

# CSC3600 ICT Professional Project

## File Metadata Harvester and Searcher

### Project Plan

Version No: 7  
Document ID: 1.7  
Aug 16, 2019

**Team Name:** *USQ Learning Emporium*

Team Members	
Name	Student ID
Jesse Hare	W0070949
James MacKeown	U1093772
Ryan Sharp	U1090072
Vincent Roberts	U1046454
Richard Dobson	U1011744

**Supervisor:** *Stijn Dekeyser*

### Version History

Version	Change	Date
1	Created Doc Template	2/08/2019
2	Added Content to Introduction	6/08/2019
3	Added content to the Specification	9/08/2019
4	Added more content to parts 1 and 2	12/08/2019
5	Addition to Risk management plan	13/08/2019
6	Added Content to Code of Conduct	16/08/2019
7	Added rest of content	16/08/2019

## Table of Contents

1. Introduction.....	3
2. Project Specification.....	5
3. Project Description.....	9
4. Work Breakdown Structure and Task Scheduling.....	11
5. Time and Cost Estimations.....	12
6. Project Schedule.....	15
7. Risk Management Plan.....	16
8. Code of Conduct.....	18
9. Conclusions.....	24
10. References.....	24

## 1. Introduction

### The Project

The purpose of this project our team will be undertaking is to provide a tool for our client to harvest metadata for files, and another tool that will use that harvested data and display it in a user-friendly GUI.

The two components of this project are:

1. A script that is invoked via the CLI (command line interface) that takes a directory path as input and returns (on Standard Output) CSV-formatted text having one line per file in the directory and listing all metadata attributes and values extracted from that file.
- 2.
3. Develop a GUI that reads the output of the CLI script (CSV file) and display a spreadsheet-like grid, that can be interacted with. Examples of interactions include sorting by specified attributes, searching for file information on multiple criteria and double clicking a file opens it.

The project is to be implemented in Python, and the desired platform that the components of the project will be implemented to operate on is Linux, although depending on time constraints we may expand support to other operating systems. (Such as Windows or MacOS).

The result of this project will be a pair of tools that will fulfil all our client's needs/requirements. Due to time constraints we will need to be careful that our project remains within a well-defined scope, which we will do so by carefully understanding what it is our client wants from the product vs what will be achievable in the timeline we have been presented with.

### Motivation

Following meetings with our client, we have come to understand their motivation behind requesting our team for this project.

Our client has a very large number of files, and they wish to have a utility to help them organise and keep track of those files. The product we will be developing for them will provide them useful file-management utility and ultimately will help provide a solution to their file management problems.

Our client wishes to be able to harvest file metadata and then be able to interact with that data. The data will be represented in a GUI, which will list all files and their attributes in a spreadsheet-like manner. The client wants to be able to perform searching, sorting and file access operations on the file records represented in the GUI.

Our client will also be able to efficiently harvest the metadata of files and display them in an interactive GUI that allow further sorting and searching by specific keywords/attributes, with each represented file having the ability to be opened from that same interface. This will offer another level of utility for our client that is unprecedented in most standard Linux distributions.

We also believe that for any operating system, that file-management is an important task that needs to be performed regularly to ensure that the filesystem remains organised and easy to navigate. We believe that this project will result in a delivered product that helps our client perform this duty and will hold onto its value for some time.

## Ultimate Goal

The suite of functionality requested by our client closely resembles those offered by several existing file-managing products. The difference between our proposed product and existing software is that the tools we are developing will allow our client to view metadata that is usually not displayed by typical file manager software, or readily accessible. This will allow our client to view file information in more detail, and to search for file information in more specific parts of the filesystem, and against more specific criteria.

We expect this tool to be invaluable in searching for specific files, whose location may no longer be known, and perhaps even the filename itself may have been forgotten. Misplaced files or files that are difficult to find should be found easily enough if the general location of the file is at least known.

But most importantly of all, our goal is to satisfy all user requirements and expectations. We hope they find our software solution as useful as we believe it is going to be, and it serves them well for some time into the future.

## General Plan

We will be using an Agile approach to how we are managing the project, utilising the Scrum framework to ensure that our team work efficiently and learns together to produce the final product. The general plan of how we will run the project is below.

1. Planning
  - Establish the general objectives of the project
  - Design the software architecture
2. Sprint Cycle (Repeat until project closure)
  1. Assessment
    - Review objectives
    - Close involvement of client to introduce requirements/tasks
    - Assign priorities and risk to product backlog/task list
  2. Selection
    - Team and customer work together to select the features and functionality to develop during the sprint
  3. Development
    - Develop the selected features and functionality agreed upon by the team and the client
  4. Review
    - Review the work done and present to the client
    - If product incomplete, begin a new Sprint cycle
    - If product is complete, and client is satisfied, proceed to next phase
3. Project closure
  - Wrap up the project
  - Complete documentation
  - Reflect on learning experiences during the project

## 2. Project Specification

### Component 2: File Metadata Searcher

The metadata searcher will make use of a GUI and will enable the user to view the metadata for each file in the CSV file, which will be organised in a spreadsheet-like grid. There will be an ability to sort and filter the data by attribute values, and a search function which can take multiple criteria. The spreadsheet will have a degree of interactivity, in that double clicking on a record for a file will open it.

#### *Metadata Searcher Technical Details*

##### Language

The Metadata Searcher will be implemented using python, and for the GUI we will be making use of packages such as Tkinter, PyQt ,Pyside etc. Our final choice on which one to use will come down to whichever package has all the required functionality we need.

##### Tools

- Computer running a Linux-based OS
- A text editor such as Atom, Vim, Sublime Text etc
- Latest version of Python will be used, for which an interpreter usually comes preinstalled on most popular Linux distributions

Most members of the team will most likely be using an IDE of some sorts, like PyCharm, Spyder etc.

##### Storage

It can be expected that the metadata searcher GUI program will have a larger storage footprint than the metadata harvester. That being said, it is expected that the metadata searcher program should not take up much storage space at all on any machine it is used on.

##### Software components

- Linux-based OS will be used for development and testing of the harvesting script and all required tools for development. It is most importantly required to run the script, as it will be designed to work with the Linux filesystems specifically.
- Latest version of Python will be used, for which an interpreter usually comes preinstalled on most popular Linux distributions
- At the minimum, we will be designing the searcher program to be invoked via the CLI. There is also an option of making it a standalone executable, which won't require much work, which will enable it to be executable like any other regular standalone program.

##### System requirements

- The system requirements for both the development and execution of this script are not very high, and it can be realistic to assume that any machine with the required OS and tools can do the job. However, it is worth noting that when harvesting large areas of the filesystem for metadata, that users may experience faster execution on machines with higher specs.

### Problems (Scope, Limitations, Assumptions)

During our initial meetings with our client, together we have come to define a scope for the project. Although this may be subject to change during the project, this will be our starting point to help outline the product we are making, and what we will and will not be implementing.

#### *Metadata Harvester Scope*

##### Scope Description:

###### In Scope:

- The script when executed will produce a CSV-formatted text as output detailing all files and their metadata within the user specified directory
- There will be an option available to specify a recursive directory search- which will enable the harvester script to harvest metadata from subdirectories of the target directory as well
- The script will only be accessible to invoke via the CLI
- The script will harvest the metadata of all file types found, and will be able to handle differing file types when formatting the CSV output
- Will be implemented to work with Linux OS distributions

###### Out of Scope:

- The script will not include options to produce output in any other formats
- The script will not be accessible via a GUI
- The script will not be designed to work with filesystems on other Operating Systems besides distributions of Linux

##### *Metadata Harvester Potential Problems:*

- There are many different file types, each with varying sets of attributes, some of which are not mandatory. This needs to be accounted for when formatting the output of the metadata harvester.
- Even files that are the same type, may have differing metadata attributes that have a value. An example of this is there may not be an author value for some text files, whereas there may be for others. This also needs to be taken into consideration.
- Due to time constraints on the project, a desired format for the output has to be agreed on as early as possible. Once a standard CSV format for output is decided then the team can also begin development on the Metadata Searcher, which will help avoid an unnecessary delay to development of the second component of this project.
- Should certain types of files be ignored? Examples of files that may be ignored are temporary files, this will need further clarification with our client.
- Should users only be able to harvest data for files of which they are the owners of? How will this be handled? What permissions will this script be executing under?

##### *Metadata Harvester Limitations:*

- Must be invoked via CLI
- Must only use open-source libraries
- Standard output must be CSV formatted to allow use of output by searcher program
- Must be implemented in Python
- Must be fully developed before or at the same time as the metadata searcher program to help facilitate the latter stages of testing.

*Metadata Harvester Assumptions:*

- Users should know how to use the CLI
- Users of the harvesting script should have authorised access to a Linux OS from which they can harvest file metadata
- Although the Harvester script doesn't require a windowed environment, it's main utility for the client comes from its pairing with the second component of this program, the metadata searcher, which requires a GUI and in turn a windowed environment.
- Users should not use the script for nefarious purposes and will only use them to harvest and search metadata for files that they have permission to access.

*Metadata Searcher Scope**Scope Description:***In Scope:**

- Searcher program will take as input the data produced from the Harvester Script only
- The Searcher program will be designed to allow user interaction via a GUI
- Files and their metadata will be displayed as records in a spreadsheet-like format
- Differing file types will be accounted for, resulting in sparsely populated fields
- File data displayed will be sortable by any metadata attribute (field)
- A search option will be implemented to enable searching by one or more attribute values, which results in only matching records being displayed
- Each file metadata record displayed will be clickable, which will then open the file with its default application for the file's type (via a double click)
- Like the Harvester script, the Searcher must be usable on Linux.

**Out of Scope:**

- The Searcher Program will not be designed to take input from sources other than the Harvester Script
- The Searcher program will not be designed to offer interfaces other than the GUI
- There will not be an option to change the format of data displayed
- Files will only be able to be opened via double clicking their associated record, no other file operations will be possible. (Such as deletion, renaming etc.)
- The Searcher program will not be implemented to run on other OSes other than Linux distros.

*Metadata Searcher Potential Problems:*

- The harvested data used by the Searcher will have some inconsistencies, such as empty metadata fields, which will need to be accounted for
- How will the files and metadata displayed be organised by default?
- What will be the default order for file metadata attributes displayed?
- If files are sorted by a field how will others be sorted that do not have a value for that field?
- Will displayed files and metadata be organised by file type in separate sections? Or will they be displayed all together, sorted depending on another metadata attribute. (such as filename)
- The format of the CSV input that the searcher requires must have a format that is agreed upon as early as possible, as without this development of the searcher program will be delayed.
- Will the program be invoked via CLI or a standalone Launcher? Or both?

#### *Metadata Searcher Limitations:*

- Must make use of a GUI
- Must only use open-source libraries
- Must be implemented in Python

#### *Metadata Searcher Assumptions:*

- Users should know how to use the CLI
- Users should have access to a Linux system that is not headless in order to make use of the Searcher program and its GUI
- User will not use the programs for nefarious purposes and will only use them to harvest and search metadata for files that they have permission to access.
- Users will only attempt to use the Searcher with CSV data produced by the companion Metadata Harvester script developed by our team.

#### Potential problems for project in general

- Changing Requirements – There is a high chance that requirements may change due to a range of factors during development. To remedy this, and minimise the impact, we will be using incremental development, so that any work that must be redone will only be limited to what was made in the previous sprint cycle.
- Team mismanagement/miscommunication – Because our development team are all based in different geographical areas, all our correspondence will be done via distance. Because each member is working by themselves, we will need to make sure that the project is well planned and laid out, and that meetings are scheduled regularly enough so that all members stay well informed on the progress of the project.

#### Objectives

- Implement all functionality requested for by the client
- Make all deliverables on time and within budget
- Complete unit testing of all software components throughout development
- Complete program documentation before project closure

#### Expected Outcome and deliverables

The expected outcome of this project is to provide both components with the full functionality desired by the client.

The specific deliverables will be:

1. The file metadata harvester program
2. The file metadata searcher program
3. Supporting documentation for both components

It is worth noting however that these deliverables will be broken down during development into smaller deliverables that will be presented to our client at the end of each sprint cycle.



## 3 Project Design

### Functional requirements

Functional requirements are activities that our system must perform. Specifically, our system traverses a file system either recursively or non-recursively to find files and folders, and subsequently extract multiple types of file metadata, dependent on the file type. This metadata is formatted in a specific order and output into a CSV format, ready for importation into a graphical user interface for data manipulation at the users' request.

The following functional requirements are split into two parts; one for back end components which take care of metadata scraping and another for the graphical user interface:

#### Back End

- Command line interface script written in Python
- Takes three arguments:
  - 1) (-r) recursively traverse through the folders of the specified path
  - 2) (/path/to/file/or/folder/) relative path of file or folder to traverse
  - 3) (out.csv) output file separated by > character
- Utilises the os.walk member functions' functionality from the core Python library to walk each layer of the selected file path for each file/folder.
- Each file is individually analysed for any metadata, beginning with name and type as core requirements, with subsequent metadata entries separated by commas.
- Each files metadata is passed in a while loop until all files are complete, output is sent through STDOUT to a formatted CSV whereby the information can be piped to the graphical user interface.

#### Front End

- Command line launcher for utilisation with Linux pipe ( | ) and back end in one line of code.
- Takes one argument, the input relative file path.
- The input CSV information is displayed in a spreadsheet-like grid format initially
- Information can be further filtered by rows and columns
- Information can be searched by either filename or metadata type
- Double clicking a file-row will open the file in the operating systems default file browser

### Non-functional requirements

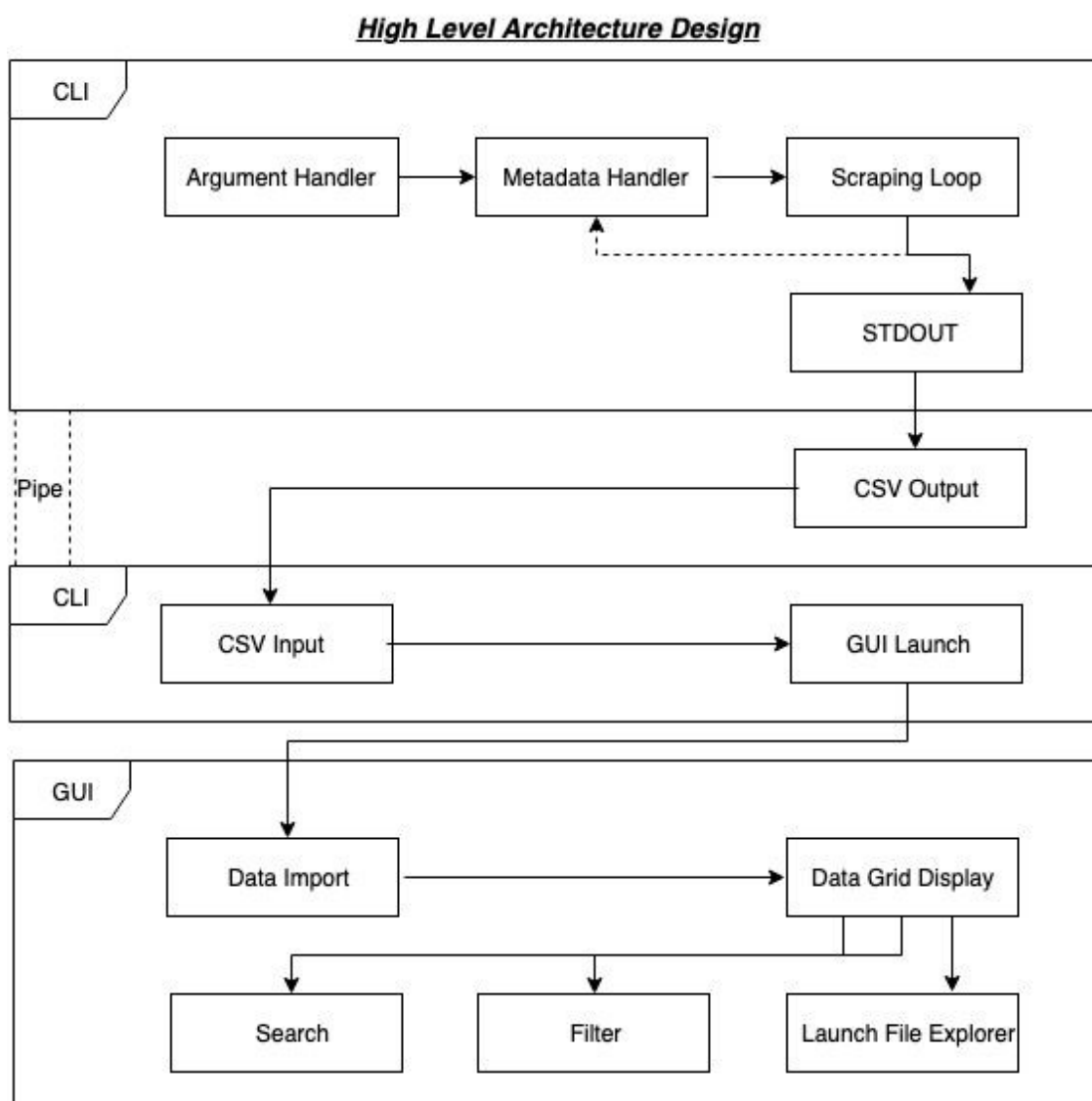
A non-functional requirement describes how the system should behave, in contrast to the functional requirement which describes what the system does.

Our system functionally depends on the information gathered by the back-end component, without this initial information gathering phase the following information manipulation through the graphical user interface is not possible.

This area covers more of what the client "wishes" than the technical details of each phase.

- Performance – Fast and consistently reproducible results
- Scalability – Potentially multi-threaded for large workloads
- Reliability – Free from bugs that prevent consistent workflow
- Data Integrity – The data pulled from the program is accurate and reliable
- Security – Coding has followed best practices to prevent memory leaks and potential buffer/heap overflows
- Environmental – Works flawlessly on the clients preferred Operating System, Debian Stretch, but could be multi-platform if possible
- Interoperability – The programs should work in tandem with each other and have inputs/outputs which are pipeable

### High Level Architecture Diagram



## 4. Work Breakdown Structure and Task Scheduling

### i. Project planning

1. Develop the work breakdown structure and build a project schedule ~ 4 hours

### ii. Project analysis / Requirements elicitation

1. Video conference with client to understand project requirements ~ 2 hours
2. Refine gathered requirements from client and discuss with development team ~ 1 hour
3. Model user tasks ~ 1 hour

### iii. Project design and specifics

1. Allocate two teams, one for Python based CLI another for TKinter/PyQT GUI ~1 hour
2. Allocate scrum master and setup sprints for each team ~ 4 hours

#### Front end team:

1. Design screen layouts for front end GUI ~ 1 hour
2. Finalize frameworks for multiplatform utilization ~ 1 hour
3. Identify main methods, classes and libraries for core functionality ~ 1 hour

#### Back end team:

1. Create pseudocode skeleton of program to assist in visualisation of workflow ~ 2 hours
2. Determine open source GPL libraries for required functionality, including metadata scraping and recursive folder traversal ~ 1 hour
3. Correspond with front end team on format of output CSV to aid in parallel programming ~ 1 hour

### iv. Project build

#### Front end team:

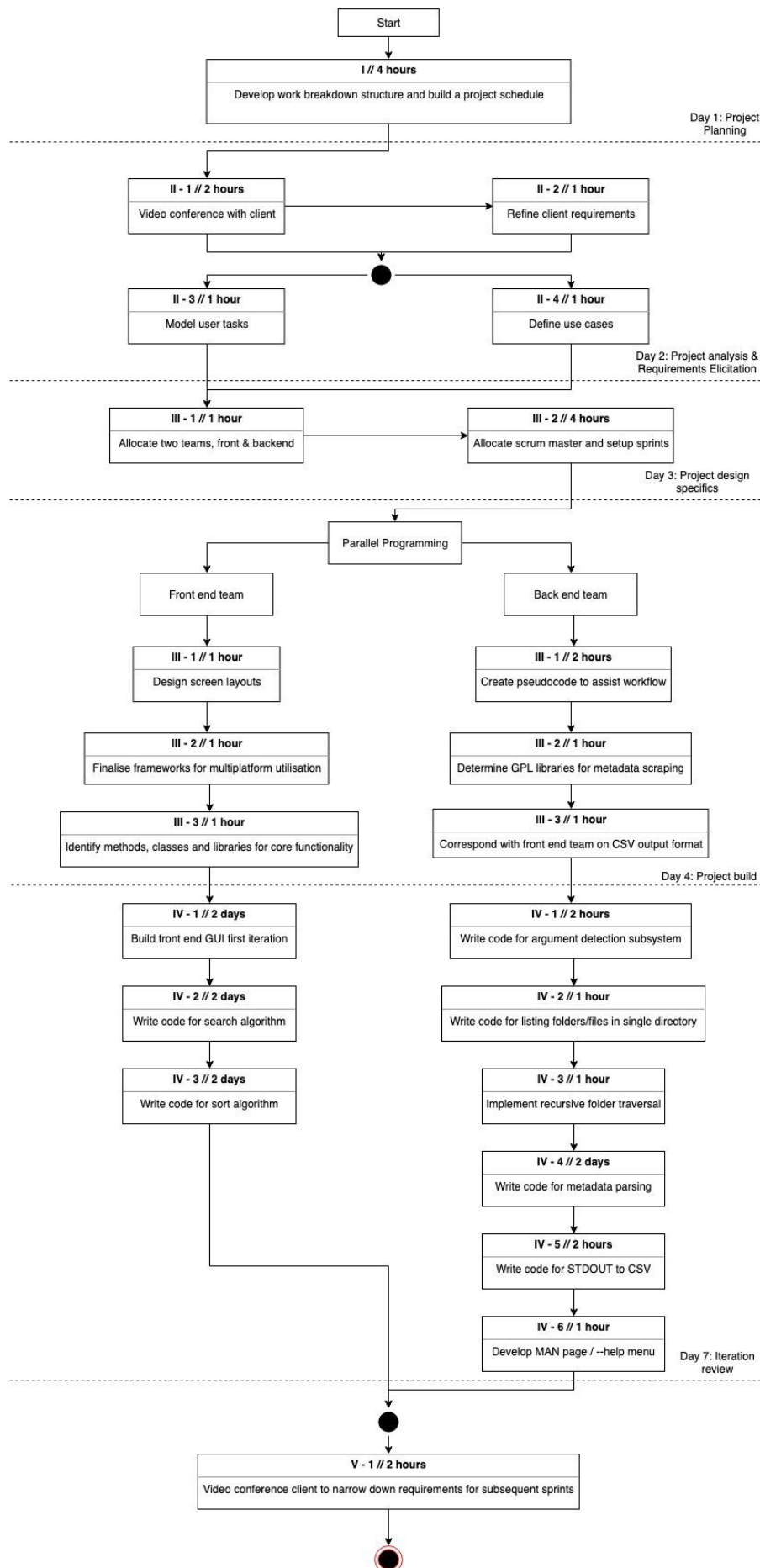
1. Build front end GUI first iteration ~ 2 days
2. Write code for search algorithm ~ 2 days
3. Write code for sorting CSV imported elements based on user input ~ 2 days

#### Back end team:

1. Write code for argument detection subsystem ~ 2 hours
2. Write code for listing folders/files within singular directory ~ 1 hour
3. Implement recursive functionality to file traversal ~ 1 hour
4. Write code for metadata parsing ~ 2 days
5. Write code for STDOUT to CSV file ~ 2 hours
6. Develop a MAN page or –help menu ~ 1 hour

### v. Iteration review

1. Video conference with client to further narrow down requirements for subsequent sprints ~ 2 hours



## 5. Time and Cost Estimation

### Cost of resources

The following table will outline the costs associated with the project. All items below are subject to change in the event of any unforeseen events.

#### Hardware Solutions

Item Required	# of Items Needed	Cost (\$AUD)	Total Cost (\$AUD)
Microsoft Surface Pro 6	5	\$1,349	\$6,745
MacBook Air 13" (128GB)	1	\$1,699	\$1,699

As the project will be worked on remotely, all staff will be required to use company provided equipment. There are 5 staff members each requiring a laptop. A Microsoft machine has been chosen as the main laptop to be worked on. VirtualBox will need to be installed in order to work with Linux distributions. Alternatively, these laptops can be configured to run in dual-boot mode.

An additional MacBook Air will be sent to the project leader, James McKeown. This is to ensure the program developed will be able to run on all major platforms. A bottom of the range MacBook has been suggested to reduce costs.

#### Software Solutions

Item Required	# of Accounts Needed	Cost (\$AUD)
Zoom Video Conferencing	5	\$0
VirtualBox	5	\$0
Slack	5	\$0
Atom	5	\$0

As stated above, the project will be worked on remotely. As such, Zoom video conferencing will need to be used in order to allow for efficient communication between staff members. This will allow for screen sharing of devices which will boost productivity. Slack will be used to facilitate group messaging between members. With it's ability to have different chat rooms, Slack is the clear choice.

VirtualBox will be installed on all Windows machines to allow members to use and test different Linux distributions. Atom will be the text editor used for the project.

## Time estimation

The project will take approximately 10 weeks to complete. In these 10 weeks it is estimated there will be five sprints. Sprint 1 will take one week, sprint 2 will take three weeks, sprint 3 will take three weeks, sprint 4 will take three weeks, sprint 5 will take one week.

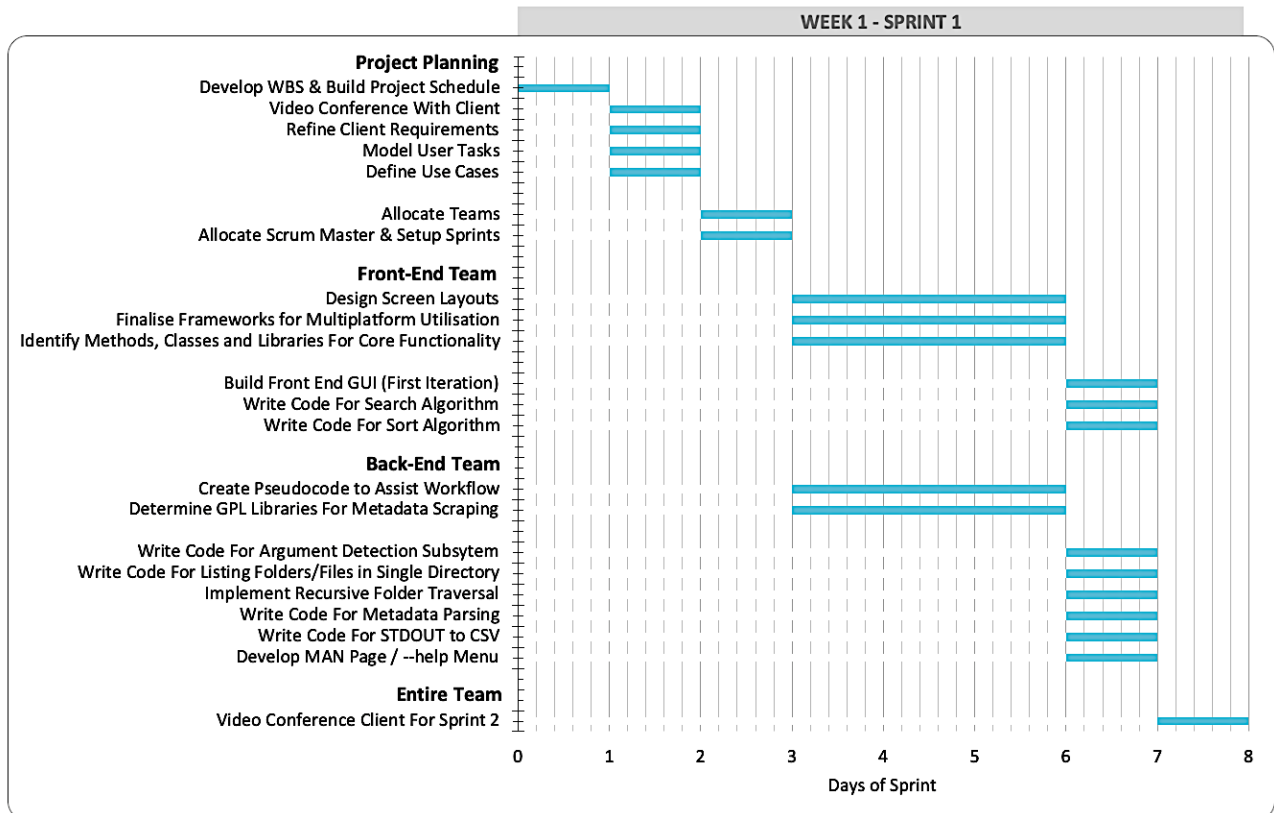
## Labour Cost

The project will be split up into 2 teams, a front-end team and a back-end team. The front-end team will consist of Jesse Hare and Ryan Sharp. The back-end team will consist of James McKeown, Richard Dobson and Vincent Roberts. The team leader will be James McKeown. The project will run for approximately 10 weeks, in which each staff member is expected to take on a part-time role. The following table is an estimation of the labour cost based on current industry standards (based on [payscale.com](https://www.payscale.com) and [indeed.com](https://www.indeed.com)).

Staff Member	Role	Hourly Rate	Hours Per Week	Estimated Total Cost
James McKeown	Team Leader/Back-End Developer	\$39.90	20	\$7,980
Jesse Hare	Front-End Developer	\$34.10	20	\$6,820
Richard Dobson	Back-End Developer	\$35.40	20	\$7,080
Ryan Sharp	Front-End Developer	\$34.10	20	\$6,820
Vincent Roberts	Back-End Developer	\$35.40	20	\$7,080

## 6. Project Schedule

### GANTT Chart



### About the Planning

As the project is being run under the agile methodology, planning past the first sprint is infeasible. As such, only the first sprint has been planned to use the GANTT chart. The first three days of the project will be spent planning and organising teams. Once teams are organised into front-end and back-end, each team will have their own progression through the project. Main tasks will be developed from day three to day seven. On day 8, a meeting with the client will be held to figure out a plan for the second sprint.

Milestones will be hit at the end of each sprint stage.

## 7. Risk Management Plan

Risk Management involves four key stages:

- Identification
- Evaluation
- Mitigation
- Action Plan

### 7.1 Identification

There are two main categories of potential risks to this project that we can foresee at this stage:

- Operational - risks to the completion of the project; and,
- Strategic - risks to reputation, professionalism and professional ethics.

The following specific risks have been identified in each of these categories:

Operational:

- Project delay. This is risk of the team falling behind in key milestones and failing to meet deadlines.
- Loss of a team member. The risk that a team member may be lost due to illness or to dropping out of the course, or otherwise incapacitated for the purposes of this project.
- Loss of equipment. Refers to the possibility that equipment being used for this project such as computers, laptops, home networks, internet connection, mobile phones, etc may be lost, stolen, or cease to function due to breakdown, therefore reducing the productive capacity of the so afflicted team members until such time as the equipment may be repaired or replaced.
- Loss of data. Refers to the loss of work that has already been completed due to failure to keep backups or other such accidents.
- Loss of necessary services. This is when a service that is being used by the team to conduct operational activities becomes unavailable, resulting in reduced cooperative productive capacity until such time as the outage can be restored or a workaround can be found.
- Unable to implement a working solution. This refers to the risk that the team will ultimately be unsuccessful in its endeavours to create a software solution suitable to the client's needs.

Strategic:

- Leak of customer's privacy. This is the risk that activities by the project team may cause undesirable exposure of the client's private details that they would have wished to have remained out of the public sphere.
- Leak of client's business confidentiality. Similarly, to the point above, this risk refers to a situation in which the client's classified business documents are exposed, whether inadvertently or maliciously, to unwanted public attention.
- Leak of intellectual property. This refers to the loss of valuable intellectual property belonging to the client, due to carelessness or corporate espionage, etc.



## 7.2 Evaluation

The two tables below enumerate each of the specific risks that have been identified above and assigns a probability score as a percentage of this risk eventuating over the course of the project. It also assigns an impact score as a percentage to assess how detrimental the outcome would be if that risk does indeed eventuate. Consideration of these two aforementioned factors combined leads to the computation of the “Status” of that risk, which is evaluated as “high”, “neutral”, or “low”.

**Table 1. Operational - Risks to the Completion of the Project**

ID	Risk and Consequences	Probability (%)	Impact (%)	Status	Mitigation Response
1.1	Project delay	70	90	High	If project begins to fall behind, tempo of team Zoom meetings to increase from weekly to bi-weekly, and if necessary, tri-weekly.
1.2	Loss of team member	25	95	Neutral	If team suffers loss of team member, client will be re-engaged to have their expectations massaged in accordance with new reality.
1.3	Loss of equipment	40	20	Low	It is not outside the realms of possibility that one of our team will have their laptop or phone die, but that's no worries, they can easily replace.
1.4	Unable to Implement a working solution	30	100	High	If either our front-end or back-end teams are unable to implement a working solution, human resources from the other team will be reassigned to the team which is struggling. If both teams are struggling, The technical advisor (Professor Dekeyser) has agreed to provide the team with relevant hints to unblock progress and assist with implementing a working solution
1.5	Loss of data	60	50	Neutral	If we lose data due to inexperience with GitHub or for another reason, we just have to do it again, and take proper backups. The team has already experienced on close call due to unfamiliarity with GitHub, therefore are keeping backups in a google drive now as well.
1.6	Loss of necessary services	10	70	Neutral	Slack and GitHub are enterprise grade systems which are unlikely to go down. USQ study desk however is a different story...

**Table 3. Strategic - Risks to Professionalism and Professional Ethics**

ID	Risk and Consequences	Probability (%)	Impact (%)	Status	Mitigation Response
2.1	Leak of customer's privacy	10	80	Neutral	Loose lips sink ships. If any team member inadvertently leaks private customer details, they will be reminded of their importance of proper professional conduct.
2.2	Leak of client's business confidentiality	10	75	Neutral	If any team member inadvertently leaks private customer details, they will be reminded of their importance of proper professional conduct.
2.3	Leak of intellectual property like system design	10	90	Neutral	If any team member inadvertently leaks private customer details, they will be reminded of their importance of proper professional conduct.

### 7.3 Mitigation

The four T's of risk mitigation are:

- Tolerate – accept the risk
- Treat – take cost effective actions to reduce the risk
- Transfer – shift the risk to a 3<sup>rd</sup> party such as a sub-contractor or an insurance company
- Terminate – eliminate the risk entirely or kill the project.

The specific risks assessed as having the highest status are project delay and inability to implement a working solution. All other risks are assessed as being Tolerable, however these two specific risks will need constant re-evaluation to see if they need to be Treated. If they do, the Treatment for these risks is outlined below.

### 7.4 Action Plan

The action plan is outlined below as a response to the two major specific risks as assessed, which are:

- Project delay; and
- Unable to implement a working solution

In the event that the project is beginning to be delayed, which is being defined as a failed sprint, i.e. a one-week sprint which fails to achieve any of its prescribed goals, then the team will intensify the frequency of communication by increasing the number of scheduled zoom calls from once per week to twice per week. Furthermore, if it looks like the team is starting to be at risk of being unable to implement a working solution, which is defined as 2 failed sprints in a row, i.e. two one-week sprints in succession which fail to bear any productive fruits, then the team will set a time an emergency zoom meeting with Professor Dekeyser to demonstrate what the holdup is and ask for any hints or advices which might be forthcoming.

## 8 Code of Conduct

### Team Principles

Team principles outline the overall guidelines to be followed by all team members in order to create the most effective possible environment for the success of the project. These are ideals that all members should keep in mind and work toward implementing throughout their conduct as part of this project.

- The project will be conducted in a manner that promotes transparency in the development of the project, its status, methods, and future. Team members will also be honest in their conduct as well as what they say to other team members and project stakeholders. This will serve to ensure that all parties are accountable for the work that they do on the project, that agreements have been followed, and that the team's conduct during the duration of the project has been consistent with high professional and ethical standards. This will also help to ensure that fixable issues are resolved where possible before they risk becoming bigger problems further down the line.
- Team members will strive to promote open communication throughout the project's run, and to foster an environment of successful collaboration in service of meeting the ultimate goals of the project. This is to ensure that the team works as a team as well as possible, that input, knowledge and experience from all of the members as well as stakeholders is used to the fullest extent, and to allow all team members to collaborate fully in the best possible team environment.
- Team members will endeavour to be respectful of both each other and anyone else involved in the project in their conduct. An environment of mutual respect is more likely to be an environment where all team members are able to contribute to their fullest, and that nobody is negatively affected to a degree where their work may suffer when it comes to project development.
- Team members will take care to ensure all relevant laws and regulations to the project and its completion are adhered to. Needless to say, not adhering to relevant legislation or regulations will only serve to potentially jeopardise the project's future, as well as potentially having other serious unforeseen consequences.
- Team members will generally attempt to conduct themselves in an ethically consistent manner. Consistent ethical conduct will ensure that the project is completed in a way that does not risk violating the trust of stakeholders, confidentiality of information, or disruption of the team.

### Team Communication and Operational Processes

These processes are guidelines which team members are to follow in order to facilitate the appropriate kinds and degrees of communication between each other so as to keep everyone informed and on the same level throughout the project's duration, as well as to promote collaboration, and ensure smooth overall operation of the project by the team.

- Any project issues that would affect the rest of the team or the status/development of the project, are to be brought up as soon as possible to the rest of the team. This is in order to ensure that problems are found at an early stage, and can thus be more easily dealt with, as well as to allow for the most time to facilitate changes or solutions that may need to be implemented.
- Decisions that affect the development of the project are to be taken only after consultation with the other group members, as well as if necessary, the client and other stakeholders.
- Throughout the duration of the project, both regular team meetings with the client as well as regular internal team meetings are to be held in order to make sure both the client and team members are on the same page and are up to date with the current status of the project. Internal team meetings are to additionally be used to organise what members are to be working on what tasks and generally allocate work, but also to ensure that all members are well informed on the status of other elements of the project and the overall progress.
- Throughout this project, the team will ensure that proper version control and management are applied, with the use of other tools and software such as git and GitHub to ensure that collaboration is effective and version contributions and updates are kept well documented.

### Professionalism and Professional Ethics

These serve to ensure that all team members keep their interactions with other team members as well as the client and stakeholders as professional as possible. Also, to ensure that general conduct when developing the project is also kept professional, and that development is done in the most ethical way practical.

- All information provided by the client or stakeholders to the project team, as well as the contents of all discussions shall be kept confidential and used only for the purposes of completing the project.
- All laws and regulations regarding Intellectual Property rights and ownership are to be followed and the general principles of the protection of Intellectual Property are to be respected throughout the development of project deliverables.
- The project team is to display respect in its interactions with, and treatment of any stakeholders or other persons related to the project, its development, and its completion.
- The client and other stakeholders are also to be treated as a valuable part of the project, with their feedback and input to be considered where relevant, and to be kept informed and consulted where appropriate to ensure the outcome that is best for all involved parties.
- Team members are to treat each other with respect in their interactions and collaboration throughout this project. This also means that team members are to consider their peers in regard to effort applied to the project, and to treat the other members with respect by ensuring that they themselves are contributing an appropriate amount to the project and not letting the work fall disproportionately on other members.
- All appropriate methods for the protection and preservation of important data and its integrity are to be undertaken, both to ensure protection of this data from external interference or attack, as well as to protect it from loss due to technical faults.
- Where possible only the use of legitimate versions of licensed software are to be used in the completion of this project.
- Additionally, to earlier mentioned respect of the protection of Intellectual Property, any assets, audio/visual resources, or other resources are to only be used when allowed by their rights holders and under the appropriate licensing or other usage agreements.
- If or when assets or resources created by an external party are used, they shall also be fully acknowledged, and the original rights holders fully credited where appropriate.

### Non-Compliance

Non-compliance from team members broadly includes any form of conduct that goes against this Code of Conduct, and in any way damages the progress or development of the project, interferes with the outcome, disrupts interactions with the client and stakeholders, or disrupts other team members. These forms of non-compliance can be broken down into major and minor non-compliance depending on the nature and severity of the conduct and its effect on the project.

### Minor Non-Compliance

Minor non-compliance covers actions that cause only minor amounts of disruptions in the overall outcome and success of the project, and do not seriously jeopardise it, these include but are not limited to:

- Team members either not doing all of the work they are assigned, or unannounced delays especially if such delays are happening repeatedly.
- Team members being resistant to or refusing to communicate to a degree that information about the project is not being shared adequately such that delays, or errors are occurring in development.
- Team members being disrespectful to each other, or otherwise creating an environment where other's ability to contribute or work are being negatively impacted by a member's behaviour.
- Team members deviating from agreed upon plans unilaterally, without communicating this to other team members or considering their input in potential changes.

### Major Non-Compliance

Major non-compliance covers forms of non-compliance that have a more major effect on the project, severely disrupting work on development, the relationship with the client, or overall project outcomes, including potentially jeopardising the project entirely. Major non-compliance includes, but is not limited to:

- Team members failing to participate at all in the project, with an extreme degree of non-participation that causes major delays or elements of the project to need to be completed by other team members instead and doing so consistently.
- Intentional disruption of work material or resources, such as damage to or destruction of data, equipment, documentation, or other materials used in the completion of the project.
- Improper handling of work material such as data, documentation, or equipment that, through a team member's inaction or negligence, said material is lost or damaged.
- Team members disrespecting or creating conflict with either the client, or other stakeholders in any form.
- Team members misusing external assets used in the project, such as plagiarised or stolen intellectual property, or fraudulent licensing and/or usage agreements.
- Gross ethical violations in the conduct of any team members through the course of the project, especially in regards to breaches of team, project, and client confidentiality and data security.
- Breaches of any legislation or relevant regulations by team members at any point during the duration of the project.

## Non-Compliance Resolution

If breaches as outlined in the previous section are found to have happened, there are a number of solutions that can be taken in order to attempt to remedy the situation, these will depend on the nature of the non-compliance, the severity, and the attitude and behaviour of the conflicting party. In cases of smaller non-compliance, especially if not malicious, solutions may be simple and easy to work through as a team, though in the case of major non-compliance especially if done so maliciously, more drastic action may need to be taken.

### Minor Non-Compliance Resolution

This section covers only minor issues of non-compliance with the Code of Conduct, such that the project is not significantly affected or jeopardised, but that may be causing delays or interruptions in workflow. These solutions may include but are not limited to:

- Members who have minor non-compliance issues may be asked to have private discussions with other team members or the team leader in order to communicate the problems that are occurring, and to attempt to work out a solution to these problems.
- If the non-compliance is affecting the team more widely, a team meeting may be held in order to discuss the issues with the non-complying team member, and as an entire team to attempt to work through a solution where possible.
- In the case that it is needed, it may be possible to draw up a small plan in order to manage correction of the non-compliance from a team member, this would be done with the member and other members of the team, and used as a guide for which to manage resolving the problem through the time needed while the project is ongoing.

### Major Non-Compliance Resolution

This covers larger, more serious problems from non-compliance with the Code of Conduct, in these cases the problems cause is more significant and pose a major disruption or obstacle to the project's development and may severely impact the ability of the team to deliver elements or jeopardise the project entirely. Resolutions to this may include but are not limited to:

- Major team meetings, if needed with the client or other stakeholders, in order to develop a plan to resolve the problem in a way that is most satisfying to all parties, as well as to keep all parties informed on the current status of the project and the issues that have arisen.
- In cases of severe non-compliance, especially such that it through malice or gross negligence jeopardises the success of the project, then immediate removal from the team may be decided for the non-complying member.
- Depending on the exact nature of the non-compliance, it may be necessary to launch legal action, such as in the case of major destruction of resources, gross ethical violations, or violations of legislation. This would only be required in the most extreme of circumstances, where no other options are realistically viable solutions, or where the non-compliance is to such a significant degree that it is warranted.

## 9 Conclusions

This concludes the project plan, which we hope helped you understand the project in detail, how it will be carried out and what to expect when it reaches completion. This plan we hope will serve as a solid reference as to what is happening throughout the design, development and deployment of all project deliverables.

Thankyou for reading, and we look forward to commencing work on this project, and the eventual delivery of a high-quality software solution that solves our client's problems and meets all their needs.

## References

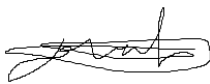

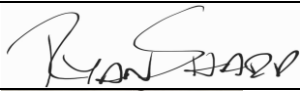
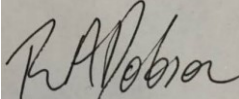

Sommerville, I. and Stevens, P. (2012). *Introduction to software engineering*. Frenchs Forest, N.S.W.: Pearson.

Rosenblatt, H. and Shelly, G. (2014). *Systems analysis and design, tenth edition*

Highsmith, J. (2011). *Agile project management*. Upper Saddle River: Addison-Wesley.

## Contribution Statement

The undersigned members of this team agree to abide by this project plan to ensure the successful completion of the project. The members also agree that the contribution percentages specified below reflect the true level of contributions made by each of the members to the works reported in this document.

Student Name	Student ID	Signature	Contribution %	Date
Jesse Hare	W0070949		20	16/08/2019
James Mackeown	U1093772		20	16/08/2019
Ryan Sharp	U1090072		20	16/08/2019
Richard Dobson	U1011744		20	16/08/2019
Vincent Roberts	U1046454		20	16/08/2019