

Assessment Specification

Module Details

Module Code	UFCFTR-30-3
Module Title	Distributed and Enterprise Software Development
Module Leader	Mehmet Aydin
Module Tutors	Mehmet Aydin, James Lear, Desmond Case, Khoa Phung, Dilshan Jayatilake
Year	2024-2025
Task	Task 2
Total number of assessments for this module	2
Weighting	60%

Dates

Date issued to students	10.02.2025
Date to be returned to students	22.05.2025
Submission Date	08.05.2025
Submission Place	Blackboard
Submission Time	14:00
Submission Notes	

Feedback

Feedback provision will be	Formative feedback will be provided through in-class standup sessions, while summative feedback will be provided verbally and via BB.
-----------------------------------	---

Contents

Module Details.....	1
Dates.....	1
Feedback	1
Contents	2
Section 1: Overview of Assessment.....	3
Section 2: Task Specification	4
Section 3: Deliverables	5
Section 4: Marking Criteria.....	5
Marking Criteria.....	6
Section 5: Feedback mechanisms.....	6
Section 6: Appendices	7
6.1 Completing your assessment.....	7
6.2 Assessment Content	8
6.3 Assessment Offences	8
6.4 Use of Generative AI (ChatGPT or similar)	8
6.5 Guidance on Referencing (inc AI):	9

Section 1: Overview of Assessment

This assignment assesses the following module learning outcomes:

- MO1. Evaluate distributed and parallel computing concepts and paradigms with their legal, ethical, social and professional implications in developing large scale software systems
- MO2. Describe and evaluate the concepts and paradigms of modern enterprise systems with particular focus on components, interfaces and services.
- MO3. Apply a current industrial project management approach when undertaking a software development project
- MO4. Identify security issues in distributed or enterprise level software systems in order to implement preventive measures.

The assessment is worth **60%** of the overall mark for the module and is a **group assignment** that requires to work in groups.

Broadly speaking, this is a group work with which students are to demonstrate their technical and teamwork skills. The group work involves large scale software development via *component development*, use of external *services*, *security*, *containerisation*, and *teamwork performance*. The students are asked to work together in group practicing modern project management methodologies such as SCRUM to demonstrate their skills as part of software development team. The contribution of each student will be assessed individually following the marking scheme released as part of this assessment brief.

The task is described in more details in Section 2.

Working on this assignment will help you to develop a prototype of enterprise systems through an agile project management methodology in which a working software is expected to be developed, tested and demonstrated. The assessment strongly relates to MOL3 and MOL4 by considering the implementation of the software-to-be with ethical and security aspects on board. The project will be done by a team of students applying sprint-wise agile approach practicing sprint planning, task delivery and sprint evaluation, while being in constant contact with customer (tutors are acting as customers), and regularly updating team-members via periodic standup meetings. It also relates to MOL1 and MOL2 with implementing the concepts and paradigms of distributed and enterprise systems including components, services and containerised deployment. Therefore, the assessment relates to all 4 module learning outcomes.

Students are advised to regularly attend the practical sessions to be in contact with tutors (i.e. customers) and be able to clarify ambiguities around the requirements etc.

Section 2: Task Specification

You are asked to design and implement a web-based software system (as groups of 5 people) using the features and functionality of the examples from the course books, the practical classes, demonstrations in the lectures or other materials referred to within the scope of the module.

The assessment **for standard CS route** is based on the case study detailed in the enclosed file, called **Case Study.docx**, as part of the bundle.

Likewise, the assessment **for AI route** is based on the case study detailed in the file called **Case Study MLAAS.pdf**.

Please pay attention to the specifications and list of requirements provided in the file.

You are expected to design and build your software system using **Python Django** framework following the MVC design patterns, (MVT of Django implementation) using a Relational database of your choice (e.g. MySQL, PostgreSQL etc.) for the backend. The system-to-be should be developed as a prototype of an enterprise software, which requires to be a distributed system, where the pieces (components) are required to be distributedly deployed on a number of containers (2 containers minimum, 3 or 4 ideally) created and run via docker.

Groups should also use **Docker** to containerise and deploy your application, where your DB server and your WebApp should be running on separate containers with different web addresses. Ideally, you are expected to deploy your frontline, your application, some of the functionalities you developed in the form of services using DRF (different from the other external services), and your DB on separate containers, which should be considered distributed.

As a team, you are required to demonstrate and discuss your working system using VS Code IDE. The team will be treated as a unit and any individual member may be requested to demonstrate complete knowledge of the system presented.

As part of the curriculum and assessment:

- (1) you will manage your group project with Agile methodology, practicing a SCRUM-like project management throughout of the term,
- (2) you will work through sprints starting from the forth (4th) week of Spring Term,
- (3) you need to practice **stand-up sessions** each week during the practical sessions. This is going to be part of the assessment, where your attendance and the quality of your verbal report during the standups will be recorded to contribute towards your individual mark.

You are required to develop your **web-based software system** in last three sprints of Spring Term as follows:

Sprint 1: Task plan and schedule of delivery. This sprint covers **Week 4, 5** and **6**, mainly requires (1) working out with planning and scheduling the tasks, (2) setting up a DB among the alternatives, creating all tables with their relations (3) authentication for most of the user types. A sprint review report for Sprint 1 should be uploaded on BB via File Exchange at the end of **Week 6**. The demo and assessment will be communicated during the practical session of **Week 7** as a face-to-face review; all members are required to attend. This sprint weighs 10% of the overall mark.

Sprint 2: Completion of backend architecture and partial completion of front-end user interface. This requires, at least, half of the tasks are completed. These tasks are expected to be completion of all model classes, some templates including Forms and building `views.py`. This sprint covers **Week 7, 8** and **9**, and be submitted at the end of **Week 9**, to be reviewed and

assessed on **Week 10** in the same way as previous sprint. This sprint weighs 30% of the overall mark.

Sprint 3: The group project is expected to be completed at the end of this sprint, which covers **Week 10, 11** and **12**. Final and complete delivery of the application will be demonstrated and assessed on **Week 12** and/or **13**. It carries 60% of overall mark.

Please note that *Week 4 of Term 2* refers to *Week 27* of university calendar, while *Week 12* of the term is *Week 39* in the university calendar.

Section 3: Deliverables

Item	Detail	Date & Submission Mechanism
Sprint 1 report	<ul style="list-style-type: none">Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.	07/03/25 File Exchange on Blackboard (BB)
Sprint 2 report	<ul style="list-style-type: none">Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.Demonstrating (<i>live or video</i>) completed tasks (10 mins).	28/03/25 File Exchange on Blackboard (BB)
Completed software	<ul style="list-style-type: none">Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.Complete project zipped and uploaded on BBDemonstrating the complete project (20 mins)Peer Assessment Form (Peer Assessment Form.docx).This form should be filled and confidentially submitted to tutors by everybody, while other deliverables are to-be in behalf of the groups.	08/05/25 Blackboard (BB) submission link

Section 4: Marking Criteria

This group project requires multiple delivery stages (Sprints), where each stage is marked independently. Sprint 1, 2 and 3 carry 10%, 30% and 60%, respectively.

Marking tools are designed into marking grids as provided in the file, called "**Marking Rubrics.pdf**" within the bundle, where each sprint is considered in a separate rubric with more specific criteria and expectations.

Marking Rubrics document also includes a marking table for identifying individual contributions. Individual contributions are broken down following inputs from each member of the groups via **Peer Assessment Form**, separately provided with the file, called "**Peer Assessment Form.docx**" included in the bundle.

The reported inputs by each group member is cross-checked with evidence provided via GitHub and other tools (to be) used.

You may note that 80% and beyond marks require outstanding excellence in the works, which are expected to exceed the requirements outlined in the Case Study document and should be genuinely exceptional.

Marking Process

As a team, you will be asked to demonstrate your developed software system in scheduled slots as applicable for each sprint where appropriate. For Sprint 3:

- You must download (from Blackboard) and unzip your project.

Your VS Code project and your DB server must be runnable on the standard CATE configured laboratory machines (or the same configuration on your own laptop which you may like to use).

- It is your responsibility to attend scheduled classes – failure to demonstrate your system in class will be treated as a non-submission.
- All group members will be awarded the same mark as long as all group members agree that all contributed equally. Otherwise, your individual marks will be generated based on your individual contributions identified based on collected and provided evidence alongside your stand-up performance. Everybody must confidentially fill and submit “**Peer Assessment Form**” for collecting inputs to determine individual contributions. A guiding scheme is included in the document called “**Marking Rubrics**”
- Any group member failing to take part in the demonstration will be assessed as a non-submission and given zero marks.
- Each group is expected **to attend at least 6 stand-up** sessions. This will be considered when individual contributions are calculated.
- The 40% of the mark for final demo will come from software performance with a number of selected test-cases
- The remaining 60% comes from the quality of the code, meeting the technical requirements, and the quality of your answers to technical questions which you will be asked to explain how you implemented the key principles into your code within the context of the enterprise software systems.
- The quality of your verbal expression in this demonstration is important – incoherent explanations will not achieve high marks. Please be advised that demonstrations last for a fixed durations, so be prepared to concisely demonstrate and explain your system.

Section 5: Feedback mechanisms

Formative verbal feedback will be provided during practical sessions through standup sessions for the group work, and after each sprint through evaluating sprint review forms. Written feedback will also be provided for each sprint review form submitted on BB.

Summative feedback will be provided following the demonstration and presentations.

Section 6: Appendices

6.1 Completing your assessment

Where should I start?

Students are expected to start this coursework with meeting the group members. No groups of friends are allowed in order to practice expected industrial standards. The following steps would be useful to consider:

1. Meet up with all group members, read the specs.
2. Read through [Case Study](#) file and start eliciting functional and non-functional requirements under the light of experiences gained from previous term, Autumn Term. Note that students taking the module UFCFUR-15-3 Artificial Intelligence will have a separate Case Study ([Case Study MLAAS](#)) which is in line with the assessment brief of that module team.
3. Make up "Product Backlog" from elicited requirements in the form of use cases or user stories.
4. Develop a rough plan for the complete project through 3 sprints, and then develop the action plan for the Sprint 1,
5. Deliver tasks planned as part of Sprint 1 and submit sprint report for evaluation; use [SprintReportTemplate.docx](#).
6. Plan Sprint 2 and deliver the tasks planned for Sprint 2, the submit sprint report,
7. Demo Sprint 2 results,
8. Plan the final sprint, Sprint 3, deliver all tasks to completion,
9. Submit the project as a compressed file to BB and confidentially submit peer assessment form using the template provided with [PeerAssessmentForm.docx](#).

What do I need to do to pass?

Reasonable contribution to the group project, which produces a working product that meets minimal requirements. These will be eventually discussed and released in due course.

How do I achieve high marks in this assessment?

Equal contribution to the teamwork that produced a working product delivers all requirements in a very/excellent/outstanding quality as described in making scheme. In order to achieve 1st class or 2:1, proper attendance to participate group-work and significantly contribute to the development.

How does the learning and teaching relate to the assessment?

All lectures and practical sessions are very relevant to this assessment. Particularly, attendance to practical sessions from Sprint 1 onward is a requirement since all individuals need to attend at least 6 stand-ups as part of individual contributions.

Students need to beware of that the final demonstration of the developed software will include Q/A to group members to assess the level of comprehension and understanding.

What additional resources may help me complete this assessment?

All relevant documents will be shared on Blackboard module site.

What do I do if I am concerned about completing this assessment?

UWE Bristol offer a range of Assessment Support Options that you can explore through [this link](#), and both [Academic Support](#) and [Wellbeing Support](#) are available.

For further information, please see the [Academic Survival Guide](#).

6.2 Assessment Content

In line with UWE Bristol's [Assessment Content Limit Policy](#) (formerly the Word Count Policy), word count includes all text, including (but not limited to): the main body of text (including headings), all citations (both in and out of brackets), text boxes, tables and graphs, figures and diagrams, quotes, lists.

6.3 Assessment Offences

How do I avoid an Assessment Offence on this module? ²

Use the support above if you feel unable to submit your own work for this module.

Please make effort to avoid similarities in your logbook and individual reports. Similarity of UML diagrams is not an issue.

UWE Bristol's [UWE's Assessment Offences Policy](#) requires that you submit work that is entirely your own and reflects your own learning. It is important to:

Ensure that you reference all sources used, using the [UWE Harvard](#) system. Use the guidance available on [UWE's Study Skills referencing pages](#).

Avoid copying and pasting any work into this assessment, including your own previous assessments, work from other students or internet sources

Develop your own style, arguments and wording. Avoid copying and changing individual words but keeping essentially the same sentences and/or structures from other sources

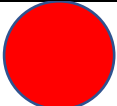
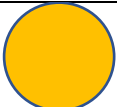

Never give your work to others who may copy it

If you are doing an individual assessment, develop your own work and preparation. Do not allow anyone to make amendments to your work (including proof-readers, who may highlight issues but not edit the work).

When submitting your work, you will be required to confirm that the work is your own. Text-matching software and other methods are routinely used to check submissions against other submissions to the university and internet sources. Details of what constitutes plagiarism and how to avoid it can be found on UWE's Study Skills [pages about avoiding plagiarism](#).

6.4 Use of Generative AI (ChatGPT or similar)

Generative AI cannot be used for authoring the report and the logs, but the final version of Individual Report can be proofread. If this has been done, it must be declared and acknowledged.

	Generative AI must not be used in this assessment, because students' evaluation and critical review skills on software development technologies are assessed.
	You can use Generative AI in this assignment for checking spelling, grammar etc.
	No touch of Generative AI in authoring the report and the logs

6.5 Guidance on Referencing (inc AI):

Please note that the aim of referencing is to demonstrate you have read and understood a range of sources to evidence your key points. You need to list the references consistently and in such a way as to ensure the reader can follow up on the sources for themselves.

[Referencing - Study skills | UWE Bristol](#)

[Using generative AI at UWE Bristol - Study skills | UWE Bristol](#)