



**Unit:**  
**Analysis, Design and Implementation**  
**Assignment title:**  
**Student Feedback Tracker**  
**Winter 2019**

**Important notes**

- Please refer to the Assignment Presentation Requirements for advice on how to set out your assignment. These can be found on the NCC Education *Campus*. Click on Policies and Advice in the left-hand menu and look under the Advice section.
- You must read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensure that you acknowledge all the sources that you use in your work. These documents are available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- You **must** complete the '**Statement and Confirmation of Own Work**'. The form is available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- Please make a note of the recommended word count. You could lose marks if you write 10% more or less than this.
- You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media that cannot be run directly, will result in a fail grade being awarded for this assessment.
- All electronic media will be checked for plagiarism.

## Scenario

A local educational establishment has decided that it would be useful if they could ensure that students had a comprehensive record of all assessment feedback they received during the course of their studies. This would include the grade they received for each piece of work, summative feedback, comments and suggestions for improvements. It would also allow for any notes they personally wish to record about the feedback they received. These notes would be personal to the student, but every other piece of feedback could be queried by the student or any member of staff.

You have been tasked with designing and developing a piece of software appropriate for this new model of feedback tracking. It should allow for administrators to register students as being on a course of study, and to record all the individual modules or subjects to which they have been registered. Each of these should permit a number of assessment entries to be added and for these to be identified by whether they are an assignment, examination, or something else.

A student will then be assigned to one or more courses, each of which will be made up of one or more subjects, and each subject will have one or more assessments. For each assessment, a student will have the following information available:

- Type of assessment (viewable by staff and the student)
- Grade (viewable by staff and the student)
- Summative feedback (viewable by staff and the student)
- Recommendations for improvement (viewable by staff and the student)
- Personal notes (viewable by the student only)

The system should allow the student to search the text of any of these fields for a piece of text, and also search by grade and grade ranges, as well as combinations. 'Give me all feedback from examinations where I received a C or a B and where the feedback included the word 'proactive', as an example.

Students should not be able to change any of the recorded data except for their own personal notes, and staff members should not be able to change feedback on modules to which they are not assigned – your system will need to keep track of user accounts to make this possible. At least three levels of access should be provided – administrators who are permitted to add users and modules. Staff who are permitted to add feedback and assessments. Students who are permitted to add notes to their own feedback.

Staff assigned to a module should also be given access to some statistical information, specifically grade breakdowns (in bar chart form) **and** word clouds of the words used most frequently within feedback. It should be possible to generate these by date as well as over the full set of data within a module. The word cloud should omit stop words such as *the*, *and*, and so on.

In addition to permitting feedback to be searched, your application should also permit for feedback to be browsed – by date of submission, by subject name, by assessment type, and so on. Browsing should provide all the key information in a suitable format such as a table.

Your application then needs to provide the following functionality:

- Allows for new accounts to be created in three different levels
- Allows for administrators to add new courses and subjects
- Allows for staff and administrators to add new assessments and feedback
- Allows for students to add private notes to the feedback they have received
- Permits students to search their feedback by grade and content
- Permits students to browse feedback in an appropriate format
- Prohibits staff from adding feedback to modules to which they are not assigned
- Permits administrators to assign staff to modules
- Provides staff with bar charts of grade breakdowns
- Provides staff with word-clouds of most commonly used words in feedback

Your solution will consist of a class diagram, a use-case diagram, **and** an activity diagram for the process of generating a word-cloud for a specific year's assessments. You should also submit the completed program code in C#.

## **Task 1 – 26 Marks**

### **Candidate class list and Diagrams**

26 Marks are available for providing an appropriate list of candidate classes, along with the supporting class diagram. Candidates classes should be justified with appropriate discussions.

## **Task 2 – 25 Marks**

### **Activity diagram**

25 Marks are available for the creation of the appropriate activity diagram, which is the generation of a word cloud for a specific year's assessments.

## **Task 3 – 8 Marks**

### **Use case diagrams**

8 Marks are available for the provision of a suitable use-case diagram.

## **Task 4 – 15 Marks**

### **Code architecture**

15 Marks are available for a code architecture that shows an appropriate level of coupling and cohesion, along with the necessary amount of inheritance and encapsulation to express the system. The code should also appropriately handle input and output.

## **Task 5 – 26 Marks**

### **System implementations**

26 marks are available for implementing the system as described and providing the completed C# code.

## Submission requirements

- Your program must be submitted as a zip file of the full project.
  - Whatever IDE you use, it should be possible to open and run the project directly from the extracted archive.
- Diagrams and materials associated with the tasks above should be presented in a word-processed document.
- All references and citations must use the Harvard Style.

## Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensured that you have acknowledged all the sources that you have used in your work?

☐

Have you completed the 'Statement and Confirmation of Own Work' form and attached it to your assignment? **You must do this.**

☐

Have you ensured that your work has not gone over or under the recommended word count by more than 10%?

☐

Have you ensured that your work does not contain viruses and can be run directly?

☐