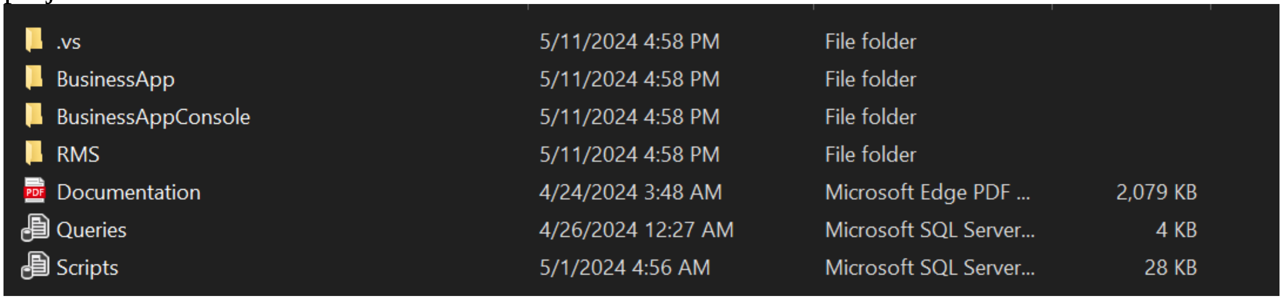
Table of Contents

1. How to Use:

- First, clone the repository or download the zip file from https://github.com/mb0227/Business-App-Project-2nd-Semester. I recommend cloning to avoid any issues.
- You should have Visual Studio installed (I recommend community version 2022).
- There are 3 Folders that are displayed at first. RMS has the complete backend of the project, BusinessApp is GUI version of the project and BusinessAppConsole is limited version of my project on Console.

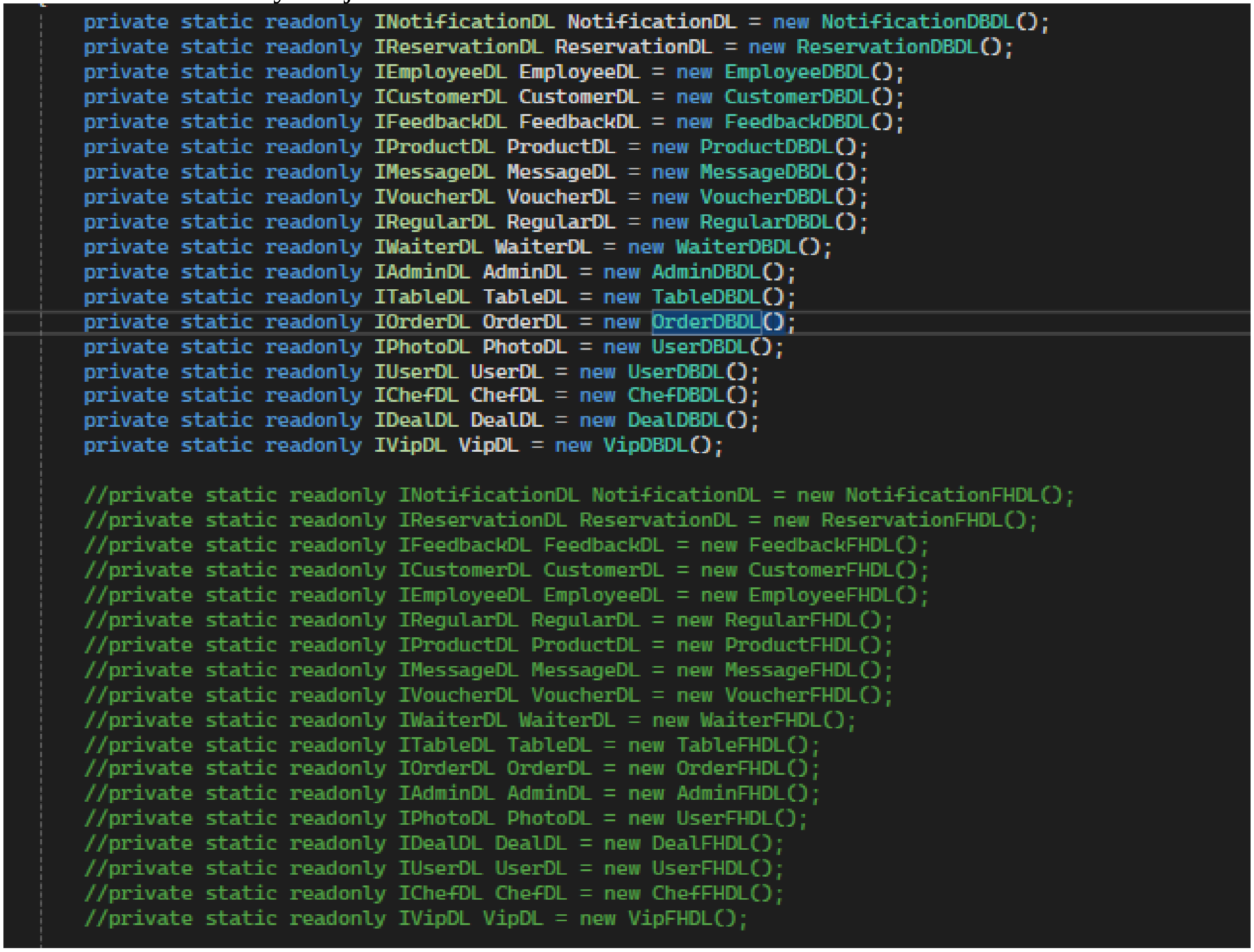


.vs	5/11/2024 4:58 PM	File folder	
BusinessApp	5/11/2024 4:58 PM	File folder	
BusinessAppConsole	5/11/2024 4:58 PM	File folder	
RMS	5/11/2024 4:58 PM	File folder	
Documentation	4/24/2024 3:48 AM	Microsoft Edge PDF ...	2,079 KB
Queries	4/26/2024 12:27 AM	Microsoft SQL Server...	4 KB
Scripts	5/1/2024 4:56 AM	Microsoft SQL Server...	28 KB

- Now there are two modes in which this project can be run. First is by using **Database** and other is by **File Handling**. I recommend using Database for better experience.

Database:

- Go to RMS>Utility>ObjectHandler.cs. Comment out all the lines which use FH.



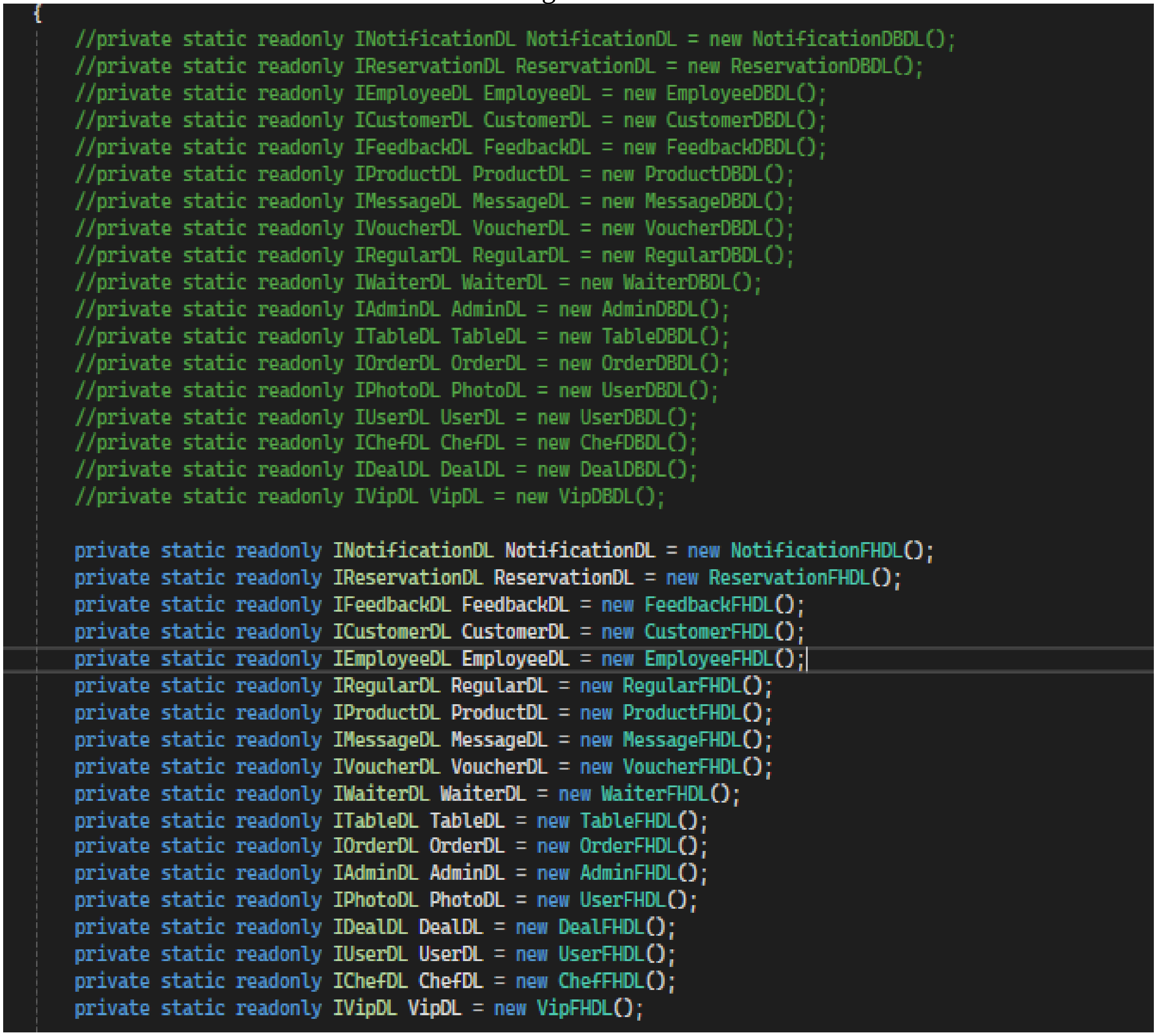
```
private static readonly INotificationDL NotificationDL = new NotificationDBDL();
private static readonly IReservationDL ReservationDL = new ReservationDBDL();
private static readonly IEmployeeDL EmployeeDL = new EmployeeDBDL();
private static readonly ICustomerDL CustomerDL = new CustomerDBDL();
private static readonly IFeedbackDL FeedbackDL = new FeedbackDBDL();
private static readonly IProductDL ProductDL = new ProductDBDL();
private static readonly IMessageDL MessageDL = new MessageDBDL();
private static readonly IVoucherDL VoucherDL = new VoucherDBDL();
private static readonly IRegularDL RegularDL = new RegularDBDL();
private static readonly IWaiterDL WaiterDL = new WaiterDBDL();
private static readonly IAdminDL AdminDL = new AdminDBDL();
private static readonly ITableDL TableDL = new TableDBDL();
private static readonly IOrderDL OrderDL = new OrderDBDL();
private static readonly IPhotoDL PhotoDL = new UserDBDL();
private static readonly IUserDL UserDL = new UserDBDL();
private static readonly IChefDL ChefDL = new ChefDBDL();
private static readonly IDealDL DealDL = new DealDBDL();
private static readonly IVipDL VipDL = new VipDBDL();

//private static readonly INotificationDL NotificationDL = new NotificationFHDL();
//private static readonly IReservationDL ReservationDL = new ReservationFHDL();
//private static readonly IFeedbackDL FeedbackDL = new FeedbackFHDL();
//private static readonly ICustomerDL CustomerDL = new CustomerFHDL();
//private static readonly IEmployeeDL EmployeeDL = new EmployeeFHDL();
//private static readonly IRegularDL RegularDL = new RegularFHDL();
//private static readonly IProductDL ProductDL = new ProductFHDL();
//private static readonly IMessageDL MessageDL = new MessageFHDL();
//private static readonly IVoucherDL VoucherDL = new VoucherFHDL();
//private static readonly IWaiterDL WaiterDL = new WaiterFHDL();
//private static readonly ITableDL TableDL = new TableFHDL();
//private static readonly IOrderDL OrderDL = new OrderFHDL();
//private static readonly IAdminDL AdminDL = new AdminFHDL();
//private static readonly IPhotoDL PhotoDL = new UserFHDL();
//private static readonly IDealDL DealDL = new DealFHDL();
//private static readonly IUserDL UserDL = new UserFHDL();
//private static readonly IChefDL ChefDL = new ChefFHDL();
//private static readonly IVipDL VipDL = new VipFHDL();
```

- You have to install **SQL Server Management Studio** and **SQL Server Configuration Manager** and set them up.
- Open SSMS and go to File>Connect Object Explorer and then copy the Server Name. Now Go to RMS folder and open its solution. Go to Utility>UtilityFunctions.cs. Now in GetSqlConnection function just paste the server name (copied from SSMS) against Data Source. It should be Like `"Data Source=YourServerName;Initial Catalog=RMS;Integrated Security=True"`
- Then first go to new query and write **CREATE DATABASE RMS.** Run this query.
- Then go to the **queries** file or **scripts** file (any One) and run them. Now the tables shall be created in database.

File Handling:

- Just comment out DB Lines. No other change needed.



```
{
    //private static readonly INotificationDL NotificationDL = new NotificationDBDL();
    //private static readonly IReservationDL ReservationDL = new ReservationDBDL();
    //private static readonly IEmployeeDL EmployeeDL = new EmployeeDBDL();
    //private static readonly ICustomerDL CustomerDL = new CustomerDBDL();
    //private static readonly IFeedbackDL FeedbackDL = new FeedbackDBDL();
    //private static readonly IProductDL ProductDL = new ProductDBDL();
    //private static readonly IMessageDL MessageDL = new MessageDBDL();
    //private static readonly IVoucherDL VoucherDL = new VoucherDBDL();
    //private static readonly IRegularDL RegularDL = new RegularDBDL();
    //private static readonly IWaiterDL WaiterDL = new WaiterDBDL();
    //private static readonly IAdminDL AdminDL = new AdminDBDL();
    //private static readonly ITableDL TableDL = new TableDBDL();
    //private static readonly IOrderDL OrderDL = new OrderDBDL();
    //private static readonly IPhotoDL PhotoDL = new UserDBDL();
    //private static readonly IUserDL UserDL = new UserDBDL();
    //private static readonly IChefDL ChefDL = new ChefDBDL();
    //private static readonly IDealDL DealDL = new DealDBDL();
    //private static readonly IVipDL VipDL = new VipDBDL();

    private static readonly INotificationDL NotificationDL = new NotificationFHDL();
    private static readonly IReservationDL ReservationDL = new ReservationFHDL();
    private static readonly IFeedbackDL FeedbackDL = new FeedbackFHDL();
    private static readonly ICustomerDL CustomerDL = new CustomerFHDL();
    private static readonly IEmployeeDL EmployeeDL = new EmployeeFHDL();
    private static readonly IRegularDL RegularDL = new RegularFHDL();
    private static readonly IProductDL ProductDL = new ProductFHDL();
    private static readonly IMessageDL MessageDL = new MessageFHDL();
    private static readonly IVoucherDL VoucherDL = new VoucherFHDL();
    private static readonly IWaiterDL WaiterDL = new WaiterFHDL();
    private static readonly ITableDL TableDL = new TableFHDL();
    private static readonly IOrderDL OrderDL = new OrderFHDL();
    private static readonly IAdminDL AdminDL = new AdminFHDL();
    private static readonly IPhotoDL PhotoDL = new UserFHDL();
    private static readonly IDealDL DealDL = new DealFHDL();
    private static readonly IUserDL UserDL = new UserFHDL();
    private static readonly IChefDL ChefDL = new ChefFHDL();
    private static readonly IVipDL VipDL = new VipFHDL();
```

- Now right click on RMS located in Solution Explorer, and click on Rebuild.
- Then go to BusinessApp Folder and open its Solution File and run it. The Project is running now, So Enjoy! You can use Console version similarly.

2. Project Description:

2.1 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.

2.2 Scope:

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

2.3 Objectives:

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

2.4 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.

3. Technical Details:

3.1 Development Environment:

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

3.2 Code Style:

I used Pascal case for my Functions naming and for variables in class.

3.3 Programming Languages:

Programming languages used are C# and SQL.

3.4 Framework:

.NET Framework

3.5 Database:

SQL Server for data storage and retrieval.

4. User Guide:

4.1 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.

4.2 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.

4.3 Functionality Overview:

4.3.1 Admin:

1. CRUD Operations for Customers and Employees.
2. Managing Tables.
3. Add new Admin.
4. View Analytics.
5. Notify Users and Reply to their messages.
6. View User Reviews.
7. Change Username/Password.

8. Delete Account.
9. Log Out.

4.3.2 Customer:

1. Can promote to VIP to access more features and get rewards.
2. Place and track orders.
3. View order history and details.
4. Make a Reservation.
5. Can upload Profile.
6. Provide feedback.
7. Messaging System.
8. Change Username/Password.
9. Delete Account.
10. Log Out.

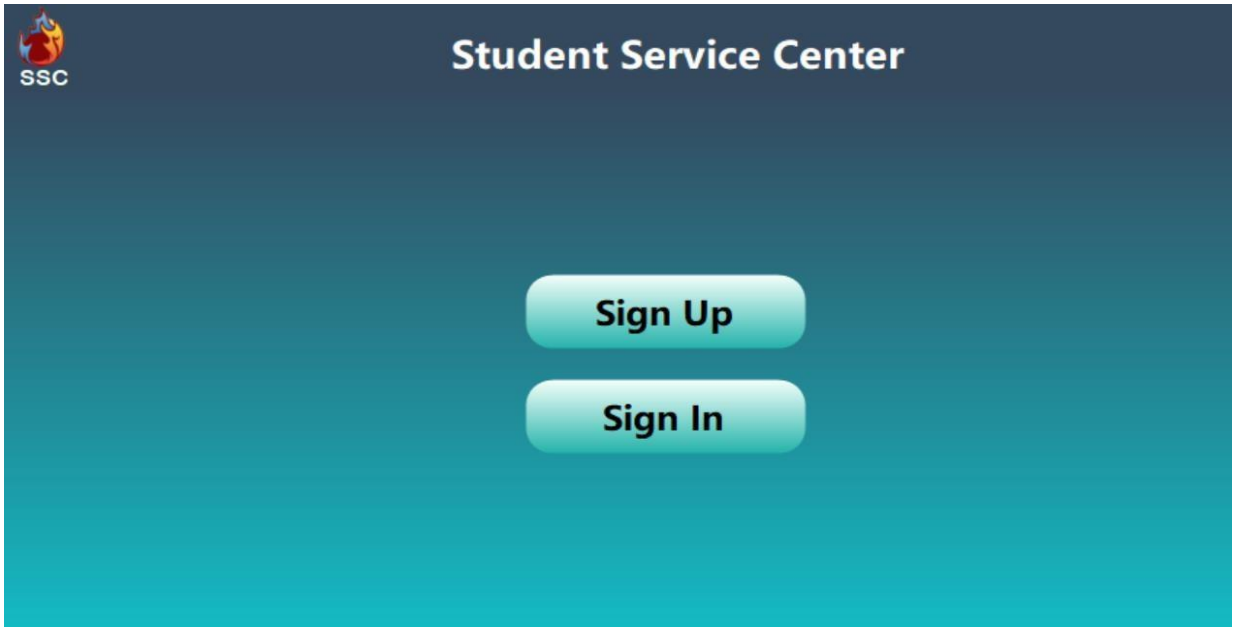
3.3.2 Staff (Waiters and Chefs):

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

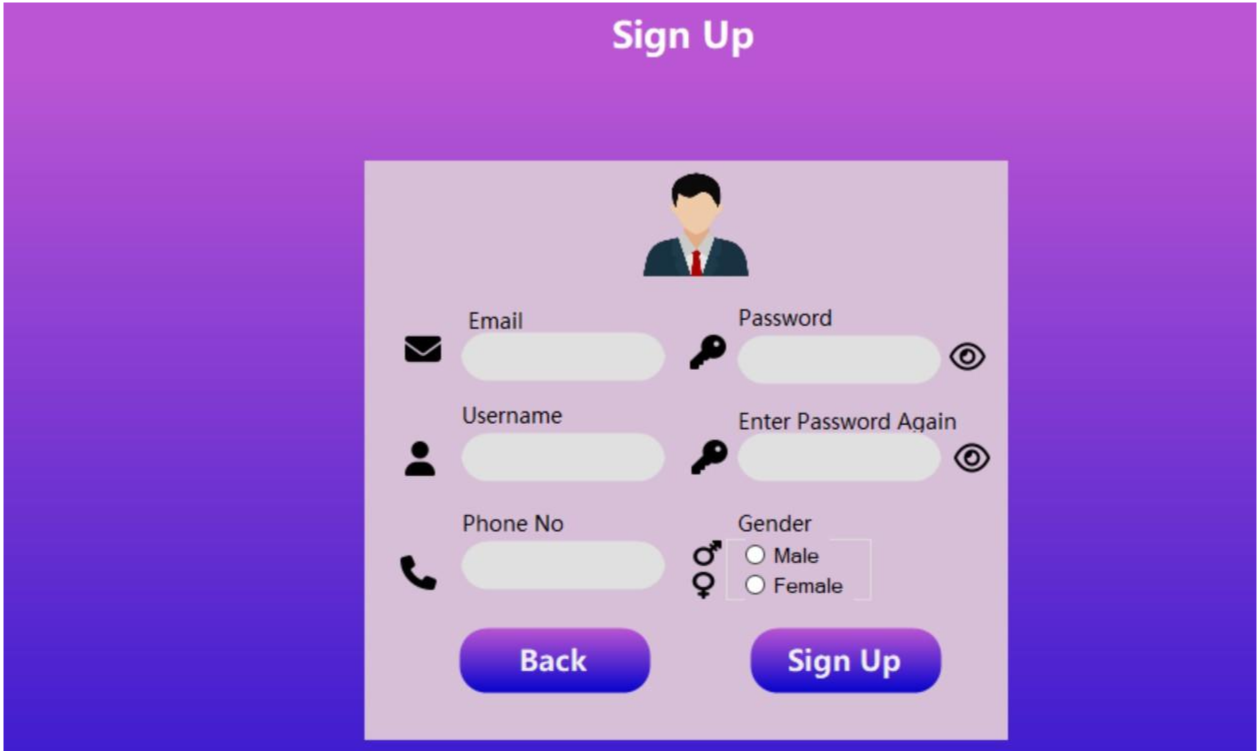
5. Wireframes:

5.1 Homepage:

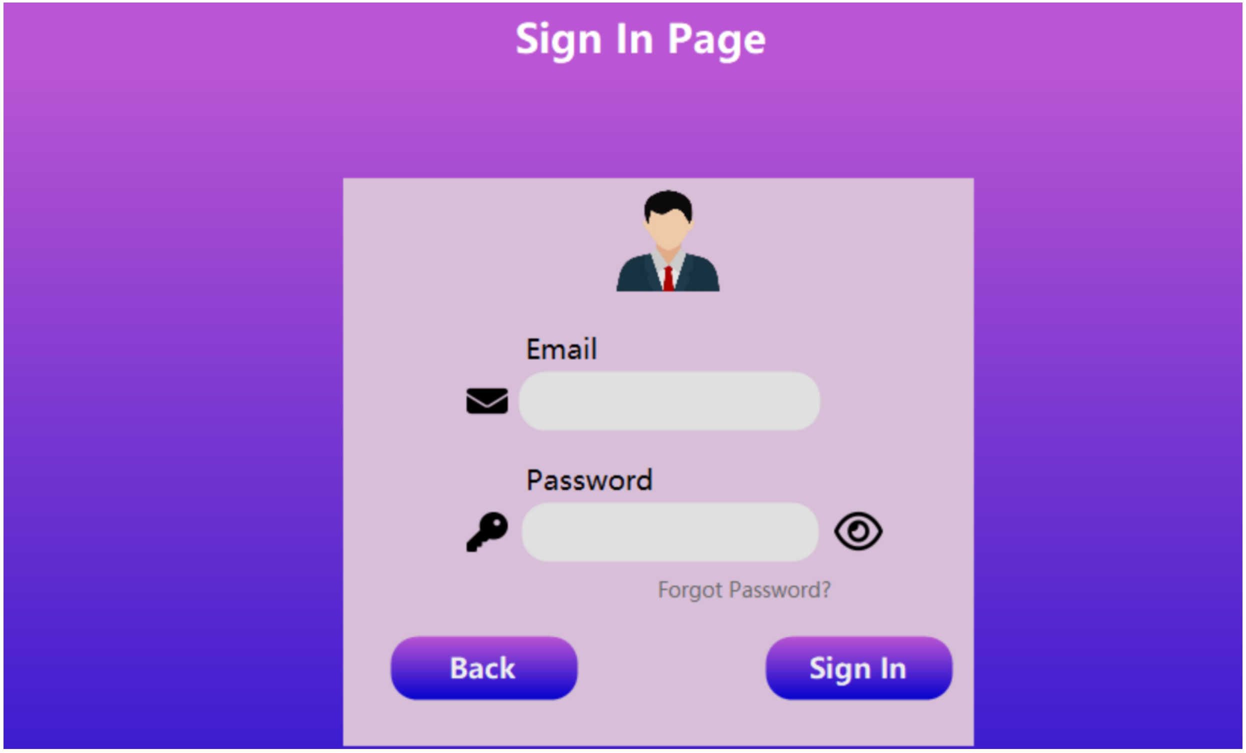
This is the landing page for all users.



5.2 Sign Up Page:

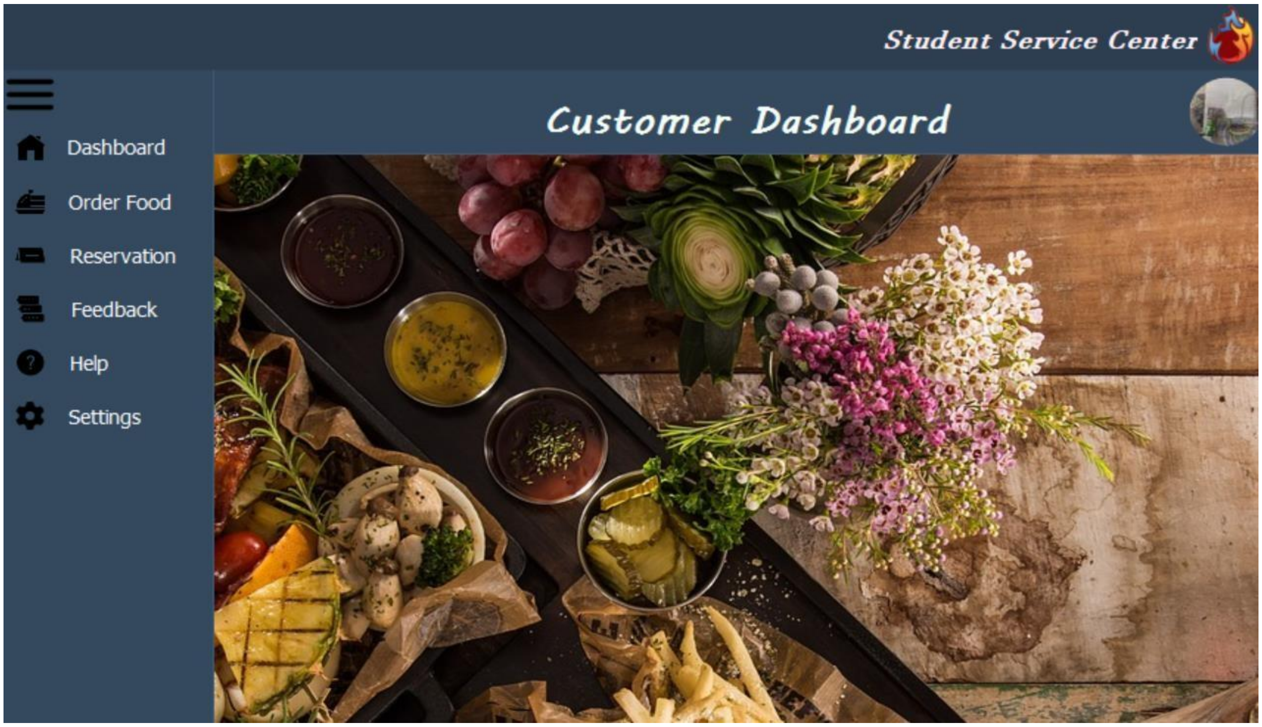


5.3 Sign-In Page:

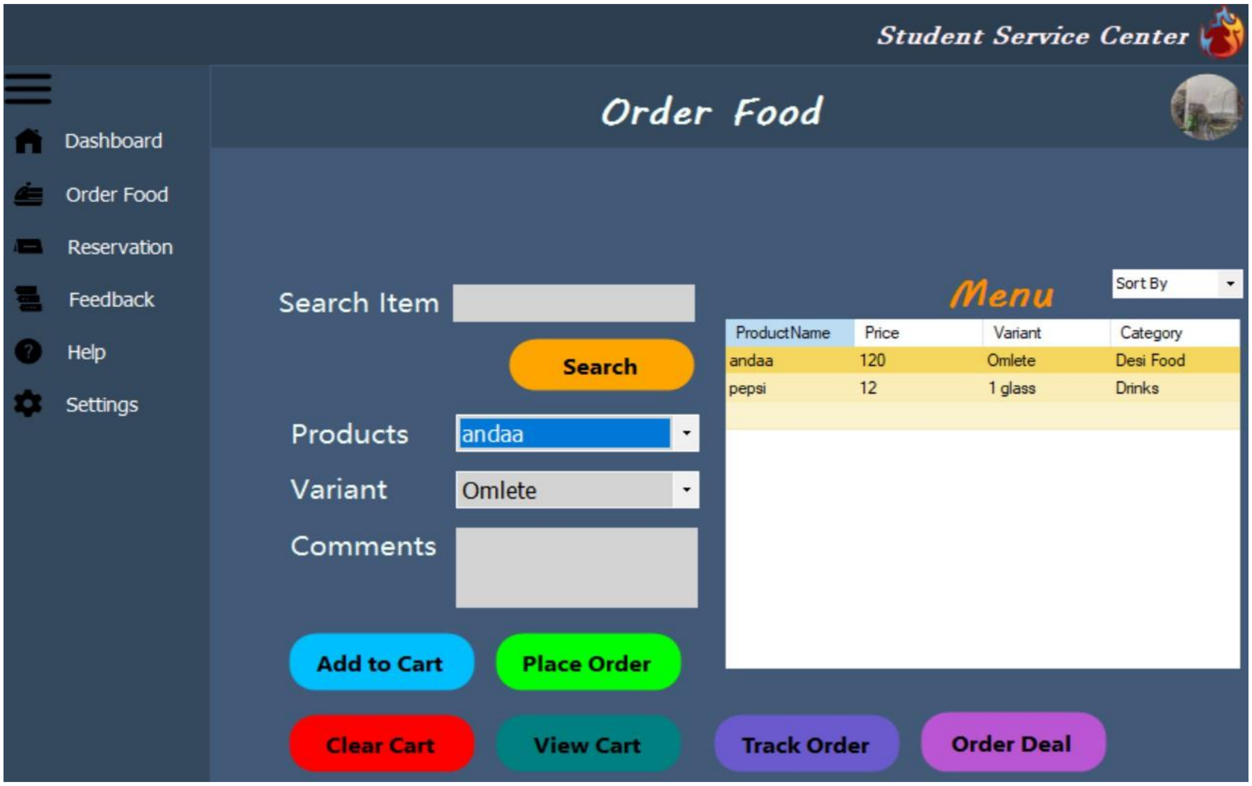


5.4 Customer:

5.4.1 Landing Page:



5.4.2 Order Food:

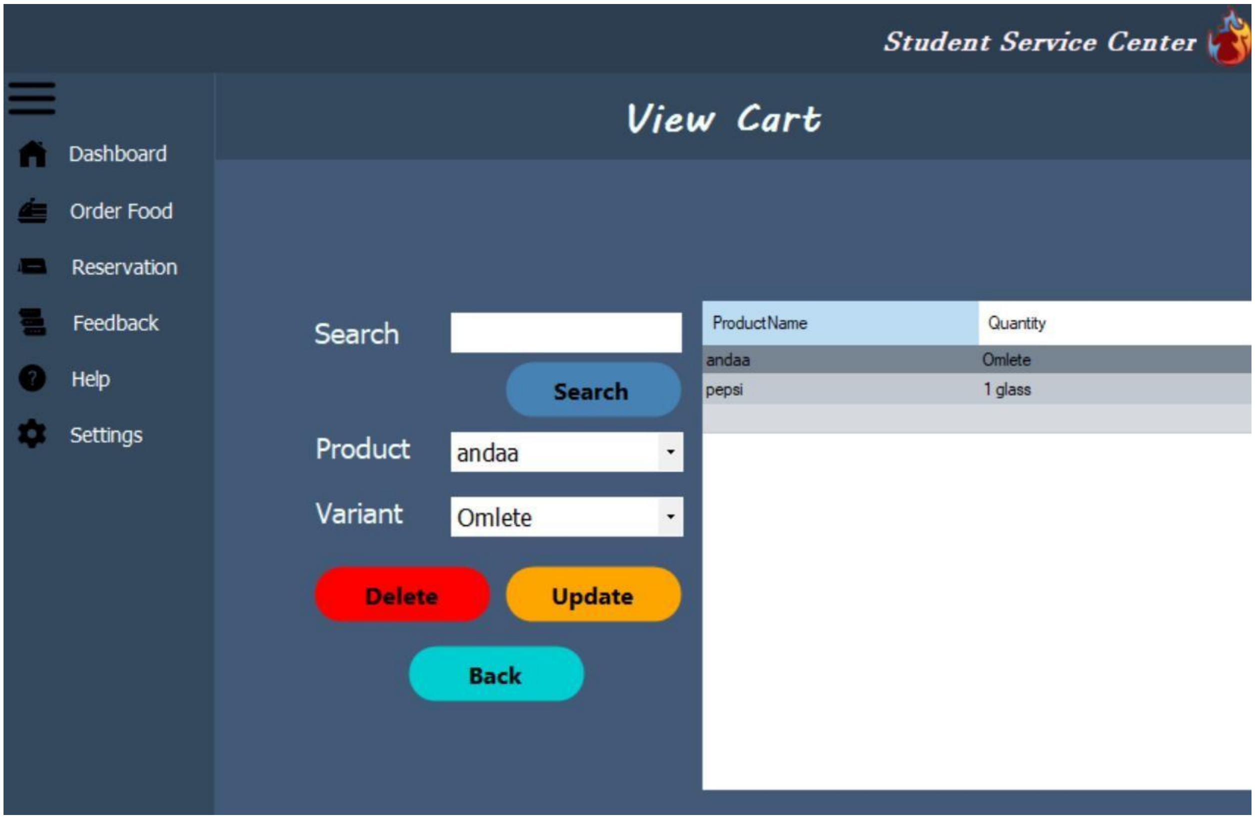


Products: andaa

Variant: Omlete

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

5.4.3 View Cart:



View Cart

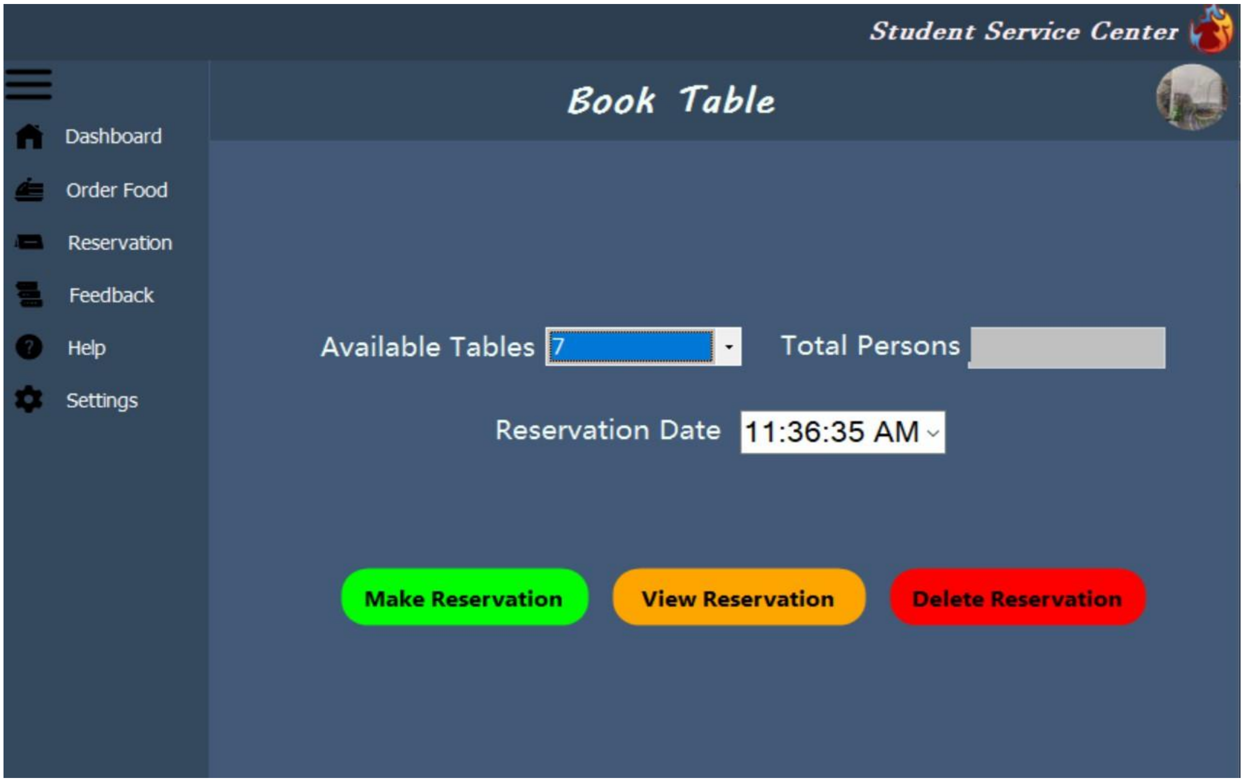
Search

Product andaa

Variant Omlete

ProductName	Quantity
andaa	Omlete
pepsi	1 glass

5.4.4 Book Table:



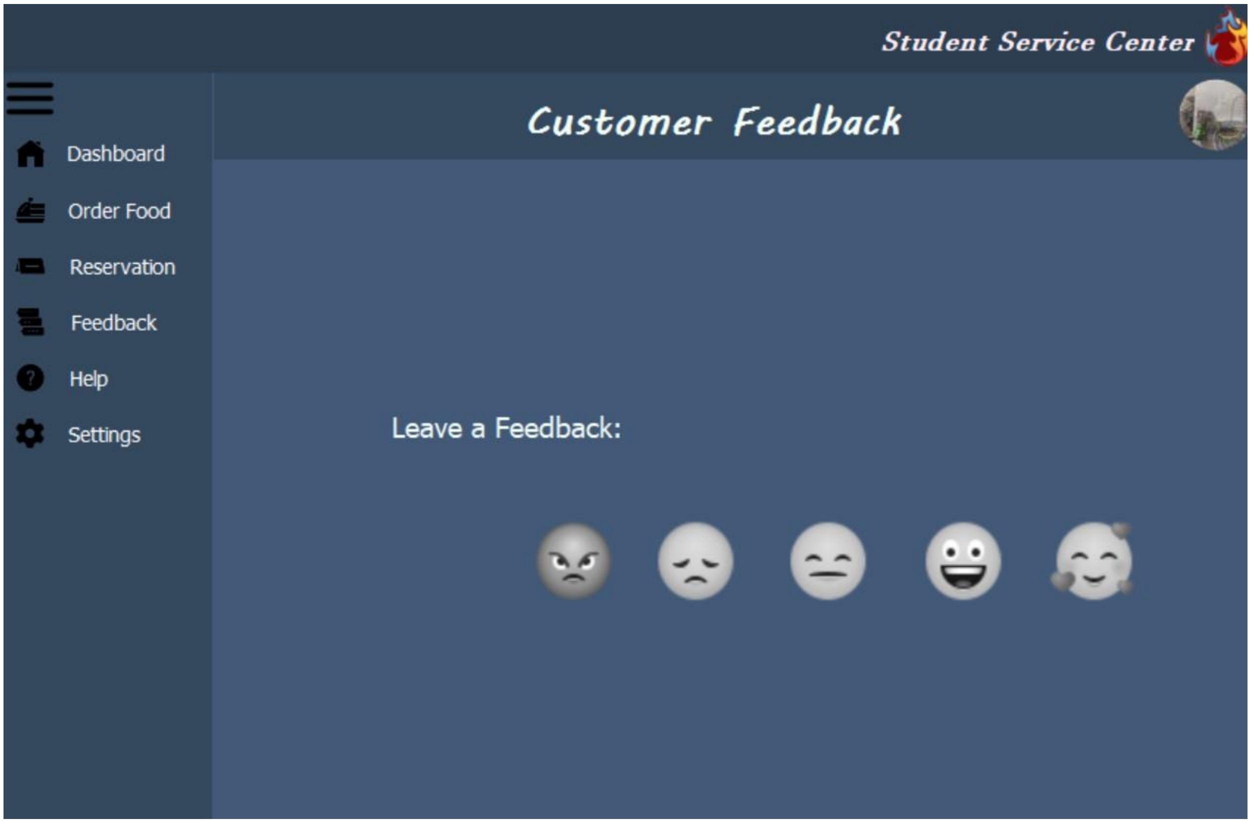
Book Table

Available Tables 7

Total Persons

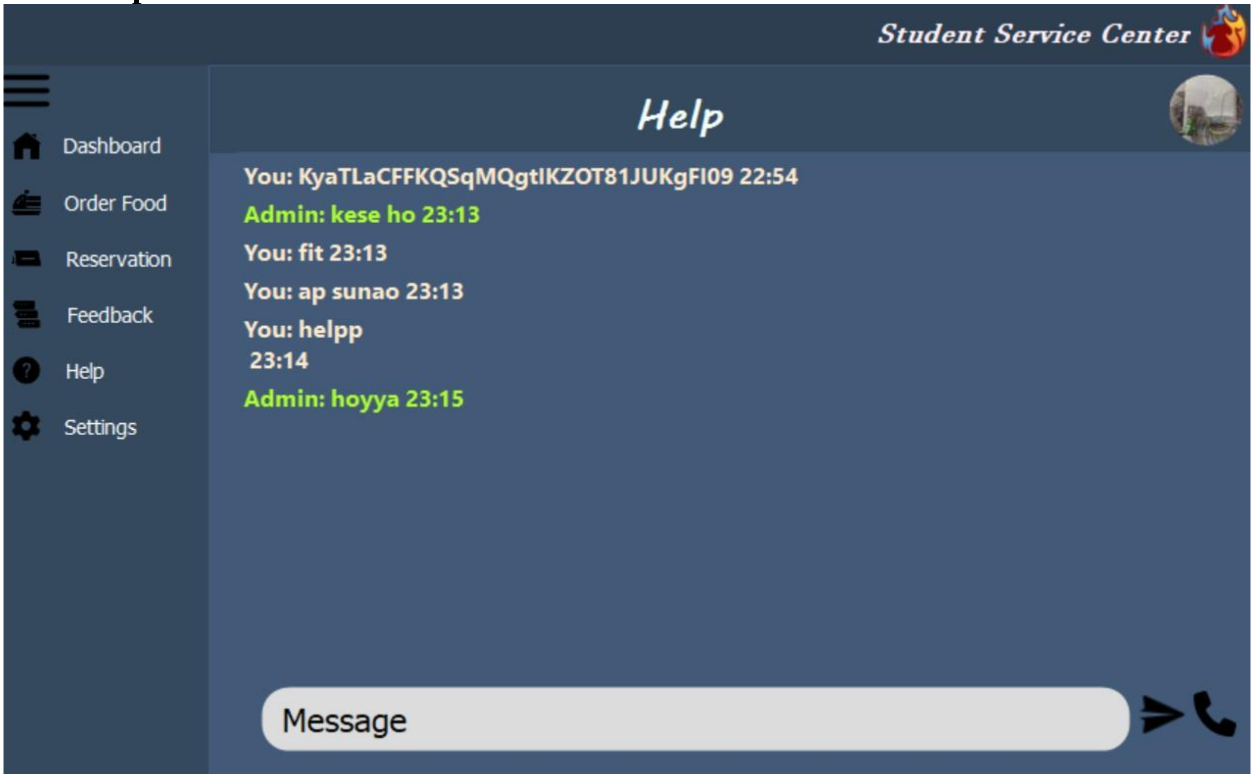
Reservation Date 11:36:35 AM

5.4.5 Feedback:



Leave a Feedback:

5.4.6 Help:



You: KyaTLaCFFKQSqMQgtIKZOT81JUKgFI09 22:54

Admin: kese ho 23:13

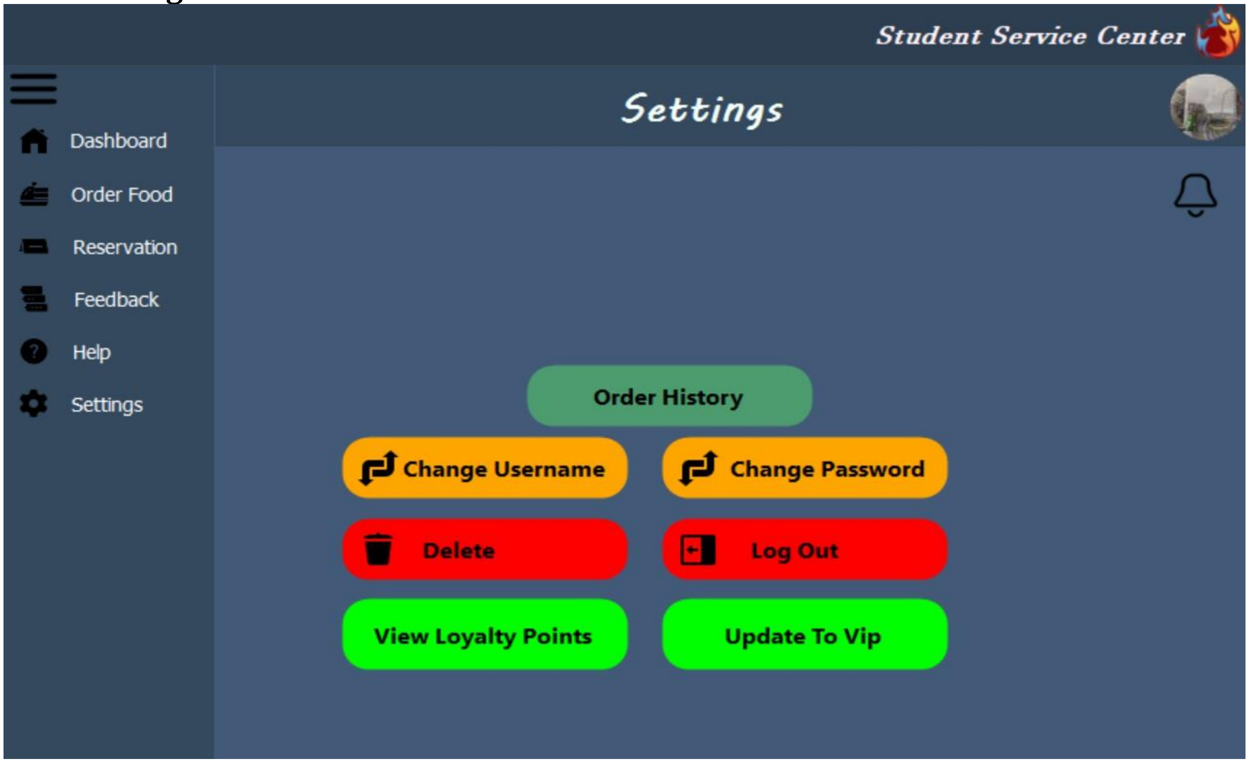
You: fit 23:13

You: ap sunao 23:13

You: helpp
23:14

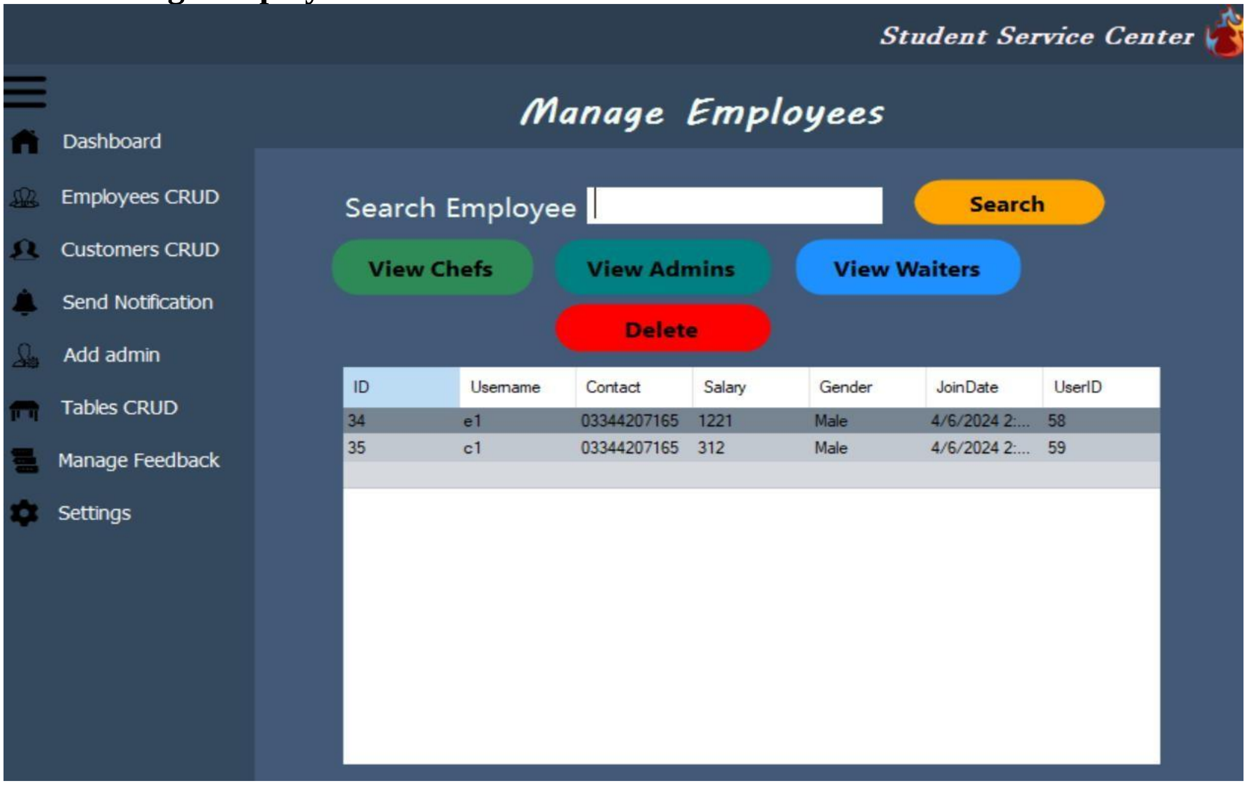
Admin: hoyya 23:15

5.4.7 Settings:



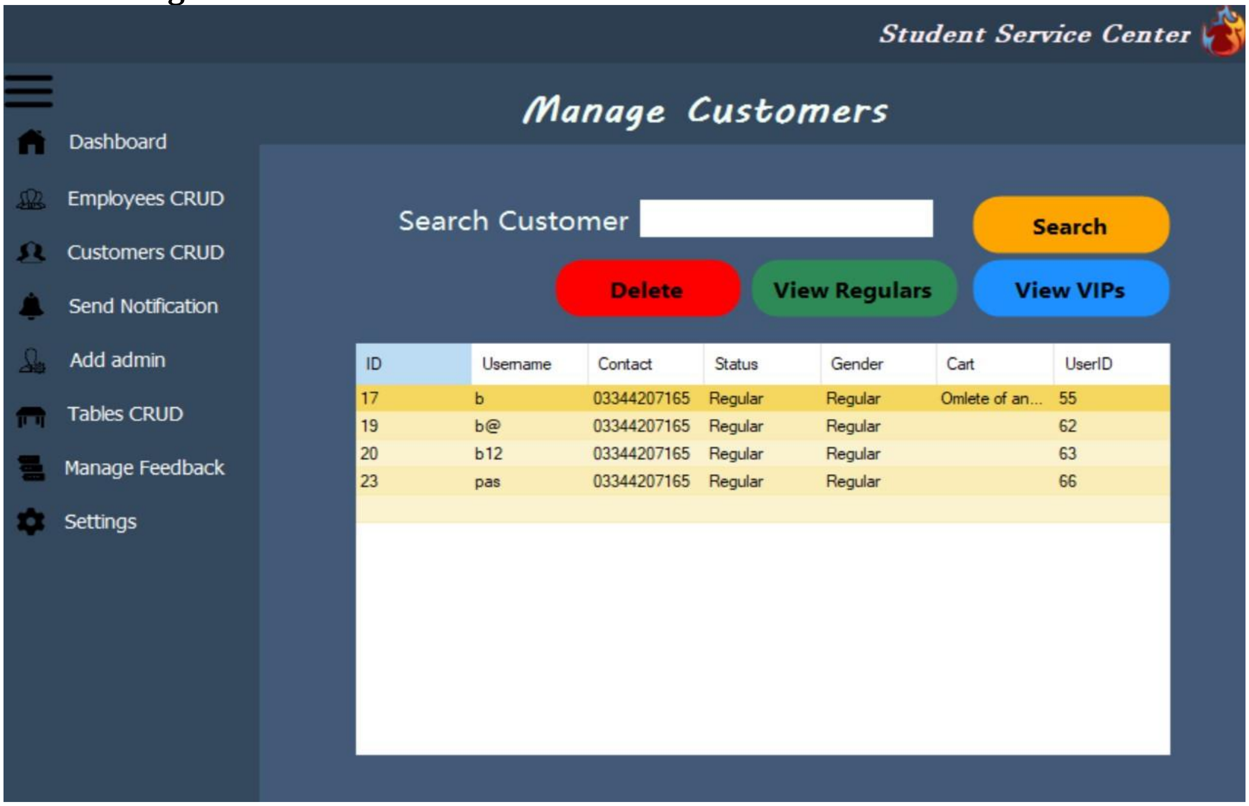
5.5 Admin:

5.5.1 Manage Employees:



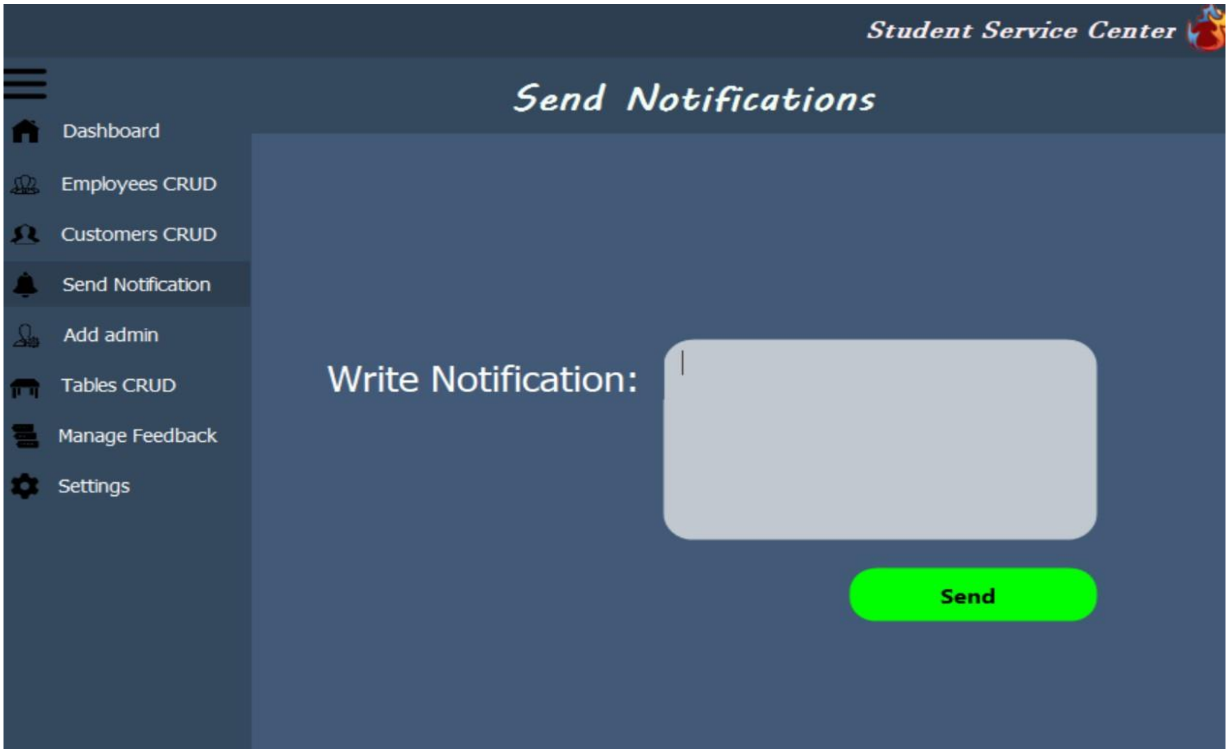
ID	Username	Contact	Salary	Gender	JoinDate	UserID
34	e1	03344207165	1221	Male	4/6/2024 2:...	58
35	c1	03344207165	312	Male	4/6/2024 2:...	59

5.5.2 Manage Customers:

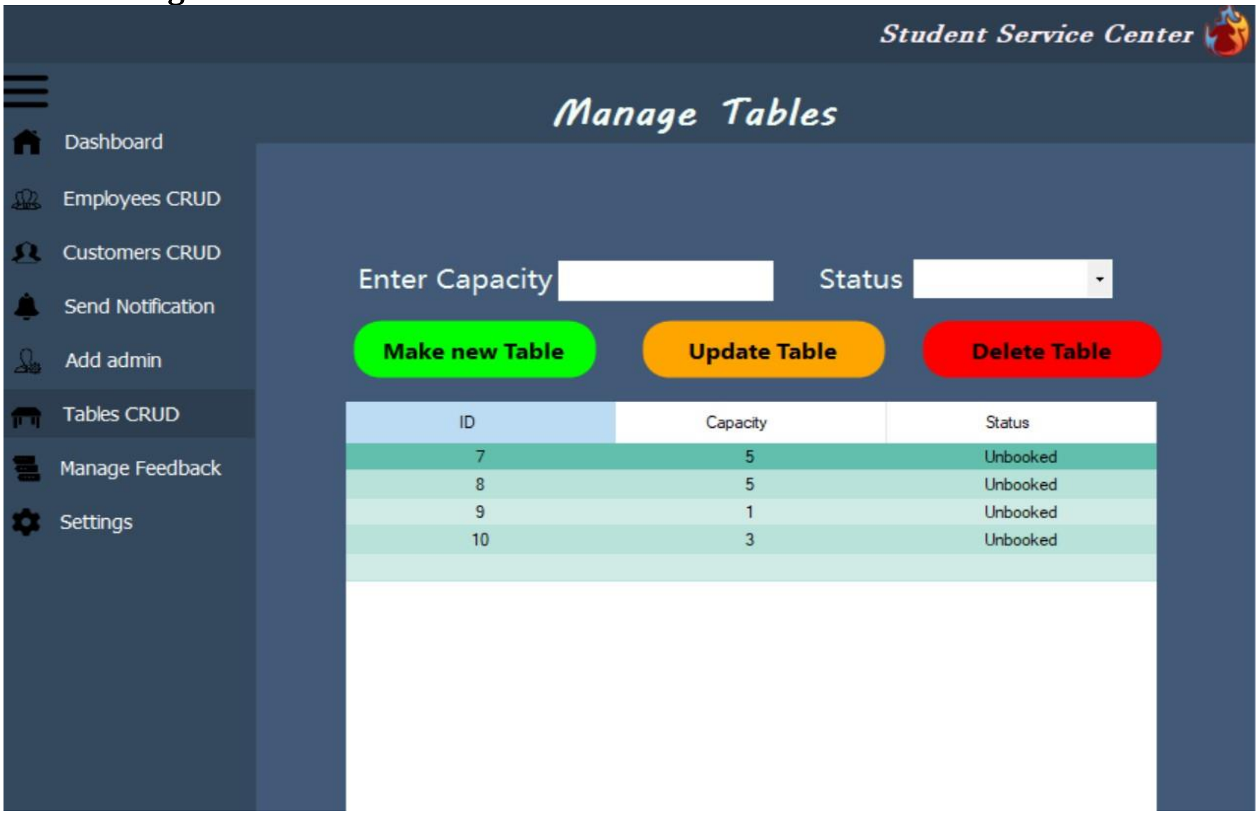


ID	Username	Contact	Status	Gender	Cart	UserID
17	b	03344207165	Regular	Regular	Omlete of an...	55
19	b@	03344207165	Regular	Regular		62
20	b12	03344207165	Regular	Regular		63
23	pas	03344207165	Regular	Regular		66

5.5.3 Send Notifications:

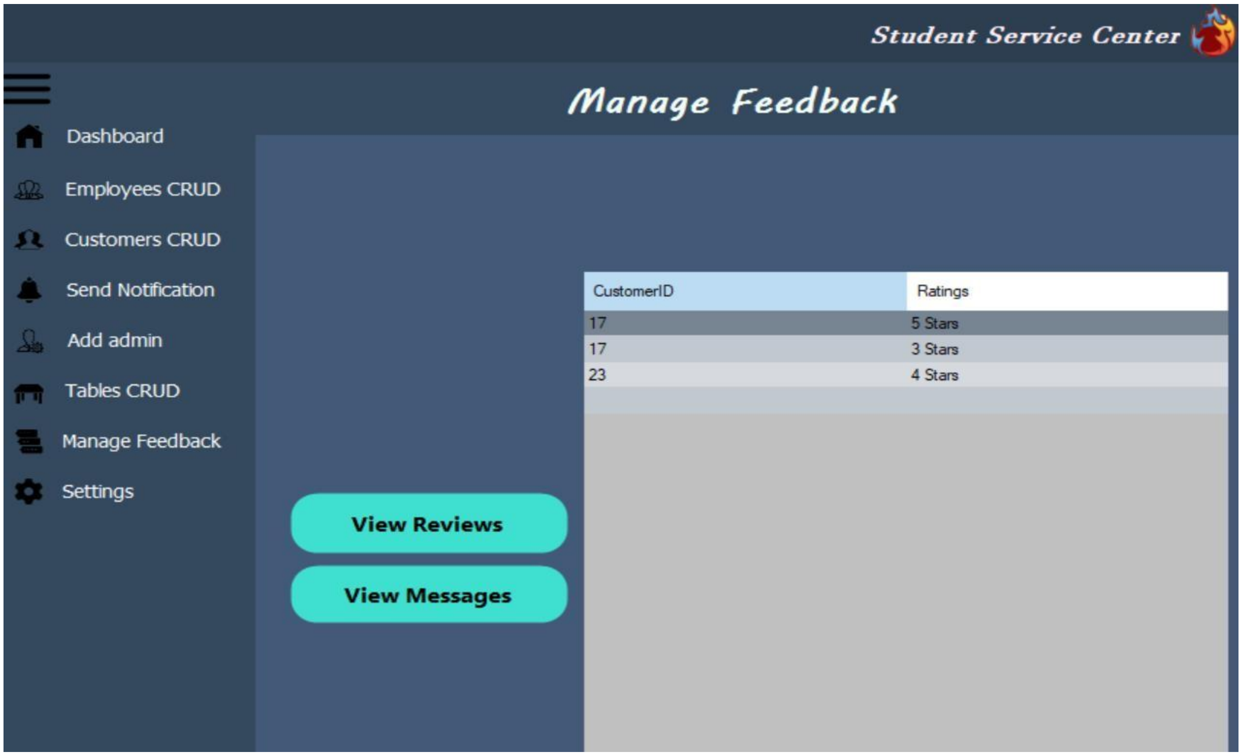


5.5.4 Manage Tables:



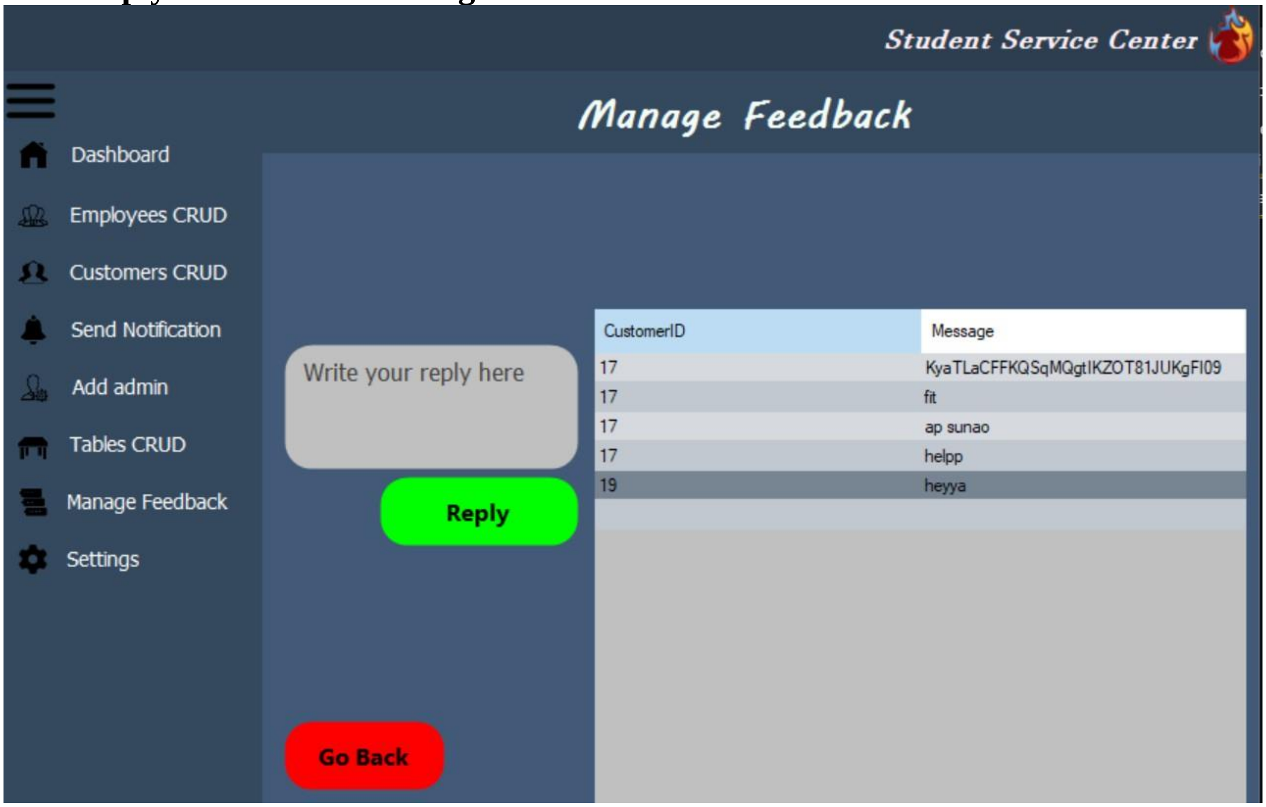
ID	Capacity	Status
7	5	Unbooked
8	5	Unbooked
9	1	Unbooked
10	3	Unbooked

5.5.5 View Feedback:



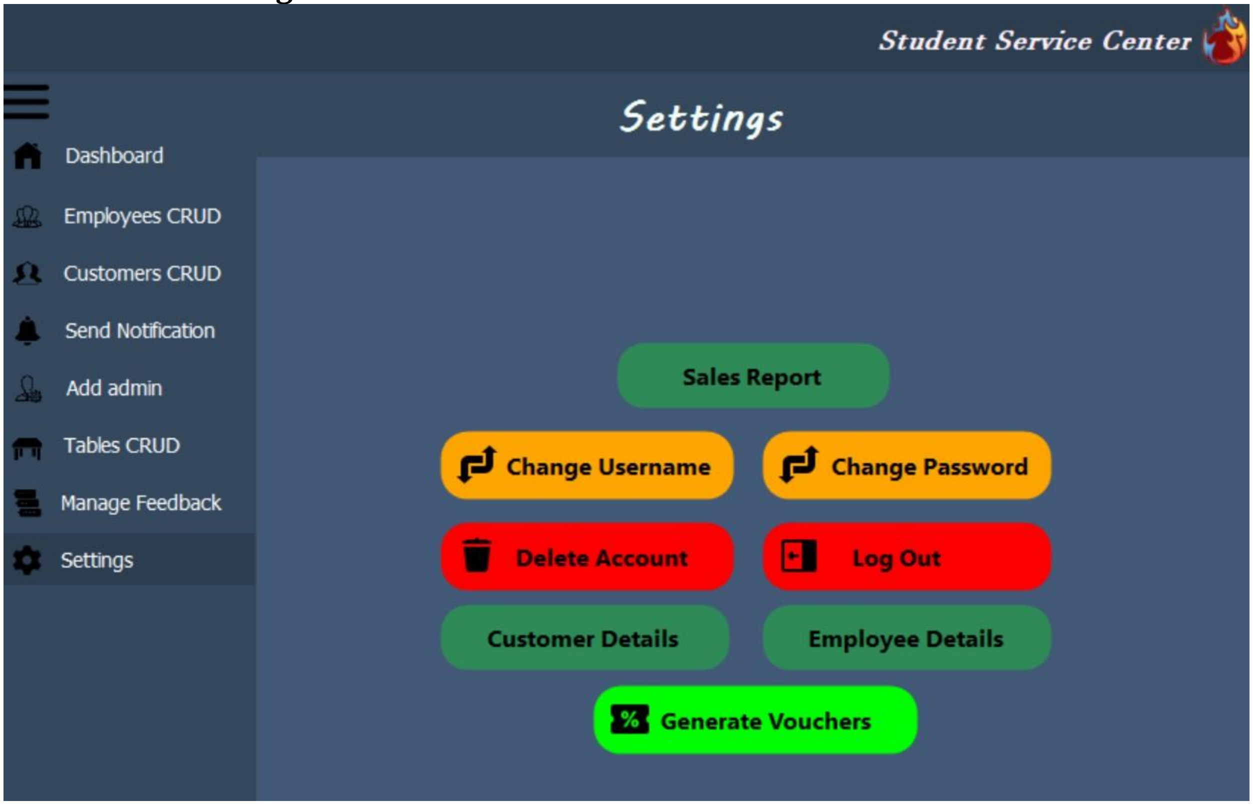
CustomerID	Ratings
17	5 Stars
17	3 Stars
23	4 Stars

5.5.6 Reply to Customer Messages:



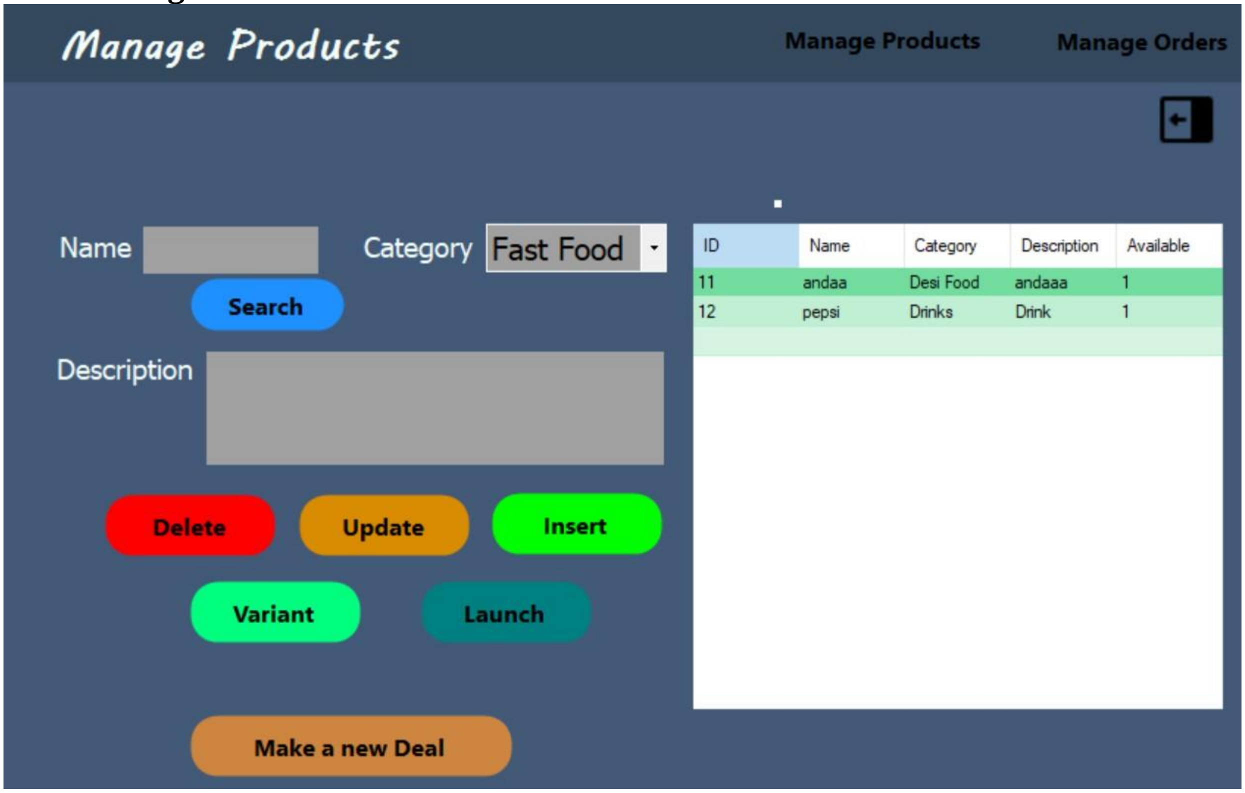
CustomerID	Message
17	KyaTLaCFFKQSqMQgtIKZOT81JUKgFI09
17	fit
17	ap sunao
17	helpp
19	heyya

5.5.7 Admin Settings:



5.6 Chef:

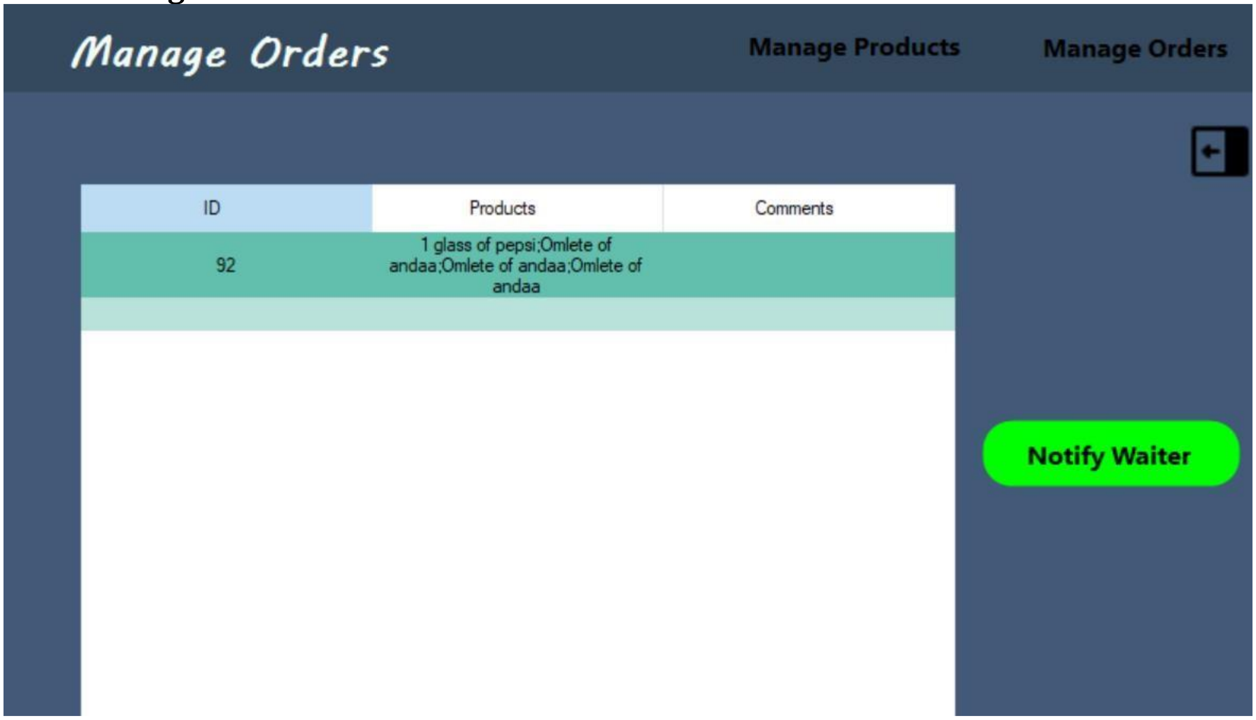
5.6.1 Manage Products:



Category Fast Food

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

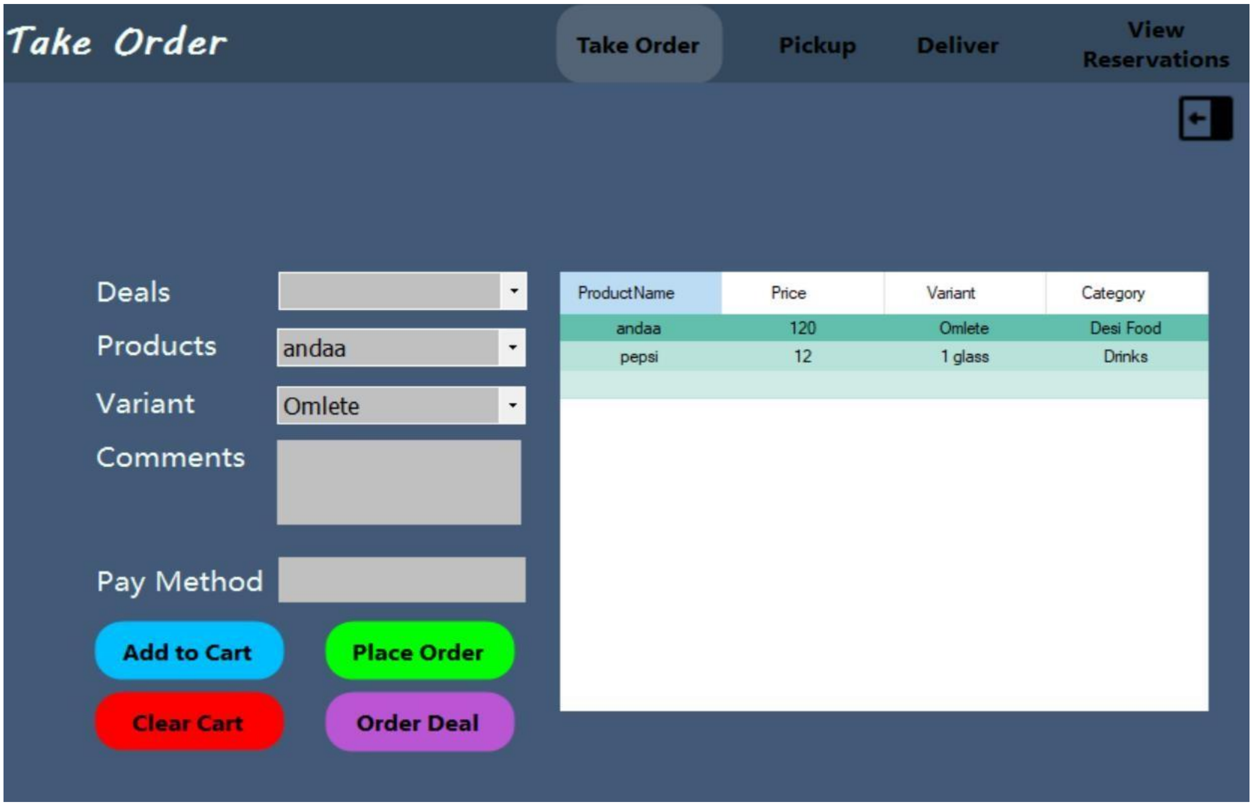
5.6.2 Manage Orders:



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

5.7 Waiter:

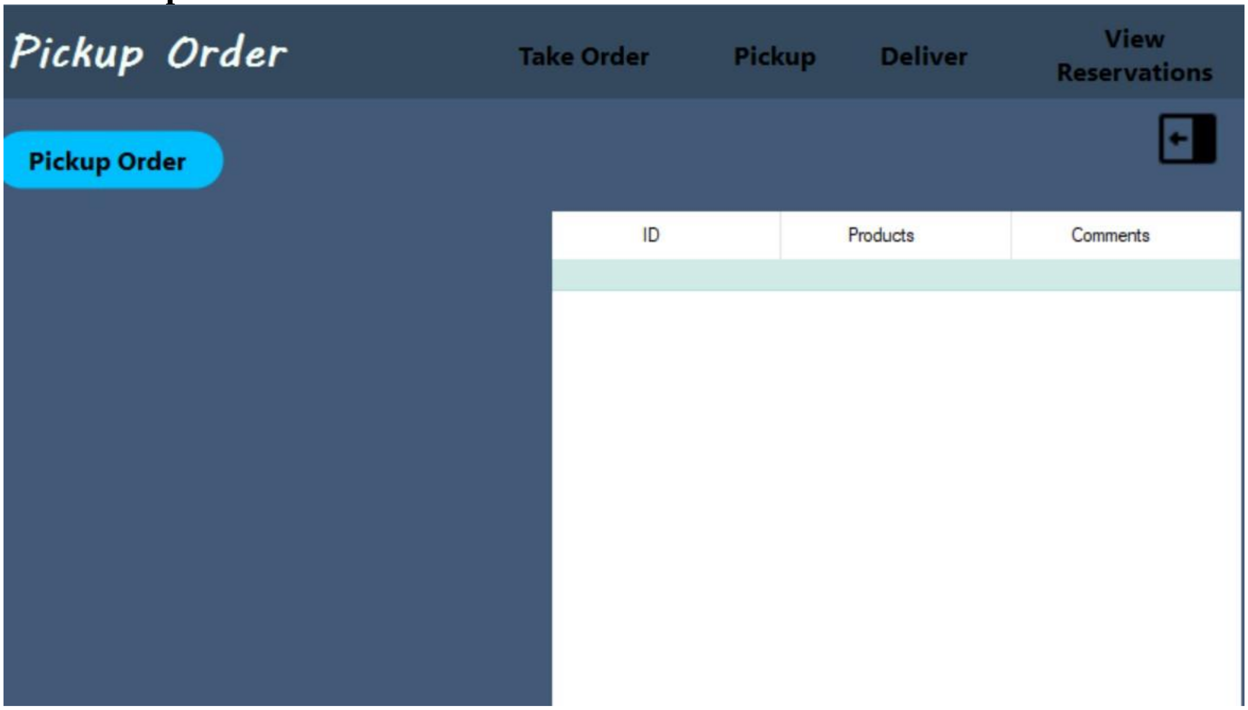
5.7.1: Take Order:



Deals
Products andaa
Variant Omlete
Comments
Pay Method

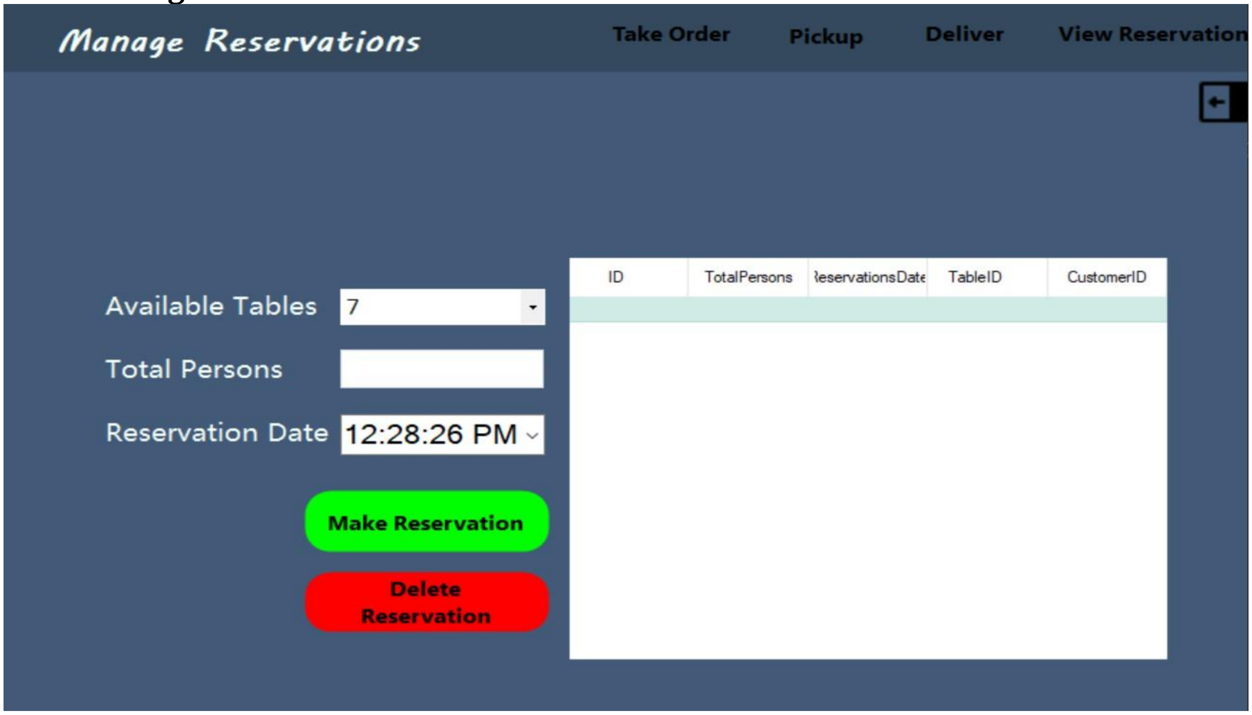
ProductName	Price	Variant	Category
andaa	120	Omlete	Desi Food
pepsi	12	1 glass	Drinks

5.7.2 Pickup and Deliver Order:



ID	Products	Comments

5.7.3 Manage Reservations:



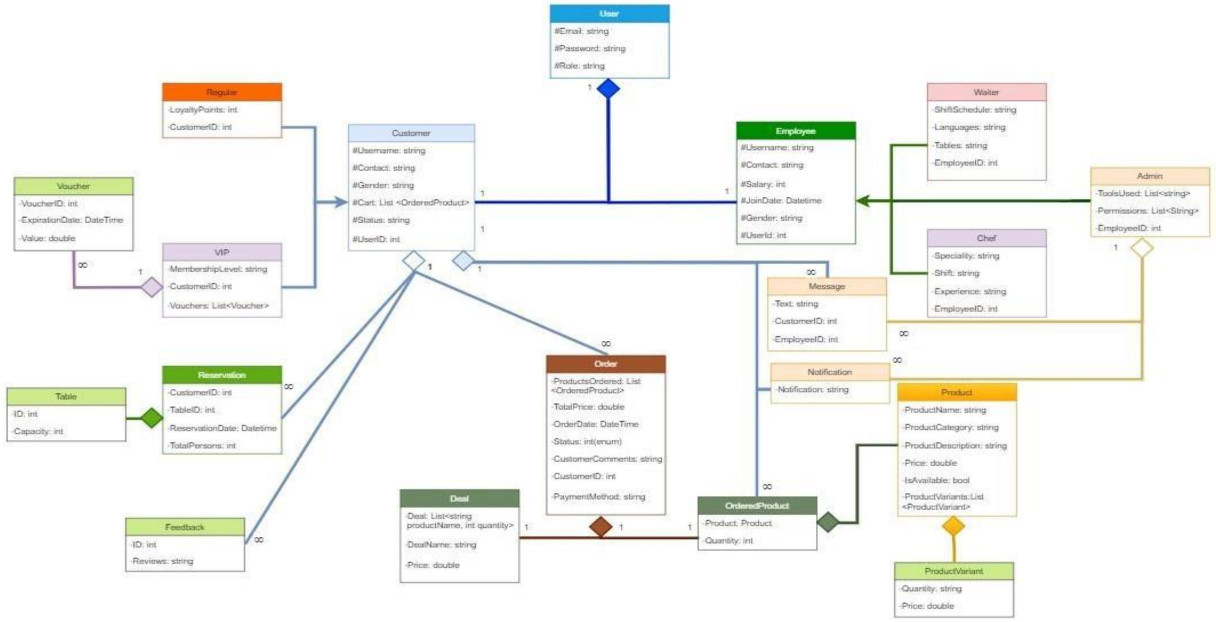
Available Tables 7

Total Persons

Reservation Date 12:28:26 PM

ID	TotalPersons	ReservationsDate	TableID	CustomerID

6. CRC Diagram:



7. Social Media Links:

https://github.com/mb0227/Business-App-Project-2nd-Semester
https://www.linkedin.com/posts/malik-bilal-234937264_im-thrilled-to-share-my-latest-project-activity-7203913414761422849-OdAM?utm_source=share&utm_medium=member_desktop

8. Conclusion:

The Restaurant Management System, developed using C# in both WinForms and console applications, is a robust solution for managing restaurant operations. It integrates Object-Oriented Programming (OOP) principles with database management and file handling to support features such as ordering, reservations, and user management for four distinct roles: administrators, managers, staff, and customers.

The system's WinForms implementation provides a user-friendly graphical interface, while the console version offers a straightforward alternative for quick interactions. Both versions ensure core functionalities are accessible and reliable, catering to different user preferences. By leveraging OOP, the system remains modular, scalable, and easy to maintain, while database integration ensures data integrity and efficiency.

Important Note: Only Customers can sign up from the first Sign up page.
Admins can only be through AddAdmin Form. So. Go to program.cs and replace Application.Run(new Homepage()); with Application.Run(new AddAdmin());
Fill in admin's info and click on add employee button. You can directly use queries too.
Now, go to program.cs and undo that change and you should be good to go.

Also, only and only Admin can add any other employee, be it chef or waiter by.
Admin must login in and go to Employees CRUD option and add employee.

Another important note is that if you want to use Forget Password Functionality, you have to go to BusinessApp. There will be Folder named Credentials, open it then the file in it. You have to paste your **encrypted** Email and App Password there. Here, you can copy the Encryption File Code in RMS and run it separately in a new file and give your email and password one by one. Copy the result and store them in comma separated style i.e. email, password.

Now you are good to go! So Have Fun.