1st SIT GROUPWORK QUESTION PAPER: **Year Long 2019**

Module Code:	**CS6004NI**
Module Title:	**Application Development**
Module Leader:	**Mr. Dhruba Sen** (Islington College)

Coursework Type:	**Group**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**Week 25**
Coursework is given out:	**Week 18**
Submission Instructions:	Submit the following to Islington College RTE department before the due date: • The software application to be developed in C# • The documentation in MS Word compatible or PDF format
Warning:	London Metropolitan University and Islington College takes Plagiarism seriously. Offenders will be dealt with sternly.

Plagiarism Notice

You are reminded that there exist regulations concerning plagiarism.

Extracts from University Regulations on Cheating, Plagiarism and Collusion

Section 2.3: “The following broad types of offence can be identified and are provided as indicative examples …..

(i) Cheating: including copying coursework.
(ii) Falsifying data in experimental results.
(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate’s assessment.
(v) Collusion to present joint work as the work solely of one individual.
(vi) Plagiarism, where the work or ideas of another are presented as the candidate’s own.
(vii) Other conduct calculated to secure an advantage on assessment.
(viii) Assisting in any of the above.

Some notes on what this means for students:

(i) Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
(ii) Taking extracts from published sources without attribution is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " e = mc2 (Einstein 1905)". A reference section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Further information in relation to the existing London Metropolitan University regulations concerning plagiarism can be obtained from http://www.londonmet.ac.uk/academic-regulations

Introduction

This assessment is a group coursework and should be done in a team of 2 to 4 students. It requires developing and documenting a web application using Visual Studio C#, ASP.NET and Microsoft SQL. Your software artefact must be submitted as a Visual Studio project. It will be marked using Visual Studio 2013 or above and any features not working in the standard installation of Visual Studio 2013 or above will not be assessed.

The coursework carries 30% of the module mark.

Coursework Submission Deadlines: Week 25

This Group coursework has 2 parts, both of which are to be submitted to RTE before 2pm, Friday.

(1) The software application to be developed in ASP.Net

(2) The documentation in MS Word compatible or PDF format

NB– Anyone not meeting the deadline must submit the work to the Undergraduate Registry with a completed *mitigating circumstances form*. It will only be marked if the mitigating circumstances are accepted. All parts of the late submission must be handed in to the Undergraduate Registry with the form. You must ensure that you have receipt from the Registry for your work.

Please note the rules on plagiarism

The application should be implemented individually. This is not a group/team effort. Any material which is a direct copy from someone else (student or other source) or a close paraphrase/code must be indicated where it is quoted i.e. it must be made clear what material is a quotation or close paraphrase e.g. by showing the text in italics or in quotation marks. It is not sufficient to show the source in a list of references or bibliography. If you are unclear, please discuss your examples with your seminar tutor or the module leader. Plagiarism is a serious offence and conviction for plagiarism may lead to suspension from the University, even for a first offence (please see the section on Academic Misconduct in the Student Handbook).

Software Development Scenario

The application is to support the operations of Everest Video Library, specifically administration of the renting and returning of DVDs. The application described is simplified and is an incorrect representation of normal practice and/or legal requirements. Please implement the system as described in the specification rather than as in the real world where there is conflict between them. Everest Video Library uses computer system to manage the stock, loan and return of DVDs to members. To borrow a DVD from a shop, a customer must become a member of that shop (membership is not transferable to other shops in the chain). Students need to design appropriate **E-R diagram** for the application.

A Brief Description of the System and the Functionality you must implement

The description of the functionality below is not a complete specification it is just a list of the functions to be supported. It is up to individual group to design their application to exploit the functionality of Visual Studio and .NET technology to deliver an appropriate implementation of the functionality, supporting the users' task as effectively as possible.

The system is to support the following functions:-

1. Allow the user to enter or select an actor's name (Lastname) and see the titles of all DVDs stocked by the shop (whether on loan or on the shelves) for which the Actor is a CastMember.
2. Allow the user to enter or select an actor's name (Lastname) and see the titles and the number of copies on the shelves of all DVDs for which the Actor is a CastMember and which have at least one copy on the shelves.
   (Hint – a DVD copy is on the shelf if it is not on loan. A DVD copy is on loan if there exists a Loan row for that copy where the DateReturned is Null)
3. Allow the user to enter or select a member by MemberNumber or MemberLastName and view a list of all the DVDs (by title and copynumber) that they have been loaned in the last 31 days (i.e. where there is a Loan row with DateOut >= Current_date-31, whether the DVD has been returned or not).
4. List the producer, studio and cast lists of all DVDs in the shop with the DVDs in increasing order of DateReleased, showing the title of each DVD and the names of all the cast members in alphabetic order of last name.
5. Allow the user to enter or select a Copy by CopyNumber and find the details of the last loan for that copy (whether it is still on loan or has been returned), ie the Name of the

Member who borrowed it, the date out, due and back (if any) and the title of the DVD.

6. Allow the User to issue a DVD Copy on loan to a member. This process records the member going to the counter with the physical DVD copy and should involve

   Checking that, if the member is under 18, that AgeRestricted on DVDCategory for the Copy's Title is False

   That the member has not already got on loan the number of DVDs they are allowed (from the corresponding MembershipCategoryTotalLoans)

   Creating a row in Loan with

   LoanNumber 1 more than the highest value currently on file (hint, this will be the loan number on the last row currently in the table)

   LoanTypeNumber corresponding to the LoanType requested by the Member (decode options on LoanType table)

   CopyNumber the copy number of the DVD being issued

   MemberNumber the member number of the member borrowing the DVD

   DateOut the date part of the current (system) date

   DateDue date out plus the LoanDuration for the LoanType chosen

   DateReturned Null

   Displaying the charge the Member has to pay (the StandardCharge for the DVD title times the LoanDuration)

7. Allow a user to record the return of a DVD copy

   This involves

   Locate the row on loan for this Copy which has DateReturned Null

   Updating DateReturned to the date part of the current (system) date, indicating the DVD has been returned

   Checking if the DateReturned is after the DateDue and, if it is, calculating what the Member has to pay which is the number of days DateReturned is greater than DateDue times the PenaltyCharge for the DVDTitle and displaying this

8. Produce an alphabetic list of all members (including members with no current loans) with all their details (with membership category decoded) and the total number of DVDs they currently have on loan. This report should highlight with the message

"Too many DVDs" any member who has more DVDs on loan than they are allowed from their MembershipCategory.

9. Allow the user to enter a new DVD Title into the system. This should allow for situations where it is the first Title involving a particular Producer, Studio and/or Actor, i.e. requiring adding row(s) to Producer, Studio and/or Actor. A new title will always require adding rows to CastMember.
   This task is particularly difficult to get perfect, so reasonable attempts can score full marks.
10. Everest Video Library has a policy of removing from stock all DVD copies which are more than 365 days old.
    This function should
    a. Produce a list of all DVD Copies which are more than a year old and which are not currently on loan
    b. Delete all these copy rows (the assumption is that the staff use the list to remove the DVDs from the shelves)
11. Produce a list of all DVD copies currently on loan (DateReturned Null on Loan), in date order of date out with the loans of a particular day ordered by the title. The list should show the Title of the DVD, the copy number and the name of the member who borrowed it and, for each date out, the total number of loans.
12. Allow the user to produce a list of all Members who have not borrowed any DVD in the last 31 days (defined on DateOut of their most recent Loan). You may ignore any Member who has never borrowed a DVD.
    This list should be in increasing order of the DateOut of their most recent loan, showing the member First and Last Names and Address and the DateOut and DVDTitle of the last Loan and the number of days since the last loan.
13. Allow the user to display a list of all DVD titles in the shop where no copy of the title has been loaned in the last 31 days (ie all titles for which there exist no copies with loan rows for the copy where the DateOut is >= Current_date – 31).
14. Allow Assistants to change **their own** passwords (on User) while Managers can change any detail on any row on User (including adding and deleting rows).
    For full marks, this function must not require the user to log in again, i.e. the application must keep track of who is logged in from their initial login (see below) and not require them to identify themselves again.

User Access Control

Access to the system is to be controlled by password. Different users should have access to different

set of the system's functions.

1. Any user can access to functions 1 and 2 bypassing the login. The other functions require to login either as Assistant or Manager as follows.
2. Assistants should have access to functions 1 to 13, and change ONLY their own password in function 14.
3. Managers should have access to all the functions 1 to 14 of the application

DELIVERABLES

Your submission should include the software project and a reflective essay as described below.

1. Your software artefact in the form of a complete Visual Studio 2013 or above project, which should include the program's source code, compiled classes, the executable file and database.
2. A reflective essay (1200 words), which concisely documents:
    a. detailed instructions to run the program
    b. concise description of your solution design for each of the implemented function of the application. Please do not include code listing here.
    c. the architecture of your software in terms of software classes and their purpose, clearly indicating which classes to be of your own work and which classes from other sources (e.g. from textbooks, online sources such as MSDN etc.)
    d. detailed description of the classes' properties and methods
    e. individual group member's reflection of own experience of using Visual Studio, C#, ASP.NET and MS SQL Server to develop a web application interfacing a backend database, e.g. which feature you like and why, what issues you experienced and your solution to overcome it.