Faculty of Computing, Engineering and Sciences

Module Name:**Enterprise Software Engineering (Distance Learning)**

Module Code: **COSE71194**

Title of Assignment: **Assignment**

Module Learning Outcomes for This Assignment

1. Demonstrate a systematic understanding of architectures and frameworks for internet applications.	Knowledge & Understanding
2. Employ design patterns appropriate to enterprise applications.	Learning Application
3. Design, implement and test an object-oriented enterprise application.	Communication Problem-solving
4. Critically reflect on and draw conclusions about the fitness for purpose of an application.	Analysis

Submission deadline

Submit before 08 January 2021 at 16:00 (via Blackboard)

Feedback deadlines

You will be given verbal feedback during the demonstration. Further written feedback will be made available via Blackboard within no later than 20 working days after your demonstration

Introduction

This assignment, which contributes 100% to the module's marks, consists of developing an enterprise application that will be assessed by demonstration and report.

To prepare for this assignment you should complete the tutorial exercises.

The task

Your task for this assignment is to specify, design, write and test an enterprise application to fit the requirements laid out in this specification.

This document will describe in general terms what your application should do. You should decide exactly what your application will do, and how it is to be done.

You should read carefully the mark scheme to understand how the grades will be allocated to each element of the assignment.

Fundamental requirements

In this assignment, you must satisfy **all** the following fundamental requirements, otherwise you could be awarded zero marks for the assignment

1. You must use Java EE to implement this assignment.
2. You must be the only author of the work you submit for assessment. You must not have help with this assignment from any person except for members of the regular teaching team. You are reminded of the university's policy about plagiarism, as described at: http://www.staffs.ac.uk/support_depts/info_centre/handbook/academic-life.jsp
3. During the demonstration, you must be able to explain in detail the software code you present for assessment.
4. You must submit your assignment via this module's Blackboard presence using the links provided in the Assessment section. Follow the instructions given with each link.

The assignment

1. Specify your system

Choose **one** scenario from the list below and write a specification for the system outlined in the scenario. Ensure you describe clearly the required functionality and any other constraints and features that are to be included.

NB: Whichever scenario you choose your system must not take payments.

Scenario 1: Student timetabling system

Specify, design, implement and test an enterprise application for managing student timetables. There must be at least:

- two user types (e.g. student and timetabler)
- three entities (e.g. modules, classes and rooms)
- five events (e.g. schedule a class, change a booking, delete a booking, etc.)

Scenario 2: Parcel delivery system

Specify, design, implement and test an enterprise application for tracking parcel delivery. There must be at least:

- two user types (e.g. recipient and driver)
- three entities (e.g. parcel, customer, driver)
- five events (e.g. Book a collection, collect a parcel, deliver a parcel, etc.)

Scenario 3: Your own system

Specify, design, implement and test an enterprise application of your own choosing. You must have your choice approved by the module leader before beginning the assignment. Your proposed system must be of a similar level of complexity and challenge as the two scenarios listed above.

2. Design your system

Use the UML to document the design of your application. You should plan only 10-12 use cases.

Your enterprise application must demonstrate at least the following features:

- A Java EE multi-tiered architecture with a web-based (JSF) user interface
- Design patterns
- Perform insert, update, and delete operations on a database in response to user interactions

Important: Review your plans with your tutor to ensure that you are on the right track for this assignment.

3. Create and populate your database

Your enterprise application must interact with a relational database.

In a text file, write SQL commands that:

- create your database, and
- insert some initial data into your database.

4. Implement your enterprise application

Implement your enterprise application using Java EE. The best approach is to implement just one use case initially. Get all the problems sorted out, and then implement the remaining use cases one by one.

You should ensure consistency between the implementation and the UML design.

5. Test your implementation

Write a test plan (automated wherever possible) and test your web application. As well as unit tests, you should include integration tests that are based on your use cases.

6. Write a report

Write a report (up to 1,000 words) in which you critically evaluate your application. You should consider:

- the design, describing alternatives and justifying your selections;
- the results of testing, making recommendations for further development.

7. Submit your work

On or before the submission deadline date, use the links on Blackboard to upload a zip file containing:

- your design (PDF documents)
- your enterprise application (NetBeans projects)
- your report (PDF document)
- any other files you have created (e.g. the SQL commands for creating and populating your database)

Please clean your projects before submission to reduce the size of the zip file.

Mark Scheme

You should refer frequently to the marking scheme during your work on this assignment.

How your work will be assessed

Your submission will be assessed using the Rubric in Appendix A, which will give you quantitative and qualitative feedback on your work. You should ensure that your report presents evidence to support the awarding of marks in each criterion of the rubric.

After submitting your work for each part, you will be required to demonstrate your software to a member of the teaching team for this module. You will run your software to prove that it works, and you will be asked questions about your code. You are expected to answer these questions correctly and confidently. Incorrect or unconfident answers might result in your marks being reduced. You are reminded of the university's policy about plagiarism, as described at: http://www.staffs.ac.uk/support_depts/info_centre/handbook/academic-life.jsp

If you do not demonstrate your software, you will score zero marks for the implementation parts of the assignment.

Instructions for the demonstration

Email Graham (g.d.mansfield@staffs.ac.uk) to make an appointment for your demonstration. You will be sent the URL for the Blackboard Collaborate room in which we will meet.

To prevent delays in the demonstration, it is essential that you are ready for the demonstration before we begin the Blackboard Collaborate session. Therefore, you should do the following before your demonstration start-time:

1. Open NetBeans, and prepare your application, which includes running the SQL commands to create and populate your database (see step 3 above);
2. Be ready to show the tutor that the dates on your files are the same as those submitted via Blackboard;
3. Make sure that your database is in its initial state (i.e. ready to run without delay);
4. Await instructions from the lecturer who will assess your demonstration.

Appendix A: Rubric

Criterion (weighting)	Mark										
	0	**15**	**25**	**35**	**45**	**55**	**65**	**75**	**85**	**95**	**100**
Complexity of database (10%)	No database	One table	Two tables	Three tables	Four tables – one 1:m relationship	Four tables – two 1:m relationships	Five tables – one 1:m relationship	Five tables – two 1:m relationships	Six or more tables – three or more 1:m relationships		
Java EE features (20%)	No application	Attempted – but with significant errors			Use of basic features such as JSF, managed beans and direct interaction with the database; Use of all CRUD operations		Additionally, use of more advanced features such as session beans, entity classes and Java Persistence API		Additionally, use of Java Messaging Service and Java EE features not covered in the module		
Design patterns used (20%)	No application	Attempted – but with significant errors			One design pattern	Two design patterns	Three design patterns	Four design patterns	Five or more design patterns		
Implementation consistent with design (20%)	No application or no consistency with design	Attempted – but with significant errors			Some consistency between code and design		Very good consistency between code and design		Code consistent with design to a professional standard		
Testing (15%)	No evidence of testing	Little evidence of testing			Some evidence of testing		Some JUnit testing, including integration tests based on use-cases		Comprehensive JUnit testing; Automated integration tests based on use-cases		
Critical evaluation (15%)	No report	Weak analysis; little supporting evidence presented				Analysis considers major aspects of the design and application, and is supported with evidence		Insightful, reflective analysis considers all required elements, and is supported with evidence			