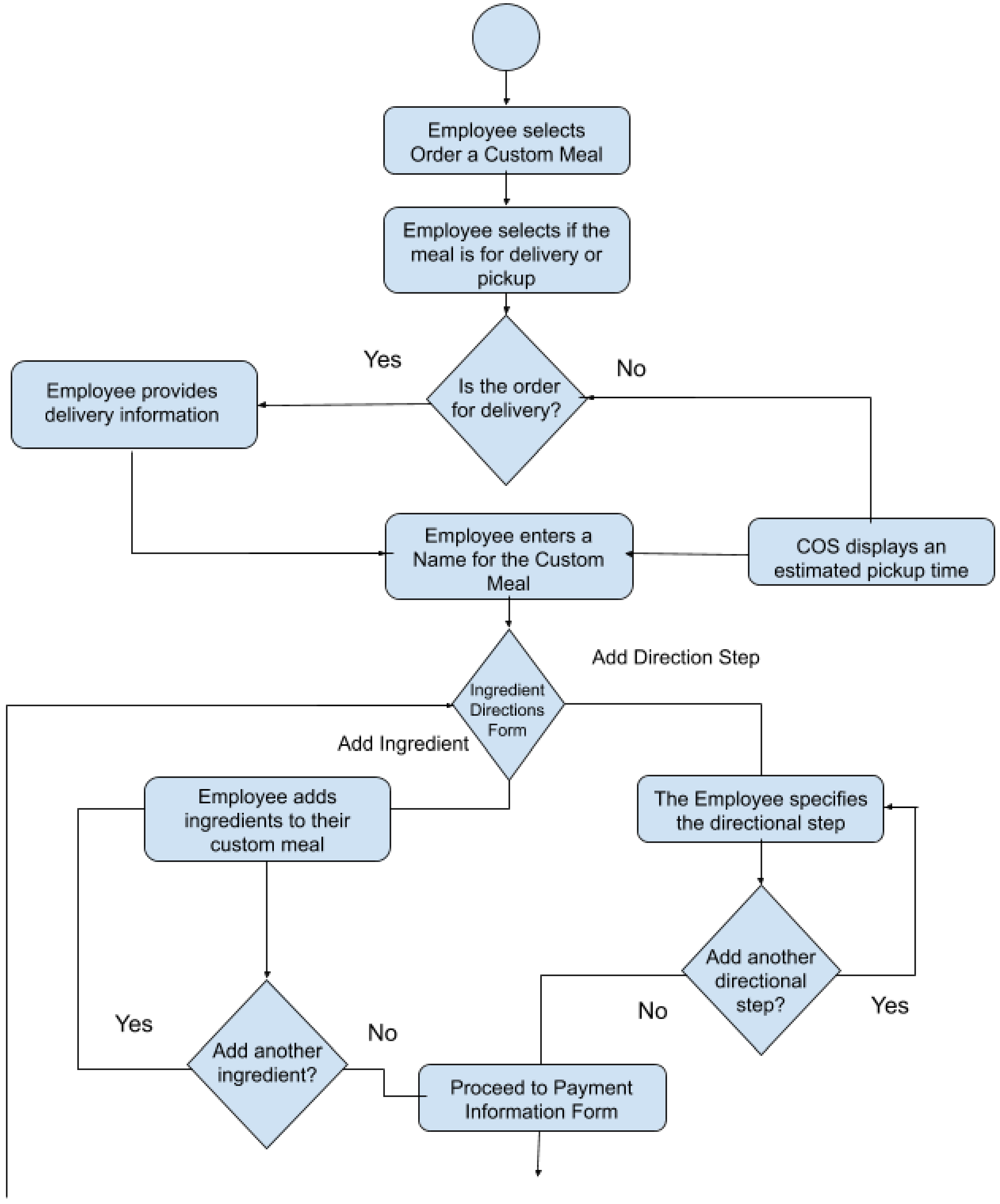
1.0 Order a Single Custom Meal

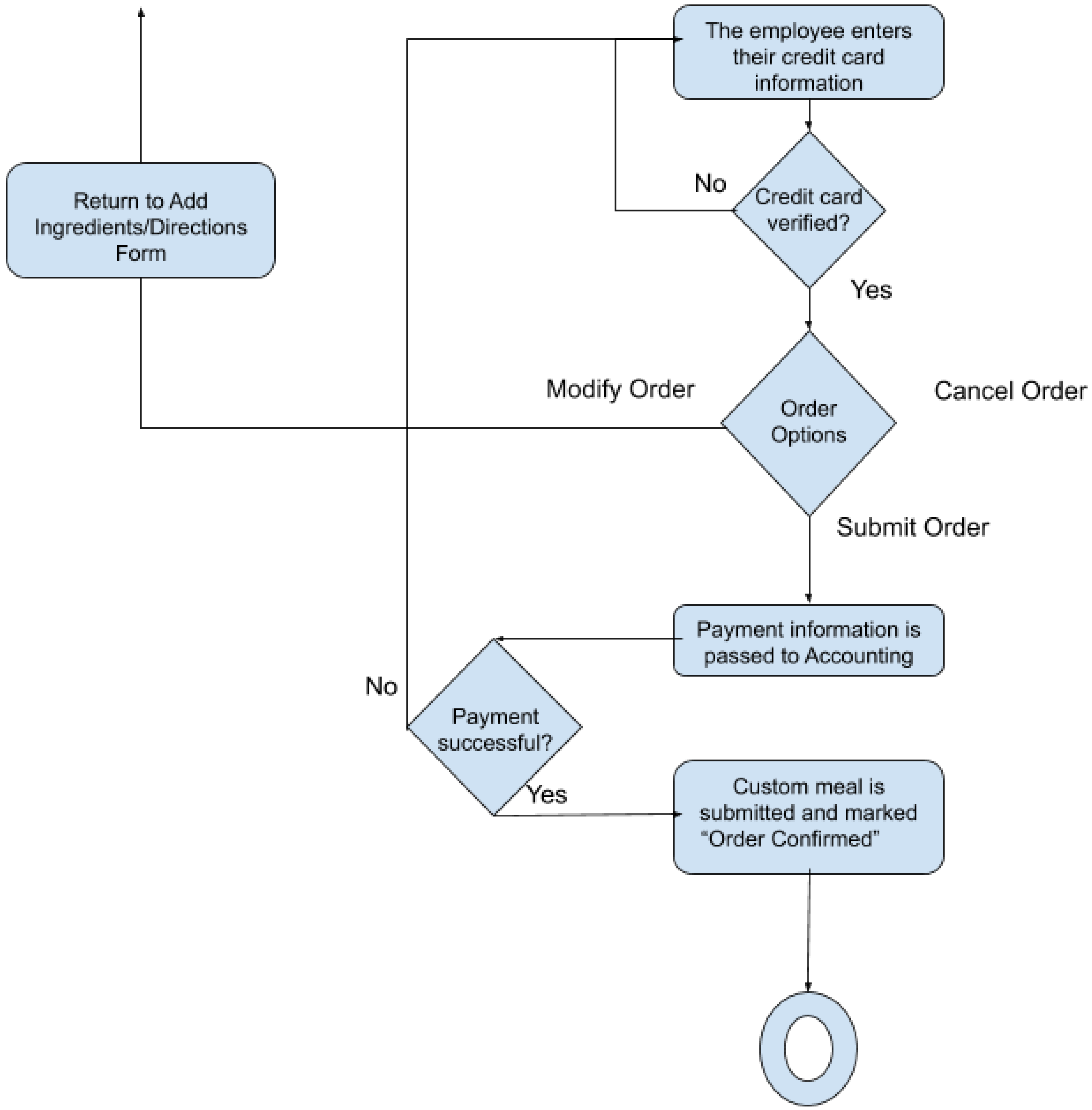
1. The use case begins when the user selects Order a Custom Meal
2. The user is prompted to select whether the order is for pickup or for delivery. If the order is for delivery, the user is asked to supply the delivery address.
3. The user is then asked to supply the delivery date
4. COS displays the available times for delivery on the specified date
5. The user selects a delivery time
6. The user enters a Name for the Custom Meal
7. The user clicks on “Add Ingredient” to create a new ingredient entry
8. The user specifies the ingredient name and quantity (1.0.E.1)
9. The system displays the price of the specified ingredient based on quantity
10. The system keeps a running “TOTAL” of items ordered as they are entered
11. The user either moves to the directions section (continue normal flow) or continues adding ingredients (return to step 7) (1.0.E.2)
12. The user clicks on “Add Direction Step” to create a new direction step entry
13. The user specifies the direction step information
14. The user either moves to the Payment Info section by selecting “Proceed to Payment” (continue normal flow) or continues adding directions (return to step 12) (1.0.E.3)
15. The user enters their credit card information
16. The user selects “Submit Order” (continue normal flow) or selects “Modify Order” to change the order details (return to step 6) or selects “Cancel Order” which ends the use case
17. Accounting verifies the credit card information, processes the payment information, saves the order into the database as pending. If incorrect credit card information is detected, the system will ask the user to make corrections (returns to step 15) (1.0.E.4)
18. If payment is successful, the custom order is marked “Order Confirmed,” an identification number is generated and passed back to the user, and the use case ends. If payment is not successful, the user is prompted to correct the information (returns to step 15) or selects “Cancel Order”, at which point the use case ends.

Postconditions:	1. If the custom order is not canceled by the user, it is saved into the system database and marked “Order Confirmed” 2. The cafeteria inventory is updated to reflect the items in the current order
Exceptions:	**1.0.E.1 Added ingredient is not in stock** 1. COS tells the user that the ingredient being added is not in the inventory of available ingredients 2. If the user cancels the Custom Meal Order, the use case is terminated by COS 3. Else the user can try specifying another ingredient **1.0.E.2 User did not specify at least one ingredient** 1. COS tells the user that there must be at least one ingredient added to the custom meal 2. If the user cancels the Custom Meal Order, the use case is terminated by COS 3. Else the user can try specifying an ingredient **1.0.E.3 User did not specify at least one directional step** 1. COS tells the user that there must be at least one directional step added to the custom meal 2. If the user cancels the Custom Meal Order, the use case is terminated by COS 3. Else the user can try specifying a directional step **1.0.E.4 Payment verification failed** 1. COS tells the user that the credit card information supplied could not be processed 2. If the user cancels the Custom Meal Order, the use case is terminated by COS 3. Else the user can try specifying payment information
Priority:	High
Frequency of Use:	Multiple instances per day

Other Information:	1. The user can cancel the order any time before confirming the order

Activity Diagram


Employee selects Order a Custom Meal
Employee selects if the meal is for delivery or pickup
Yes
Is the order for delivery?
No
Employee provides delivery information
Employee enters a Name for the Custom Meal
COS displays an estimated pickup time
Add Direction Step
Ingredient Directions Form
Add Ingredient
Employee adds ingredients to their custom meal
The Employee specifies the directional step
Add another directional step?
No
Yes
Yes
Add another ingredient?
No
Proceed to Payment Information Form

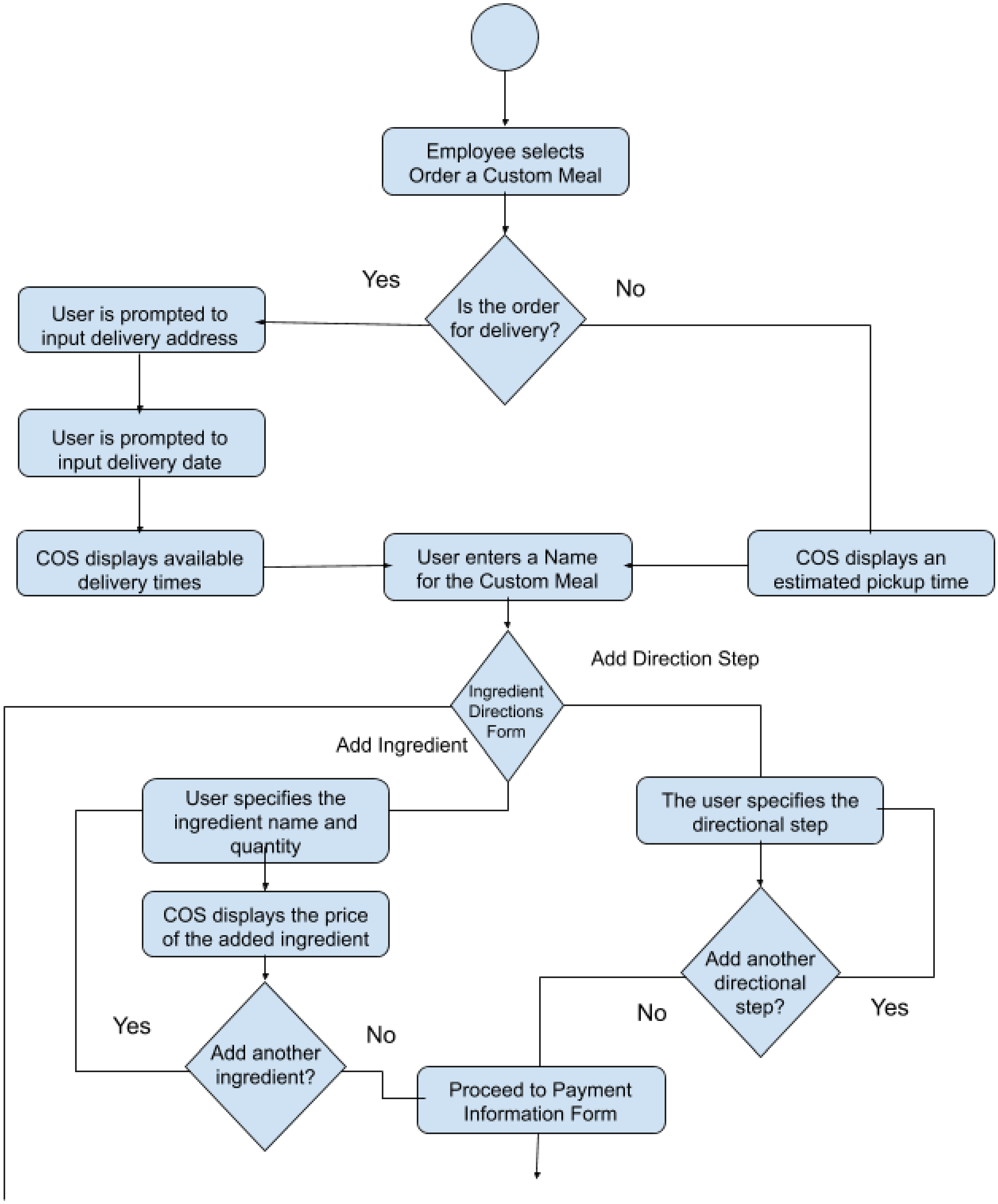

The employee enters their credit card information
No
Credit card verified?
Yes
Return to Add Ingredients/Directions Form
Modify Order
Order Options
Cancel Order
Submit Order
Payment information is passed to Accounting
No
Payment successful?
Yes
Custom meal is submitted and marked "Order Confirmed"

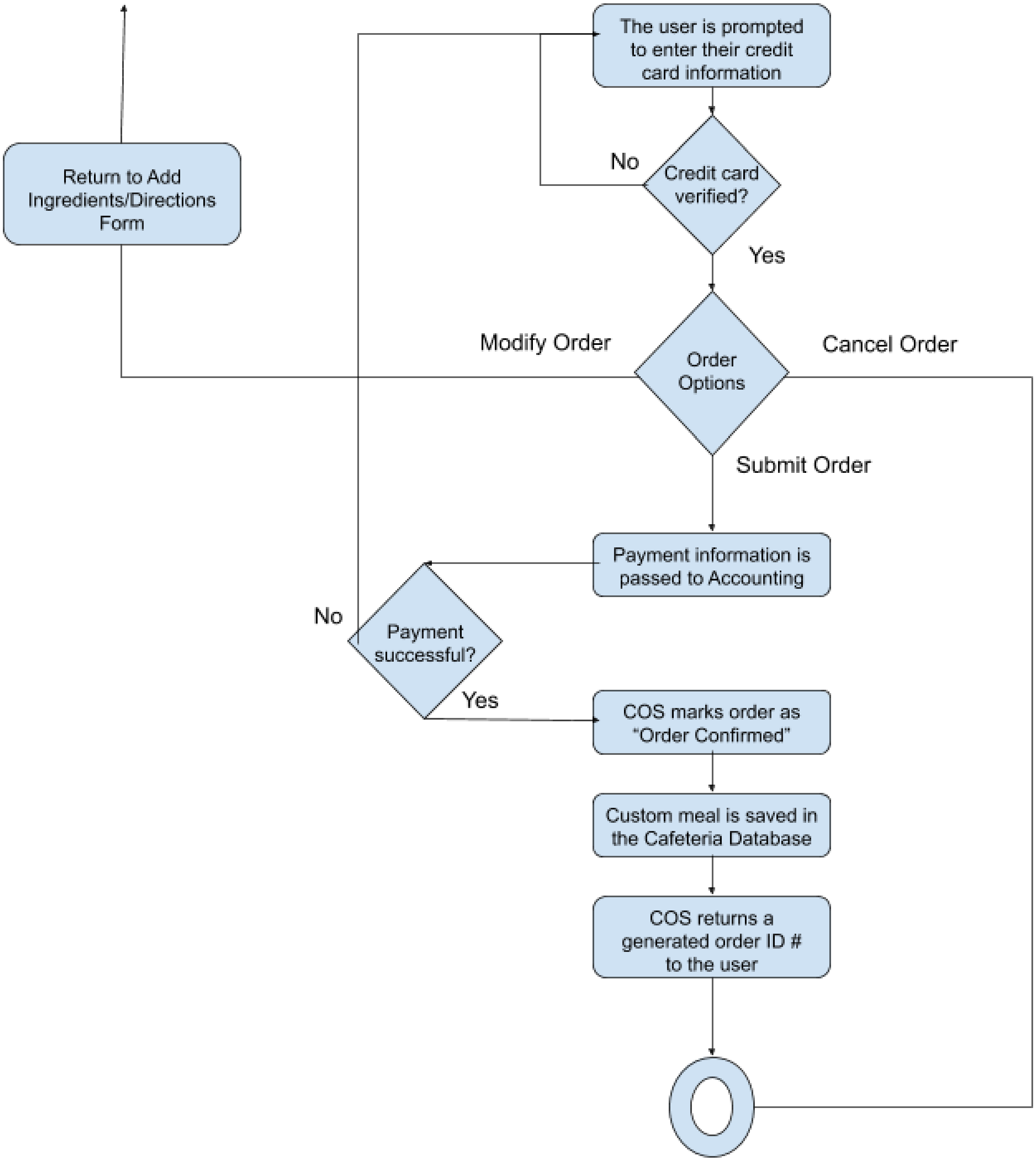
The above diagram outlines the process of creating a custom meal for order.

The employee is first asked to decide whether their order is for delivery or not. After deciding on delivery, the employee is able to create a new name for their custom meal. They then have the ability to add or items from the cafeteria database to their custom order. Likewise, the employee is able to add directional steps which specify how the meal should be prepared. When the user is finished adding steps and/or ingredients, they are prompted to proceed to entering their credit card information.

Upon successful verification of this information, COS will inform the user, “Order Confirmed.” Returning an order identification number to the employee, and finally, saving the custom meal into the custom menu database. If the user decides to cancel the order any time before receiving their confirmed order identification number, then no changes will be made anywhere.

Data Flow Diagram


Employee selects Order a Custom Meal
Is the order for delivery?
Yes
No
User is prompted to input delivery address
User is prompted to input delivery date
COS displays available delivery times
User enters a Name for the Custom Meal
COS displays an estimated pickup time
Ingredient Directions Form
Add Direction Step
Add Ingredient
User specifies the ingredient name and quantity
The user specifies the directional step
COS displays the price of the added ingredient
Add another directional step?
No
Yes
Yes
Add another ingredient?
No
Proceed to Payment Information Form

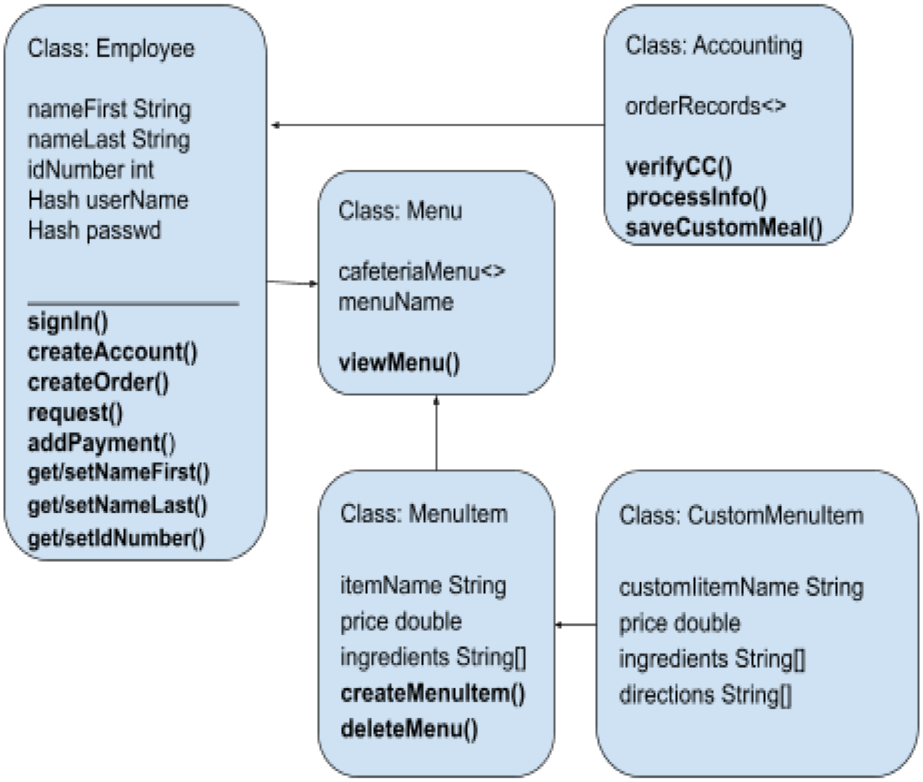

The user is prompted to enter their credit card information
Credit card verified?
No
Yes
Return to Add Ingredients/Directions Form
Modify Order
Order Options
Cancel Order
Submit Order
Payment information is passed to Accounting
No
Payment successful?
Yes
COS marks order as "Order Confirmed"
Custom meal is saved in the Cafeteria Database
COS returns a generated order ID # to the user

The above diagram outlines the data flow when creating a custom meal for order.

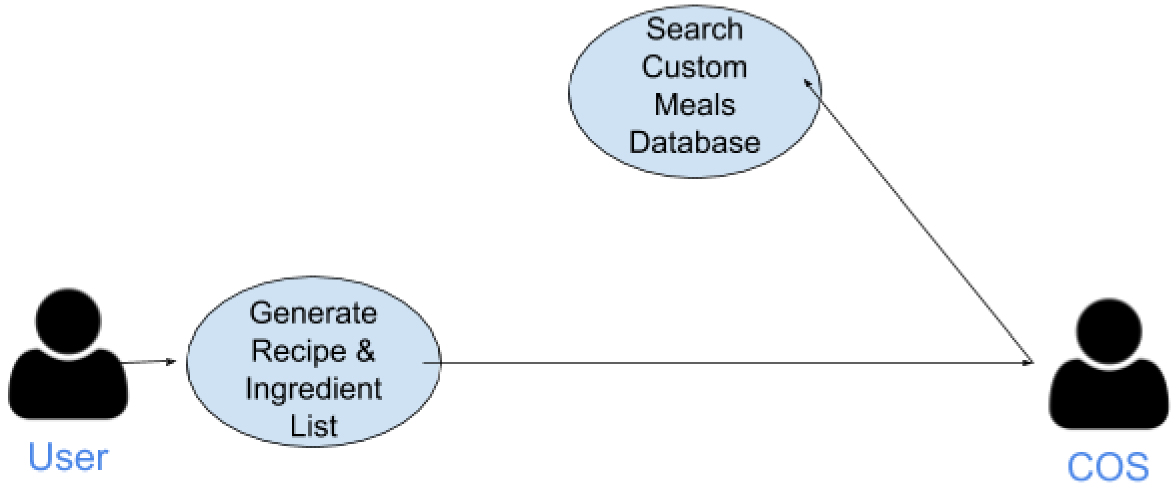
The employee must first input whether their order is for delivery or not. This information is then passed to COS which displays the corresponding delivery or pickup time windows. The employee then passes a new name for their custom meal to COS. Ingredients are then added or removed from the custom order, being temporarily stored in the database. Likewise, directional steps are added or removed from the custom order, being temporarily stored in the database. The employee then passes COS their credit card information, which is then verified by accounting. If the information is erroneous, the employee is brought back to the form for entering their credit card information.

Upon successful verification of this information, COS will inform the user, “Order Confirmed.” Returning an order identification number to the employee, and finally, saving the custom meal into the custom menu database. If the user decides to cancel the order any time before receiving their confirmed order identification number, then no changes will be made anywhere.

UML


Class: Employee
nameFirst String
nameLast String
idNumber int
Hash userName
Hash passwd
signIn()
createAccount()
createOrder()
request()
addPayment()
get/setNameFirst()
get/setNameLast()
get/setIdNumber()
Class: Accounting
orderRecords<>
verifyCC()
processInfo()
saveCustomMeal()
Class: Menu
cafeteriaMenu<>
menuName
viewMenu()
Class: MenuItem
itemName String
price double
ingredients String[]
createMenuItem()
deleteMenu()
Class: CustomMenuItem
customlitemName String
price double
ingredients String[]
directions String[]

Recipe/Ingredient List Builder


Search Custom Meals Database
Generate Recipe & Ingredient List
User
COS

8.1 Search Custom Meals Database

An employee is capable of searching through the database of custom meals that have been created by not only themselves but also by other employees as well. The module will provide access menus and tabs that direct to specific custom menu entry categories in addition to the tab that lists every custom meal created, sorted in descending order according to creation date.

8.2 Generate Recipe & Ingredient List

An employee is capable of selecting a custom meal through the custom meal database and asking for the recipe and ingredients used to compile the custom meal. COS will query the database on behalf of the employee and return the list of ingredients and the attached recipe.

Use Case: Produce recipes and ingredient lists from custom meals from the cafeteria

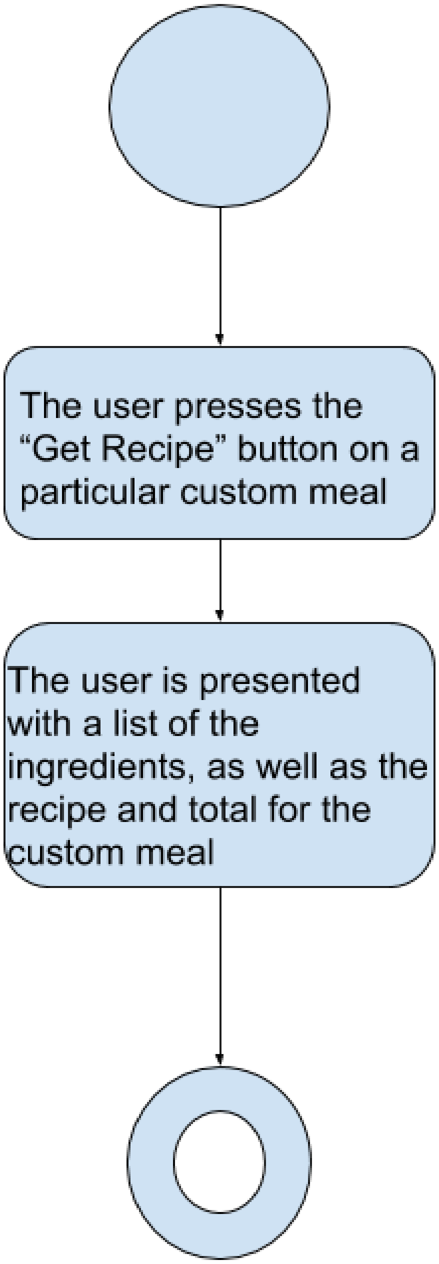
The following table showcases the details of the use case for producing the recipe and ingredient list for custom meals from the cafeteria.

Produce recipes and ingredient lists from custom meals from the cafeteria			
Created by:	Christian Ramirez		
Primary Actor:	User	Secondary Actor:	Cafeteria Ordering System
Description:	A user accesses the Cafeteria Ordering System from the corporate intranet or from home and requests the recipe and ingredients list for a custom item listed in the cafeteria meals database.		
Trigger:	The user indicates that they want to get the recipe and ingredients for a particular custom meal.		
Preconditions:	1. The user has successfully logged into the COS. 2. The custom meal must exist within the cafeteria meals database		
Normal flow:	1. The use case begins when the user selects Get Recipe on a Custom Meal 2. The COS accesses the custom meals database 3. The user is returned the list of ingredients, recipes, and the queried meals price		
Priority:	Low		
Frequency of use:	Multiple instances per day		

Activity Diagram

The above flowchart describes the flow an employee would take to generate the list of ingredients and recipe for a custom meal entry.

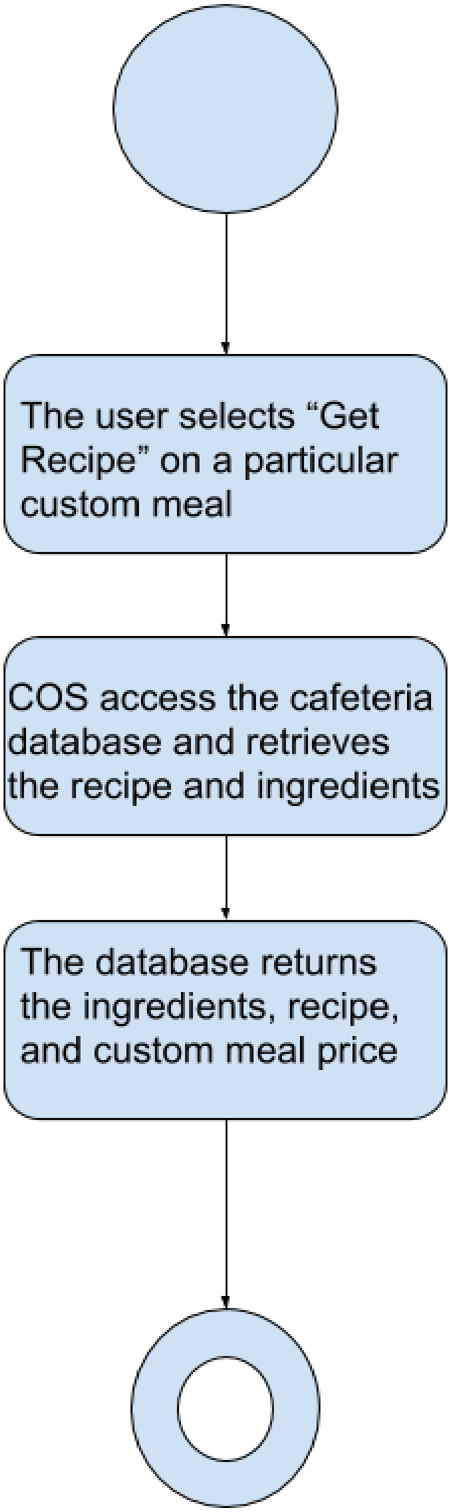
The employee begins by tapping on the “Get Recipe” button which is displayed on any custom meal entry which is in the custom meal database. The requested information will be queried and returned to the employee where they will be able to review the recipe and ingredients for the requested custom meal.


The user presses the “Get Recipe” button on a particular custom meal
The user is presented with a list of the ingredients, as well as the recipe and total for the custom meal

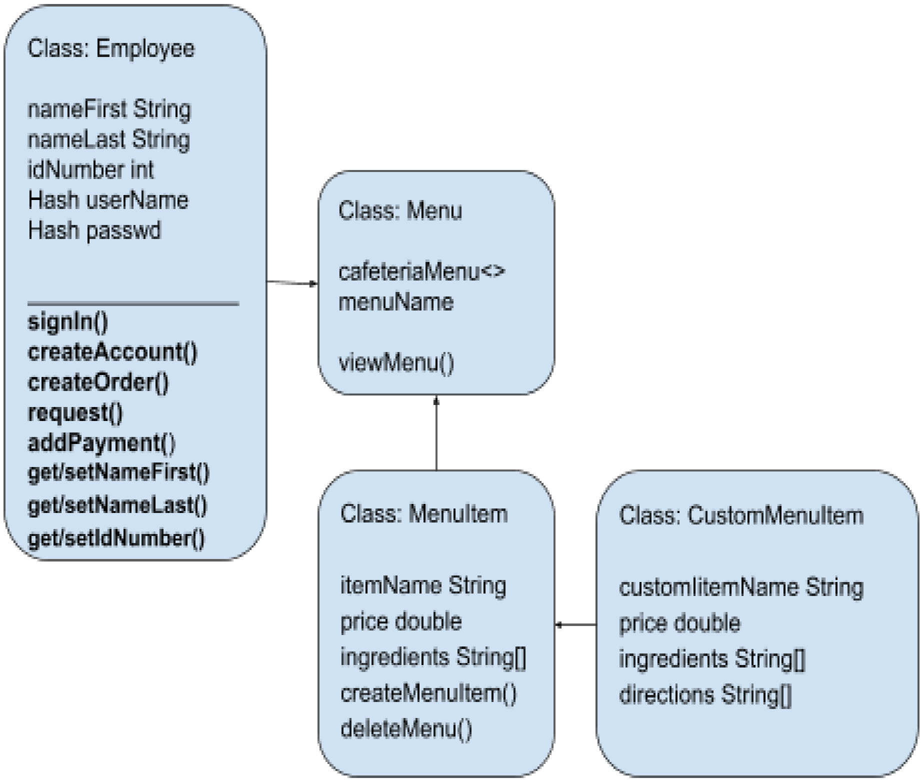
Data Flow Diagram

The above flowchart describes the flow of data that happens when a list of ingredients and recipes are generated for a custom meal entry.

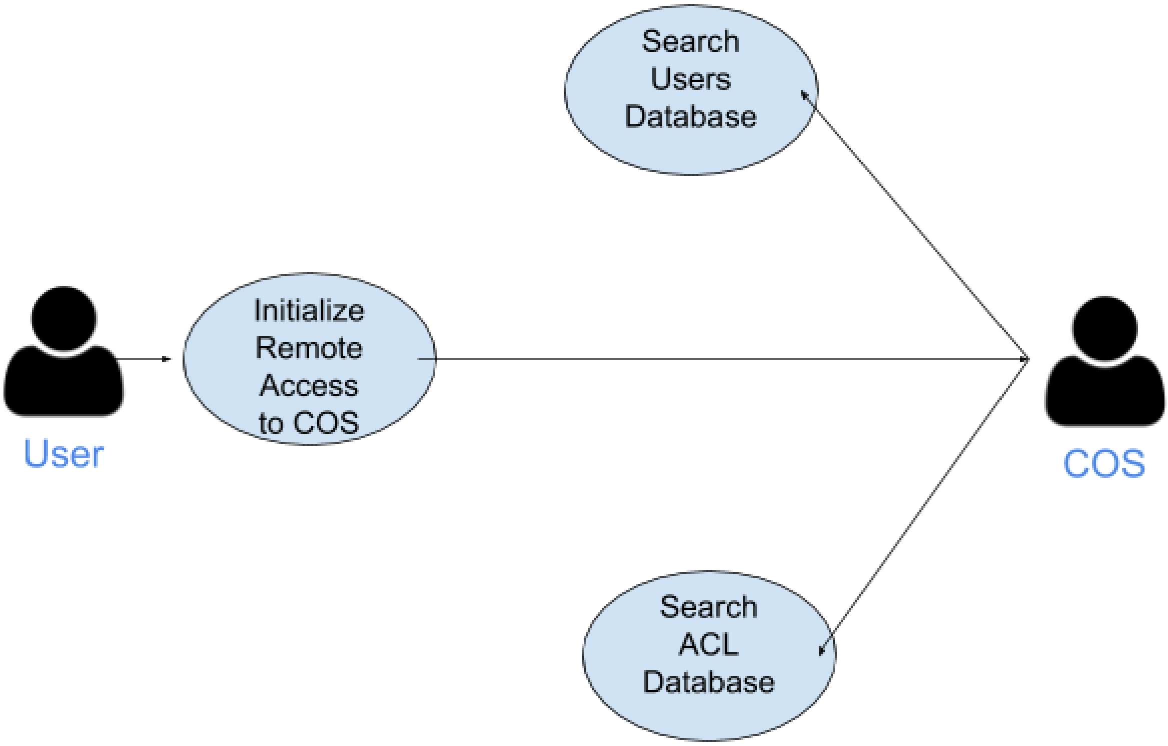
When the employee selects the “Get Recipe” button, COS queries the custom meal database for the particular custom meal. The requested information is then returned to the employee where they will be able to review the recipe and ingredients for the requested custom meal.


The user selects “Get Recipe” on a particular custom meal
COS access the cafeteria database and retrieves the recipe and ingredients
The database returns the ingredients, recipe, and custom meal price

UML Diagram


Class: Employee
nameFirst String
nameLast String
idNumber int
Hash userName
Hash passwd
signIn()
createAccount()
createOrder()
request()
addPayment()
get/setNameFirst()
get/setNameLast()
get/setIdNumber()
Class: Menu
cafeteriaMenu<>
menuName
viewMenu()
Class: MenuItem
itemName String
price double
ingredients String[]
createMenuItem()
deleteMenu()
Class: CustomMenuItem
customlitemName String
price double
ingredients String[]
directions String[]

Remote System Access



9.1 Remote Portal

Should an employee wish to connect to the Cafeteria Ordering System remotely, the employee is able to remotely access the Ordering System portal. This page displays the company logo, a field to enter a username, and a field to enter the users password.

9.2 Initialize Remote Access to COS

The employees credentials are entered, hashed, and then compared to the hashes currently stored within the database. If the hashed credentials are found to be in the database then the user is compared against an ACL to further see if the current employee has the permissions to use the Cafeteria Ordering System remotely.

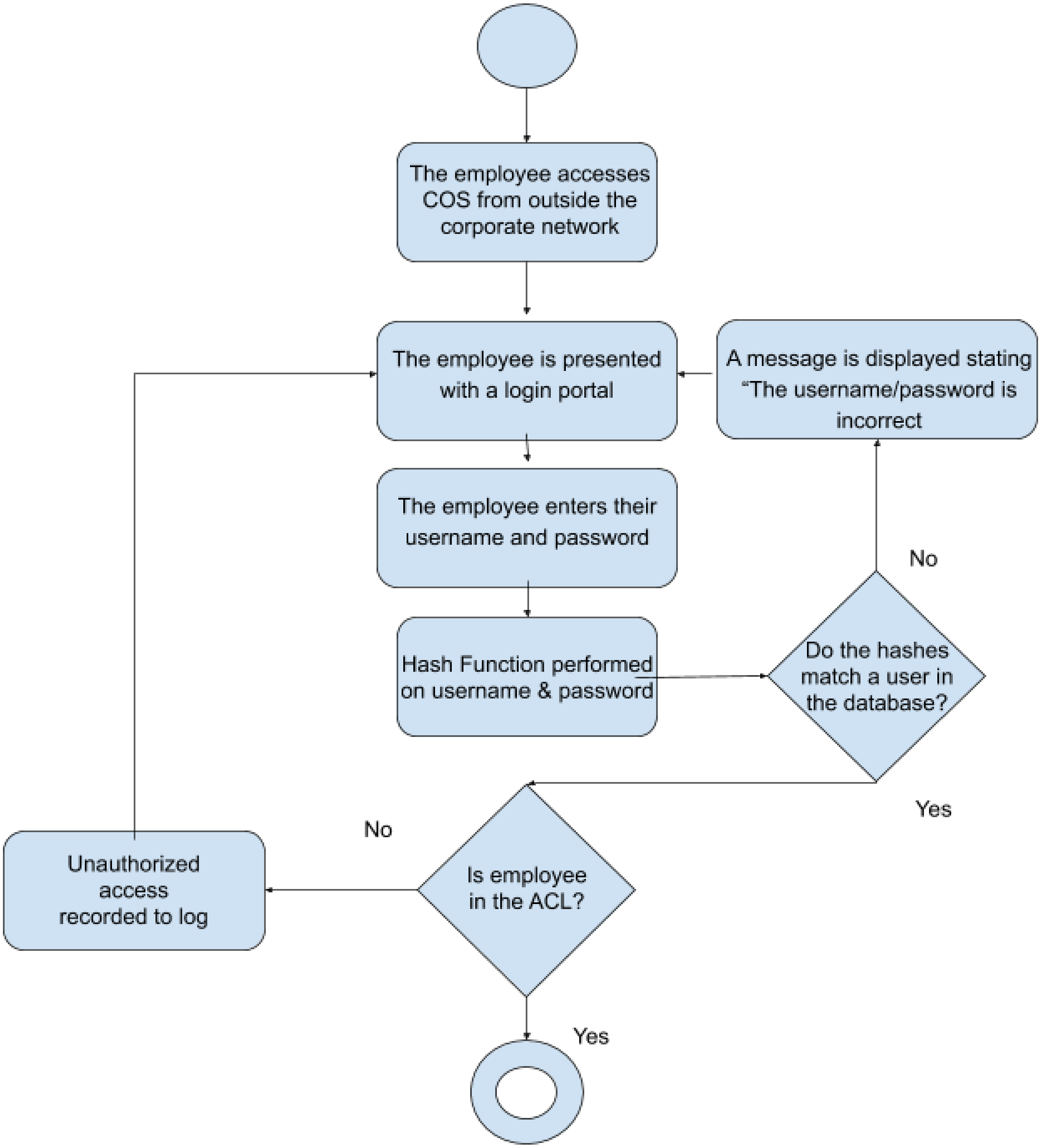
Use Case: Accessing the Cafeteria Ordering System remotely

The following table showcases the details of the use case for providing access to the Cafeteria Ordering System through outside Internet access for authorized employees.

Provide system access through outside Internet access for authorized employees			
Created by:	Christian Ramirez		
Primary Actor:	User	Secondary Actor:	Cafeteria Ordering System
Description:	A user requests accesses to the Cafeteria Ordering System from a remote location		
Trigger:	The user tries to access the COS from outside of the corporate network		
Preconditions:	1. The username password combination is valid and resides within the COS users database 2. The user trying to log on to COS remotely must be authorized to do so		
Normal flow:	1. The use case begins when the user accesses the COS from outside of the corporate network 2. The user is prompted to specify their username 3. The user is prompted to specify their password 4. COS compares the hash of the supplied username and password to hash values stored within the users database 5. If the hashes are equivalent, COS checks the access control list to see if the user is authorized to access COS remotely. If the user is on the ACL, then the user is granted access to the COS, the user login is recorded to a log file, and the use case ends. Otherwise, if the user is not on the ACL, the illegal login attempt will be recorded to the unauthorized access log file, the user is denied access to COS, and the use case ends. (E.2) 6. If the hashes are not equivalent, the user is denied access based on the hash comparison, and the use case ends. (E.1)		

Exceptions:	**E.1 The specified username/password combination is incorrect** 1. COS tells the user that the username password combination is incorrect 2. If the user cancels the login authorization process, the use case is terminated by COS 3. Else the user can try specifying a username and password combination **E.2 The specified username/password combination is incorrect** 4. COS tells the user that the username password combination is incorrect 5. If the user cancels the login authorization process, the use case is terminated by COS 6. Else the user can try specifying a username and password combination
Priority:	High
Frequency of use:	Multiple instances per day

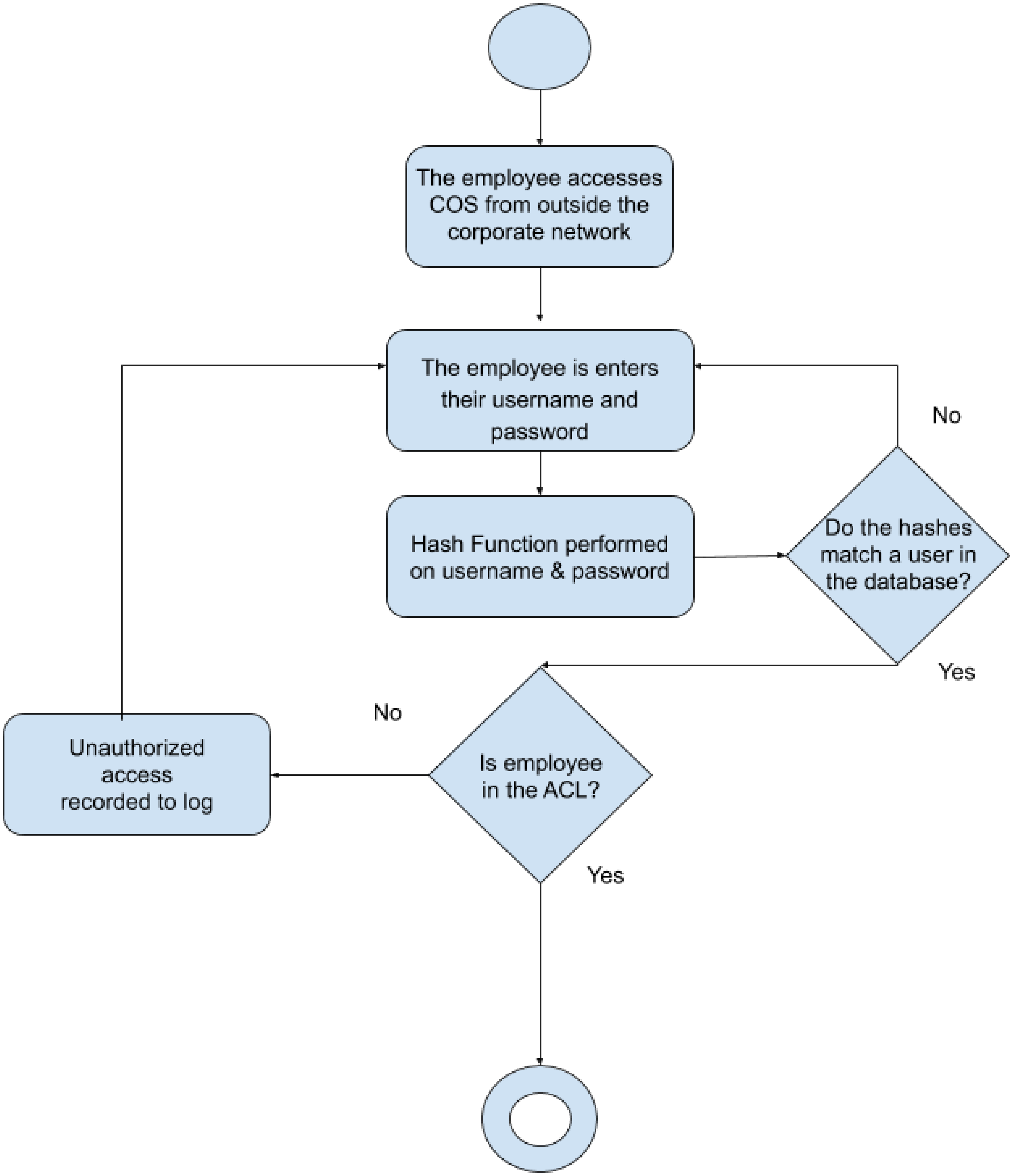
Activity Diagram


The employee accesses COS from outside the corporate network
The employee is presented with a login portal
A message is displayed stating "The username/password is incorrect
The employee enters their username and password
No
Hash Function performed on username & password
Do the hashes match a user in the database?
Yes
No
Unauthorized access recorded to log
Is employee in the ACL?
Yes

The above diagram represents the employees flow when they decide to access COS remotely.

The chart begins when the employee is presented with a login portal when they try to navigate to COS while outside of the corporate network. The employees login credentials will then be run through the hashing function. It will then be compared to the list of current employees to verify the user. If the hashed data is found in the database, the data is then checked to see if the current employee is within the Access Control List (ACL). If the employee is found in this list, they are granted access to COS, otherwise, the login attempt is logged to a file, and a message is displayed stating, “Unauthorized Access has been recorded.”

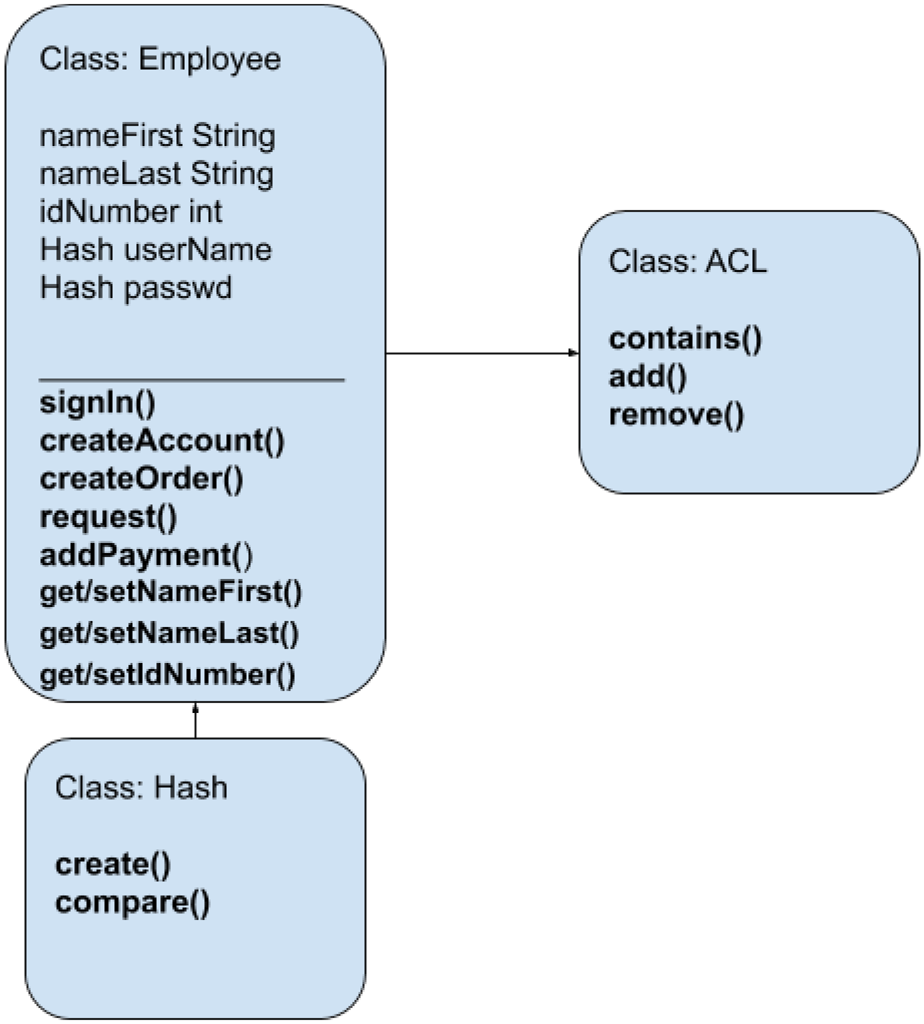
Data Flow Diagram


The employee accesses COS from outside the corporate network
The employee is enters their username and password
No
Hash Function performed on username & password
Do the hashes match a user in the database?
Yes
No
Unauthorized access recorded to log
Is employee in the ACL?
Yes

The above diagram represents the flow of data when an employee decides to access COS remotely.

The employee will provide their username and password used for logging onto COS. The data, both the username and the password, will be run through the hashing function. It will then be compared to the list of current employees to verify the user. If the hashed data is found in the database, the data is then checked to see if the current employee is within the Access Control List (ACL). If the employee is found in this list, they are granted access to COS, otherwise, the login attempt is logged to a file, and a message is displayed stating, “Unauthorized Access has been recorded.”

UML Diagram


Class: Employee
nameFirst String
nameLast String
idNumber int
Hash userName
Hash passwd
signIn()
createAccount()
createOrder()
request()
addPayment()
get/setNameFirst()
get/setNameLast()
get/setIdNumber()
Class: ACL
contains()
add()
remove()
Class: Hash
create()
compare()

Index

Glossary

Activity Diagram**:** An analysis model that depicts a process flow proceeding from one activity to another. Similar to a flowchart.

Actor - A person performing a specific role, a software system, or a hardware device that interacts with a system to achieve a useful goal. Also called a user role.

Administrator- The person of a multi-user computer system also called system admin. The system administrator designs the system and manages its use.

Cafeteria Ordering System(COS) - an Internet Based application that will accept individual or group meal orders, process payments, and trigger delivery of the prepared meals to a designated location on the XYZ Solutions campus.

Data - Information; raw facts. Data can be input into a computer and processed in various ways.

Data Flow Diagram - An analysis model that depicts the processes, data stores, external entities, and flows among them that characterize the behavior of data flowing through business processes or software systems.

Data Models - The particular objects, including their specific properties, that are part of a process.

Database - A large collection of data stored together and organized for rapid search and retrieval.

Exception - A condition that can prevent a use case from concluding successfully. Unless some recovery mechanism is possible, the use case's post-conditions are not reached and the actor's goal is not achieved.

Field - A specific area, within the record, used for a particular category or item, for example, name, zip code, city.

Flowchart - An analysis model that shows the processing steps and decision points in the logic of a process. Similar to an activity diagram.

Input - To enter data into the application.

Modules - An independent piece of software which forms part of one or more larger programs.

Normal Flow - The default sequence of steps in a use case, which leads to satisfying the use case's post-conditions and letting the user achieve his goal. Also known as the normal course, main course, normal sequence, and main success scenario.

Pre-condition - A condition that must be satisfied or a state the system must be in before a use case can begin.

Post-condition - A condition that describes the state of a system after a use case is successfully completed.

Remote Access -Any combination of hardware and software to enable the remote access tools or information that typically reside on a network of IT devices

SQL - Structured Query Language - A language used to interrogate and process data in a relational database.

User - A customer who will interact with a system either directly or indirectly (for example, by using outputs from the system but not generating those outputs personally). Also called end user.

Use Case - A description of a set of logically related possible interactions between an actor and a system that results in an outcome that provides value to the actor. Can encompass multiple scenarios.