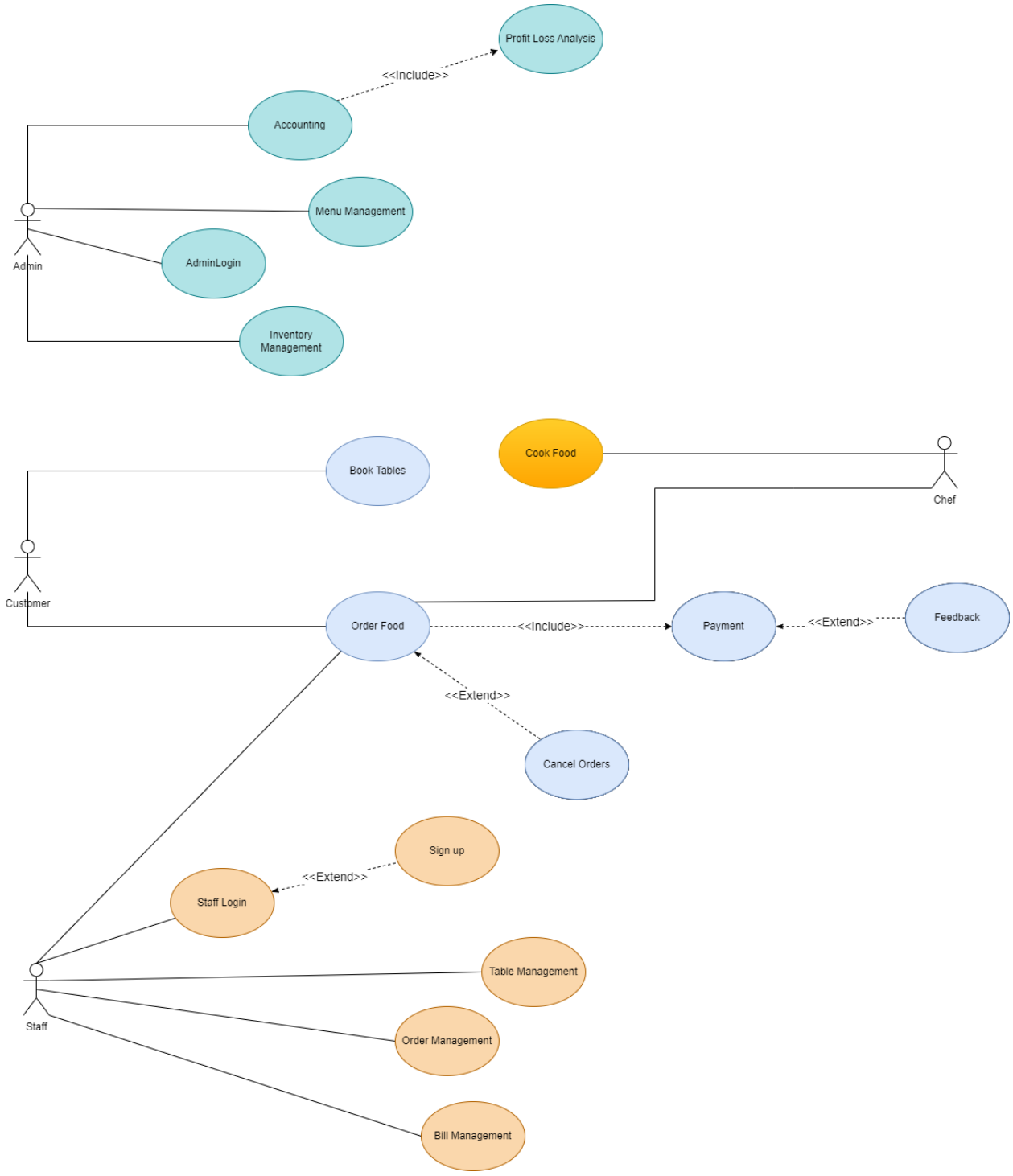
IT 314 - Software Engineering

Lab 3 - Group 14

Date - 23/02/2023

Team Members	Student ID
Pulkit Khandelwal	202001120
Shreyansh Khemesara	202001121
Prajapati Vrutik Navneetbhai(Team Leader)	202001124
Baraiya Dhruv Pravinbhai	202001140
Mandaviya Soham Hiteshbhai	202001142
Shah Nishadhkumar Nirajbhai	202001151
Buddhdev Harsh Nitesh	202001157
Chhagani Krunal Ajaybhai	202001158
Thakor Harshal Dipaksinh	202001169

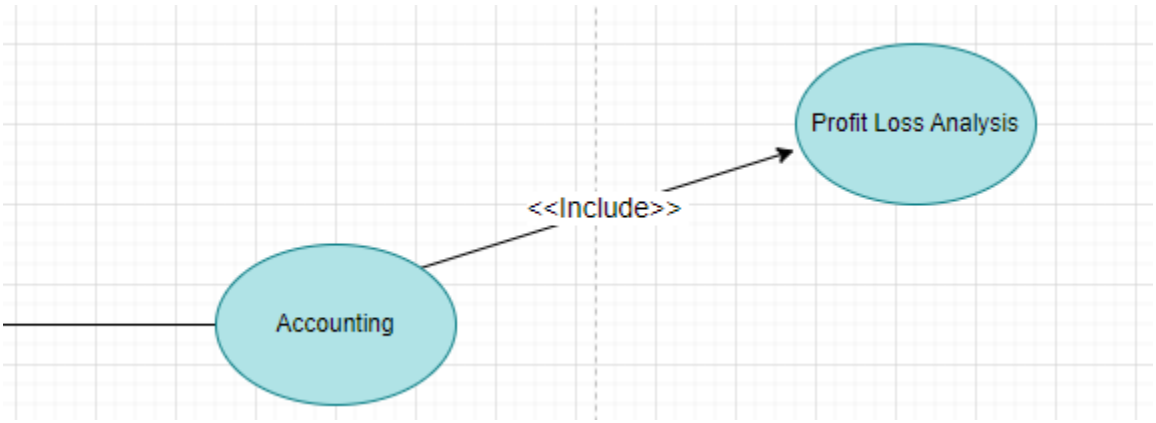
(1)Use case diagram :



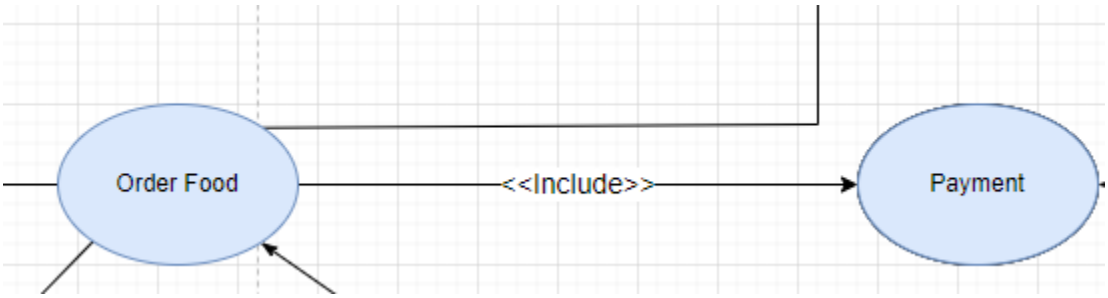
Relationship among use cases and actors

- **Include Relationship between two use cases:**

1. In this admin will use Accounting use case and to see the details for payments and it will include profit loss analysis.

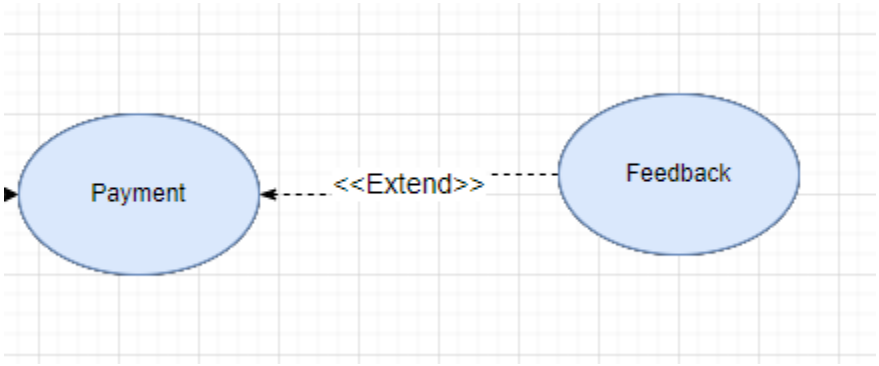


2. After the customer has ordered food and food is served by staff , at the end customer will have to pay the bill which is included in the Payment use case.

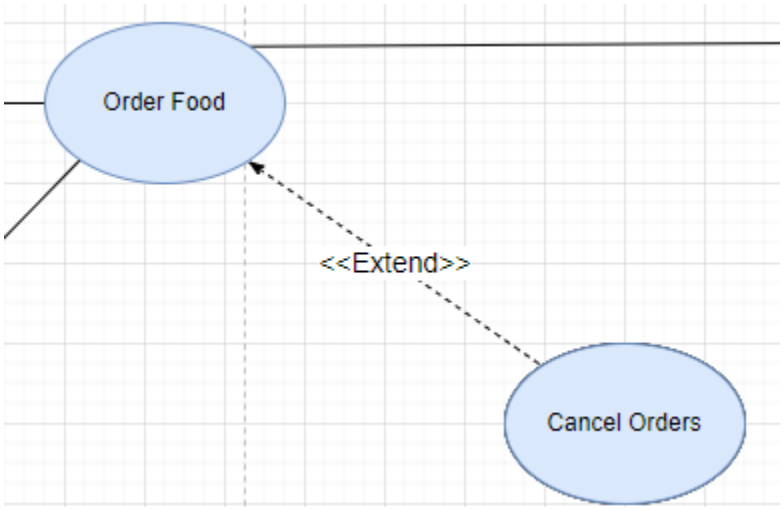


- **Extend Relationship between three use cases :**

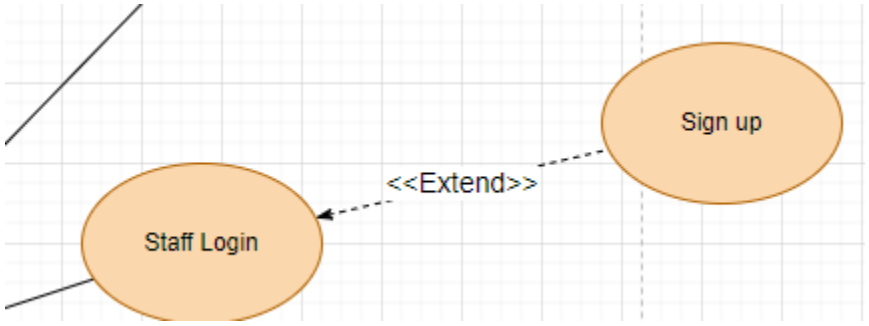
1. After the customer has paid the bill , he / she will be asked for feedback. It's optional to give feedback so it is extended functionality.



2. After the customer has given the food order , if there are any items that are not available or the customer wants to cancel the order then it can be done. So cancel orders is extended functionality.



3. If staff logins and staff's account does not exist then there should be a sign up option for staff so Sign up is the extended functionality.



(2) The Use Case Description for each use case :

Use case : 1

1. **Name**: Order Food
2. **Actors**: Customer ,Chef, Staff
3. **Goals**: Fulfilling the customer food order.
4. **Preconditions**: Customer has reviewed the menu and decided what to order.
5. **Postconditions**: Either the order has been placed or rejected.
6. **Description**: When the customer orders the food , the staff member receives the order and sends it to the chef for preparing the order.
7. **Trigger**:
    7.1. When an order request is made by the customer.
    7.2. When a customer eventually cancels the order.
8. **Main flow**:
    8.1. The customer selects the items which they wish to order.
    8.2. The staff takes the order and confirms it.
    8.3. The staff sends the order to the chef.
    8.4. The chef receives the order and prepares the food.
    8.5. The waiter serves the food to the customer.
    8.6. The order is marked as complete in the system.
9. **Alternate flow**:

8.2.a. If there is any mistake in order then staff cross check it with the customer and correct the mistake.

8.2.b. If the ingredients for making the order are not available then staff will notify the customer to choose an alternative option.

Use case : 2

1. **Name**: Inventory Management
2. **Actors**: Admin
3. **Goal**: Keep updating the system with the current inventory levels.
4. **Preconditions**: Admin has logged in into the system.
5. **Description**: The staff notifies the admin about a shortage and the admin updates the status to unavailable for that food item.
6. **Basic flow**:
    a. The staff informs the inventory levels as items are used or received.
    b. The admin updates inventory to the system and places orders as needed.

Use case: 3

1. **Name**: Table management
2. **Goal**: Manage the booked tables by the customer.
3. **Actors**: Customer, staff
4. **Preconditions**: The staff has logged in into the system.
5. **Postconditions**: Either the request will be accepted or rejected.
6. **Basic flow**:
    a. The customer requests booking for a table.
    b. The system will search for available tables.
    c. The system confirms and updates the table status.

Use Case : 4

1. **Name**: Menu management.
2. **Actors**: Admin
3. **Goal**: To successfully add a new item in the existing menu
4. **Preconditions**: Admin has logged in into the system.
5. **Postconditions**: Menu should be updated as needed.
6. **Description**: Admin logs into the system and adds a new item by entering item data and images. This item should now reflect into the updated menu of the restaurant.
7. **Trigger**:
   7.1. When the admin clicks add a new item.
   7.2. When admin enter item details and confirms to add the item
8. **Basic Flow:**
   8.1. Admin selects the add/remove item option from the interface.
   8.2. System shows a bunch of details to the manager to be filled out like Item name, description, price, image etc.
   8.3. Admin fills all the required information and clicks next button
   8.4. System confirms the information - stores data of new items and confirms to add it to the menu.
   8.5 Admin confirms the item to be added and the item is added to the menu successfully.

Use Case: 5

1. **Name**: Admin Login, Staff Login/SignUp
2. **Actors**: Staff, Admin
3. **Goal**: To successfully login to the system and perform the desired operations.
4. **Preconditions**: Admin account should exist.

5. **Post Conditions**: Successful login to the system where admin can manage the restaurant and staff can see past and current orders.
6. **Description**: After entering the valid credentials staff and admin gets logged in to the system.
7. **Trigger**: When a new user wants to place an order a sign-up page gets triggered.
8. **Basic Flow**:
   8.1. Actor selects the login/signUp button.
   8.2. System prompts for form where actor enters credentials.
   8.3. System validates the credentials and logs in.
9. **Alternate Flow**:
   9.1. If the emailID / password of the actor is incorrect, the system sends an error and an option of “forgot password” where the actor can reset the password.

Use Case :6

1. **Name**: Accounting
2. Actors: Admin
3.Goal: To generate reports on sales, and other relevant metrics.
4.**Preconditions**: The admin must have logged into the system successfully before generating the report
5.**Postconditions**: Either report has been generated or not.
6. **Description** : The Admin uses the System to generate reports on sales and other relevant metrics. The system generates and displays the requested report.
7. **Trigger** :
   7.1 When Admin clicks on generate report.

7.2 Admin selects the period and clicks on generate report.

8. **Basic Flow** :

8.1 Admin clicks on the “Generate report” button.

8.2 System shows the options for selecting the duration for which the admin wants to see the report.

8.3 Admin selects the duration

8.4 Admin clicks on generate report

8.5 System generates the report and show it to the admin

9. **Alternate Flow** :

9.1 If admin has not selected the duration, and he clicked on generate the report then show notification to enter or select the duration.

Use case: 7

1. **Name:** Order management
2. **Actor:** staff
3. **Goal:** To efficiently process customer orders and ensure timely delivery of food ordered along with tracking the order.
4. **Preconditions:**
   - 4.1. The system is properly installed and functional.
   - 4.2. The menu items and prices are correctly entered.
   - 4.3. The inventory is properly tracked and updated.
5. **Postcondition:**
   - 5.1. The customer receives the correct order on time.
   - 5.2. The inventory is updated to reflect the items used for the order.
   - 5.3. The system can track sales, inventory and other metrics.
6. **Main flow:**
   - 6.1. Staff updates the order taken.
   - 6.2. Updates in order:

6.2.1. Processing
6.2.2. Completed
6.2.3. Delivered

Use Case : 8

1. **Name**: Payment
2. Actors: Customer, Staff
3.Goal: To receive payment from the customer.
4.**Preconditions**: The customer has eaten the food and does not want to order anything else.
5.**Postconditions**: After successful payment, the staff will enter its details in the system.
6. **Description** : After the customer has eaten the food and doesn't want anything , staff will give a bill to the customer and the customer will pay according to the preference of the method of payment.
7. **Trigger** :

7.1 When staff will ask for payment.

8. **Basic Flow** :

8.1 The staff asks for payment and method of payment to the customer.

8.2 Customer selects the payment method.

8.3 Customer pays the required amount.

8.4 Staff confirms the payment and asks customers for feedback.

8.5 Customer gives feedback if he wants to.

9. **Alternate Flow** :

8.4.a If the customer doesn’t want to give feedback he won’t give it and check out of the restaurant.

Non functional requirements:

Non-functional requirements are attributes of a system that describe how the system should behave, rather than what it should do. Here are some non-functional requirements for a restaurant automation system:

- **Security:** The system should be secure and protect sensitive information from unauthorized access and data breaches. In the restaurant automation system only admin should have access to management of: menu, accounting, inventory, etc. No third person should have access to it.
- **Reliability:** The system should be reliable and consistently available, with minimal downtime and technical issues. The feedback part of the system should immediately report to the admin as to rectify the error ASAP.
- **User-Friendliness:** The system should be easy to use and navigate, with a user-friendly interface that requires minimal training. The restaurant owner/ Admin and staff should be able to easily navigate and use the system without any pre-training.
- **Responsiveness:** The system should be responsive, irrespective of the platform on which it is running. The restaurant automation system should be platform independent.
- **Scalability:** The system should be scalable and flexible, able to accommodate the growing needs of the restaurant over time. The system should take food orders concurrently and should be able to handle workload as needed.
- **Performance:** The system should have fast and efficient performance, with quick load times and minimal lag. Workload management and concurrent orders should be handled efficiently.
- **Support:** The system should be backed by a responsive and knowledgeable support team, with access to technical assistance and updates as needed. The feedback system should directly be connected to the support team and admin to sort out the priorities of issues accordingly.