



Unit: Analysis, Design and Implementation

Assignment title: Wiki Wiki Wiki

June 2016

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

Given the success of Wikipedia and other collaborative platforms, you have been hired by Educational Resources Incorporated to put together the design and initial prototype for a wiki application. As part of its remit to provide good educational software, the company has a number of special adaptations that it needs taken into account, and these would prohibit the use of any existing wiki software to meet their needs. You can read more about wikis here: <https://en.wikipedia.org/wiki/Wiki>

Your job is to create a wiki system that allows for three levels of user to be registered:

- Student
- Tutor
- Administrator

Students can read all pages, and edit pages only if they have been given permission.

Tutors can read and edit any page, but cannot grant read or write permissions.

Administrators can read and edit all pages, and also provide the necessary access to students and tutors.

Educational Resources Incorporated have requested that pages work through the implementation of 'Markdown', which is a standard, simple format for the writing of wiki pages. You can read more of that here: <https://help.github.com/articles/markdown-basics/>.

When a page is edited, the application should output the entire contents of the tool to static HTML files so that these can be later uploaded to a website without needing the software to be available. Permissions apply only within the confines of the application, they do not pertain to the deployed static HTML pages.

Each user then consists of a username, a classification, and routines that determine if they have access to read and write pages, or grant access. Each page contains a title, an original author, and the content of the page. It also contains details as to who has been specifically added as having edit permission. The pages should also have a 'history', which contains the previous versions of the page and the user responsible for changing them. An administrator can 'revert' a page to an earlier version if needed.

Your application then needs to provide the following functionality:

- Allows for new users to be added;
- Allows for pages to be added and edited;
- Implements appropriate read/write checks for user types;
- Stores a revision history of changes;
- Convert markdown into HTML;
- Outputs all content into static HTML pages;
- Allows for an administrator to revert a change;
- Allows for an administrator to set user permissions;
- Saves the entire state of the current wiki in a suitable data format.

Your solution will consist of a class diagram, a use-case diagram, an activity diagram for the process of converting a single wiki page from markdown into HTML and then writing it out to a static web page, and the completed program code in Java.

Task 1 – Candidate Class List and Diagrams (26 Marks)

The candidate class list should incorporate justifications and discussion as to why each class was selected for inclusion, and how its relationship to other classes was derived. The class diagram should show attributes, operations, scope and relationship of classes to each other.

Task 2 – Activity Diagram (25 Marks)

The activity diagram should incorporate the classes involved in a user specifying a computer system.

Task 3 – Use Case Diagrams (8 Marks)

The use case diagram should incorporate each of the user activities indicated in the brief.

Task 4 – Code Architecture (15 Marks)

This involves creating 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.

Task 5 – System Implementations (26 Marks)

This is for implementing the system as described and providing the completed Java 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? ☐