Case Study: Student Grading System

Student Grading System is required, in which;
1. Student can ask the system
 > about grade they obtain for certain unit,
 > units that they take for that semester, and view unit description.
2. Lecturer also can access the same information as students do, in addition the lecturer can;
 > create a new unit,
 > edit the unit details,
 > create the assessment category,
 > update the assessment score and grading for students.

My Classes

Implementation Details

Disclaimer: the syntax in this descriptive text does not heed the programming rules of c# but rather the mentions allude to the syntax.

1. User Abstract Class.

The user class provides a template for both the lecturer and student to inherit from. The below listed are the implementation details of the User class.

Protected attributes

Firstname - User's first name

Lastname - User's last name

Password - User's log in password

LoggedIn - User's log in status, which is a Boolean type value

Methods

Constructor - takes a first name, last name and password, but the login defaults to false.

LogIn - method that sets log in to true

LogOut - method that set the login attribute to false

Print - Abstract methods to be specified for each user

2. Lecturer Class

Inherits from the User abstract class. The lecturer class contains the classroom (which contains the Unit, list of students, a System.Collections.Hashtable containing student results and the class number) and a lecturer ID

Methods

Most methods, in the Lecturer require the lecture ID since they might need to call a method in another class to which it'll pass the lecturer ID in order to execute.

Constructor - takes first name, last name, password, and Lecturer ID. Classroom defaults to a class object with an empty list of students, null unit, empty studentResults and a randomly assigned classNumber. Lod in attrubute of the lecturer defaults to false.

PostResults - Posting the assessment results

UpdateResults - updating the assessment results. This method calls assessment.UpdateAssessment() which requires the lecturer ID to execute.

CreateUnit - creating a new Unit

DeleteUnit - delete and existing unit

UpdateUnit - update unit details. This method calls unit.UpdateUnit() which requires the lectureID to execute.

GetStudentDetails - show all students and their details. This method calls the student.Print() method in a foreach loop and allows the teacher to see all details in the students in her class

PostGrade - this method posts the gradde but needs the lecturer ID to execute

Print - Prints out the lecturer Details. THis method also calls classRoom.StudentList() which displays student details in the lecturer's class. It also calls classRoom.StudentGrades() which displays the student grades next to the student IDs

3. Student Class

The student class inherits from the User abstract class. The student contains additional attributes, student ID and a list of units they take.

Methods

Constructor - takes first name, last name, password, and student ID. List of units defaults to an empty list, and logged in defaults to false.

ViewGrade - This method displays the students grades in a unit, including the unit's assessment scores. This method calls that unit.Display which requires a student ID and returns results they attained in the unit being taken in that class.

SemesterUnits - this method display all the units the student is taking in that semester. This method calls unit.Print in a foreach loop, which returns each unit's details

Print - This method prints out the students details. It outputs the student name, student ID, the units they're taking and the grades including semester scores. This method makes calls to unit.Display which prints out entire details of the unit the student in taking.

4. ClassRoom Class

This class contains a list of students, one unit, a System.Collections.Hashtable (a key value pair with containing student ID (as keys) and their Unit grades (as Unit type values) providing a quick look up for queries) and a public class number.

Methods:

Constructor - takes a unit parameter, the list of students defaults to an empty list, hashtable defaults to an empty table. the class number is assigned a random number.

Unit - the property for the unit attribute

AddStudents - Adds a studen to the class' list of students. This method requres the lecturer ID to execute.

RemoveStudents - removes a student from the class' list of students. This method requres the lecturer ID to execute.

StudentList - Displays student details inculding their grades and assessment scores for the specific unit. This method calls student.Print method in the student object.

(1) StudentGrades - returns a list of all students and their grades in the specific unit including assessment scores. This method makes calls to the student.Print method in the student object.

(2) StudentGrades - takes astudent ID parameter and allows the lecturer to view the student results of a specific student.

5. Unit Class

This class instantiates the units taken. Since the student contains the unit classes some methods require the lectureID in order to execute. It contains the below listed attributes:

Name - the name of the unit.

Semester - the semester during which the unit is taken.

Description - the descriptive information about the unit.

Assessments - the list of assignments the unit has.

Grade - the grade attained in the unit

Methods

This class contains properties for each and every attribute. In addition it also contains the below:

Constructor - Requires a name, semester, and destription input to instantiate and object. The grade defualts to null, the assessments default to an empty List of Assessment.

AddAssessment - This method requires the Lecturer ID to execute. It adds an assessment to the assessment list in the unit.

RemoveAssessment - This method requires the Lecturer ID to execute. It removes an assessment to the assessment list in the unit after checking whether or not the assessment exists in the list.

UpdateUnit - requires a unit object from which it copies the details passed to itself using the properties. This method also requires the lecturer ID to execute.

Display - This method returns the details of teh unit including teh student's scores in assessments and overall grade of the unit. This method requires a student ID.

Print - This method prints out information about the unit exclusive of grade, assessment scores

6. Assessment Class

private data members/attributes:

Name - name of the assessment

StartDate - start date of the assessment

DueDate - end date of the assessment

Feedback - the feeback provided for the assessment

Description - descriptive information about the content of the assessment

Open - a boolean value that determines accessibility. If the data of the day in question is within the start adn end date it defaults to true and students can attampt it otherwise it stays false.

Catergory - the assessment categories. This attribute utilises the Category enumeration.

Score - the attained score in that assessment.

Methods

This class contains props for each attribute mentioned above. In addition it also contains the below methods

Constructor - takes the parameters name, start date, due date, description, category. Feedback defaults to an empty string, score defaults to null, open defaults to false

UpdateAssessment - This method takes a lectureID input along with a copy of an assignment and updates its details to the passed assignment then deletes the copy passed to it.

CloseAvailability - This method checks the day's date the to make the assessment closed, if the date is within the range start to end dates it'll require lectuerer ID to change the status.

MakeAvailable - This method checks the day's date the to maek the assessment open, if the date is outside the range start to end dates it'll require lectuerer ID to change the status.

Print - This methods prints out every detail of the assessment. It prints a "_" for the null values.

7. << Enumeration >> Grade

This is an exhaustive enumeration that contains the different assignable grades.

HighDistinction

Distinction

Credit

Pass

Fail

8. << Enumeration >> Category

This is an exhaustive enumeration that contains the different assessment categories offered, namely:

GroupWork

OnlineTest

PassTask

CreditTask

DisctinctionTask

HDTask