PROJECT REPORT

Restaurant Management System

Table of Contents

CHAPTER 1

INTRODUCTION

This project, titled **Restaurant Management System**, is designed to streamline the operations of a restaurant by providing a user-friendly interface for administrators, customers, and staff. It offers features such as menu management, order placement, order history, and logout options, ensuring efficient interaction between different roles in the restaurant ecosystem.

1.1 SYSTEM OVERVIEW

The **Restaurant Management System** is a console-based application divided into distinct modules for different user roles:

1. **Main Menu**: Provides options for users to log in, register, or exit the system.
2. **Admin Menu**: Allows administrators to manage the restaurant's menu by adding, removing, updating, and displaying items.
3. **Customer Menu**: Enables customers to place orders, view their order history, and browse the restaurant menu.
4. **Staff Menu**: Designed for staff members to view customer order history and display the menu.

Each menu is tailored to the specific needs of the user roles, ensuring seamless operation and effective task management.

1.2 WHY DEVELOP THIS SYSTEM?

The **Restaurant Management System** was developed to address the following needs:

1. **Efficiency**: Streamlines restaurant operations by automating tasks such as menu management, order processing, and history tracking.
2. **User Role Segmentation**: Provides dedicated functionalities for administrators, customers, and staff, ensuring clarity and precision in operations.
3. **Ease of Use**: Offers a simple, console-based interface that minimizes the learning curve for users.
4. **Data Organization**: Keeps track of menu items and order histories, enabling better decision-making and service quality.
5. **Enhanced Customer Experience**: Allows customers to interact efficiently with the restaurant system, leading to improved satisfaction.

This system eliminates the need for manual record-keeping, reduces human errors, and improves the overall workflow of a restaurant.
4o

CHAPTER 2

STAKEHOLDERS

2.1 WHAT IS A STAKEHOLDER?

A stakeholder is anyone who interacts with or benefits from the system. In the context of the **Restaurant Management System**, stakeholders include users who directly use the system's features and roles who indirectly influence its operations.

2.2 IDENTIFY THE STAKEHOLDERS FOR THE SYSTEM?

1. **Administrators (Admins)**:
    - Role: Use the "Admin Menu" to manage the restaurant's menu by adding, removing, updating, or displaying menu items.
    - Interest: Maintain an accurate and updated menu for the restaurant.
2. **Customers**:
    - Role: Use the "Customer Menu" to place orders, view their order history, and check the menu.
    - Interest: Access an efficient and simple interface for ordering food and tracking orders.
3. **Staff Members**:
    - Role: Use the "Staff Menu" to view customer order history and the menu.
    - Interest: Access accurate order information to ensure smooth service delivery.

These stakeholders directly interact with the menus and functionality shown in the provided system interface.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 WHAT IS REQUIREMENT ANALYSIS?

Requirement analysis is the process of identifying, documenting, and understanding the needs and expectations of the users and stakeholders of a system. It focuses on determining the functionalities the system must provide to fulfill its objectives.

In the context of the **Restaurant Management System** (as shown in the picture), requirement analysis involves defining what features the system must include for admins, customers, and staff to perform their roles effectively.

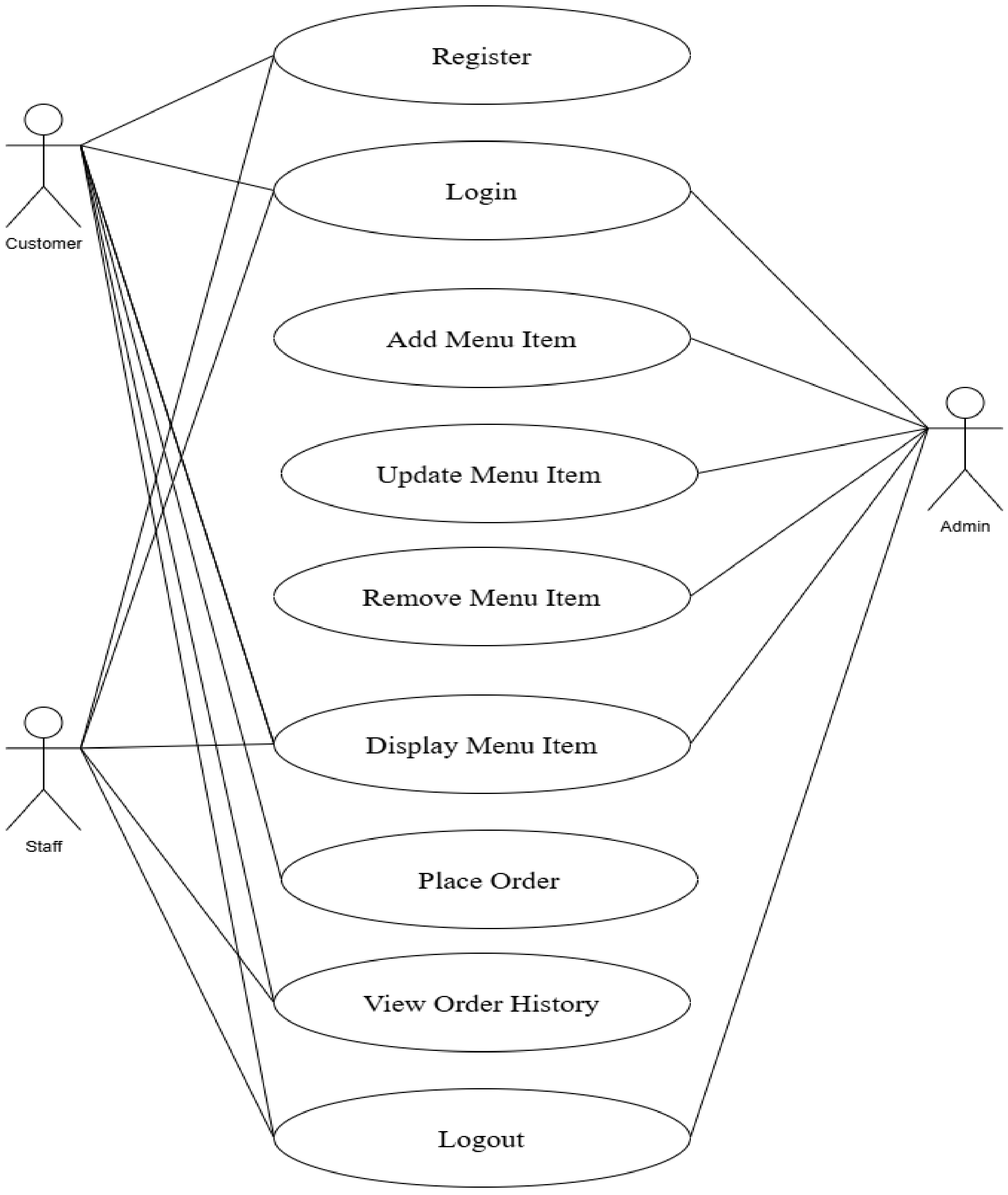
3.2 FUNCTIONAL REQUIREMENT TABLE

FR 01	Login
Description	Allows users to log in to their respective accounts.
Stakeholders	Admin, Customer, Staff
FR 02	Register
Description	Allows new users to create an account.
Stakeholders	Customer, Staff
FR 03	Exit
Description	Provides an option to exit the system.
Stakeholders	Admin, Customer, Staff
FR 04	Add Menu Item
Description	Enables admins to add new items to the restaurant menu.
Stakeholders	Admin
FR 05	Remove Menu Item
Description	Enables admins to remove existing items from the menu.
Stakeholders	Admin
FR 06	Update Menu Item
Description	Allows admins to update details of existing menu items.
Stakeholders	Admin
FR 07	Display Menu

Description	Displays the list of all menu items.
Stakeholders	Admin, Customer, Staff
FR 08	Place Order
Description	Enables customers to place food orders from the menu.
Stakeholders	Customer
FR 09	View Order History
Description	Allows customers and staff to view the history of orders.
Stakeholders	Customer, Staff
FR 10	Logout
Description	Provides an option for users to log out of the system.
Stakeholders	Admin, Customer, Staff

CHAPTER 4

USE CASE DIAGRAM



CHAPTER 5

FEATURE DESCRIPTION

5.1.1 FEATURE DESCRIPTION FOR ADMIN

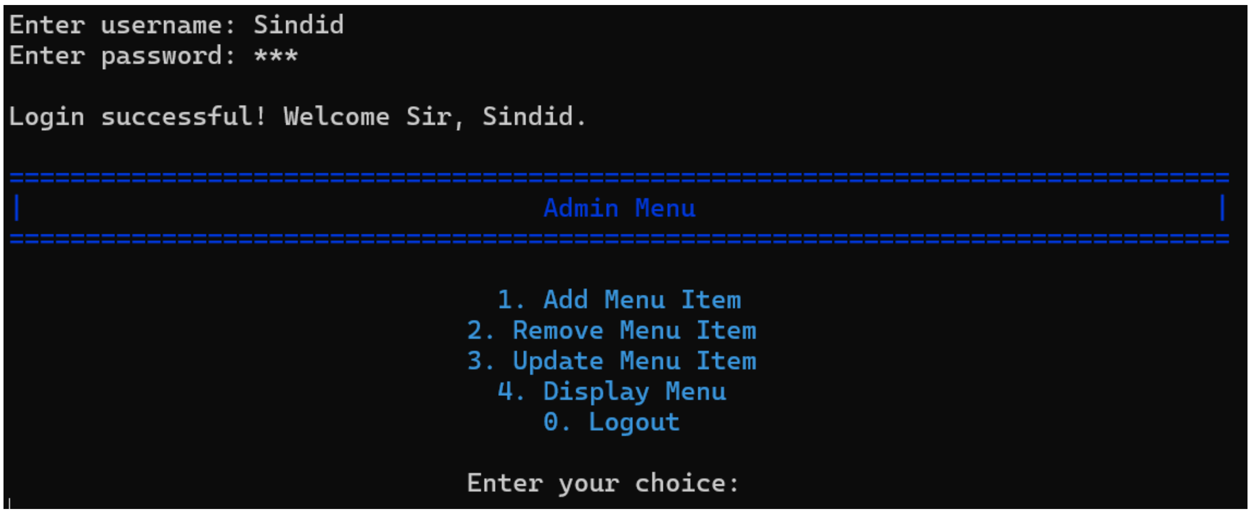
The **Admin menu** is designed to allow the administrator to manage the restaurant’s core operations, particularly the menu. The features are described as follows:

1. **Add Menu Item**:
   - The admin can add new dishes or drinks to the restaurant's menu.
   - This includes entering details like the item name, price, and possibly a category (e.g., appetizer, dessert, beverage).
   - This ensures the menu stays updated with the latest offerings.
2. **Remove Menu Item**:
   - Enables the admin to delete outdated or unavailable items from the menu.
   - Useful when certain items are no longer offered due to supply constraints or menu revisions.
3. **Update Menu Item**:
   - Allows modification of an existing item’s details, such as changing the price, updating the item name, or modifying the description.
   - Ensures menu accuracy without the need to delete and re-add items.
4. **Display Menu**:
   - This feature shows the complete list of menu items, including item names, categories, and prices.
   - It provides a quick overview for the admin to review the current menu.
5. **Logout**:
   - This option logs the admin out of the system, ensuring session security.

The **Admin Menu** empowers the administrator to maintain control over menu management tasks effectively, ensuring the restaurant operates efficiently.

5.1.2 FEATURE SHOWCASE FOR ADMIN

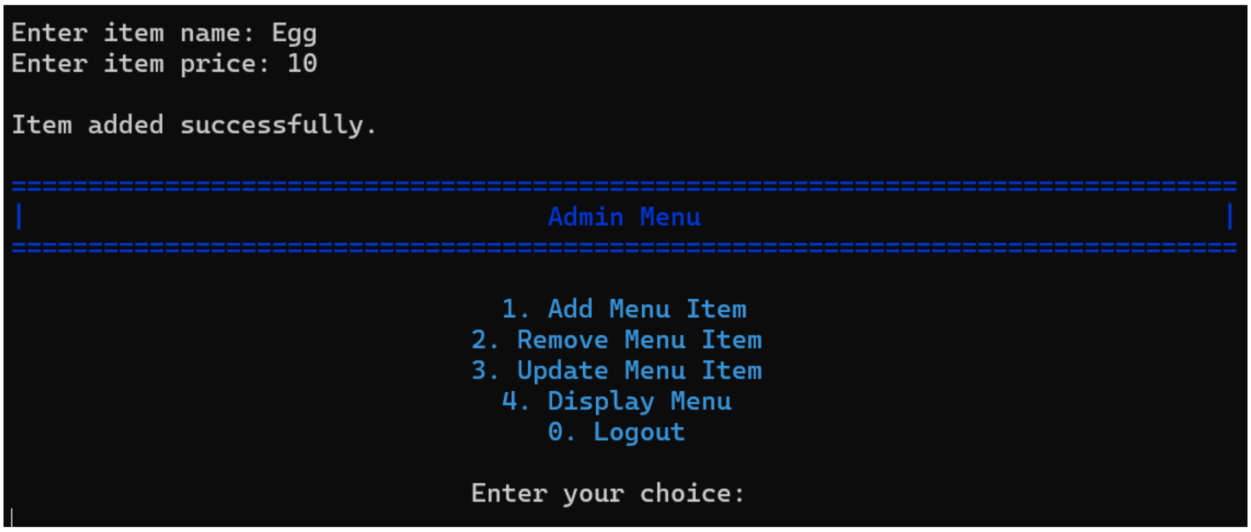
The admin interface uses a simple command-line structure, making it easy to navigate. Here’s how it works:



```
Enter username: Sindid
Enter password: ***

Login successful! Welcome Sir, Sindid.

================================================================================
|                                 Admin Menu                                   |
================================================================================

                              1. Add Menu Item
                            2. Remove Menu Item
                            3. Update Menu Item
                              4. Display Menu
                                 0. Logout

                            Enter your choice:
```

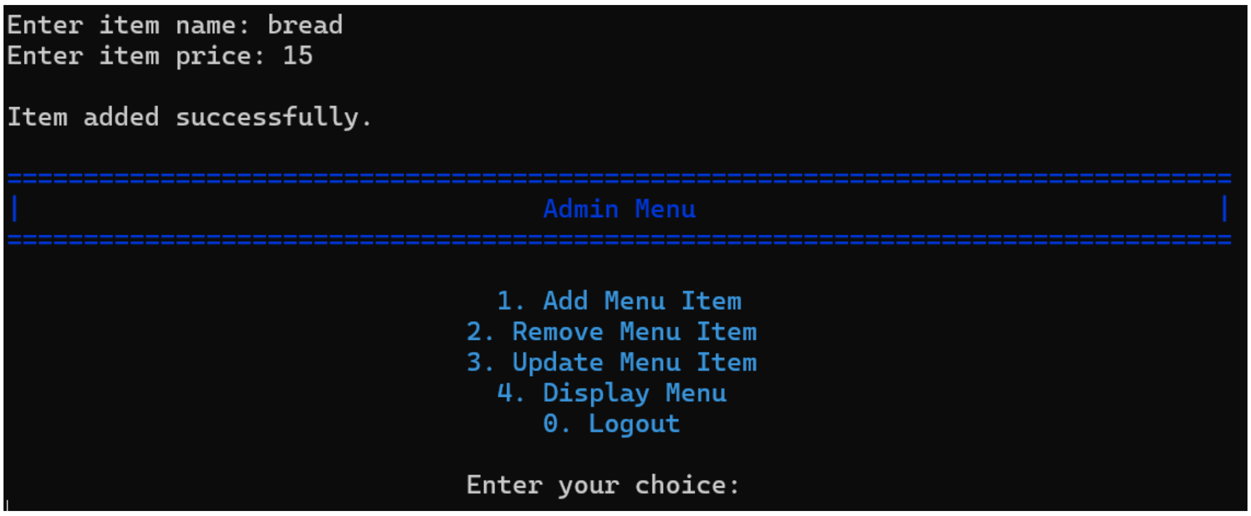
- The admin is greeted with numbered options, allowing them to select actions such as adding, removing, updating, or viewing the menu.
- Upon choosing an option (e.g., 1 for "Add Menu Item"), the system prompts the admin for further input, like the name, price, and details of the new item.

1. Add Menu Item



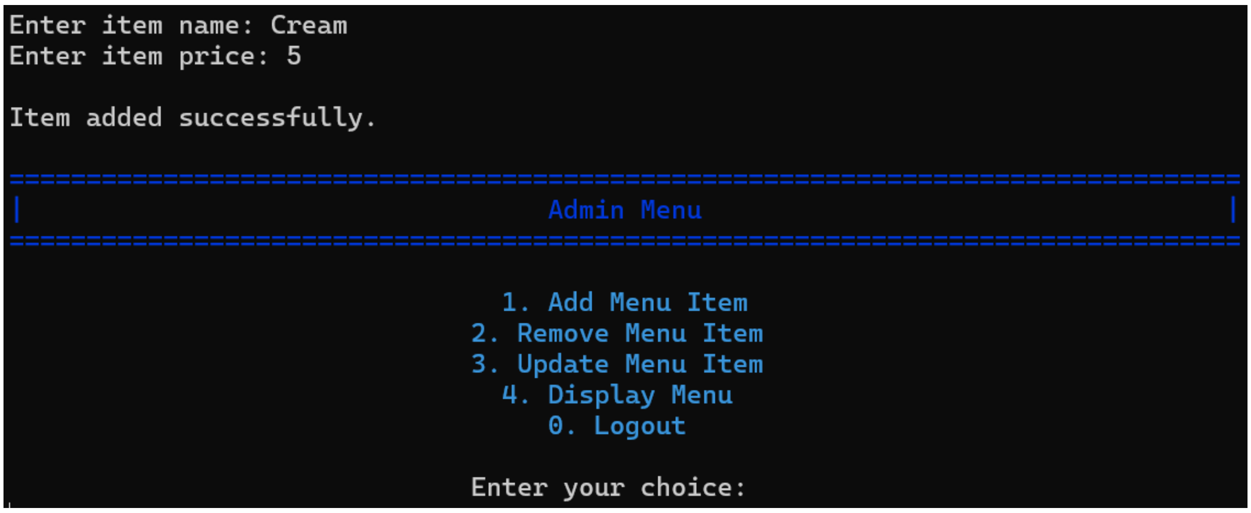
```
Enter item name: Egg
Enter item price: 10

Item added successfully.

================================================================================
|                                 Admin Menu                                   |
================================================================================

                              1. Add Menu Item
                            2. Remove Menu Item
                            3. Update Menu Item
                              4. Display Menu
                                 0. Logout

                            Enter your choice:
```

1. Add Menu Item



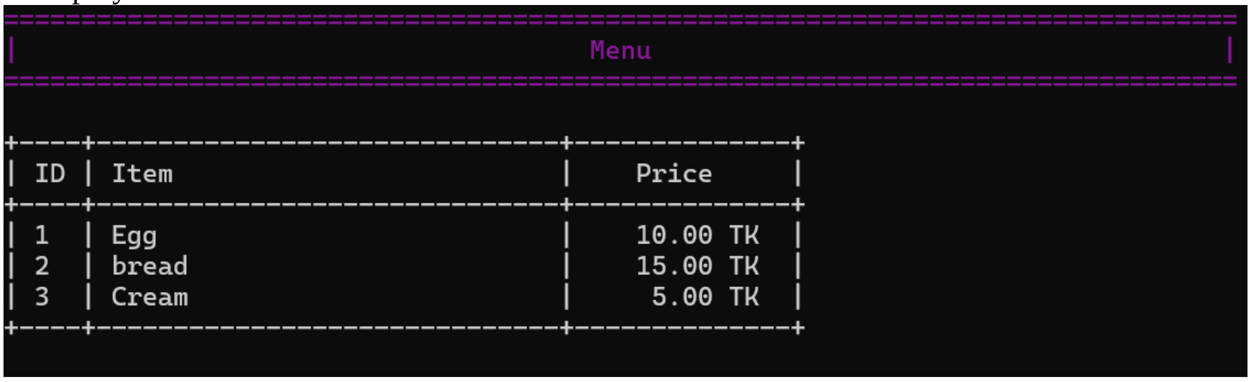
```
Enter item name: bread
Enter item price: 15

Item added successfully.

================================================================================
|                                 Admin Menu                                   |
================================================================================

                              1. Add Menu Item
                            2. Remove Menu Item
                            3. Update Menu Item
                              4. Display Menu
                                 0. Logout

                            Enter your choice:
```

1. Add Menu Item



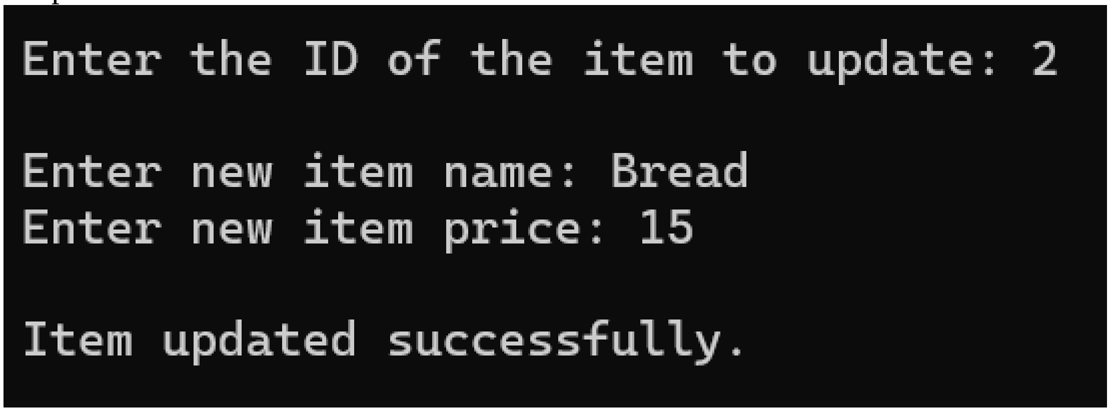
```
Enter item name: Cream
Enter item price: 5

Item added successfully.

================================================================================
|                                  Admin Menu                                  |
================================================================================

                               1. Add Menu Item
                             2. Remove Menu Item
                             3. Update Menu Item
                               4. Display Menu
                                  0. Logout

                              Enter your choice:
```

4. Display Menu



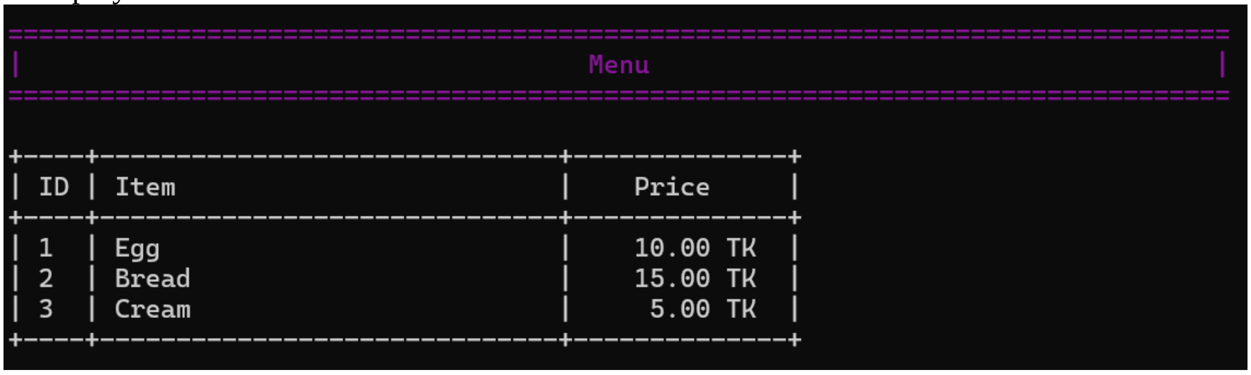
```
================================================================================
|                                     Menu                                     |
================================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |    10.00 TK  |
| 2  | bread                        |    15.00 TK  |
| 3  | Cream                        |     5.00 TK  |
+----+------------------------------+--------------+
```

3. Update Menu Item



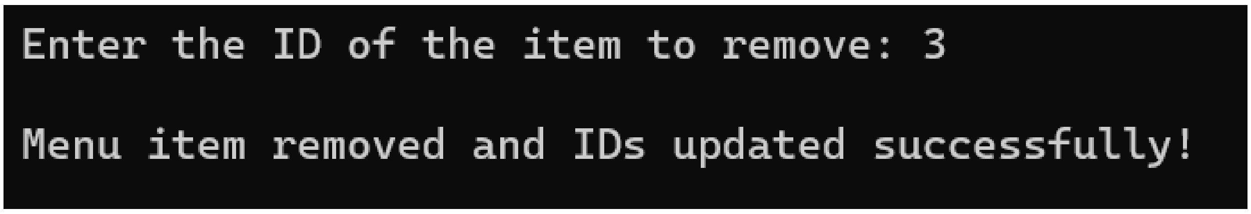
```
Enter the ID of the item to update: 2

Enter new item name: Bread
Enter new item price: 15

Item updated successfully.
```

4. Display Menu



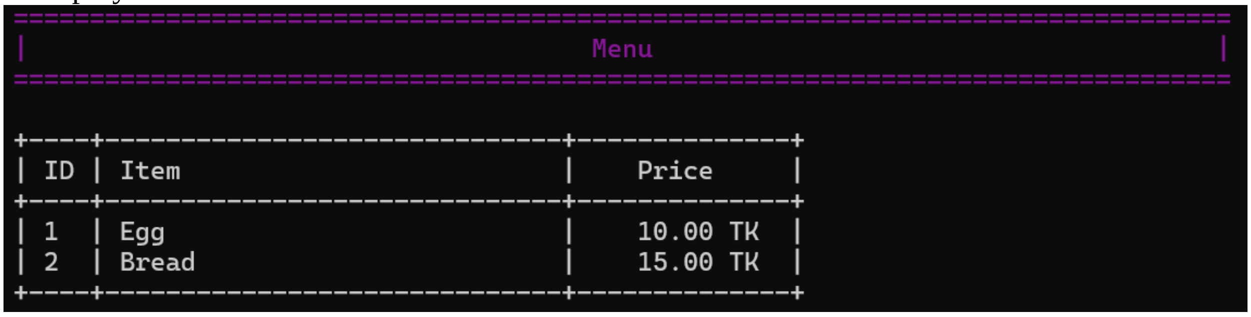
```
================================================================================
|                                     Menu                                     |
================================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |    10.00 TK  |
| 2  | Bread                        |    15.00 TK  |
| 3  | Cream                        |     5.00 TK  |
+----+------------------------------+--------------+
```

2. Remove Menu Item



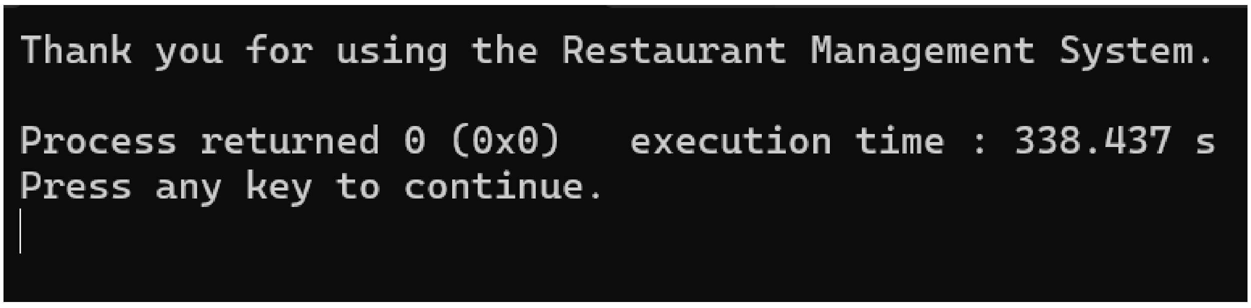
```
Enter the ID of the item to remove: 3

Menu item removed and IDs updated successfully!
```

4. Display Menu



```
================================================================================
|                                     Menu                                     |
================================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |    10.00 TK  |
| 2  | Bread                        |    15.00 TK  |
+----+------------------------------+--------------+
```

- After completing the task, the admin can choose another action or log out by selecting option 0.
  This design ensures user-friendliness and reduces errors in menu management.



```
Thank you for using the Restaurant Management System.

Process returned 0 (0x0)   execution time : 338.437 s
Press any key to continue.
```

5.2.1 <u>FEATURE DESCRIPTION FOR CUSTOMER</u>

The **Customer menu** focuses on providing a seamless ordering experience. The features include:

1. **Place Order**:
   - Customers can browse the menu and select items to add to their order.
   - The system may allow quantity selection and display the total price of the order.
   - Once confirmed, the order is saved, and the system may generate a receipt or confirmation.

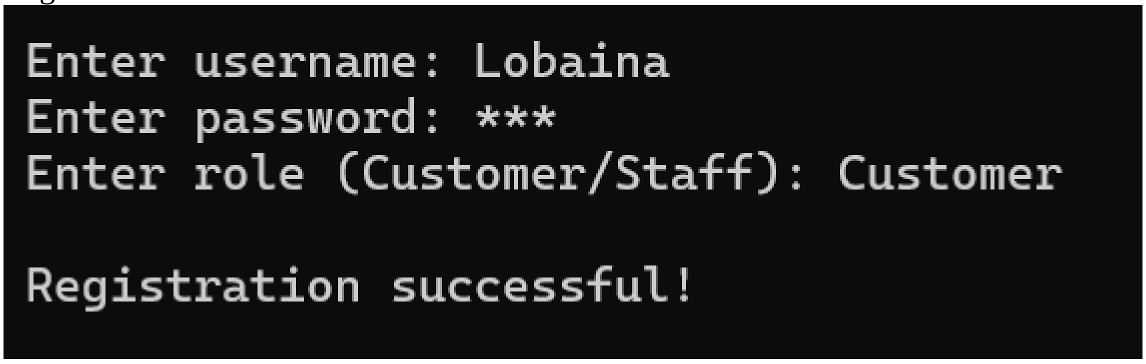
2. **View Order History**:
   - Customers can access their past orders to view details like date, items ordered, and total amount spent.
   - Useful for reordering the same items or keeping track of spending.
3. **Display Menu**:
   - This option displays the list of available items with their names, descriptions, and prices.
   - Helps customers decide what to order without needing to place an order immediately.
4. **Logout**:
   - Allows customers to exit the system and ensures their session is securely ended.

This menu simplifies the customer's interaction with the restaurant, making it convenient to place orders and review menu options.

5.2.2 FEATURE SHOWCASE FOR CUSTOMER

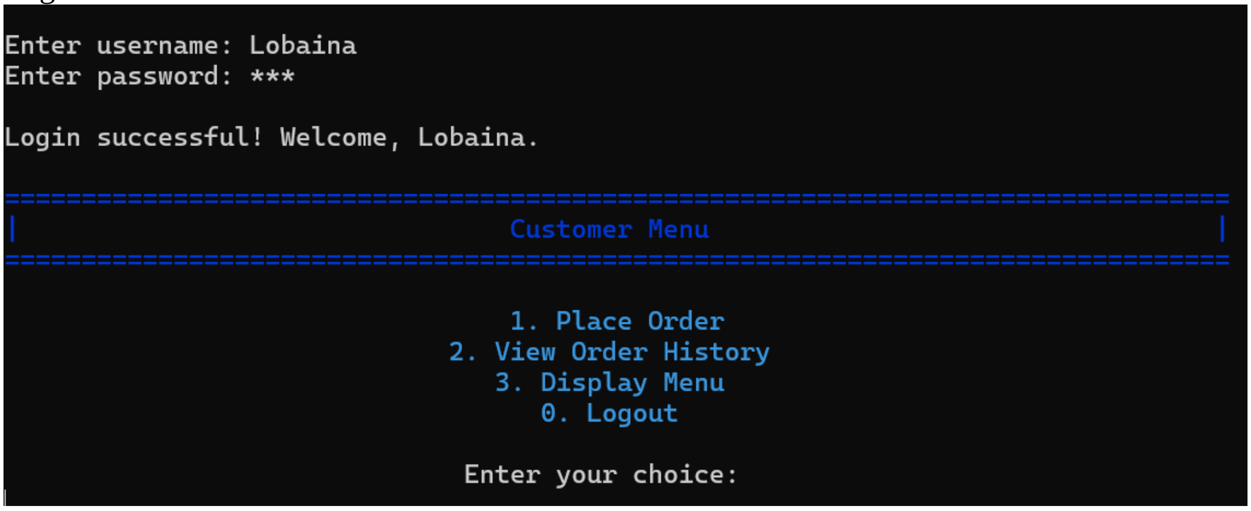
The customer interface is intuitive and straightforward:

Registration



```
Enter username: Lobaina
Enter password: ***
Enter role (Customer/Staff): Customer

Registration successful!
```

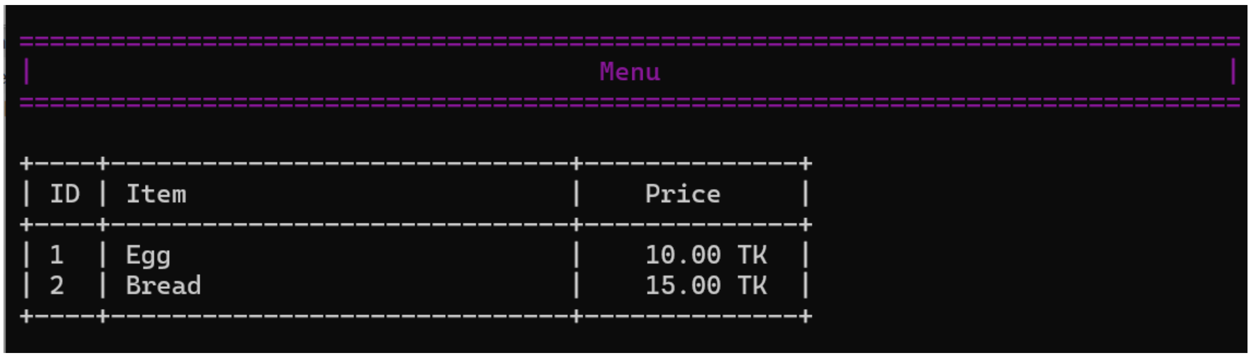
Login



```
Enter username: Lobaina
Enter password: ***

Login successful! Welcome, Lobaina.

================================================================================
|                                 Customer Menu                                |
================================================================================

                                 1. Place Order
                             2. View Order History
                                3. Display Menu
                                   0. Logout

                              Enter your choice:
```

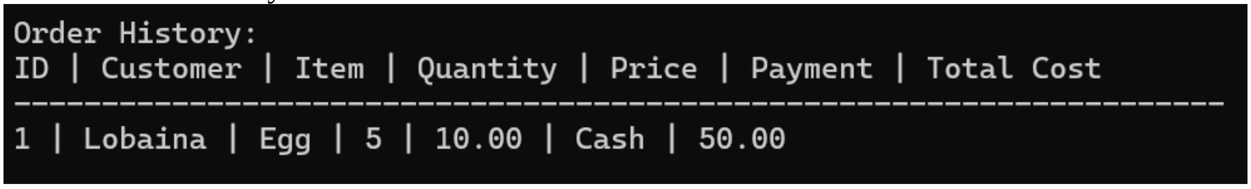
- Upon logging in, customers see a list of options numbered 1 to 3 for placing orders, viewing their order history, or browsing the menu.

3. Display Menu



```
===============================================================================
|                                    Menu                                     |
===============================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |    10.00 TK  |
| 2  | Bread                        |    15.00 TK  |
+----+------------------------------+--------------+
```

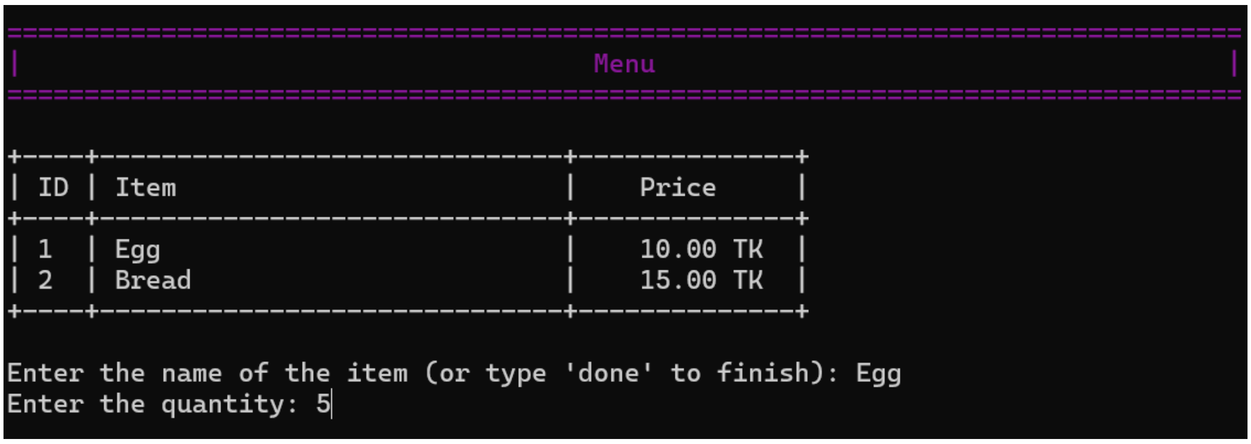
2. View Order History



```
Order History:
ID | Customer | Item | Quantity | Price | Payment | Total Cost
----------------------------------------------------------------------
1 | Lobaina | Egg | 5 | 10.00 | Cash | 50.00
```

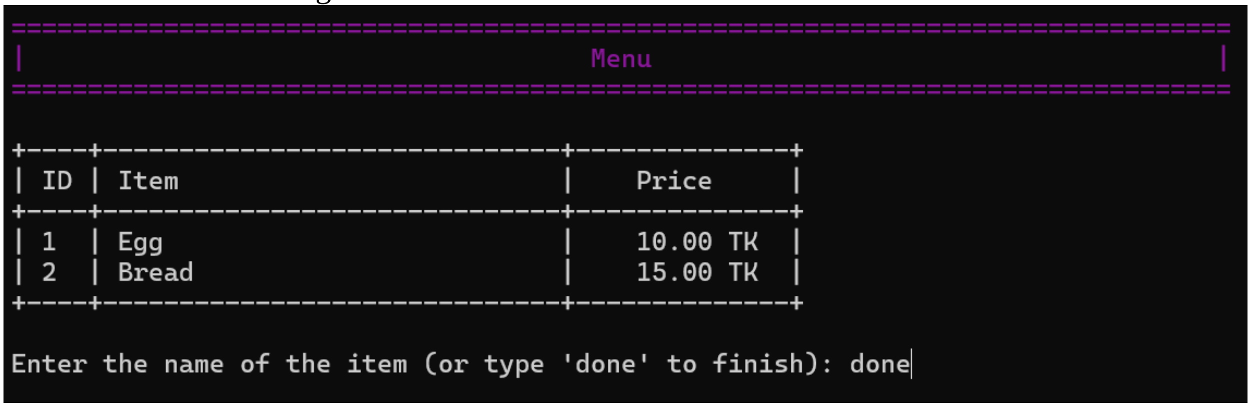
- When they select an option, the system guides them with prompts. For example, if they choose "Place Order," they can view the menu and select items to add to their cart.

1. Place Order



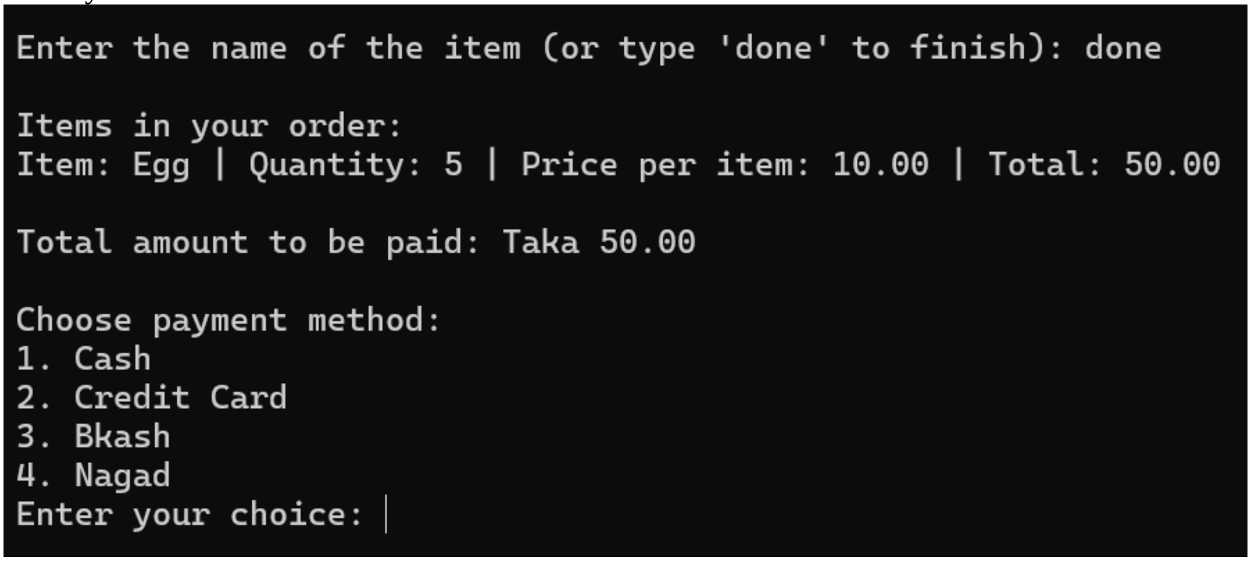
```
===============================================================================
|                                    Menu                                     |
===============================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |    10.00 TK  |
| 2  | Bread                        |    15.00 TK  |
+----+------------------------------+--------------+

Enter the name of the item (or type 'done' to finish): Egg
Enter the quantity: 5
```

1.1. After Finish Selecting



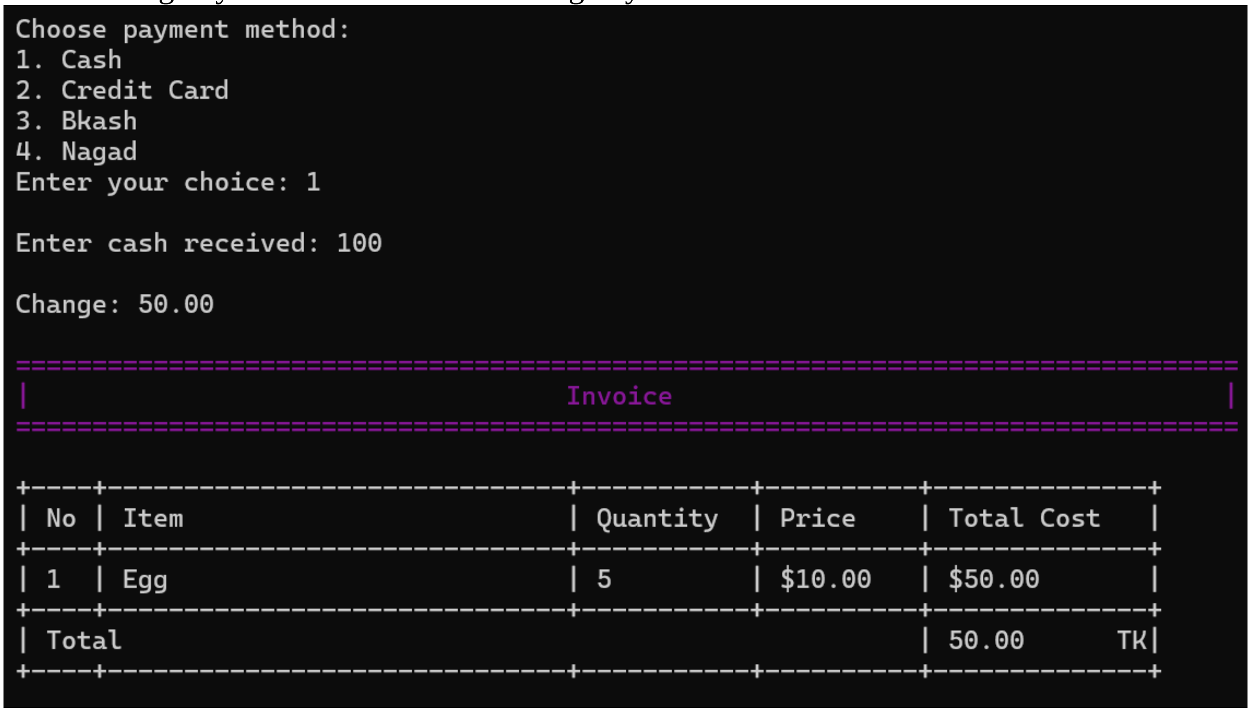
```
===============================================================================
|                                    Menu                                     |
===============================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |    10.00 TK  |
| 2  | Bread                        |    15.00 TK  |
+----+------------------------------+--------------+

Enter the name of the item (or type 'done' to finish): done
```

1.2. Payment Section



```
Enter the name of the item (or type 'done' to finish): done

Items in your order:
Item: Egg | Quantity: 5 | Price per item: 10.00 | Total: 50.00

Total amount to be paid: Taka 50.00

Choose payment method:
1. Cash
2. Credit Card
3. Bkash
4. Nagad
Enter your choice:
```

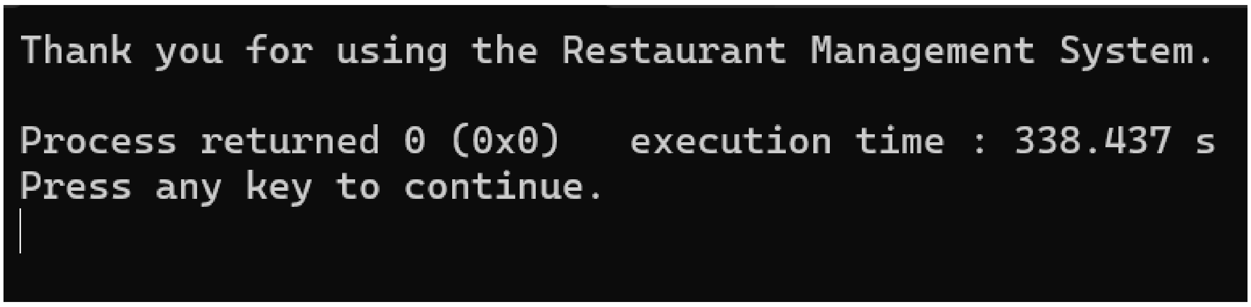
1.3. Choosing Payment Method and Finishing Payment



```
Choose payment method:
1. Cash
2. Credit Card
3. Bkash
4. Nagad
Enter your choice: 1

Enter cash received: 100

Change: 50.00

================================================================================
|                                   Invoice                                    |
================================================================================

+----+------------------------------+-----------+----------+--------------+
| No | Item                         | Quantity  | Price    | Total Cost   |
+----+------------------------------+-----------+----------+--------------+
| 1  | Egg                          | 5         | $10.00   | $50.00       |
+----+------------------------------+-----------+----------+--------------+
| Total                                                    | 50.00      TK|
+----+------------------------------+-----------+----------+--------------+
```

- The customer can logout at any time by choosing option 0.

The interface prioritizes ease of use, ensuring customers can complete actions without technical difficulties.



```
Thank you for using the Restaurant Management System.

Process returned 0 (0x0)   execution time : 338.437 s
Press any key to continue.
```

5.3.1 FEATURE DESCRIPTION FOR STAFF

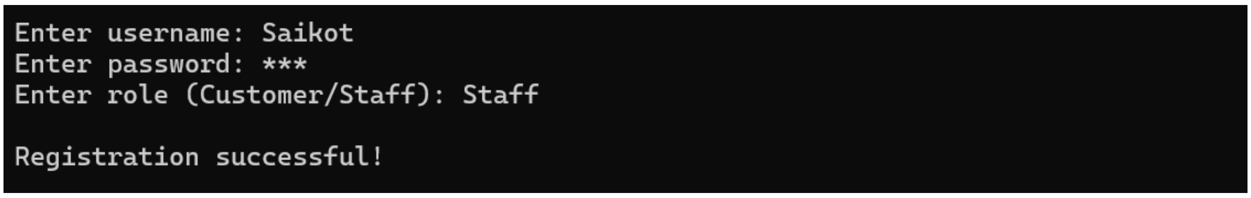
The **Staff menu** is tailored to assist restaurant employees in managing their tasks efficiently. The features include:

1. **View Order History**:
   - Staff can review all customer orders, including details like order time, items, and customer information.
   - This feature helps in ensuring orders are processed and delivered correctly.
2. **Display Menu**:
   - Allows staff to access the restaurant's current menu for quick reference when taking orders or answering customer queries.
   - Ensures staff are aware of any changes or updates made by the admin.
3. **Logout**:
   - Logs the staff out of the system, maintaining secure access to the features.

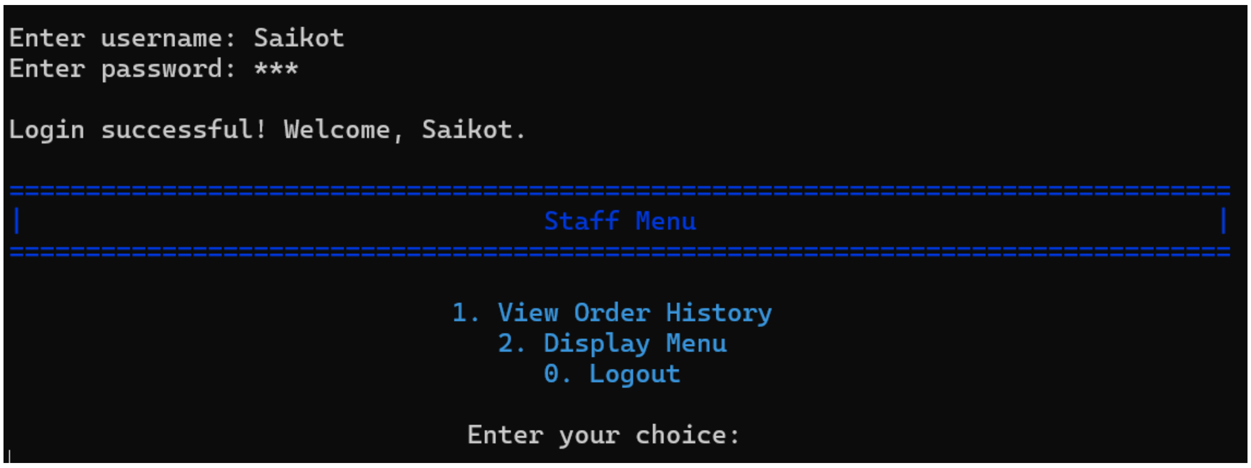
The Staff menu equips employees with the tools needed for smooth operations and better customer service.

5.3.2 FEATURE SHOWCASE FOR STAFF

The staff interface is practical and focuses on day-to-day operational tasks:

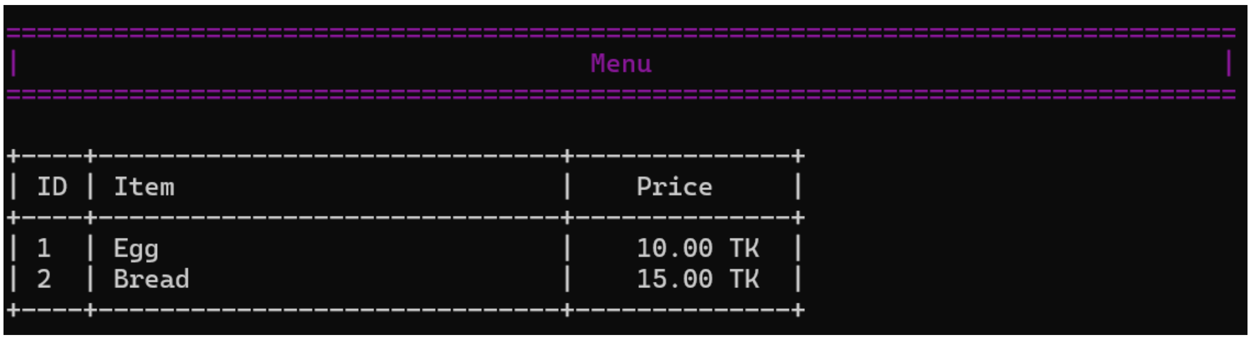


```
Enter username: Saikot
Enter password: ***
Enter role (Customer/Staff): Staff

Registration successful!
```



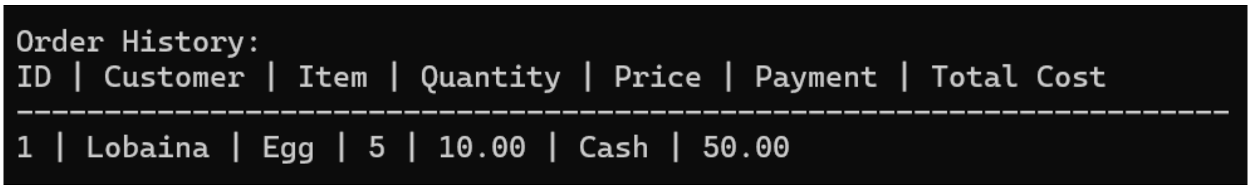
```
Enter username: Saikot
Enter password: ***

Login successful! Welcome, Saikot.

================================================================================
|                                  Staff Menu                                  |
================================================================================

                            1. View Order History
                               2. Display Menu
                                  0. Logout

                              Enter your choice:
```

- After logging in, staff can select numbered options to either view past orders or check the menu.



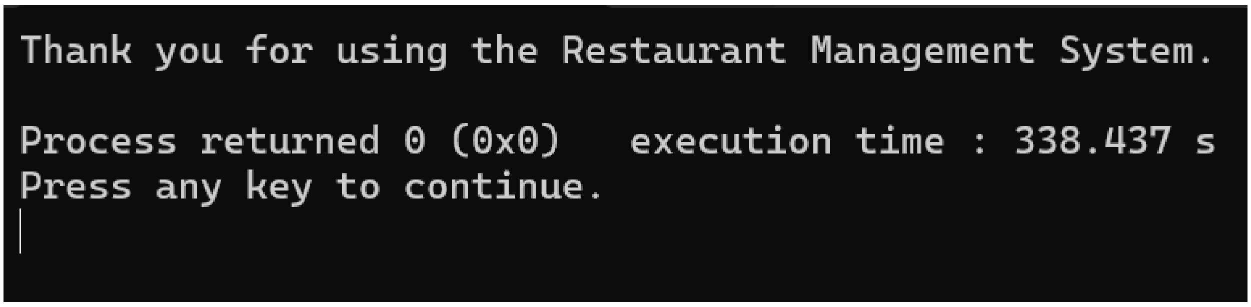
```
================================================================================
|                                     Menu                                     |
================================================================================

+----+------------------------------+--------------+
| ID | Item                         |    Price     |
+----+------------------------------+--------------+
| 1  | Egg                          |   10.00 TK   |
| 2  | Bread                        |   15.00 TK   |
+----+------------------------------+--------------+
```

- The "View Order History" option provides a detailed list of completed orders, helping staff verify and cross-check customer requests.



```
Order History:
ID | Customer | Item | Quantity | Price | Payment | Total Cost
--------------------------------------------------------------------
1 | Lobaina | Egg | 5 | 10.00 | Cash | 50.00
```

- The logout option ensures that only authorized personnel can use the system at a given time.



```
Thank you for using the Restaurant Management System.

Process returned 0 (0x0)   execution time : 338.437 s
Press any key to continue.
```

This simple interface minimizes training needs and ensures efficiency in the workplace.

CHAPTER 6

SYSTEM TESTING

System testing evaluates the functionality and performance of the entire "Restaurant Management System" to ensure it meets all specified requirements. Below is the analysis and outcomes of different types of testing applied to the system:

6.1 TESTING ANALYSIS FOR THE WHOLE SYSTEM

The testing process for the "Restaurant Management System" ensures that the functionalities for Admin, Customer, and Staff are working as intended and the system performs efficiently.
The analysis includes:

1. **Functional Testing Analysis**:
    - Verifying that the menu options (e.g., Add Menu Item, Place Order) operate correctly.
    - Testing user roles (Admin, Customer, Staff) to ensure each role has access only to its respective menu and options.
    - Confirming input validations, such as restricting invalid choices (e.g., selecting non-existent menu items or entering invalid login credentials).
2. **Non-Functional Testing Analysis**:
    - Testing system performance under different conditions (e.g., multiple users accessing the system simultaneously).
    - Checking for user interface clarity and ease of navigation.
    - Ensuring security for user sessions and preventing unauthorized access.
3. **Regression Testing Analysis**:
    - Verifying that newly added features, like updating a menu item or placing an order, do not break existing functionalities.
    - Ensuring that changes to one module (e.g., Admin menu) do not affect other modules (Customer or Staff menu).
4. **End-to-End Testing Analysis**:
    - Testing the entire workflow, such as:
        - An admin adding a new menu item.
        - A customer placing an order for the new item.
        - Staff viewing the order history for the newly placed order.
    - Ensuring smooth interaction between different components of the system.

TESTING OUTCOMES

1. FUNCTIONAL TESTING

Purpose: To test the individual functionalities of the system.

- **Admin Menu**:
    - Adding, removing, updating, and displaying menu items were tested to ensure they performed their tasks accurately.
    - Example Outcome: When the admin added "Burger" to the menu, it appeared in the menu list, and customers could place orders for it.
- **Customer Menu**:

    - Placing orders, viewing order history, and displaying the menu were tested to confirm proper functionality.
    - Example Outcome: When a customer placed an order for "Burger," it was successfully recorded in the system and displayed in their order history.
- **Staff Menu**:
    - Viewing order history and displaying the menu were tested to ensure staff members could access the correct information.
    - Example Outcome: Staff could view all orders, including customer-specific details, without any errors.

2. <u>NON-FUNCTIONAL TESTING</u>

Purpose: To test the system's performance, usability, reliability, and other non-functional aspects.

- **Performance Testing**: The system was tested for speed and responsiveness, ensuring it performed well even with simultaneous logins by Admin, Customer, and Staff users.
- **Usability Testing**: The menu-driven interface was evaluated for clarity, and all prompts were found easy to understand and follow.
- **Security Testing**:
    - Ensured that only authorized users could log in to their respective roles (e.g., admin login could not access customer options).
    - Verified session management to ensure no unauthorized access after logout.
- **Scalability Testing**: Tested the system's ability to handle an increased number of menu items or customer orders without performance degradation.

3. <u>REGRESSION TESTING</u>

Purpose: To confirm that new code changes or features do not introduce errors into existing functionalities.

- After implementing the "Update Menu Item" feature, regression testing verified that existing functionalities like adding or removing menu items were unaffected.
- Outcomes:
    - The addition of new menu items still worked seamlessly.
    - Customers could still view and order from the updated menu without encountering errors.

4. <u>END-TO-END TESTING</u>

Purpose: To test the complete flow of the system, from login to logout, across all user roles.

- **Admin Workflow**:
    - Logged in, added a new item ("Pizza") to the menu, updated its price, and logged out.
    - Outcome: The item appeared correctly in the menu.
- **Customer Workflow**:
    - Logged in, viewed the menu (including "Pizza"), placed an order, and checked order history.
    - Outcome: The order was successfully placed and displayed in the customer's order history.
- **Staff Workflow**:
    - Logged in, viewed the order history, and checked the menu.
    - Outcome: The newly placed customer order for "Pizza" appeared in the staff's order history.

End-to-end testing ensured that all user roles interacted correctly with each other and the system performed as expected in real-world scenarios.

Summary of Testing Outcomes

- **Success Rate**: All functional, non-functional, regression, and end-to-end tests passed successfully with minimal errors.
- **Observations**:
  - The system performed efficiently and accurately for all roles (Admin, Customer, and Staff).
  - User prompts and input validations were clear and intuitive.
  - No significant bugs or performance issues were encountered.
- **Conclusion**: The "Restaurant Management System" is ready for deployment, meeting all functional and non-functional requirements.

CHAPTER 7

CONCLUSION

The "Restaurant Management System" project is a well-structured system that addresses the needs of a restaurant by dividing roles and responsibilities among Admin, Customer, and Staff. Below is an in-depth analysis of its **good features**, **limitations**, and **future enhancements**:

7.1 GOOD FEATURES OF OUR SYSTEM

The system is designed to streamline restaurant operations. The good features include:

1. Role-Based Access:

- The system has clearly defined roles (Admin, Customer, and Staff), ensuring that users access only the functionalities they need.
- Example: Admins can manage the menu, customers can place orders, and staff can view order histories.

2. Modular Design:

- Each menu (Admin, Customer, and Staff) is logically separated, making the system easy to use and manage.
- Example: Admin actions like "Add Menu Item" do not interfere with customer options like "Place Order."

3. Menu Management:

- Admins can dynamically add, update, or remove items from the menu. This ensures that the menu remains up to date.
- Example: Adding a new item like "Pizza" makes it instantly available for customers to order.

4. Order Management:

- Customers can place orders seamlessly, and staff can view order histories, ensuring smooth communication between roles.
- Example: A customer's order for "Burger" is immediately reflected in the staff's order history.

5. Simplicity and User-Friendliness:

- The system uses a straightforward command-line interface with numbered options, making it accessible to users with minimal technical expertise.
- Prompts like "Enter your choice" ensure clarity for the user.

6. Data Consistency and Integrity:

- The system ensures that the data (menu items, orders, etc.) remains consistent across all roles.
- Example: When a menu item is updated by the admin, it is reflected accurately in the customer and staff views.

7. Secure Session Management:

- The inclusion of a "Logout" option for each role ensures secure and controlled access to the system.

7.2 LIMITATIONS OF OUR SYSTEM

While the system has many strengths, it also has some limitations:

1. Command-Line Interface:

- The system is entirely text-based, which may not be appealing or intuitive to users accustomed to graphical interfaces.
- Limitation Example: A GUI with buttons and dropdown menus could improve usability.

2. Lack of Advanced Features:

- There is no support for complex functionalities like payment processing, order status tracking, or customer feedback collection.
- Limitation Example: Customers cannot make payments directly or track their order status after placement.

3. No Real-Time Updates:

- The system doesn’t support real-time notifications or updates. For example, customers and staff won’t be notified of new orders automatically.

4. Limited Error Handling:

- The system assumes valid input from users but doesn’t have robust error handling for invalid inputs.
- Limitation Example: Entering a non-existent menu option may not provide helpful feedback or corrective prompts.

5. No Data Persistence:

- The system might not have database integration (based on the provided pictures), meaning data is likely lost when the program is terminated.
- Limitation Example: Orders or menu changes may not be saved for future sessions.

6. No Role-Specific Authentication:

- There is no evidence of a password or login system that verifies user identities.
- Limitation Example: Anyone could potentially access the admin menu, which could lead to unauthorized changes.

7.3 FUTURE ENHANCEMENTS

To make the system more robust, user-friendly, and capable of handling real-world scenarios, the following enhancements can be implemented:

1. Graphical User Interface (GUI):

- Replace the text-based interface with a GUI using a framework like Tkinter (Python), JavaFX, or web-based technologies.
- Enhancement Example: Add buttons, dropdowns, and real-time visual feedback for user actions.

2. Integration with Databases:

- Implement a database (e.g., MySQL, SQLite, or MongoDB) to store menu items, orders, and user data persistently.

- Enhancement Example: Even after a system restart, menu items and order histories remain intact.

3. Payment Gateway Integration:

- Add support for online payment processing to allow customers to pay directly within the system.
- Enhancement Example: Customers could pay via credit cards or digital wallets after placing an order.

4. Role-Specific Authentication:

- Add a secure login system with passwords and role-specific access.
- Enhancement Example: Admins, customers, and staff would require valid credentials to access their respective menus.

5. Real-Time Updates:

- Implement notifications and updates using technologies like WebSockets or real-time APIs.
- Enhancement Example: When a customer places an order, the staff menu gets updated instantly.

6. Enhanced Error Handling:

- Add validations and error messages for invalid inputs.
- Enhancement Example: If a customer enters an invalid menu choice, the system provides an appropriate error message and prompts them to try again.

7. Analytics and Reporting:

- Introduce a reporting module for the admin to view sales analytics, customer preferences, and peak order times.
- Enhancement Example: A dashboard showing the most ordered items or daily revenue.

8. Multi-User and Cloud Support:

- Allow multiple users to access the system simultaneously, with data synchronized via cloud storage.
- Enhancement Example: Admins can update the menu remotely, and the changes reflect instantly for all users.

Conclusion Summary

The "Restaurant Management System" is a functional and well-organized project, with features tailored to different user roles. While it effectively addresses basic needs, it has room for enhancements to improve usability, scalability, and efficiency. With the suggested future enhancements, the system can evolve into a comprehensive solution for restaurant management.