CIS 3342 Project 3 –   
Restaurant Review System

This project will give you more experience building multi-tier web applications. In this project you will develop a small-scale restaurant review system. This system should be a multi-page web application. You will also gain experience working with databases and designing a good data model for your web application to use. You will demonstrate data modelling skills learned from the pre-requisite database class by creating a good data model and implementing it by creating database tables.

**Database Table Requirements:**

Create a data model and implement it by creating database tables for the following system. You will be graded on how well you designed the database for this project.

**Restaurant Review System Requirements:**

Your project needs to implement the following transactions.

1. Allow for different types of users to use the application. There are three types of application users: reviewers, restaurant representatives, and simple site visitors. The application must implement the ability to add a new account for either a review writer or restaurant representative:

*Review writers are a different type of user than a restaurant representative in this web application. A reviewer account is a user that has an account with our site that can view reviews for restaurants, create new reviews, modify existing reviews, and make reservations to dine at a restaurant. A restaurant representative account has a different level of access; they can view and manage information related to their restaurant and manage reservations that were made by other users of the application. Also, the site can be used by simple site visitors that have no level of access other than viewing restaurants and reviews. Therefore, they all have different levels of access to the application and the data managed by it.   
  
A site like this would rely on a login page to determine the user and their access privileges. However, you do not need to create a login page for this application. In the interest of time, you can devise a simpler solution that allows the users to choose whether they are a reviewer or representative, enter some information to identify and locate them in the database, and take them to the pages related to their level of access as a reviewer or representative.*

1. Add a new restaurant:

*Create an ASPX page to allow the user (reviewer) or restaurant representative to add a new restaurant. There are exactly two situations in which a restaurant can be added to the database: the restaurant representative adds the restaurant, or the reviewer adds the restaurant when creating the first review for a restaurant. Creating the first review for a restaurant that does not exist in the database will require adding the restaurant to the database. In this situation, there will not be a representative assigned to the restaurant since the reviewer cannot possibly know the person to assign to it.   
  
When a restaurant representative creates a restaurant, you must include the ability to assign the restaurant representative to a restaurant. You can assume that only one representative will be assigned to a restaurant. Also, you should allow the restaurant representative to select an already existing restaurant to represent; this could occur when the reviewer adds the restaurant to the database upon writing the first review for it.*

1. Add Reviews:

*Create an ASPX page to allow the user (reviewer) to add a new review. The review must include written comments pertaining to the restaurant and giving the restaurant a rating for 4 elements: a rating from 1 to 5 stars where 5 is the best for the food quality, a similar rating for the service, a rating from 1 to 5 for the restaurant’s atmosphere, and a price level rating from 1 to 5 dollars where 5 is the most expensive. If the restaurant doesn’t exist, you should direct the user to add the restaurant and its information before allowing them to review it. Hint: you will need to keep track of the total number numbers of 1 star, 2 stars, 3 stars, 4 stars, and 5 stars ratings in order to determine the average rating for a restaurant, and the same goes for the rating for the expense ratings.*

1. Delete Reviews and Modify an existing review’s Information:

*You can use the same ASPX page to implement the* ***delete review transaction*** *and the* ***modify existing review transaction****, because they both require access to a user’s reviews. The user should only have access to their reviews, not another person’s review.*

1. Perform a search for restaurants information based on a category:

*A user can enter the initial letters of a category name and see a display of the complete list of restaurants that match the category. You could implement this with a search or by displaying a list of categories. This search should produce a list of restaurants so the user can choose one and view its information along with the reviews for it. You must allow the user to choose more than one category (for example, the user wants to see Italian and American restaurants). This display is to be presented in a dynamic display using a GridView, but you can use a Repeater.  
  
Any user can search for restaurants and view information regarding restaurants, not just users like reviewers and restaurant representatives that have accounts with the site.*

1. Display Reviews for a restaurant:

*The application should display the restaurant’s information, reviews, and display the average ratings for the food, service, and price. This display is to be presented in a dynamic display using a GridView, but you can use a Repeater.*

*Any user can view reviews for a restaurant, not just users that have accounts.*

1. Make a reservation:

*The application must allow any user (simple site visitor, reviewer, and restaurant reps) to find a restaurant, view information about it, and make a reservation. Update the database to record the reservation for the restaurant.*

1. Delete reservations and modify existing reservations:

*The application must allow the restaurant representative to view, modify, and delete reservations they created.*

1. Allow restaurant representatives to manage restaurant information and reservations:
   1. *The application must allow a restaurant representative to view and update information related to the restaurant. The application must allow the restaurant representative to assign an image used in the display. Hint: simply store a URL to an image on the web instead of storing images in the database.*
   2. *Allow the representative to view all reservations for a particular restaurant as well as modify and delete reservations. This display is to be presented in a dynamic display using a GridView, but you can use a Repeater.*
   3. *Reviewers must not be allowed to view this information nor should any user that isn’t associated with the specific restaurant.*
2. Make the site professional and attractive:

*You can use Bootstrap, CSS, or any other tools to style your site, but I expect your site to look professional through the use of colors, styles, and images.*

1. Make use of Stored Procedures:

*You must use stored procedures for all database operations.*

1. Use server-side input validation for all transactions where necessary.
2. Design principles:
   1. This application will be used by three types of users: a user visiting the site to view restaurants and reviews; a member of the site (review writer) who can write a review, manage previously written reviews, and make reservations; an employee of the restaurant that wishes to review reservations and manage restaurant information. Think of this perspective when designing the application and database.
   2. Provide a consistent and logical navigation system. The user should never have to use the browser’s Back and Forward buttons to move between pages.
   3. The user should be presented with an opening screen that presents the various transactions with links to respective pages to perform the selected transaction.
   4. Make your presentation clear to the user, providing on-screen instructions wherever needed both for data entry and error correction. If required data is omitted or entries are incorrect, the user should not have to re-enter data that is already correct.
   5. Create a good data model and implement the data model by creating the necessary tables in the database. I expect you to design and implement your own data model. You will be graded on the implementation of your data model.
   6. You need to use a proper naming convention for all controls and in your code. I expect you to properly name your classes, variables, functions, etc…
   7. Your programs shouldn’t crash for any reason; it’s poor design to have a program crash. Make sure to implement exception handling in appropriate places that can cause errors and handle them gracefully.
   8. **You must use component-based software design. This means creating code that is reusable by writing as much of the code as possible in classes and methods of classes instead of in the GUI.**

**Due:**  
See project posting under the Assignments section of Canvas.

**Submission:**You need to upload zip file containing the solution with all your code to Canvas.

**Important…**

In addition to normal submission process described above, you need to provide a reasonable number of records to allow us to run and test your application. Before submitting the project, you must populate your tables with a minimum of 5 restaurant categories and a minimum of 5 restaurants for each category.

Project’s that are not submitted properly will not be graded.   
  
You need to zip the root folder for your solution into a single zip file and submit the assignment in Canvas. To submit the assignment, you need to click the Assignment’s Title “Project 3” to view the submission form and upload the file.

**Make sure you properly submit your assignment and that it works. Programs that don’t run or don’t contain all the necessary files will not be graded and marked late.**

Please be sure to save your work periodically as you proceed and also back it up. You may want to store it on your flash drive. If you are going to zip an application in order to store it, BE SURE TO FIRST CLOSE Visual Studio. If you do store information on your flash drive be sure to copy it to a hard drive on your computer before working with the project.