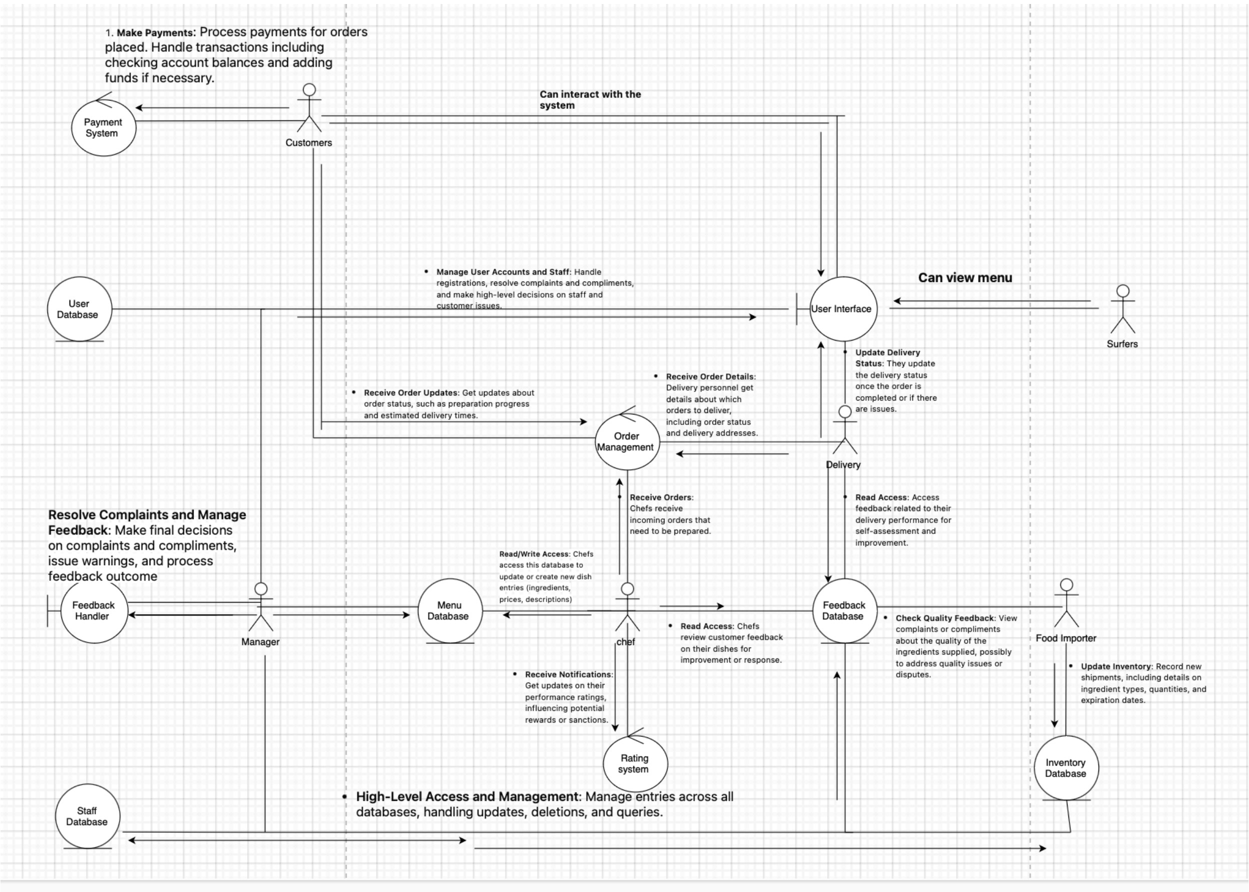
PHASE 2 REPORT

Group [S]: Akshay Raju, Veronica Koval, Mehereen Meem, Aleika Chery, Bismah Hasan

This report is meant to provide the data structure and logic to carry out the functionalities dictated by the specification.

1. Introduction

an overall picture of the system using collaboration class diagram:



1. **Make Payments:** Process payments for orders placed. Handle transactions including checking account balances and adding funds if necessary.

- **Manage User Accounts and Staff**: Handle registrations, resolve complaints and compliments, and make high-level decisions on staff and customer issues.
- **Receive Order Updates**: Get updates about order status, such as preparation progress and estimated delivery times.
- **Receive Order Details**: Delivery personnel get details about which orders to deliver, including order status and delivery addresses.
- **Update Delivery Status**: They update the delivery status once the order is completed or if there are issues.
- **Receive Orders**: Chefs receive incoming orders that need to be prepared.
- **Read Access**: Access feedback related to their delivery performance for self-assessment and improvement.

Resolve Complaints and Manage Feedback: Make final decisions on complaints and compliments, issue warnings, and process feedback outcome

Read/Write Access: Chefs access this database to update or create new dish entries (ingredients, prices, descriptions)

- **Read Access**: Chefs review customer feedback on their dishes for improvement or response.
- **Check Quality Feedback**: View complaints or compliments about the quality of the ingredients supplied, possibly to address quality issues or disputes.
- **Receive Notifications**: Get updates on their performance ratings, influencing potential rewards or sanctions.
- **Update Inventory**: Record new shipments, including details on ingredient types, quantities, and expiration dates.
- **High-Level Access and Management**: Manage entries across all databases, handling updates, deletions, and queries.

2. ALL USE CASES

Use Cases (Scenarios)

- Chefs
    - Prepare order
        - There will be a priority queue of orders that need to be prepared
        - Once the order is in the system, they will get distributed to the chefs currently working based on priority
        - If ingredients are present (check inventory database), the meal will be prepared

    - Once finished, the chef can indicate it is done, which will remove the item from the queue and add it to another queue (such as for delivery if it is a takeout)
  - Special order handling (VIP customers)
    - VIP customer object will have a function to request a certain meal
    - This then gets added to a database of dishes the restaurant prepares
    - It also adds the meal to a priority queue of orders needing to be prepared in the kitchen
  - Raise quality issue (Food importer)
    - Will have a queue of issues and complaints, each with a severity rating attached to it
    - Raising an issue will add a complaint to this queue with details on:
      - Type (what is being complained about)
      - Severity
      - Date created
    - This queue can be viewed by the management to address the problems
  - Decide menu
    - Chef has the ability to create a new entry into the database of dishes the restaurant makes
    - Populating this database with the ingredients will also create new entries into the ingredients database if not present initially
    - Current menu can be selected from entries in the database
- Delivery:
  - Accept delivery
    - Once the order leaves the meals to make queue for the chef, if it is a delivery then it gets added to a delivery priority queue
    - Then one of the delivery drivers has the option of picking up the order first in the queue
    - This assigns the delivery task to them (driver object will have attribute for current delivery)
  - Can receive and report complains/complements
    - Upon completing a delivery, delivery personnel are available to receive feedback from customers.
    - They have an option to report complaints/ compliments
    - The type of feedback (complaint or compliment) is categorized and recorded for further review and processing.
    - All recorded feedback is transmitted to the central system where it is stored and processed.
    - The feedback data is accessible to both management and the delivery team for quality control and service improvement purposes.
  - View delivery ratings
    - Customer ratings will be a list of integers
      - Values between 1 and 5 where 1 is terrible and 5 is excellent
    - Delivery drivers can access summary statistics on this list such as the mean/median/etc. and a histogram of the distribution of ratings
- Food importers:
  - Manage inventory
    - When new food items get delivered, then the food importer has access to update the database of ingredients with the following attributes:
      - Ingredient name
      - Ingredient type (dairy/meat/vegetable/etc.)
      - Date of shipment
      - Date of expiration
      - Count
  - Raise quality issue

        - Will use the same functions as the raise quality issue use case found in Chefs (see above)
- Manager
    - Manage staff
        - Managers will have access to a database of staff profiles who work at the restaurant
        - This database will have:
            - Keys – Unique Employee ID
            - Attributes:
                - Name
                - Position/Role
                - Pay
                - Number of Complaints
                - Number of Compliments
                - Tier (after promotion/demotion)
        - Will have a set of functions that perform the use cases below:
            - Promote staff
                - Create function that if the number of compliments = 2, then update the tier of the employee by 1
                - Also reset value of compliments to 0
            - Demote staff
                - Similar to the promote staff function except it updates the tier by -1
            - Hire staff
                - Add entry to the database with all relevant information
                - This includes generating a unique employee ID number to function as their key
            - Fire staff
                - Function that is executed when the tier level of an employee reaches 0 (been demoted multiple times)
                - Remove entry from database that corresponds with the employee ID's key value
    - Handle Reviews
        - Reviews will all be stored as a table with classifications on who wrote it and if it's a compliment or complaint
        - Managers will have the option to remove reviews by deleting their corresponding entries from table
        - Will also call functions to issue warnings to the customer who wrote it
    - Issue Warning to Customer
        - Each customer will have a number of warnings in the system's customer database
        - Function to issue warning would increment the number of warnings in the database associated with that customer
    - Demote/Deregister Customer
        - If a customer receives multiple warnings, then a condition is satisfied and this function is called
        - This would check whether the customer is a VIP customer or not
            - If VIP, then change database attribute from VIP to regular
            - If not VIP, then would pop this customer from the database altogether, thus deregistering them
        - Would then clear warnings for said customer
- Customers/Surfers:
    - Common to Customers and Surfers
        - View menu
            - Would allow the surfer to view entries in the database of dishes the restaurant

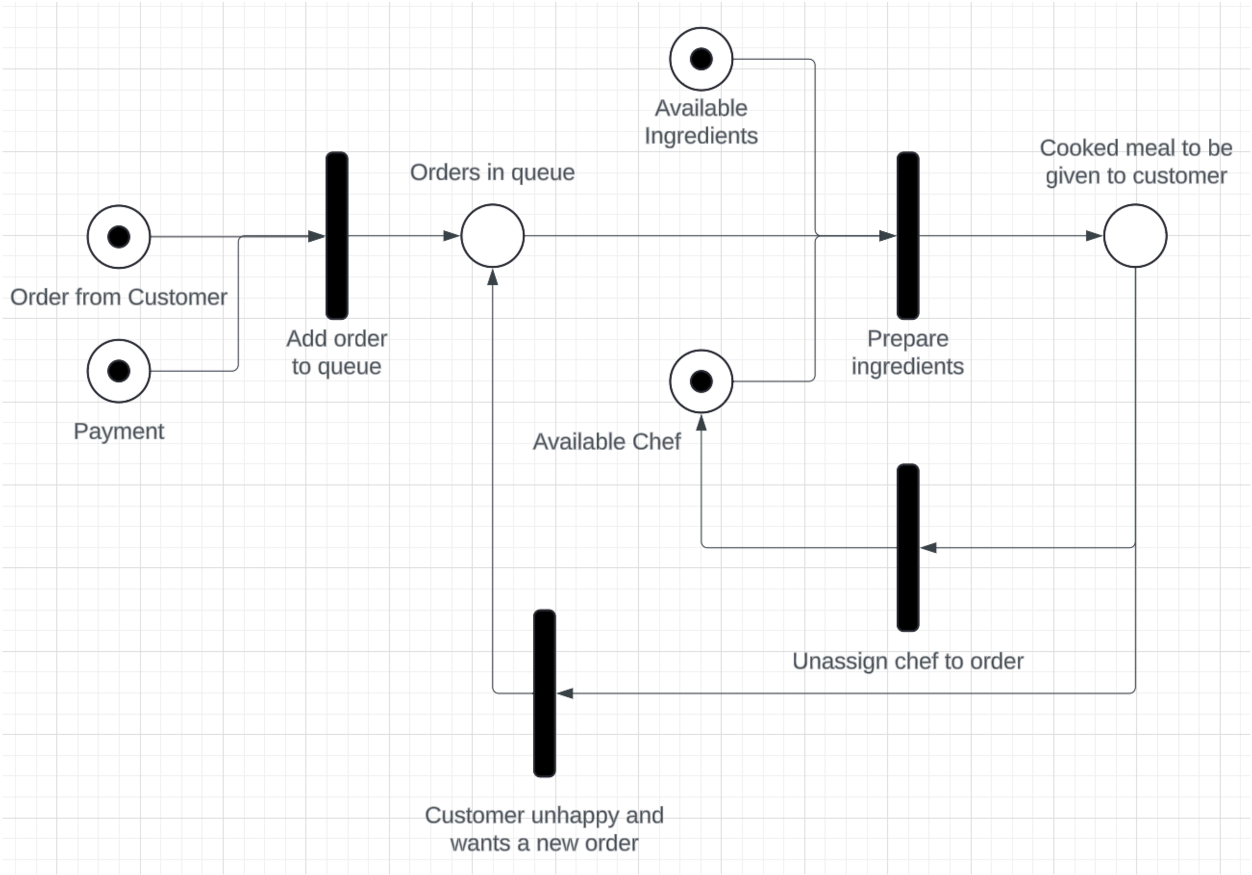
  - However only ones that have the attribute of ‘currently making’ that the chef decides, hides all other dishes restaurant isn’t making at the moment
  - Would use a series of click down menus to expand on the dish’s ingredients, calories, etc.
- View ratings
  - Would allow the user to view summary statistics about the list of ratings of the chefs in the form of a histogram
  - Would also allow them to view ratings of each individual dish in a similar manner

- Only Customer Use Cases
  - Place order
    - Would allow the user to click on a menu item and add it to a cart
      - This cart would be a stack of items which would continuously add more elements as more things get added
      - When customer views cart and clicks checkout, these items then will go to the queue of meal preparation that the chef handles
    - See figure in Supporting Information (Section 4) for more details on this process
  - Add rating
    - This would give the customer a survey with multiple choice input fields
    - Each input corresponds to a number to describe the quality of service
    - Then has an optional stage of adding a review
      - If clicked, will show a text field where the customer can write a review
      - Also have multiple choice question field for if it’s a compliment or complaint
- Only VIP Customer Use Cases
  - Access Exclusive Offers
    - Would have a separate tab in the system to show exclusive offers
    - Then VIP customers can click an exclusive offer which would add an element to the stack in the cart that would be of negative value (discount on food)
  - Special food request
    - Would be a separate menu item which will include a text field along with sub-text fields for specific ingredients
    - This order will be placed on the queue of incoming orders to the chef, as well as to the database of ingredients so that we can check if the restaurant has the necessary ingredients to fulfill the order

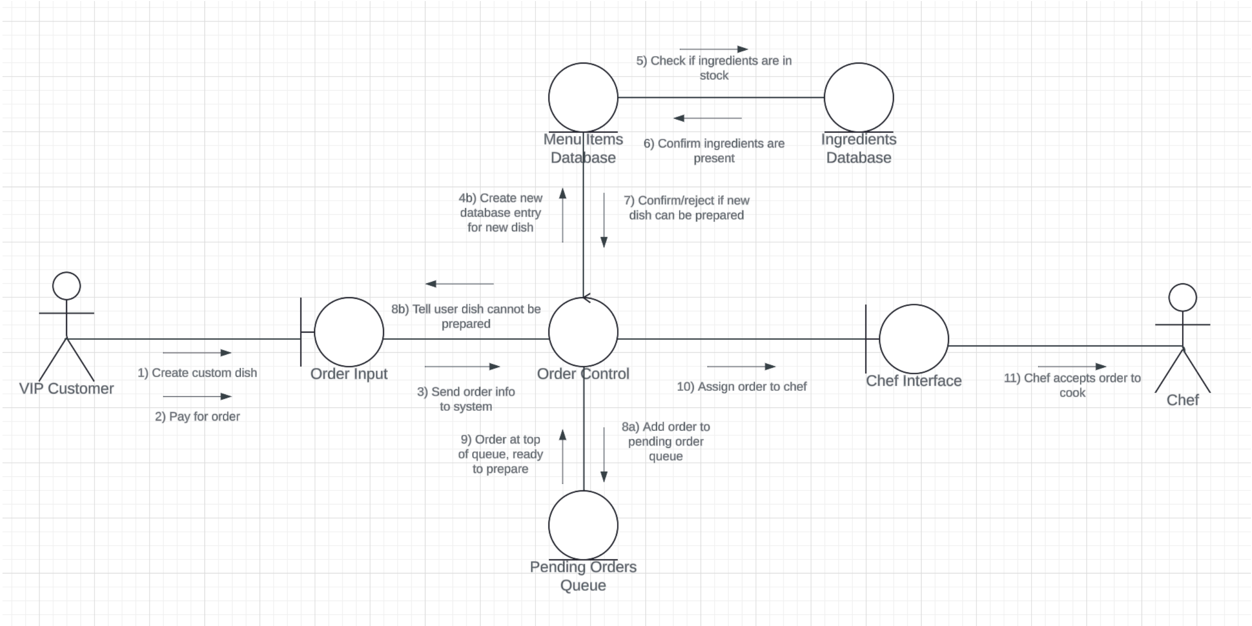
Diagrams:

CHEF

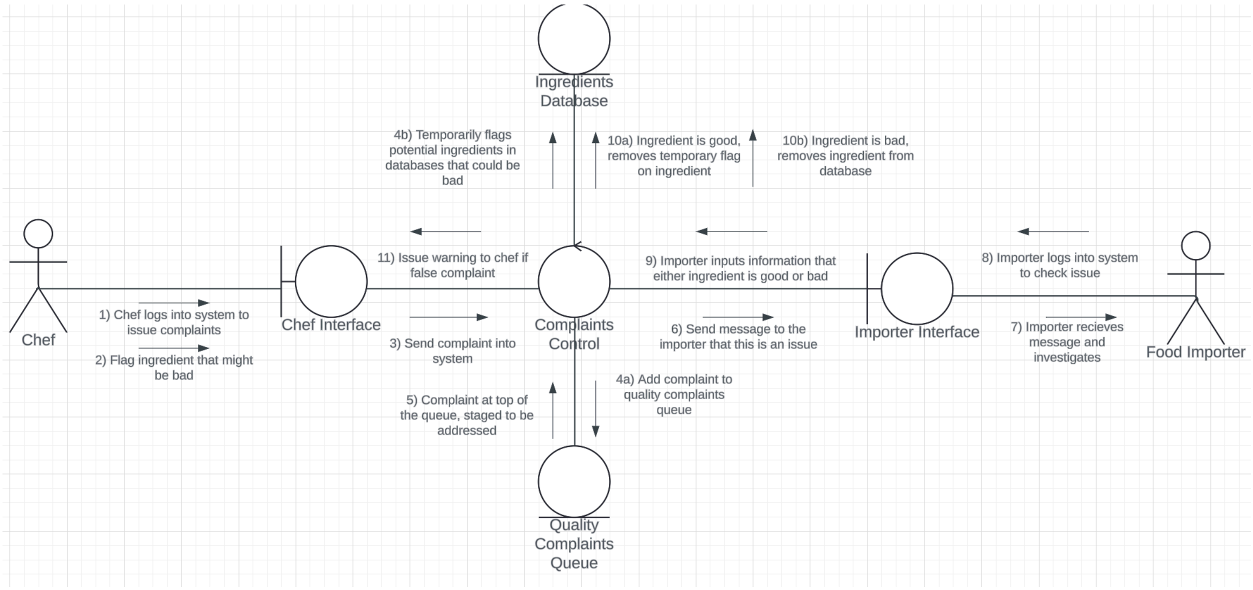
Use Case 1 (Prepare Order): Petri Net



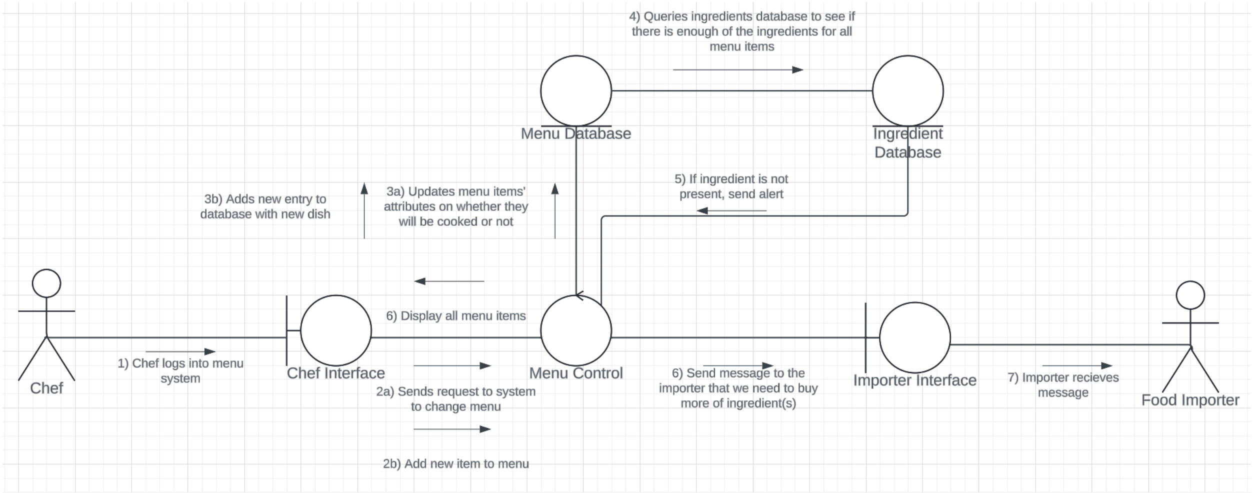
Use Case 2 (Special Order Handling): Collaborative Class Diagram


5) Check if ingredients are in stock
Menu Items Database
6) Confirm ingredients are present
Ingredients Database
4b) Create new database entry for new dish
7) Confirm/reject if new dish can be prepared
8b) Tell user dish cannot be prepared
VIP Customer
1) Create custom dish
2) Pay for order
Order Input
3) Send order info to system
Order Control
10) Assign order to chef
Chef Interface
11) Chef accepts order to cook
Chef
9) Order at top of queue, ready to prepare
8a) Add order to pending order queue
Pending Orders Queue

Use Case 3 (Raise Quality Issue): Collaborative Class Diagram

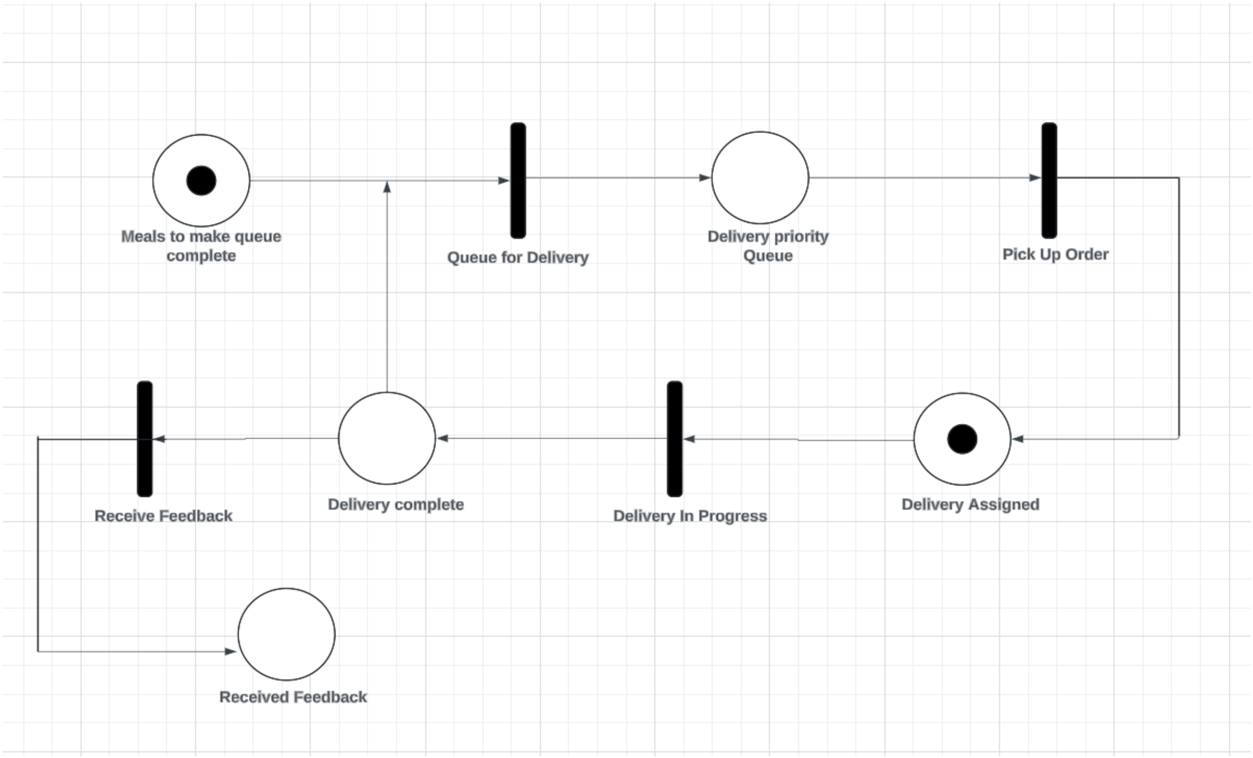

Ingredients Database
4b) Temporarily flags potential ingredients in databases that could be bad
10a) Ingredient is good, removes temporary flag on ingredient
10b) Ingredient is bad, removes ingredient from database
Chef
11) Issue warning to chef if false complaint
9) Importer inputs information that either ingredient is good or bad
8) Importer logs into system to check issue
1) Chef logs into system to issue complaints
2) Flag ingredient that might be bad
Chef Interface
3) Send complaint into system
Complaints Control
6) Send message to the importer that this is an issue
Importer Interface
7) Importer recieves message and investigates
Food Importer
5) Complaint at top of the queue, staged to be addressed
4a) Add complaint to quality complaints queue
Quality Complaints Queue

Use Case 4 (Decide Menu): Collaborative Class Diagram

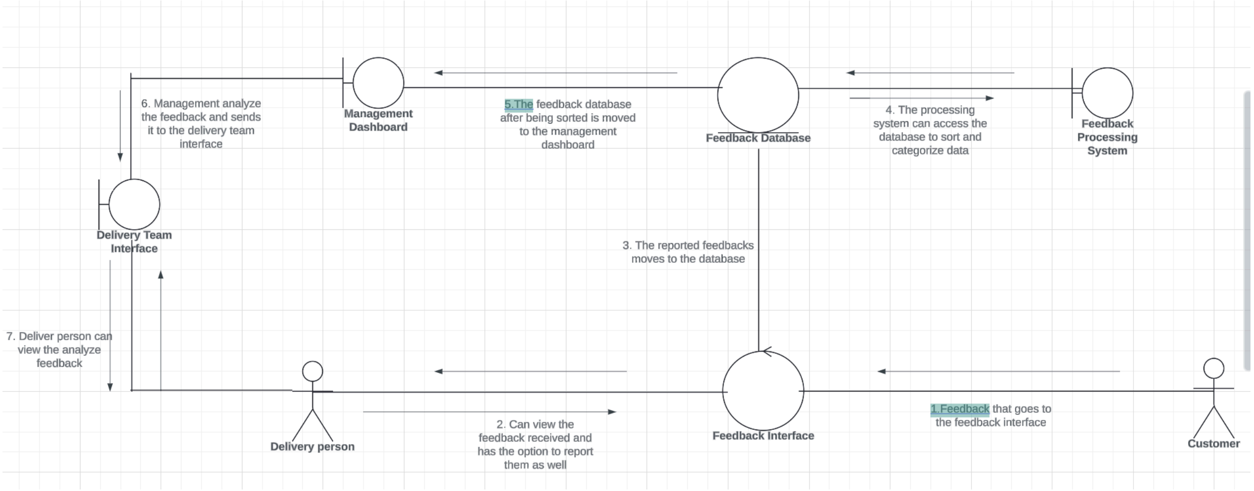

4) Queries ingredients database to see if there is enough of the ingredients for all menu items
Menu Database
Ingredient Database
3b) Adds new entry to database with new dish
3a) Updates menu items' attributes on whether they will be cooked or not
5) If ingredient is not present, send alert
Chef
6) Display all menu items
1) Chef logs into menu system
Chef Interface
Menu Control
2a) Sends request to system to change menu
2b) Add new item to menu
6) Send message to the importer that we need to buy more of ingredient(s)
Importer Interface
7) Importer recieves message
Food Importer

DELIVERY

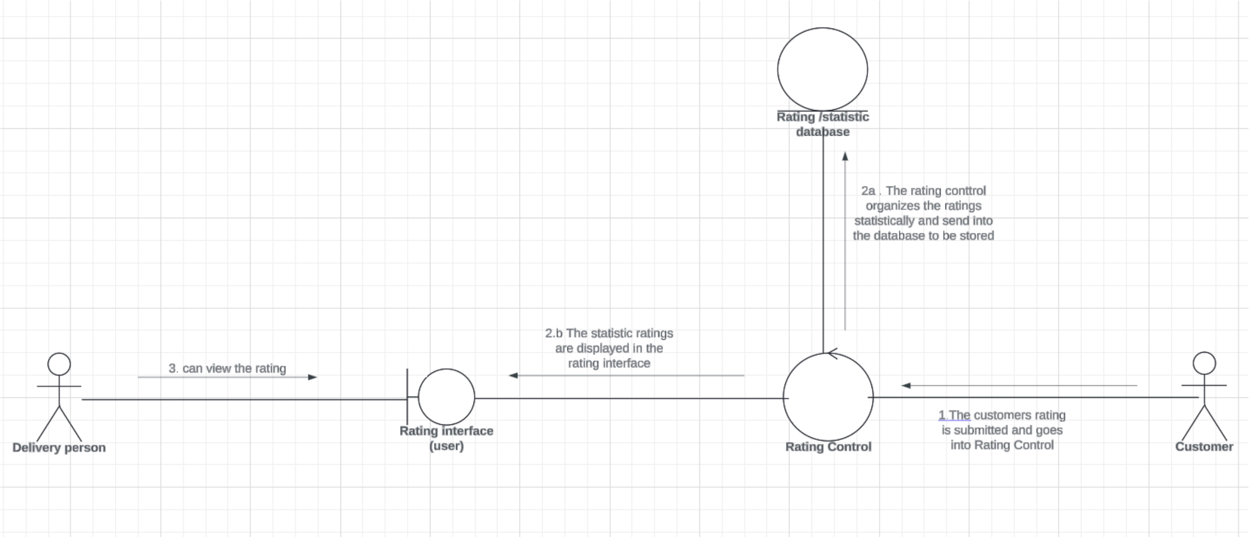
Use Case 1 (Accept delivery): Petri net



Use Case 2 (Can receive and report complains/complements): Collaborative Class Diagram

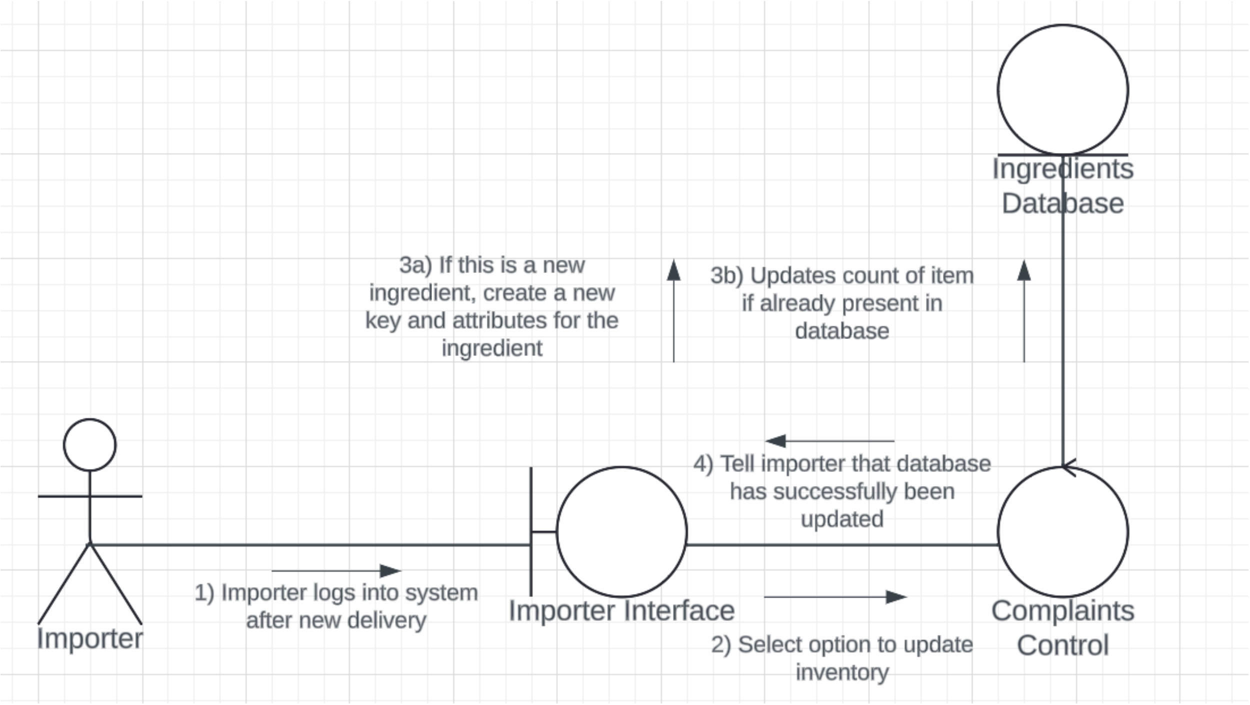

6. Management analyze the feedback and sends it to the delivery team interface
Management Dashboard
5.The feedback database after being sorted is moved to the management dashboard
Feedback Database
4. The processing system can access the database to sort and categorize data
Feedback Processing System
Delivery Team Interface
3. The reported feedbacks moves to the database
7. Deliver person can view the analyze feedback
Delivery person
2. Can view the feedback received and has the option to report them as well
Feedback Interface
1.Feedback that goes to the feedback interface
Customer

Use Case 3 (View delivery ratings): Collaborative Class Diagram

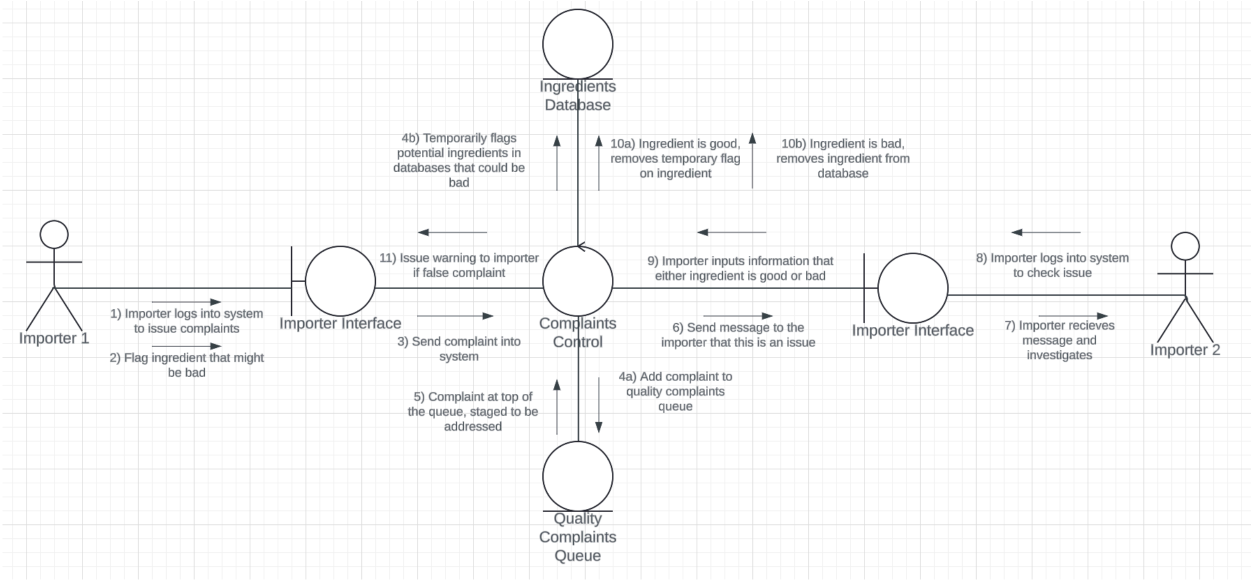

Rating /statistic database
2a . The rating conttrol organizes the ratings statistically and send into the database to be stored
2.b The statistic ratings are displayed in the rating interface
3. can view the rating
Delivery person
Rating interface (user)
Rating Control
1.The customers rating is submitted and goes into Rating Control
Customer

FOOD IMPORTERS

Use Case 1 (Manage Inventory): Collaborative Class Diagram

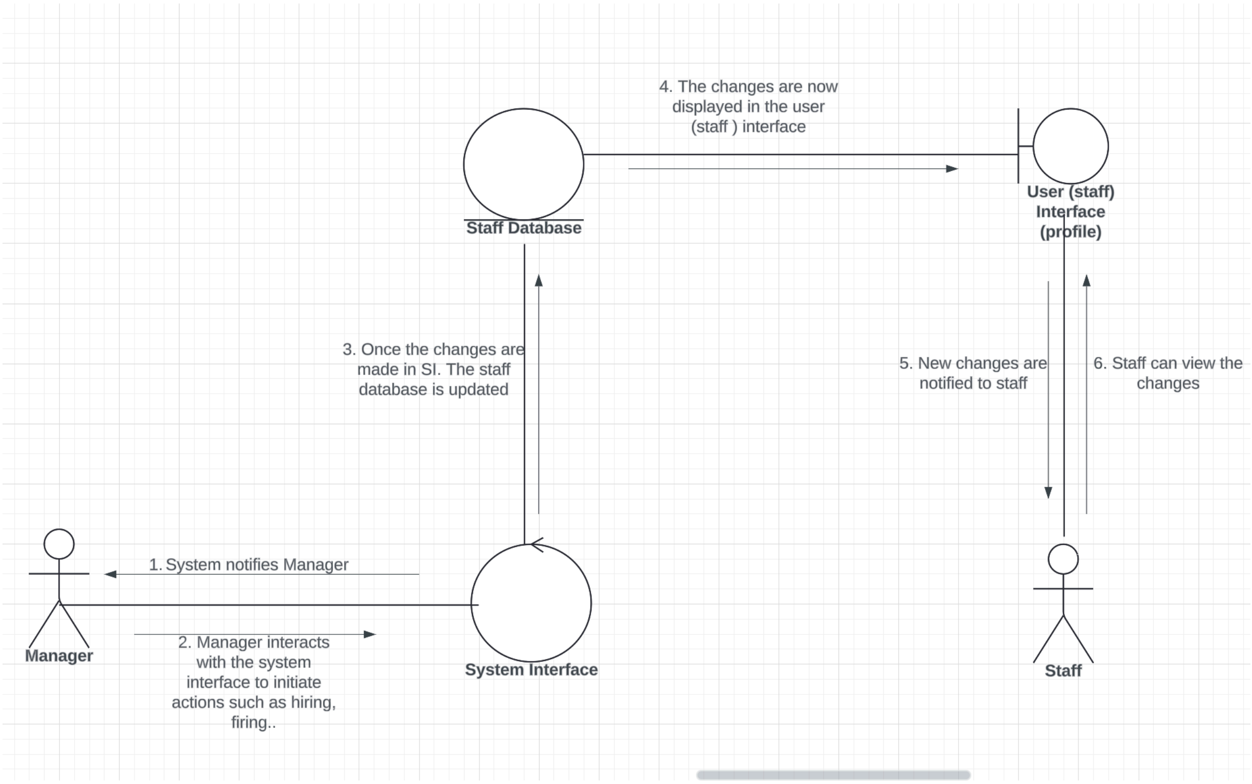

Ingredients Database
3a) If this is a new ingredient, create a new key and attributes for the ingredient
3b) Updates count of item if already present in database
4) Tell importer that database has successfully been updated
1) Importer logs into system after new delivery
Importer
Importer Interface
2) Select option to update inventory
Complaints Control

Use Case 2 (Raise Quality Issue): Collaborative Class Diagram

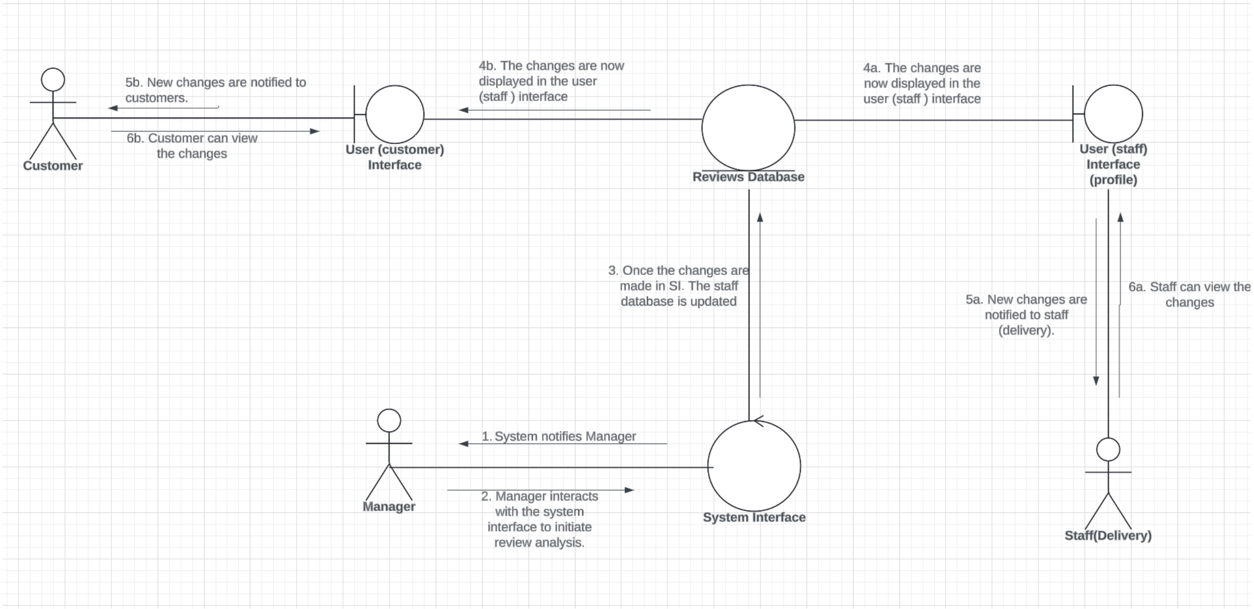

Ingredients Database
4b) Temporarily flags potential ingredients in databases that could be bad
10a) Ingredient is good, removes temporary flag on ingredient
10b) Ingredient is bad, removes ingredient from database
Importer 1
1) Importer logs into system to issue complaints
2) Flag ingredient that might be bad
Importer Interface
11) Issue warning to importer if false complaint
3) Send complaint into system
Complaints Control
9) Importer inputs information that either ingredient is good or bad
6) Send message to the importer that this is an issue
Importer Interface
8) Importer logs into system to check issue
7) Importer recieves message and investigates
Importer 2
5) Complaint at top of the queue, staged to be addressed
4a) Add complaint to quality complaints queue
Quality Complaints Queue

MANAGERS

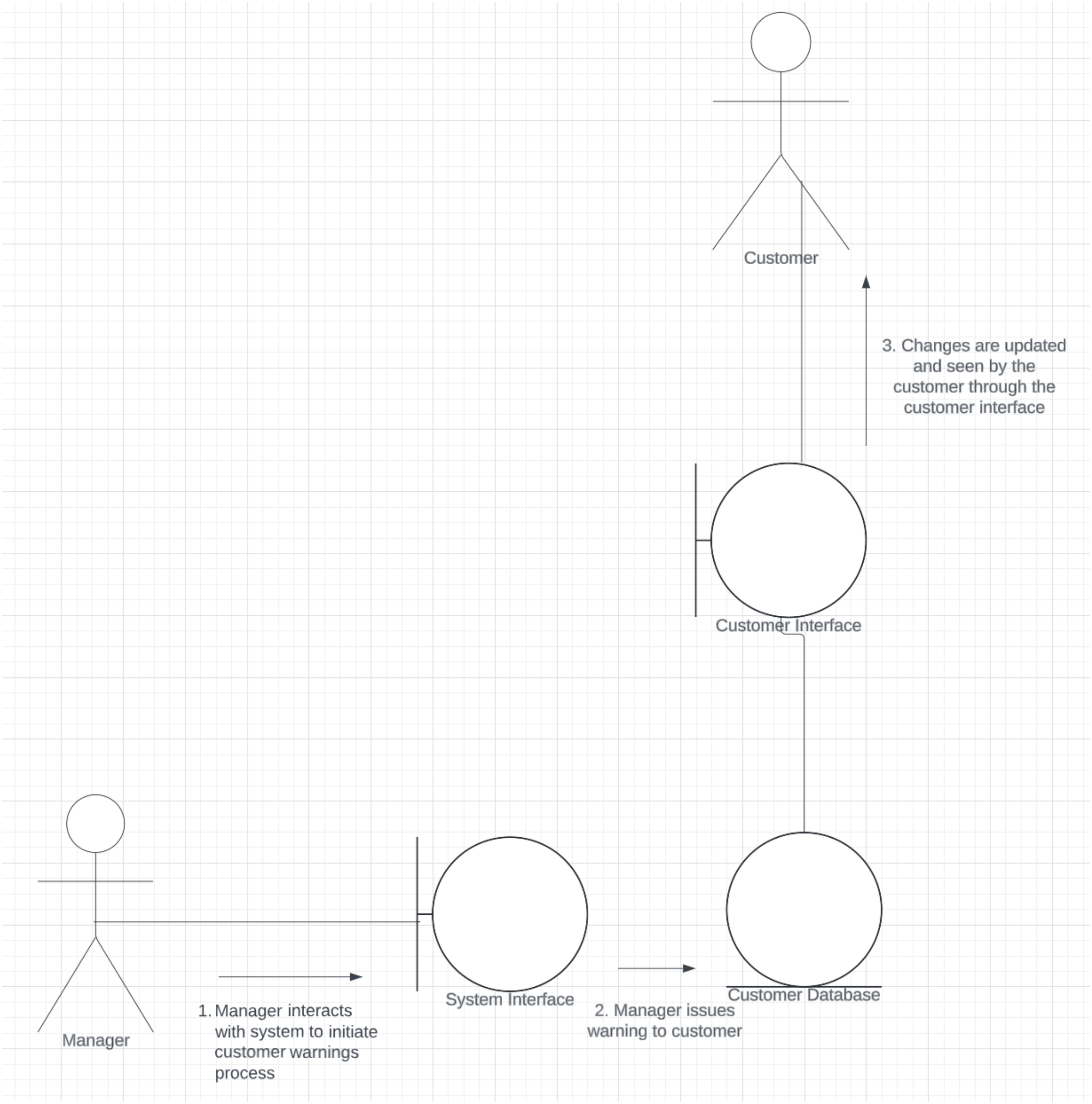
Use Case 1 (Manage staff): Collaborative Class Diagram


4. The changes are now displayed in the user (staff) interface
Staff Database
User (staff) Interface (profile)
3. Once the changes are made in SI. The staff database is updated
5. New changes are notified to staff
6. Staff can view the changes
1. System notifies Manager
2. Manager interacts with the system interface to initiate actions such as hiring, firing..
Manager
System Interface
Staff

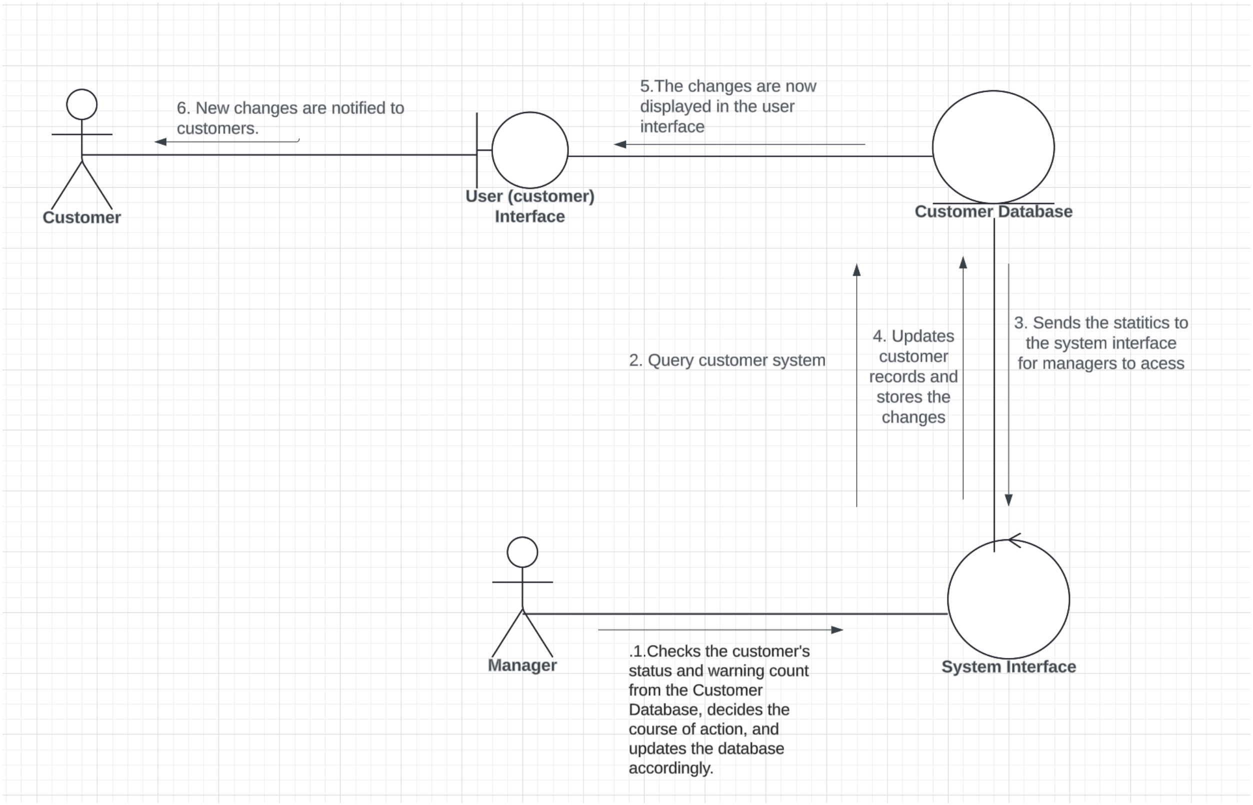
Use Case 2 (Handles review): Collaborative Class Diagram


5b. New changes are notified to customers.
6b. Customer can view the changes
Customer
User (customer) Interface
4b. The changes are now displayed in the user (staff) interface
Reviews Database
4a. The changes are now displayed in the user (staff) interface
User (staff) Interface (profile)
3. Once the changes are made in SI. The staff database is updated
5a. New changes are notified to staff (delivery).
6a. Staff can view the changes
1. System notifies Manager
Manager
2. Manager interacts with the system interface to initiate review analysis.
System Interface
Staff(Delivery)

Use Case 3 (Issue warning to customers): Collaborative Class Diagram

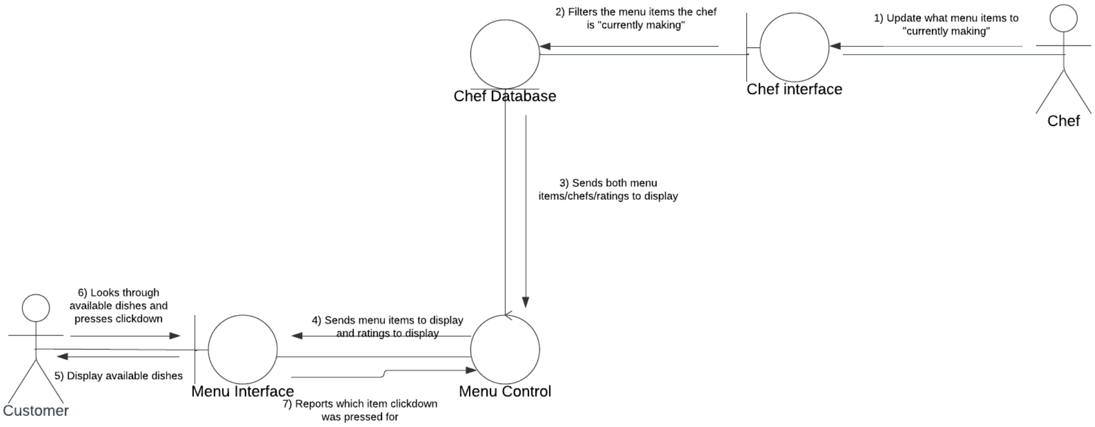

Customer
3. Changes are updated and seen by the customer through the customer interface
Customer Interface
Manager
1. Manager interacts with system to initiate customer warnings process
System Interface
2. Manager issues warning to customer
Customer Database

Use Case 4 (Deregister): Collaborative Class Diagram

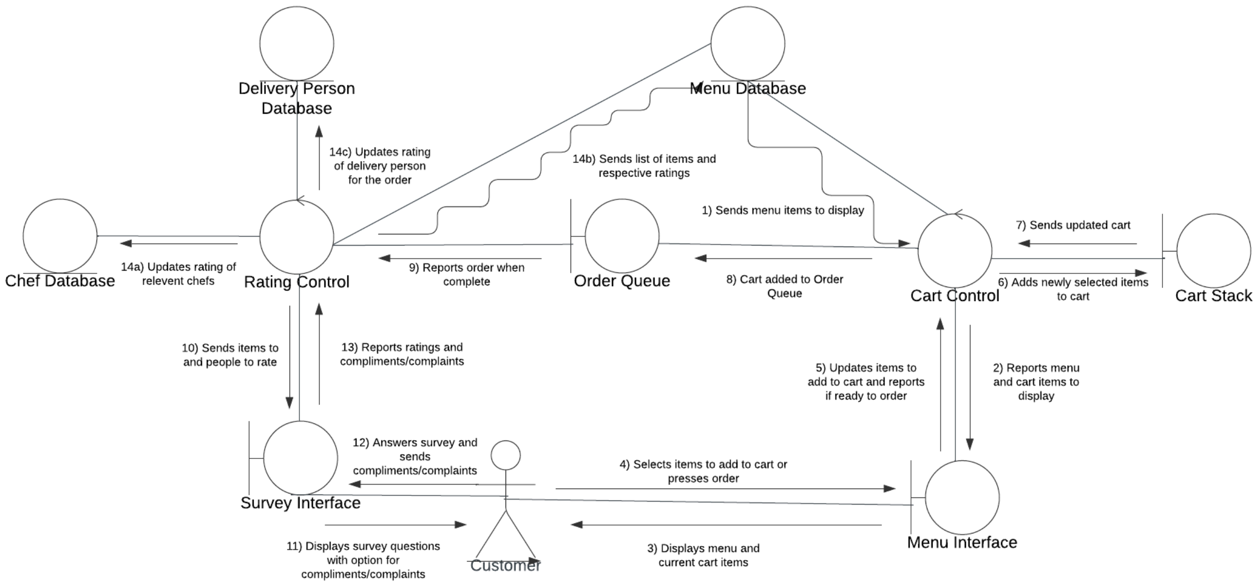

6. New changes are notified to customers.
5.The changes are now displayed in the user interface
Customer
User (customer) Interface
Customer Database
2. Query customer system
4. Updates customer records and stores the changes
3. Sends the statitics to the system interface for managers to acess
Manager
System Interface
.1.Checks the customer's status and warning count from the Customer Database, decides the course of action, and updates the database accordingly.

CUSTOMERS

Use Case 1 (Common to customers and surfers/ view menu): Collaborative Class Diagram

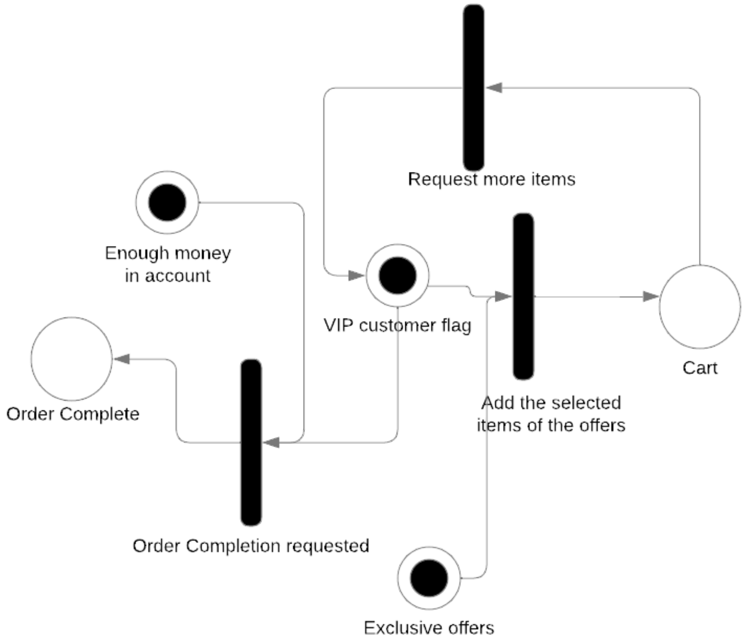

2) Filters the menu items the chef is "currently making"
1) Update what menu items to "currently making"
Chef Database
Chef interface
Chef
3) Sends both menu items/chefs/ratings to display
6) Looks through available dishes and presses clickdown
4) Sends menu items to display and ratings to display
5) Display available dishes
Customer
Menu Interface
Menu Control
7) Reports which item clickdown was pressed for

Use Case 2 (Only customers use case): Collaborative Class Diagram

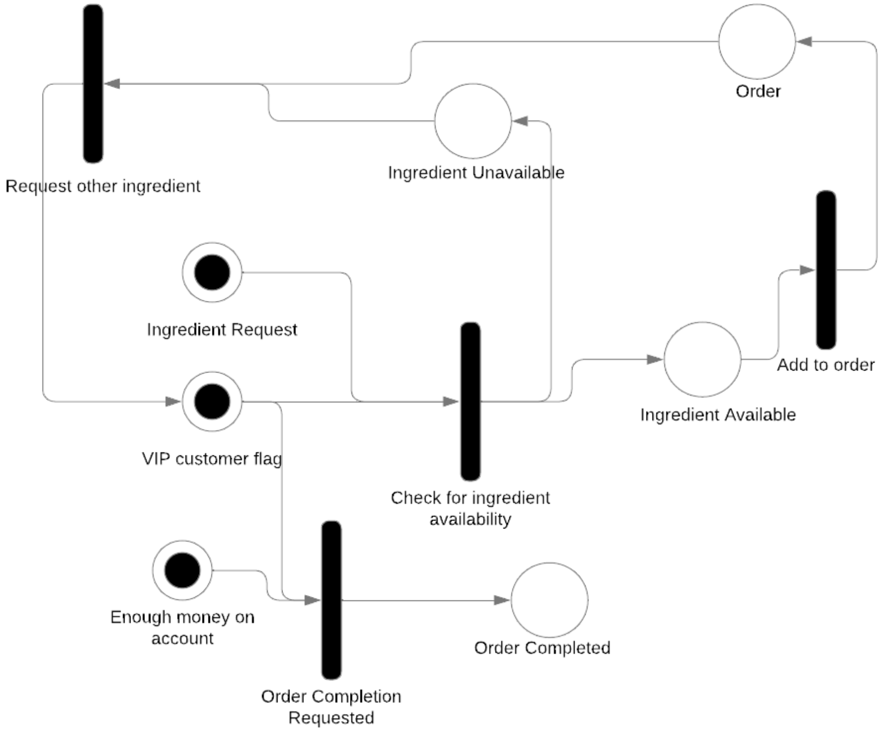

Delivery Person Database
Menu Database
14c) Updates rating of delivery person for the order
14b) Sends list of items and respective ratings
1) Sends menu items to display
7) Sends updated cart
Chef Database
14a) Updates rating of relevent chefs
Rating Control
9) Reports order when complete
Order Queue
8) Cart added to Order Queue
Cart Control
6) Adds newly selected items to cart
Cart Stack
10) Sends items to and people to rate
13) Reports ratings and compliments/complaints
5) Updates items to add to cart and reports if ready to order
2) Reports menu and cart items to display
12) Answers survey and sends compliments/complaints
4) Selects items to add to cart or presses order
Survey Interface
Menu Interface
11) Displays survey questions with option for compliments/complaints
Customer
3) Displays menu and current cart items

Use Case 3 (only VIP customers use case): Petri-net

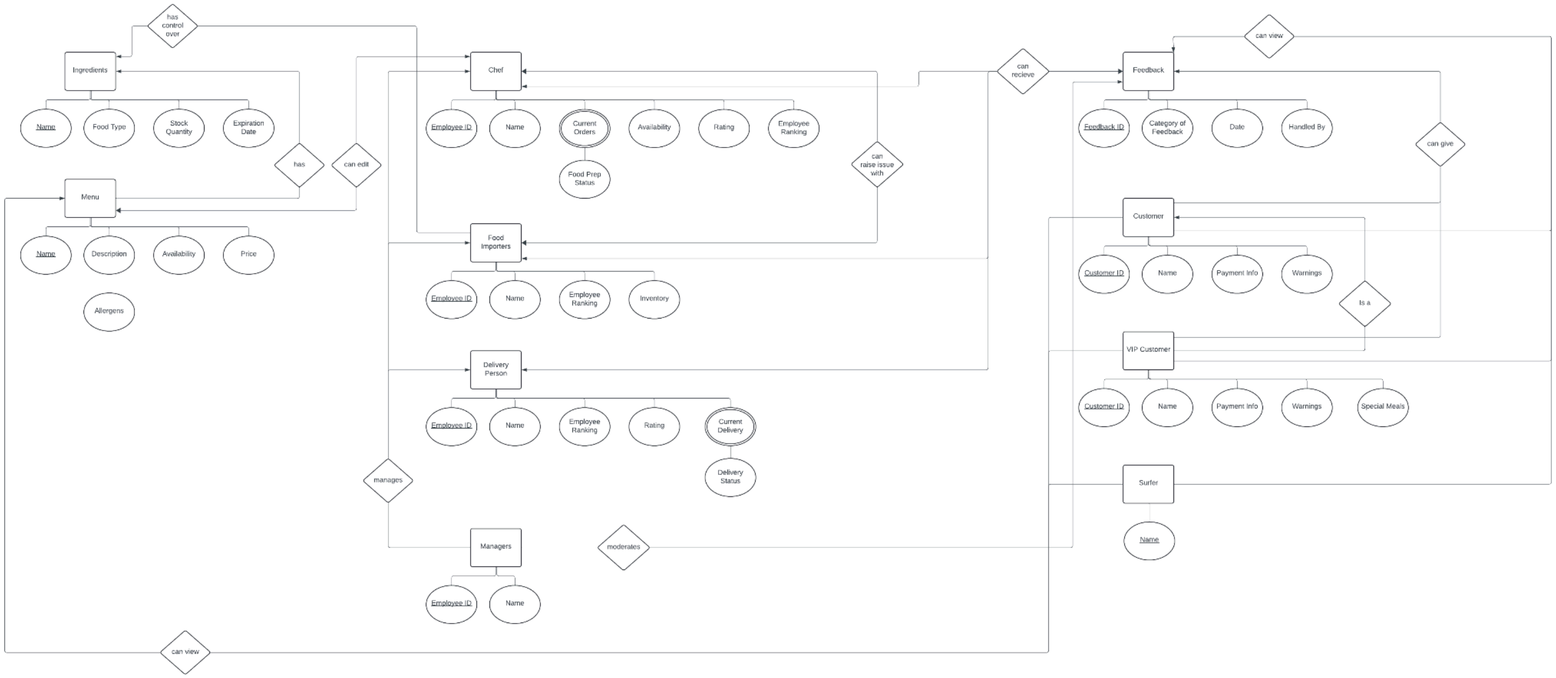
a) *Exclusive Offers*



b) *Customized order*



3. E-R diagram for the entire system attributes and key for each class should be provided


has control over
Ingredients
Name
Food Type
Stock Quantity
Expiration Date
has
can edit
Menu
Name
Description
Availability
Price
Allergens
Chef
Employee ID
Name
Current Orders
Food Prep Status
Availability
Rating
Employee Ranking
can raise issue with
Food Importers
Employee ID
Name
Employee Ranking
Inventory
Delivery Person
Employee ID
Name
Employee Ranking
Rating
Current Delivery
Delivery Status
manages
Managers
Employee ID
Name
moderates
can recieve
can view
Feedback
Feedback ID
Category of Feedback
Date
Handled By
can give
Customer
Customer ID
Name
Payment Info
Warnings
Is a
VIP Customer
Customer ID
Name
Payment Info
Warnings
Special Meals
Surfer
Name
can view

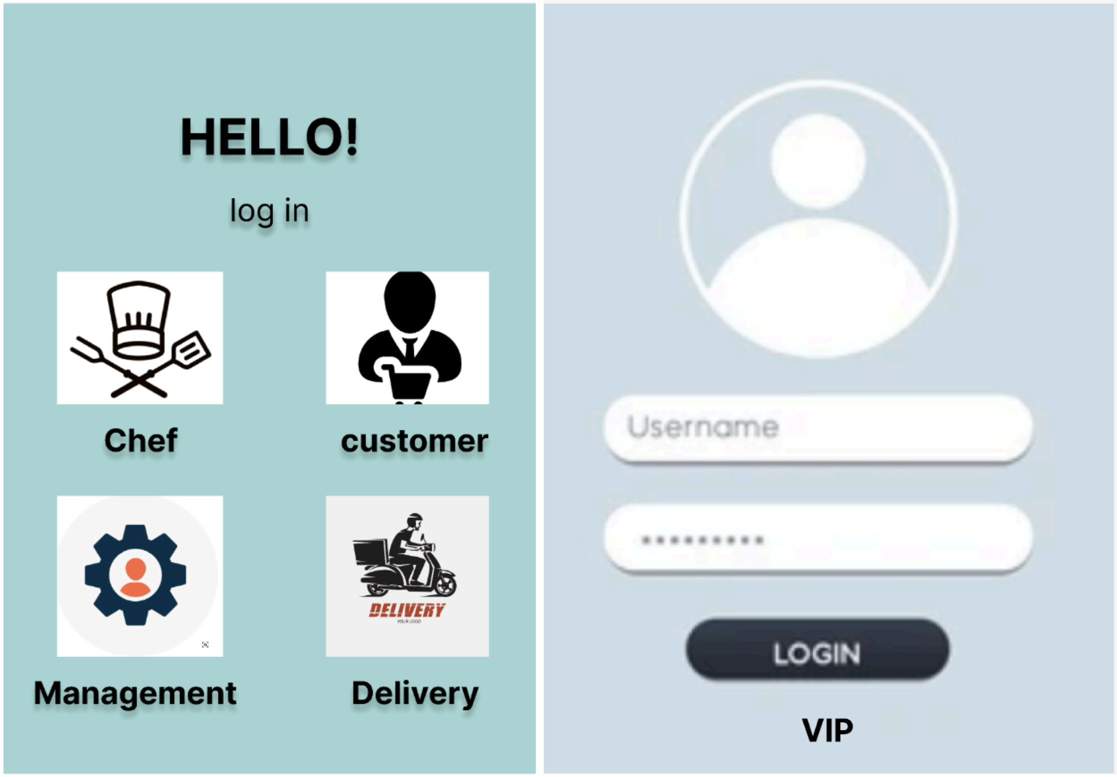
4. Detailed design:

for EVERY method, pseudo-code to delineate the input/output and main functionalities

THE PSEUDO CODE CAN BE ACCESSED THROUGH GITHUB
https://github.com/VerKoval/CulinaryCloud

5. System screens:

demonstrate major GUI screens of the system and a sample prototype of one functionality of your own choice.



6. Memos of group meetings and possible concerns of team work

Meeting on April 6, notes:

- Set up Github: https://github.com/VerKoval/CulinaryCloud
- Design 2 phase

****Order****

- ➔ All use cases
  - ◆ Scenarios for each use case: normal AND exceptional scenarios
  - ◆ Collaboration or sequence class diagram for each use case, choose 3 or more use cases: draw the Petri-nets instead of class diagrams
- ➔ E-R diagram for the entire system
  - ◆ attributes and key for each class should be provided
- ➔ Detailed design:
  - ◆ for EVERY method use pseudo-code to delineate the input/output and main functionalities
- ➔ System screens:
  - ◆ demonstrate major GUI screens of the system and a sample prototype of one functionality of your own choice.
- ➔ Introduction
  - ◆ an overall picture of the system using collaboration class diagram

Concern: Question on number 5: For prototypes, do we have actual code?

For question number 2, confused about how to explain the scenarios

Additional mention for, All use cases:

- Condense the use case scenarios
- Collaboration class diagram
- Choose three→ petri-nets

Meeting on April 12th notes:

General review of completed diagrams

Questions/concerns for Chef use case diagrams

- Raise quality issue: number 11
- Decide menu: 6 split into 6a and 6b

Meeting on April 17th notes:

ER DIAGRAM

Entities:

1. Chefs
   - Name
   - Employee ID
   - Current orders
     - Food prep status
   - Availability
   - Ratings
   - Status
2. Customers
   - Customer ID
   - Name
   - Payment information
   - VIP Status
   - Warnings
3. Managers
   - Employee ID
   - Name
4. Food importers
   - Employee id
   - Name
   - Status
5. Delivery person
   - Employee ID

    - Name
    - Status
    - Ratings
    - Current delivery
        - Delivery status
6. Ingredients
    - Name
    - Type
    - Stock quantity
    - Expiration date
7. Menu
    - Name
    - Description
        - Allergens
    - Price
    - Availability
8. Feedback/reviews
    - Category
    - Feedback ID
    - Date
    - Handled by

7. Address of the git repo (github, gitlab, bitbucket, etc) of your team's work so far - put all materials including this report there

Repo Link: https://github.com/VerKoval/CulinaryCloud