

COM6103:Team Software Project

Assignment Requirements

This document specifies the requirements for you to complete the assessment for this module.

1. Overview

The assignment is the only assignment for the module and is worth 100% of the marks; please see the Blackboard site for the exact deadline. The final report submission is electronic via Blackboard and you will also give a brief demonstration in the week before. Your team Gitlab repository and Blackboard Group Forum will also be viewed when needed.

Workload and Teams

The assignment is intended to account for >100 hours of each person's work towards the module as a whole, within a team of 5 to 7 (as is typical in industry).

It is important that you record each team member's contributions throughout your sprints and summarise this information in your final report (you may also include further detail in an Appendix). It is required that each person not only contributes to each stage of the assignment, but also shows evidence of their contribution, e.g. through Gitlab commits, issues, etc. Allocating work within the team is the responsibility of the individuals and the team. Buddy Check will also be used for Peer Review and to adjust individual marks within teams; this will not raise any individual's mark by more than 10%, but may lower marks by more than 10%.

Deadlines

The deadline is absolutely fixed and you should note that group assignment extensions are only given in very exceptional circumstances that affect every member of a group. You should therefore aim at finishing your work at least a few days before the deadline – do not leave it until the deadline, in case anything goes wrong, and you then find that you are late.

Note that the Computer Science department applies [fairly severe penalties for handing late assignments](#). N.B. All team members are responsible for ensuring that the team's work is submitted on time; any late penalties will always apply to the whole team.

Material Provided

Please see Lecture material, lab class examples and resources on Blackboard, which focus on the methodology/process to be used for your team work.

External sources

The work submitted **MUST BE YOUR OWN WORK**. If you include code from external sources, then you **must** reference other's work; for which you will **earn zero marks**.

You may use third-party libraries that **support** your software development. If you use libraries (or frameworks) that **replace** a substantial portion of the team's software development, then you will find it much harder to show evidence of your added value and likely earn little, if any, marks for the resulting work, e.g.

- using React or Flask is acceptable
- using Django (a CMS - Content Management System/Framework) carries a substantial risk, since it can mean that you are unable to show evidence of adding (software development) value. This is why Flask is recommended for Python.

If in doubt, please ask **early** on the assignment forum in Blackboard.

2. Requirements

As a team, you are required to develop software to the specification of a client. The initial project briefs are provided through blackboard. You will meet with your Client (typically arranged by yourselves) to better understand the requirements, identify hidden (or less obvious) requirements and to allow you to make your own, client approved, changes and proposals.

3. Assessment areas

The following areas/criteria will be used to assess your ability to work in a team to build software to the specification of a client:

- 3.1 Teamwork (30%)
- 3.2 Documentation (40%)
- 3.3 Delivered software (30%)

The **learning objectives** of this assignment are to 'learn by doing' **AS A TEAM** and you will need to show evidence that you have:

- considered the user and usage through design, e.g. by showing hand drawn sketches
- worked together as a team including upskilling team members where possible and appropriate

3.1 Teamwork : 30% Weighting

For teamwork, we want to understand:

- How you organised **your team** and the tools and processes you used
- How **the team** communicated throughout the project and ensured members were supported and (when relevant) upskilled
- How well the workload was balanced across team members; *n.b. beware of **excluding** team members*

To assess this, the following artefacts/evidence will be reviewed

- Final team report
- Weekly Team Adviser meeting notes
- Feedback/notes from Client meetings
- GitLab Issues Board, and commit history

3.2 Documentation : 40% Weighting

Here you should be showing

- how your team carried out its projects
- how you adequately describe/present the solution developed to the client - including how it could be used in your absence

A large part of the team documentation mark will be based on the quality of your team report, the extent to which it details all the requested information, including changes to stories and basic information about using your system.

Team documentation will be graded using a number of mediums which include

- Final team report
- User guide (Gitlab)
- Setup guide (Gitlab, inc. deployment instructions)
- Test results; preferably automated

3.3 Delivered Software : 30% Weighting

You should show how well your developed and delivered software meets the requirements.

Particularly, we are interested in answering the following questions

- How usable is the product?
- Does it meet the client's requirements?
- Is the product developed to specification?

The software will be graded using a number of mediums some of which include

- The basic organisation of your code into units, and its overall structure
- The amount of work completed to date, and the extent to which you are on track to delivering a good quality product
- Client Presentation given in Week 10
- Basic functional test of the product

Note: The last commit in the teams' GitLab repository before the final report deadline will be considered as the final deliverable. You must ensure that last-minute changes/commits do not cause defects in your software. If for any reason you would like a different commit/tag in the repository to be graded, it must be specified in the report.

N.B. Do not send files after the deadline that you forgot to commit.

Please note the following important points in regard to the use of the Git repository:

1. This must be the University of Sheffield Team Gitlab repository and no other will be accepted. Ensure you commit to your team repository - not your personal repository
2. All members should be making regular commits. In marking your work, we will check that you have been making regular commits, not just a few 'big' commits or a sudden rush just before the deadline; both of these would fail to satisfy the learning outcomes of the module.
3. You should attempt a clean install/run before your demonstration and your final commit

4. Client Demonstration

In week 10 during the session, you will give a 10 minute demonstration of your app to show it working. You are strongly encouraged to time yourself beforehand. This is an opportunity for you to demonstrate your application working. Note: You should have more than one laptop/device ready to demonstrate on. You will also be given some feedback which you may then use to improve your final submission (report).

You will likely find that some parts of your app will only be visible through your report; also some may be added/updated after the demonstration.

Your client will be considering many aspects of your demonstration including requirement completeness, interface and interaction suitability, reliability and the quality of the solution; a simplistic (purely functional) solution will attract fewer marks.

5. Final Team Report

As evidence of progress and how the team self-managed, all teams are required to **submit a 10-18 page report (max ~8,000 words)** through Blackboard.

In this report, you should describe:

- how your team carried out the project, i.e. the process you followed
- What programming language(s) were used?
- what features were delivered
- what features you could/did not deliver?
- what challenges you faced as a team

- how you tested the software
- how you communicated within your team.

As a minimum the report should contain, in this order

- **Cover sheet with Team name and list of members**
- **Introduction/background to the project**
- **Project scope and objectives**
- **Product backlog** – A list of user stories that make up the complete product
- **Analysis & Design** - System architecture, UML diagrams, Algorithm/Database design.
- **Evidence of Testing** - Test plan, Test documentation and Test results
- **Team management & communication**
 - Describe how you managed your team, who did what?
 - Describe how you communicated as a team and challenges that were involved (if any)
 - Include two sample meeting minutes (you can include more in an Appendix if you wish). Ideally, these should be sprints 2 and 3
- **Planned & Completed Features**
 - Present the plan, including prioritisation, and completed user stories in each iteration
- **Uncompleted Features**
 - What features were not completed and why?
- **Screenshots of relevant pages**
- **Conclusion**
 - Ideally, this should include what was learnt, including how the team improved, challenges faced, and how you resolved challenges
 - You should also include a short section on proposals for future work, e.g. you might include the plan for a 4th Sprint
- **Appendix**
 - A. User guide (maximum 3 pages)
 - B. Setup guide (maximum 3 pages)
 - Other documents you think are relevant can be added here

Note - The Appendix is not included in the word count, but please do not include unnecessary items.

6. Assigning Individual Marks for Team Projects

Team marks will be scaled for each student in proportion to their individual performance. The scaling factor is derived from the scores assigned to you by each of your teammates in a peer review. We will ask you to score yourself and your teammates on a scale of 1–5, according to the following aspects:

- Attendance and punctuality (to group meetings, joint working sessions, etc.)
- Ability to work effectively with other team members

- Contribution to content and organisation of project deliverables
- Quality of contributions
- Timeliness of contributions

You will be expected to support the scores you give with explanatory comments.

The scaling factor is computed from these scores such that 1.0 corresponds to the average team member contribution and anything above or below 1 represents an above or below the average effort.

Please note the following:

- **Scores assigned to individuals and accompanying comments will be kept confidential.** We will disclose the factors, and where appropriate anonymised comments.
- **We reserve the right to manually adjust factors in the case of certain individuals or teams**, if we have reason to believe that scores have been assigned unjustly. This is department and University standard policy.

7. Assessment Criteria

Teamwork		Weighting: 30%
70%-100%	Excellent teamwork. The feedback from the team advisors and clients were excellent. The teamwork section of the final report provided excellent evidence of how the team self-managed and handled challenges. Very good references to the tools and processes used to ensure clear communication. The team's processes were explained in-depth and example evidence provided. The Gitlab repository shows even contributions from the team members.	
60%-69%	Good teamwork. The feedback from the team advisors and clients were good. The teamwork section of the final report provided good evidence of how the team self-managed and handled challenges. Good references to the tools and processes used to ensure clear communication. The team's processes were explained, and some example evidence was provided. The Gitlab repository shows almost even contributions from the team members.	
50%-59%	Average teamwork. The feedback from the team advisors and clients evidenced average teamwork. The teamwork section of the final report provided sufficient evidence describing how the team self-managed and handled challenges. Sufficient reference to the tools and processes used to ensure communication. The team's processes were presented, and some example evidence was provided. The Gitlab repository shows contributions from the team members were not even.	
40%-49%	Limited teamwork. The feedback from the team advisors and clients evidenced limited teamwork, only a few individuals did most of the work. The teamwork section of the final report provided limited evidence of team management and communication. The team's processes were not explained	

	in-depth, and limited evidence provided. The Gitlab repository shows only a few individuals did most of the work.	
Documentation		Weighting: 40%
70%-100%	The deliverables are complete and excellent. The final report presented a very good introduction, clearly written user stories, very good analysis & design, and very good evidence of testing. The setup guide and user guide are very well written, clear to read and easy to follow.	
60%-69%	Well presented and almost complete deliverables. The final report presented a good introduction, user stories, good analysis & design, and good evidence of testing. The setup guide and user guide are well written, clear to read and easy to follow with minor limitations.	
50%-59%	The deliverables are partially complete. The final report contained a partially complete introduction, partially complete user stories, partially complete analysis & design and partially complete testing documentation. The setup guide and user guide are sufficient but not easy to follow.	
40%-49%	The deliverables are limited. The final report contained limited introduction, user stories, analysis & design and test documentation. The setup guide and user guide are scant and not easy to follow.	
Delivered Software		Weighting: 30%
70%-100%	The overall product is excellent, can be used and adequately meets the client's requirements. The specification and priorities of the client have been taken into consideration. Many of the user stories are complete or almost complete. The client's feedback on the quality of the product is excellent. The demonstration shows the main functionalities of the software and does not exceed five minutes.	
60%-69%	The overall product is good, can be used and meets some of the client's requirements. Some of the specification and priorities of the client have been taken into consideration. Many of the user stories are complete or almost complete. The client's feedback on the quality of the product is good. The demonstration shows the main functionalities of the software and does not exceed five minutes.	
50%-59%	The overall product is partially complete when compared to the client's requirement. Not more than 50% of the specification and priorities of the client have been taken into consideration. Some of the user stories are complete but with many incomplete. The client's feedback on the quality of the product is that it is partially complete. The demonstration shows some functionalities of the software.	
40%-49%	The overall product is limited when compared to the customers' requirement. Many of the user stories are incomplete. The specification and priorities of the client have not been taken into consideration. The client's feedback on the quality of the product is that it is limited.	

Submission

The final report must be submitted in **PDF format** using Blackboard (MOLE).

The submission deadline is **3pm on Thursday 9th May 2024**.

The team is responsible for ensuring their work is submitted on time. The [standard penalties for late submission of work apply](#).

Unfair Means

This is a **team** assignment and you must **not** collaborate with other teams.

The [standard rules concerning unfair means apply](#).

Questions

If you have any questions concerning what is required by this assignment please post on the Blackboard Discussion Board: the Assignment Forum, or exceptionally, i.e. if the question is personal, email them to a.stratton@sheffield.ac.uk