

Course: SWE6140 GROUP WORK IN LARGE SCALE SOFTWARE DEVELOPMENT
Lecturer: Dr. Lawrence Nderu
Contribution: 60% of course
Due: 17/11/2016

Learning Outcomes: A, B, C, D

Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing or buying coursework from someone else and submitting it as your own work. Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University.

All material copied or amended from any source (e.g. internet, books) must be placed in quotation marks and in italics, with a full reference to the source directly underneath the material. Your work will be submitted for electronic plagiarism checking. Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.

Coursework Submission Requirements

- An electronic copy of your work for this coursework should be fully uploaded by midnight (local time) on the Deadline Date.
- The last version you upload will be the one that is marked.
- For this coursework you must submit a single Acrobat PDF document. In general, any text in the document must not be an image (i.e. must not scanned) and would normally be generated from other documents (e.g. MS Office 2007 using "Save As .. PDF").
- For this coursework you must also upload a single ZIP file containing supporting evidence.
- There are limits on the file size.
- Make sure that any files you upload are virus-free and not protected by a password otherwise they will be treated as null submissions.

Coursework Regulations

1. Coursework submitted late without an Extenuating Circumstances claim will receive a ZERO grade.
2. All coursework must be submitted as above.

Design and building a Distributed Information System.

This coursework must be completed individually

Detailed Specification

The scenario

You have been asked to prepare an initial design for enabling a large private medical care service provider with a distributed healthcare solution. The care service provider has over 100 outlets across the East Africa offering a wide range of laboratory testing, diagnoses and treatment services. The medical functions are supported by admin staff that manage appointments and resources (a resource is defined as staff and bookable facilities, i.e. specialised rooms).

There are three categories of user: -

- Administrative staff who need to track patients, staff, resources and book appointments but cannot see their medical history.
- Doctors need access to patient information, their medical history, arrange for lab testing but cannot book appointments.
- Laboratory staff needs to track samples associated with a patient.

Your task is to design and implement a prototype 3-tier application for the business.

All information used in your system should as a minimum be stored at a central location – although distribution would be preferred.

Based on the above requirements it is necessary to build the prototype as a distributed system with a number of different user interfaces, one for each of the users. You should implement a 3-tier architecture. The system should persistently save data in a database. The choice of technology is yours.

Likewise the user interfaces can be built using any technology of your choosing. Care should be taken to achieve a 3-tier architecture.

You should produce and submit a document of your UML design for the proposed system along with screen shots of your implementation and a brief description of your implemented product. You should also produce a ZIP file containing your software code and a brief report describing what you have achieved.

Deliverables

Based on the above requirements it is necessary to build the prototype as a distributed system with a number of different user interfaces, one for each of the users identified in your design. You should design and implement a 3-tier architecture.

You should document a UML design for the proposed system.

The implemented prototype system should persistently save data in a database. The choice of appropriate technology this includes the technology for distribution¹ (middleware, RMI, web services, etc.), the database

(Oracle, SQL server, MYSQL, Postgres, MS Access, etc.) and also the application/user interface technology (Java, Java Servlets, .NET, ASP.NET, VB.NET, etc.).

Specific Deliverables

You are expected to submit two items:

A report containing only the following items:

D1: An ERD describing your backend database.

D2: An UML Deployment diagram detailing your system architecture.

D3: UML Use-case diagrams for the scenario, these should distinguish the various functions and users of the system.

D4: UML Conceptual Class diagram for the scenario.

D5: A critical evaluation of the design and implementation.

D6: A short statement of the implemented functionality along with screen shots of the application functionalities.

No software code is to be submitted in the report (code is submitted as a separate ZIP file)

Accompanying text should be kept to key assumptions of any designs and brief explanations of the software functionality.

You will be required to demonstrate your software prototype. Lack of a demonstration will be classed as a non-attempt at implementation and thus will fail the course work.

Deliverables & Grading Criteria

The System design (maximum of 50% will be awarded for your design) to include:

- a. An UML use-case diagram documenting the prototype system. (10%)
- b. A backend database design– ERD (10%)
- c. An UML design class diagram (10%)
- d. An UML deployment diagram (10%)
- e. Critical evaluation (reflection / limitations / strengths of the work). (10%)

Implementation (Maximum of 50% will be awarded for you implementation) to include:

- f. 3-tier implementation with a clear separation between tiers and using appropriate technology,
- g. Doctors functionality
- h. Laboratory functionality
- i. Administrators functionality

Grading Criteria

70% and above (distinction)

Year 2016

A very well written and professionally presented piece of work which shows a complete model and matches the implementation of the required system.

All deliverables must be covered to a high standard and quality that warrants a distinction to be awarded. Critical evaluation shows insight and a thorough grasp of DIS concepts.

Between 60 and 69 (merit)

The work lacks some detail but shows a complete design model and a working implementation of the system.

All deliverables must be covered to a standard & quality that warrants a merit to be awarded.

The design and implementation is of a good standard but missing some detail and or incomplete in some areas.

Your critical evaluation of your system is good, but lacking some detail and insight.

Between 50 and 59

The work lacks some details but demonstrates a fair degree of understanding of both design and implementation issues.

The critical evaluation is mainly factual but shows some reflection.

Between 40 and 49

The work lacks some details in either section but the design model and implementation demonstrate a pass level understanding of the core principles in distributed systems.

The design is incomplete but shows a basic understanding of the scenario and how it could be realised as a DIS. Critical evaluation is factual but ultimately unable to reference the design or implementation.

Below 40 - FAIL

A poor submission with an inadequate implementation which fails to demonstrate basic understanding of the principles involved Lack of implementation.