OO Design – CA2 – 10% - UML in Documentation

This assessment is designed for you to demonstrate your understanding of UML diagrams and how to employ these diagrams in documenting your systems. The assessment may be completed EITHER as a presentation to the class OR as a written document. In both cases, your UML diagrams should be provided as image files as well as embedded within the document/presentation.

Motivation:

Over the past year, you have been the sole developer in your company working on a particular three tier web application project. Your company has now hired several additional employees to work on this application alongside you.

In order to bring the new employees up to speed, your employer has requested that you provide high-level documentation on overall structure of the back end of the system, as well as in-depth documentation on some specific functionality you will hand over to the new employees to work on. You have been told that you can provide this documentation in the form of a written document or a presentation, but you must employ UML diagrams to support the information and should refer to these UML diagrams in your explanation of the system.

Word count: Your document should be no shorter than 500 words and no longer than 1500 words.
Presentation length: Your presentation should be no shorter than 15 minutes.

Diagrams

Part 1) High-level documentation – Conceptual Class Diagram

Create a **conceptual class diagram** of the back-end (server code only) of your system. This diagram should show any inheritance or implementation of interfaces that occur. Where a class USES (i.e. creates an instance of within itself) another class (E.G a service class creating a DAO to access the database), you should show this relationship using the dependency notation (a dashed line with a regular arrowhead).

Note: You are only required to show up to 10 classes in this diagram. If your system contains more than 10 classes, you should select 10 and include these. If your system contains classes that contain other classes (such as seen in referential integrity exercises), use other classes, employ inheritance or implements interfaces, you must include at least some of these within the selected 10 classes.

Part 2) In-Depth documentation – Specification Class Diagrams

Create **specification class diagrams** of two specific sections of the back-end of the system. You should select two tables from within your database and create a specification class diagram for the classes related to each table, i.e. any DAO classes, service classes and DTO classes related to these tables. For example, where there is an Employees table, the system could contain an Employee DTO (that implement the Comparable interface), an Employee DAO class (that inherits from a DAO superclass) and an EmployeeService class. These would all appear in the class diagram for the Employee table. However, you would not be required to show any Commands that use the EmployeeService class.

Part 3) In-Depth documentation – Sequence Diagrams

Create a sequence diagram for two specific pieces of functionality within your system, such as logging in, registering a new user, searching for some entry in the system etc. You may consider your JSP pages as actors/boundary classes that trigger the sequences. Your diagrams must include the opt, alt and loop notations, but may also include constraints.

Documentation

Part 1) Introduction

You should begin your documentation (presentation or document) with an introduction to the system – i.e. what it does, what is currently in developed and how it might be extended in the future.

Part 2) High-level documentation – Conceptual Class Diagram

Your documentation (presentation or document) should explain what each class does within the system and how they are connected. This may be achieved in the form of bullet points or paragraphs of text.

Part 3) In-Depth documentation – Specification Class Diagrams

Your documentation (presentation or document) should discuss the sections you chose to document in your specification class diagrams in greater detail. This includes a short explanation of what each method does (excluding getters and setters) and the reasoning for including multiple constructors if this is the case.

Part 4) In-Depth documentation – Sequence Diagrams

Your documentation (presentation or document) should clearly explain the sequence of events occurring within the sequences you chose to document in your sequence diagrams. This includes what functionality the sequence of events is providing, what each method does when called during that sequence and the meaning of what is returned.

In the case of all sections of your documentation, you should refer to your diagrams and specific aspects of their notation when discussing them.

Submission Details

Upload a zipped folder containing your UML diagrams (as individual image files) and either a presentation file (e.g. PowerPoint) or document (e.g. word document or pdf). The zipped file name should include your name and class group.

Submission deadline: 23:55 April 28th.