

Main Project Guidelines

1. Overview of Assessment

The assessment is worth **60%** of the overall mark for the module.

This is a group assignment, which requires **4/5 people** to form a group for software development and demonstration.

Working on this assignment will help you to develop your technical development skills and your teamwork skills, which aims to improve your employability level by providing a practice of real working environment.

2. Task Specification

You (as a group/team of 5 people) are asked to design and implement a web-based software system using the features and functionality of the examples from the course materials, 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 SmartCare case study, which is detailed in the enclosed file, called **Case Study.pdf**.

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 retrieve a list of requirements provided in the case. You may see a list of hard requirements included in the case brief.

You should 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.

You 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 will be required to demonstrate and discuss your working system using VS Code IDE. Your team will be treated as a unit and any individual member may be required to demonstrate complete knowledge of the system you are presenting.

As part of our curriculum and assessment:

- (1) you will manage your group project with Agile methodology,
- (2) you will be practicing a SCRUM-like project management throughout of the term;
- (3) you will work through sprints starting from the first week of Spring Term,

- (4) you need to practice **stand-up sessions** each week during the practical sessions. This is going to be part of the assessment, where each individual's attendance will be recorded and contributing towards your individual mark.

You are required to develop your **web-based software system** through the following three sprints:

Sprint 1: This sprint starts from the first week of Term 2 covering **Week 13, 14** and **15**. You are expected to work out through analysis and design stages to elicit the requirements and make up Product Backlog. You will try to practice sprints with sprint planning, task delivery and sprint review. Your sprint review report for Sprint 1 should be uploaded on BB via File exchange at the end of **Week 15**. Your group effort will be evaluated through Sprint Report, and formative feedback will be produced accordingly. There will be **no mark** for this sprint.

Sprint 2: Task plan and schedule of delivery. This sprint covers **Week 16, 17** and **18**, 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 the majority of the user types. Your sprint review report for Sprint 2 should be uploaded on BB via File exchange at the end of **Week 18**. Your group effort will be assessed through Sprint Report, and feedback will be produced accordingly. This sprint weighs **10%** of the overall mark.

Sprint 3: 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 19, 20** and **21**, its review is required at the end of **Week 21**, a demo and sprint report is required. Your group effort will be assessed through the sprint report and a short demo at the end of this sprint. Demos will be scheduled on **Week 21**. This sprint weighs **30%** of the overall mark.

Sprint 4: The group project is expected to be completed at the end of this sprint, which covers **Week 22, 23** and **24**. Final and complete delivery of the application will be demonstrated and assessed after the submission deadline. It carries **60%** of overall mark.

3. Deliverables

You are required to submit your group work Sprint-by-Sprint as described above and with more detailed in Case Study document. The final version of the work must be submitted through the link to be provided on Blackboard VLE as a zipped project.

Sprint-wise Deliverables:

Sprint 2:

- Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.

Sprint 3:

- Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.
- Demonstrating completed tasks (10 mins).

Sprint 4:

- Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.
- Complete project zipped and uploaded on BB
- Demonstrating 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.

4. Marking Criteria

This group project requires multiple delivery stages (Sprints), where each stage is marked independently. Sprint 2, 3 and 4 carry 10%, 30% and 60%, respectively.

Marking tools are designed into marking matrixes as provided in the file, called “[Marking Matrix.pdf](#)” within the bundle, where each sprint is considered in a separate matrix with more specific criteria and expectations.

Marking Matrix 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 4:

- 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 Matrix**”
- 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 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.

5. Feedback mechanisms

Feedback will verbally be provided immediately after the demonstration at each stage (Sprint). On your request, you can receive group-based and personalised feedback, written or verbally.