IST 256 Applications Programming for Information Systems

Assignment 2

RESTAURANT REVIEWS

Due Wednesday, October 19, 2011

at the beginning of class

(revised 10/10/11)

We are going to make user interface forms for a web site that allows people to rate restaurants in their area. People can give quick reviews in terms of rating by a number of stars, or they can make a longer review by adding text. In order to encourage reviews of different types of restaurants, the web site awards reviewer points based on the type of restaurant and the type of review. In this assignment, we will make the user interface form and allow the user to see how many reviewer points they earn. (We won't implement the back end to the web site that would save the reviews.)

The form will have three parts. The first part will ask the user to input their name and the region of their reviews. The second part will allow them to review one restaurant by giving the name of the restaurant, the type of the restaurant, the type of cuisine, the number of stars, and an optional text review. They can add as many reviews as they want by repeating this part of the form. The third part of the form will be a check-out where they can see how many reviewer points they have accumulated so far and receive a reviewer status.

ln the first part of the form, the user types their name and the name of the area where they wish to review restaurants. There should be a button here for the user to start the reviews, and if the user omits either of the two pieces of input information, a message box should give an alert and ask them to fill these in.

In the second part of the form, the user can give a review of one restaurant. They should

- *Give the name of the restaurant*
- Select between 3 types of restaurants: Cheap Eats, Casual Dining, Fine Dining
- Select one or more cuisines: (You pick 4 types of cuisine to list here.)
- Select or type a number of stars for ratings, from 1 to 5 stars.

There should also be a TextArea where the user can (optionally) type a text review.

The user receives reviewer points based on what type of restaurant they review and whether they typed a text review or not:

Quick Review (just stars)	Cheap Eats	5
	Casual Dining	8
	Fine Dining	10
Text Review (adds text to stars)	Cheap Eats	15
	Casual Dining	20
	Fine Dining	25

After giving a review, the user can click a button to submit the review. The form will compute how many reviewer points they get for that review and add it to the total points in the shopping cart variable. In addition to the reviewer points listed above, the user gets an additional 10 reviewer points for their first review. The user can then go on to review more restaurants.

The third part of the form is the checkout where the user can see how many reviewer points they have received so far. In addition to the reviewer points for each review and the initial reviewer 10 points, the user will receive a bonus of 10 points for every 3 reviews that they give. If the user has accumulated 80 or more reviewer points, they will also receive a message giving them the accolade of "Top Reviewer".

When the user clicks the checkout button on the third part of the form, it should display

- the user name
- the area of the restaurant reviews
- the number of reviews that they submitted
- the number of reviewer points that they accumulated in part 2
- the number of any bonus points
- the total number of points
- the "Top Reviewer" message if the total is over 50

Finally, there should be another button where the user can clear the reviews and start over (possibly with a new name and area).

Computation for the form:

This application will require class level variables to keep information in the shopping cart. In these variables, you should keep all the information that is needed in more than one button. In particular, note that you must save the all the information that must be displayed on the checkout form.

There is also a programming requirement for this assignment. This is that you are required to write and use a method to compute the reviewer points and add them in to the shopping cart in the second part of the form.

To finish this assignment (worth 120 points):

- Design the layout of the form and make it. Give meaningful names to the components.
- Write the programs
    - Write comments on the top of each program, including **your name** and what the program is for
    - Write (simple) comments before each button actionPerformed method with the description of what it does
    - Write the code, including the method; design variables as appropriate; and add any additional comments that will make the code clear to read

For this assignment, you will be graded on:

Whether the programs runs and adds reviewer points correctly
Checking for missing input from the first two textboxes and giving an alert
Use of radiobuttons or checkboxes
Format, clarity and appearance of user interface
Design of variables and variable names
Design and use of a method
Documentation
Clarity and overall design of programs

Bonus Problem (worth up to 30 extra points):

The bonus problem is that when the user clicks the checkout button on the third part of the form, they should also receive a list of all the reviews that they submitted. The form of this list is that there will be multiple lines of text in either a label or textarea, where each line contains the information from one review, concatenated with blanks or commas. The information would include the name of the restaurant, the type of the restaurant, the cuisine, the stars, and the text, if given. For example:

> Rosie's Cantina, Casual Dining, Mexican, ***, "Good food and service, boring décor"

One way to implement this is to add an array of Strings to the shopping cart variables at the class level. Each String would have the combined restaurant review String as described in the previous paragraph. Every time that the user submits a review, you would concatenate the review information and add it to the array. You can limit the size of the array to 10 reviews, but alert the user if they try to submit too many. Then in the report in the third part of the form, you would display this text in a multi-line label or textarea.

Here are some ideas to improve the clarity and appearance of the user interface

Add colors labels or background, add image(s).
Make sections of the form in containers and enable/disable them to direct the user.
In the clear button, clear the selections of the radiobuttons and checkboxes.
. . .

How to Submit your Homework:

In your project directory, create a **zip file** of the sub-directory that has your homework project in it and submit it to the assignment 2 dropbox on the iLMS system.