A PROJECT REPORT
ON

RESAURANT REVIEW SYSTEM

BY

Smit bhoraniya (CE017) 19CEUOS057
Het Desai (CE001) 19CEUON074

Bachelors of Technology
Semester VI

Subject: (CE – 619) Service Oriented Computing
Month-Year: March - 2022

Guided by:

Prof. Prashant M Jadav

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical/term work carried out in the subject of Software Engineering and recorded in the journal is the bonafide work of

Smit bhoraniya (CE017) 19CEUOS057

Het Desai (CE001) 19CEUON074

Of
B. Tech Semester VI in branch of
Computer Engineering
during the academic year 2021-2022

Prof. Prashant M. Jadav
Assistant Professor,
Dept. Of Computer Engg.
Faculty of technology
Dharmsinh Desai University,
Nadiad

Dr. C. K. Bhensdadia
Head,
Dept. of Computer Engg.
Faculty of technology
Dharmsinh Desai University,
Nadiad

Table of content:

1. Abstract and Introduction:

Abstract:

Restaurants nowadays prefer taking online orders. It not only helps in getting effective customer feedback but also useful for managing orders easily. We are moving towards an automated and digital world. Having a significant online presence is necessary for any restaurant to be successful and prosperous. Getting customer feedback and analyzing them in an effective manner makes the difference.

Information:

Restaurant review management system is designed to take reviews for a particular restaurant. In this app we have implemented functionalitieslike add restaurant, add review forms, view responses.

Technologies/Tools Used:

Platform/Tools used: Visual Studio 2019, Microsoft SQL Server Management Studio 18
Technology used: ASP.NET framework
Language: C#

2. Software Requirement Specification (SRS):

1. Account Management:

R.1.1 : User Account

R.1.1.1 : Login for User
Description: Enter credentials for login purpose.
I/p: Email and Password
O/p: Logged In

2. Restaurant Management:

R.2.1 : Add Restaurant.
Description: Add restaurants.
I/p: Enter Restaurant.
O/p: Restaurant will be added to the Website.

R.2.2 : Edit Restaurant.
Description: Edit restaurants
I/p: Edit Restaurant.
O/p: Restaurant will be Updated

R.2.3 : Remove Restaurant
Description: Remove Restaurant
O/p: Restaurant details will be Removed.

R.2.4 : Restaurant List
Descripation: Show all Restaurant
O/P: Show all Restaurant

3. Restaurant Review Management:

R.3.1. Add Review
Description: Add Review
I/P: Add Review Details
O/P: Review will be added to the Website

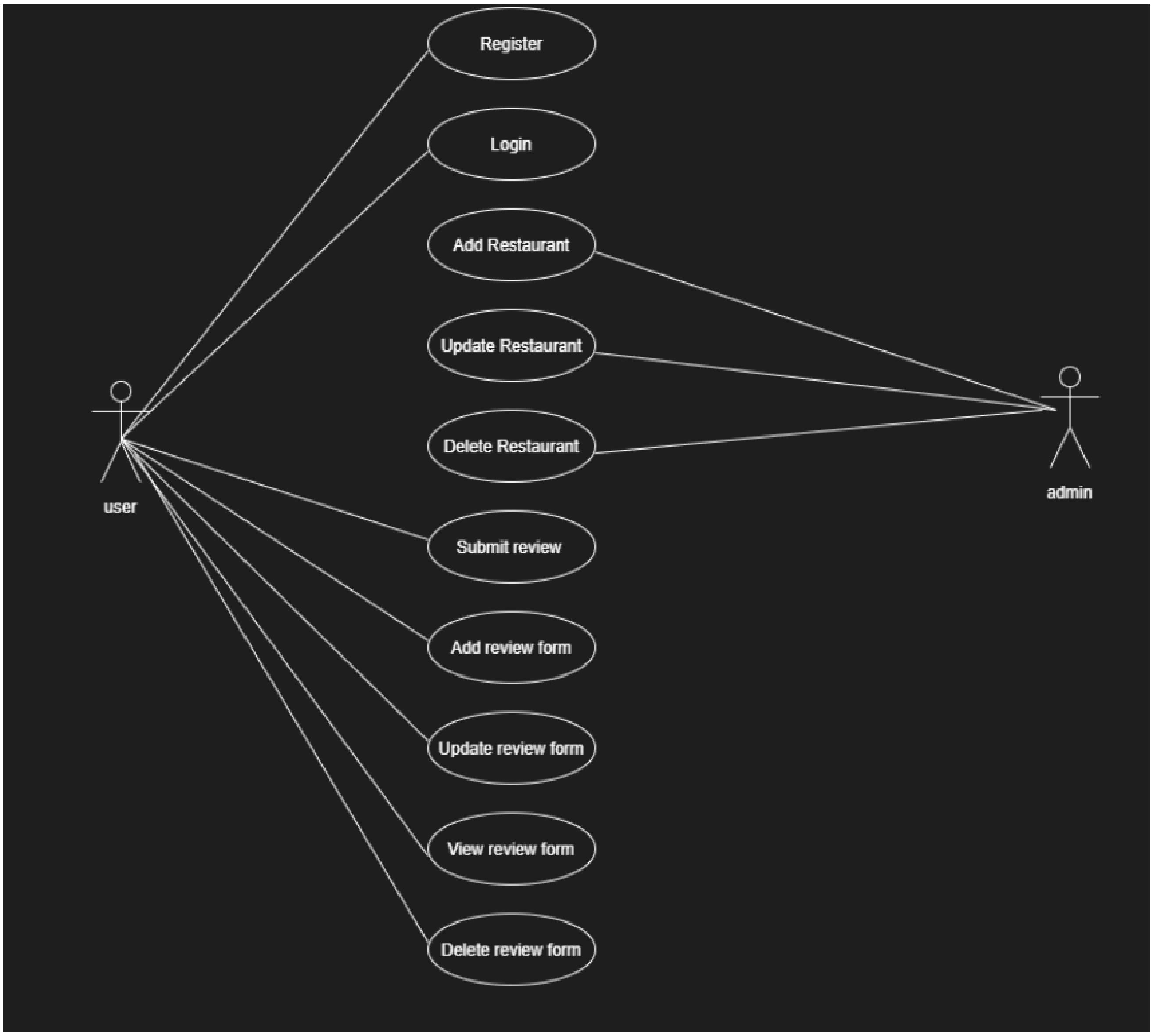
R.3.2. Update Review
Description: Update Review
I/P: Update Review Details
O/P: Review will be updated to the Website

R.3.3. Remove Review
Description: Remove Review
O/P: Review will be removed

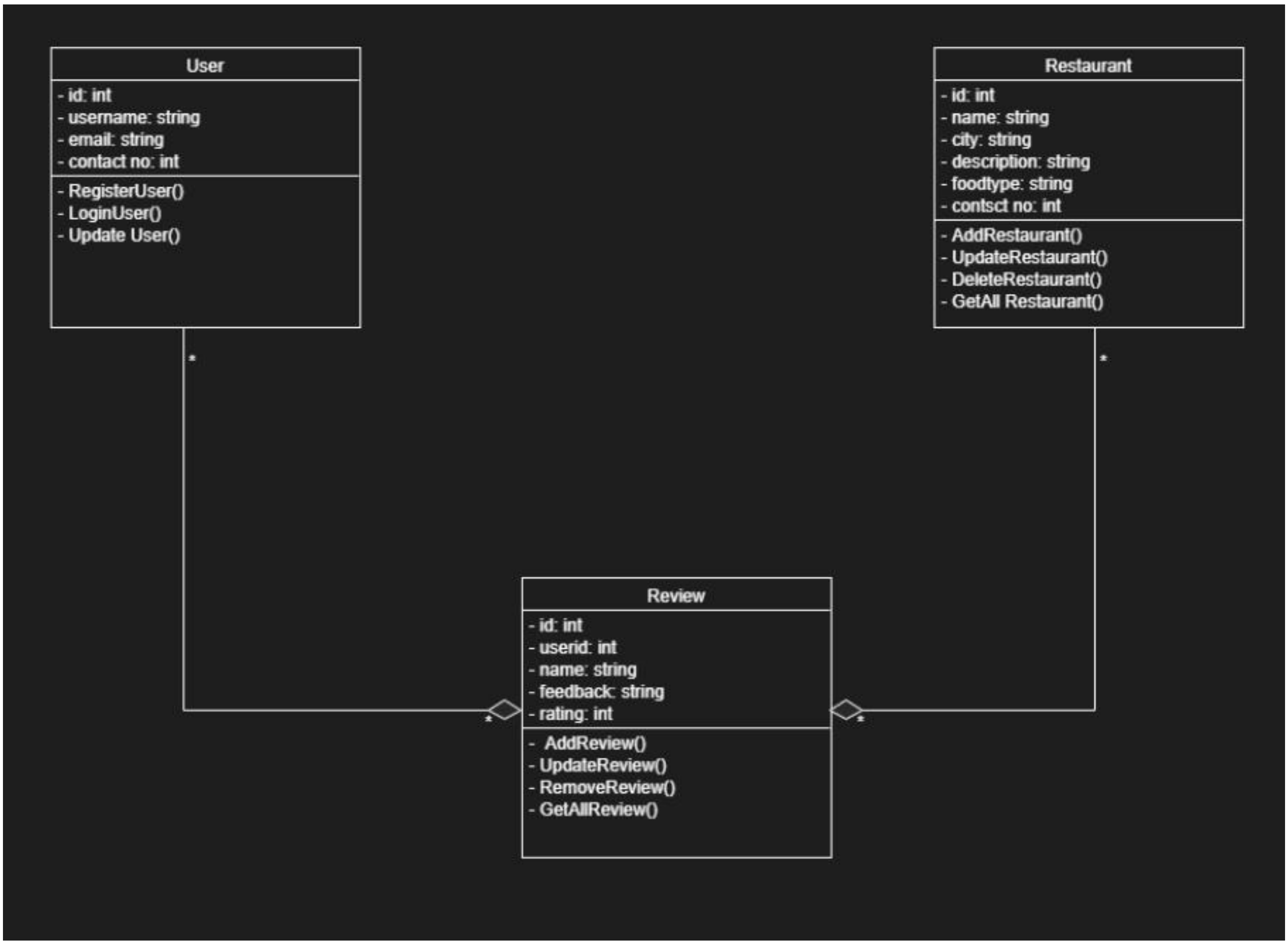
R.3.4. Review List
Description: Review List
O/P: Show all Review

4. Design Documents:

USE CASE DIAGRAM:



CLASS DIAGRAM:



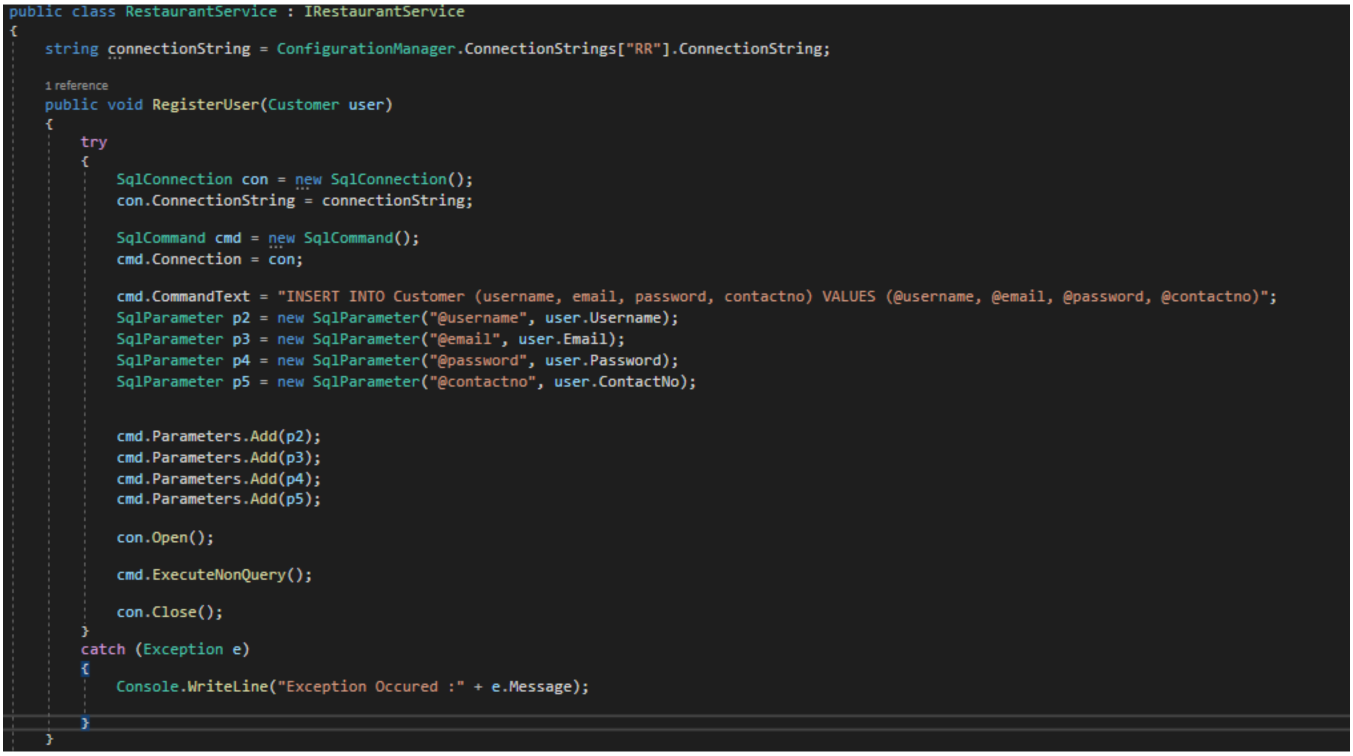
5. <u>Implementation Details:</u>

Modules:

1. **User Module:** In this module we have Email address, username, contact no and Password with which person can Login into our application.

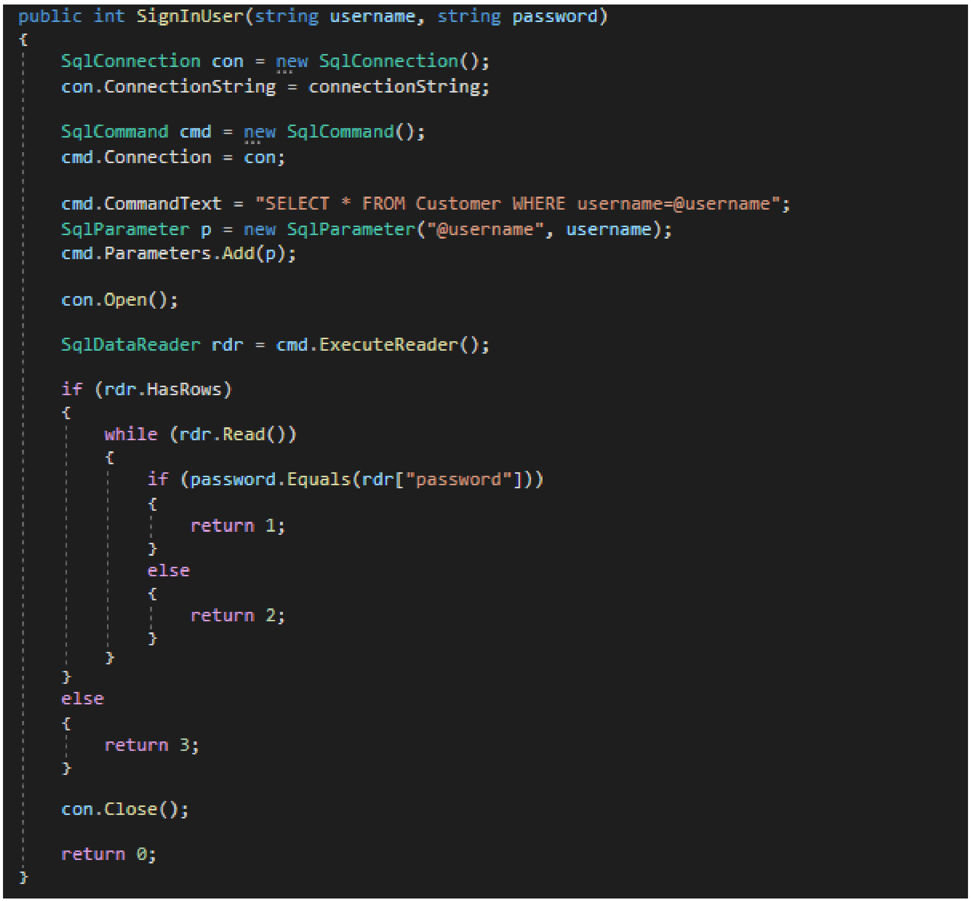
   Test: In this module the user can login.

Register User:



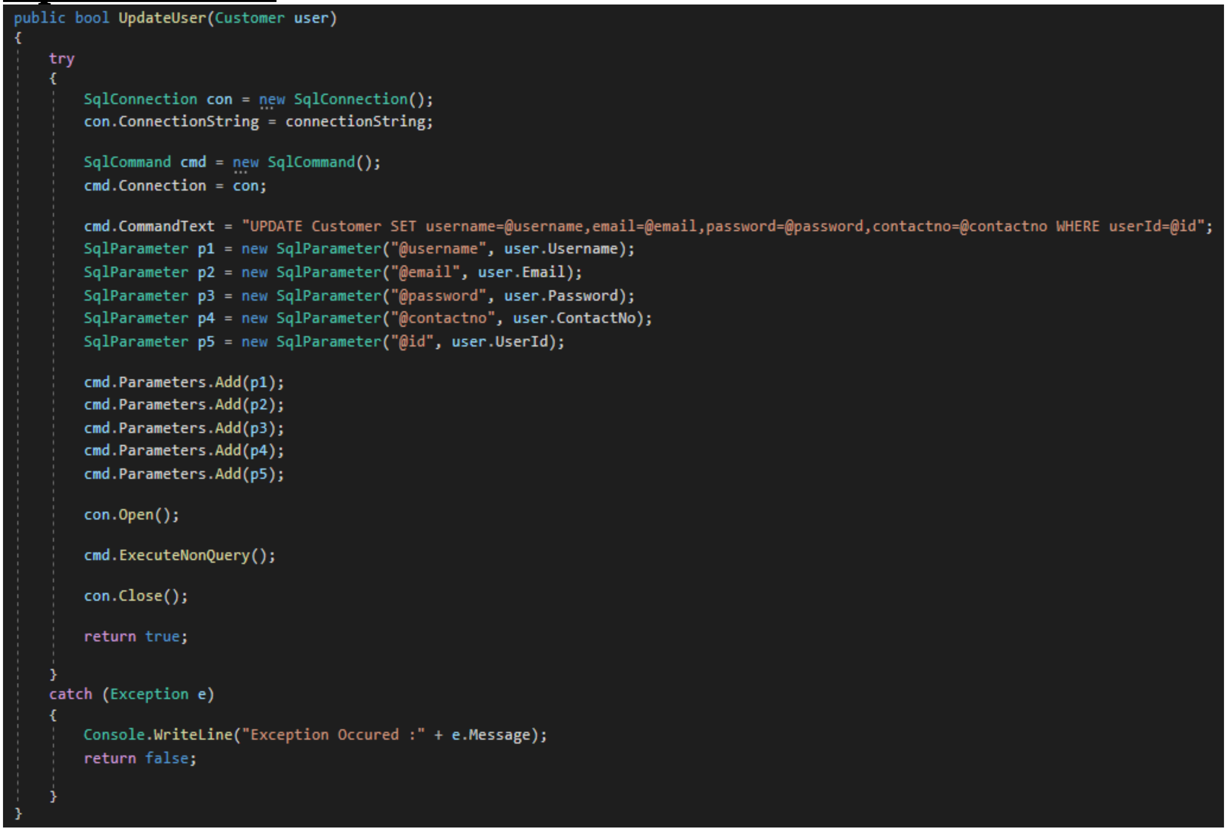
```
public class RestaurantService : IRestaurantService
{
    string connectionString = ConfigurationManager.ConnectionStrings["RR"].ConnectionString;

    1 reference
    public void RegisterUser(Customer user)
    {
        try
        {
            SqlConnection con = new SqlConnection();
            con.ConnectionString = connectionString;

            SqlCommand cmd = new SqlCommand();
            cmd.Connection = con;

            cmd.CommandText = "INSERT INTO Customer (username, email, password, contactno) VALUES (@username, @email, @password, @contactno)";
            SqlParameter p2 = new SqlParameter("@username", user.Username);
            SqlParameter p3 = new SqlParameter("@email", user.Email);
            SqlParameter p4 = new SqlParameter("@password", user.Password);
            SqlParameter p5 = new SqlParameter("@contactno", user.ContactNo);


            cmd.Parameters.Add(p2);
            cmd.Parameters.Add(p3);
            cmd.Parameters.Add(p4);
            cmd.Parameters.Add(p5);

            con.Open();

            cmd.ExecuteNonQuery();

            con.Close();
        }
        catch (Exception e)
        {
            Console.WriteLine("Exception Occured :" + e.Message);

        }
    }
```

Login User:



```
public int SignInUser(string username, string password)
{
    SqlConnection con = new SqlConnection();
    con.ConnectionString = connectionString;

    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;

    cmd.CommandText = "SELECT * FROM Customer WHERE username=@username";
    SqlParameter p = new SqlParameter("@username", username);
    cmd.Parameters.Add(p);

    con.Open();

    SqlDataReader rdr = cmd.ExecuteReader();

    if (rdr.HasRows)
    {
        while (rdr.Read())
        {
            if (password.Equals(rdr["password"]))
            {
                return 1;
            }
            else
            {
                return 2;
            }
        }
    }
    else
    {
        return 3;
    }

    con.Close();

    return 0;
}
```

Update User:



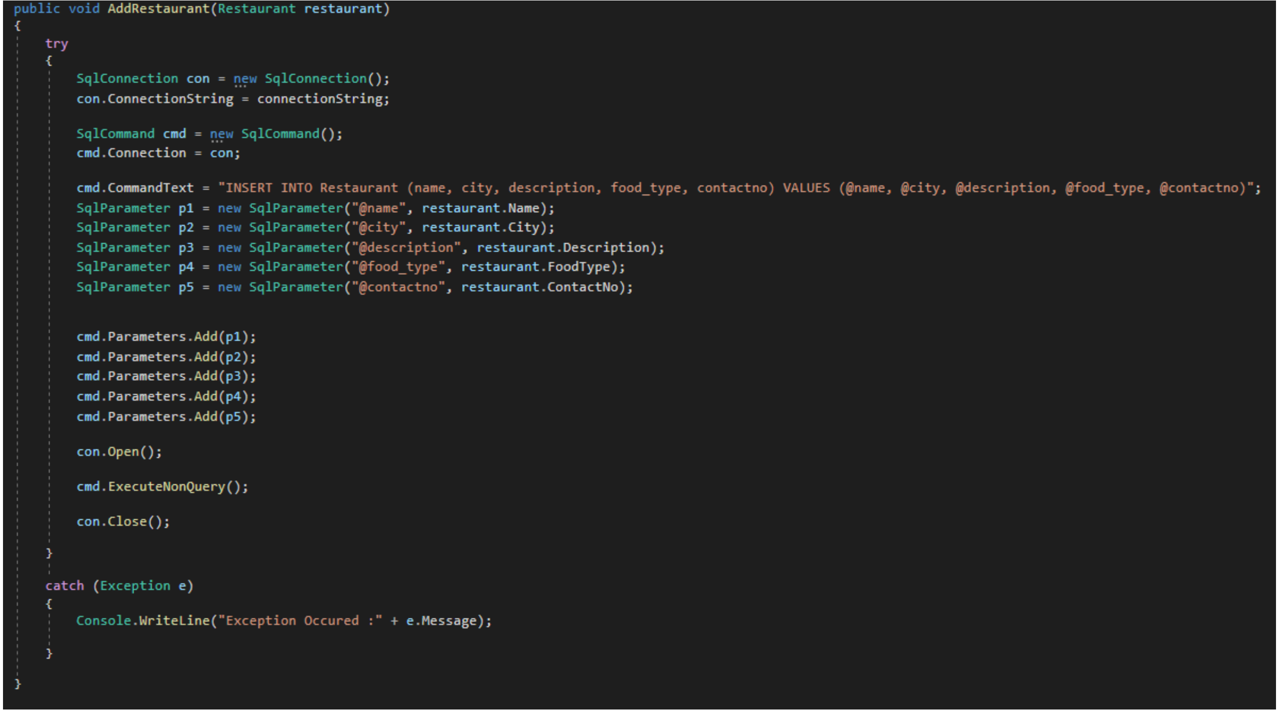
```
public bool UpdateUser(Customer user)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "UPDATE Customer SET username=@username,email=@email,password=@password,contactno=@contactno WHERE userId=@id";
        SqlParameter p1 = new SqlParameter("@username", user.Username);
        SqlParameter p2 = new SqlParameter("@email", user.Email);
        SqlParameter p3 = new SqlParameter("@password", user.Password);
        SqlParameter p4 = new SqlParameter("@contactno", user.ContactNo);
        SqlParameter p5 = new SqlParameter("@id", user.UserId);

        cmd.Parameters.Add(p1);
        cmd.Parameters.Add(p2);
        cmd.Parameters.Add(p3);
        cmd.Parameters.Add(p4);
        cmd.Parameters.Add(p5);

        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();

        return true;

    }
    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);
        return false;

    }
}
```

2. Restaurant Module:

In This Module we have Restaurant details, we can add, edit, remove the Restaurants.

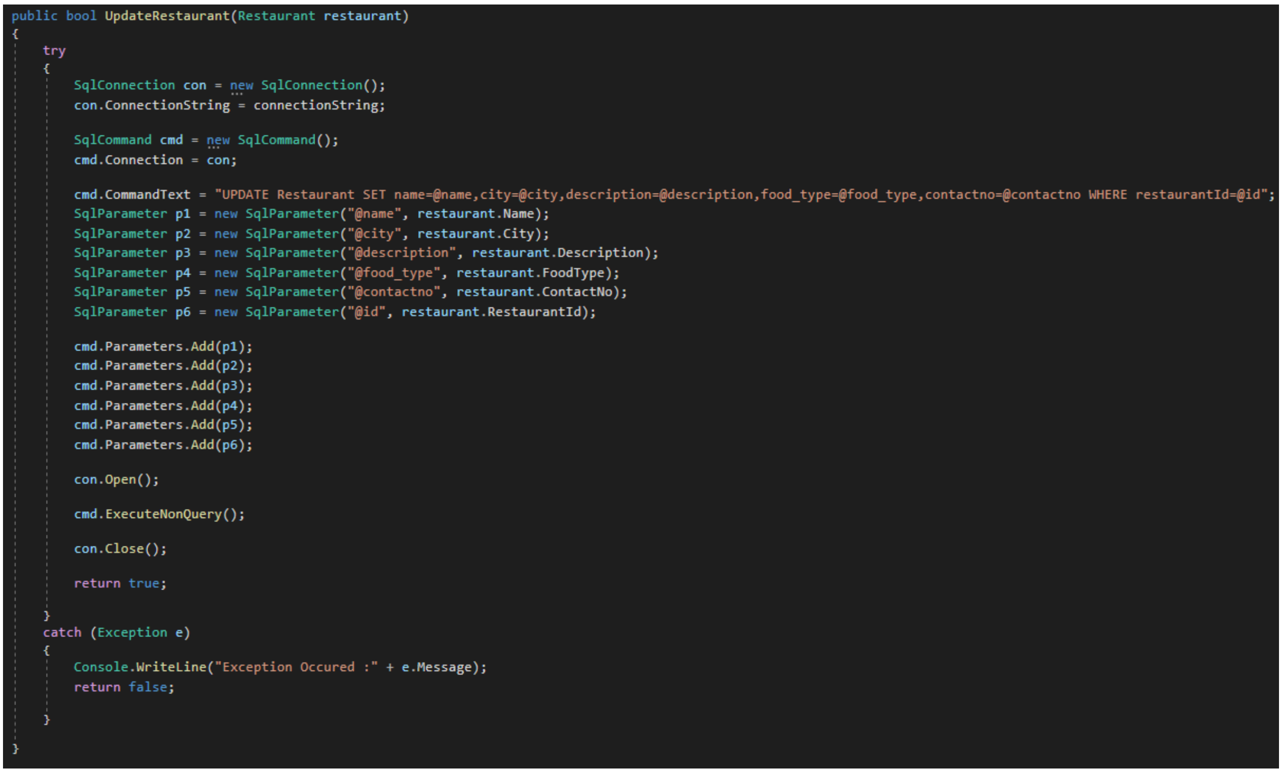
Test: in this admin can manage the restaurant

Add Restaurant:



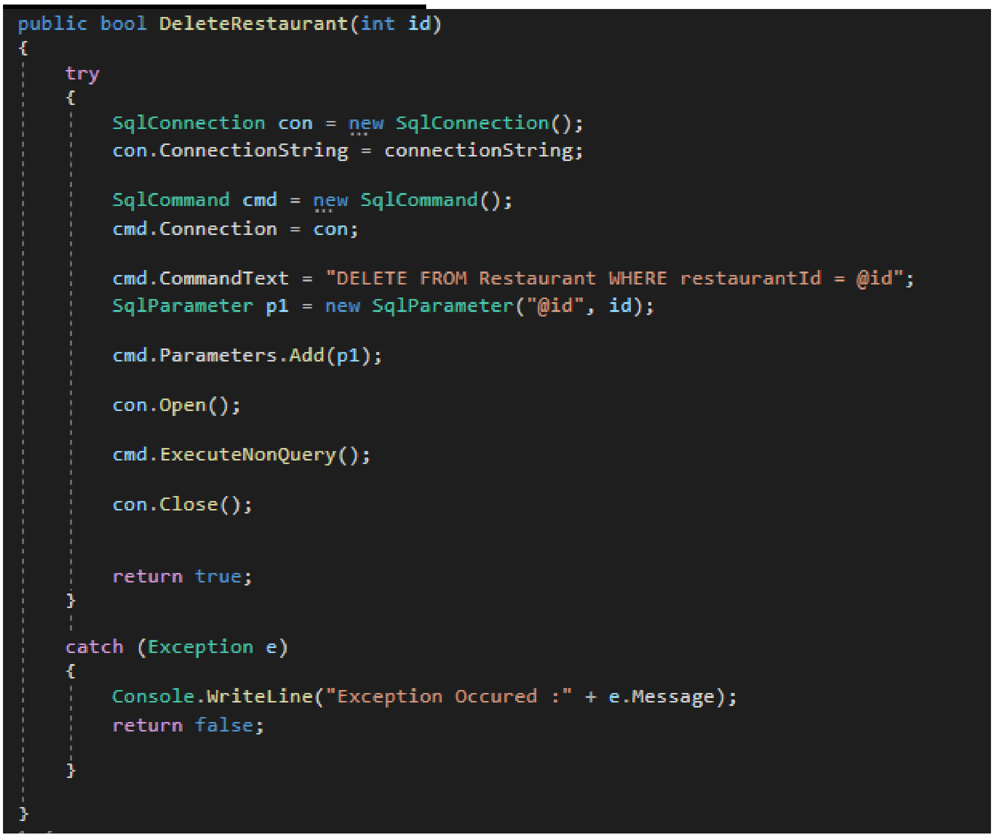
```
public void AddRestaurant(Restaurant restaurant)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "INSERT INTO Restaurant (name, city, description, food_type, contactno) VALUES (@name, @city, @description, @food_type, @contactno)";
        SqlParameter p1 = new SqlParameter("@name", restaurant.Name);
        SqlParameter p2 = new SqlParameter("@city", restaurant.City);
        SqlParameter p3 = new SqlParameter("@description", restaurant.Description);
        SqlParameter p4 = new SqlParameter("@food_type", restaurant.FoodType);
        SqlParameter p5 = new SqlParameter("@contactno", restaurant.ContactNo);


        cmd.Parameters.Add(p1);
        cmd.Parameters.Add(p2);
        cmd.Parameters.Add(p3);
        cmd.Parameters.Add(p4);
        cmd.Parameters.Add(p5);

        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();

    }

    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);

    }
}
```

Update Restaurant:



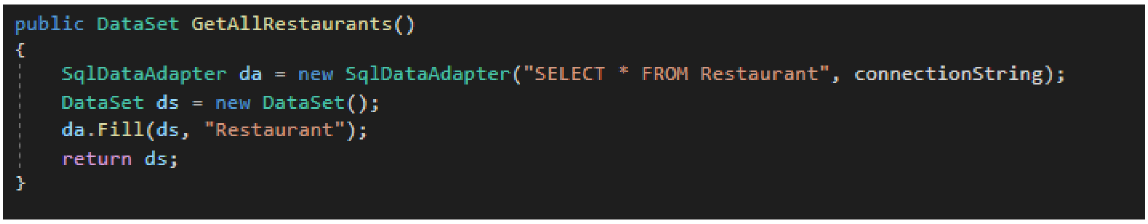
```
public bool UpdateRestaurant(Restaurant restaurant)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "UPDATE Restaurant SET name=@name,city=@city,description=@description,food_type=@food_type,contactno=@contactno WHERE restaurantId=@id";
        SqlParameter p1 = new SqlParameter("@name", restaurant.Name);
        SqlParameter p2 = new SqlParameter("@city", restaurant.City);
        SqlParameter p3 = new SqlParameter("@description", restaurant.Description);
        SqlParameter p4 = new SqlParameter("@food_type", restaurant.FoodType);
        SqlParameter p5 = new SqlParameter("@contactno", restaurant.ContactNo);
        SqlParameter p6 = new SqlParameter("@id", restaurant.RestaurantId);

        cmd.Parameters.Add(p1);
        cmd.Parameters.Add(p2);
        cmd.Parameters.Add(p3);
        cmd.Parameters.Add(p4);
        cmd.Parameters.Add(p5);
        cmd.Parameters.Add(p6);

        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();

        return true;

    }
    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);
        return false;

    }
}
```

Remove Restaurant:



```
public bool DeleteRestaurant(int id)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "DELETE FROM Restaurant WHERE restaurantId = @id";
        SqlParameter p1 = new SqlParameter("@id", id);

        cmd.Parameters.Add(p1);

        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();


        return true;
    }

    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);
        return false;

    }

}
```

RestaurantList:



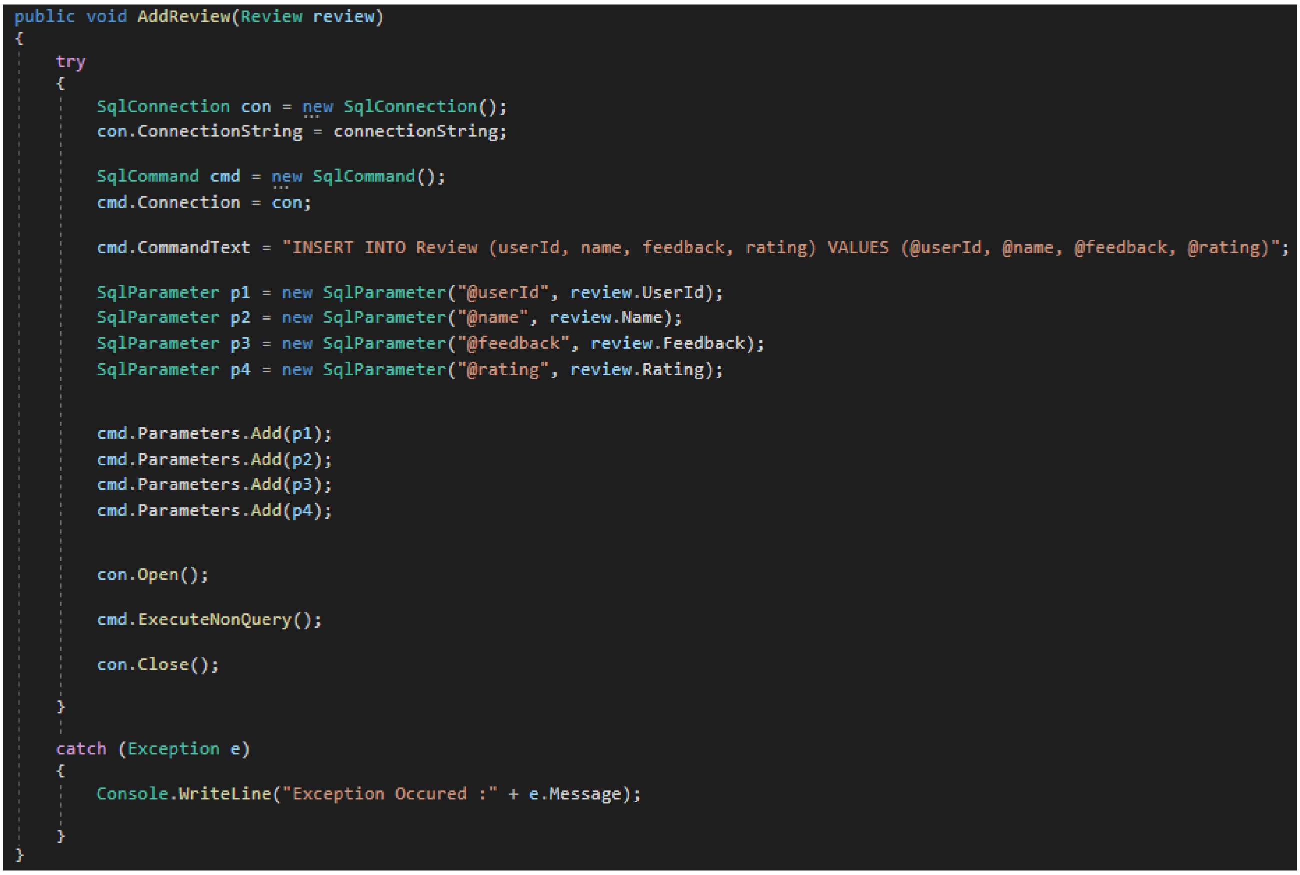
```
public DataSet GetAllRestaurants()
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Restaurant", connectionString);
    DataSet ds = new DataSet();
    da.Fill(ds, "Restaurant");
    return ds;
}
```

3. Review Module:

In This Module we have Reviews, we can add, edit, remove the Review.

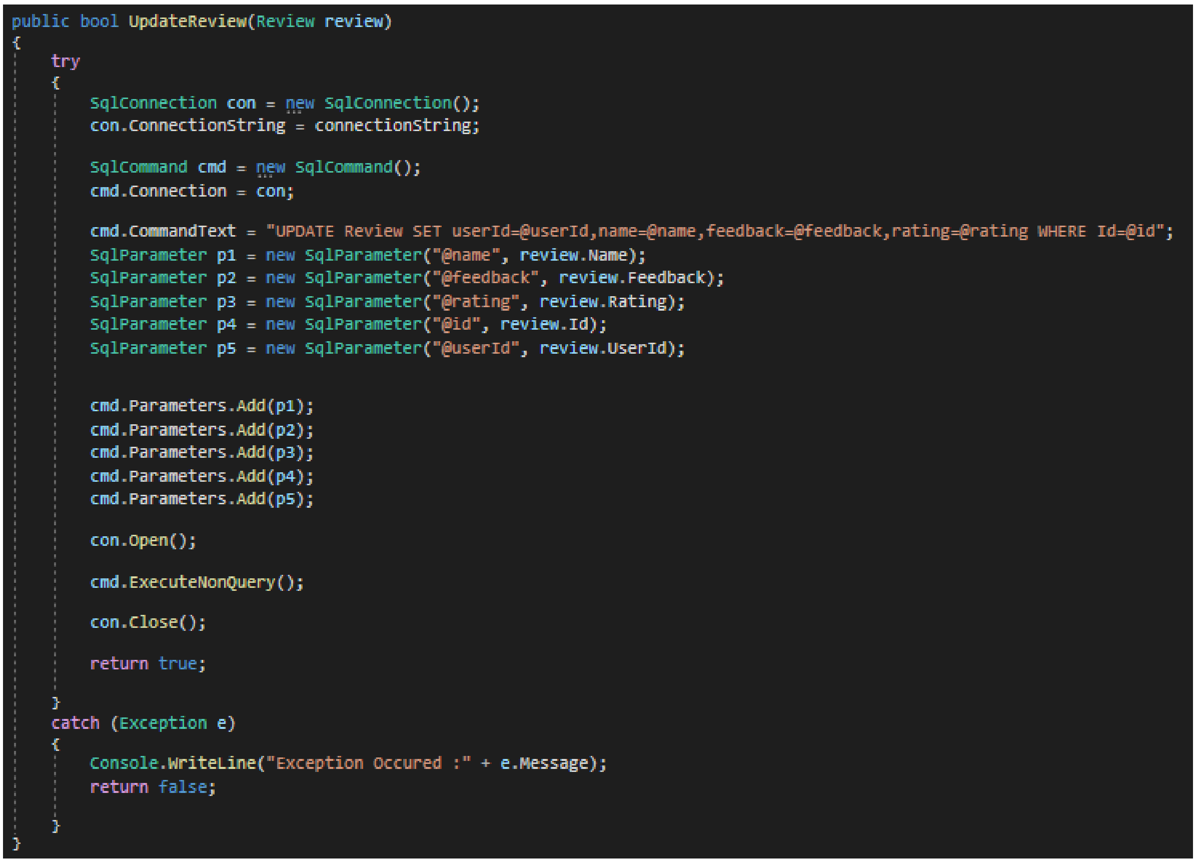
Test: in this user can give Review

Add Review:



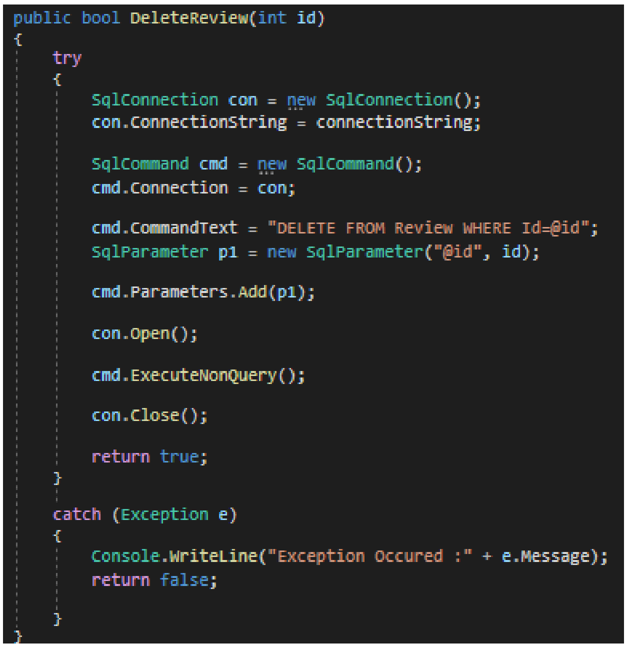
```
public void AddReview(Review review)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "INSERT INTO Review (userId, name, feedback, rating) VALUES (@userId, @name, @feedback, @rating)";

        SqlParameter p1 = new SqlParameter("@userId", review.UserId);
        SqlParameter p2 = new SqlParameter("@name", review.Name);
        SqlParameter p3 = new SqlParameter("@feedback", review.Feedback);
        SqlParameter p4 = new SqlParameter("@rating", review.Rating);


        cmd.Parameters.Add(p1);
        cmd.Parameters.Add(p2);
        cmd.Parameters.Add(p3);
        cmd.Parameters.Add(p4);


        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();

    }

    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);

    }
}
```

Update Review:



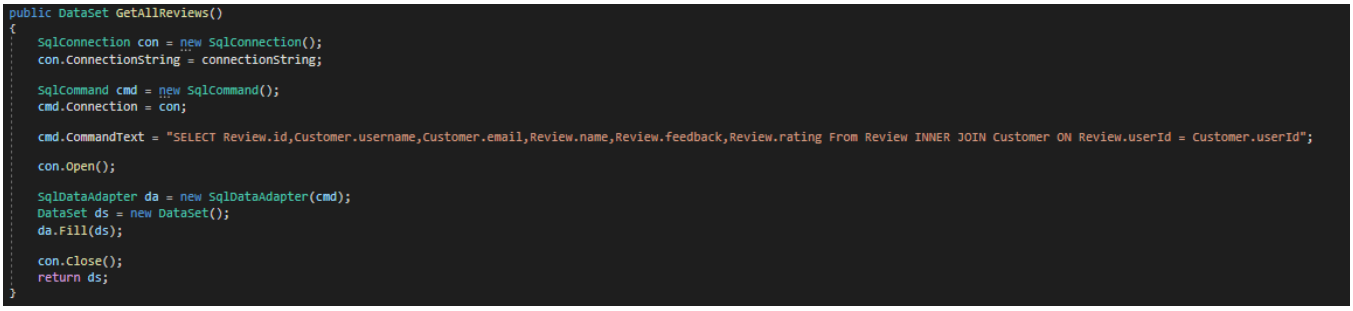
```
public bool UpdateReview(Review review)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "UPDATE Review SET userId=@userId,name=@name,feedback=@feedback,rating=@rating WHERE Id=@id";
        SqlParameter p1 = new SqlParameter("@name", review.Name);
        SqlParameter p2 = new SqlParameter("@feedback", review.Feedback);
        SqlParameter p3 = new SqlParameter("@rating", review.Rating);
        SqlParameter p4 = new SqlParameter("@id", review.Id);
        SqlParameter p5 = new SqlParameter("@userId", review.UserId);


        cmd.Parameters.Add(p1);
        cmd.Parameters.Add(p2);
        cmd.Parameters.Add(p3);
        cmd.Parameters.Add(p4);
        cmd.Parameters.Add(p5);

        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();

        return true;

    }
    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);
        return false;

    }
}
```

Remove Review:



```
public bool DeleteReview(int id)
{
    try
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = connectionString;

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = con;

        cmd.CommandText = "DELETE FROM Review WHERE Id=@id";
        SqlParameter p1 = new SqlParameter("@id", id);

        cmd.Parameters.Add(p1);

        con.Open();

        cmd.ExecuteNonQuery();

        con.Close();

        return true;
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception Occured :" + e.Message);
        return false;

    }
}
```

List Review:



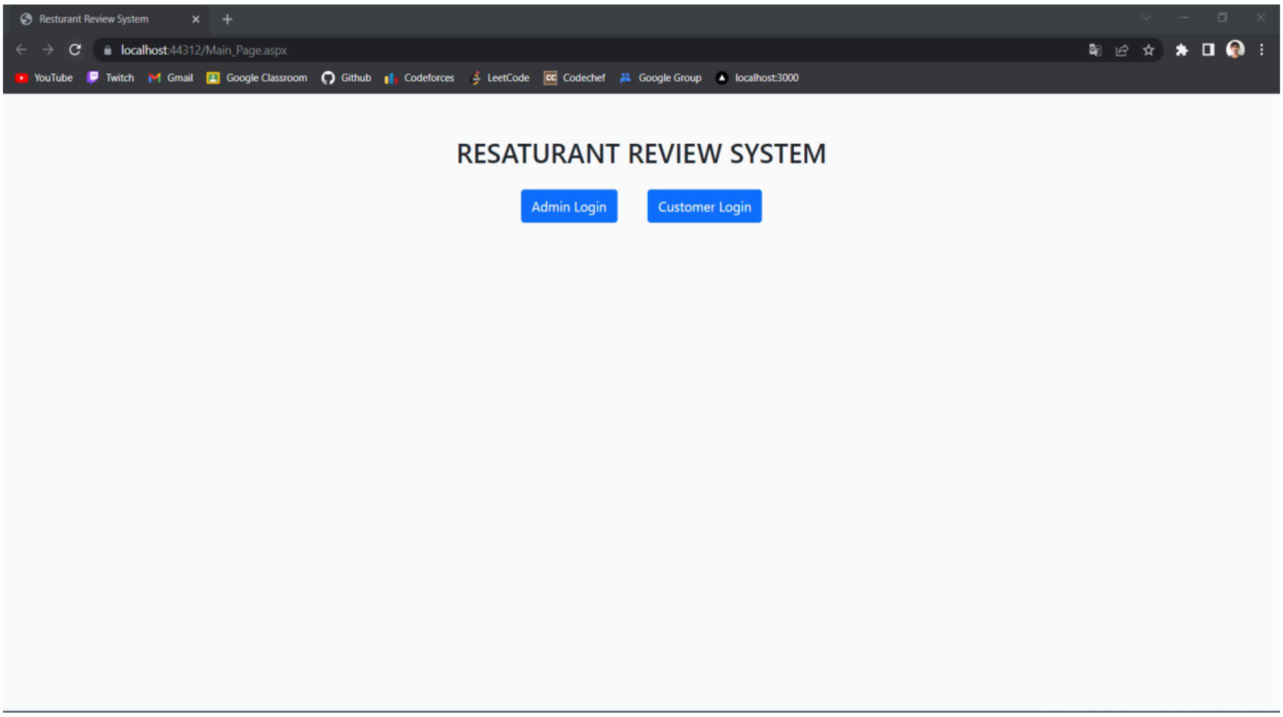
```
public DataSet GetAllReviews()
{
    SqlConnection con = new SqlConnection();
    con.ConnectionString = connectionString;

    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;

    cmd.CommandText = "SELECT Review.id,Customer.username,Customer.email,Review.name,Review.feedback,Review.rating From Review INNER JOIN Customer ON Review.userId = Customer.userId";

    con.Open();

    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataSet ds = new DataSet();
    da.Fill(ds);

    con.Close();
    return ds;
}
```

6. Testing:

Testing Method: Using Manual Testing

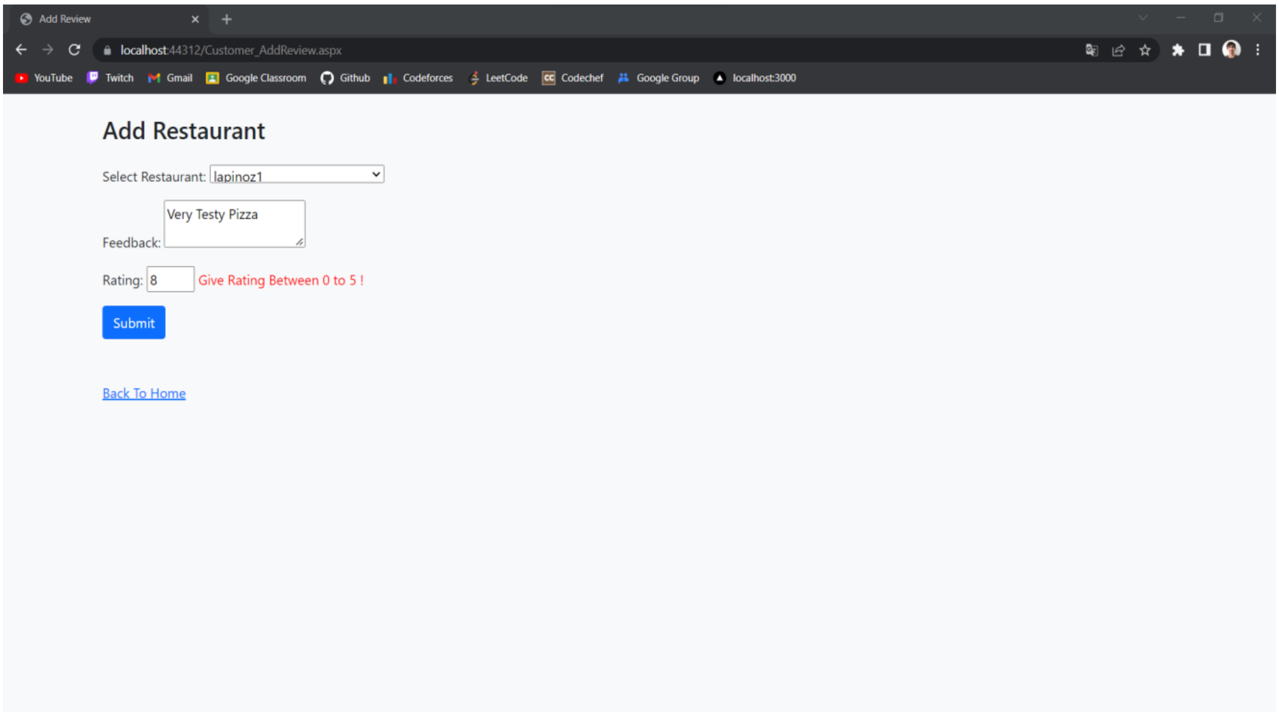
Testing CRUD operations	Input	Output
Login	Email and Password	You are successfully logged in and new form is opened
Add Restaurant	Add Restauration Details	Restaurant Will Be Added
Add Review	Give a Review Details	Review is added in database.

Screenshot



RESATURANT REVIEW SYSTEM

Login Page



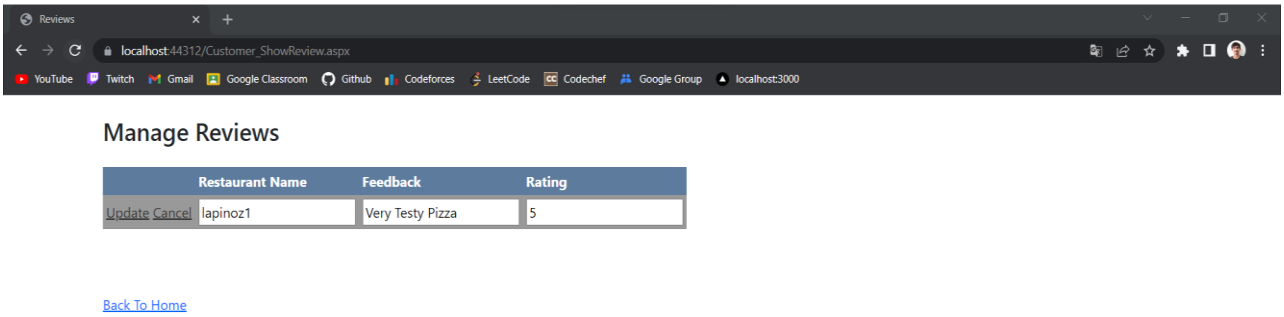
Add Restaurant

Select Restaurant: lapinoz1

Feedback: Very Testy Pizza

Rating: 8 Give Rating Between 0 to 5 !

Add Review

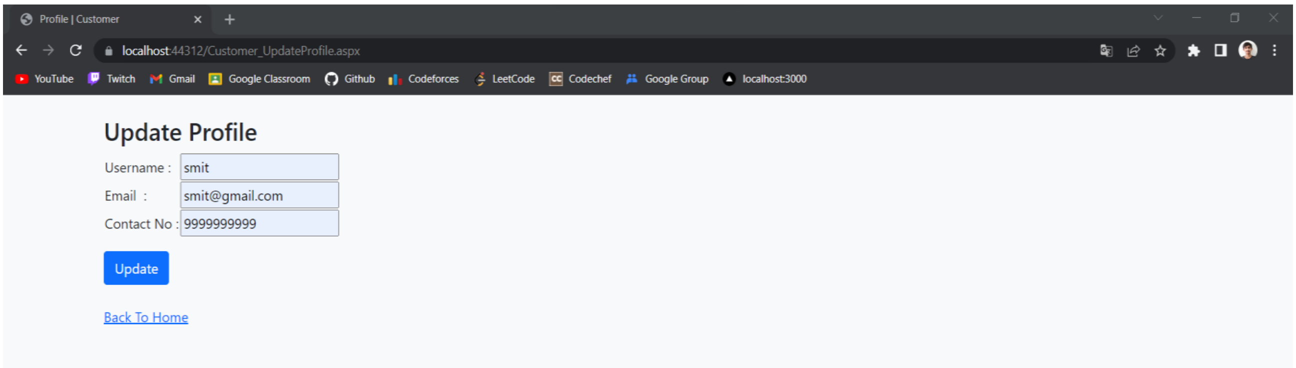


Manage Reviews

	Restaurant Name	Feedback	Rating
Update Cancel	lapinoz1	Very Testy Pizza	5

Back To Home

Edit Review



Update Profile

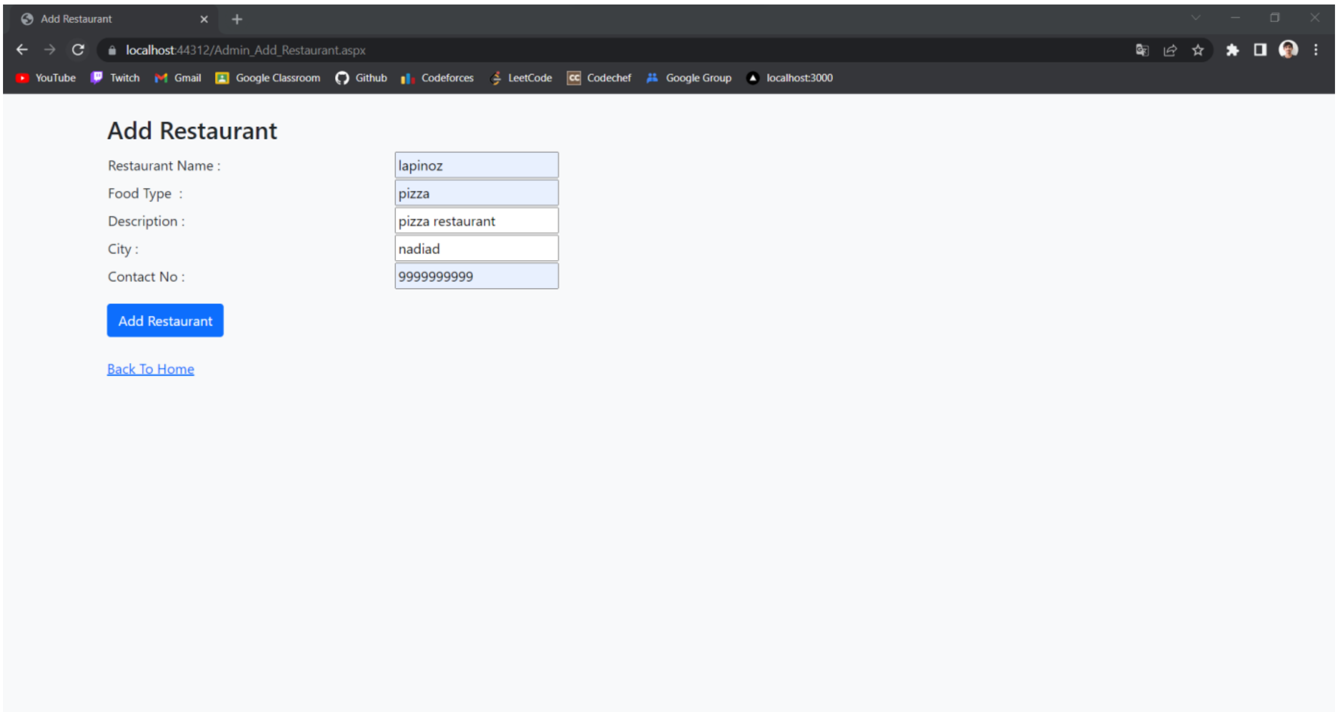
Username : smit

Email : smit@gmail.com

Contact No : 9999999999

Back To Home

Update Profile



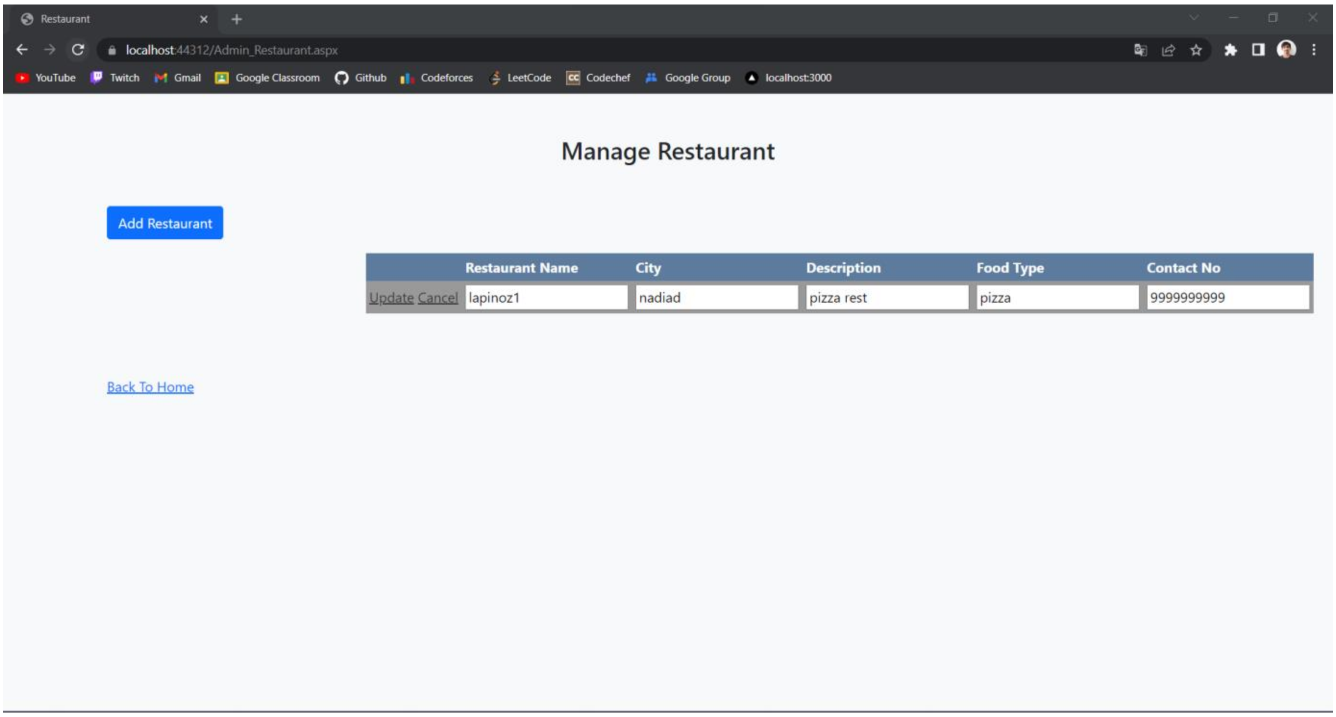
Add Restaurant

Restaurant Name :	lapinoz
Food Type :	pizza
Description :	pizza restaurant
City :	nadiad
Contact No :	9999999999

Add Restaurant

Back To Home

Add Restaurant



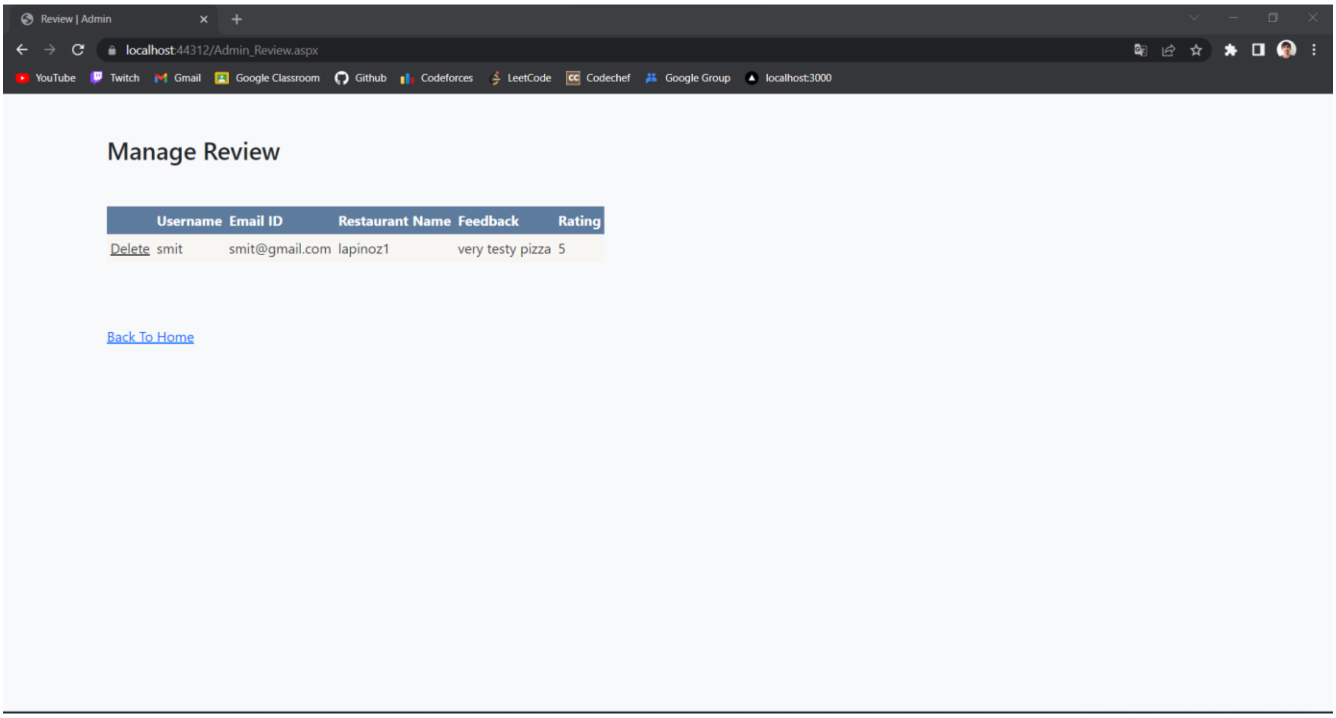
Manage Restaurant

Add Restaurant

	Restaurant Name	City	Description	Food Type	Contact No
Update Cancel	lapinoz1	nadiad	pizza rest	pizza	9999999999

Back To Home

Edit Restaurant



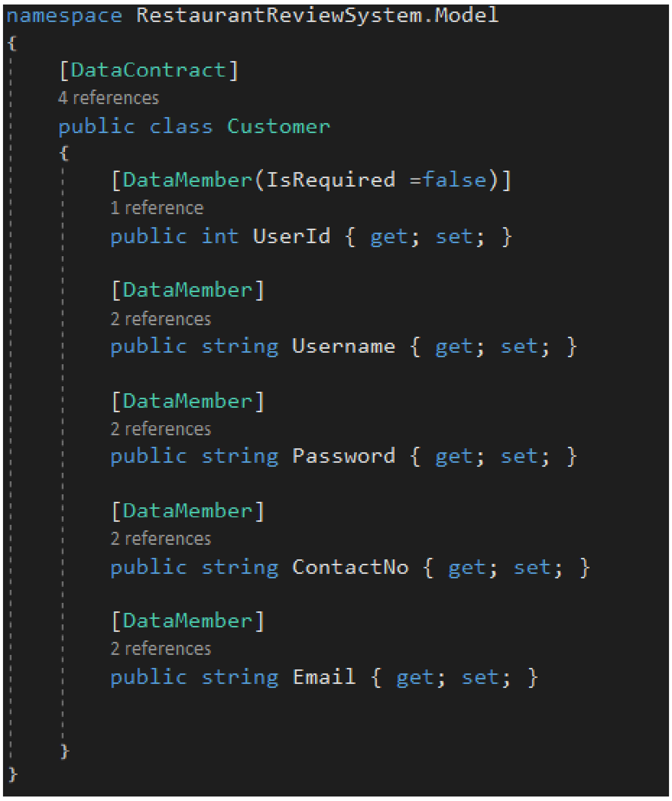
Manage Review

	Username	Email ID	Restaurant Name	Feedback	Rating
Delete	smit	smit@gmail.com	lapinoz1	very testy pizza	5

Back To Home

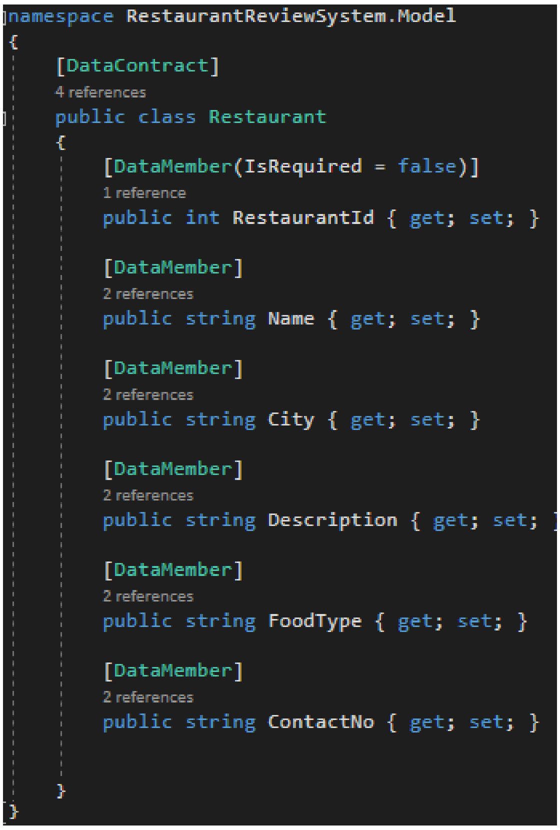
Review List

7. Screenshots:



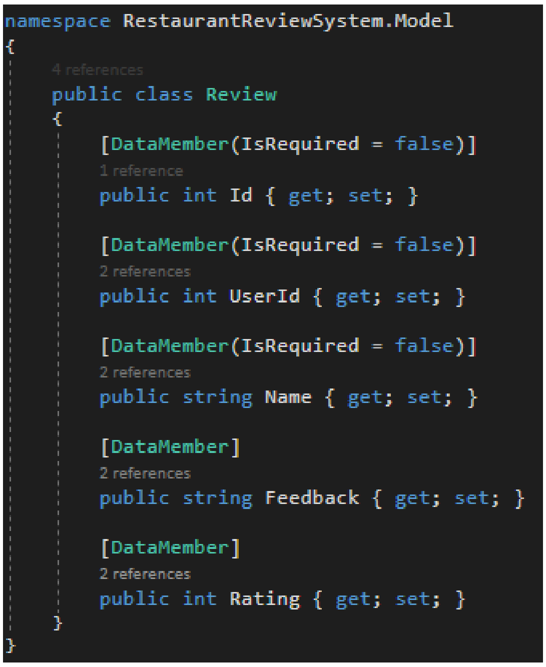
```
namespace RestaurantReviewSystem.Model
{
    [DataContract]
    4 references
    public class Customer
    {
        [DataMember(IsRequired =false)]
        1 reference
        public int UserId { get; set; }

        [DataMember]
        2 references
        public string Username { get; set; }

        [DataMember]
        2 references
        public string Password { get; set; }

        [DataMember]
        2 references
        public string ContactNo { get; set; }

        [DataMember]
        2 references
        public string Email { get; set; }


    }
}
```

Customer.cs



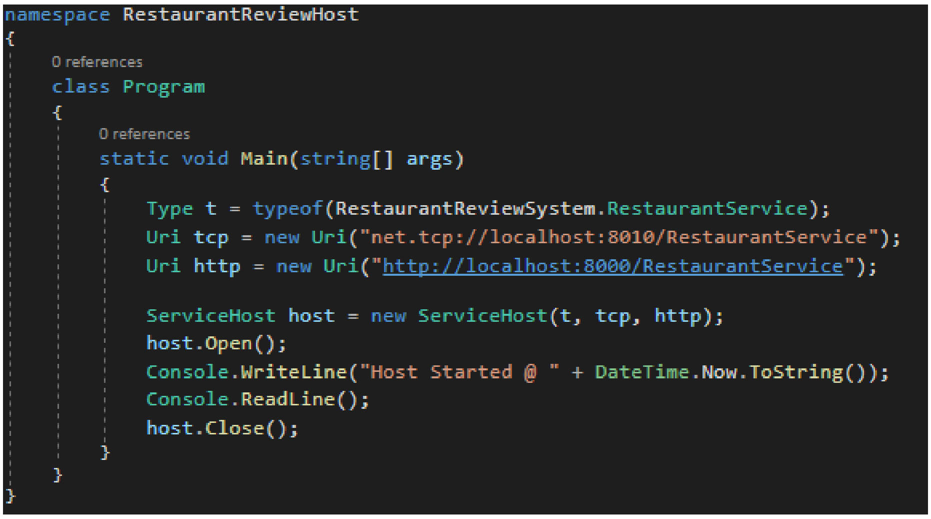
```
namespace RestaurantReviewSystem.Model
{
    [DataContract]
    4 references
    public class Restaurant
    {
        [DataMember(IsRequired = false)]
        1 reference
        public int RestaurantId { get; set; }

        [DataMember]
        2 references
        public string Name { get; set; }

        [DataMember]
        2 references
        public string City { get; set; }

        [DataMember]
        2 references
        public string Description { get; set;

        [DataMember]
        2 references
        public string FoodType { get; set; }

        [DataMember]
        2 references
        public string ContactNo { get; set; }


    }
}
```

Restaurant.cs



```
namespace RestaurantReviewSystem.Model
{
    4 references
    public class Review
    {
        [DataMember(IsRequired = false)]
        1 reference
        public int Id { get; set; }

        [DataMember(IsRequired = false)]
        2 references
        public int UserId { get; set; }

        [DataMember(IsRequired = false)]
        2 references
        public string Name { get; set; }

        [DataMember]
        2 references
        public string Feedback { get; set; }

        [DataMember]
        2 references
        public int Rating { get; set; }
    }
}
```

Review.cs



```
namespace RestaurantReviewHost
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Type t = typeof(RestaurantReviewSystem.RestaurantService);
            Uri tcp = new Uri("net.tcp://localhost:8010/RestaurantService");
            Uri http = new Uri("http://localhost:8000/RestaurantService");

            ServiceHost host = new ServiceHost(t, tcp, http);
            host.Open();
            Console.WriteLine("Host Started @ " + DateTime.Now.ToString());
            Console.ReadLine();
            host.Close();
        }
    }
}
```

Hos

8. Conclusion:

In this project we have successfully implemented 8 major functionality which are

- Add Restaurant.
- Update Restaurant.
- Delete Restaurant.
- Restaurant list
- Add review.
- View review.
- Delete review

After the implementation and coding of system, testing was performed on the system to determine the errors and possible flaws in the system.

9. Limitation and Future Enhancements:

We have implemented basic functionalities as mentioned above.

In future we will try to add more functionality like reviews for different field like ambience ,food quality ,service etc.

10.Bibliography:

Following links and websites were referred during the development of this project:

- https://stackoverflow.com/
- https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0
- https://youtube.com/

Project Git Repository:

https://github.com/sb011/WCF_Restaurant_Re

view_System.git