
Software Design and Implementation

TrekStar Software System

Callum Axon (N0727303), Callum Carney (N0741707),
Matthew Robinson (N0724629)



Table of Contents

- Table of Contents
- Todo - ***remove me before submission pls***
 - User Guide
- Contributions
 - Callum Axon (N0727303) - Member A
 - Callum Carney (N0741707) - Member C
 - Matthew Robinson (N0724629) - Member B
- System Description
 - Cohesion and Coupling Considerations
 - * Cohesion
 - * Coupling
- Component Diagram
- Deployment Diagram
 - Individual Installation
 - Company-Wide Installation
- Design Pattern
 - Factory Design Pattern
- Planned Architecture
- Included C++ Libraries
 - JSON for C++ (GitHub Link)
 - GTest (GitHub Link)
 - spdlog (GitHub Link)
- Internal Data Structures
- Search/Sorting Algorithm
- UI Screenshots
- Software Testing Procedure
- User Manual
 - Software Introduction
 - Using the Main Menu
 - Using View Mode

- * Using List Projects
- * Using Search Projects
- * Using List Materials
- Using Maintenance Mode
 - * Using Add Project
 - * Using Update Project
 - Edit Title
 - Edit Summary
 - Edit Released
 - Edit Playing In Theaters
 - Edit Existing Keywords
 - * Using Remove Project
 - * Using Update Project Materials
 - Edit Audio Format
 - Edit Run Time
 - Edit Language
 - Edit Retail Price
 - Edit Subtitles
 - Edit Frame Aspect
 - Edit Packaging
 - * Using Remove Project Materials
- Conclusion
- Appendix
 - Managing group work
 - Meeting Minutes

Todo - *remove me before submission pls*

User Guide

- User guide details required for when the application is able to manage crew and materials/projects fully.
- Add back in image that shows keywords when adding a project when this is reimplemented.
- Change all figure values where text is CHANGEFIGUREVALUE once all images added.
- Complete Using Update Project Materials once the material id bug has been fixed.
- Resolve image issue on Edit Packaging once bugs have been fixed
- Look into new line for # # # # headers
- Add class diagram (and any other landscape picture pages) to the TOC on last save

Contributions

Callum Axon (N0727303) - Member A

- Class Diagram.
- One of the data structures - stack.
- A description of how cohesion and coupling have been considered in the class design.
- A justification and explanation of how cohesion and coupling have been considered in the design.
- 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).
- An explanation of any design pattern used.

Callum Carney (N0741707) - Member C

- A cover page (showing the full name and student ID of all members).
- A table of contents page and identifying who has contributed to which individual tasks.
- A general description of the system.
- Component diagrams.
- Deployment diagrams.
- An explanation of the internal data structures used and the reason of the choices.
- Examples (screen shots) of user interface.
- A user manual and instruction of the software. (regardless if your software is based on GUI or console interface).
- An explanation of any C++ library used.
- An explanation about the software testing process and metrics.
- Meeting minutes.

Matthew Robinson (N0724629) - Member B

- A sequence diagram for a case/scenario of interest.
- State machine diagram for a class.
- One of the sorting studied during the lectures - merge sort.
- One of the searching algorithms studied during the lectures - binary search.
- A console interface that allows user to interact with the software system.
- An explanation of the search or sorting algorithm used and the reason of the choices. Explain how the algorithm will work in the system with detailed steps.

- 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).

All other contributions have been completed as part of a group effort.

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

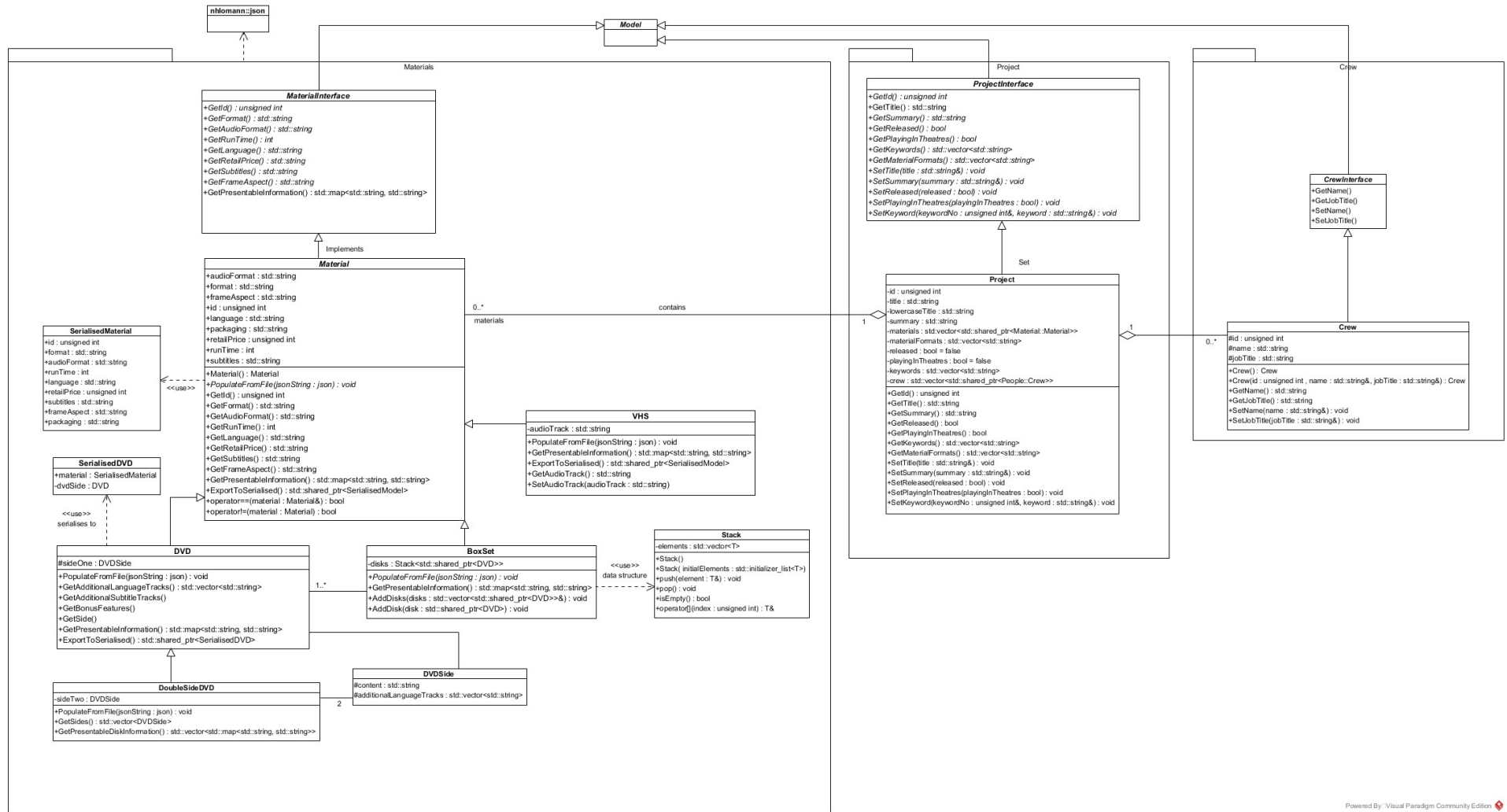


Figure 1: Class diagram

Cohesion and Coupling Considerations

Cohesion

The concept of cohesion is defined as “the degree to which all elements of a component are directed towards a single task, within a single component, or to which the responsibilities of a class are related”.

Within the design, it was important to separate out the logic for the presentation of the information from the business logic. We have achieved this using a Model - View - Controller architecture. In order for the views to gather data from the Models themselves, a large number of *so-called* getter functions are present. This is so that the models are not responsible for the presentation and **logical cohesion** does not occur. Utilising these functions also provides an example of how communicational cohesion has been considered within the design, whilst the separation of these elements has provided the situation of functional cohesion.

Another example of how **logical cohesion** has been avoided is the approach taken to importing data from files. This logic is contained within its own object and the only data which is passed into their respective objects is a *serialised* version - separating the concerns surrounding the importing of files.

Further details on cohesion, relating to specific design patterns, can be found in that section of the document.

Coupling

Due to the relationships between different objects within the system, some coupling does occur. An example of this is that a Project has Materials associated with it - thus creating a dependency between the two objects. However, the relationships defined within the class diagram show that the two related objects don't modify the data of each other, but store data about each other independently. The data between these two objects could therefore be considered to be uncoupled.

Sequence Diagram

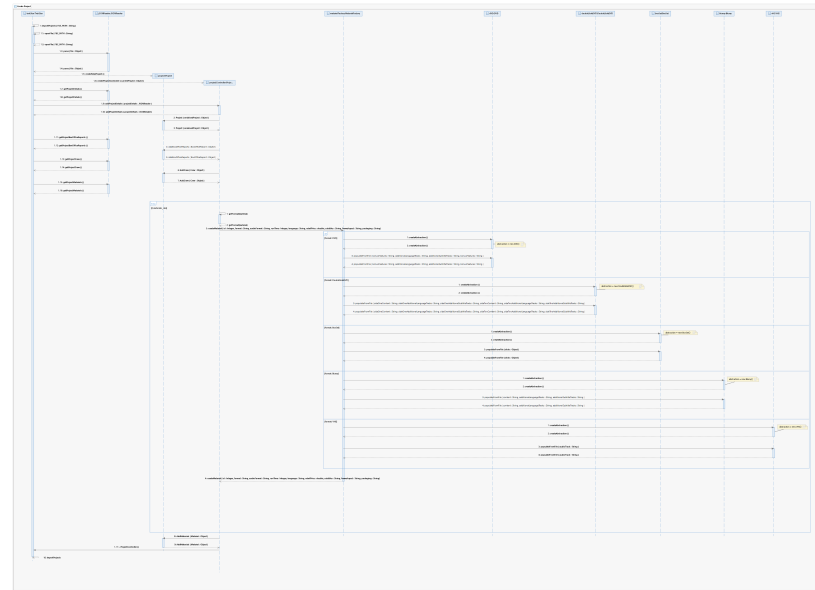


Figure 2: Sequence diagram

State Diagram

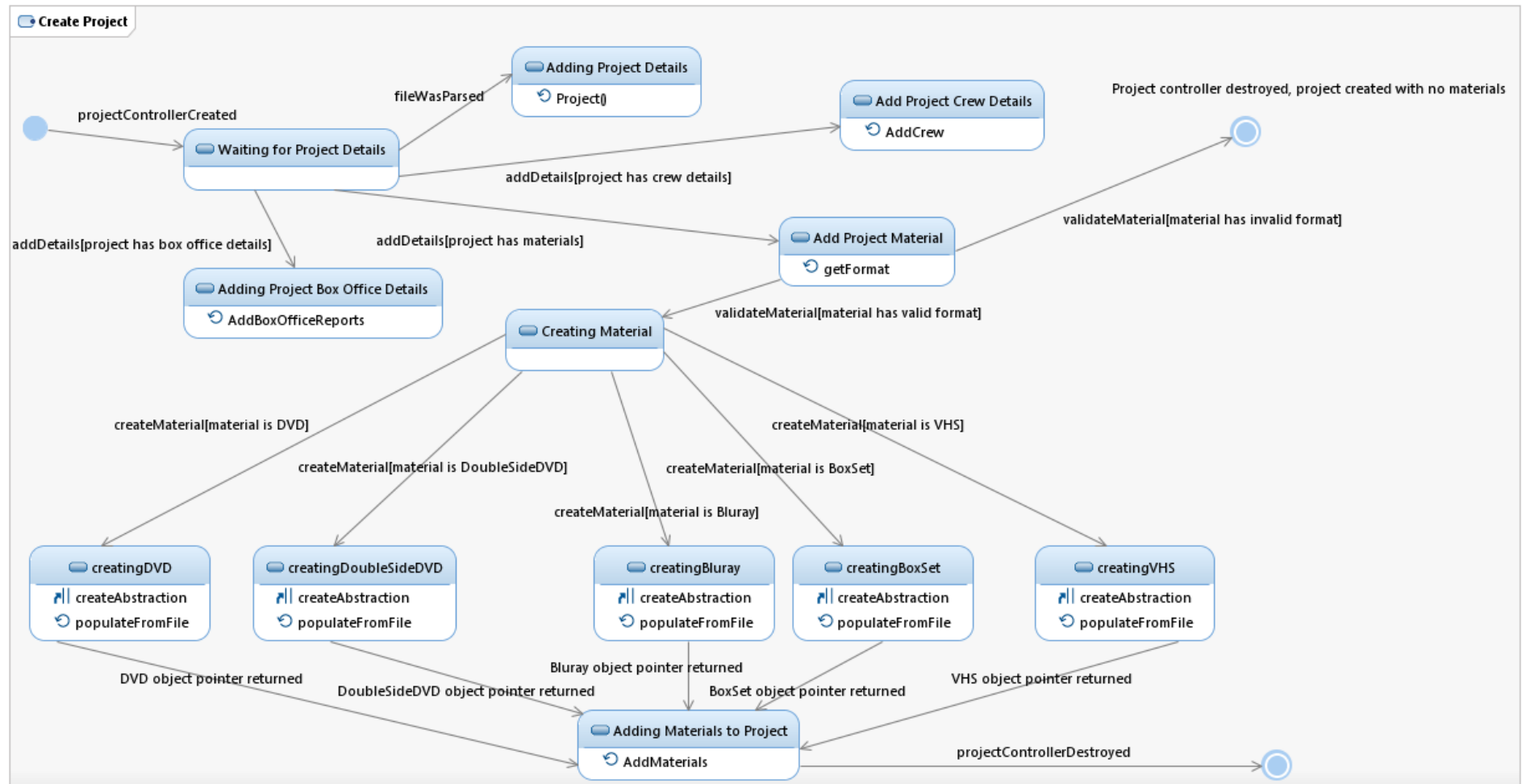


Figure 3: State diagram

Component Diagram

Include component diagram here

Deployment Diagram

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

1. An individual 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 aforementioned Deployment scenarios.

Individual Installation

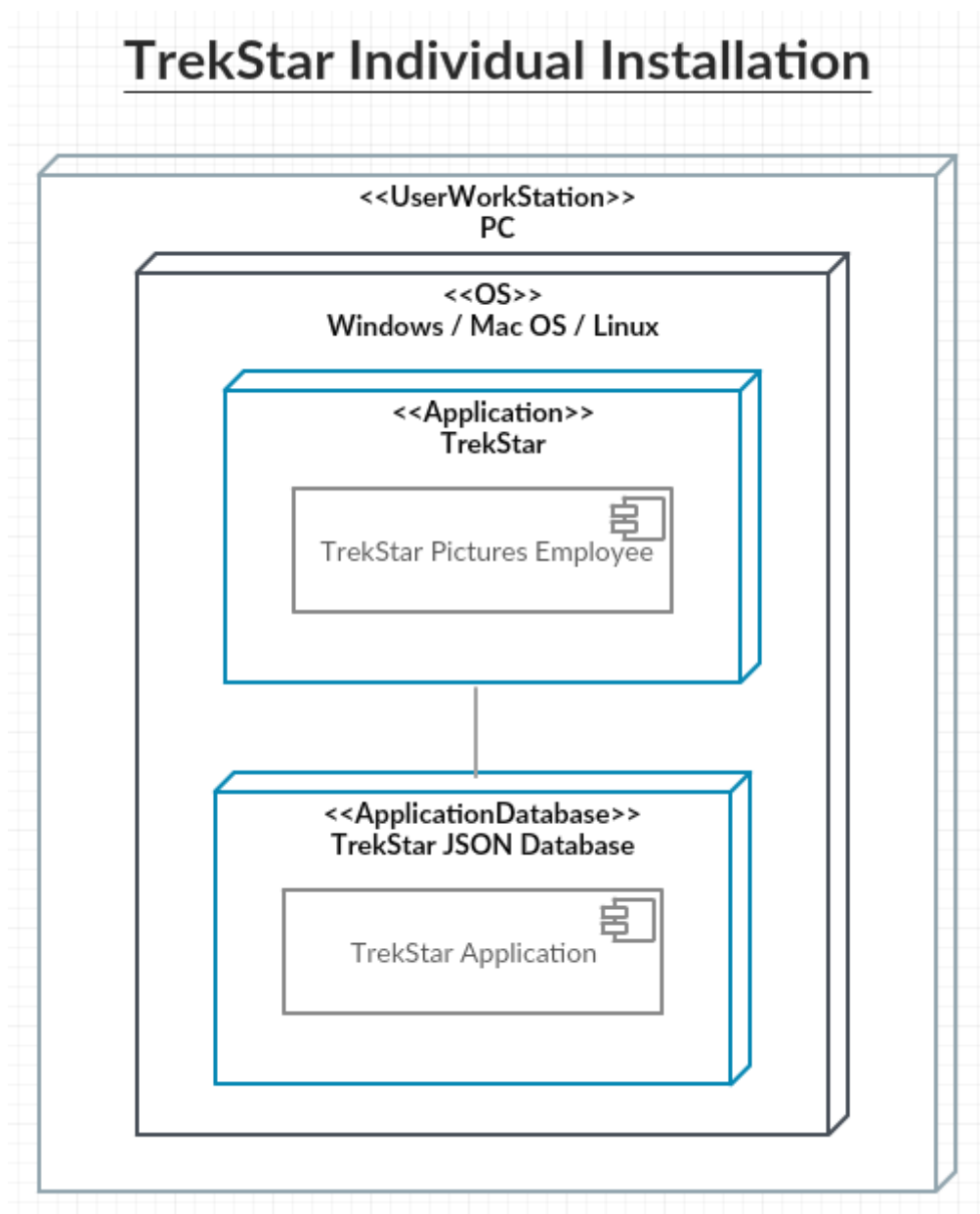


Figure 4: Visual Representation of Individual TrekStar installation

Company-Wide Installation

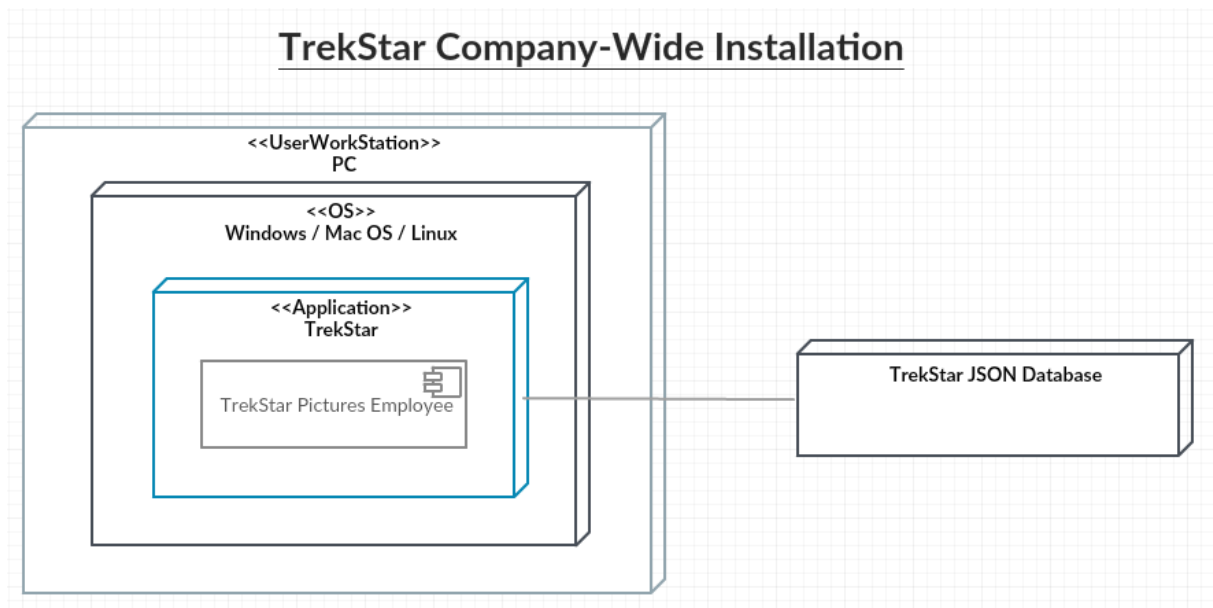


Figure 5: Visual Representation of Company-Wide TrekStar installation

Design Pattern

Factory Design Pattern

Include explanation of any design patterns used

With a large number of *materials* defined within the business rules, it was appropriate to use a **factory design pattern** to generate different material types based upon an input provided by the user. The specific implementation involved defining a return type as the base material class - allowing a covariant return type. This design pattern was possible as we have a common interface for every material. A UML representation of this pattern can be found below.

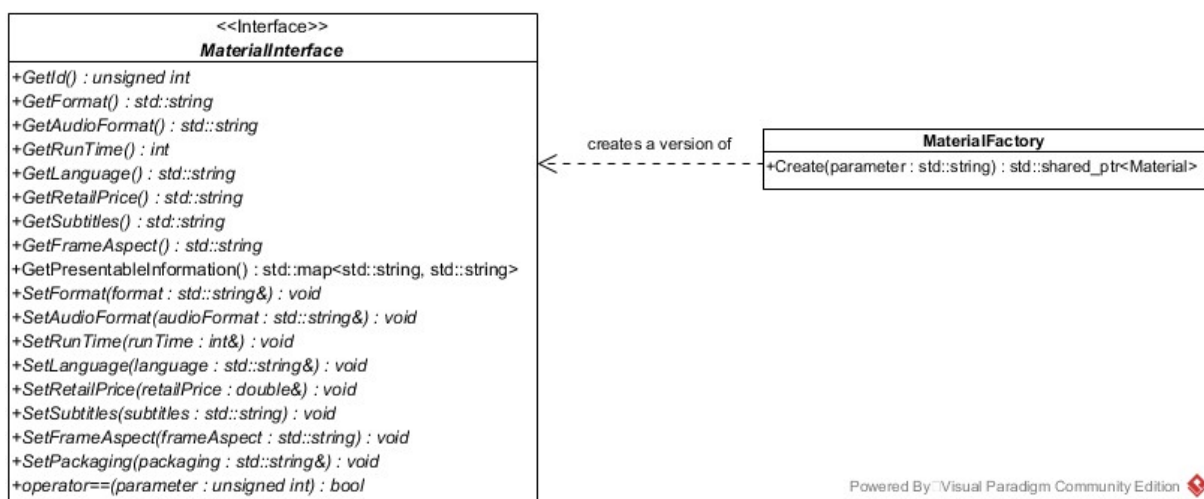


Figure 6: MaterialFactory UML Class Diagram

The advantages of this pattern is that new materials types can easily be added by adding a small conditional statement in this class. This provides a level of abstraction and means that this class has only a single responsibility - to construct materials - removing this logic from the consuming object. This helps with the issue of coupling as functional cohesion is achieved. Because of the abstraction, it wouldn't mean any changes would necessarily have to occur in the consuming class.

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

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 explanation of the included C++ libraries and why we chose to use them:

JSON for C++ ([GitHub Link](#))

It was 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](#))

It was 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 solidified 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 lightweight.
- 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](#))

It was 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

During the design phase of the TrekStar project management application it was decided that an internal data structure of Stack would be used. While there are pros and cons to each data structure, ultimately it was decided that the Stack data structure fit the best with the requirements of our application, you can find the in-depth reasoning behind the decision to use Stack below:

- The Stack data structure allows for the sequential ordering of disks, and make sense within our use cases. For example, if you have a box set of 4 disks and you would like to add a new disk, its likely that the new disk will be the latest episode, therefore it will be inserted sequentially at the end of the list. Another example would be when a user would like to remove a disk from a boxset, it is unlikely that you will want to remove the 3rd episode within the boxset, however if the 3rd episode was to be removed then the 4th episode (the last one in the underlying vector) can be removed allowing for the 3rd episode to then be removed, therefore the Stack data structure is still appropriate.
- Another advantage to the Stack data structure is that data can be accessed quicker sequentially, this is because you can iterate though the underlying vectors data finding items that a requested.
- A final advantage to the Stack data structure is that it uses general programming through the use of templating, as a consequence to this, the Stack data structure supports multiple types.

Search/Sorting Algorithm

Include explanation of the search/sorting algo used.

UI Screenshots

Include screenshots of console app here.

Software Testing Procedure

Software Testing is an important part of the SDLC and within our project we have multiple constraints in place to ensure that tested, bug-free code is committed to the GitHub repository. Committing working code to the repository is important for a number of reasons, with some of these being:

- Developers will never pull broken code that will lead to a slow down in development.
- Developers will never have to fix someone else's broken code, avoiding time wasted on parts of the application that is not allocated to that developer.
- In case of a rollback, the repository will always be in a good state, which helps to avoid further rollbacks.

Some of the constraints that we have in place are:

- Developers will only commit code that has the associated unit tests created and those unit tests must pass.
- Developers must never commit broken code that could impede the efficiency of other developers, if a commit is required a new branch should be created.
- If there are known bugs, developers must log these within the GitHub Issues to ensure that other developers know that it is known/being worked on in case they come across it.

Our group is confident that these constraints will allow us to develop the application with fewer interruptions and ensure that only high quality, tested code is being pushed to the repository.

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

```
-----
#####  #####  #####  ##  ##  #####  #####  ###  #####
##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
##  #####  #####  #####  #####  ##  ##  ##  #####
##  ##  ##  ##  ##  ##  ##  ##  ##  #####  ##  ##
##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
##  ##  ##  #####  ##  ##  #####  ##  ##  ##  ##  ##
-----

Main Menu
-----
1 - View Mode
2 - Maintenance Mode
3 - Quit
-----
>
```

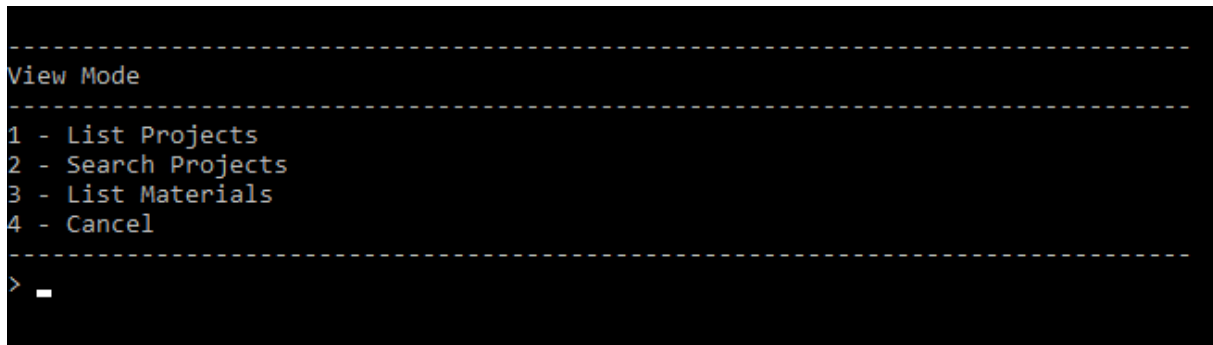
Figure 7: Visual representation of the Main Menu

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



```
-----
View Mode
-----
1 - List Projects
2 - Search Projects
3 - List Materials
4 - Cancel
-----
> _
```

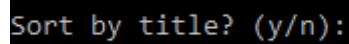
Figure 8: Visual representation of View Mode

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 projects	Using Search Projects
3	Inputting this value will cause the application to navigate back to the main menu	Main Menu

Using List Projects

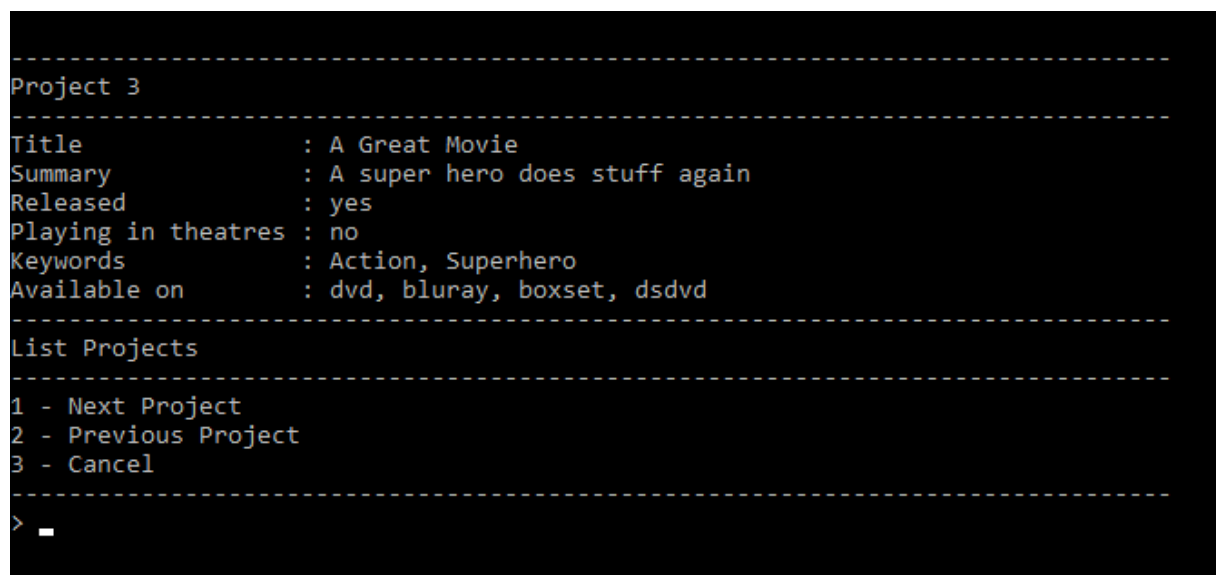
Once you have entered the List Projects functionality you will be asked if you wish to sort all of the projects by their title, you can enter *y* to confirm that you wish for the projects to be order by their titles and *n* to decline.



```
Sort by title? (y/n):
```

Figure 9: 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.



```
-----
Project 3
-----
Title           : A Great Movie
Summary         : A super hero does stuff again
Released        : yes
Playing in theatres : no
Keywords        : Action, Superhero
Available on     : dvd, bluray, boxset, dsdvd
-----
List Projects
-----
1 - Next Project
2 - Previous Project
3 - Cancel
-----
> _
```

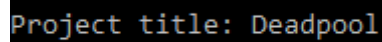
Figure 10: Visual representation of project listings

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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

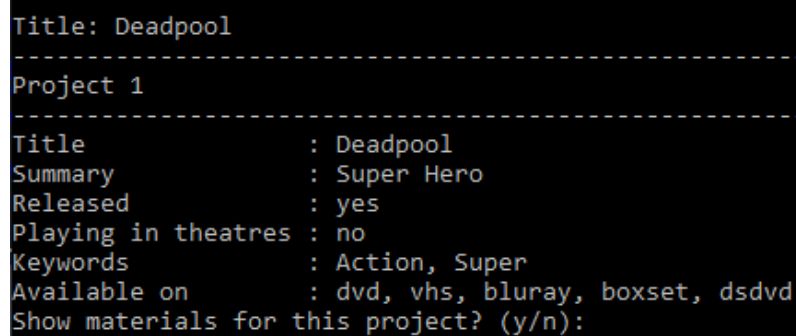
Once you have entered the Search Projects functionality you will be asked to input a search query, this can be any value and data will only be returned if there is an associated project title held within the database.



```
Project title: Deadpool
```

Figure 11: 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.



```
Title: Deadpool
-----
Project 1
-----
Title           : Deadpool
Summary         : Super Hero
Released        : yes
Playing in theatres : no
Keywords        : Action, Super
Available on    : dvd, vhs, bluray, boxset, dsdvd
Show materials for this project? (y/n):
```

Figure 12: Visual representation of a project being discovered from a search query

As you can see in Figure CHANGEFIGUREVALUE, there is the option to display the associated materials for this project, you can enter y to confirm that you wish to view the materials for the associated project, which will bring up the following information:

```
Title: Deadpool
-----
Project 1
-----
Title           : Deadpool
Summary         : Super Hero
Released        : yes
Playing in theatres : no
Keywords        : Action, Super
Available on    : dvd, vhs, bluray, boxset, dsdvd
Show materials for this project? (y/n): y
-----
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
Packaging :
Retail Price : 9.990000
Run Time : 120
Subtitles : ENG
-----
List Materials
-----
1 - Next Material
2 - Previous Material
3 - Cancel
-----
> _
```

Figure 13: Visual representation of a projects materials being displayed

As you can see in `CHANGEFIGUREVALUE`, there are more options available 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.

You can also enter *n* to confirm that you do not want to view the materials for the associated project, which will bring up the following information:

```
Title: Deadpool
-----
Project 1
-----
Title           : Deadpool
Summary         : Super Hero
Released        : yes
Playing in theatres : no
Keywords        : Action, Super
Available on    : dvd, vhs, bluray, boxset, dsdvd
Show materials for this project? (y/n): n
-----
View Mode
-----
1 - List Projects
2 - Search Projects
3 - Cancel
-----
> _
```

Figure 14: Visual representation of a projects materials not being displayed

As you can see in CHANGEFigureValue, there are more options available 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.

```
-----  
ID: 3, Title: A Great Movie  
ID: 4, Title: Back To The Future  
ID: 1, Title: Deadpool  
ID: 2, Title: Superheroes  
-----  
Project ID: 3_
```

Figure 15: 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 : Dolby  
Bonus Features 1 : Director's Comments  
Content : big movie  
Format : dvd  
Frame Aspect : 16:9  
Language : ENG  
Packaging :  
Retail Price : 9.990000  
Run Time : 120  
Subtitles : ENG  
-----  
List Materials  
-----  
1 - Next Material  
2 - Previous Material  
3 - Cancel  
-----  
>
```

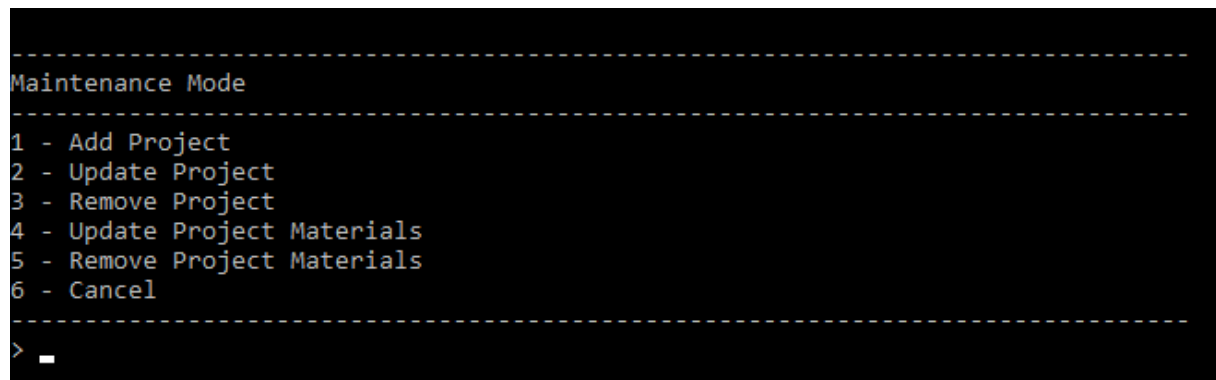
Figure 16: Visual representation of material listings

As you can see in Figure 10, there are more options available 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



```
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Remove Project
4 - Update Project Materials
5 - Remove Project Materials
6 - Cancel
-----
> _
```

Figure 17: Visual representation of Maintenance Mode

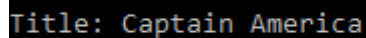
Once on this screen, you can use the following keyboard inputs to navigate around maintenance mode, to find out more about each feature, click the associated information link.

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 update project	Updating Projects
3	Inputting this value will cause the application to enter remove project	Removing Projects
4	Inputting this value will cause the application to enter update project materials	Updating Project Materials
5	Inputting this value will cause the application to enter remove project materials	Removing Project Materials
6	Inputting this value will cause the application to navigate back to the main menu	Main Menu

Using Add Project

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

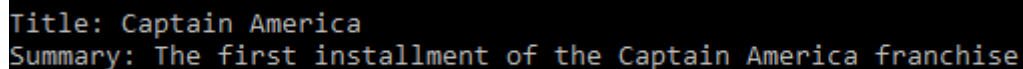
Firstly, you need to enter the Project Title, this is the name that you want your project to be called. Any value is accepted.



```
Title: Captain America
```

Figure 18: Visual representation of the application asking for a Project Title

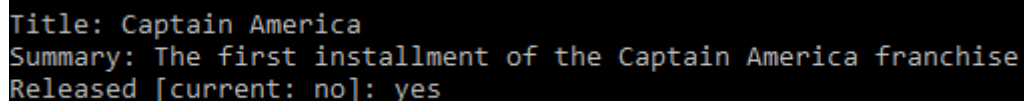
Next, you will be asked for a project summary. Any value is accepted.



```
Title: Captain America  
Summary: The first installment of the Captain America franchise
```

Figure 19: Visual representation of the application asking for Project Summary

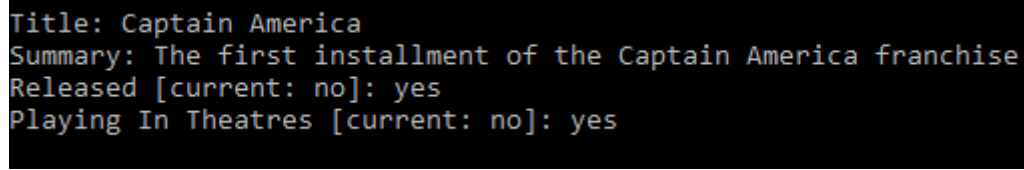
Next, you will be asked if the project has been released. Only the values *y* and *n* are accepted. With *y* confirming that the project has been released and *n* confirming that the project has not been released.



```
Title: Captain America  
Summary: The first installment of the Captain America franchise  
Released [current: no]: yes
```

Figure 20: 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.



```
Title: Captain America
Summary: The first installment of the Captain America franchise
Released [current: no]: yes
Playing In Theatres [current: no]: yes
```

Figure 21: 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.

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 keyword requests depends on the previously entered numeric value. Any value is accepted.

Visual representation of the application asking for a keyword

As you can see in `CHANGEFIGUREVALUE`, there are more options available 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 Update Project

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

Firstly, you need to enter 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.

```
-----  
[1]: Title: A Great Movie  
[2]: Title: test  
[3]: Title: title  
-----  
Project ID [1 - 3]: 1
```

Figure 22: Visual representation of selecting a Project ID to update the project for

Next, you will be presented with multiple options, you can use the following keyboard inputs to update certain aspects of the project, to find out more about each update, click the associated information link.

Input Value	Command Action	Information Link
1	Inputting this value will cause the application to enter edit title	Edit Title
2	Inputting this value will cause the application to enter edit summary	Edit Summary
3	Inputting this value will cause the application to enter edit released	Edit Released
4	Inputting this value will cause the application to enter edit playing in theatres	Edit Playing In Theatres
4	Inputting this value will cause the application to enter edit existing keywords	Edit Existing Keywords
5	Inputting this value will cause the application to navigate back to the maintenance mode menu	Maintenance Mode Menu

Edit Title Once you have entered the Edit Title functionality you will be asked to enter the following information:

You need to enter the new Project Title, this is the new name that you want your project to be called. Any value is accepted.

```
Title [current: A Great Movie]: An even greater movie
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 23: Visual representation of the application asking for a new Project Title

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Summary Once you have entered the Edit Summary functionality you will be asked to enter the following information:

You need to enter the new Project Summary, this is the new summary that you want your project to have. Any value is accepted.

```
Summary [current: A super hero does stuff again]: A great superhero movie
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 24: Visual representation of the application asking for a new Project Summary

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Released Once you have entered the Edit Released functionality you will be asked to enter the following information:

You need to enter the new Released value, this is the new released state that you want your project to have. Only the values *y* and *n* are accepted. With *y* confirming that the project is released and *n* confirming that the project is not released.

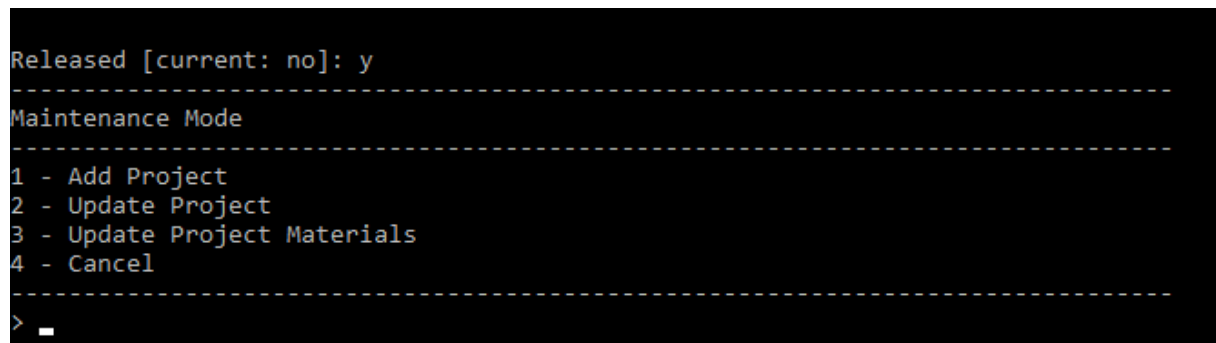
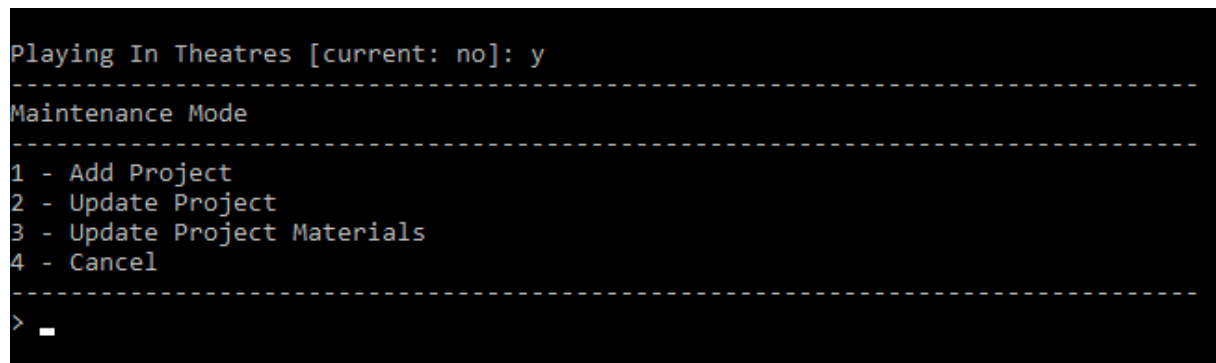
A terminal window with a black background and light blue text. The text shows a prompt 'Released [current: no]: y' followed by a dashed line separator. Below the separator is the text 'Maintenance Mode' followed by another dashed line. Underneath is a list of four options: '1 - Add Project', '2 - Update Project', '3 - Update Project Materials', and '4 - Cancel'. This is followed by a third dashed line and a prompt '> ' with a cursor.

Figure 25: Visual representation of the application asking for a new Released value

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Playing In Theaters Once you have entered the Edit Playing In Theaters functionality you will be asked to enter the following information:

You need to enter the new playing in theatres value, this is the new playing in theaters state that you want your project to have. 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.



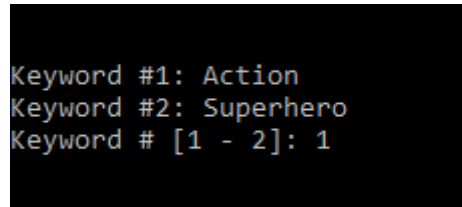
```
Playing In Theatres [current: no]: y
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 26: Visual representation of the application asking for a new Project Title

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Existing Keywords Once you have entered the Edit Existing Keywords functionality you will be asked to enter the following information:

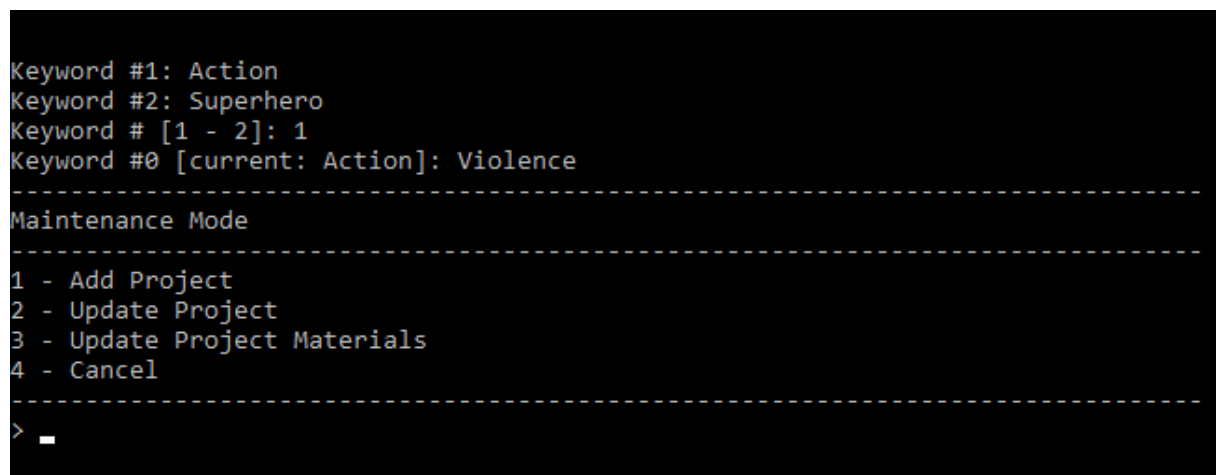
You need to enter the keyword identifier you wish to update, you can find the identifier in the provided list, an example of this can be seen below:



```
Keyword #1: Action
Keyword #2: Superhero
Keyword # [1 - 2]: 1
```

Figure 27: Visual representation of the application showing the current Keyword identifiers

Once you have inputted a valid keyword identifier you will be asked to enter your new Keyword, any value is accepted.



```
Keyword #1: Action
Keyword #2: Superhero
Keyword # [1 - 2]: 1
Keyword #0 [current: Action]: Violence
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 28: Visual representation of the application asking for a new keyword

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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 Project

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

You only need to enter 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.

Once you submit this Project ID it will be removed from the database.

```
-----  
[1]: Title: A Great Movie  
[2]: Title: Deadpool the movie  
[3]: Title: Deadpool 2  
-----  
Project ID [1 - 3]: 3
```

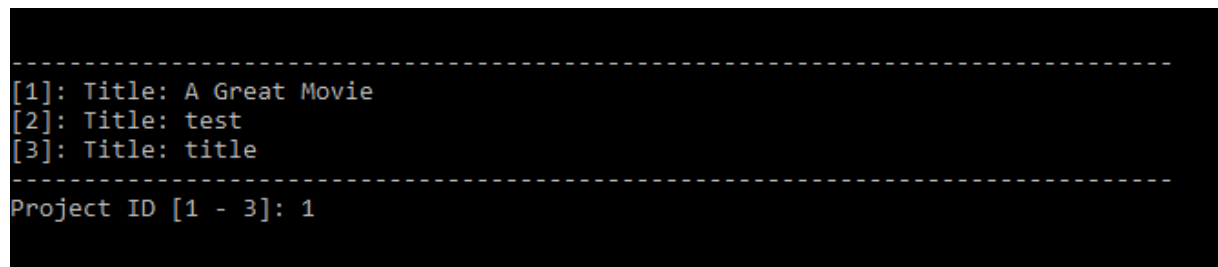
Figure 29: Visual representation of selecting a Project ID to update the project for

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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 Update Project Materials

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

Firstly, you need to enter 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.



```
-----  
[1]: Title: A Great Movie  
[2]: Title: test  
[3]: Title: title  
-----  
Project ID [1 - 3]: 1
```

Figure 30: Visual representation of selecting a Project ID to proceed with selecting a material to update

Next, you will be asked to enter the material identifier you wish to update within the previously selected project, you can find the identifier in the provided list, an example of this can be found below:

Visual representation of selecting a Material ID to update the material for

Next, you will be presented with multiple options, you can use the following keyboard inputs to update certain aspects of the material, to find out more about each update, click the associated information link.

Input Value	Command Action	Information Link
1	Inputting this value will cause the application to enter edit audio format	Edit Audio Format
2	Inputting this value will cause the application to enter edit run time	Edit Run Time
3	Inputting this value will cause the application to enter edit language	Edit Language
4	Inputting this value will cause the application to enter edit retail price	Edit Retail Price
5	Inputting this value will cause the application to enter edit subtitles	Edit Subtitles

Input Value	Command Action	Information Link
6	Inputting this value will cause the application to enter edit frame aspect	Edit Frame Aspect
7	Inputting this value will cause the application to enter edit packaging	Edit Packaging
8	Inputting this value will cause the application to navigate back to the maintenance mode menu	Maintenance Mode Menu

Edit Audio Format Once you have entered the Audio Format functionality you will be asked to enter the following information:

You need to enter the new Audio Format, this is the new audio format that you want your material to have. Any value is accepted.

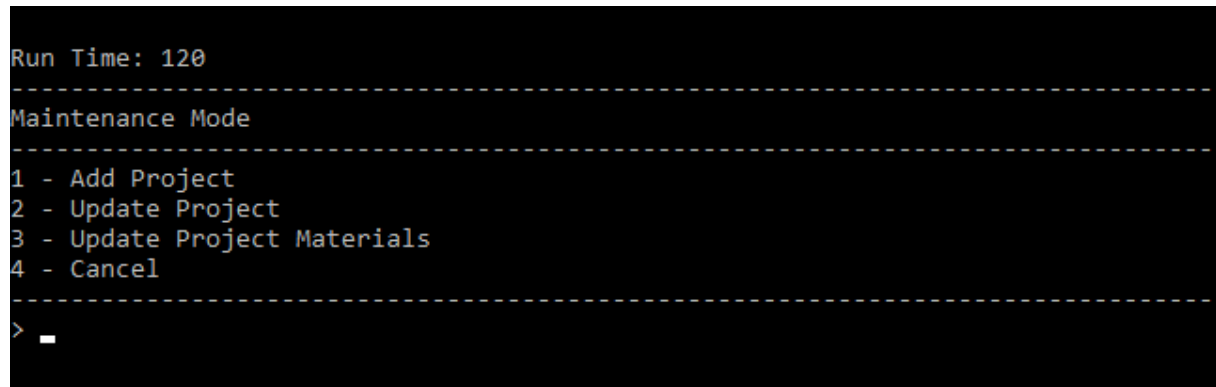
```
Audio Format: dolby
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 31: Visual representation of the application asking for a new Audio Format

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Run Time Once you have entered the Edit Run Time functionality you will be asked to enter the following information:

You need to enter the new Run Time, this is the new run time that you want your material to have. Any numeric value is accepted.

A screenshot of a terminal window with a black background and light blue text. The text shows the current Run Time as 120, followed by a dashed line separator. Below the separator is the heading 'Maintenance Mode', followed by another dashed line. A list of four options is displayed: '1 - Add Project', '2 - Update Project', '3 - Update Project Materials', and '4 - Cancel'. This is followed by a third dashed line and a prompt character '>' with a cursor.

```
Run Time: 120
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 32: Visual representation of the application asking for a new Run Time

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Language Once you have entered the Edit Language functionality you will be asked to enter the following information:

You need to enter the new Language, this is the new language that you want your material to have. Any value is accepted.

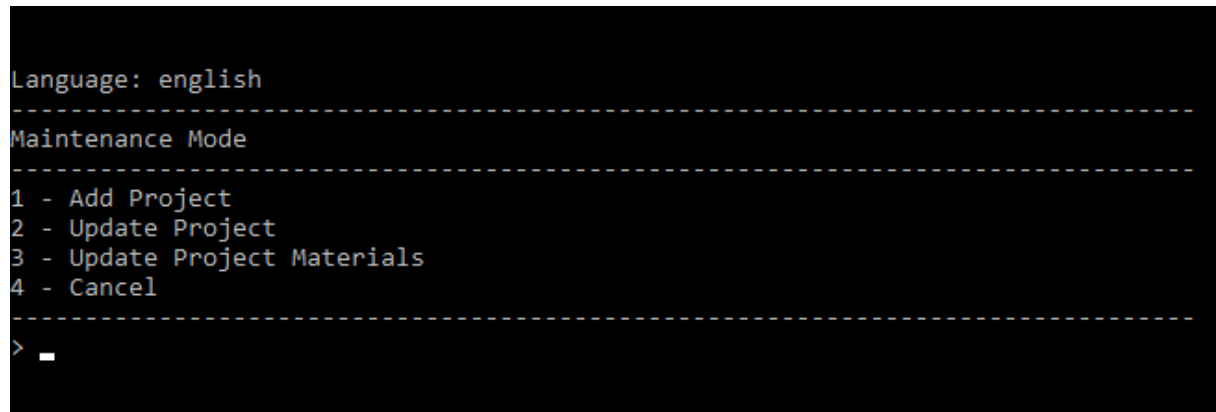
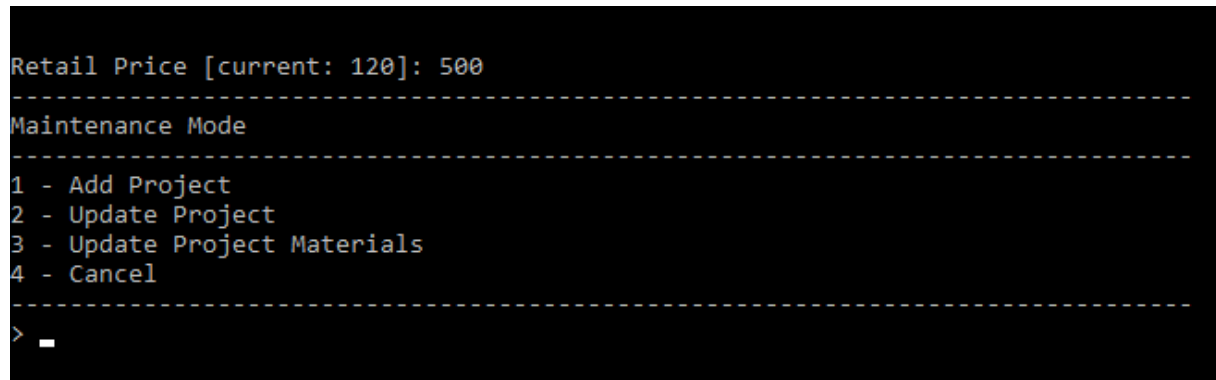
A terminal window with a black background and light blue text. The text shows the current language is 'english', followed by a dashed line separator. Below the separator is the heading 'Maintenance Mode', followed by another dashed line. A list of four options is displayed: '1 - Add Project', '2 - Update Project', '3 - Update Project Materials', and '4 - Cancel'. This is followed by a third dashed line and a prompt '> ' with a cursor.

Figure 33: Visual representation of the application asking for a new Language

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Retail Price Once you have entered the Edit Retail Price functionality you will be asked to enter the following information:

You need to enter the new Retail Price, this is the new retail price that you want your material to have. Any numeric value is accepted.

A terminal window with a black background and white text. The text shows a prompt 'Retail Price [current: 120]: 500' followed by a dashed line separator. Below the separator is the text 'Maintenance Mode' followed by another dashed line. Underneath is a list of four options: '1 - Add Project', '2 - Update Project', '3 - Update Project Materials', and '4 - Cancel', separated by dashed lines. At the bottom, there is a prompt character '>' followed by a small white cursor bar.

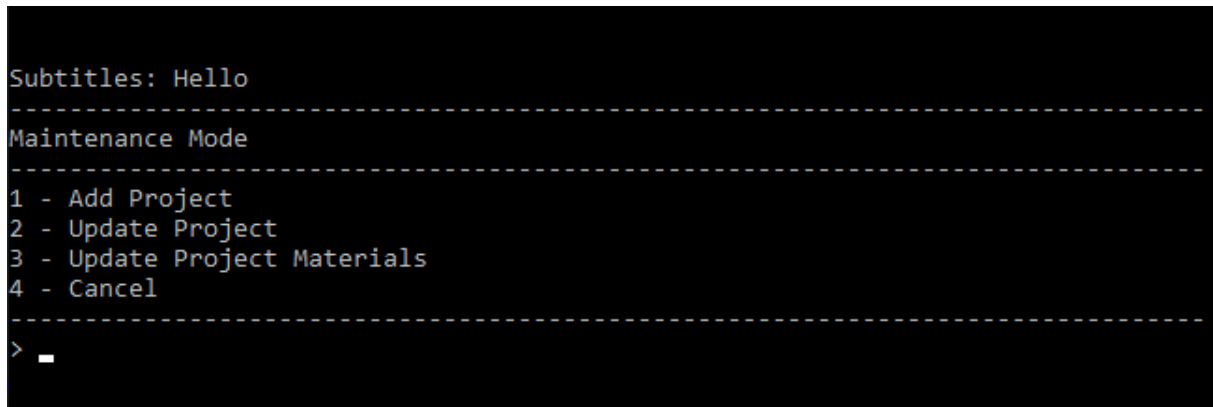
```
Retail Price [current: 120]: 500
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 34: Visual representation of the application asking for a new Retail Price

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Subtitles Once you have entered the Edit Subtitles functionality you will be asked to enter the following information:

You need to enter the new Subtitles, this is the new subtitles that you want your material to have. Any value is accepted.



```
Subtitles: Hello
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 35: Visual representation of the application asking for new Subtitles

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Frame Aspect Once you have entered the Edit Frame Aspect functionality you will be asked to enter the following information:

You need to enter the new Frame Aspect, this is the new frame aspect that you want your material to have. Any value is accepted.

```
Frame Aspect [current: 16:9]: 12:9
-----
Maintenance Mode
-----
1 - Add Project
2 - Update Project
3 - Update Project Materials
4 - Cancel
-----
> _
```

Figure 36: Visual representation of the application asking for a new Frame Aspect

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

Edit Packaging Once you have entered the Edit Packaging functionality you will be asked to enter the following information:

You need to enter the new Packaging details, this is the new packaging information that you want your material to have. Any value is accepted.

Visual representation of the application asking for new Packaging information

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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 Project Materials

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

Firstly, you need to enter 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.

```
-----  
[1]: Title: A Great Movie  
[2]: Title: Deadpool the movie  
[3]: Title: Deadpool 2  
-----  
Project ID [1 - 3]: 3
```

Figure 37: Visual representation of selecting a Project ID to select Material IDs for

Once you have selected a Project ID you will be presented with the associated Material IDs, you need to now select a Material ID from this list and it must be a numeric value.

```
[1]: Format: dvd  
[2]: Format: bluray  
[3]: Format: boxset  
[4]: Format: dsdvd  
Material ID [1 - 4]: 4  
-----  
Main Menu  
-----  
1 - View Mode  
2 - Maintenance Mode  
3 - Quit  
-----  
> _
```

Figure 38: Visual representation of selecting a MaterialID to delete

As you can see in Figure CHANGEFIGUREVALUE, there are more options available 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.

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 distribute 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 across 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 aforementioned 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 39: 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 40: 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 41: 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 42: 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 43: Meeting 5 - Minutes

SDI Strand Group Meeting 6 | MINUTES

Meeting date | time 23rd April | 10:00am | Meeting location Skype

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

AGENDA TOPICS

Time allotted | 7 hours 32 minutes | Agenda topic Quality Assurance |

The group felt it necessary to convene in order to perform the final checks on the completed Application and Documentation.

During this time some bugs were discovered and resolved and the final aspects of the documentation were put together and proofread by group members.

Action items	Person responsible	Deadline
N/A	N/A	N/A

Meeting closed at 5:32 pm.

Figure 44: Meeting 6 - Minutes