**C# II Project - Semester 2 (Software Development) [100]**

**Due date: Thursday, 11th October before 17:00 – there will be NO late acceptance.**

* **Project solution folder and database files, Project Demo Presentation (Video) and Documentation must be loaded to the Learn site as a zipped file**
* **A folder with the database files, solution folder, Video and documentation on your folder( folder is a folder with your student number) in the Students folder on the Shared Drive, ONT2000\_2018/Projects**

A system needs to be written that will enable a lecturer to manage student projects, project tasks and project teams of students. The lecturer must be able to assign marks to students This project is also used to allocate marks for the students for the project and the project’s tasks.

You must use SQL Server, stored procedures and classes. You must use a 3-tier architecture for the system.

System: This system will enable:

1. The entry and updating of a project’s details. The information that should be stored for a project is:  
   - project ID, project title, project description, project start date, project due date, project owner, project supervisor, project status.
2. The entry and updating of a student’s details. To be assigned to a project a student’s details need to be stored. The information that should be stored for a person is:  
   - student ID, first name, surname, cell no, student status
3. The entry and updating of a supervisor’s details. To be assigned as a supervisor for a project a supervisor’s details need to be stored. The information that should be stored for a person is:  
   - supervisor ID (can be staff ID), first name, surname, cell no, supervisor status
4. The assignment of a student to a team for a project. To assign a student to a project, both the project and the student must exist in the database. The information that should be stored for an assignment is:  
   - assignment ID (can be autonumber), project ID, student ID, date of assignment, role on project, student’s mark for project, student’s buddy mark, comment, assignment status
5. The entry and updating of project tasks. A project can be divided into smaller tasks. These tasks can also be allocated a mark for the specific task. The information that should be stored for a project task is:  
   - project task ID (can be autonumber), project ID, start date of task, end date of task, completed date, task status
6. A student on the project team can be assigned to specific tasks of the project and can be allocated a mark for the specific task. The information that should be stored for a student for the specific task is:  
   - taskAssignmentID (can be autonumber), project task ID, student ID, assignment role, date of assignment, assignment mark, comment, assignment status
7. Entry/Update of Buddy Rating for a project. Team members can rate each other’s contribution to the project. The information that should be stored for a student for the specific task is:   
   - buddy ID (can be autonumber), project ID, student ID, evaluator student ID, mark
8. Calculate Buddy mark for student for a project and update the assignment buddy mark for the student.
9. Queries on the projects, project tasks, student assignment to a project, tasks and marks

**Requirements for System**

The minimum requirement for this system is that it should:

1. Enter/Update a project
2. Enter/Update a student/supervisor
3. Assign a supervisor to a project
4. Assign a student as a team member of a project

* Assign a student to a project. The student’s role can be specified. The date of assignment must be specified.

1. Enter/Update a task for a specific project
2. Assign a student to a specific task (you can have multiple team members to a task  
   - Assign a mark for a student for a specific task
3. Entry/Update buddy ratings for a project
4. Calculate and update the project buddy mark
5. Enter/Update marks for project and team members
6. Enter/Update of completion date of a project task
7. Reports/Queries required:
   1. All projects
   2. All Projects and team members
   3. All Projects and project tasks
   4. All projects and marks
   5. Query a specific project  
      - team members assignments  
      - tasks and assignments  
      - marks
   6. Query a specific student  
      - assigned project and tasks  
      - marks
   7. Query a supervisor  
      - assigned projects
   8. Buddy Rating for all projects
   9. Buddy rating for a specific project
   10. Projects with no team members
   11. All project tasks that are overdue

These could simply queries in grids etc OR you can load into a WORD doc.

[70]

1. **Documentation** - You must hand in the following:
   1. User document for your programs
      1. Functions in your application
      2. Screenshots and details on how to use the screens
   2. Programmer document
      1. Hierarchy Diagram of form classes/methods
      2. Details on classes
      3. Database details
      4. Code in programmed methods
   3. Video

[30]

**More on Documentation Details**

Remember that you must have an index as well page numbers and sections.

You need to provide the following:

Programmer/System documentation:

1. Introduction to your system – Here give a short description of your project
2. Database Details - ERD – This will show your database ERD, you can give a screenshot of the SQL Server ERD
   1. ERD
   2. Stored Procedures (some examples)
   3. Make sure that if you have some complex SELECT statements, document those
3. Details of classes – for each class you need to provide
   1. Class name
   2. Field/Properties
   3. Methods – just a list and maybe short description
   4. Do for DataAccessLayer and all the Business Layer classes (eg Project, ProjectTasks, Person (Student/Supervisor), Assignment, etc)
4. Hierarchy Diagram of classes
   1. Show a class diagram of your system. Each form is a class and if that form calls another form then that is a child in the hierarchy of the original form. This will show the flow of the system and the forms involved.
5. Programmed methods
   1. This is about discussing how you have programmed.
   2. If you are using 3-tier architecture, then specify this and discuss and give examples of some of the more detailed methods, not ALL methods as I can look at code.
   3. Again this is where I can look to see if there is anything interesting or complex coding

User documentation

1. This part of the document is for a user so that they know how to use the system.
2. Break your system into sections (eg Main form, Projects, Assignments, Queries/Reports etc)
   1. Have a screenshot of form and a brief description on how to use it
   2. If a form calls another form, then you can give screenshot of that form and describe that forms operation.
   3. This is to show the flow of your system
   4. Remember, that when I mark, I refer back to this document to remind myself of the system.