Not for public distribution
Document adopted and modified from ProjectManagementDocs.com

BUSINESS CASE: MOSS GUI (PROGRAMMING EDUCATION)

TEAM RNG
[redacted], **ISAAC BASQUE-RICE,** [redacted], [redacted], [redacted], [redacted], **AND** [redacted]

ABERTAY UNIVERSITY
DUNDEE, DD1 1HG

DATE: 14/12/2021

Team Member	Sections
[redacted]	1-3
[redacted]	4-8
Isaac Basque-Rice	9-12 + Appendix A – My Section with Grey Highlight
[redacted]	13-16 + Appendix D
[redacted]	17-19 + Appendix B
[redacted]	20-22 + Appendix C
[redacted]	Project Charter (separate)

1. Executive Summary

This business case details the approach taken to address concerns from Dr Suzanne Prior, a lecturer at Abertay University, about the current process taken to check plagiarism within student submitted code. Dr Prior has approach team RNG in creating software that is highly efficient to decrease time taken to mark submission as much as possible.

The business case aims to explain why team RNG, and the chosen approach are the best suitable candidates for the task in creating a solution for the issue.

A section detailing what the teams' recommendations and justifications for the project are and aims to clearly state what is believed to be the best possible approach to take to tackle the issue and the justifications for these recommendations.

All possible risks and costs that could occur throughout the development and implementation of the project have been considered and detailed plans on how these will be managed and reduced by the team during development are presented within this document.

Using factors such as efficiency, simplicity, cost and other factors, a detailed analysis of different possible approaches that could be taken to address the issue was conducted and why alternative approaches were not chosen.

A detailed overview of the development timeline has been created with clear targets for each milestone in development. This overview details when development of each part of the project is expected to be started and completed.

1.1. Issue

Dr Suzanne Prior, a cybersecurity lecturer at Abertay University, who is interested in early-stage programming education, has issues with plagiarism within programming assignments being handed in within these early stages. Unlike written assignments, which already have automatic plagiarism detection tools (such as TurnItIn), programming code does not have any such tools. As plagiarism increases within these early-stages the need for such tools has become highly desired by lecturers. Currently the lecturer must manually download assignments, unzip them, and then run them through the MOSS checker tool, which is a command line tool that submits to an external server (and as such has significant workflow issues), to detect possible plagiarism.

1.2. Anticipated Outcomes

With the development of a web-based automatic plagiarism detection tool for code being handed in will help reduce lecturer workload by automating the currently manual task. With the creation of a web-based tool that automates the process of checking for plagiarism it is hoped that the time taken by lecturers to mark student code is greatly reduced therefore reducing lecturer workload and stress. With the software designed to be as efficient as possible less time will be spent on checking code so more time can be spent on creating more personalised feedback for students. Such tools will also increase fairness in marking as it makes plagiarism easier to detect, ensuring cheating students are caught at a higher rate, this will improve student trust in the marks being handed out.

1.3. Recommendation

With the given problem, the development of a solution could be approached in many ways. After much discussion within the development team as well as the client, the approach described within this document was chosen as the best applicable to ensure optimal cost, efficiency, and ease of use. The recommended solution will replace the current manual method of checking code for plagiarism with new web-based, easy to use, web application which will automate the process. The implementation of this new system will greatly reduce workload for lecturers marking submitted assessments by automating the plagiarism checking stage of marking. The use of a web-based application allows the use of the checker on any device that the lecturer may try to access it on.

- Multiple files can be uploaded and checked at a time
- Web application allows for easy integration and use across different devices
- Increases efficiency of plagiarism checking
- Decreases lecturer workload by simplifying process

1.4. Justification

The use of a web application was chosen over the alternatives due to its ability to be accessed and used on any machine the user desires without the need to download any files. It will also make sending files to the external MOSS system a lot simpler, faster, and more efficient as it will allow multiple files to be uploaded and checked at once. Other possible solutions would either require external files to be downloaded onto each device using the tool or would only allow single files to be uploaded at once. With

other solutions that would require external files to be installed this would end up costing extra time for markers which in turn would negate the time saved from having a new system, so these solutions were discarded. The decrease in time for solutions only able to upload single files at a time would be almost negligible so therefore these options were also discarded. The solution chosen would allow for the greatest decrease in time taken to mark submissions as well as the greatest ease of use.

1.5. Team Experience

The team has in depth experience in creating web-based applications. The team also has knowledge with scripting in the python programming language. We therefore believe we are best suited for the task in hand. Any questions or enquiries can be directed towards our team lead, Isaac at: 1901124@uad.ac.uk

2. Business Case Analysis Team and Stakeholders

2.1. Business Case Analysis Team

The following individuals comprise the business case analysis team:

Role	Description	Name/Title
Project Manager	Manages the business case and project team	Isaac Basque-Rice, Process Team Lead

2.2. Project Team

The following individuals comprise the project team:

Role	Description	Name/Title
Project Manager	Manages the business case and project team	Isaac Basque-Rice, Process Team Lead
Designer	Mainly responsible for the design of the graphical user interface used by staff	[redacted], Project Team
Programmers	Mainly responsible for coding the system	[redacted], [redacted], [redacted]
Testers	Mainly responsible for system testing and bug fixing	[redacted], [redacted]

2.3. Client and External/Internal Stakeholders

We have identified the following stakeholder:

Role	Description	Name/Title
Client (Abertay University)	Sponsor for project	Dr Suzanne Prior, Cybersecurity Lecturer

3. Problem Definition

3.1. Problem Statement

Dr Suzanne Prior, a lecturer for the school of design and informatics at Abertay university, has detailed a problem with the current process for checking for plagiarism in student submitted code. Currently for written assessments the process is automated with a tool embedded in MyLearningSpace called TurnItIn, however there is no such tool for code being submitted. Currently a tool called MOSS, developed by Stanford University, is used to check code for plagiarism, but use of this tool requires lecturers to manually upload each file that needs check. This process of manually uploading each file is very slow and inefficient and in turn increases the time it takes to mark each student's submission. Due to this process taking up time, lecturers may rush other parts of the marking process, such as taking care with looking over the quality of the work submitted or giving detailed and personalised feedback, to meet their marking time targets. The team has been told by Dr Prior that currently it can at most take a full day to mark submissions from only one student.

3.2. Organizational Impact

With the MOSS GUI project being the first tool of its kind to be available to lecturers, it will positively impact the way lecturers mark and evaluate code that has been submitted. The following section describes some of the impacts this project will have.

Tools: Since there are not currently any automated tools available to lecturers to check code for plagiarism, this project will give lecturers a new much simpler and more efficient way of checking student submitted code for plagiarism this will greatly improve speed and quality of marking. As the software aims to be a web-based application, it will be possible for lecturers to easily access the software at any place or device they decide to use.

Process: The new web-based application will completely change the current process, which currently requires lecturers to manually unzip and check each file individually through MOSS and replace it with an automated process that allows lecturers to upload and check multiple files at once.

Software: Since there is not currently any software available to lecturers to check code for plagiarism, there will not be a phasing out period of old systems. This allows the project to be used straight away by lecturers once it's development and implementation has been completed.

Staff: As any company, Abertay University aims at ensuring lecturers are not overworked and overstressed as this can cause many issues for their staff such as burnout and decrease in moral within its staff. This project aims at decreasing workload for staff which helps decrease stress placed on staff when trying to meet targets for getting submissions back to students.

Training: As the project aims to be as simple as possible no training on how to use the software will be needed for lecturers. Therefore, no extra cost will be placed on the university to create and provide training.

3.3. Technology Integration and/or Migration

Due to the lack of a current system in place, the new web-based application will be able to be put online immediately for instant use by the lecturers. No data will need to be migrated as none exists. This means integration of the software will be immediate on the system. Due to the lack of need for training, as the system is designed to be as easy to use as possible, there will not need to be a period for training lecturers on how to use the new software.

4. PROJECT OVERVIEW

The MOSS GUI project overview describes the current problem this project will address. It consists of a project description, goals and objectives, performance, assumptions, constraints, and major milestones of the project. As the project is approved and moves forward, each of these sections will be expanded to include a greater level of detail in working toward the project plan.

4.1. Project Description

As detailed in the project brief from the client, plagiarism in programming code is an ongoing problem in the academic environment. It's difficult to check for this plagiarism in an easy way. This project will be delivered using the incremental model and completed once the tool has met the client's needs with a simple plagiarism checker program.

The project will consist of checking programming code for this plagiarism to hopefully combat this issue. Our team believe the best approach to solving this business problem is through a web application. Using a web app, the programming code will be submitted in a zip file to the plagiarism checker tool, MOSS, with an easy graphical user interface. No command line utility will be needed anymore to use MOSS. The MOSS program will check this code and highlight any potential cases of plagiarism. However, to improve productivity and ease of use, these results will be displayed directly to the project's web app. Navigating to MOSS's results website will not be necessary anymore.
This project will result in reduced plagiarism in code in the academic environment whilst providing an easy interface to do so. It aims to remove the hassle that comes with using the MOSS program currently and meeting the client's goals to achieve maximum efficiency.

4.2. Business Goals and Objectives

The MOSS GUI Project directly supports several of the corporate goals and objectives established by the client. The following table lists the business goals and objectives that this project supports and how it supports them:

Business Goal/Objectives	Description
Easy to use GUI	Web based tool will allow an easy interface to submit programming code.
Improve staff efficiency	Staff will be able to submit programming code fast and easily, with very few steps involved compared to the current method.
Display MOSS output to web app	Instead of making the client go to MOSS's web page for the output, we will display this output on our website after submission.
Compatibility	The web app will be easily available on most devices with an internet connection.

4.3. Project Performance

The following table lists the key resources, processes, or services and their anticipated business outcomes in measuring the performance of the project. These performance measures will be quantified and further defined in the detailed project plan.

Key Resource/Process/Service	Performance Measure
Time	The web app will reduce time taken to upload programming code to check for plagiarism. This will be achieved by the easy user interface to upload and send the code to MOSS.
Efficiency	Elimination of multiple steps to generate a plagiarism report for the code. This will be achieved by displaying the report on the same submission web page.
Accessibility	Reduction of requirements needed to access and use the MOSS tool. The MOSS tool will be accessible in a web browser. Staff will not need programming languages/software to use MOSS.

4.4. Project Assumptions

The following assumptions apply to the MOSS GUI project. As project planning begins and more assumptions are identified, they will be added accordingly.

- Stanford University will provide their MOSS tool to the project team for use in this project.
- The project won't need more resources provided by first-year costs of the hosting plan.
- Funding is available to continue hosting the web app with DigitalOcean if need be.
- Funding is available for purchasing hardware/software for programming the web app.
- All department heads will provide necessary support for successful project completion.
- Project has continued support and backing from the client.
- The project team will follow the incremental methodology as planned until the project is complete.
- Finally, the project plan addressed in this document will stay the same.

4.5. Project Constraints

The following constraints apply to the MOSS GUI project. As project planning begins and more constraints are identified, they will be added accordingly.

- As implementation will be done by our internal team and not by the product developers, there will be limited support from MOSS.

- A money constraint is the web app hosting provider is offering our team $100 free resources to host the web app. If the project requires more time or resources to host the web app this might be difficult. Although the project team is confident this will be sufficient, the client should be aware that costs for the project would increase if the project requires more hosting resources.
- The timeframe for delivering this project is twelve weeks. Due to this timeframe our approach to the project will be limited, such as the extent of frameworks and programming languages used.
- The server that the project will use may be restricted by the hosting provider. For example, the hosting company may have monthly bandwidth restrictions.
- Finally, a highly likely constraint is the team having to meet and work remotely over services such as Microsoft Teams, Discord and GitHub if Covid-19 impacts in person communications.

4.6. Major Project Milestones

The following are the major project milestones identified at this time. As the project planning moves forward and the schedule is developed, the milestones and their target completion dates will be modified, adjusted, and finalized as necessary to establish the baseline schedule.

Milestones/Deliverables	Target Date
Project Plan Review and Completion	14/12/2021
Project Kick-off	25/01/2022
Working GUI Complete	10/03/2022
Web app styling Complete	07/04/2022
MOSS Tasks Complete	07/04/2022
Security and Software Testing Complete	14/04/2022
Completed Product	02/05/2022
Closeout/Project Completion	02/05/2022

5. Strategic Alignment

The MOSS GUI Project is in direct support of Abertay's Strategic Plan and the client's goals. By directly supporting these goals, this project will improve the day-to-day life at Abertay and help the lecturers in their everyday tasks.

Goal	Relationship to Project
Comply with Abertay University's strategic plan	To meet the estimation for Abertay in 2025, the MOSS GUI project will support innovative use of digital technology which will be used by staff of Abertay University.
Increase Productivity	The time taken to prepare all the student's code for plagiarism is having an impact on staff productivity. The project aims to meet the goals of the client and improve this situation.

6. Cost Benefit Analysis

First year costs for the MOSS GUI project are detailed below. However, next year the total net year cost will increase, such as web hosting, if the decision is made to keep the web app running.

Action	Action Type	Description	First year costs
Project Manager	Cost	The total amount to hire the Project Manager for 12 weeks (234 hours).	£20 an hour = £4,680
Front-End Web Developer	Cost	The total amount to hire the front-end developer for 12 weeks (200 hours).	£12.24 an hour = £2,448
Programmers/Back-End	Cost	The total amount to hire the three programmers for 12 weeks (200 hours).	£11.54 an hour * 3 = £6,974
Testers	Cost	The total amount to hire the two testers for 12 weeks (200 hours).	£8.50 an hour * 2 = £3,400
Domain Name	Cost	Obtaining the domain name for the web app.	£2-8 (Higher cost is usually for domain privacy, to hide domain information from the public such as client name)
Purchase Web hosting	Savings	Initial investment for the MOSS GUI Project - Digital Ocean.	$100 free hosting (Otherwise $12 a month)
SSL Certificate	Savings	To keep our website secure, an SSL certificate is needed.	LetsEncrypt/Cloudflare offer free SSL certificates.
MOSS tool acquisition	Savings	The MOSS plagiarism checker tool provided by Stanford University is provided free of charge.	Free
Net First Year Cost			**£17,510**

7. Alternatives Analysis

The following alternative options have been considered to address the current business problem. These alternatives were not selected for several reasons which are also explained below.

No Project (Status Quo)	Reasons For Not Selecting Alternative
Abertay's Lecturers continue using MOSS as a command line utility	• Hassle to check the code for plagiarism this way, affecting productivity • Time for doing the task is increased • Lack of automation
Alternative Option	**Reasons For Not Selecting Alternative**
Outsource the implementation of a web-app to an external vendor.	• Significantly higher cost • Expertise already exists in house • Vendor's lack of familiarity with the University and internal requirements
Alternative Option	**Reasons For Not Selecting Alternative**
The project team develop their own plagiarism checking software internally and don't use MOSS at all.	• Lack of qualified resources • Significant cost associated with software design • Timeframe required is too long • Team expertise does not exist to develop this software internally to the best quality it can be • Tool has already been developed and used over many years. Trustworthy.

8. APPROVALS

The signatures of the people below indicate an understanding in the purpose and content of this document by those signing it. By signing this document, you indicate that you approve of the proposed project outlined in this business case and that the next steps may be taken to carry out the project in accordance with the details outlined herein.

Approver Name	Title	Signature	Date
Prior, Suzanne	Client (Abertay University), Lecturer.	SPrior	14/12/2022

Not for public distribution
Document adopted and modified from ProjectManagementDocs.com

PROJECT MANAGEMENT PLAN: MOSS GUI (PROGRAMMING EDUCATION)

TEAM RNG
[redacted], **ISAAC BASQUE-RICE**, [redacted], [redacted], [redacted], **AND** [redacted]
ABERTAY UNIVERSITY
DUNDEE, DD1 1HG

DATE: 14/12/2021

Not for public distribution
Document adopted and modified from ProjectManagementDocs.com

9. Introduction

In all academic environments, plagiarism is and has always been a significant issue that teachers, lecturers, and professors have had to deal with. in recent years with the advent of the internet, plagiarism checkers such as TurnItIn have been integrated into virtual learning environments to significant effect. However, these tools are usually geared towards essays and other such documents intended for reading. There are few tools for performing such plagiarism checks against programming code.

Dr Suzanne Prior, a cybersecurity lecturer at Abertay, has approached the RNG team to develop one such tool, designed to run through the MOSS plagiarism checker, developed by Stanford University for this purpose, run though a zip file provided by the client, and check the content of programming code files for plagiarism. This program will take the form of a GUI Front-End with some Back-End capability and is intended to streamline the problem of plagiarism checking, as the current process is an arduous and repetitive task that is not embedded into VLEs currently.

10. Project Management Approach

The Project Manager (PM) on this project, Isaac Basque-Rice, has the responsibility of managing and executing the project plan to its fullest extent possible in the given timeframe, this will mean being responsible for decision making, task delegation, and general administrative tasks.

If a decision made by the Project Manager is contentious then the decision will be put to a vote of the relevant parties (I.e., the personnel on the project team) and decided this way. The project team consists of individuals in one of three groups. These groups are Programming (comprised of [redacted], [redacted], and [redacted]), Design ([redacted]), and QA/Testing ([redacted] and [redacted]). Technical Documentation will be expected to be provided by individuals as they work.

Documentation regarding the project, its management, funding decisions (as appropriate), and delegation of approval authority to the Project Manager will be approved by the Project Sponsor.

Individuals from each group will report to the Project Manager on a weekly basis throughout the duration of the project, the PM will be responsible for communication between themselves and the representatives from each group with regards to project resources and progress.

The RNG team will be taking an incremental and prototyping based approach to this project for a few key reasons. Primarily, although the complexity and uncertainty are both low, it may not be unreasonable to imagine the timeframe for this project will be short. As a result of this the team has decided to focus efforts on prototyping the application prior to the official beginning of the Development phase, this will be to gain a feel for the technologies the project will be using, improve the delivered application through testing what works and what may not, and establishing a valuable feedback loop whereby the testers on the team regularly come back to the other members of the team suggesting what need to be added, changed, or removed.

11. Project Scope and Milestone List

The scope of the MOSS GUI Project includes the planning, GUI design, development, testing (both security and functionality), deployment, and eventual transition of the project to a university administrator. Crucially, the design and implementation of a new plagiarism checker is out of scope for this project, as this project concerns a graphical implementation of an already existing plagiarism checker. The application will also meet all legal requirements set forth in the Equality Act (specifically with regards to disability discrimination/accessibility), as well as the disability discrimination act, to this end the developers will implement several accessibility features as appropriate (e.g., alt text in images for the benefit of those who use screen readers, and the usage of plain English text).

This project's scope also includes the production of relevant documentation, both for the benefit of future developers (development documentation) and the user (a user manual). The project will be considered completed when full documentation is complete, the development process as laid out in this document is complete, and both are handed over as a package to the client to be signed off, at which point the client will push into production.

All work on the MOSS GUI project will be performed internally to Team RNG, with no outsourcing. Due to the nature of the project as a client-side web application, the project is system/platform agnostic and as such the scope requires no changes in requirements to operating system, platform, software, updates, or revisions outside of what is deemed reasonable.

The table below outlines the major milestones for the MOSS GUI project. Each of these milestones represent overarching sections of the project and have many smaller milestones within them, for example the milestone regarding the Back-End development has within it multiple sections relating to file discovery and manipulation, as well as integration, file searching, fetching results, and so on. Many of these smaller milestones were achieved through a process of functional decomposition.

In the event of a scheduling delay that would affect the delivery of any of these milestones, said delay will be communicated to the project manager who will proactively consult the relevant parties regarding actions that can be taken to mitigate these delays. Changes to milestones, dates, and other targets will be approved by the project manager. In cases where the hand over date is affected by the delay the client will be notified after steps to mitigate this delay have been agreed upon and put in place.

Milestone	Description	Date
Complete Requirements Gathering	The Requirements Gathering process is a continuous process that began upon assignment of the project by the client and continued through the first few weeks of work through dialogue between the team and the client, requirements gathering should end prior to any development.	30/01/22
Complete MOSS GUI User Interface Design	The front-end design for the web application completed, beginning with wireframing, and then altered and improved iteratively over time.	07/04/22
Complete Back-End Development	Back-end development for the site completed, comprising server-side scripting, website functionality, and interaction with MOSS.	15/04/22
Complete Cloud Management and Configuration	Configuration of the cloud hosting service complete, will be done alongside back-end development by a dedicated cloud engineer.	15/04/22
Complete Testing, Bug fixing, and adjusting requirements	All programming and configuration completed, and errors corrected, this is also an ongoing process.	15/04/22
Complete Documentation	All documents produced for the client completed and approved.	04/05/22
Hand Over to Client	Completed project and all documentation and code transferred to the client to begin deployment.	04/05/22

The following products will be handed over to the client:

Product	Description	Date
Developer Documentation	This documentation will outline the process of development as well as all aspects of the program from a developer's perspective to facilitate	04/05/22

	maintenance and further development should the user's needs and requirements change over time after handover	
User manual	A manual outlining the system from a user's perspective, as mentioned previously due to the requirements of the project emphasising efficiency this document should not be particularly large	04/05/22
Source Code	The source code of the project, this includes both server and client-side code, HTML, CSS, JavaScript, Server-Side scripts, and other assets required for normal function of the application	04/05/22

12. Work Breakdown Structure (WBS)

The WBS for the MOSS GUI Project is comprised of a collection of work packages. These packages do not exceed 20 hours of work but are at least 2 hours' worth of work, the reasoning for a smaller timeframe of work constituting a work package is that the target for management of this project is functional decomposition, I.e., being able to make a large task many smaller tasks to improve efficiency during work. Each Work Package was developed by the project manager in conjunction with the rest of their team.

The MOSS GUI project schedule was derived from the Work Breakdown Structure with appropriate feedback and input from all team members. The schedule was completed and approved by the Project Sponsor and in the event of any changes to requirements this process can be repeated. The schedule will be maintained by the project manager in two forms, a Kanban board hosted on Trello, and as a Gantt chart hosted on GanttPRO, both online tools.

As previously mentioned, in the event of alterations to the work plan during development the Project Sponsor will be consulted in the form of a change request, this document will be an assessment of the change on schedule, cost, resources, scope, and risks. In the event of a change in this document exceeding acceptable boundaries in any of these categories the Project manager will consult the Project Sponsor for her approval.

The MOSS GUI boundary conditions that the team has agreed upon are that +/- 0.1 will result in caution from the team and moderate action will be taken to remedy this, +/- 0.2 or more will be considered critical, wherein immediate, and critical action must be taken.

The Work Breakdown Structure is provided in Appendix A.

13. Change Management Plan

This section of the project proposal will describe the change control process of the MOSS GUI project. It details the steps we will take to decide whether a change will be taken or not and identifies who can approve changes and who will be involved within the decision.

1. The need for a change is identified.

2. A member of the group requests to make a change to a section of the project.

3. The group will have a meeting about the proposed change, discussing the reasons it's needed decide whether the change is necessary or not.

4. If it is agreed that a change is needed the team will, decide whether the change will require approval from the Project Sponsor (proposed change that such as substantial changes to UI design would require approval).

5. 5.1 if it is not a large the change will be implemented immediately by the group

   5.2 Any larger changes we will contact the Project Sponsor (Dr Suzanne Prior) for approval before implementing the change.

Changes to the project can be suggested by any member of the team. The discussion will involve all members of the team and in-order for a change to be approved by the group it must have at least most of the group's approval. If the change requires approval from the Sponsor, she will be contacted by the Project Manager (Isaac Basque-Rice) the change will only be implemented after the group has approval from the client.

14. Communications Management Plan

Meetings:

Each week we will have a meeting with all members of the project team lead by Isaac Basque-Rice (Project Manager) where we will discuss the work done in the previous week, discuss any issues and what we intend to-do over the following week. All members of the team should arrive on time to these meeting or if this is not possible inform the rest of the group before the start of the scheduled meeting. During the meeting, all members should have their phone switched off to ensure people are not distracted and are involved in the meeting. Notes will

also be taken throughout the meeting by a member of team. These notes will be distributed to the rest of the team after the meeting.

Emails:

Any emails sent by a member of the team related to the MOSS GUI project should contain proper grammar, correct spelling and the language used within the email should be professional i.e., the email should not contain profanity or unrelated topics. Any attachments on emails should be in a format agreed upon by all members of the team. Emails should also contain proper subjects and only be sent to those who it is necessary to send it to. Any email that raises an issue that needs to be addresses should clearly state what the issue is and either suggest a time for a meeting to produce a solution or suggest a solution to be considered.

Informal communication:

During the MOSS GUI project, the group will use discord to communicate more informally with the rest of the team. To ensure proper documentation of anything important discussed this way it should also be briefly mentioned what the discussion in discord was about during the weekly meeting therefore all discord discussions will be recorded alongside the weekly meeting notes. While communicating informally team members should still avoid using profanity and or insulting other team members.

Communication Type	Description	Frequency	Format	Participants / Distribution	Deliverable/ Product	Owner
Weekly Status Report	Email summary of project status	Weekly	Email	Project Client, and team	Status Report	Project Manager
Weekly Project Team Meeting	Meeting to review work over the week and discuss progress also discuss what will be done next	Weekly	In Person	Project Team	Updated Action Register	Project Manager
Monthly Project review	Review status of project	Monthly	Email	Project client, and team	Project status presentation	Project manager

Name	Title	E mail
Suzanne Prior	Project Sponsor	s.prior@uad.ac.uk
Isaac Basque-Rice	Project Manager	1901124@uad.ac.uk
[redacted]	Testing	1900189@uad.ac.uk
[redacted]	Design	1900112@uad.ac.uk
[redacted]	Testing	1203685@uad.ac.uk
[redacted]	Programming	1901193@uad.ac.uk
[redacted]	Programming	1900609@uad.ac.uk
[redacted]	Programming	2102430@uad.ac.uk

15. Cost Management Plan

Overall, the project will cost a minimal amount as everything being used will either be free or cost an exceedingly small amount

To ensure the project remains on budget throughout the project any spending will first have to be approved by the Project Manager and the Project Sponsor/ client this will then be stored within a document that will be monitored throughout the project. Overall control over budget and spending will be decided by the Project Sponsor, although it will be monitored by the [project manager or give somebody role of accountant], this person will also be responsible for working out the MOSS GUI projects financial performance with will be reported regularly monthly to the Project Sponsor during the monthly project review

The CPI (Cost Performance Index) and SPI (Schedule performance Index) will be used to calculate the project's finical performance. Smaller changes in performance may not present an issue although larger changes will and will therefore require quick corrective action to keep the project on budget. All changes must be reported during the monthly update to keep the whole team informed about the cost and whether we are on schedule or not, even if it is decided that being slightly behind schedule will have no effect on the overall state of the project. If corrective action is required the change must be approved using the method stated above in the change management plan. Severe problems caused will be reported to the Project Sponsor immediately to ensure corrective action can be taken quickly to ensure the project remains within the budget and on schedule.

The CPI (cost performance index) is a calculation designed to show how closely a project follows its budget. If the result of the calculation is 1.0 then the project is on budget, above this and the

project is underbudget. Anything below 1.0 means that the project is overbudget. The SPI (Schedule performance index) is a similar calculation used to show how closely the project follows its schedule. Similarly, to the CPI, if the result of the SPI calculation is above 1.0, it means the MOSS GUI project is ahead of schedule, under 1.0 means it is behind schedule and 1.0 means that the project is on schedule. Results from either of these calculations of 0.8 or lower will require immediate corrective action 1.2 or above will also require action.

16. Procurement Management Plan

During the MOSS GUI project, it is expected that there will be a minimal amount of procurement involved. The Project Manager will be able to approve any procurement that will not cause the project to exceed the budget, anything exceeding the budget must be approved by the Project Sponsor.

If it is decided that procurement of something is required then the group will have a meeting to discuss all the advantages and disadvantages of the options available, several factors will be considered this includes cost, capabilities, limitations, accessibility, and efficiency. Once a product that suits the project's needs is selected with the approval of the Project Sponsor the Project Manager will then contact the vendor of the product to purchase the agreed product. This will be recorded and reported as part of the monthly update on the status of the project.

17. Project Scope Management Plan

The scope of this project will be measured and verified by the work breakdown structure, Project Scope and Milestone List, scope statements and the Definition of done. These documents will identify projects goals and objectives, this will help with preventing scope creep which could be dangerous in this project as we will be working within a very limited timeframe.

The Project Manager will be responsible for scope management, he will keep track of work breakdown structure, Project scope and Milestone List, scope statements, definition of done and update them when necessary.

The Project Sponsor and all project team members can initiate a scope change. All change requests will be democratically discussed as a group, in a case of a stalemate Project Manager will have a decisive vote. Upon acceptance of the scope change request, the Project Manager will submit a proposed scope change request to the Project Sponsor. Upon approval of the scope changes by the Project Sponsor the Project Manager will announce the changes to all project team members, Project Sponsor and update the project documentation.
The scope changes initiated by the Project Sponsor will follow a similar process, the changes will be discussed but there will be no voting, changes initiated by the Project Sponsor will automatically be accepted.

The Project Sponsor will be responsible for formally accepting the project's final deliverable. This acceptance should be based on the -

- Definition of done.
- Work breakdown structure.
- Testing results.
- Completion of all the necessary work packets.
- Deliverables presented.

The Project Manager will validate and compare completed work packets to the work breakdown structure, definition of done and milestone list at each increment to prevent scope creep and to make sure that all the requirements are met. If the requirements are not met for any reason. The Project Manager should try to find the cause of the problem and implement appropriate solutions.

The definition of done (DOD) is a list of criteria that needs to be done for a project or a project feature to be considered complete. The Project Manager should work with the Project Sponsor to create DOD by creating a DOD checklist for release and/or a DOD checklist with user stories. Although a Gantt chart with deadlines was created (look Appendix B) a DOD is an innovative idea as it will specify when a task is truly done.

18. Schedule Management Plan

Project schedules for MOSS GUI will be created using Ganttpro.com, MS Excel and Drawm.io and then it will be maintained using Trello. Trello was chosen as it is easy to use, provides the visualization tools and is free.
Project schedules will include -

- Gantt Chart
- Precedence Network
- Work Breakdown Structure (WBS)
- Resource Calendar

The WBS will identify work packages that must be completed to deliver all required deliverables and to successfully complete the project.
Precedence network and Gantt chart will be used to define the order of work packages, find critical paths, and assign relationships between project activities. The duration of each activity will be estimated using the planning poker method as this gives everyone a chance to share their ideas, propose better solutions and better understand work packages.
A resource calendar will be used to assign resources to each work package to complete work packages to an acceptable standard and on time.

Once the preliminary schedule has been completed, it will be reviewed by the project team. The project team must agree to the proposed work package assignments, durations, and schedule. Once this is achieved the Project Sponsor will review and approve the schedule and it will then be baselined.

The following will be designed as milestones for project schedule (Look at 11. Project Scope and Milestone List for description of these points) -

- Complete Requirements Gathering
- Complete MOSS GUI User Interface Design
- Complete Back-End Development
- Complete Cloud Management and Configuration
- Complete Testing, Bug fixing, and adjusting requirements
- Complete Documentation
- Hand Over to Client

These milestones were selected because they: highlight important dates, help with project evaluation and the definition of done.
Roles and responsibilities for schedule development are as follows:

The Project Manager will be responsible for facilitating and validating the project schedules listed above. The Project Manager will also present these schedules to the Project Sponsor to obtain final approval.

The project team will assist in estimating work package duration by playing planning poker. The project team will also review and validate the proposed schedule and perform assigned activities once the schedule is approved by the Project Sponsor.

The Project Sponsor will participate in reviews of the proposed schedule documents (Gantt chart, Work breakdown structure and Precedence Network) and approve the final schedule before it is baselined.

The Gantt chart is usually associated with waterfall methodology, but a Gantt chart will be utilized in this project as we are an inexperienced team that will benefit from additional guidance, fixed deadlines and details present in the Gantt chart. Also, an expert insisted on using the Gantt chart despite my suggestion of using a product road map instead.
A Gantt chart and precedence network with a critical path is provided in Appendix B.

19. Quality Management Plan

All members of the RNG team will take part in quality management, each team member should make sure that their individual work packages and other contributions are delivered on time and to an acceptable level.

The Project Manager will be responsible for -

- Documenting the progress logs.
- Communicating updates to the team members and the Project Sponsor.
- Working with a client (Project Sponsor) in project scope and requirements definition.
- Presenting the project prototypes at each increment to the client and then logging and communicating the client's feedback and expectations to all team members. Client tests should be conducted using a screen capture tool like TestComplete for analysis and to make sure that the GUI is intuitive and can be operated quickly as saving clients time is the main feature of this project.
- Validating whether DOD requirements/user stories are met.
- Working together with testers to ensure the quality of the Project by following the Quality Management Plan, ensuring that all tasks are done on time and according to the plan.

The Testers will be testing products functionality and design to make sure that our project meets the client's requirements and expectations. Also, Testers will be assisting the Project Manager with documenting progress logs and in quality management.

The Project Sponsor will be responsible for working with a Project Manager at reviewing product prototypes and quality standards additionally Project Sponsor will sign off the final deliverables. The Project Sponsor will also be responsible for approving the definition of done.

The quality management plan will be an iterative process consisting of four steps described below. This quality assurance method was selected as it fits the incremental development methodology and allows for client’s engagement and feedback in the development which is necessary for intuitive and easy-to-use GUI.

The ISO 9126/25010 quality characteristics below will be used to calculate the weighted scores of the future prototypes. Importance rating values were selected by evaluating the client's requirements and goals, the client's main goal is to reduce the workload and amount of time spent finding plagiarism, therefore usability, reliability and functionality are important for this project.

Characteristic	Importance rating
Functionality	4
Reliability	4
Usability	5
Efficiency	3
Maintainability	1
Portability	1

The Project Manager will follow this process to ensure that all quality standards are met.

- Creating a quality baseline.

This step is about creating a quality baseline by reviewing our previews work, prototypes, and weighted scores this will allow us to get a good understanding of the project's current position.

- Finding client's requirements and expectations.

This step is about finding out what client wants and what are hers though about the work we have done and our plans moving forward. During the prototype presentation, the Project Manager should gather the client's feedback and expectations.

- Performing a gap analysis

Now with all the information gathered, the quality baseline can be compared with the client's requirements and expectations to identify the areas which need improvements.

- Improve

After the problem has been identified the Project manager need to inform the rest of the project team about the problem to think of a solution to that problem, update the scope and requirements and/or schedule plan and finally assign resources and implement that change. Then at the next increment, this process will repeat.

We will adopt the black box testing methodology as it focuses on making sure that the project fulfils its basic functionality and meets the user requirements.

At this moment, the project does not need to meet any legal/GDPR standards. But it is likely to change during the development process. The login system and cloud optimization work packages are likely to need a GDPR consideration for example a login system might need a cookie consent pop-up and/or terms of service agreement. Whether these solutions will be needed or not will become known during the development process. The designer will design and implement appropriate solutions in case these legal considerations are needed.

20. Risk Management Plan

To ensure risks are managed correctly during the project, the team will come together to highlight what potential risks could arise. Once risks have been identified they will be ranked based on how likely they are to happen and impact, so they can be avoided or managed appropriately. This will be a recursive process to ensure any new risks are identified as the project progresses and to monitor currently identified risks. The team will be proactive to ensure risks are managed or identified prior to them interfering with the progress of the project. All members of the team will be responsible for helping manage potential risks.

As mentioned above the project team will be proactive and the risk management process recursive. The way the team will achieve these goals, is by having weekly meetings in which they will discuss the current risks and any new risks that arise during the project. These will also be tracked by the project manager who can assign specific team member to deal with these when needed. Using communication tools such as discord the team will also be in constant contact, this further enabling them to tackle and manage risks through the whole process.

Once the project is complete the information gained during the risk management process will allow the project team to garner experience on what has went well and what could be improved. This will help build personal knowledge as well as group knowledge on how to deal with risks during a software project.

The highest priority risks that have been identified have been added to Appendix C alongside the other identified risks and their solutions.

21. Staffing, Resource and Cost

The project team will operate under a projected organizational structure with the project manager being responsible for any final decisions made during the project. The project team consists of seven members with the following roles:

Project Manager (1 position)

The project manager will be responsible for all management related tasks during the project. This will consist of but not limited to the following, assigning tasks to team members, keeping track of progress, relaying communications to team members, reports, and communications with clients/ representatives. The project manager will also deal with final decisions when the team disagrees on what direction to take during the project.

Programmer (3 positions)

The programmers will be responsible for code-based tasks during the project and will help the project manager split the work into manageable sections from which the programming team can work on. They will also be responsible for ensuring the work is compliant with the standards and expectation set out by the client alongside ensuring risks are identified and managed as per the risk management policy.

Designer (1 positions)

The designer is responsible for all design basked tasks. They will help the project manager split this workload into manageable sections that can be tracked for progress. They are also responsible for making sure the Front-End of the project meets the client's expectations whilst still being functional and fitting into the scope of what the programmers can do. They will also be responsible for ensuring risks are identified and recorded as required.

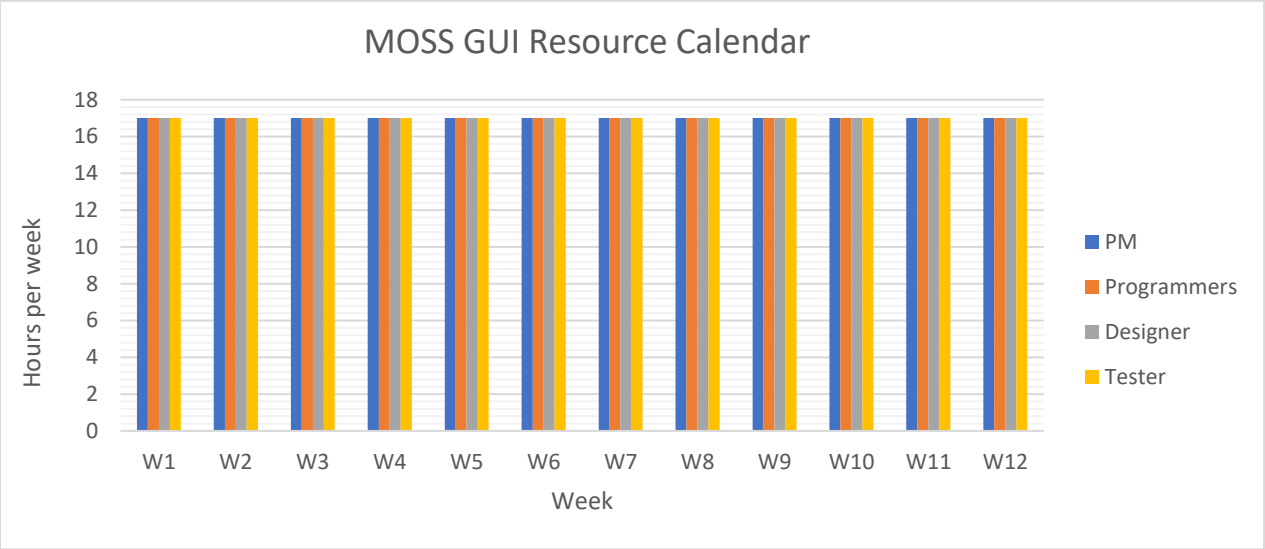
Tester (2 positions)

The testers are responsible for all testing/ bug finding activities. They will work with the project manager to split this workload into manageable and trackable tasks. They are responsible for ensuring the work done is fully functional and meets the client's expectations whilst being free from errors. They will also ensure that no risks are posed from the work that has been completed that could affect the success of the project.

Resources

The programing education project is scheduled to last 12 weeks, with all team members being required throughout. During the various phases of implementation, the hours required from team members may vary. The project will operate under 17-hour work weeks. The 17-hour work week means that team members will have an equal amount of time, to spend on the other two modules they are required to complete during this period. This adding up to just more than what would be expected in full time employment 40 hours a week.

The resource calendar bellow displays that each member of the team will be working for roughly the same amount of time each week during the course of the project. This may vary during implementation due to scheduled tasks taking longer or being completed faster than expected. However, with this being a group task the hours have been set so each member contributes the same amount of effort towards the project. In reality it is unlikely each member will be required for the full time each week but as with projects done by professional companies elsewhere, people have a scheduled work week regardless of timescales on tasks. The calendar does consider the total working hours of 1428 displayed on the Gantt chart in Appendix B.



There are various resources that the project team will make use of during implementation of the project. Due to the nature of the project these can be broken down into two categories, Free and paid.

Free Resources

During the project various python modules will be used, these come at no cost as they are public libraries.

Access to the MOSS server where the processing of the submitted files will take place is an essential resource and is provided by the client.

Use of various computer system on which the application will be coded and tested from. This is a resource that the project team will supply themselves or is already provided by the university.

Source code editing software, the project team will be using software such as Microsoft Visual Studio and Visual Studio Code to create, edit and maintain the project source code. Due to being enrolled at the University of Abertay the project team has access to this software for free.

Offline Hosting software, during the coding/implementation of the application the team will make use of offline hosting software to cut down costs of the project. This means the server for hosting the application only needs to be used towards the end of the project.

GitHub organization, this will be used to allow all team members to work on the project when together and remotely, this ensuring efficiency is high and acts as a backup for the project files.

Paid Resources

The project team will make use of PyCharm IDE, this is a paid feature and will cost around £150 for an active license. However, the project team will have access to this for free during the project due to being students.

Hosting server, the application requires a server from which it can be hosted and run, this will require a regular payment to be maintained for continued operation. This again will come at no charge to the project team, Digital Ocean offers students free credit for hosting services and this will cover the costs incurred by the project.

Cost

The cost baseline bellow will cover all costs to complete the MOSS GUI Project. These costs are calculated by assuming each member is paid the expected salary a graduate would earn after completion of their studies, for their specific role. The hours used bellow are based on the Gantt chart provided in appendix B.

The hourly wages have been calculated by assuming 8-hour days and the following,

Project Manager: £42,635 yearly, £20.49 hourly (PayScale 2021, Project Manager).

Programmers: £24,804 yearly, £11.92 hourly (Glassdoor, Junior Programmer).

Testers: £24,874 yearly, £11.95 hourly (PayScale 2021, Software Tester).

Web Developer/Designer: £26,007 yearly, £12.50 hourly (PayScale 2021, Web Developer).

Project Phase	Budgeted Total	Comments
Planning	£966.62	119 Working hours, 17 hours per each member of the team.
Project Management	£4794.66	Total calculated by 234 working hours * 20.49(Hourly rate)
Design	£2500	Total calculated by 200 working hours * 12.5(Hourly rate)

Coding	£7271.2	Total calculated by 276 working hours * 11.92 (Hourly rate)
Testing	£4636.6	Total calculated by 388 working hours * 11.95 (hourly rate)
Documentation	NA	Included in working hours of project members.

22. Quality Baseline

The Quality baseline will outline the conditions that the MOSS GUI Project must meet to be classed as successful. These are the conditions that have been agreed upon by the client. The project team will also ensure these are factors that shape decision making during the course of the project. Further details can be obtained from quality matrix in Appendix D.

Item	Acceptable Level	Comments
Successful Data Return	100% data return from MOSS servers is expected.	
Device compatibility	The application is expected to run on windows, Linux and OSX operating systems.	
Submitting files	Submitting files should take less than 30 seconds, this is to be a drag and drop feature.	
Supporting Documentation	User manual and technical manual. These documents should be sufficient in detail that a user can operate or edit the functionality of the application without referring to the project team.	

SPONSOR ACCEPTANCE

Approved by the Project Sponsor**:**

SPrior

Date: 13/12/2021

Dr Suzanne Prior
Lecturer, Abertay University

APPENDIX A: WORK BREAKDOWN STRUCTURE

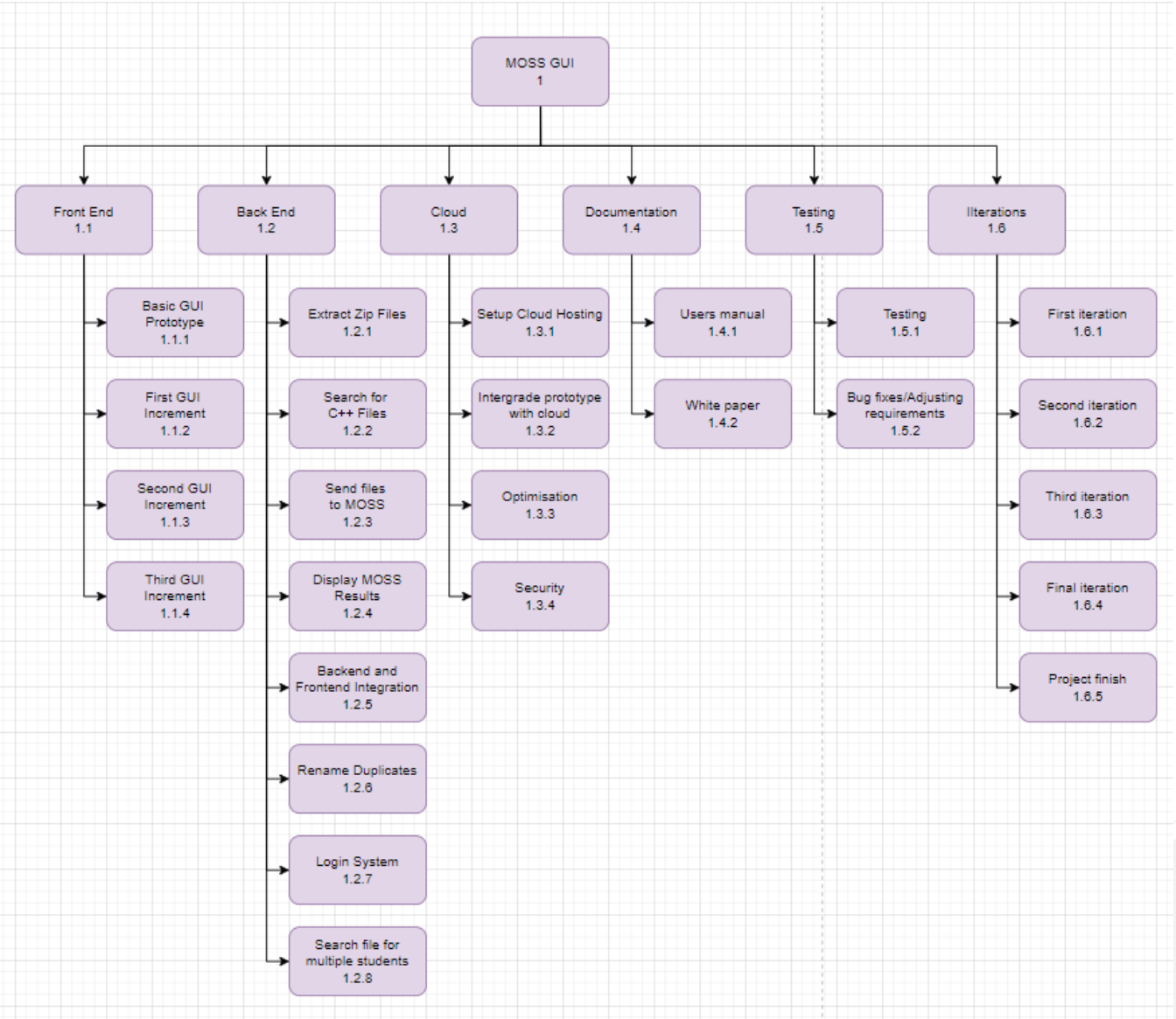
Introduction

The Work Breakdown Structure presented here represents all the work required to complete this project.

Outline View

1. MOSS GUI
    - 1.1 Front-End
        - 1.1.1 Basic GUI Prototype
        - 1.1.2 First GUI Increment
        - 1.1.3 Second GUI Increment
        - 1.1.4 Third GUI Increment
    - 1.2 Back-End
        - 1.2.1 Extract Zip Files
        - 1.2.2 Search for C++ files
        - 1.2.3 Send files to MOSS
        - 1.2.4 Display MOSS Results
        - 1.2.5 Back-End and Front-End Integration
        - 1.2.6 Rename Duplicates
        - 1.2.7 Login System
        - 1.2.8 Search File for Multiple Students
    - 1.3 Cloud
        - 1.3.1 Setup Cloud Hosting
        - 1.3.2 Integrate Prototype with Cloud
        - 1.3.3 Optimisation
        - 1.3.4 Security
    - 1.4 Documentation
        - 1.4.1 User Manual
        - 1.4.2 Whitepaper
    - 1.5 Testing
        - 1.5.1 Testing
        - 1.5.2 Bugfixes/Adjusting Requirements
    - 1.6 Iterations
        - 1.6.1 First iteration
        - 1.6.2 Second iteration
        - 1.6.3 Third iteration
        - 1.6.4 Fourth iteration
        - 1.6.5 Project Finish

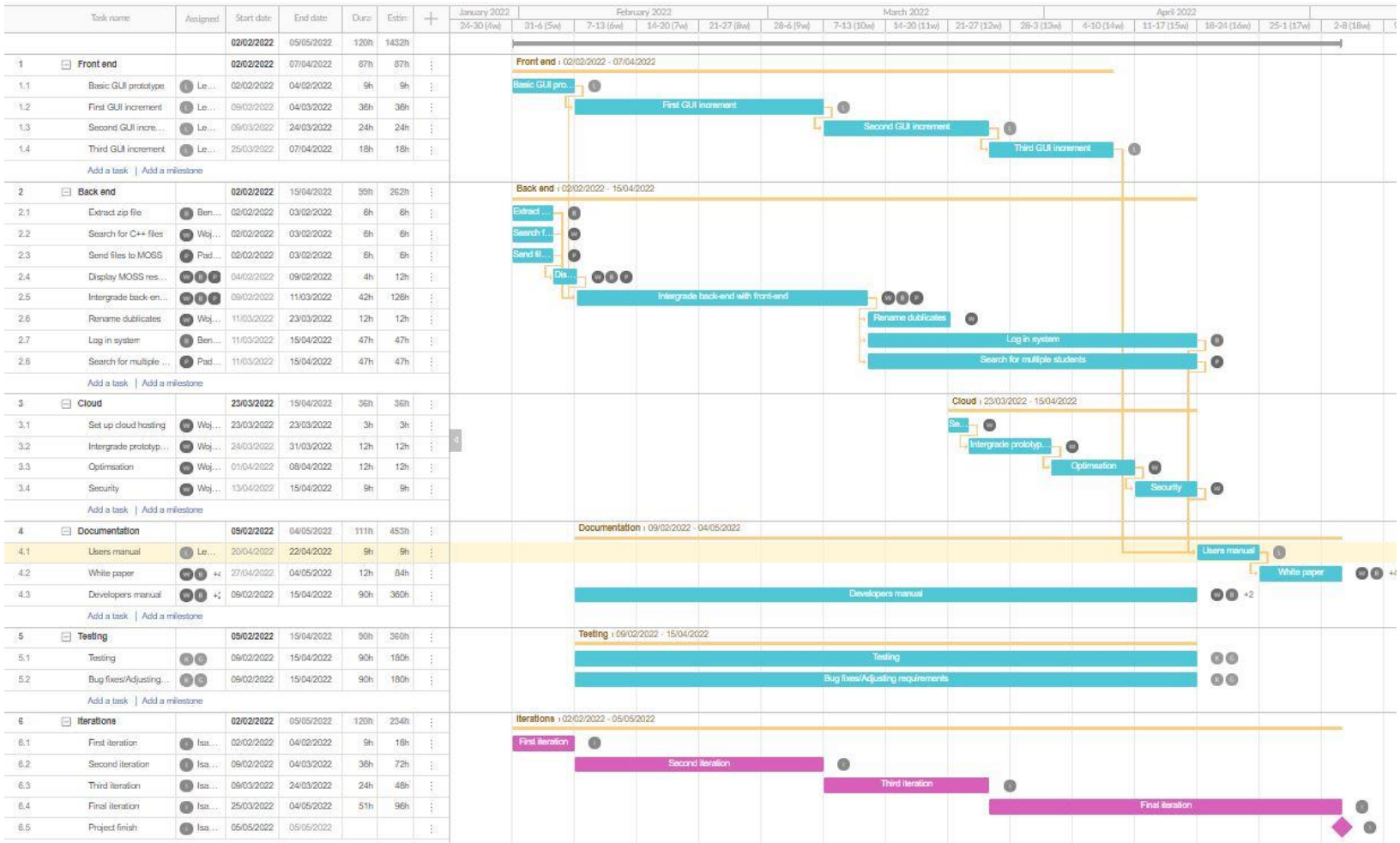
Tree Structure View



Glossary of Terms

Term	Definition
Back-End	The part of the application not directly accessed by the user, wherein (in this project) the code is processed and sent to the MOSS server, and the results are parsed and sent to display
Cloud	AN external server on which code will be hosted, data stored, and processed to serve it to the user.
CPI	Cost Performance Index, the financial efficiency of a project.
Front-End	See: GUI
GUI	Graphical User Interface, a method of interacting with an application using a series of graphical icons, this is as opposed to CLI, Command Line Interface, how MOSS is used currently, where text is input as commands to an application.
Level of Effort (LOE)	Level of Effort (LOE) is how much work is required to complete a task.
MOSS	Measure of Software Similarity, a code plagiarism checker developed by Stanford University, used by lecturers worldwide to check the validity of their students' code submissions.
SPI	Schedule Performance Index, the progress of the project against the performance plan.
Strategic Plan Document	A document created by the team to communicate the goals of the project
TeamRNG	The name of the Project Team
WBS Code	A unique identifier assigned to each element in a Work Breakdown Structure for the purpose of designating the elements' hierarchical location within the WBS.
WBS Component	A component of a WBS which is located at any level. It can be a Work Package or a WBS Element as there is no restriction on what a WBS Component is.
WBS Element	A WBS Element is a single WBS component, and its associated attributes located anywhere within a WBS. A WBS Element can contain work, or it can contain other WBS Elements or Work Packages.
Work Package	A Work Package is a deliverable or work component at the lowest level of its WBS branch.

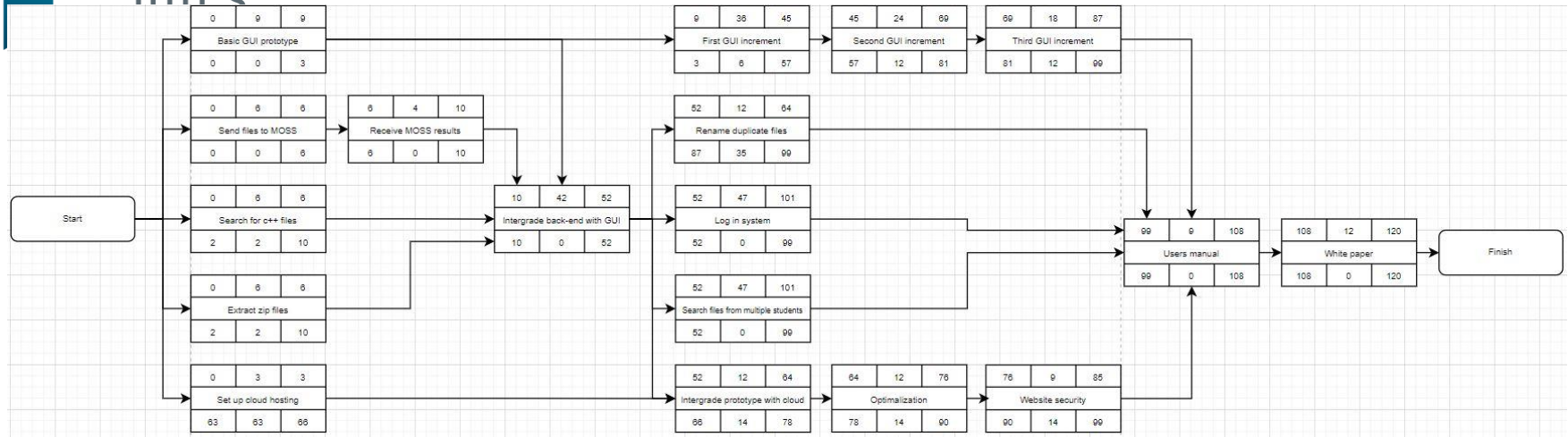
APPENDIX B: GANTT CHART AND PRECEDENCE NETWORK



	Task name	Assigned	Start date	End date	Durat	Estim
			02/02/2022	05/05/2022	120h	1432h
1	Front end		02/02/2022	07/04/2022	87h	87h
1.1	Basic GUI prototype	Le…	02/02/2022	04/02/2022	9h	9h
1.2	First GUI increment	Le…	09/02/2022	04/03/2022	36h	36h
1.3	Second GUI incre…	Le…	09/03/2022	24/03/2022	24h	24h
1.4	Third GUI increment	Le…	25/03/2022	07/04/2022	18h	18h
	Add a task | Add a milestone					
2	Back end		02/02/2022	15/04/2022	99h	262h
2.1	Extract zip file	Ben…	02/02/2022	03/02/2022	6h	6h
2.2	Search for C++ files	Woj…	02/02/2022	03/02/2022	6h	6h
2.3	Send files to MOSS	Pad…	02/02/2022	03/02/2022	6h	6h
2.4	Display MOSS res…		04/02/2022	09/02/2022	4h	12h
2.5	Intergrade back-en…		09/02/2022	11/03/2022	42h	126h
2.6	Rename dublicates	Woj…	11/03/2022	23/03/2022	12h	12h
2.7	Log in system	Ben…	11/03/2022	15/04/2022	47h	47h
2.8	Search for multiple …	Pad…	11/03/2022	15/04/2022	47h	47h
	Add a task | Add a milestone					
3	Cloud		23/03/2022	15/04/2022	36h	36h
3.1	Set up cloud hosting	Woj…	23/03/2022	23/03/2022	3h	3h
3.2	Intergrade prototyp…	Woj…	24/03/2022	31/03/2022	12h	12h
3.3	Optimsation	Woj…	01/04/2022	08/04/2022	12h	12h
3.4	Security	Woj…	13/04/2022	15/04/2022	9h	9h
	Add a task | Add a milestone					
4	Documentation		09/02/2022	04/05/2022	111h	453h
4.1	Users manual	Le…	20/04/2022	22/04/2022	9h	9h
4.2	White paper		27/04/2022	04/05/2022	12h	84h
4.3	Developers manual		09/02/2022	15/04/2022	90h	360h
	Add a task | Add a milestone					
5	Testing		09/02/2022	15/04/2022	90h	360h
5.1	Testing		09/02/2022	15/04/2022	90h	180h
5.2	Bug fixes/Adjusting…		09/02/2022	15/04/2022	90h	180h
	Add a task | Add a milestone					
6	Iterations		02/02/2022	05/05/2022	120h	234h
6.1	First iteration	Isa…	02/02/2022	04/02/2022	9h	18h
6.2	Second iteration	Isa…	09/02/2022	04/03/2022	36h	72h
6.3	Third iteration	Isa…	09/03/2022	24/03/2022	24h	48h
6.4	Final iteration	Isa…	25/03/2022	04/05/2022	51h	96h
6.5	Project finish	Isa…	05/05/2022	05/05/2022		

PROJECT MANAGEMENT DOCS
ProjectManagementDocs.com

PRECEDENCE NETWORK WITH IDENTIFICATION OF CRITICAL PATH



APPENDIX C: RISK ASSESSMENT

Top Three Risks:

R1 Team member leaves the University
As the project team consists of seven students from Abertay university it is possible that one or more students may leave the university. This would result in the workload that they were working on not being completed. The project manager will divide this workload between remaining members to solve this issue.

R2 Pandemic restrictions reinstated
Due to the project taking place after a recent pandemic lockdown there is always a chance that restrictions may be reinstated. This would make communication between team members more difficult due to limiting the ability to hold meetings to discuss projects aspects. To counter act this situation the project team has set up communication and file sharing utilities such as Discord for voice and video chat, alongside a GitHub organization to share the project related files.

R3 End results is not what the client expected
As with any project that is being carried out, there is always a risk of the product not being as the client expected and thus not successful. To avoid this situation, the project manager is regularly contacting the client regarding various aspects of the project to ensure they are happy with the direction the team is taking, alongside this the project team are ensuring to follow the provided brief closely.

Other risks:

R4 Feature Creeping
An issue that often arises during the process of a software-based project is too many unneeded features being added into the mix and this can have various negative effects such as making useability difficult. To avoid this situation, the team will follow the information from the project brief and correspondences from the client on what is expected from the application. The project manager will also ensure the team is focusing on the required functionality that the application should possess.

R5 Server-side resources unavailable
The application will be hosted online, and this comes with the possibility of these servers not being accessible, rendering the application unusable. To avoid this situation, the project team

will have back up hosts available as well as providing the client with the necessary files to host the application themselves.

R6 Issues accessing the MOSS servers

For the application to work successfully it needs to be able to submit and receive information from the Stanford University MOSS servers where the bulk of the processing is performed. If the application does not submit data as expected or authorise in the correct manner the MOSS servers may not respond to the request as expected, this resulting in the application not working. To avoid this situation the project team will do extensive research into how data should be passed to and from the MOSS servers to ensure expected results.

R7 Application security

One of the requested features from the client is that it is securely designed. This concerns both functionality and access. To ensure access is secure the application will require users to login. Ensuring the functionality of the application cannot be tampered with best practice coding will be used throughout and the project team consist of two testers who will ensure the application is free from errors and security flaws prior to release to the client.

R8 Readable application output

As the MOSS servers are currently designed for users to submit a single request for data and the application is to allow the user to submit multiply requests, the project team will need to ensure the response is relayed to user in a readable format. To ensure this the project team will design a custom layout that will allow the user to easily compare submissions.

Risk Probability-Impact Matrix

Probability of Risk				
	High		R1	
	Moderate		R2	
	Low	R4	R6 R7 R8	R3 R5
		Low	Moderate	High

Impact of Risk

APPENDIX D: QUALITY METRICS

Based on the Project Sponsor's requirements and the project requirements, the following metrics have been established for the MOSS GUI project. These metrics have been approved by the MOSS GUI team.:

a. **Successful data return**: To be considered successful, the MOSS GUI project must return data from the MOSS servers 100% of the time because to ensure the client's needs are met the product must be reliable, returning data without fail. To ensure that data is 100% of the time before the product can be considered complete the product will be tested with test files several times to ensure that the project returns accurate data every time.

b. **Device compatibility**: This product will be designed to work on the three most common operating systems Windows operating systems, Linux operating systems and OSX (Mac OS). By ensuring that the application is compatible with these operating systems our product will be easily accessible to everyone who needs access regardless of the operating system their device is using. To ensure that the product is compatible with all three of these operating systems tests will be run on the product using devices running on Linux, windows and OSX operating systems.

c. **Submitting files**: For the MOSS GUI project to be considered successful it must allow the user to submit files to the web application in 30 seconds or less. This will ensure that the application works efficiently for the user. The MOSS GUI will use a drag and drop system to allow users to upload files, by designing the application to use a drag and drop system it will ensure that the user can submit files easily and efficiently.

d. **Supporting documentation**: To ensure that the product is easy to use for any user supporting documentation including a user manual that will explain in detail how to use the application, there will also be a technical document that will detail everything about the project on the technical side such as how everything works. With these two documents the user should be easily able to understand both how the application works and how to use the application without issues. This will improve the users experience while working with this application and increase the accessibility of product allow user with less technical knowledge to use it.

Metric	Standard	Frequency	Report

Data return	100%	Every time the application is used	Monthly project status update
Device compatibility	Must work with Linux, windows and OSX	When product is updated	Monthly project status update
Submitting files	Must be submitted within 30 seconds	Every time files are uploaded to the MOSS GUI project	Monthly project status update
Supporting documentation	Must be detailed enough to make use of application simple	When the product is updated	Monthly project status update

REFERENCES

Business Case:

University, Abertay. (2020). Strategic Plan. Available at: https://www.abertay.ac.uk/about/the-university/strategic-plan/ [Accessed 22 Nov 2021]

Payscale. (2021) Average hourly rate for a Project Manager. Available at: https://www.payscale.com/research/UK/Job=Project_Manager%2C_Information_Technology_(IT)/Salary [Accessed 27 Nov 2021]

Payscale. (2021) Average hourly rate for a Web Developer. Available at: https://www.payscale.com/research/UK/Job=Web_Developer/Salary [Accessed 27 Nov 2021]

Glassdoor. (2021) Average hourly rate for a Junior Programmer. Available at: https://www.glassdoor.co.uk/Salaries/junior-programmer-salary SRCH_KO0,17.htm [Accessed 27 Nov 2021]

Payscale. (2021) Average hourly rate for a Software Tester. Available at: https://www.payscale.com/research/UK/Job=Software_Tester/Salary [Accessed 27 Nov 2021]

Project Plan:

Editorial team. (2021, February 1). Quality management plan: What is it and how to create it? Bit.Ai. https://blog.bit.ai/quality-management-plan/