



AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

**Dept. of Computer Science
Faculty of Science and Technology**

CSC 2210: OBJECT ORIENTED PROGRAMMING 2

Fall 2025-2026

Section: AA

Group No: 7

Project Report On

Restaurant Management System

Supervised By

WAHIDUL ALAM RIYAD

Submitted By:

Name	ID
1. TANVIR MAHTAB	23-50106-1
2. MD. DAUD IBRAHIM	22-49760-3
3. MD. FARDIN RAHMAN MRIDUL	22-49521-3
4. KAZI ARMAN ALAM	23-50089-1

CO2: Display and verify the mean of a real-life Project using the concepts of C# Graphical User Interface based environment with database integration to depict a desktop-based application.

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)
Evaluation Criteria	Evaluation Definition				Total =
Requirement fulfillment	Properly demonstrate a real-life scenario-based project with proper functional requirement identification for the Object-Oriented Programming project development activities.				
Validation	Ensuring the ability of students' proper demonstration on validation forms in their system in terms of dealing with the data.				
Verification	Identifying if the students can verify the system data along with proper functional requirements in terms of data flow.				

Table of Contents

1. INTRODUCTION	3
2. FEATURE LIST	3
2.1 User 1	
2.2 User 2	
2.3 User 3	
3. ER DIAGRAM	5
3.1 Normalization	5
3.2 Finalization	7
4. UML DIAGRAM	8
4.1 Use Case Diagram.....	8
4.2 Activity Diagram.	9
5. CONCLUSION.....	11

1. INTRODUCTION

This Restaurant Management System is a desktop-based application developed using C# with Windows Forms (WinForms). For the data, we used SQL Server in the back. Basically, it's all about making restaurant work easier by giving them one spot to deal with menus, folks using the system, orders, bills, and sales reports.

A lot of restaurants still do things by hand, like taking orders and keeping records. Doing it that way wastes time and can lead to mistakes, lost info, and just not getting things done fast. This project tries to fix those problems by using a computer system to keep everything safe digitally and do things right away.

The system is built with what we call a three-tier setup: how it looks (Presentation), how it works (Logic), and how it stores data (Access). This keeps things tidy, easy to fix, and ready to add more onto later. It can let different people get to different stuff. Admins get to control users and menus, customers place orders, and staff print receipts.

So, this Restaurant Management System is reliable and easy to use, which makes running a restaurant simpler. It also shows how helpful object-oriented programming is while demonstrating the practical use of database management, and CRUD operations

2. FEATURE LIST

The Restaurant Management System is designed using a role-based access approach, where different users are assigned, different permissions based on their responsibilities. The system is flexible and scalable, allowing the addition of more user roles in the future if required. The following subsections describe the features available to different categories of users.

2.1 User 1 (Administrative User)

- User 1 represents administrative-level users who are responsible for managing and controlling the overall system.
- Secure login and logout functionality
- Access to a centralized administrative dashboard
- Add new food items to the restaurant menu
- View all available menu items
- Update existing menu information such as food name, category, and price
- Delete menu items that are no longer available
- Create new user accounts with appropriate access roles
- View all registered users in the system
- Update user credentials and role assignments
- Delete inactive or unnecessary user accounts
- View all customer order records
- Generate sales reports
- Automatic calculation of total sales revenue

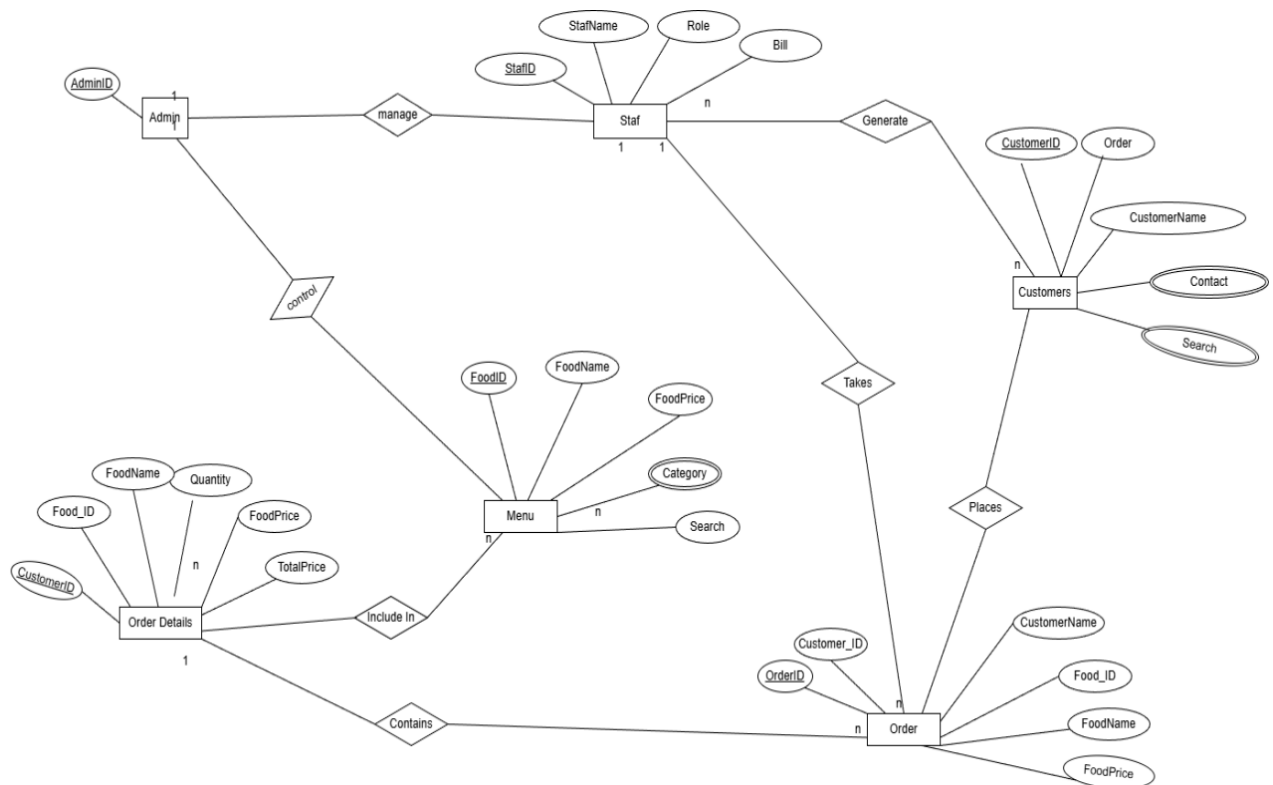
2.2 User 2 (Customer User)

- User 2 represents customers who use the system to view the menu and place food orders.
- Customer registration through a sign-up process
- Secure login and logout functionality
- View the complete restaurant menu
- Browse food items with category and pricing information
- Select food items and specify quantities
- Automatic calculation of total order price
- Place food orders successfully
- View order summary including food items, quantity, and total price

2.3 User 3 (Operational / Staff User)

- User 3 represents operational staff members responsible for handling customer orders and billing.
- Secure login and logout functionality
- View all customer orders in a structured format
- Select specific customer orders for billing
- Generate professional bills containing:
 - Customer identification details
 - Ordered food items
 - Total payable amount
 - Date and time of transaction
- Preview bills before printing
- Print bills using the system's printing feature

3. ER DIAGRAM



Normalization

Admin manage Staff:

Relation: One to Many

1NF:

Multivalue attribute: *Staff_Contact*

AdminID, StaffID, StaffName, Role

2NF:

- AdminID (PK)
- StaffID (PK), Staff_Contact
- StaffID (PK), StaffName, Role, AdminID (FK)

Admin control Menu:

Relation: One to Many

1NF:

Multivalue attribute: *Category*

AdminID, FoodID, FoodName, FoodPrice, Category

2NF:

- AdminID (PK)
- FoodID (PK), Category
- FoodID (PK), FoodName, FoodPrice, AdminID (FK)

Customers place Order:

Relation: One to Many

1NF:

Multivalued attribute: Contact

CustomerID, CustomerName, Contact, OrderID

2NF:

- CustomerID (PK), Contact
- CustomerID (PK), CustomerName
- OrderID (PK), CustomerID (FK)

Staff take Order:

Relation: One to Many

1NF:

StaffID, OrderID

2NF:

- StaffID (PK)
- OrderID (PK), StaffID (FK)

Order contains Order Details:

Relation: One to Many

1NF:

Multivalued attribute: Food Items

OrderID, FoodID, FoodName, Quantity, FoodPrice, TotalPrice

2NF:

- OrderID (PK)
- FoodID (PK), FoodName, FoodPrice
- OrderID (FK), FoodID (FK), Quantity, TotalPrice

Order Details include Menu:

Relation: Many to One

1NF:

FoodID, FoodName, FoodPrice

2NF:

- FoodID (PK), FoodName, FoodPrice

Staff generates Bill:

Relation: One to Many

1NF:

- StaffID, CustomerID, Bill

2NF:

- StaffID (PK)
- CustomerID (PK)
- BillID (PK), StaffID (FK), CustomerID (FK)

Customers search Menu:

Relation: One to Many

1NF:

CustomerID, Search

2NF:

- CustomerID (PK)
- SearchID (PK), CustomerID (FK)

Finalization

- 1) AdminID (PK)
- 2) StaffID (PK), StaffName, Role, AdminID (FK)
- 3) StaffID (PK), Staff_Contact
- 4) CustomerID (PK), CustomerName
- 5) CustomerID (PK), Contact
- 6) FoodID (PK), FoodName, FoodPrice, Category, AdminID (FK)
- 7) OrderID (PK), CustomerID, StaffID (FK)
- 8) OrderID (PK), FoodID, Quantity, TotalPrice
- 9) BillID (PK), CustomerID, StaffID (FK)
- 10) SearchID (PK), CustomerID (FK)

4. UML DIAGRAM

4.1 Use Case Diagram

The Restaurant Management System (RMS) is a desktop-based application designed to manage the daily activities of a restaurant through an integrated digital platform. Previously, restaurant operations such as menu handling, order taking, billing, and sales monitoring were performed manually, which often led to delays, errors, and inefficiencies. The Restaurant Management System automates these operations to improve accuracy, efficiency, and overall service quality.

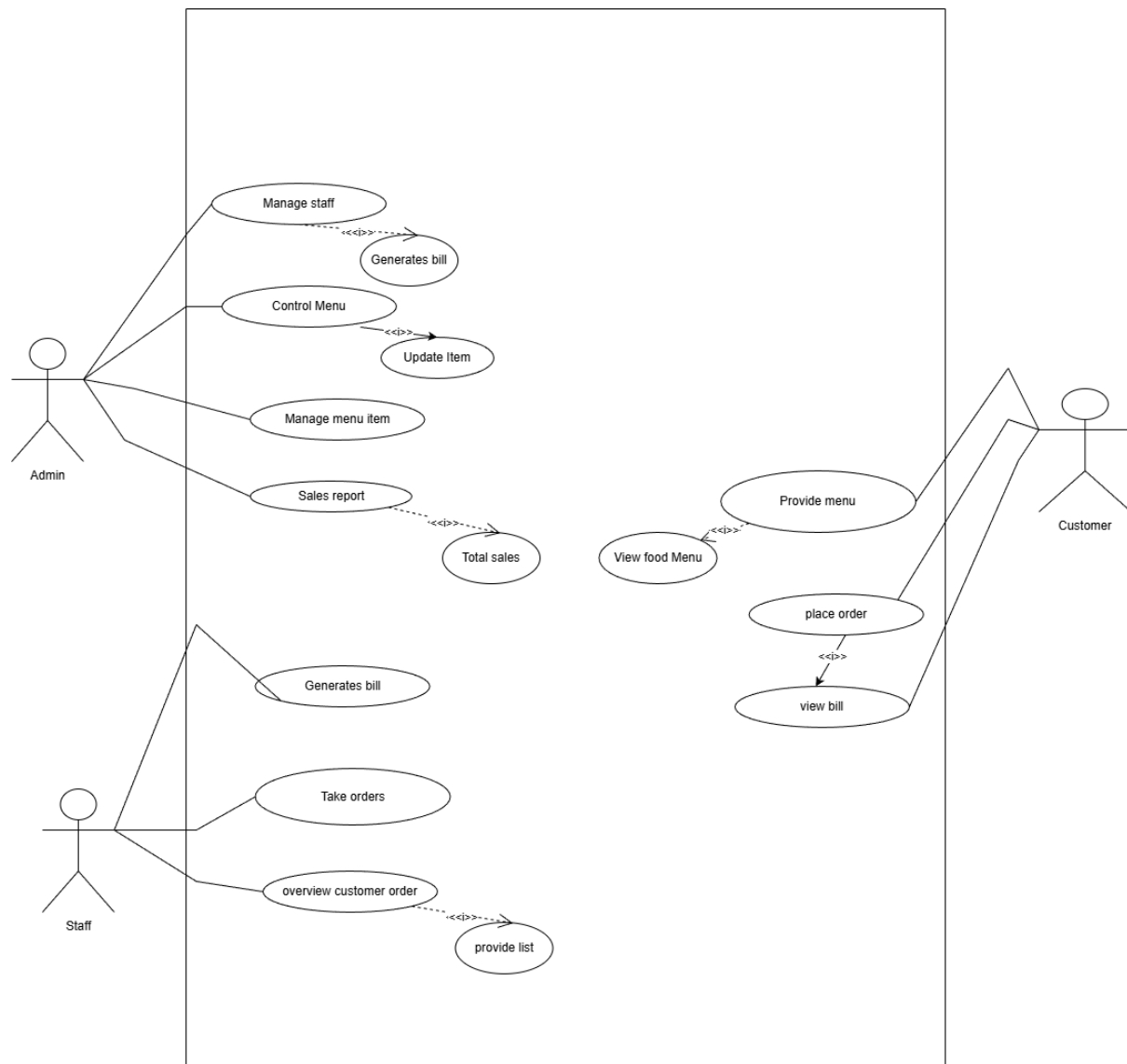
The system consists of three main types of users Admin, Staff, and Customer each having distinct roles and responsibilities as illustrated in the Use Case Diagram.

The Admin has full control over the restaurant system. The Admin is responsible for managing staff members and generating bills related to restaurant operations. The Admin can control the menu and update menu items when necessary. Additionally, the Admin can manage menu items and generate sales reports to view total sales. Through these functionalities, the Admin monitors restaurant performance and ensures that all operational activities are properly managed.

The Staff is responsible for handling operational tasks related to customer orders. Staff members take orders from customers and generate bills as required. They can also give an overview of customer orders and provide order lists to support proper order processing. These actions help maintain smooth coordination between order handling and billing activities.

The Customer interacts with the system to access restaurant services. Customers can view the food menu provided by the system and place orders based on the available items. After placing an order, customers can view their bills to review order details. This allows customers to easily access menu information, place orders, and verify billing information in a convenient and user-friendly manner.

Overall, the Use Case Diagram demonstrates how the Restaurant Management System integrates administrative control, staff operations, and customer interaction into a single unified system. By automating menu management, order processing, billing, and sales reporting, the system reduces manual errors, improves operational efficiency, and enhances customer satisfaction.



4.2 Activity Diagram.

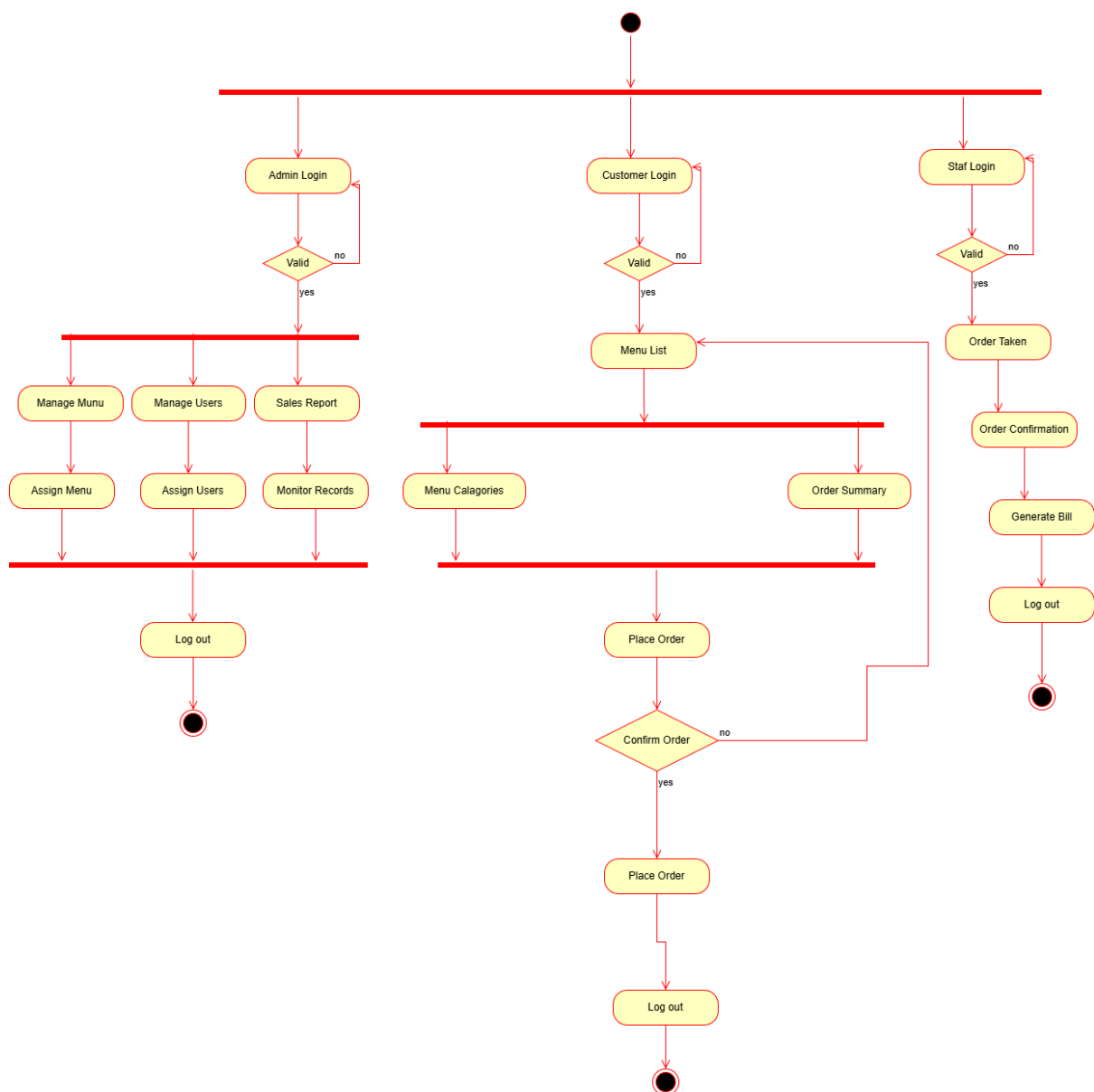
The process of the Restaurant Management System (RMS) starts with a common point where various users, such as Admin, Customer, and Staff, log in to the system based on their respective roles. Every user must go through a secure login process.

After logging in, the Admin process begins with the verification of login credentials. If the login is successful, the Admin obtains access to administrative operations. The Admin has the capability to handle the menu, handle users, and produce sales reports. Menu handling involves assigning and updating menu items, while user handling involves assigning system roles to staff members. The Admin also has the ability to view records and sales data to monitor restaurant activities. After all administrative operations are done, the Admin logs out of the system, thereby completing the Admin process flow.

The Customer activity starts with customer login and validation. After a successful login, the customer can see the menu list. The menu consists of various categories, making it easy for the customer to navigate through the food items. The customer can see the order summary and then proceed to place an order. Before finally submitting the order, the customer

confirms the order. Without confirming the order, the process goes back to the menu list for correction. After the ordering process, the customer logs out, thus completing the customer activity process.

The Staff activity begins with the staff login and validation of credentials. After the successful login, the staff receives customer orders and moves on to the order confirmation process. After the order is confirmed, the staff prepares the bill for the customer. Finally, the staff logs out of the system after completing the billing and order processing tasks. Each user's activity path is concluded with a logout operation, which ensures a secure logout process. The activity diagram shows how the Admin, Customer, and Staff activities are well coordinated, which is evident in the logical flow of activities from login to logout. This activity diagram portrays an efficient workflow that enables proper restaurant management, order processing, and customer service.



5. CONCLUSION

The Restaurant Management System has been successfully designed and implemented as a desktop application using C#, Windows Forms (WinForms), and SQL Server. The system successfully automates major restaurant operations such as menu management, user management, order processing, billing, and sales reporting.

Through the replacement of the traditional manual process by a computerized system, the processing time is reduced, and the accuracy of the data is ensured. The use of role-based access control enables users to carry out their duties while ensuring that the system is secure.

The implementation of three-tier architecture involving the Presentation Layer, Business Logic Layer, and Data Access Layer improves the modularity, scalability, and maintainability of the application. The project also shows the proper application of principles of object-oriented programming, database design, and the implementation of CRUD operations.

Overall, the Restaurant Management System is a very useful, user-friendly, and efficient solution that can be applied to small to medium-sized restaurants. From an academic standpoint, the project is an excellent example of how the frontend design of the user interface, and the backend functionality of the database can be combined using fundamental software engineering principles. The system is designed in such a way that it can be easily expanded in the future with new functionalities like online ordering, reporting, and security features.