

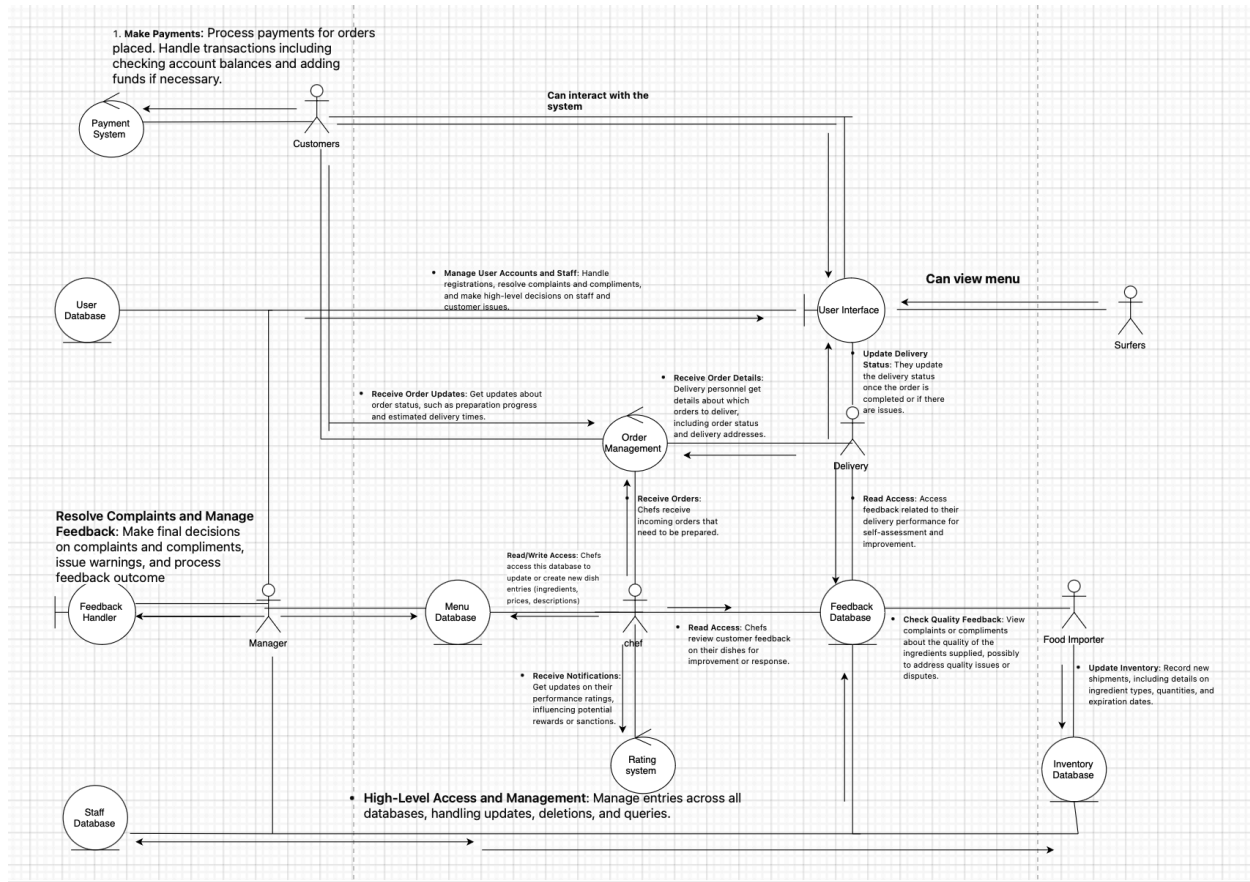
PHASE 2 REPORT

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:



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

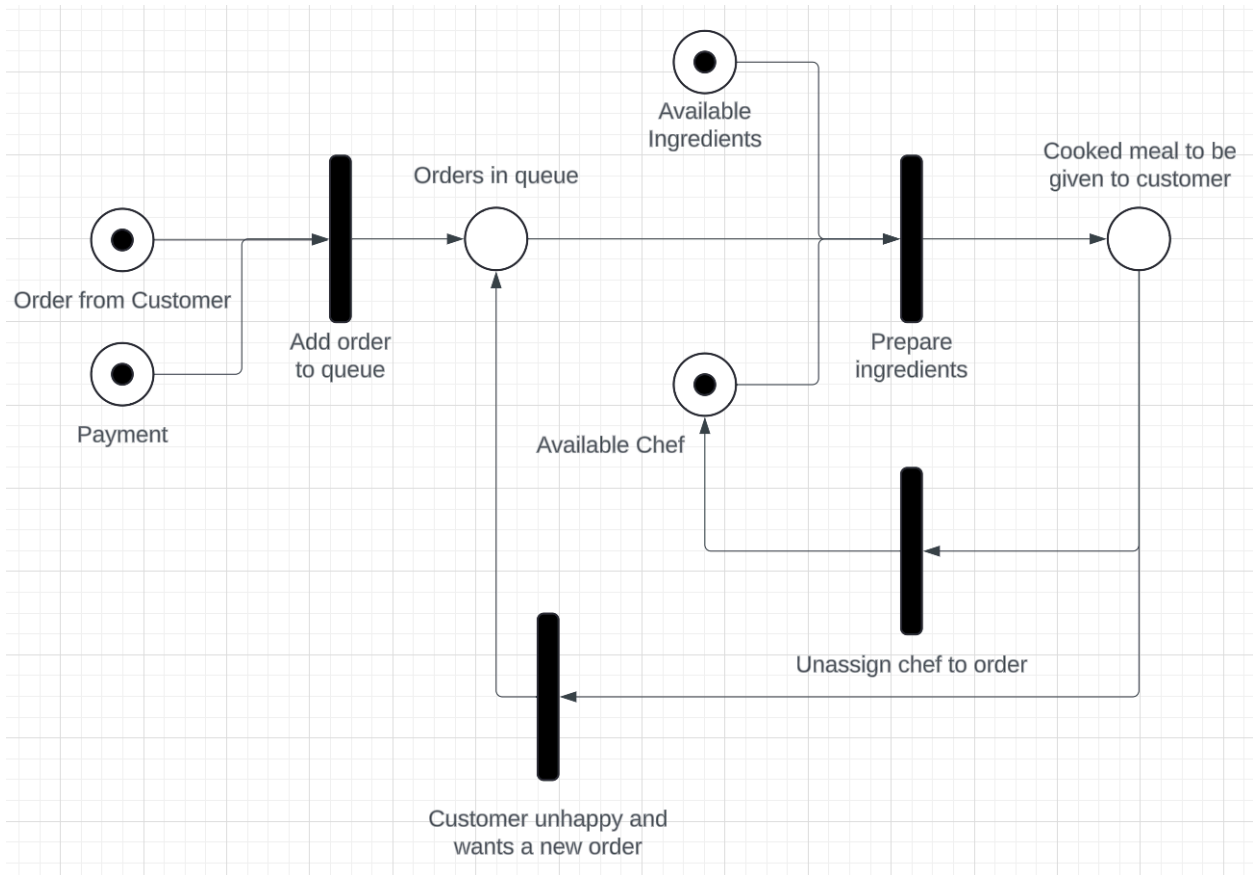
- Will use the same functions as the raise quality issue use case found in Chefs (see above)
- Manager
 - Manage staff
 - Managers will have access to a database of staff profiles who work at the restaurant
 - This database will have:
 - Keys – Unique Employee ID
 - Attributes:
 - Name
 - Position/Role
 - 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

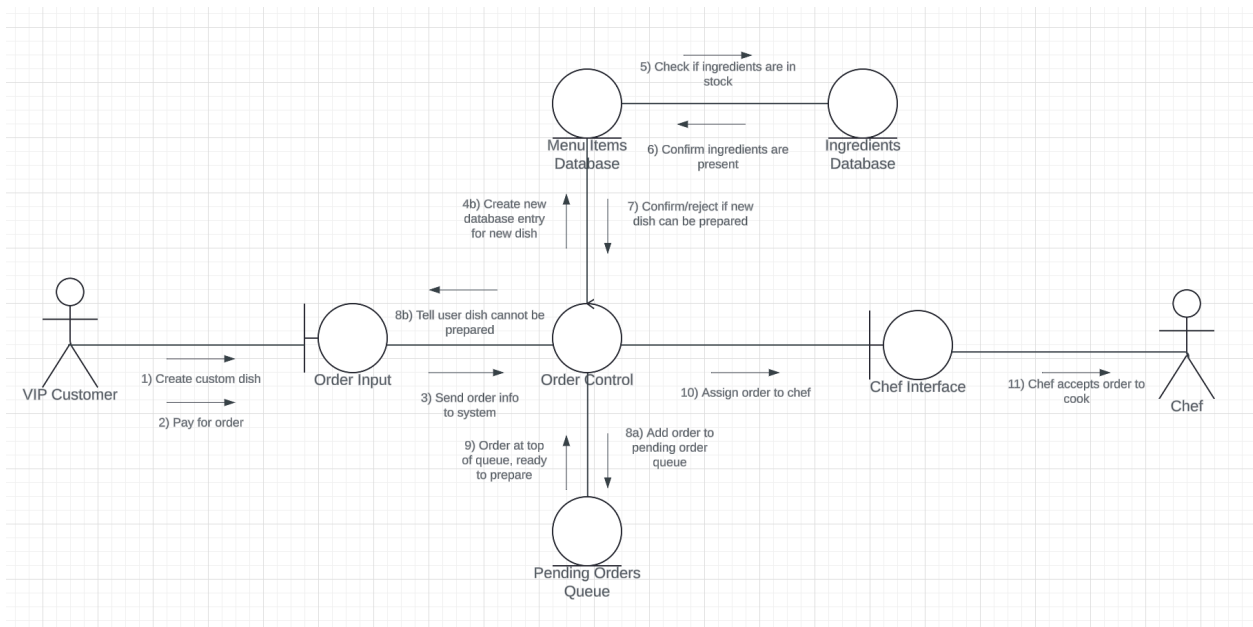
Diagrams:

CHEF

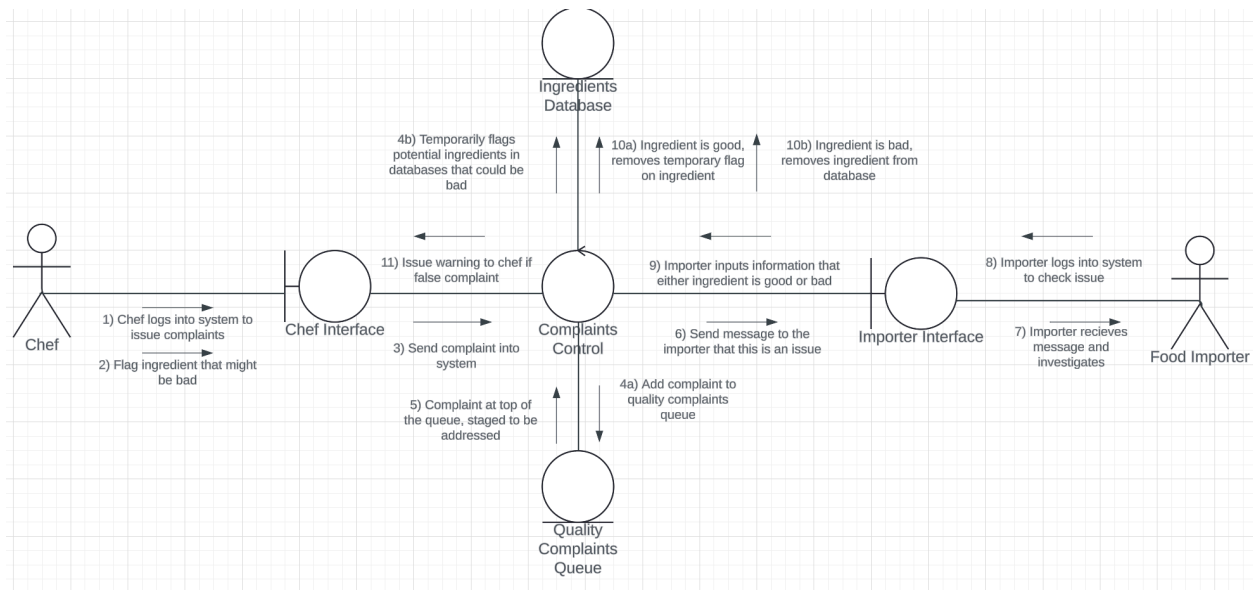
Use Case 1 (Prepare Order): Petri Net



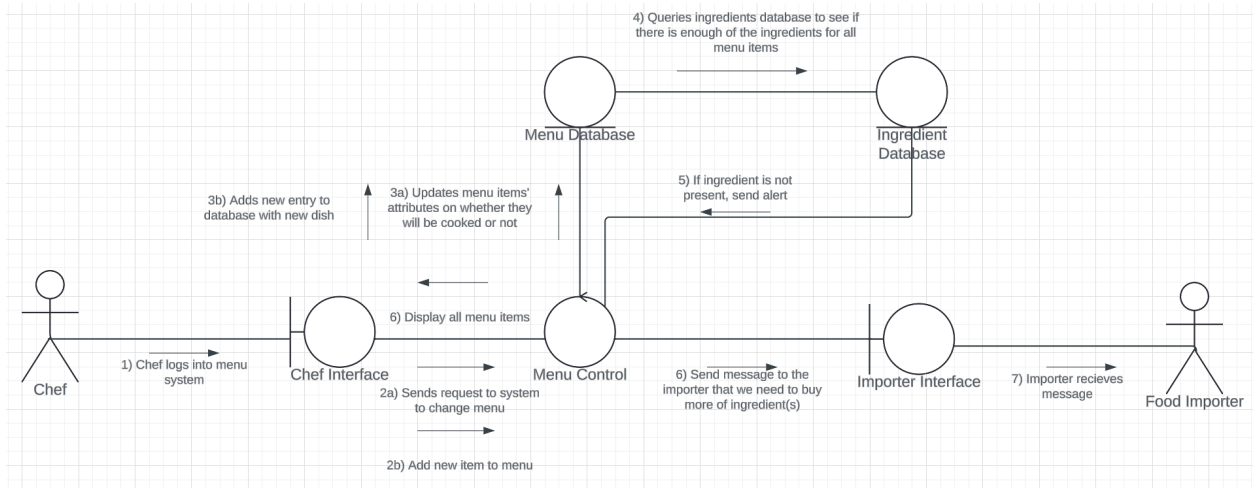
Use Case 2 (Special Order Handling): Collaborative Class Diagram



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

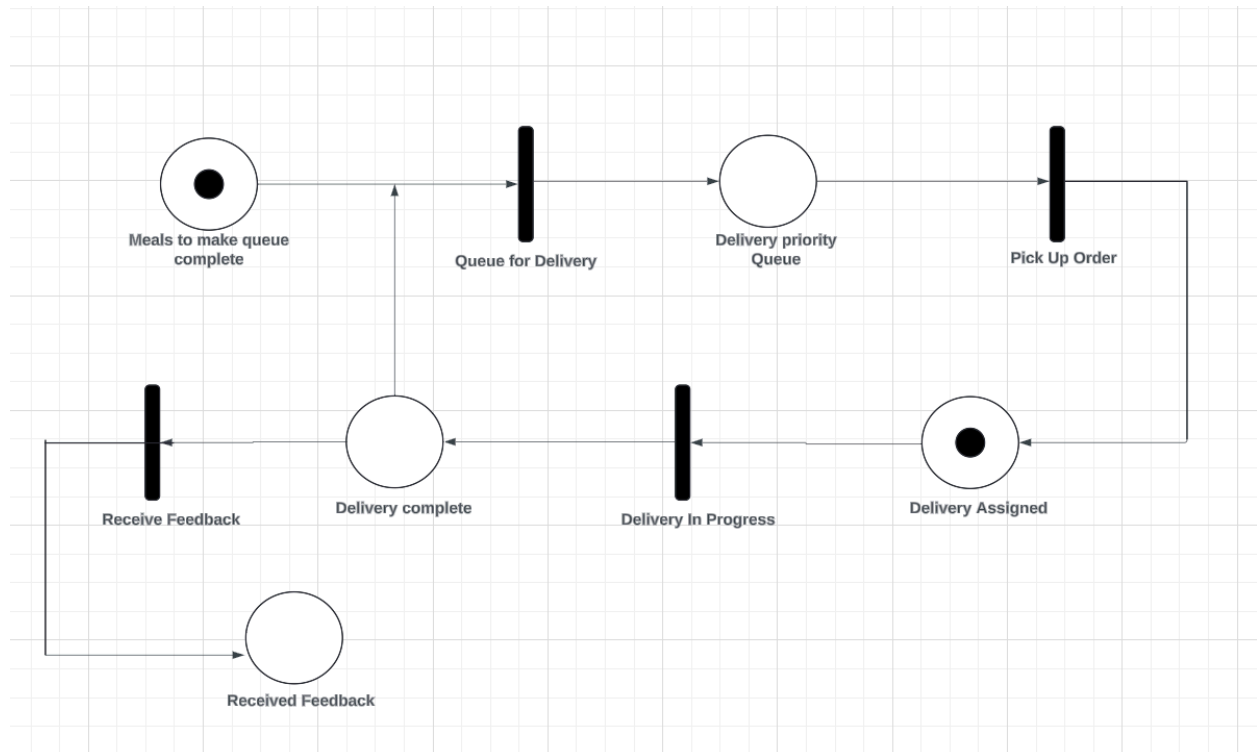


Use Case 4 (Decide Menu): Collaborative Class Diagram

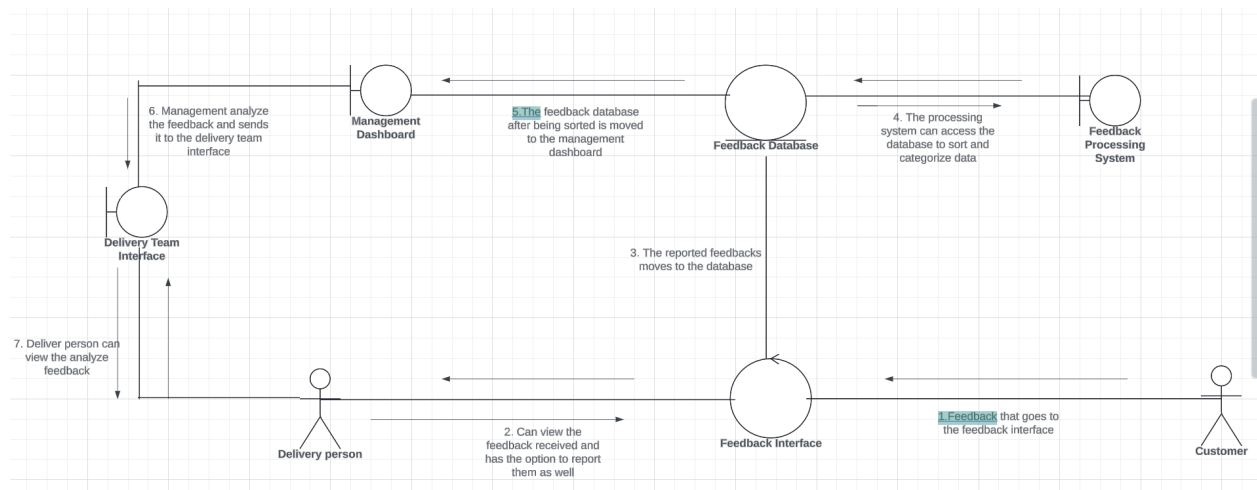


DELIVERY

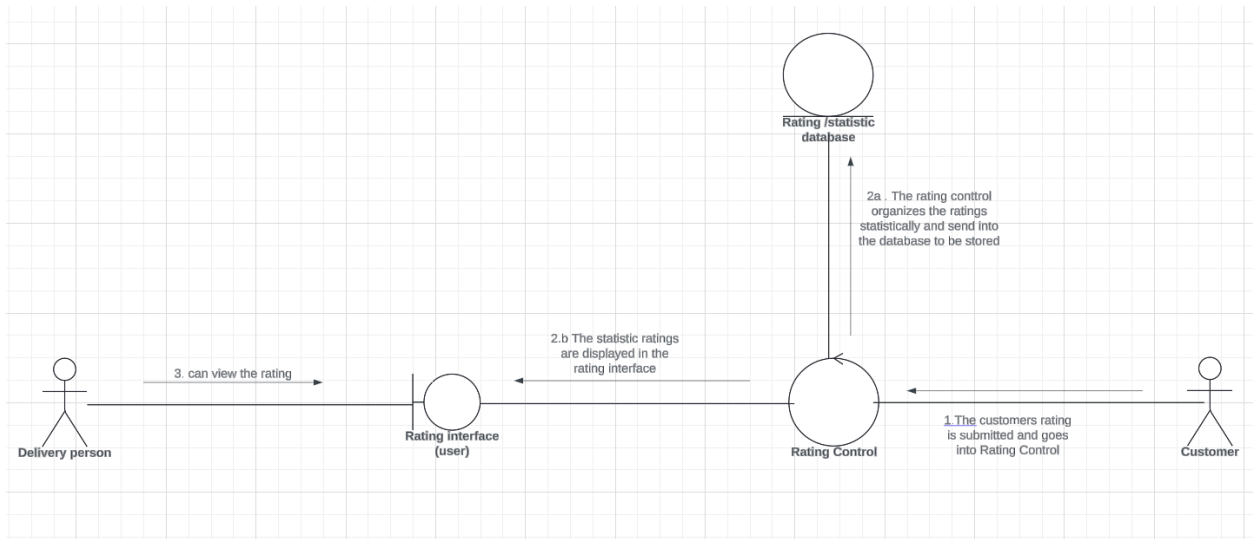
Use Case 1 (Accept delivery): Petri net



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

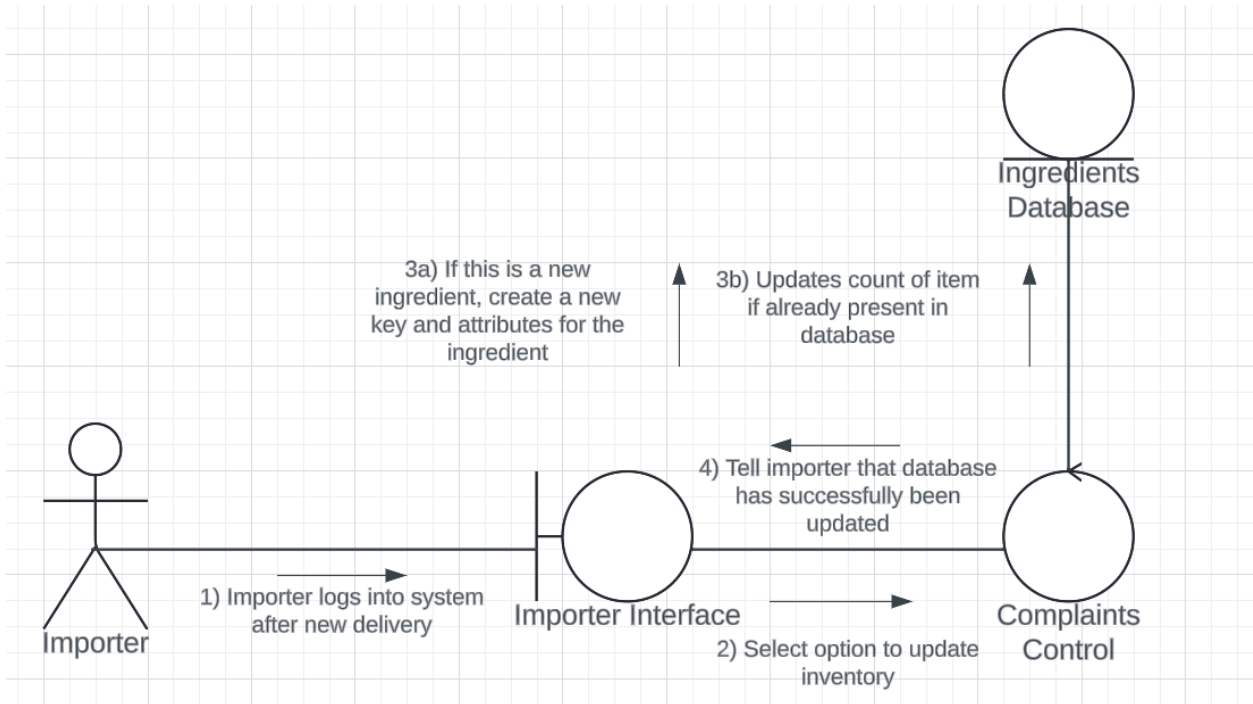


Use Case 3 (View delivery ratings): Collaborative Class Diagram

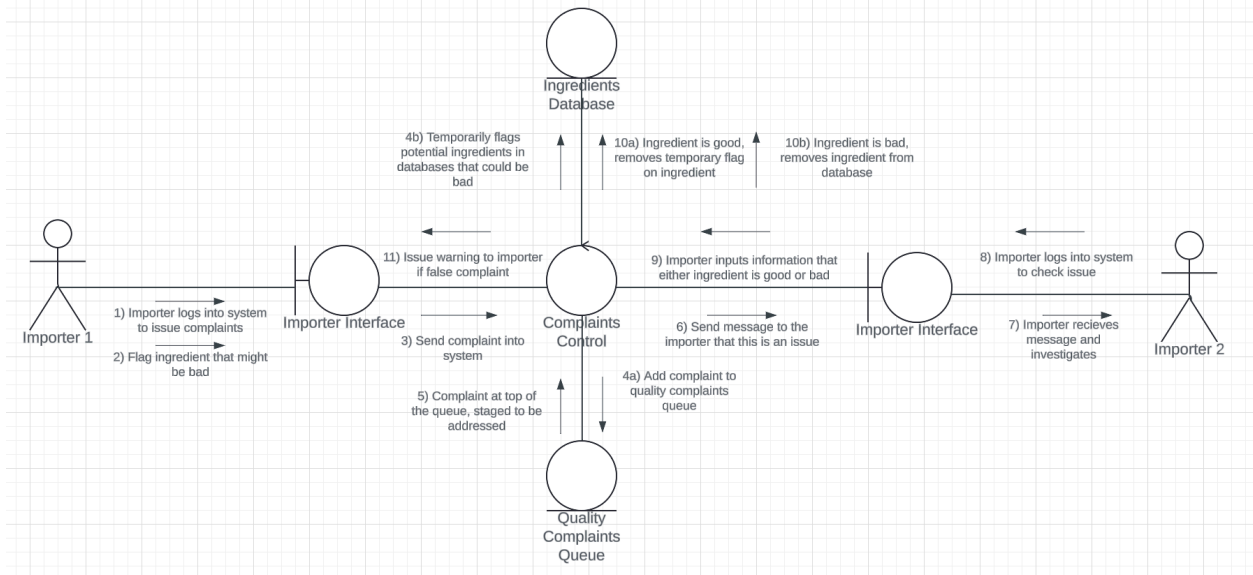


FOOD IMPORTERS

Use Case 1 (Manage Inventory): Collaborative Class Diagram

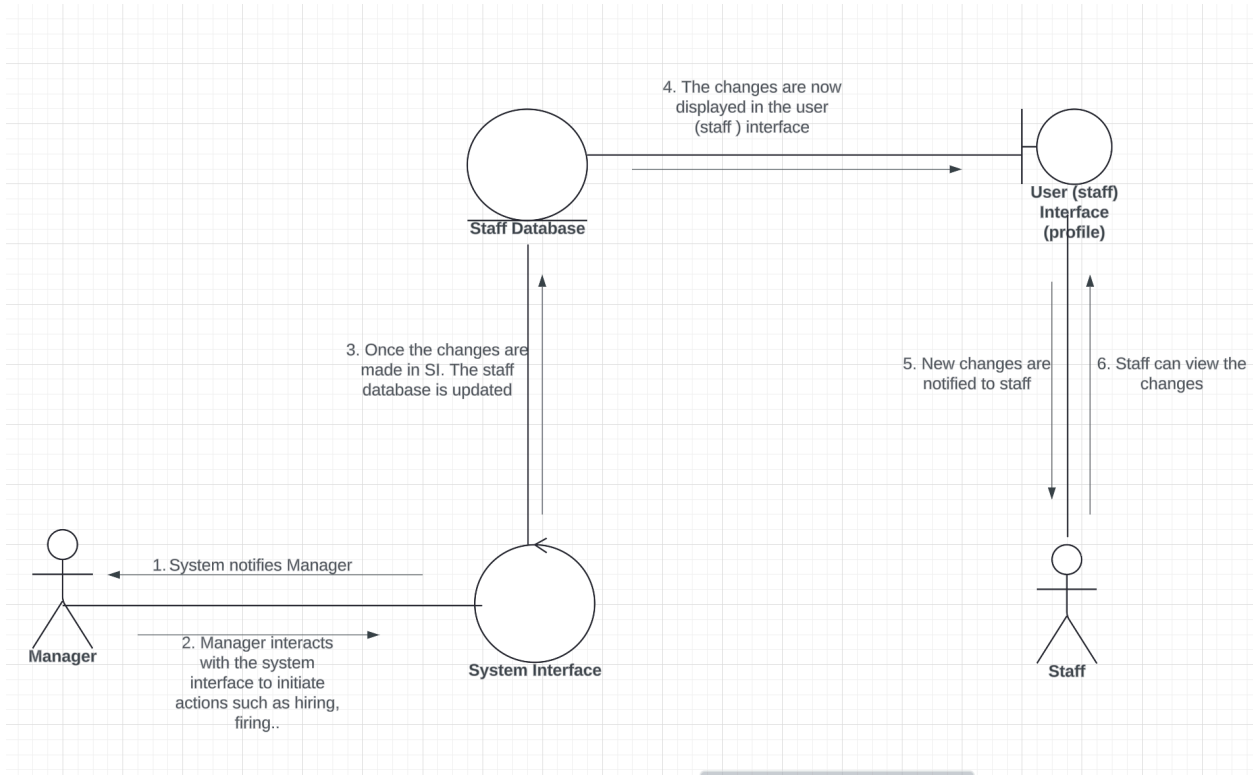


Use Case 2 (Raise Quality Issue): Collaborative Class Diagram

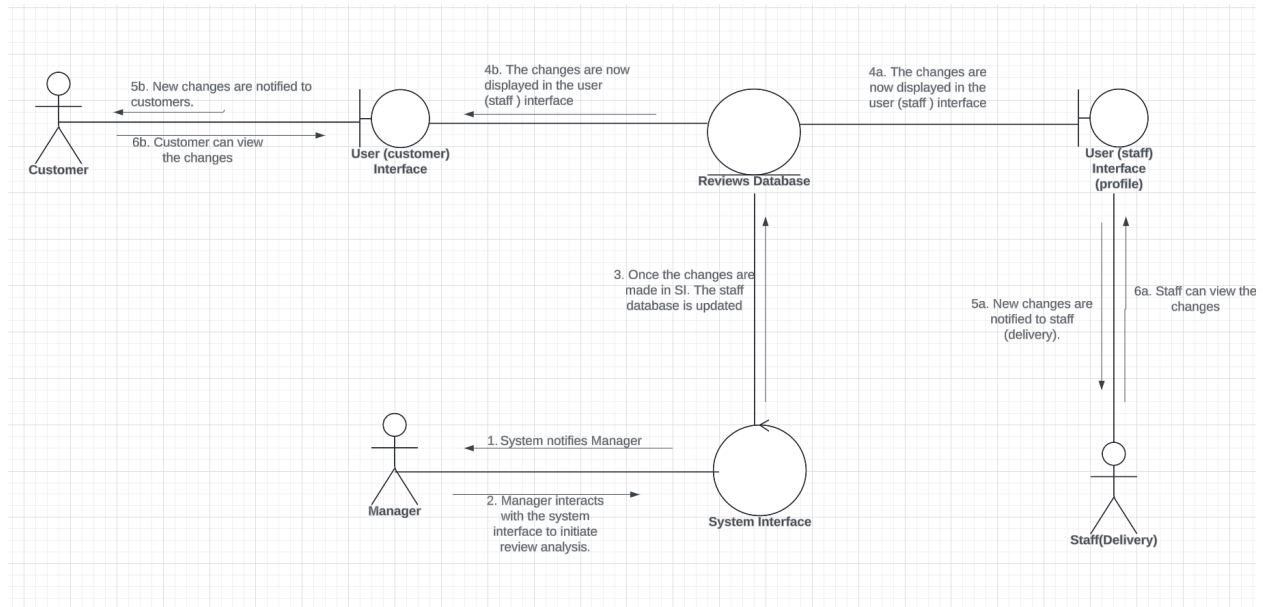


MANAGERS

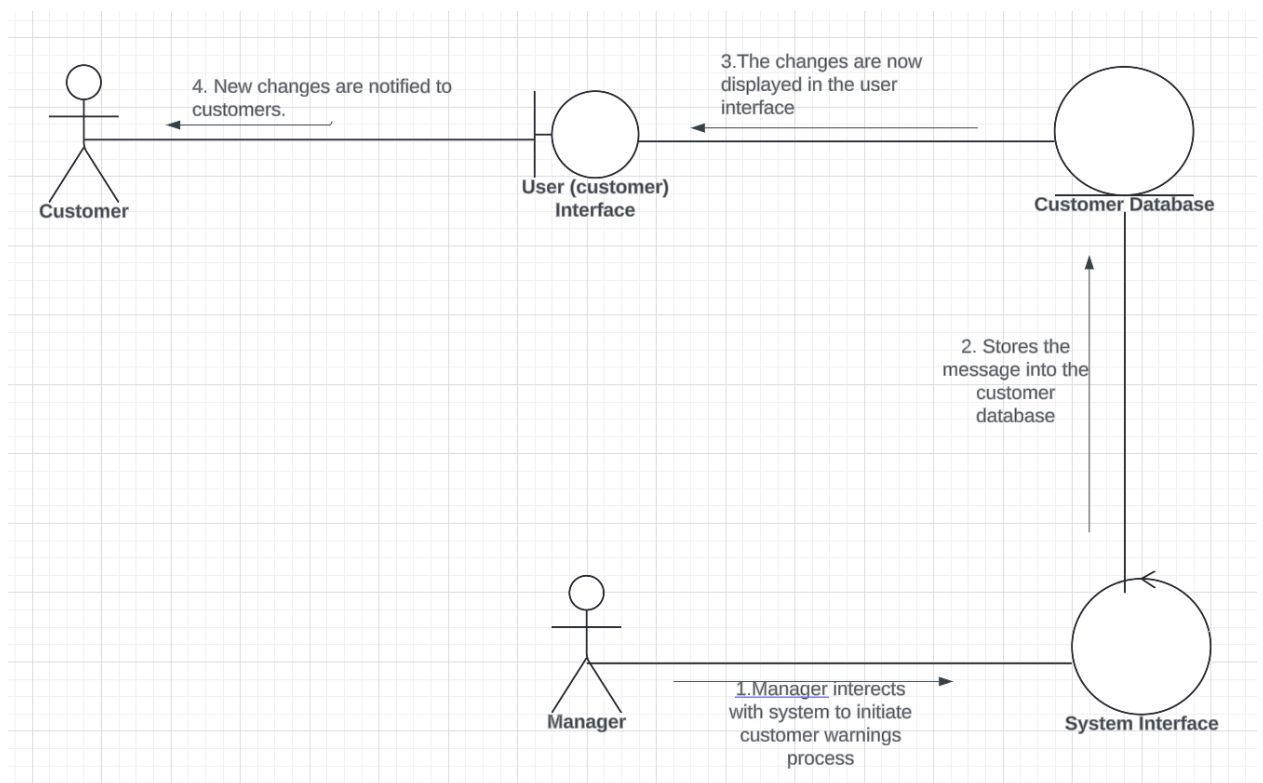
Use Case 1 (Manage staff): Collaborative Class Diagram



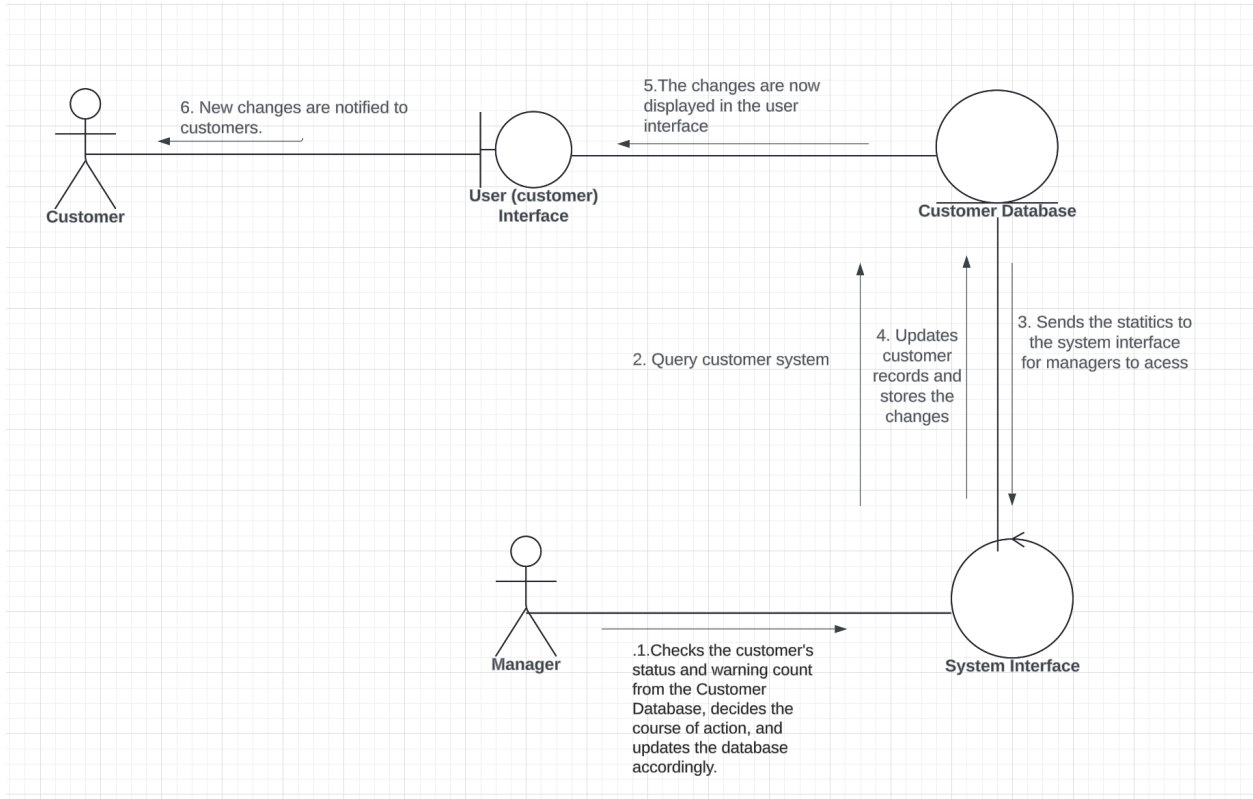
Use Case 2 (Handles review): Collaborative Class Diagram



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

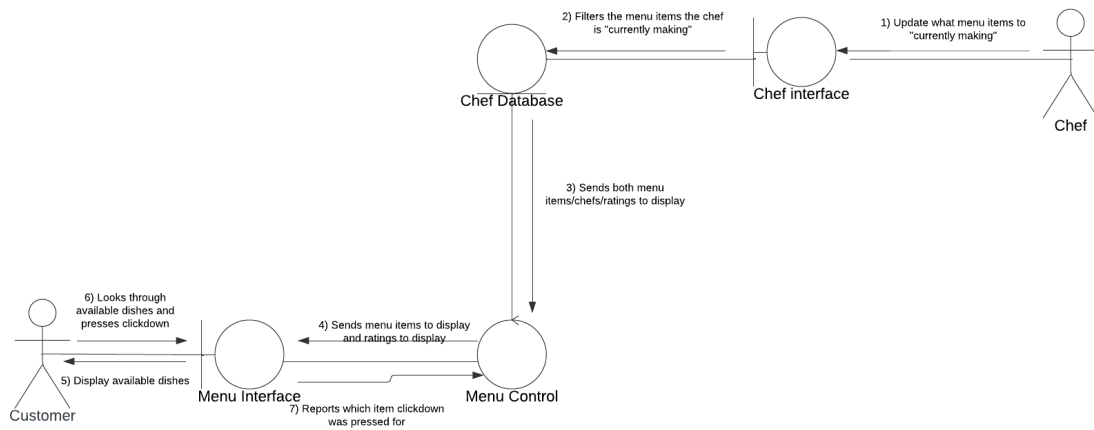


Use Case 4 (Deregister): Collaborative Class Diagram

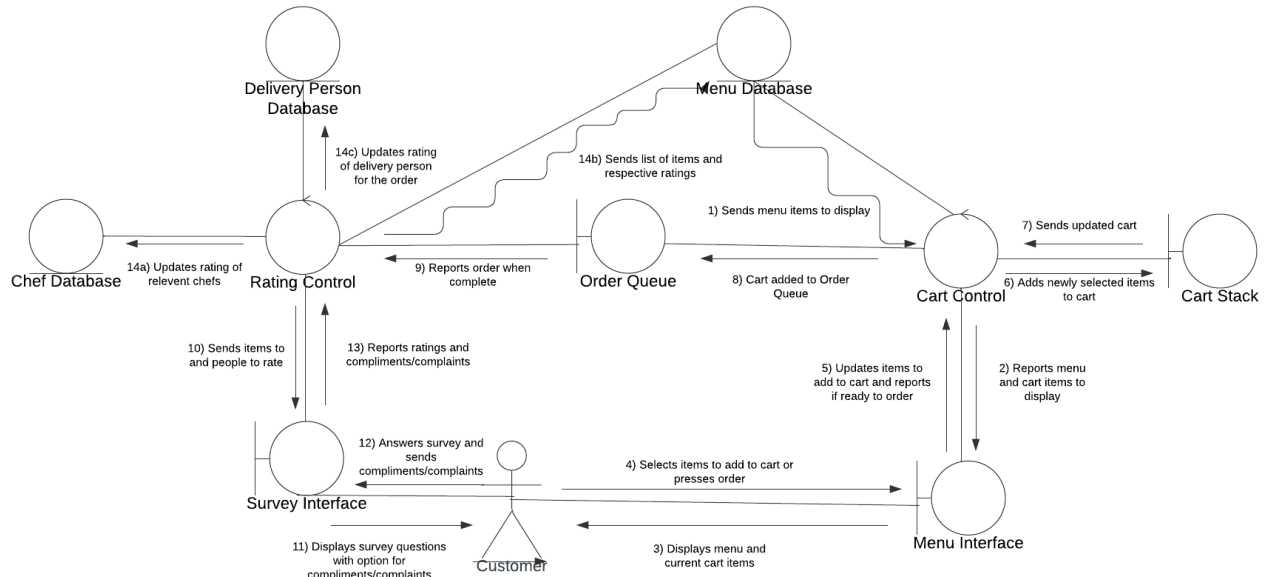


CUSTOMERS

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

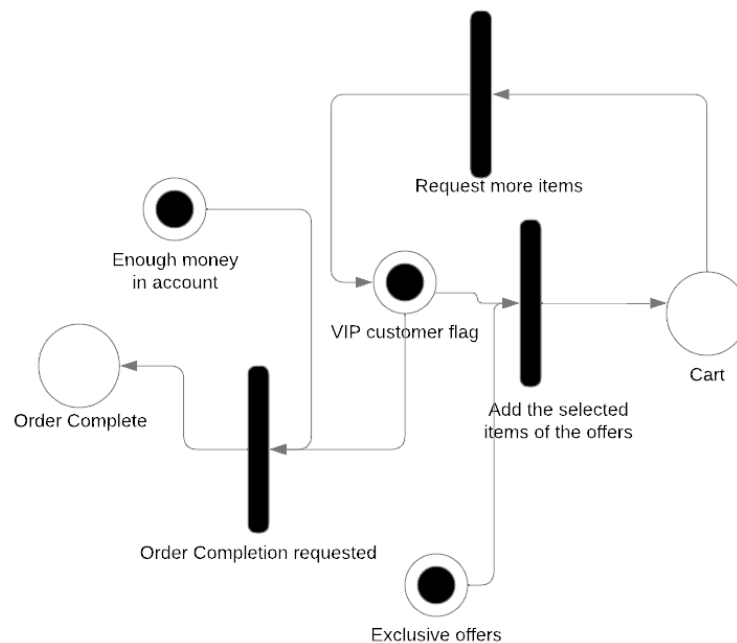


Use Case 2 (Only customers use case): Collaborative Class Diagram

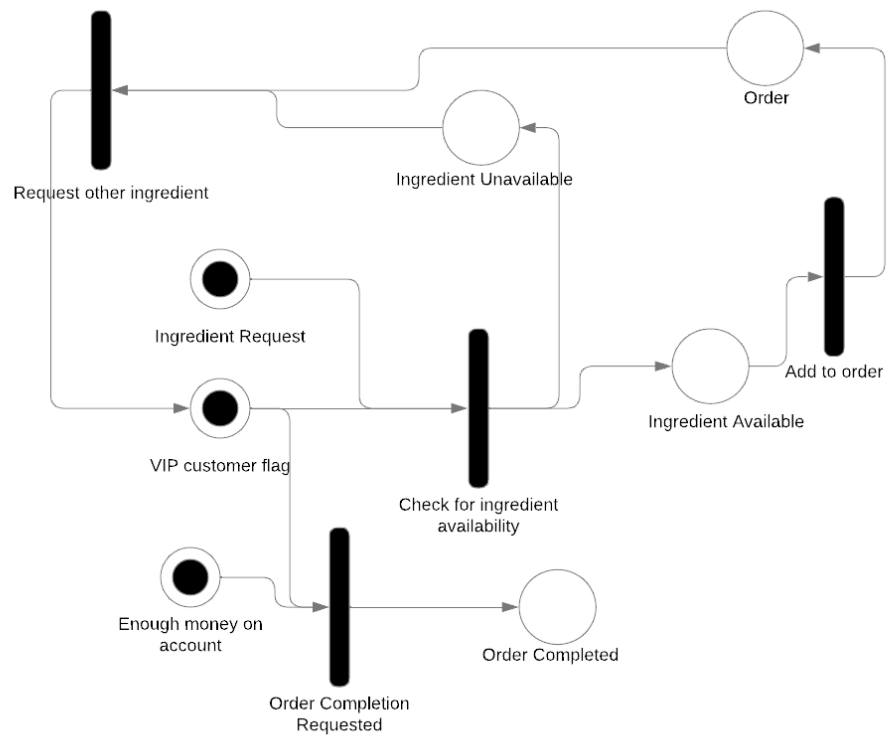


Use Case 3 (only VIP customers use case): Petri-net

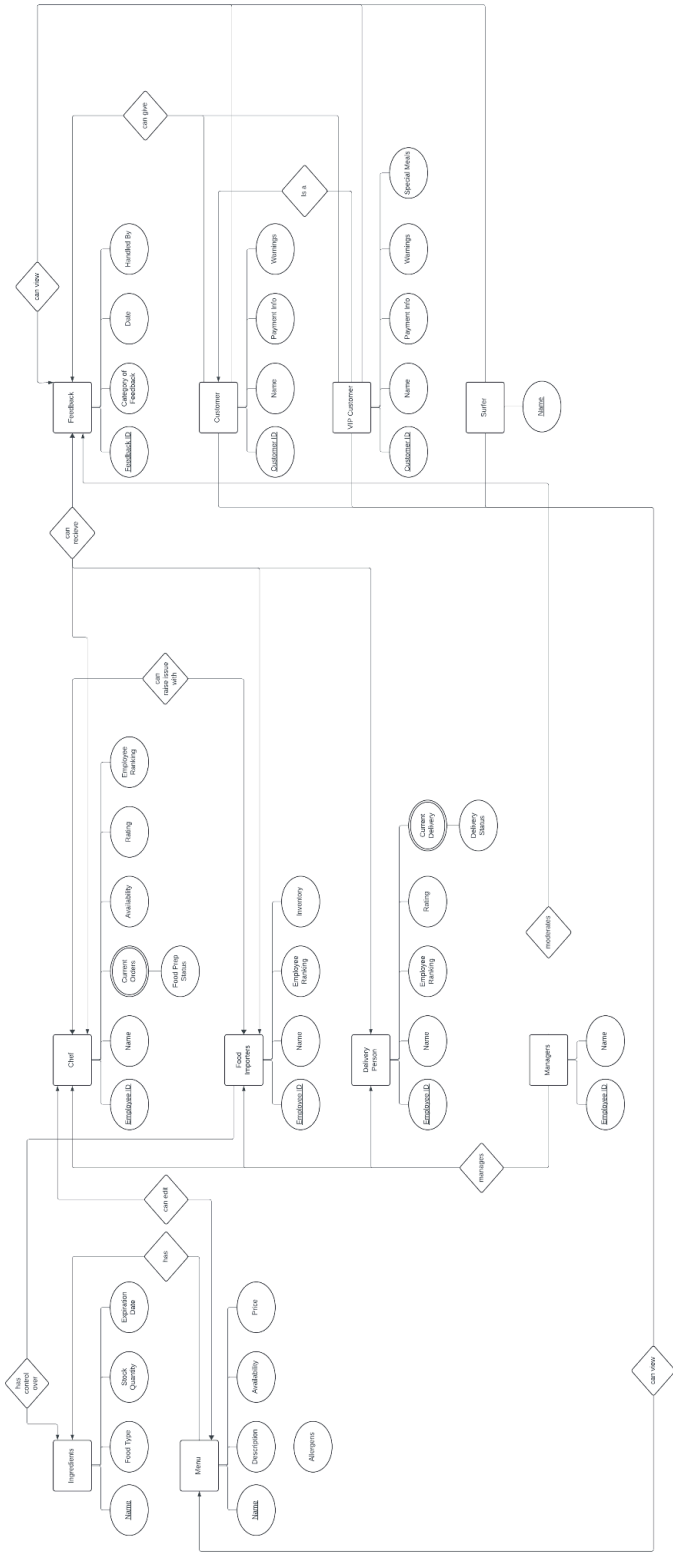
a) Exclusive Offers



b) Customized order



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



4. Detailed design:

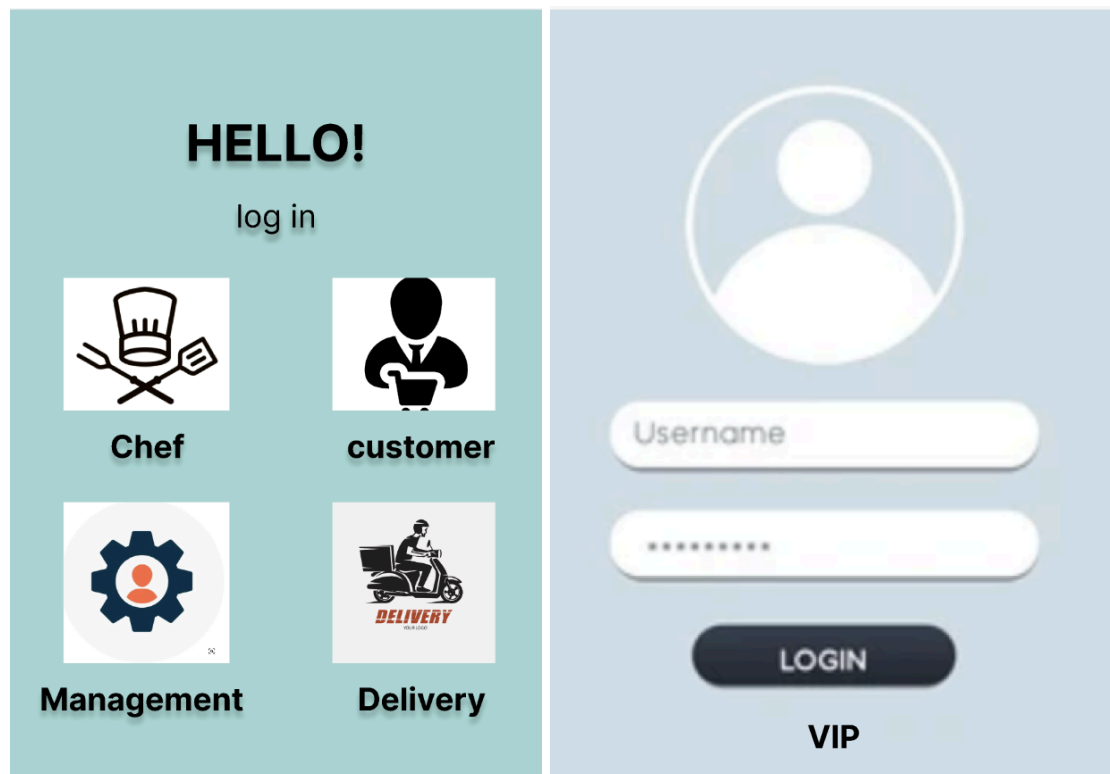
for EVERY method, pseudo-code to delineate the input/output and main functionalities

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>