

Restaurant Management System



Session: 2023 – 2027

Submitted by:

Mohammad Bilal

2023-CS-168

Supervised by:

Sir Laeeq uz Zaman

Khan Niazi

Department of Computer Science
University of Engineering and Technology
Lahore, Pakistan

Table of Contents

1. Project Description:	4
1.1 Purpose:	4
1.2 Scope:	4
1.3 Objectives:	4
1.4 Major Features:	4
2. Technical Details:	4
2.1 Development Environment:	4
2.2 Programming Languages:	5
2.3 Framework:	5
2.4 Database:	5
3. User Guide:	5
3.1 Getting Started:	5
3.2 User Roles and Permissions:	5
3.3 Functionality Overview:	5
3.3.1 Admin:	5
3.3.2 Customer:	6
3.3.2 Staff (Waiters and Chefs):	6
4. Wireframes:	6
4.1 Homepage:	6
4.2 Sign Up Page:	7
4.3 Sign-In Page:	7
4.4 Customer:	8
4.4.1 Landing Page:	8
4.4.2 Order Food:	8
4.4.3 View Cart:	9
4.4.4 Book Table:	9
4.4.5 Feedback:	10
4.4.6 Help:	10
4.4.7 Settings:	11
4.5 Admin:	11
4.5.1 Manage Employees:	11

4.5.2 Manage Customers:	12
4.5.3 Send Notifications:	12
4.5.4 Manage Tables:.....	13
4.5.5 View Feedback:	13
4.5.6 Reply to Customer Messages:	14
4.5.7 Admin Settings:	14
4.6 Chef:	15
4.6.1 Manage Products:	15
4.6.2 Manage Orders:	15
4.7 Waiter:	16
4.7.1: Take Order:.....	16
4.7.2 Pickup and Deliver Order:	16
4.7.3 Manage Reservations:.....	17
5. CRC Diagram:	17
6. Code:	18
7. GitHub Link:	21

1. Project Description:

1.1 Purpose:

The purpose of the Restaurant Management System is to streamline the operations of the any booming restaurant that is in need to expand its operations. It aims to provide an efficient and user-friendly interface for placing orders, managing orders, inventory, and other related activities.

1.2 Scope:

The system is designed to cater specifically to the needs of the restaurants focusing on order processing, inventory management, and overall restaurant administration.

1.3 Objectives:

- Automate order placement and processing.
- Manage inventory efficiently.
- Provide a user-friendly graphical interface for both customers and staff.

1.4 Major Features:

- User-friendly GUI for order placement and processing.
- Inventory management for tracking product availability.
- Role-based access control for secure operations.
- Real-time order tracking.
- Reservation management.
- Forgot password functionality.
- Messaging and notification system.
- Allow customers to upload profile
- Interactive and Responsive UI.
- Incentives for customers to attract them to this app.
- Generating reports for sales, order history, users and many more.
- Customer's cart info is saved even after closing app.
- Implemented complete functionality with both DB and FH.
- Complete implementation of OOP Features.

2. Technical Details:

2.1 Development Environment:

- Integrated Development Environment (IDE): Visual Studio
- Version Control: Git

2.2 Programming Languages:

Programming languages used are C# and SQL.

2.3 Framework:

.NET Framework

2.4 Database:

SQL Server for data storage and retrieval.

3. User Guide:

3.1 Getting Started:

On launching the application, the customers lands on homepage and can navigate to Sign in or Sign-up page, after successfully signing up, he can sign in using valid credentials. From here on out the user will explore the options and features available.

3.2 User Roles and Permissions:

Administrator:

Full access to all the management features.

Staff:

Limited access for order processing and products management. They include *Chefs* and *Waiters*.

Customer:

Limited access to many features including placing orders, making reservations, giving feedback etc.

3.3 Functionality Overview:

3.3.1 Admin:

- CRUD Operations for Customers and Employees.
- Managing Tables.
- Add new Admin.
- View Analytics.
- Notify Users and Reply to their messages.
- View User Reviews.
- Change Username/Password.

- Delete Account.
- Log Out.

3.3.2 Customer:

- Can promote to VIP to access more features and get rewards.
- Place and track orders.
- View order history and details.
- Make a Reservation.
- Can upload Profile.
- Provide feedback.
- Messaging System.
- Change Username/Password.
- Delete Account.
- Log Out.

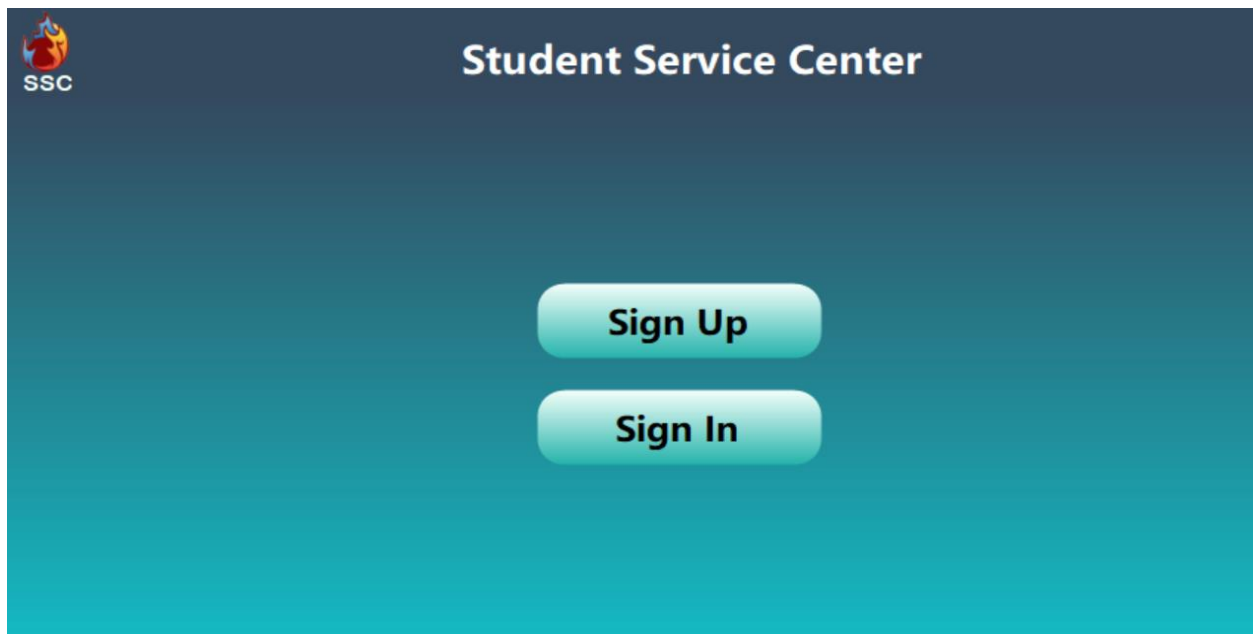
3.3.2 Staff (Waiters and Chefs):

- Table Reservation Management.
- Order Processing.
- Products and Deals Management.
- Order Status Updates.
- Log Out

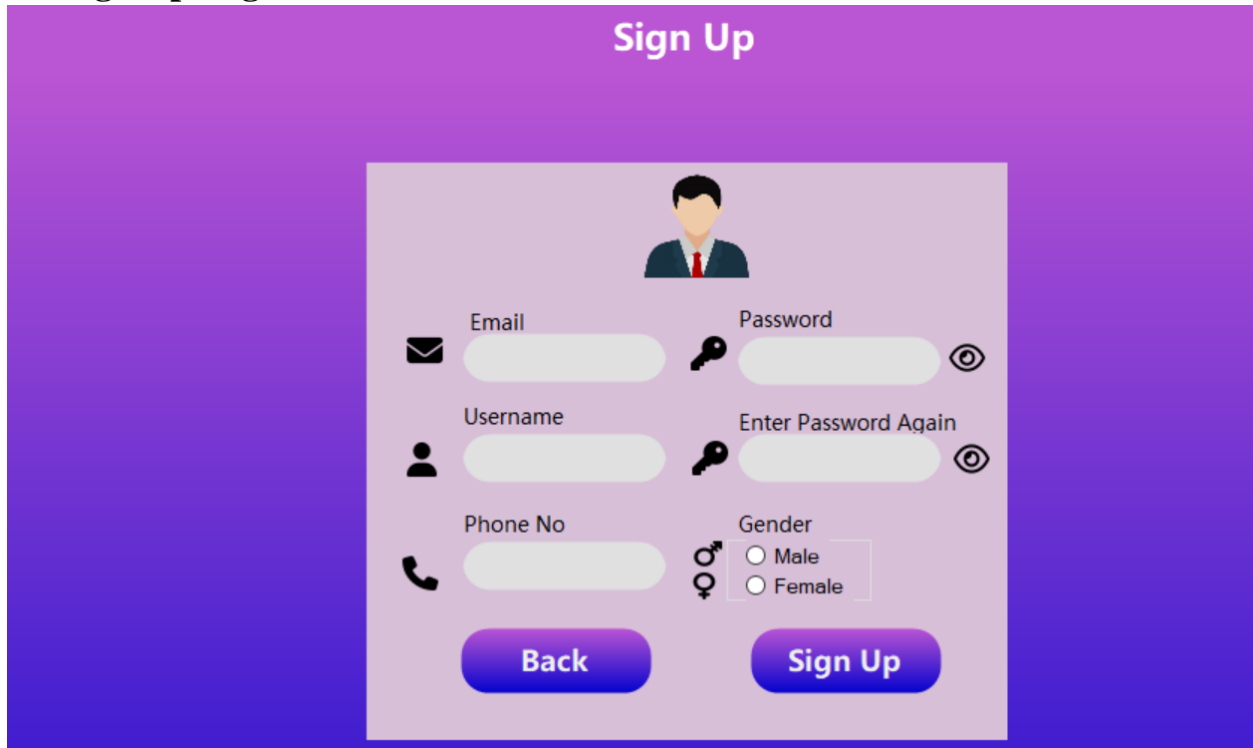
4. Wireframes:

4.1 Homepage:

This is the landing page for all users.

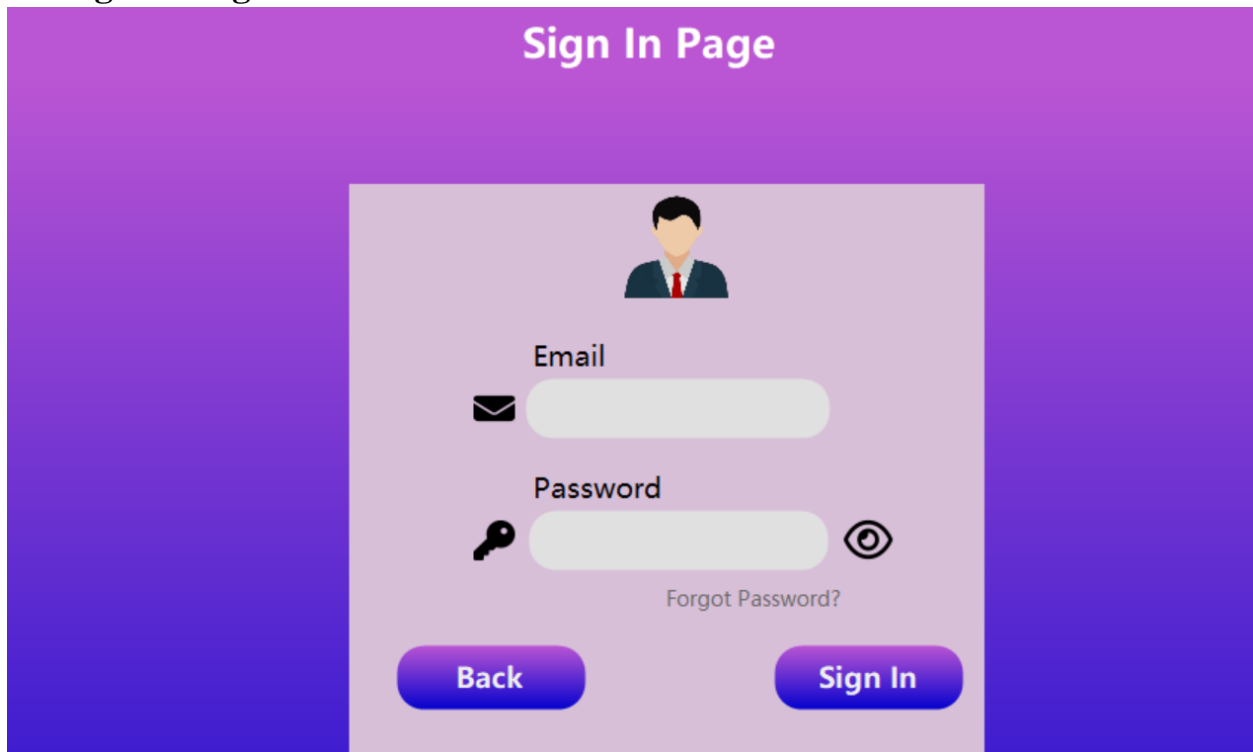


4.2 Sign Up Page:



The Sign Up page features a purple gradient background. At the top center, the text "Sign Up" is displayed in white. Below this, a white card contains a user icon at the top. The form fields are arranged in two columns: "Email" (with an envelope icon) and "Password" (with a key icon and a toggle eye icon) in the first row; "Username" (with a person icon) and "Enter Password Again" (with a key icon and a toggle eye icon) in the second row; and "Phone No" (with a phone icon) and "Gender" (with male and female icons and radio buttons for "Male" and "Female") in the third row. At the bottom of the card are two buttons: "Back" and "Sign Up".

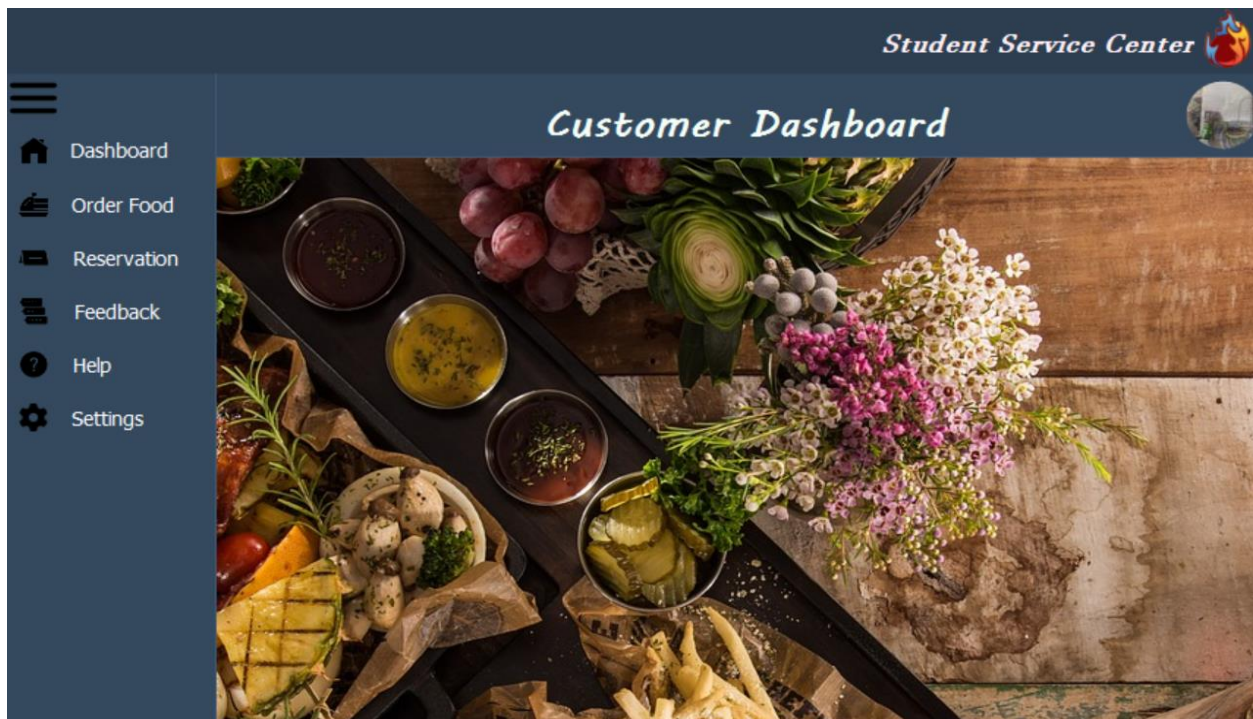
4.3 Sign-In Page:



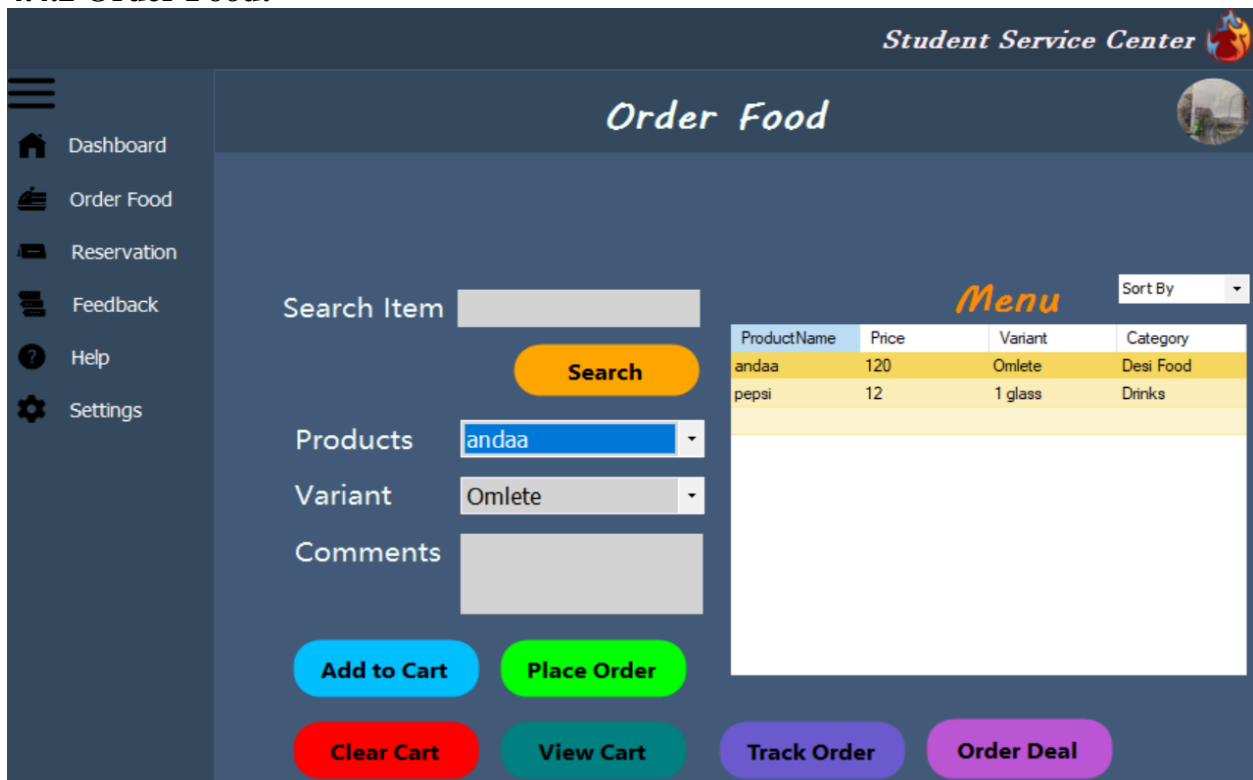
The Sign In page features a purple gradient background. At the top center, the text "Sign In Page" is displayed in white. Below this, a white card contains a user icon at the top. The form fields are arranged in two rows: "Email" (with an envelope icon) in the first row and "Password" (with a key icon and a toggle eye icon) in the second row. Below the password field is a link that says "Forgot Password?". At the bottom of the card are two buttons: "Back" and "Sign In".

4.4 Customer:

4.4.1 Landing Page:



4.4.2 Order Food:



4.4.3 View Cart:

Student Service Center

View Cart

Dashboard

Order Food

Reservation

Feedback

Help

Settings

Search

Product

Variant

Delete

Update

Back

ProductName	Quantity
andaa	Omlete
pepsi	1 glass

4.4.4 Book Table:

Student Service Center

Book Table

Dashboard

Order Food

Reservation

Feedback

Help

Settings

Available Tables

Total Persons

Reservation Date

Make Reservation

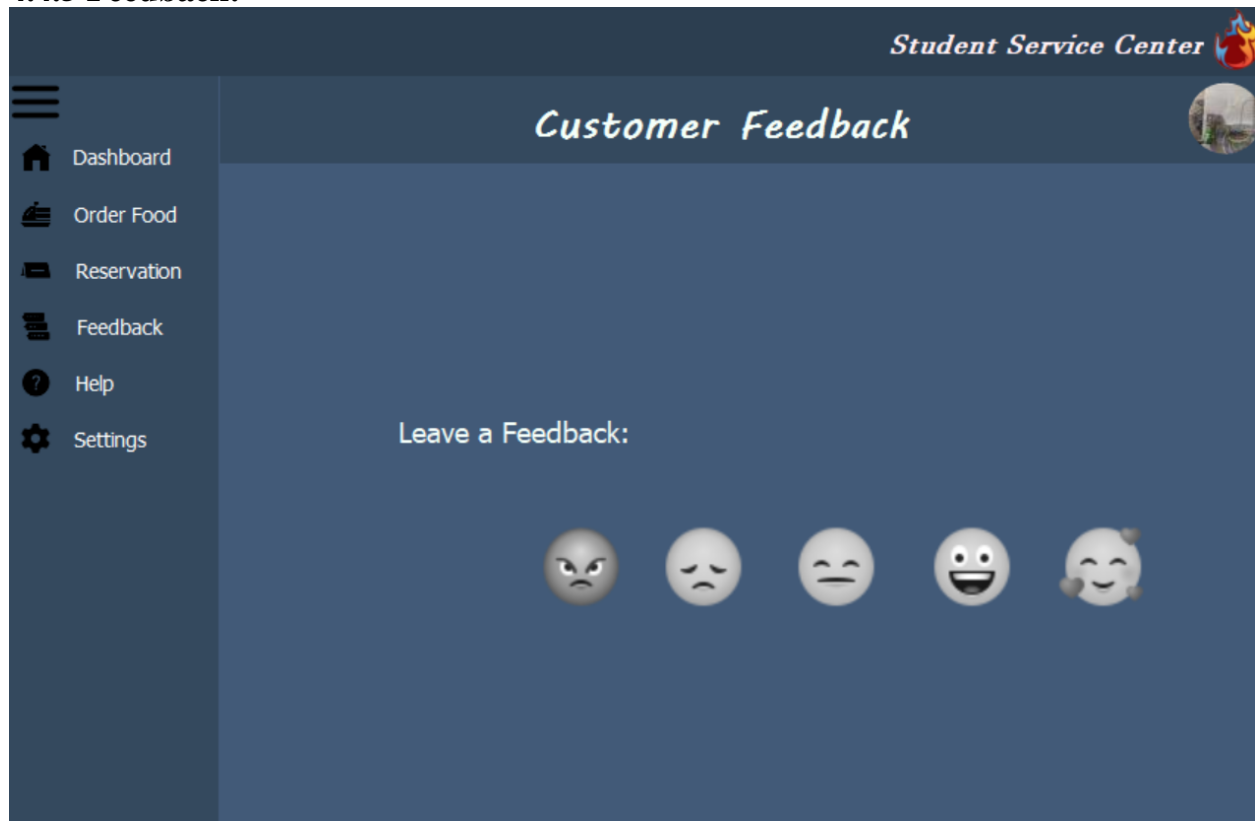
View Reservation

Delete Reservation

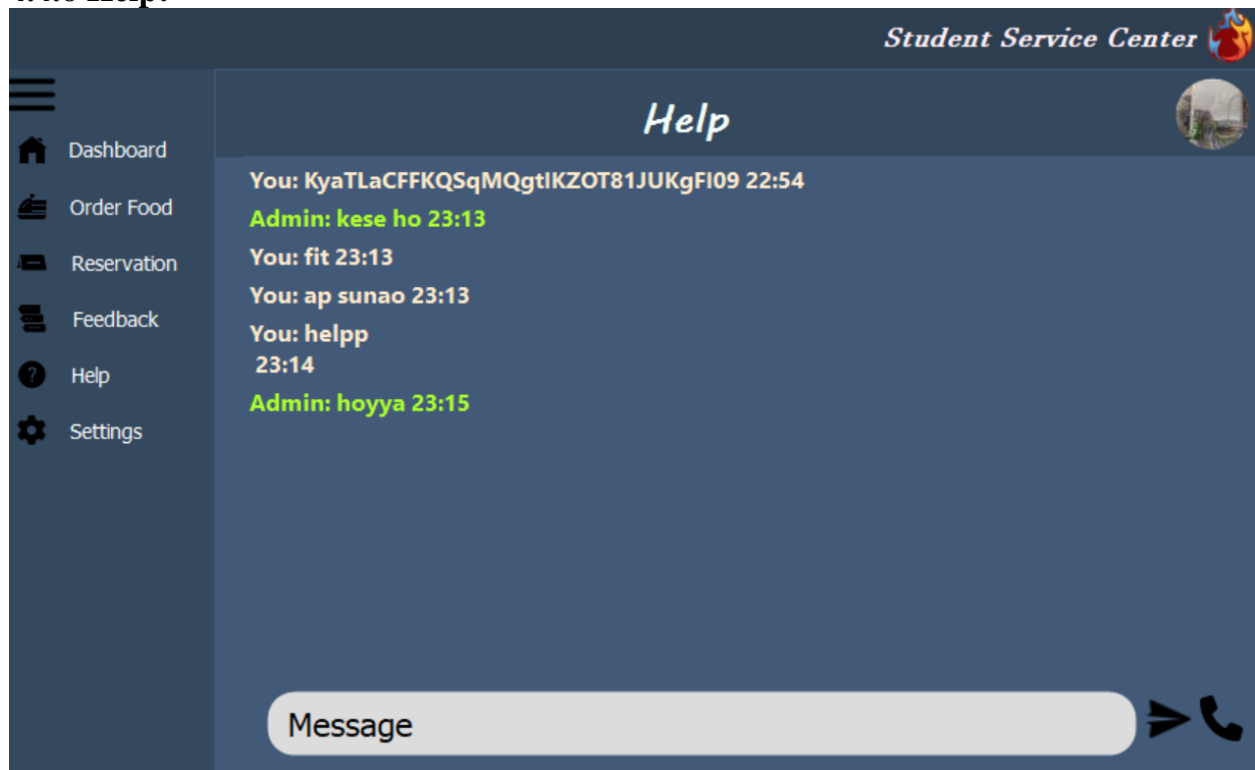
7

11:36:35 AM

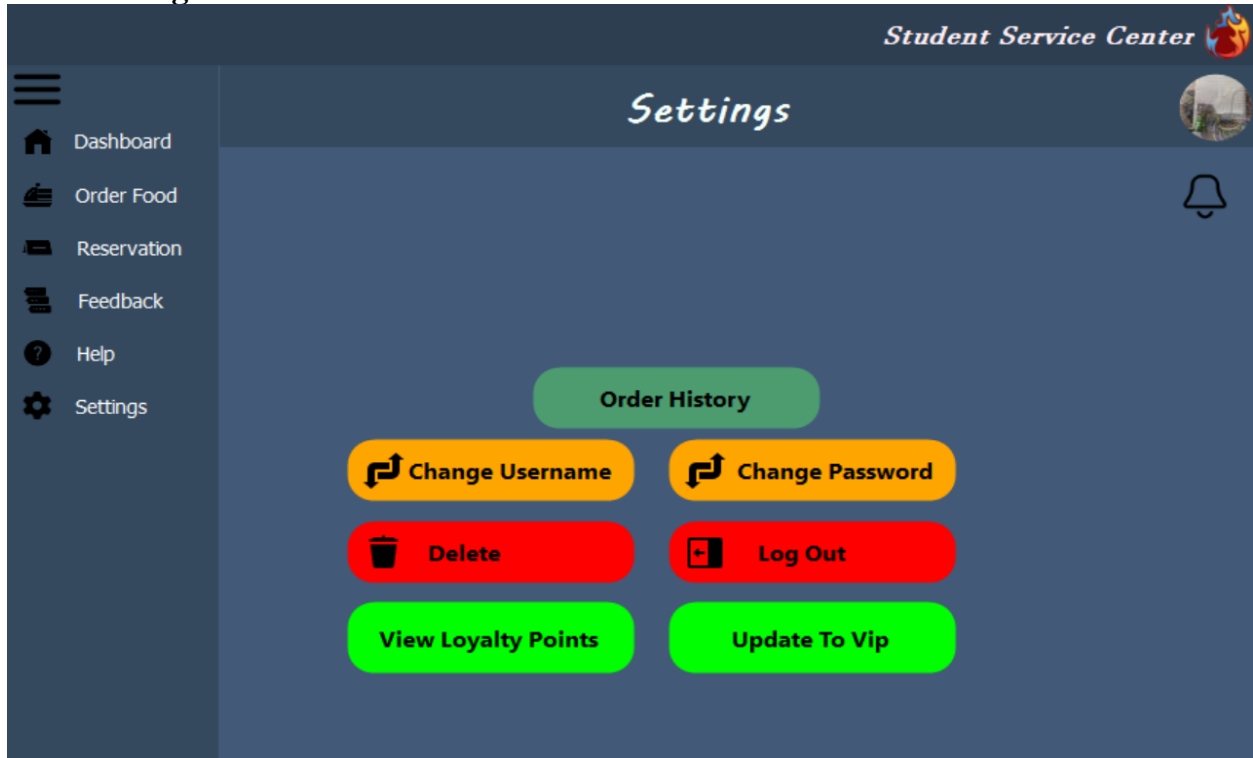
4.4.5 Feedback:



4.4.6 Help:

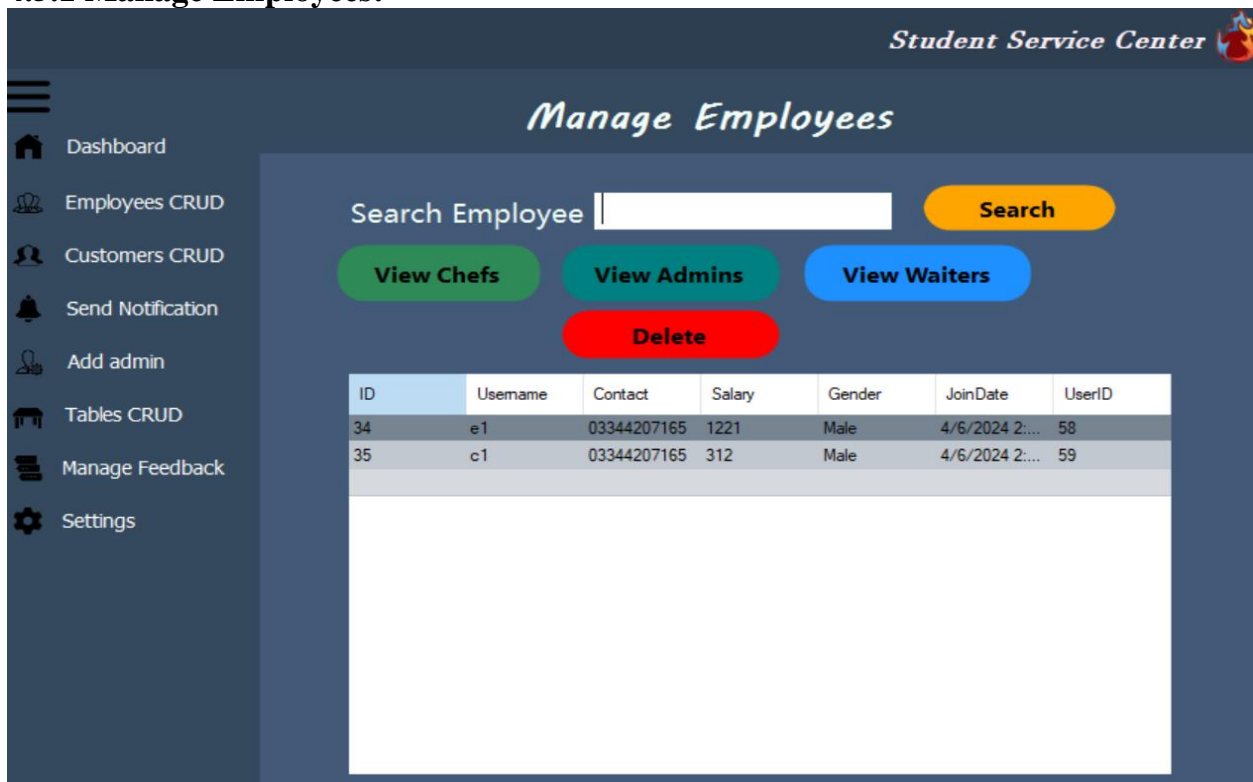


4.4.7 Settings:

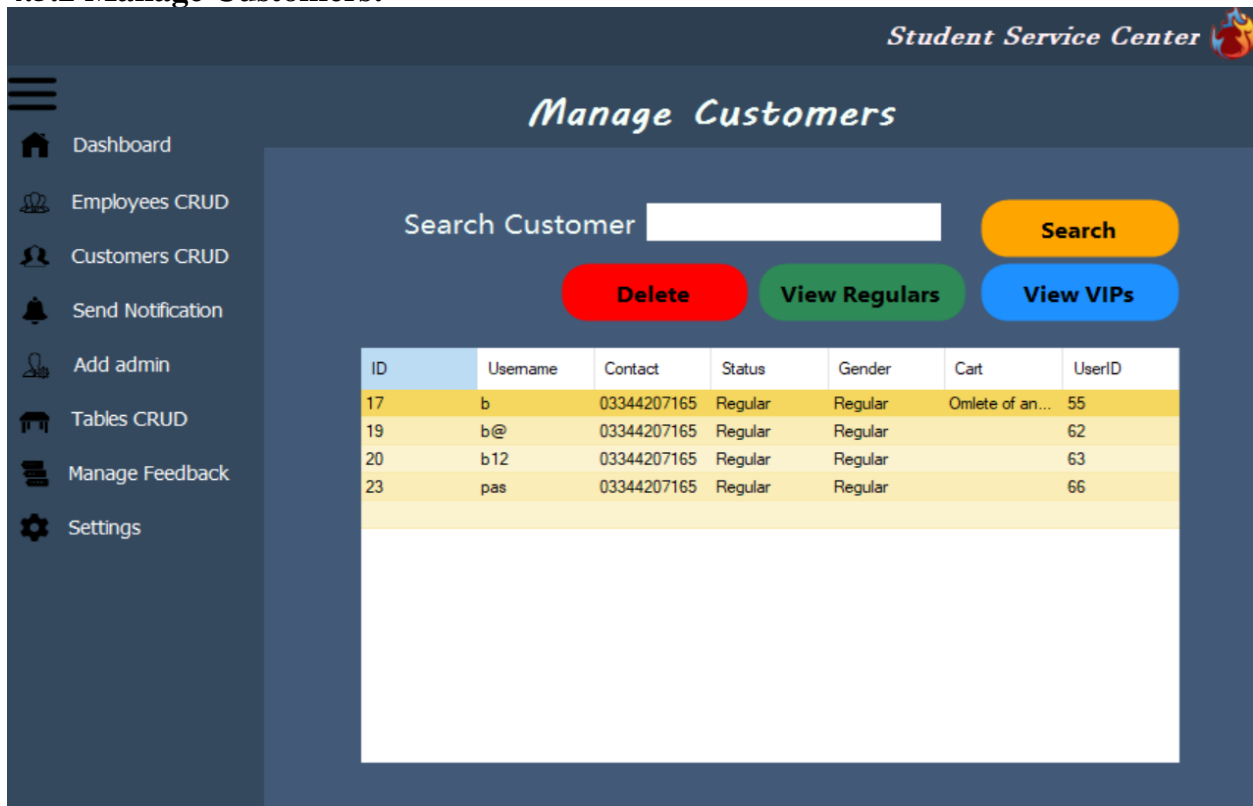


4.5 Admin:

4.5.1 Manage Employees:



4.5.2 Manage Customers:



Student Service Center

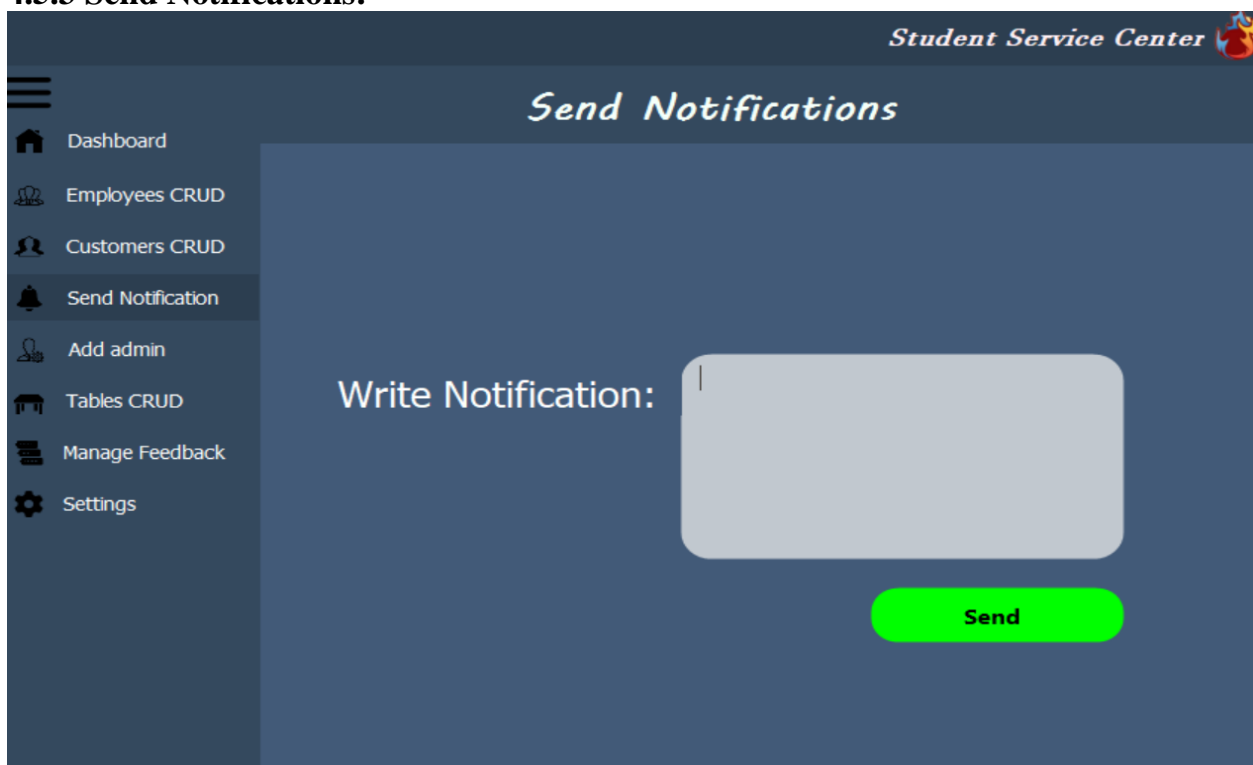
Manage Customers

Search Customer **Search**

Delete **View Regulars** **View VIPs**

ID	Username	Contact	Status	Gender	Cart	UserID
17	b	03344207165	Regular	Regular	Omlete of an...	55
19	b@	03344207165	Regular	Regular		62
20	b12	03344207165	Regular	Regular		63
23	pas	03344207165	Regular	Regular		66

4.5.3 Send Notifications:



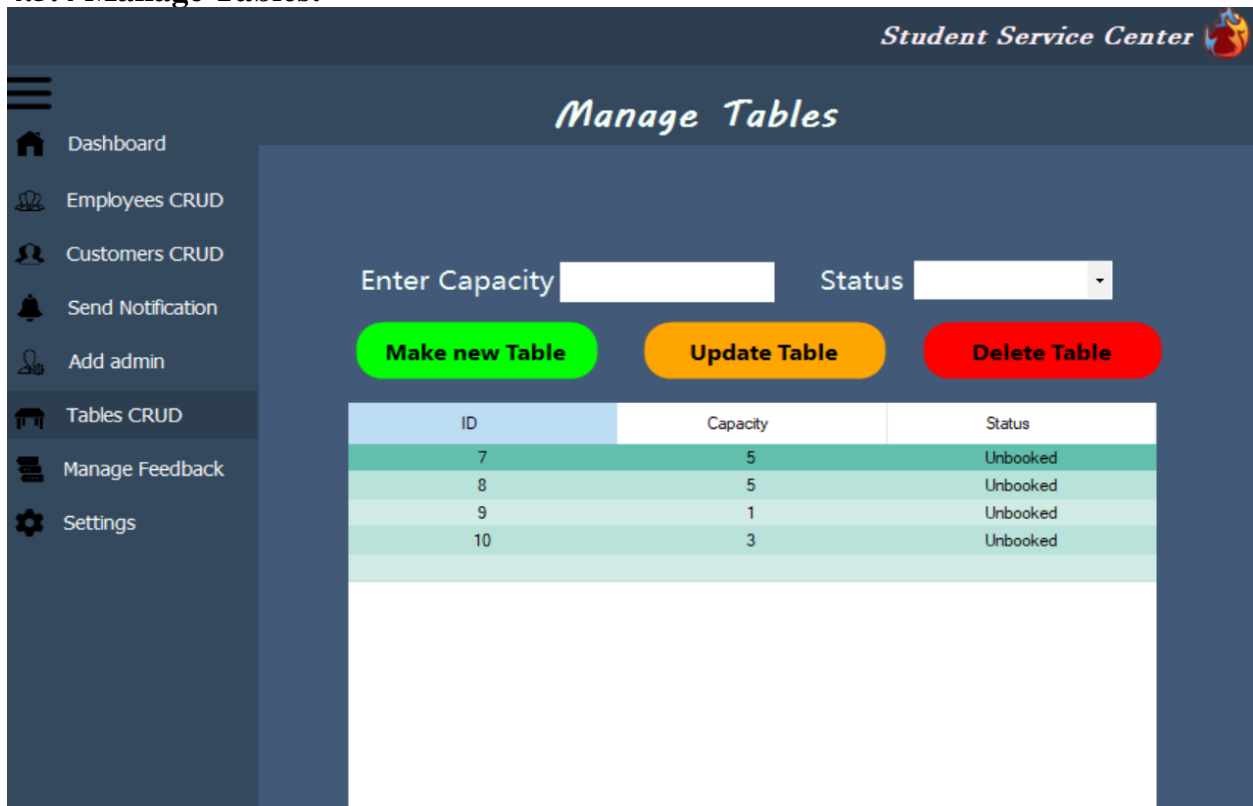
Student Service Center

Send Notifications

Write Notification:

Send

4.5.4 Manage Tables:



Student Service Center

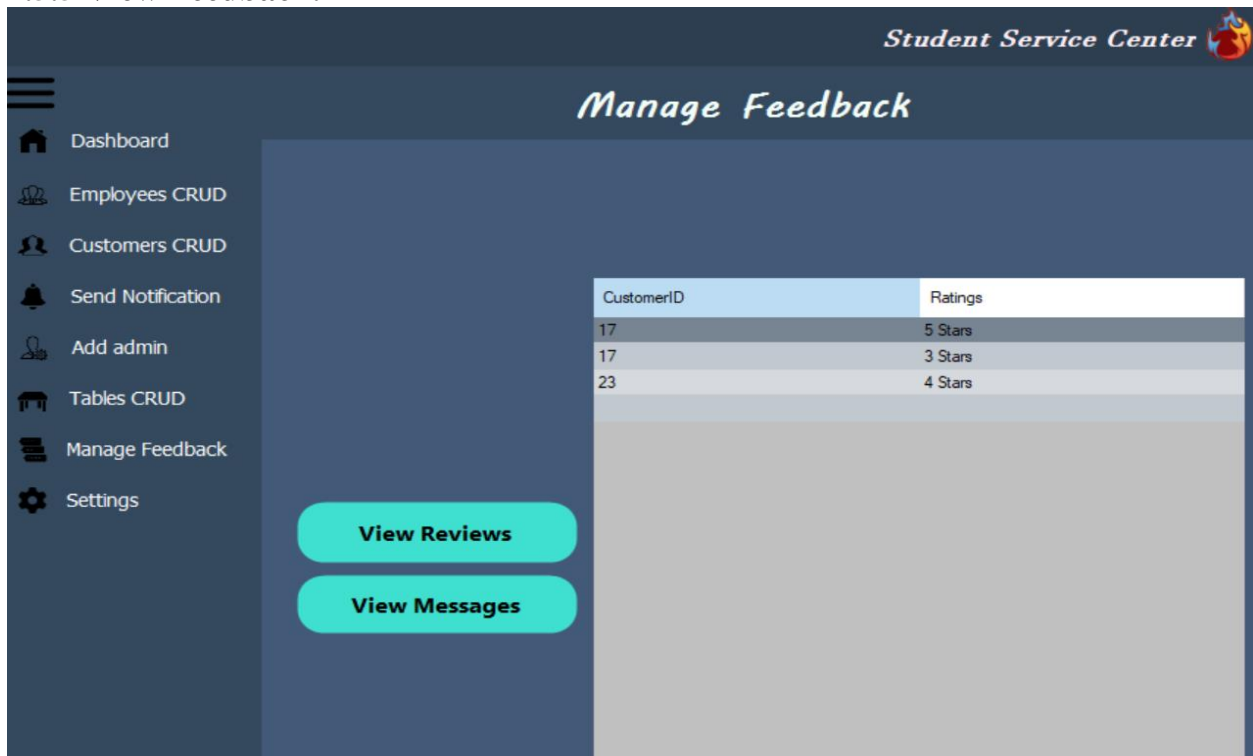
Manage Tables

Enter Capacity Status

Make new Table **Update Table** **Delete Table**

ID	Capacity	Status
7	5	Unbooked
8	5	Unbooked
9	1	Unbooked
10	3	Unbooked

4.5.5 View Feedback:



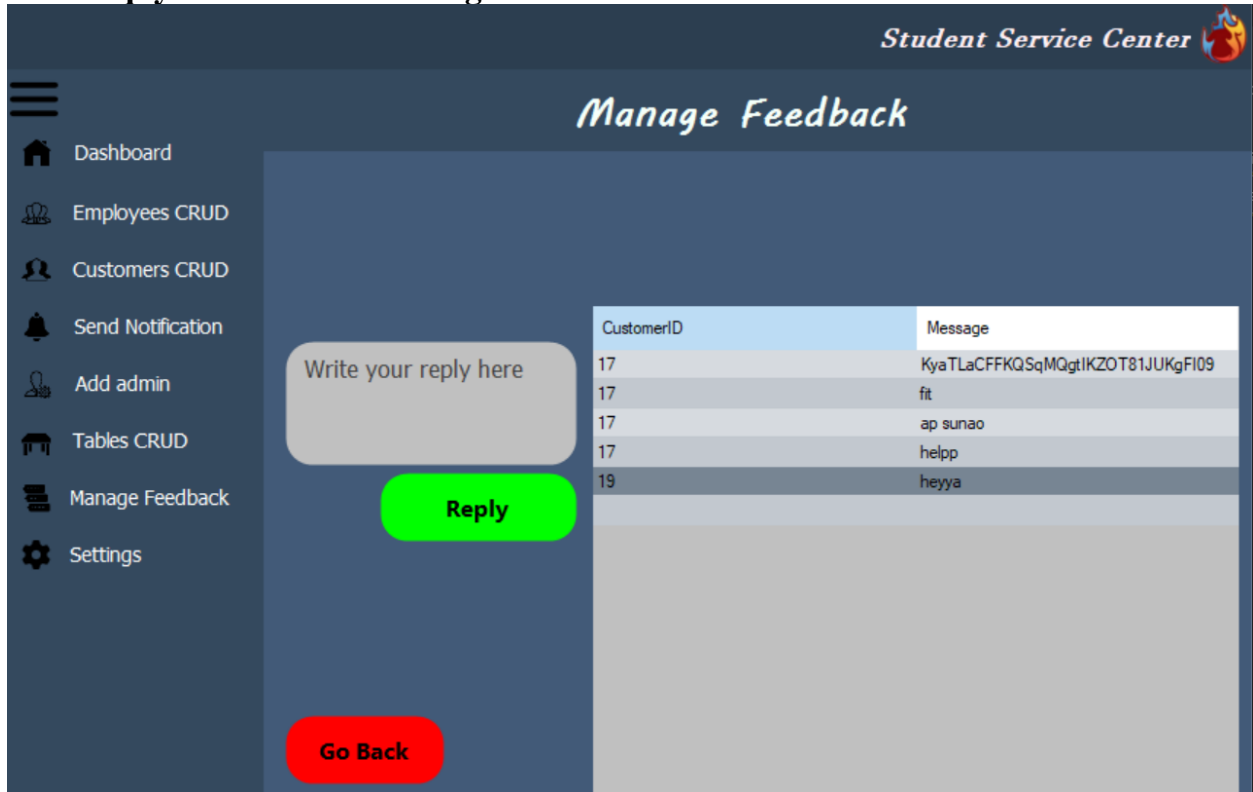
Student Service Center

Manage Feedback

CustomerID	Ratings
17	5 Stars
17	3 Stars
23	4 Stars

View Reviews **View Messages**

4.5.6 Reply to Customer Messages:



Student Service Center

Manage Feedback

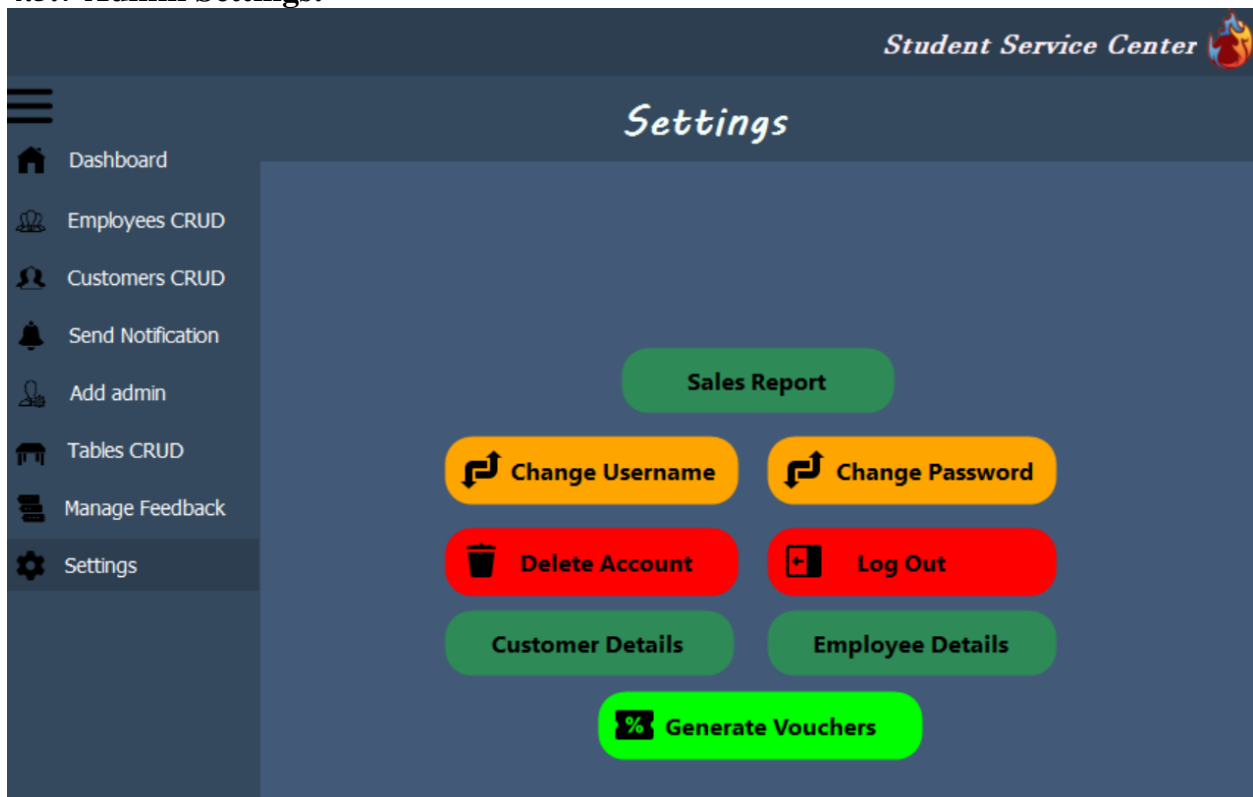
Write your reply here

Reply

Go Back

CustomerID	Message
17	KyaTLaCFFKQsqMQgtIKZOT81JUKgFI09
17	fit
17	ap sunao
17	helpp
19	heyya

4.5.7 Admin Settings:



Student Service Center

Settings

Sales Report

Change Username **Change Password**

Delete Account **Log Out**

Customer Details **Employee Details**

% Generate Vouchers

4.6 Chef:

4.6.1 Manage Products:

Manage Products

Manage ProductsManage Orders

Name

CategoryFast Food

Search

Description

Delete

Update

Insert

Variant

Launch

Make a new Deal

ID	Name	Category	Description	Available
11	andaa	Desi Food	andaaa	1
12	pepsi	Drinks	Drink	1

4.6.2 Manage Orders:

Manage Orders

Manage ProductsManage Orders

ID	Products	Comments
92	1 glass of pepsi;Omlate of andaa;Omlate of andaa;Omlate of andaa	

Notify Waiter

4.7 Waiter:

4.7.1: Take Order:

Take Order

Pickup

Deliver

View Reservations

Take Order

Pickup

Deliver

View Reservations

←

Deals

Products

Variant

Comments

Pay Method

Add to Cart

Place Order

Clear Cart

Order Deal

ProductName	Price	Variant	Category
andaa	120	Omlete	Desi Food
pepsi	12	1 glass	Drinks

4.7.2 Pickup and Deliver Order:

Pickup Order

Take Order

Pickup

Deliver

View Reservations

Pickup Order

Take Order

Pickup

Deliver

View Reservations

←

ID	Products	Comments
----	----------	----------

4.7.3 Manage Reservations:

Manage Reservations

Take Order

Pickup

Deliver

View Reservation

Available Tables

Total Persons

Reservation Date

Make Reservation

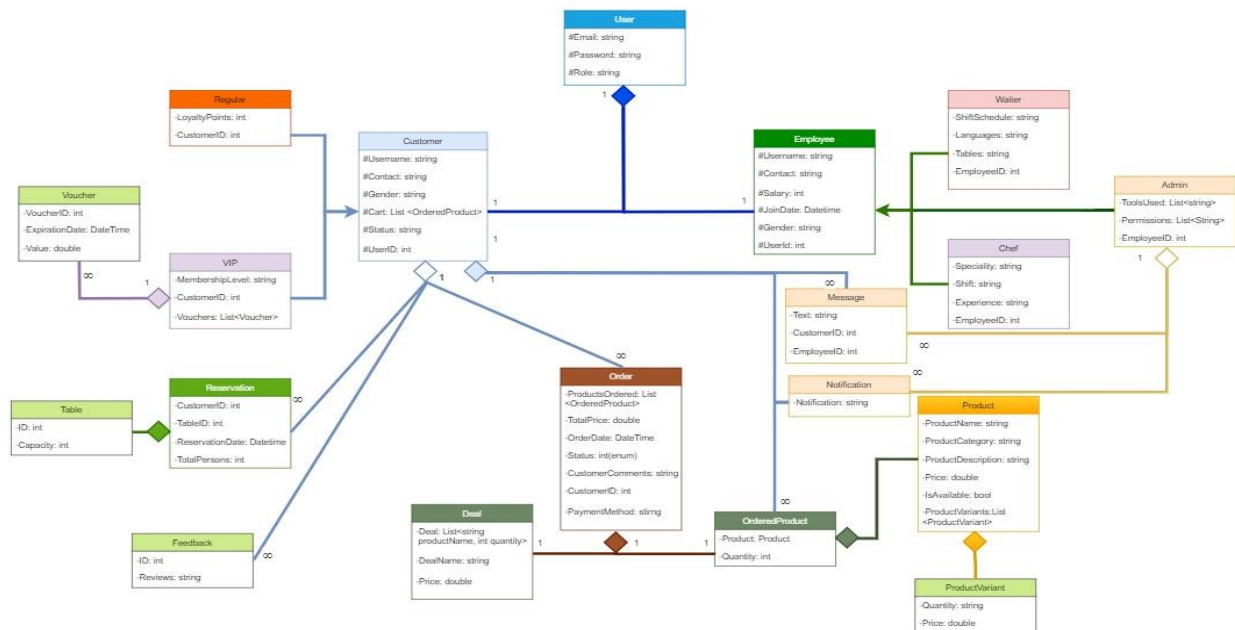
Delete Reservation

7

12:28:26 PM

ID	TotalPersons	reservationsDate	TableID	CustomerID
----	--------------	------------------	---------	------------

5. CRC Diagram:



6. Code:

```
public class Feedback
{
    private string Review;
    private int CustomerID;

    public Feedback(string review, int customerID)
    {
        Review = review;
        CustomerID = customerID;
    }

    public string GetReview()
    {
        return Review;
    }

    public int GetCustomerID()
    {
        return CustomerID;
    }

    public void SetCustomerID(int id)
    {
        CustomerID = id;
    }
    public void SetReview(string review)
    {
        Review = review;
    }
}

public interface IFeedbackDL
{
    void SaveReview(Feedback feedback);
    List<Feedback> GetReviews();
}

public class FeedbackDBDL : IFeedbackDL
{
    public void SaveReview(Feedback feedback)
    {

```

```
        using (SqlConnection connection = UtilityFunctions.GetSqlConnection())
        {
            connection.Open();
            SqlCommand command = new SqlCommand("INSERT INTO Feedback(Reviews,
CustomerID) VALUES (@Reviews, @CustomerID)", connection);
            command.Parameters.AddWithValue("@Reviews", feedback.GetReview());
            command.Parameters.AddWithValue("@CustomerID", feedback.GetCustomerID());
            command.ExecuteNonQuery();
        }
    }

    public List<Feedback> GetReviews()
    {
        using (SqlConnection connection = UtilityFunctions.GetSqlConnection())
        {
            connection.Open();
            SqlCommand command = new SqlCommand("SELECT * FROM Feedback",
connection);
            SqlDataReader reader = command.ExecuteReader();
            List<Feedback> feedbacks = new List<Feedback>();
            while (reader.Read())
            {
                feedbacks.Add(new Feedback(reader.GetString(1), reader.GetInt32(2)));
            }
            return feedbacks;
        }
    }
}

public class FeedbackFHDL : IFeedbackDL
{
    public void SaveReview(Feedback feedback)
    {
        string path = UtilityFunctions.GetPath("Feedbacks.txt");
        int id = UtilityFunctions.AssignID(path);
        using (StreamWriter writer = File.AppendText(path))
        {
            writer.WriteLine($"{id},{feedback.GetReview()}, {feedback.GetCustomerID()}");
        }
    }

    public List<Feedback> GetReviews()
    {

```

```
List<Feedback> feedbacks = new List<Feedback>();
string path = UtilityFunctions.GetPath("Feedbacks.txt");

if (File.Exists(path))
{
    string[] lines = File.ReadAllLines(path);
    foreach (string line in lines)
    {
        string[] parts = line.Split(',');
        if (parts.Length == 3)
        {
            string review = parts[1].Trim();
            int customerID = Convert.ToInt32(parts[2].Trim());

            Feedback feedback = new Feedback(review, customerID);
            feedbacks.Add(feedback);
        }
    }
}
return feedbacks;
}

public class FeedbackUI
{
    public static int TakeFeedback()
    {
        Console.ForegroundColor = ConsoleColor.DarkBlue;
        Console.WriteLine("\n\t\t\t\t\t5 Stars");
        Console.WriteLine("\t\t\t\t\t4 Stars");
        Console.WriteLine("\t\t\t\t\t3 Stars");
        Console.WriteLine("\t\t\t\t\t2 Stars");
        Console.WriteLine("\t\t\t\t\t1 Stars");
        Console.WriteLine("\t\t\t\t\tLog Out");
        Console.ResetColor();
        return ConsoleUtility.MovementOfArrow(53, 7, 1, 6);
    }

    public static void PrintReviews()
    {
        Console.ForegroundColor = ConsoleColor.DarkBlue;
        foreach (var feedback in ObjectHandler.GetFeedbackDL().GetReviews())
        {
```

```
        Console.WriteLine("\t\t\t\t\tFeedback: " + feedback.GetReview() + " Customer ID: " + feedback.GetCustomerID());
    }
    Console.ForegroundColor = ConsoleColor.DarkBlue;
}
}
```

7. GitHub Link:

<https://github.com/mb0227/Business-App-Project-2nd-Semester>