2) Draw a coffee coffee day ordering system. A coffee coffee day shop vending machine dispenses coffee to customers. Customers order coffee by selecting a recipe from a set of recipes. Customers pay for the coffee using coins. Change is given back, if any, to the customers. The 'service assistant' loads ingredients (coffee powder, milk, sugar, water, chocolate) into the coffee machine. The 'service assistant' adds recipe by indicating the name of the coffee, the units of coffee powder, milk, sugar, water, chocolate to be added as well as the cost of the coffee. The service assistant can also edit and delete a recipe. Develop the use case diagram for the specification above.

Aim:

To develop a use case diagram for the Coffee Coffee Day Ordering System, which models the interactions between customers, the vending machine, and the service assistant for ordering and managing coffee recipes.

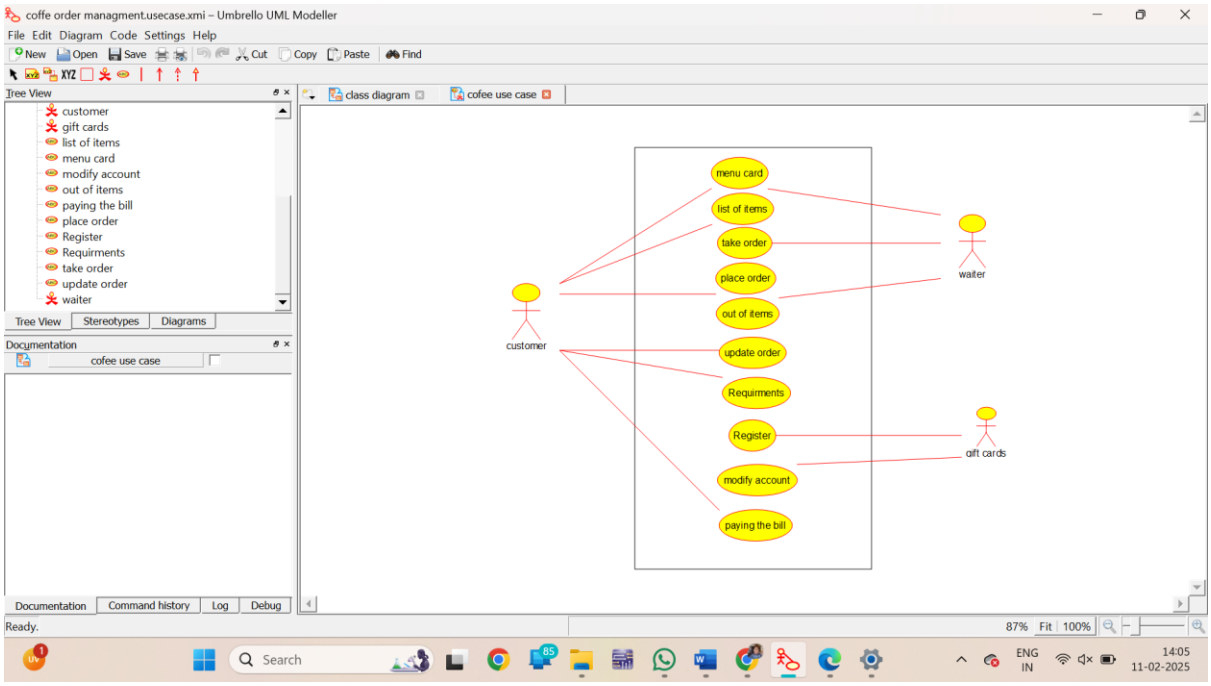
Procedure:

1. Identify the Use Cases:
    - Understand the functionalities of the coffee vending system, such as ordering coffee, paying for coffee, getting change, loading ingredients, and managing recipes.
2. Define Actors:
    - Identify the key actors:
        - Customer (who orders and pays for coffee)
        - Vending Machine (dispenses coffee and handles payments)
        - Service Assistant (manages ingredients and recipes)
3. Draw the Use Case Diagram Elements:
    - Represent actors as stick figures.
    - Represent use cases as ovals.
    - Use associations (lines) to show interactions between actors and use cases.
4. Create Use Cases for Customer:
    - Order Coffee
    - Select Recipe
    - Pay for Coffee
    - Receive Coffee
    - Get Change (if applicable)
5. Create Use Cases for Service Assistant:
    - Load Ingredients

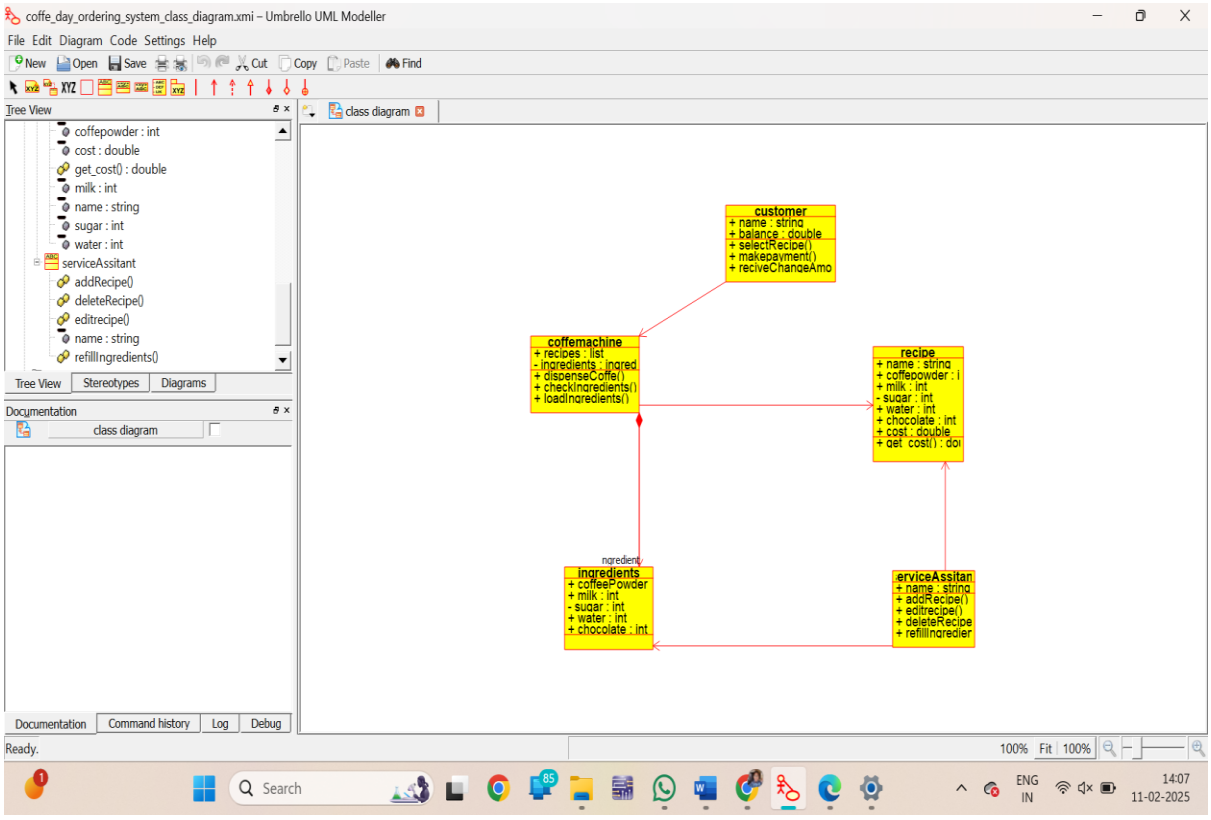
- Add Recipe
- Edit Recipe
- Delete Recipe

6. Draw System Boundaries and Relationships:
    - Enclose the use cases inside a system boundary box labeled "Coffee Vending Machine System".
    - Connect actors to their respective use cases using solid lines.
7. Verify and Validate the Use Case Diagram:
    - Ensure all functionalities are captured and the relationships are correctly mapped.
    - Check for completeness and correctness of interactions between the system and actors.

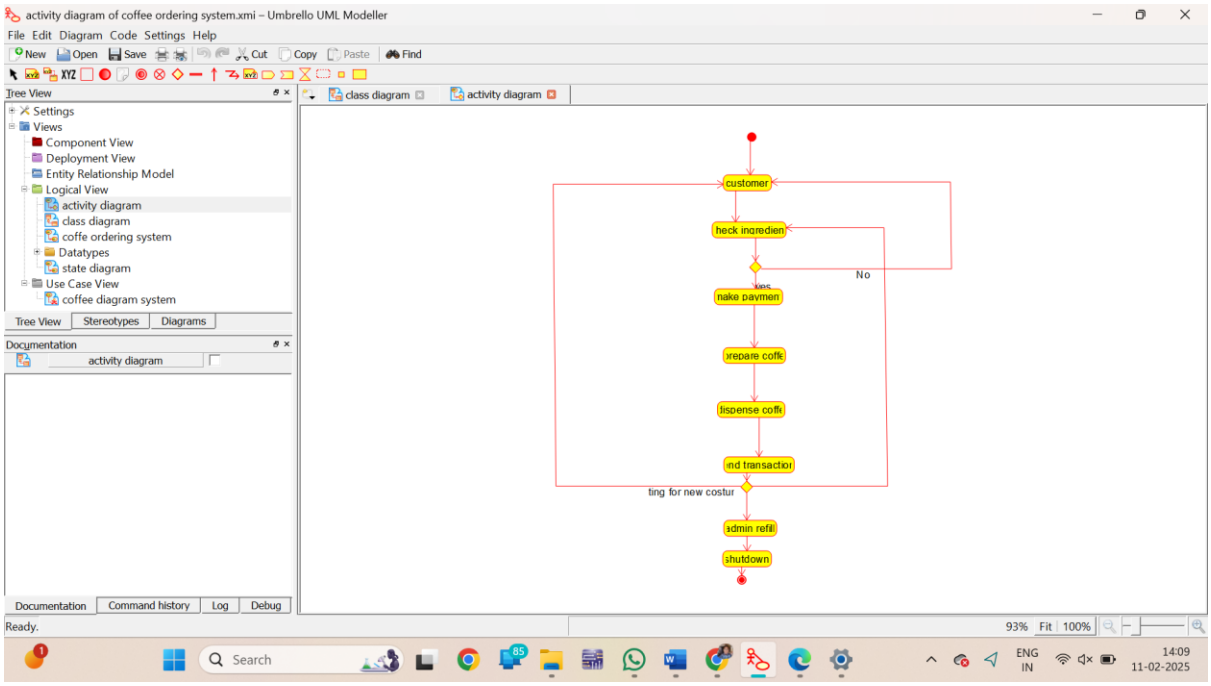
Use Case Diagram:



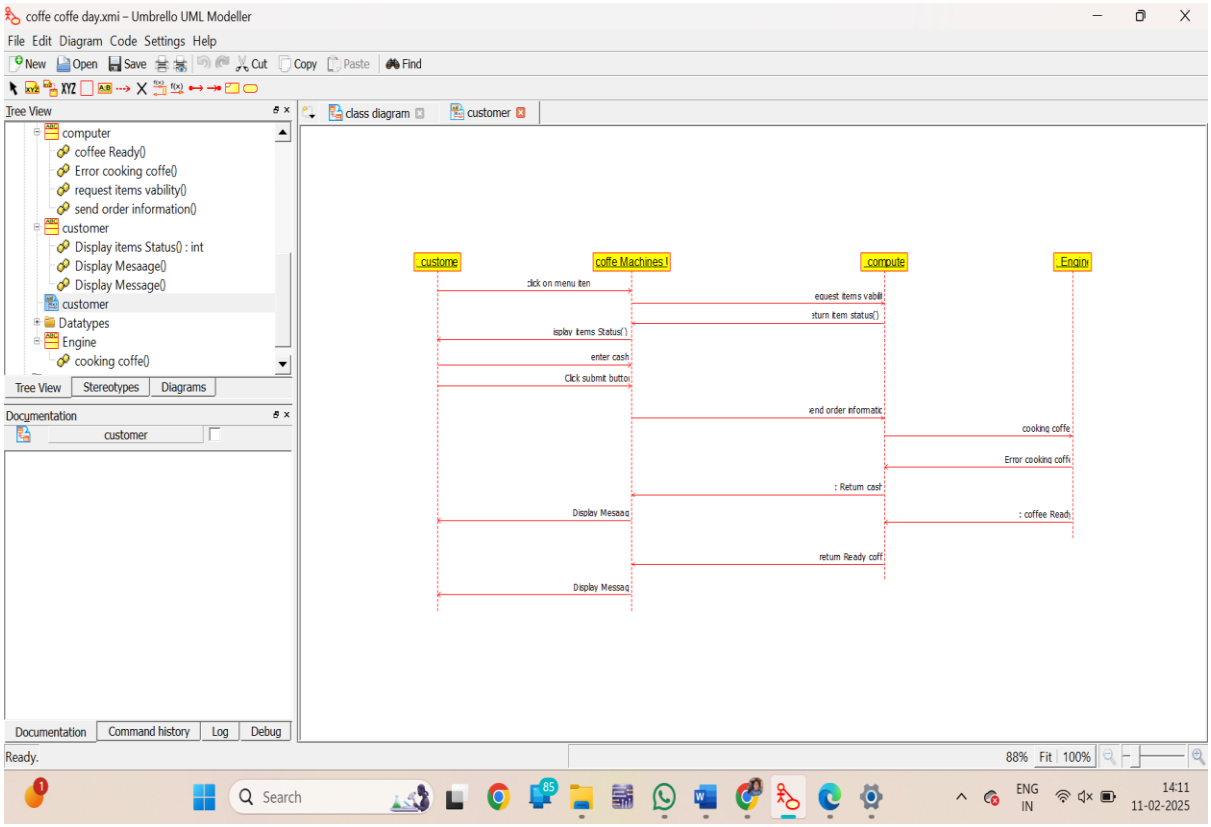
Class Diagram:



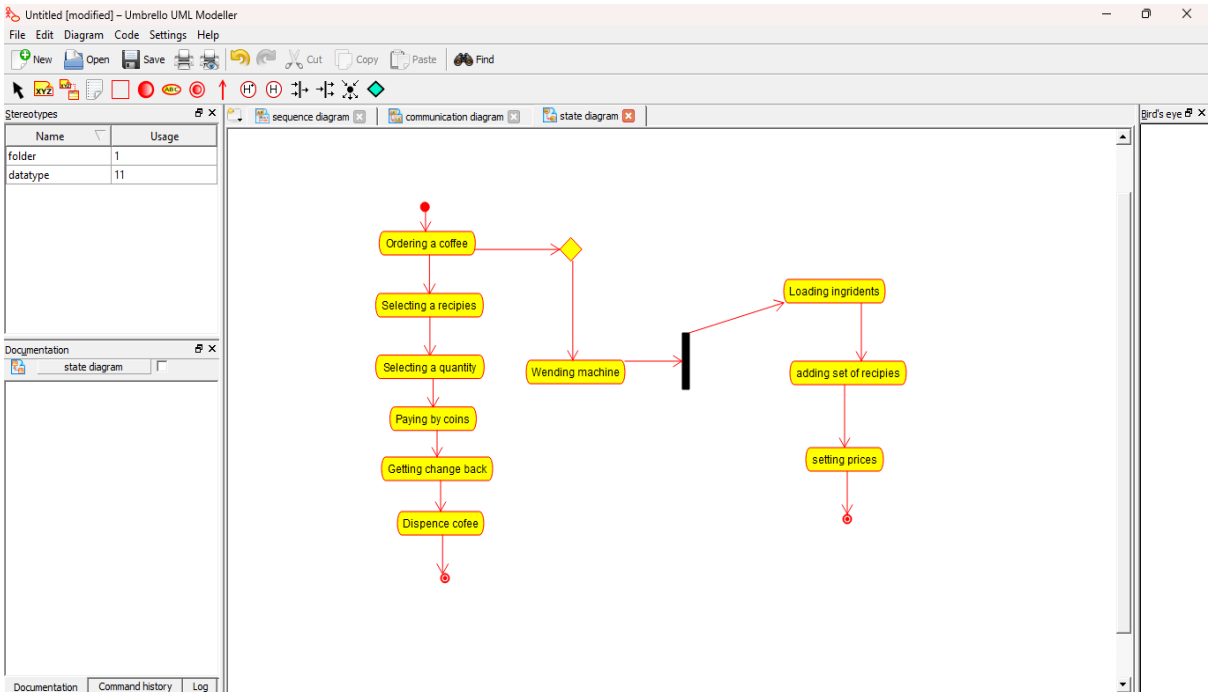
Activity Diagram:



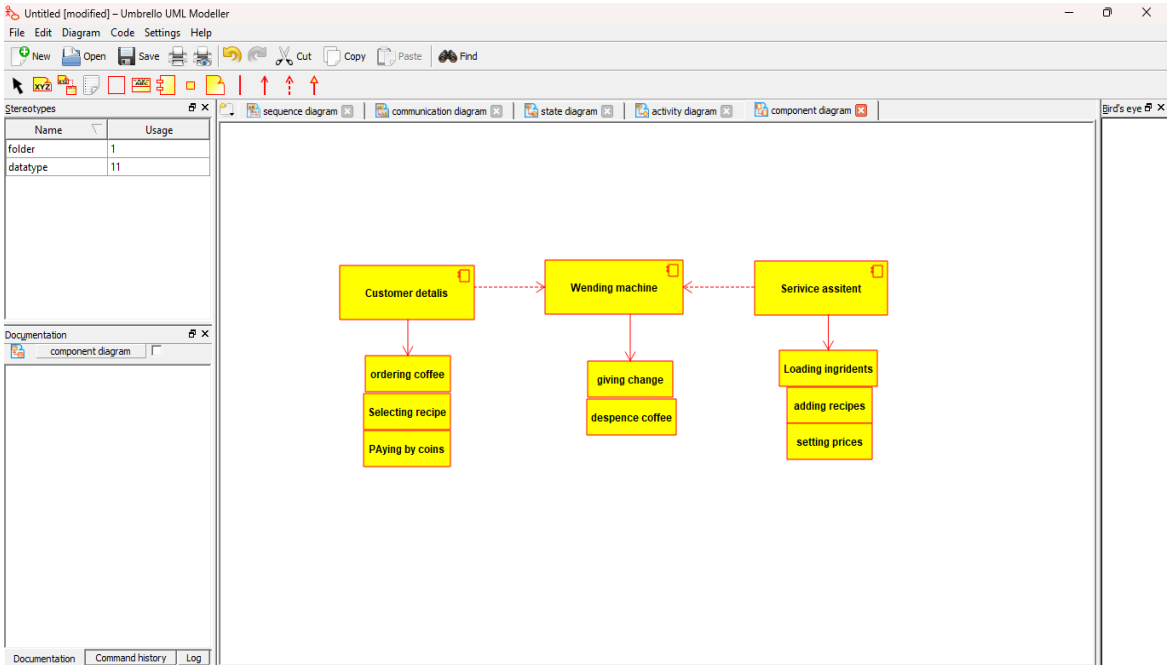
Sequence Diagram:



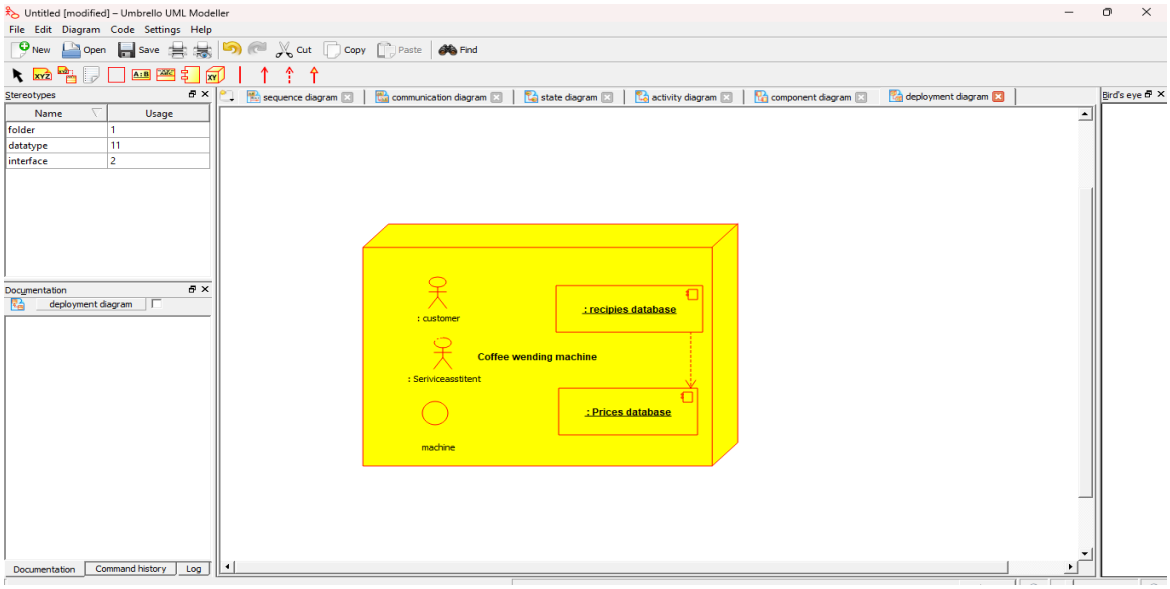
State Diagram:



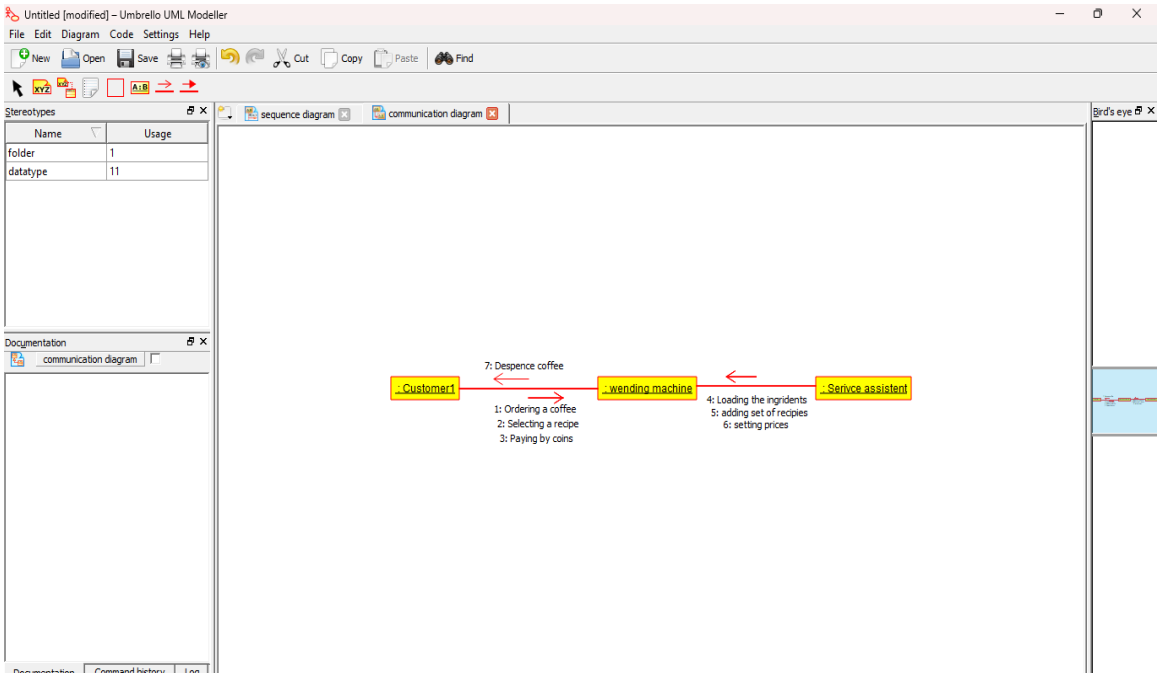
Component Diagram:



Deployment Diagram:



Communication Diagram:



Result:

A use case diagram for the Coffee Coffee Day Ordering System is successfully developed, illustrating how customers interact with the vending machine to order and pay for coffee, and how service assistants manage ingredients and coffee recipes.