Advanced Object Oriented Programming Final Assignment

(Group size : 4 people)

Due: **Friday** 4th December 2020 at 11:30 pm on Blackboard

Instructions:

Your group must develop an Event management system with a complete graphical user interface. You must also implement appropriate error checking and messages for all fields in your system. Please note the marking scheme below.

Main points:

* It must adhere to the “Eight Golden Rules” of Interface design. Covered in class and link included in the assignment description on blackboard.
* It must be a professional application that an event agency or event manager will be willing to buy.
* It must have a consistent interface following the guidelines that will be covered in lectures and labs.
* It must have error checks for field input and must be basically crash proof.

Submission Instructions:

1. At the top of you main code you must comment the name and id numbers of all group members.
2. Upload to blackboard your **complete solution compressed as a zip file** (you may submit as many times as you wish). Only the last submission will be marked.
3. You must also submit a Microsoft word document with your classes copied and pasted neatly into it.
4. **Only ONE member from each group must upload the solution zip file.**

|  |  |
| --- | --- |
| Core UI Design Principles(menu and layout consistency, readability, ease of use, etc) -20% |  |
| Design and Functionality –All operations work as expected. The design is a correct Object Oriented One taking all principles into account -40% | NOTE THE SUPPLIED STARTING CODE CARRIES NO MARKS AS IT IS GIVEN TO YOU. |
| Robustness of application (error handling without crashing, checks for duplicate data etc) -15% | Tính mạnh mẽ của ứng dụng (xử lý lỗi mà không bị treo, kiểm tra dữ liệu trùng lặp, v.v.) -15% |
| standards (is the code indented, commented, appropriate variable names, classes in capitals,etc) -10% |  |
| Completeness and professionalism (interface and overall presentation) -15% |  |

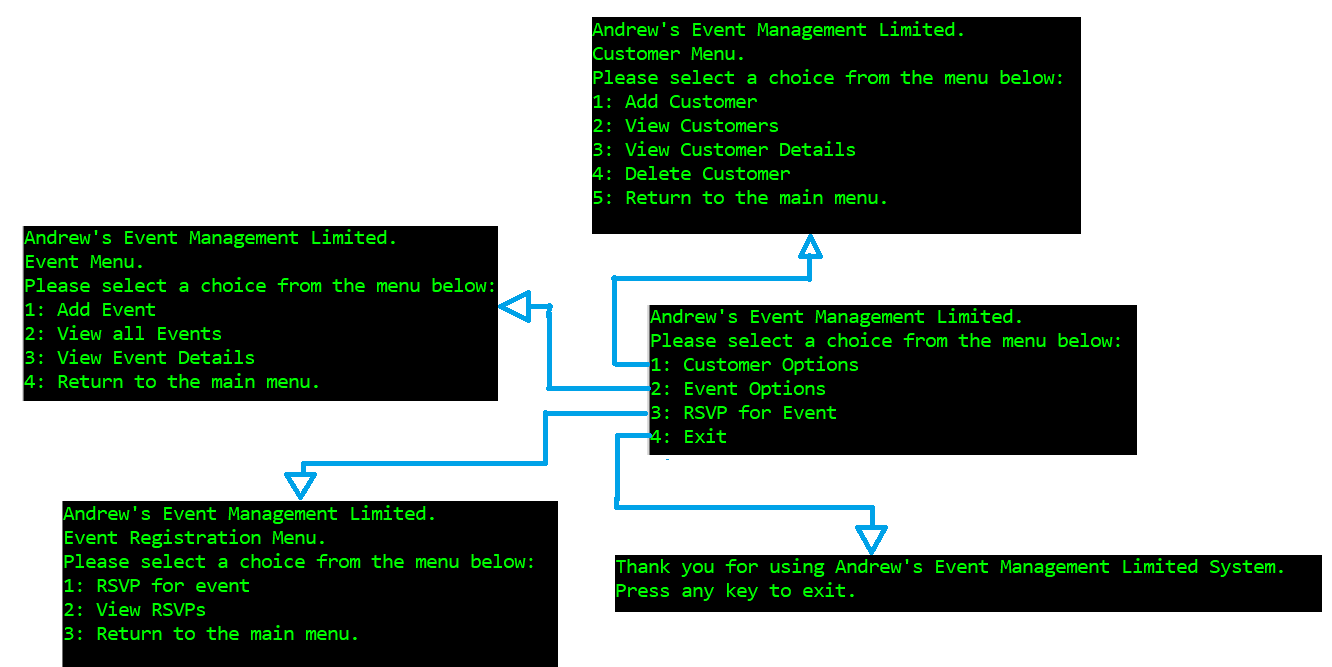
It is highly recommended that you initially design and create a text based system. If it is implemented perfectly this will give you 80% of the marks. The last 20% of the marks is related to **the graphical user interface**.

**PLEASE NOTE THIS IS ALSO A RESEARCH PROJECT ON GUI COMPONENTS.**

**Approach:**

1. **Develop the complete system with a CONSOLE interface by extending the starting code given.(This should be completed by the 21st November)**
2. **Create a new Graphical User Interface (GUI) project that reuses the fully working back end classes tested in part 1(Completed By December 4th).**

**An example of the console application menues (this menu layout is different from the one in the starting code)**



Additional Notes:

* You cannot add two events at the same venue on the same day (you must modify my code to accommodate this)
* Bạn không thể thêm hai sự kiện tại cùng một địa điểm trong cùng một ngày (bạn phải sửa đổi mã của tôi để phù hợp với điều này)
* When you “View a particular event” (View event details option) all the customers who are attending must also have their customer identification number and name displayed.

You must extend the core design and add the functionality to allow a customer to register for an event using the system.

The information that must be recorded on an RSVP (Rapid serial visual presentation)

is as follows:

* The date of the RSVP was done ( a string)
* A unique RSVP number must be assigned by the system. (NOT entered by the user)
* The event object the RSVP is being made for.
* The customer object the RSVP is being made for.

Note : you can use the following code to get the date and time from your system as a string…

string date = DateTime.Now.ToString(@"MM\/dd\/yyyy h\:mm tt");

Your main menu should have the new options:

* RSVP for event
* View RSVPs

RSVP for event

When the “Make RSVP” option is selected, a list of all customers and all events should be displayed first. The user must then be asked for the Customer id and event id to make the RSVP for.

A RSVP can only be made if there is free space in the event, the customer id exists and the event id exists. All associated objects should be updated. An appropriate message as to whether the RSVP was successful or not should be displayed.

VIEW RSVPs

When “View RSVPs” is selected, the date, RSVP number, customer name and event id must be displayed for each RSVP.

Your main class/program (the default one created when you make a new project) will be responsible for creating an EventCoordinator object as well as the complete menu system

Ensure that the interface/s generated by your program is user-friendly and gives a professional look.