**EPTIA21**

**Semester 1**
**Feb – May 2021**

**Group Assessment #1**

**Module Title:** **OOP Programming**
**Module Code:** EPTIA2021
**Assessment Type**: Practical Assessment
**Weighting:** **80%**
**Maximal Possible Mark:** 100 marks
**Final Submission Date:** 25/05/2021
**Weekly Submission Dates start:** 19/04/2020

# Assignment brief

| **Assignment title** | ***Concepts, design and implementation of an OOP application*** |
|---|---|
| **Language: C# with asp.net MVC** | |
| **Purpose of this assignment**<br>To enable learners to understand the concepts of complex OOP applications and apply the skills acquired to develop and test the project. | |
| **Groups must commit weekly updates to GitHub April 19th. The commit message (or build notes) must note what has been achieved that week and what is planned for the following week. Presentations demonstrating progress will begin the following week**<br><br>**Groups must use Trello to track their 'sprints' each week. The Group leader will keep the trello board updated and present it each week in class. The lecturer will explain how to use Trello. It is up to the Group leader to decide how much they want to commit and deliver each week to reach their deadline.**<br><br>**There is 10 marks for delivering weekly commits to github.** | |

Virtual Global College (VGC) is an old college with 3 branches in Ireland. They currently run a paper-based (word files, excel files) system to manage the 3 branches. As a developer you have been assigned to design and implement an application to register and manage courses, students and faculty members across the branches.

Note: this project can be completed either using file-storage or SQL storage (depending on the course curriculum)

This application should be able carry out the following tasks:

- Registering of Students for courses and their ongoing attendance for courses
- Managing Student contact and personal identifying information
- Tracking of Academic Progress of students (results for assignments and exams)
- Tracking Payment of fees – including partial payments
- Tracking of Exams, Assignments and dates and results (uploading not required)

Faculty Members should be able to access details relevant to them (i.e. only their students):

- Student's Grade Book (Assignment results)
- Student's Attendance
- Contact detail of students for tutors only.

Students should be able to login and see all details relevant to them (except provisional exam results before being released).

You must implement the following :

1. Administrator Login with Super Privileges for creating courses, timetables etc.
2. Faculty Login with Privileges pertaining to their students
3. Student Login with privileges that only permits the to see details pertaining to them.
4. Manage Student Registration and Enrolment in courses.

You must also complete at least 3 of the following features:

1. Manage Academic Calendar (Exam dates, assignment dates)
2. Manage Exams, Assignments and Results.
3. Manage Student Attendance.
4. Manage Network – Adding more branches.
5. Manage Course Lesson Plan.
6. Manage Time Table.
7. Management of fees paid

**Tasks**

1. Complete and deliver a UML program showing the objects, models and actors in your design.
2. Complete and deliver an entity relationship model of your database schema.
3. Complete and deliver a working version of your software using the Model View Controller pattern.
4. Complete and deliver any configuration files to enable your project can be recreated
5. All projects should be submitted to github and show multiple commits and branches from the team over the duration of development. If you choose to deliver a SQL version of the project, your sql code should be part of your github project.
6. A short report explaining your project, describing the areas you have completed and those that you have not completed. You must highlight all areas you have tackled in your project (see coding guidelines) and provide pointers to the code where you have demonstrated the various features.
7. Build Notes (or a readme) explaining how to rebuild your project and integrate any files or databases.
8. Note: If using a database, configuration should be stored as a constant so that the project can be rebuilt quickly.
9. A short video of your project working (uploaded to youtube, non public mode)

## Coding Guidelines

All code files should contain your student names AND and number.

All git repos should contain the a readme.md file with student numbers .All reports should contain your student number in the name.

Github projects will be recorded early on in the cycle after naming guidelines have been issued

Negative marking will be applied for code that fails to compile from github. You should get a colleague to test your upload to confirm it compiles correctly.

You are expected to make extensive use of course concepts including

Abstract Classes, Concrete Classes, Inheritance
Functions
Polymorphism and Interfaces
Collections
Entity Framework
Strings, Characters and Regular Expressions
Files, Streams and Object Serialization
Generic Collections: Lists, Stacks and Queues
Searching and Sorting
Custom Generic Data Structures
Threading

## Follow up questions / clarifications

You will need to make many assumptions as you develop the project. You should document these assumptions in your report so they are taken into account during assignment evaluation: for example, you might assume that no results or students can deleted from the system.

The following section will be updated as the course and projects progress to clarify any ambiguities or additional information required to complete the project.

**UML Software**
You can find a free (community) UML diagramming tool here: https://www.visual-paradigm.com/editions/

Lucid chart also offer free version: https://www.lucidchart.com/
If you cannot export from these tools, take screenshots

## Marking Guide

| | |
|---|---|
| UML Diagram | 5 |
| Report | 5 |
| Task Completion | |
| Completeness of Login for admin, faculty and students | 10 |
| Completeness of 4 optional features | 10 (10* 4) = 40 |
| Code Quality, appropriate use of OOP | 10 |
| Design of UI (ease of use, UX, visual impact) | 10 |
| | |
| **Weekly Progress Commits** | 20 (approximately 2 marks per week) |
| | |
| | |
| | |
| **Negative Marking** | |
| Missing student name and number in files | -10 |
| Compilation of Builds from git hub failing due to code errors or missing libraries | -10 |
| | |
| **Submission Checklist** | |
| Link to GitHub | Important!! |
| Link to YouTube video walkthrough | Important!! |
| Copy for your Report (Word document) | |
| UML Diagrams supplied in a Word or Powerpoint document | |
| Screen Grabs of your working application (in Word or PowerPoint | Very Important!! This is used as guide to check your software. Missing screens will result in significantly reduced marking. |
| Does your report contain a contribution section? | |
| Do your files contain your student name and number? | |