Software Design and Implementation

TrekStar Software System

Callum Axon (N0727303), Callum Carney (N0741707),

Matthew Robinson (N0724629)

2019-04-28

Table of Contents

Todo - *remove me before submission pls*

User Guide

- Using List Materials requires screenshot that shows projectid listings when id issue is resolved.
- Using Maintenance Mode requires screenshot of the maintenance main menu when remove project is added.
- User guide details required for when the application is able to manage crew and materials/projects fully.
- Update Using Add Project to reflect changes to saving a project.

Contributions

Callum Axon (N0727303)

- Stuff

Callum Carney (N0741707)

- Other Stuff

Matthew Robinson (N0724629)

- More Stuff

System Description

The Trekstar system has been developed in order to allow TrekStar Pictures to accomplish the following

- Create and Manage Projects including any relevant metadata
- Create and Manage Project Materials (single-sided DVDs, Blu-rays, etc)
- Create Projects that are "unreleased" and cannot be modified once created
- Include details in regards to the crew working on a Project

Class Diagram

Include class diagram here.

Cohesion and Coupling Considerations

A justification and explanation of how cohesion and coupling have been considered in the design.

Sequence Diagram

Include sequence diagram here

State Diagram

Include state diagram here

Component Diagram

Include component diagram here

Deployment Diagram

There are two possible deployment scenarios for the TrekStar management system, these being:

1. An indivdual user will have the TrekStar project management system installed on their machine, using a locally stored JSON Database.
2. TrekStar Pictures will release the TrekStar project management system company-wide, using a shared JSON Database for each installation.

These two deployments are different due to the change in the data storage location, in which the latter will support company-wide collaboration through the application.

However, no matter where the application is installed the following Operating Systems are supported:

- Windows
- Mac OS
- Linux

On the following pages you can find a visual representation of the aformentioned Deployment scenarios.

Individual Installation
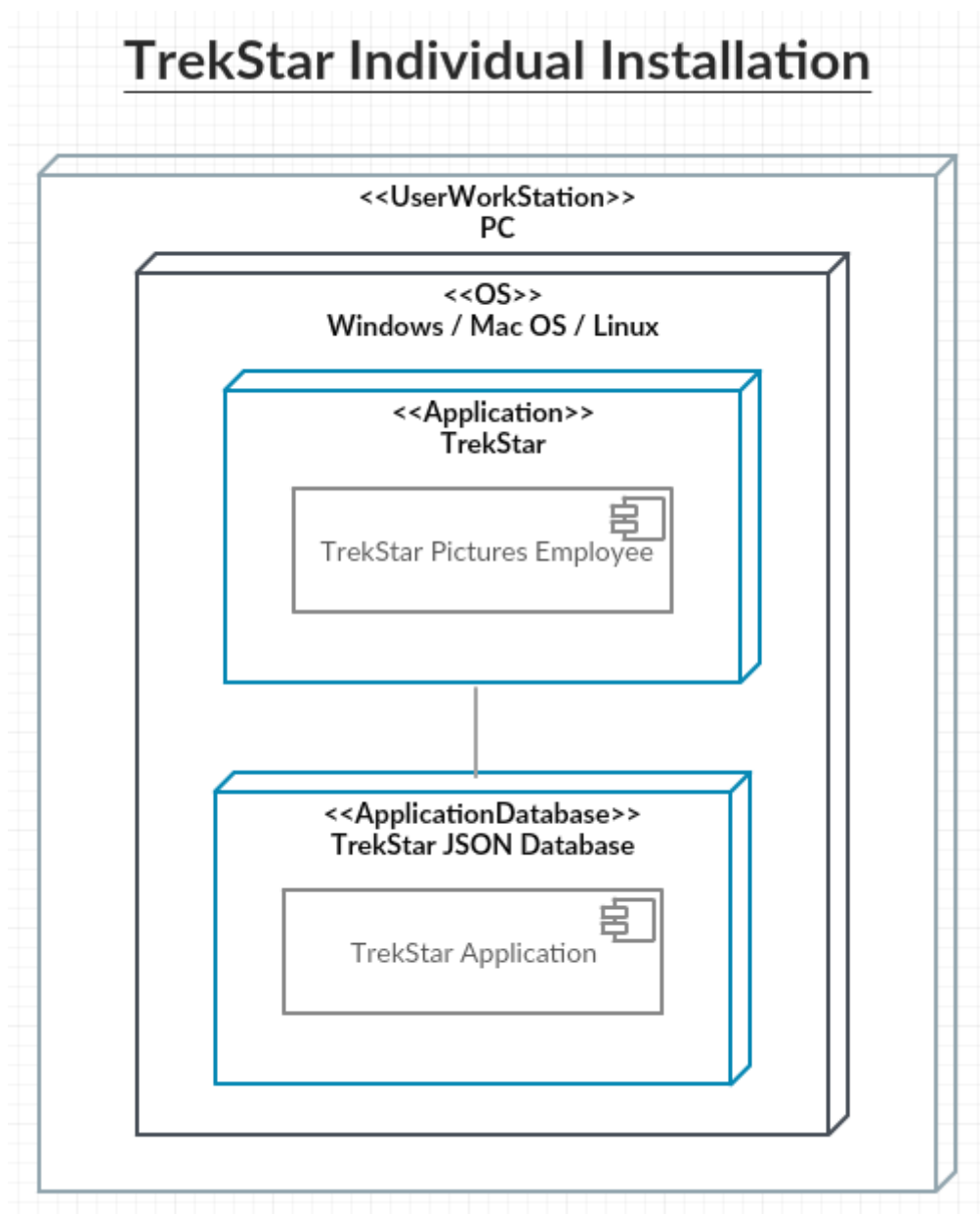
Figure 1: Visual Representation of Individual TrekStar installation

Company-Wide Installation

Figure 2: Visual Represenation of Company-Wide TrekStar installation

Design Pattern

Include explaination of any design patterns used

Planned Architecture

An explanation of the planned architecture and the reason of the choices according to ATAM (follow step 4 and 5, i.e., identify possible architecture styles and choose one with respect to the identified utility tree, you need to explain the reason).

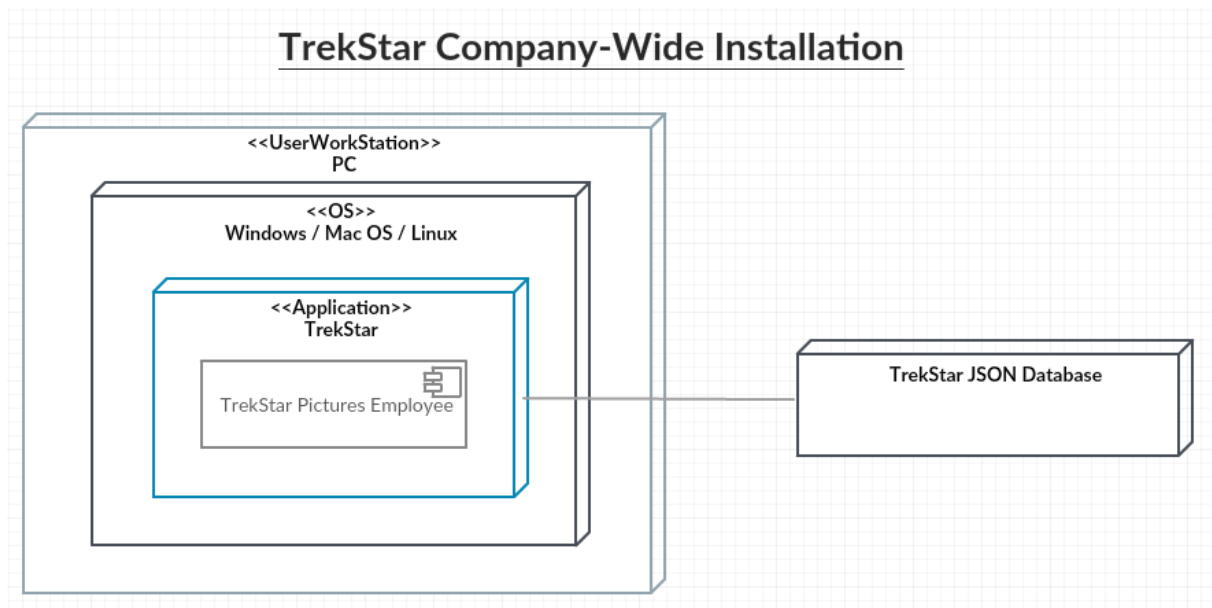# Included C++ Libraries (GitHub Link)

Within the TrekStar project management application we have used some C++ Libraries for parsing data, providing access to unit tests and including core functionality that would otherwise not be present. Below you can find an exaplaination of the included C++ libraries and why we chose to use them within the project:

JSON for C++

We decided to use the JSON for C++ library within the application because we are using JSON as our database to store all of the users data. The JSON for C++ library allows us to parse JSON data, request data based on certain keys within a JSON Array or Object and create new JSON Arrays or Objects to be inserted into the database. The JSON for C++ library is quite easily the most feature complete and well documented library for parsing and creating JSON in C++, this was the main driving force behind using this library as we required something that is reliable and suitable for the applications use cases.

GTest (GitHub Link)

We decided to use GTest as our primary unit testing library within the application, whilst we could have used Boost for unit testing, we decided that GTest had the following advantages which solidifed our decision to use GTest:

- GTest is much more lightweight when compared to Boost, while GTest is specifically a testing library, Boost has more functionality that also needs to be imported to allow for the Unit Testing section of the library to function, consequently making it larger and less lighweight.
- When compared to other Unit Testing libraries there are more features that we can make use of in the project within GTest, for example, we can use GTests ability to mock objects, this is useful when we need to test the interaction between objects as we can achieve this more efficiently when compared to other libraries. We can also avoid duplication of test setup (creating objects, etc) within each unit test by specifying test fixtures which can be referenced within the unit tests.

spdlog (GitHub Link)

We decided to use spdlog for our logger library, when compared to other libraries spdlog is lightweight, fast and only requires header imports to get started. spdlog also supports all of the operating systems that our application has been developed to work on, which is an important feature to have.

spdlog also allows us to easily generate the required daily reports based on the data we pass from the JSON database, the simplicity of this library was an important consideration as it allows future developers to quickly understand how this function works incase a new feature or bug needs to be added/fixed.

Internal Data Structures

Include explaination of what internal data structures were used.

Search/Sorting Algorithm

Include explaination of the search/sorting algo used.

UI Screenshots

Include screenshots of console app here.

Software Testing Prodcedure

Include explaination of software testing procedure here.

User Manual

Software Introduction

The Trekstar project management application has been developed for TrekStar Pictures and allows them to create and manage Projects along with their associated product materials and metadata. This guide will cover how to use the core features of the application, which includes:

- Using the main menu
- Using view mode
- Using maintenance mode

Using the Main Menu

To use the Main Menu, open the application and you will be presented with the following screen

Figure 3: Visual representation of the Main Menu

Once on this screen, you can use the following keyboard inputs to navigate around the application, to find out more about each feature, click the associated information link.

Input Value	Command Action	Information Link
1	Inputting this value will cause the application to enter View Mode	Using View Mode
2	Inputting this value will cause the application to enter Maintenance Mode	Using Maintenance Mode
3	Inputting this value will cause the application to safely exit	Not required

Using View Mode

Once you have reached the View Mode page you will be presented with the following screen

Figure 4: Visual representation of View Mode

Once on this screen, you can use the following keyboard inputs to navigate around view mode, to find out more about each feature, click the associated information link.

Input Value	Command Action	Information Link
1	Inputting this value will cause the application to list projects	Listing Projects
2	Inputting this value will cause the application to enter search mode for materials	Using Search Projects
3	Inputting this value will cause the application to list materials	Listing Materials
4	Inputting this value will cause the application to navigate back to the Main Menu	Not required

Using List Projects

Once you have entered the List Projects functionality you will be asked if you wish to sort all of the projects by their title, you can enter *y* to confirm that you wish for the projects to be order by their titles and *n* to decline.

Figure 5: Visual representation of the application asking if you wish to sort by title

You will now see the projects that are held within your applications database, you can see an example of this below.
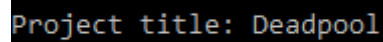
Figure 6: Visual representation of project listings

As you can see in Figure 6, there are more options avaliable to you, these options allow you to navigate through the projects. You can see the associated input actions below.

Input Value	Command Action
1	This will cause the application to move onto the next project in the database
2	This will cause the application to move onto the previous project in the database
3	This will cause the application to navigate back to View Mode

Using Search Projects

Once you have entered the Search Projects functionality you will be asked to input a search query, this can be any value and data will only be returned if there is an associated project title held within the database.

Figure 7: Visual representation of the application asking for a search query

You will now see the associated projects that matched to your search query, you can see an example of this below.
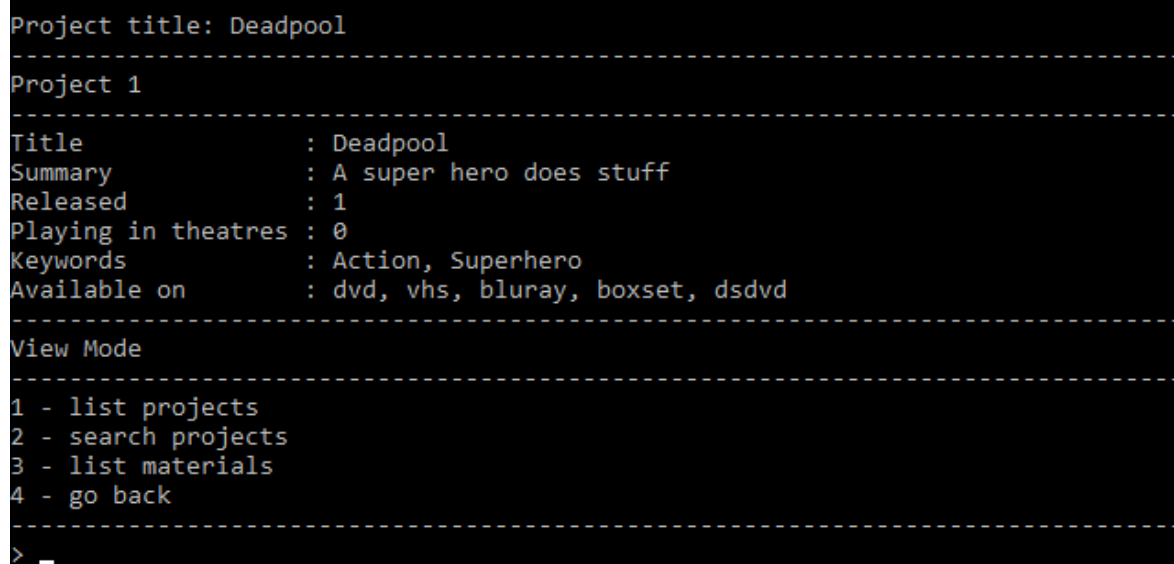

Figure 8: Visual representation of a project being discovered from a search query

As you can see in Figure 8, there are more options avaliable to you, these are the options associated with View Mode, therefore you can find more information by reading the View Mode section of this user guide.

Using List Materials

Once you have entered the List Materials functionality you will be asked to input a Project ID, this must be a numeric value, you can find the accepted Project IDs above the input location, select an ID from this list.

Visual representation of selecting a Project ID to list materials for

You will now see the materials linked to the requested project that are held within your applications database, you can see an example of this below.

```
-------------------------------------------------------------------------
Material 0
-------------------------------------------------------------------------
Additional Language Track 1 : FRA
Additional Language Track 2 : GER
Additional Subtitle Track 1 : FRA
Additional Subtitle Track 2 : GER
Audio Format : DTS
Bonus Features 1 : Director's Comments
Content : big movie
Format : dvd
Frame Aspect : 16:9
Language : ENG
Retail Price : 9.990000
Run Time : 120
Subtitles : ENG
-------------------------------------------------------------------------
List Materials
-------------------------------------------------------------------------
1 - next material
2 - previous material
3 - go back
-------------------------------------------------------------------------
>
```

Figure 9: Visual representation of material listings

As you can see in Figure 10, there are more options avaliable to you, these options allow you to navigate through the materials. You can see the associated input actions below.

Input Value	Command Action
1	This will cause the application to move onto the next material in the database
2	This will cause the application to move onto the previous material in the database
3	This will cause the application to navigate back to View Mode

Using Maintenance Mode

Once you have reached the Maintenance Mode page you will be presented with the following screen
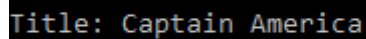
Visual representation of Maintenance Mode

Once on this screen, you can use the following keyboard inputs to navigate around maintenance mode, to find out more about each feature, click the associated information link.

Input Value	Command Action	Information Link
1	Inputting this value will cause the application to enter add project	Adding Projects
2	Inputting this value will cause the application to enter remove project	Removing Projects
3	Inputting this value will cause the application to save all changes to the database	Not required
4	Inputting this value will cause the application to navigate back to the Main Menu	Not required

Using Add Project

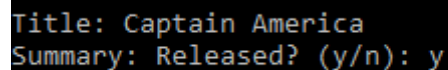Once you have entered the Add Project functionality you will be asked to enter the following information:

Firstly, you will be asked to enter the Project Title, this is the name that you want your project to be called. Any value is accepted.

Figure 10: Visual representation of the application asking for a Project Title
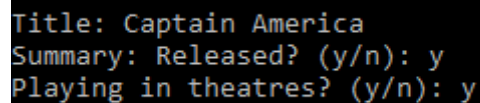
Next, you will be asked if the project has been released. Only the values *y* and *n* are accepted. With *y* confirming that the project has been released and *n* confirming that the project has not been released.

Figure 11: Visual representation of the application asking if the project has been released
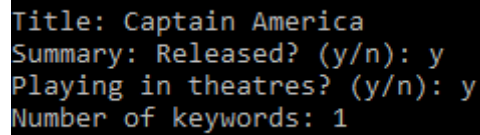
Next, you will be asked if the project will be playing in theaters. Only the values *y* and *n* are accepted. With *y* confirming that the project will be playing in theatres and *n* confirming that the project will not be playing in theatres.

Figure 12: Visual representation of the application asking if the project will be playing in theatres
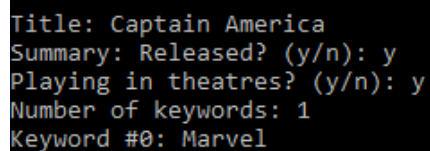
Next, you will be asked how many keywords you would like associating with the project. Any numeric value is accepted.

Figure 13: Visual representation of the application asking how many keywords should be associated with the project

Finally, you will be asked what keywords you would like associated with the project, the amount of keywords requests depends on the previously entered numeric value. Any value is accepted.

Figure 14: Visual representation of the application asking for a keyword

As you can see in Figure 16, there are more options avaliable to you, these are the options associated with Maintenance Mode, therefore you can find more information by reading the Maintenance Mode section of this user guide.

Using Remove Projects

todo

Conclusion

Discussion and conclusion about your results (reflection on testing approach, reflection on performance such as computational efficiency, reliability, security, portability, maintainability, scalability, etc. design of system complexity using e.g. big O- notation).

Appendix

Managing group work

Whilst working on this project our group used a multitude of methods to ensure that we were collaborating efficiently and getting the most amount of work completed as possible, these methods included, but are not limited to:

- Using Facebook Messenger to converse with other group members about certain aspects of the program or documentation, this allowed us to keep in touch and ensure we were on track with the workload.
- Using GitHub to its full potential, this includes the use of GitHub Issues and Pull Requests in conjunction with code reviews for each major feature. Whenever a bug or issue was discovered the group member that uncovered the issue would create a GitHub issue. If the bug was within another group members individual work then that member would be added as a participant so that they could fix the issue. Once large features were added to the program, pull requests were used with each group member being requested to review the code before merging into the master branch.

Overall, our experience of working as a group has been productive and enjoyable, being able to distrubute the workload across group members was an important aspect to the project and allowed us to complete it efficiently. We have all taken learning points from this experience that could be applied to the workplace, or even in our upcoming placements. For example:

- The use of GitHub's features (such as pull requests, issues and code reviewing) will be an excellent skill to have within the workplace, as more often than not, organisations will have some sort of code review system/Quality Assurance (QA) procedure in place and being able to understand how this all works without training will show that you are a competent employee.
- The use of a messaging platform was also important, sometimes organisations have employees working accross the globe and having the ability to efficiently collaborate on messaging systems (such as Slack) is an important skill to have.

Meeting Minutes

During group meetings a member was nominated to write minutes, these minutes were an important factor during the development of the application as they allowed us to go back and check what work we have all be allocated. On the following pages you can find the minutes for the aformentioned meetings.

SDI Strand Group Meeting 1 | MINUTES

Meeting date | time *4ᵗʰ February* | *11:00am* | Meeting location *Pavilion*

Type of meeting	Introduction	Callum Axon (CA)
		Callum Carney (CC)
		Matthew Robinson (MR)

AGENDA TOPICS

Time allotted | *10 minutes* | Agenda topic *SDI Introduction & Setup* |

It has been agreed by members that Software Design & Implementation Meetings are going to take place before the lecture which takes place on Mondays between 13:00pm and 14:00pm. These meetings are likely to last up to an hour.

CA to create the GitHub repository that will hold all of the code for the project and documentation.

MR to create the project and commit the basic files to GitHub in which CA & CC can then clone the repo.

Time allotted | *5 minutes* | Agenda topic *Documentation formatting* |

CA brings up the point of using LaTex instead of Microsoft Word for creating the documentation, CC & MR agree with CA.

CC brings up the possibility of using Pandoc with a LaTex template that will allow us to easily display code and documentation content easily as Markdown is supported.

CA and MR agree, CC to create the base documentation files and commit to git.

Action items	Person responsible	Deadline
Create GitHub Repository	CA	ASAP
Create project and commit basic files	MR	ASAP
Create LaTex documentation files	CC	ASAP

Meeting closed at 11:15 am.

Figure 15: Meeting 1 - Minutes

SDI Strand Group Meeting 2 | MINUTES

Meeting date | time *18th February | 11:00am |* Meeting location *Pavilion*

Type of meeting   Progress Check | Callum Axon (CA)
                                      | Callum Carney (CC)
                                      Matthew Robinson (MR)

AGENDA TOPICS

Time allotted | *15 minutes* | Agenda topic *Progress of previous meetings action items* |

CA confirms that the GitHub repository has been created and our accounts have been added as collaborators.

MR confirms that the project files have been created and committed to git. MR shows CA & CC the structure of the project files.

CC confirms that the LaTeX files have been created and points CA & MR to a guide created by CC on how to setup LaTeX on their local machine in order to edit the documentation.

Time allotted | *5 minutes* | Agenda topic *Member allocations* |

CA states that we need to allocate member letters in accordance with the Spec.

CA agrees to be Member A, CC agrees to be Member C, MR agrees to be Member B

All members agree that we should start working on our individual tasks.

CC states that he will create a Facebook Messenger group chat in order for us to collaborate more efficiently.

Action items	Person responsible	Deadline
Create Facebook Messenger group	CC	ASAP
Work on associated individual tasks	All	ASAP

Meeting closed at 11:20 am.

Figure 16: Meeting 2 - Minutes

SDI Strand Group Meeting 3 | MINUTES

Meeting date | time *4th March* | *11:00am* | Meeting location *Pavilion*

Type of meeting   Progress Check   Callum Axon (CA)

Callum Carney (CC)

Matthew Robinson (MR)

AGENDA TOPICS

Time allotted | *10 minutes* | Agenda topic *Checkup* |

CC created the group chat and all group members have been collaborating.

MR shows the progress on his individual tasks, so do CA & CC.

All group members are happy with the current progress.

Action items	Person responsible	Deadline
Work on associated individual tasks	All	ASAP

Meeting closed at 11:10 am.

Figure 17: Meeting 3 - Minutes

SDI Strand Group Meeting 4 | MINUTES

Meeting date | time *18th March* | *11:00am* | Meeting location *Pavilion*

Type of meeting   Progress Check   Callum Axon (CA)

Callum Carney (CC)

Matthew Robinson (MR)

AGENDA TOPICS

Time allotted | *10 minutes* | Agenda topic *Checkup* |

CA shows the progress on his individual tasks, so do CC & MR.

All group members are still happy with the current progress and no interventions need to be made.

MR thinks that the group no longer needs to convene, unless the rate of progression decreases. CA & CC agree. Group meetings will no longer occur on Mondays.

Action items	Person responsible	Deadline
Work on associated individual tasks	All	ASAP

Meeting closed at 11:10 am.

Figure 18: Meeting 4 - Minutes

SDI Strand Group Meeting 5 | MINUTES

Meeting date | time 8ᵗʰ April | 14:00pm | Meeting location Skype

Type of meeting   Discussion   Callum Axon (CA)

                                                       Callum Carney (CC)

                                                       Matthew Robinson (MR)

AGENDA TOPICS

Time allotted | 45 minutes | Agenda topic *GUI discussion* |

The group felt it necessary to convene in order to discuss the point of having a GUI versus a Command Line Interface.

The group discusses the possibility of using Qt for the GUI, amongst other potential GUI libraries.

MR brings up the point of the GUI being an optional point and that we should focus on other tasks before considering the implementation of a GUI.

CA & CC agree, CC brings up that the Specification states that for a first the application must have an "Detailed user interface and its instruction", this does not state Graphical User Interface. Therefore the group can still achieve a first without the implementation of a GUI.

All group members agree that a GUI should be held off and reconsidered later.

Action items	Person responsible	Deadline
Work on associated individual tasks	All	ASAP

Meeting closed at 14:45 pm.

Figure 19: Meeting 5 - Minutes