

# School of Technology Business and Arts

## BSc (Hons) Computing

### Level 5 Module: Software Engineering

Assessment:	Assessment 1: Project
Module Tutor:	Professor Nicholas Caldwell
Weighting in Module:	100%
Hand out:	Monday 8th January 2024
Hand in:	<b>on or before 12 noon 18th March 2024</b>
	Online submission portal

#### What is required?

- **FIVE** distinct elements, each weighted at 20% of the total marks
- (1) A **requirements specification** and supporting requirements modelling material, which may include use cases, activity diagrams, class-based modelling, CRCs, control flow, data flow and/or behavioural models.
- (2) A **design specification** which will include design documentation and diagrams
- (3) **software implementation**, which will include all code
- (4) **tests and test results**, which will include a suite of verification and validation tests, and appropriate screenshots
- (5) A **reflective essay** of up to 1,000 words discussing and justifying the key design, implementation and testing techniques used. Word count limits apply to the essay

#### Learning outcomes to be assessed:

1. Demonstrate proficiency in their knowledge and understanding of software engineering;
2. Demonstrate proficiency in their knowledge and understanding of modern software development techniques;
3. Selection of appropriate techniques and tools to plan, design, implement, test and manage software projects, including software engineering methods, data structures and algorithms, object-oriented tools (e.g. UML) and appropriate user interfaces
4. Produce a software solution to meet the requirements of an enterprise, demonstrating its quality through effective design, efficient implementation, and sound testing

#### Assessment & Grading Criteria:

1. Demonstrated proficiency in the knowledge and understanding of software engineering;
2. Demonstrated proficiency in the knowledge and understanding of modern software development techniques
3. Demonstrated the ability to select appropriate techniques and tools to plan, design, implement, test and manage a software project, utilising appropriate software engineering methods, data structures and algorithms, modelling techniques and appropriate user interfaces
4. Produced a software solution that meets the requirements of an enterprise, demonstrating its quality through effective design, efficient implementation and sound testing

#### Assessment Brief.

See the attached brief.

## Assessment Criteria

## Level 5 Module Grading Criteria

Assessment category	Pass mark, demonstrating achievement of all associated learning outcomes				Marginal fail	Fail	
	D: 70% – 100%	M: 60% – 69%	GP: 50% – 59%	P: 40% – 49%	35% – 39%	20% – 34%	< 20%
<b>Knowledge and critical understanding of well-established concepts and principles of the subject(s)</b>	High quality work showing detailed understanding of the established concepts and principles of the subject(s).	Work of solid quality showing competent and consistent understanding of the established concepts and principles of the subject(s).	Adequate work showing understanding of the established concepts and principles of the subject(s), but lacking depth and breadth.	Simple factual approach showing limited understanding of the established concepts and principles of the subject(s). Narrow or misguided selection of material, with elements missing or inaccurate.	Weak work showing limited but fragmentary understanding of the established concepts and principles of the subject(s), for example through inaccuracies, inclusion of irrelevant material and/or absence of appropriate information.	Unsatisfactory work showing weak and flawed understanding of the established concepts and principles of the subject(s), for example through serious inaccuracies, inclusion of a significant amount of irrelevant material and/or absence of appropriate information.	Highly unsatisfactory work showing major gaps in understanding of the established concepts and principles of the subject(s). Inclusion of largely irrelevant material, absence of appropriate information and significant inaccuracies.

<b>Cognitive and intellectual skills</b>	Use of excellent critical, analytical and evaluative skills in order to develop highly logical and coherent judgements / arguments, supported by a range of relevant evidence. Evidence of independent thinking and creativity. Explicit discussion of other stances and a strong awareness of the limits of their knowledge.	Use of sound critical, analytical and evaluative skills in order to develop logical and coherent judgements / arguments, supported by relevant evidence. Some evidence of originality. Demonstrates a sound awareness of other stances and a good awareness of the limits of their knowledge.	Evidence of use of evaluation and critical analysis to support the development of logical and coherent judgements / arguments, supported by relevant evidence. An awareness of other stances and of the limits of their knowledge.	Limited and inconsistent use of evaluation and critical analysis to support emerging judgements or arguments, although not always logical or coherent and with inaccuracies. Limited awareness of other stances and the limits of their knowledge.	Largely descriptive work, with very little effort made to use evaluation and critical analysis to develop judgements or arguments. Information accepted uncritically, with unsubstantiated opinions evident.	Descriptive work with no effort made to use evaluation or critical analysis to develop judgements or arguments. Views expressed are often illogical, invalid or irrelevant. Minimal or no use of evidence to back up views.	Work is largely irrelevant or inaccurate, characterised by descriptive text and unsubstantiated generalisations. Minimal or no use of evidence to back up views.
<b>Application of theory to practice (for courses with a professional practice element)</b>	Excellent application of theory to practice, with the student making highly appropriate, developed and articulated links between the two.	Sound application of theory to practice, with the student making appropriate, well-developed and articulated links between the two.	Consistent and accurate application of theory to practice, with the student making appropriate links between the two.	Relevant theoretical knowledge and understanding applied in practice, but with students not always making logical links between the two.	Limited understanding of the application of theory to practice, with the student often not making appropriate links between the two.	Weak understanding of the application of theory to practice, with only occasional evidence of the student making appropriate links between the two.	Very weak theoretical knowledge and understanding, with no evidence of appropriate application in practice.

<b>Reading and referencing</b>	Critical engagement with a wide range of relevant reading, including research-informed literature where relevant. Consistently accurate application of referencing.	Engagement with a wide range of relevant reading. Sound application of referencing, with no inaccuracies or inconsistencies.	Engagement with an appropriate range of reading beyond essential texts. Referencing may show minor inaccuracies or inconsistencies.	Evidence of reading, largely confined to essential texts, but mainly reliant on taught elements. Referencing may show inaccuracies and/or inconsistencies.	Poor engagement with essential texts and no evidence of wider reading. Heavily reliant on taught elements. Inconsistent and weak use of referencing.	Limited evidence of reading and/or reliance on inappropriate sources. Limited engagement with taught elements. Very poor use of referencing.	No evidence of reading or engagement with taught elements. Absent or incoherent referencing.
<b>Presentation, style and structure *</b>	Highly effective presentation of work that is coherently structured and clearly expressed throughout.	Competent presentation of work in terms of structure and clarity of expression.	Work is structured in a largely coherent manner and is for the most part clearly expressed.	Ordered presentation in which relevant ideas / concepts are reasonably expressed.	Work is loosely, and at times incoherently, structured, with information and ideas often poorly expressed.	Work is poorly presented in a disjointed and incoherent manner. Information and ideas are very poorly expressed, with weak English and/or inappropriate style.	Work is extremely disorganised, with much of the content confusingly expressed. Very poor English and/or very inappropriate style.

\* Work that significantly exceeds the specified word limit may be penalized.

**Assignment Brief:**

1. This assignment *normally* involves the creation of a software solution to the enterprise needs explored in the database module. Thus if your database was an ecommerce database, you would develop the back-end logic, database interface and front-end interface in the assignment for this module. Your interface may be web-based or desktop-based.
2. If you are repeating the module from last year (when it was known as Software Design Development and Engineering), you will be exceptionally allowed to reuse materials and the project concept from last year.
3. Now consider the needs of the enterprise from the perspective of its business logic and what are the services it provides to the users/customers. Use sound requirements elicitation and modelling techniques to identify, analyse and record the functional and non-functional requirements of a software system to satisfy the enterprise needs. Create a **requirements specification** document with appropriate requirements models (which may include any or all of use cases, activity diagrams, class-based modelling, CRCs, control flow, data flow and/or behavioural models) as you deem appropriate to render the requirements explicit. Justify chosen requirements and modelling choices with appropriate rationale.
4. Using your requirements specification, create a **design** that satisfies the requirements. You will find it helpful to stepwise refine the diagrams and models produced in the requirements specification into a design-level representation of the classes, methods and other components required for later implementation (this may include UML diagrams, flowcharts, decision tables pseudocode, etc.)
5. Using your created design, implement it as executable **source code** using appropriate modern object-oriented and procedural programming techniques for classes, methods and other components.
6. Design, develop and undertake a suitable **set of tests** to demonstrate that your implemented code satisfies the requirements specification and functions as expected.
7. You should submit a **requirements specification, a set of design documents, the final source code, your test set and test results** (including screenshots of correctly and incorrectly functioning code and features) – note these are **FOUR** separate elements.
8. **In addition**, you should write a **reflective essay** of up to 1,000 words discussing and justifying your approach to the task in terms of key requirements, design, implementation and testing techniques. This is the **FIFTH** element of the assignment.
9. If your essay exceeds the 1000 word count by up to 10% then there will be no penalty applied. Submissions that exceed the word count by more than 10% will be applied a fixed penalty of 5 percentage points (i.e., 5 marks). In all cases, the penalised mark will not be reduced below a pass level, assuming the work merits a pass. Tables, diagrams (including associated legends), appendices, reference lists, tables of contents, footnotes, and endnotes are excluded from the word count however should be used appropriately. It is for the Module Leader to decide if there is an excessive or inappropriate use of components excluded from the word count.
10. Ensure that your work (submitted electronically via the online submission portal) is bundled into a zip with a filename matching the pattern sXXXXXX-SDDEAssign2.zip where sXXXXXX is

your UoS userid. Ensure that all documents are marked with your UoS userid. Your name should not appear anywhere in any of the files.