Windows File Content Search System Displayed Via Tree Diagram

by

Richard Liaw Zhi Cheng

10973

Dissertation submitted in partial fulfillment of

the requirements for the

Bachelor of Technology (Hons)

(Information & Communication Technology)

MAY 2011

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Windows File Content Search System Displayed Via Tree Diagram

by
Richard Liaw Zhi Cheng

A project dissertation submitted to the
Information & Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,

[illegible]

GOH KIM NEE

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
MAY 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

[illegible]

RICHARD LIAW ZHI CHENG

ABSTRACT

The purpose of this project is to first study on the application of the mind mapping technique in a Windows search program. We will first compare between a mind mapping search method to the conventional Windows 'Search' function. This study will prove beneficial because of the rise in the use of computers. As the reliance on computers increases, so will the volume of data and information stored in them. This means that sooner or later, someone has got to develop a more intelligent or effective search method. As for the scope of the system, a new search program that also provides the basic functionalities of receiving users' input and displaying search results is to be developed. The initial design of a mind mapping search program is defined first. The directory specified by users will be the root of the mind map and its immediate branches will be the directory's highest level contents. The branches will subsequently branch out to display the contents of subdirectories in the specified directory. The branching will end by displaying the end files as clickable titles. These titles (the name of the files) will allow users to directly open the files by simply clicking them. However, more study is needed to determine whether a mind mapping search program is a good alternative search program or not. Ultimately, this project is to recommend an alternative file searching program and develop a prototype to demonstrate its capabilities. This proposed alternative search program might or might not incorporate the mind mapping search technique. The following report contains a literature review – a study on the advantages and disadvantages of using the mind map method as well as a comparison between mind mapping and a few other visual representation methods. In the end, a conclusion is drawn on whether the mind mapping technique is suitable to be incorporated into a search program. If not, an alternative search method is to be proposed. The methodology chosen for this project will be the Extreme Programming (XP) technique of Agile Development which will be further discussed. The results of the study which includes a survey, as well as a recommendation and conclusion section will be presented.

Table of Contents

List of Figures

Abbreviations and Nomenclatures

GUI	graphical user interface
KISS	keep it short and simple
XP	Extreme Programming
JRE	Java Runtime Environment

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 Background

In our modern world, the use of computers has become essential in our everyday lives. As the reliance on computers rises, so does the volume of information and data stored in it. The importance of data storage and access facilities or methods is not to be taken for granted. Windows Operating Systems offer users with ways to organize and search for files and folders.

There are three common ways to search for a file or folder stored in our computer or storage devices. However, are these three conventional ways convenient for all occasions? Are they able to provide users with an extensive view of where files and folders are stored or the way they are arranged? Should we explore the possibility of having an alternative way to perform a file search which might be more convenient in certain circumstances? Is there a searching technique that can serve as an alternative method to what is already available to us now? Well, this project is dedicated to finding the answers for the questions above.

1.2 Problem Statement

Firstly, let us look at the three common practices for file search. The most common practice would be the conventional manual search. This is how most users search for their files and folders. They would iteratively open directories and subdirectories one after another, in Window's 'explore' view, until they find what they are looking for. This method makes use of the user's memory of where he or she had stored them. If the user is the one who stored the file and clearly remembers the location, this method might be the fastest way to locate a file. However, obvious disadvantages are present. The first disadvantage would be that this method would be time consuming if the user is not the person who stored the file in the first place or that, he or she actually forgets where it was previously stored. Another disadvantage would be that searching for a file would be frustrating if there are many subdirectories to open. In other words, it is not the best way to search for files where location is unknown.

The second method used to search for files/folders is to type out the directory path of the file location. In Windows, there is always a directory path displayed on the top bar. For example, my photos are stored under '*C:\Users\Richard Liaw\Pictures*'. However, this method requires one to remember exactly the location of the file. Obviously, if the directory path is too long, as in the file is located within levels and levels of subdirectories, this method would be rendered unfeasible. It is difficult enough to recall the location of the intended file, what more to say its whole directory path. Of course, a combination of the first two methods is possible. A search process could involve both typing directory paths and searching through the 'explore' view. However, the problem of long and tedious search still exists.

The third method would be the use of the search function. Most operating systems which are common to users would have a search function which would allow the users to specify the file name or any other criteria like keywords associated with the file or folder name, date last modified, file type or even the date created to search for files. At first glance, this method seems sufficient to locate files and folders. However, as we explore deeper, we would see the limitations of this method. Firstly, the results of the search function are displayed in a list. This means that the users would not be able to clearly see the way files and folders are organized. The only way to see where files are stored would be to check the directory paths one by one. Another problem with this method would be the case when users themselves cannot recall the filename or any other details related to the file which they intend to find. In most cases, users are able to recognize and recall the filename if he or she actually sees the actual filename. This method lacks in the way that it does not provide an extensive and hierarchical view of the location of all files and subdirectories in a certain directory. We can conclude here that although these search methods have their uses, they are not effective for all occasions. I will proceed to introduce an alternative search method that would prove to be useful in certain occasions. Of course, one potential method that we will be discussing heavily would be the mind mapping technique in searching. This new method has the potential to provide an extensive view of all contents of a specified directory graphically in a timely manner. However, more study is needed to determine its real value.

1.3 Objectives

1. To develop a search program that could serve as an alternative or even a complement to the Windows 'Search' program.

2. To develop a system that provides the basic functionalities of a search program and a consistent interface with Windows applications to improve user familiarity.

3. In whole, study on the potential of the mind mapping technique in search programs and determine whether it is best suited as an alternative search program. If not, propose a better alternative.

1.4 Scope of Project

1.4.1 Scope of Study

Firstly, before we can even conclude that the mind mapping technique is a suitable technique to be utilized in this project, a study on the pros and cons of various visual representation methods should be carried out. In the end, a clear conclusion should be drawn on which method is comparatively better than the rest in terms of file or folder search. In our study, we will be looking at a few common and widely used techniques. They are the tag clouds, organization diagrams and mind mapping techniques.

After clearly explaining why the mind mapping method is used in this study, we will go on and compare this method with the conventional Windows 'Search' function. Ultimately, a conclusion will be made if the mind mapping technique is suitable when incorporated into a search program. If not, an alternative search program will be proposed.

1.4.2 Scope of System Functionalities

As previously mentioned, the system must be alternative search program. If a mind mapping search program were to be developed, the system must first provide a user interface to obtain user input, which in this case, is a directory path. The mind map should display the contents of that directory in a hierarchical manner with subdirectories subsequently branching out to display their contents. The branching will end when there are only files and no subdirectories left. The filenames would be displayed as links which users can click to directly open those files. This enables users to directly view the contents of those files. However, if the study shows that mind mapping is not an effective technique in our case, an alternative would be proposed. The main functionalities required here are just the capabilities to receive users' input, be it directory paths or any search criteria, and display the search results. This system will be a standalone application. Efficient programming techniques will be practiced to ease the process of further modifications and improvements to the system in the future. Since the Java programming language will be used to develop the prototype, it would be extremely versatile.

As for the feasibility of this project, a concise schedule would be drawn out and the fulfillment of requirements and scope are plotted into this schedule to ensure that a complete system and study are delivered on time. As mentioned earlier, file and folder searches are frequently carried out so the findings of this study will definitely prove useful to future studies and researches. Besides that, this study could be used as a reference for the future development of intelligent search engines.

CHAPTER 2

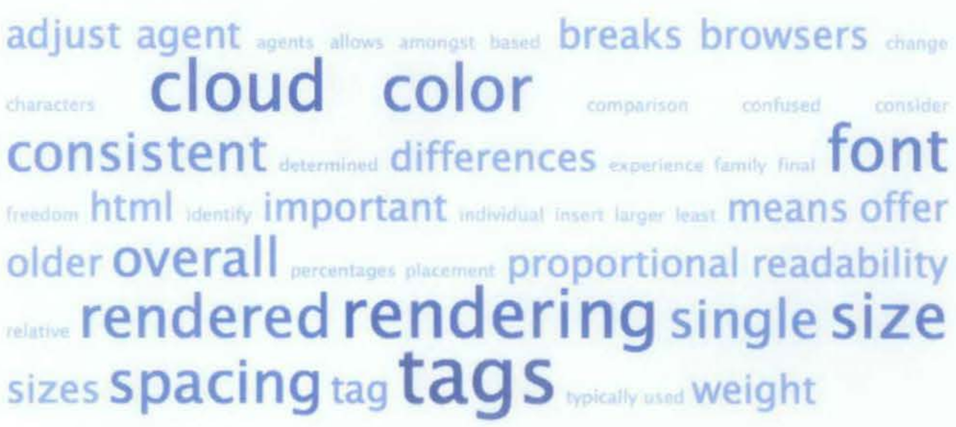
LITERATURE REVIEW AND THEORY

2. LITERATURE REVIEW AND THEORY

In this section, we are going to look at a few alternatives of information visualization. These methods will be compared by studying their pros and cons. In the end, the best among them will be selected for this project. We will be comparing between tag clouds, hierarchical diagrams and mind mapping techniques.

2.1 Tag Clouds

This method is a fairly new technique to represent information visually. Figure 1 is an example of a tag cloud.



adjust agent agents allows amongst based breaks browsers change
characters cloud color comparison confused consider
consistent determined differences experience family final font
freedom html identify important individual insert larger least means offer
older overall percentages placement proportional readability
relative rendered rendering single size
sizes spacing tag tags typically used weight

Figure 1: Example of a Tag Cloud

Phelps (2006) describes a tag cloud as a collection of tags which are each weighted by the amount of times they are used. Each tag is a description of the content or context of an object. To provide further clarification, Hearst and Rosner (n.d.) defines tag clouds as "visual representations of social tags, displayed in paragraph style layout, usually in alphabetical order, where the relative size and weight of the font for each tag corresponds to the relative frequency of its use".

To apply this visual method into this project, larger font tags can be higher level subdirectories and the smallest font will be used to represent end files. Now, the advantage that we can reap from this method is that a hierarchical view of the organization of subdirectories and files can be established. Users can roughly see the way the contents of a directory are organized. Contents of different subdirectories can be divided by drawing boxes or boundaries to improve hierarchy visualization. These tags could also work as links which users can click to directly open them. This function fulfills one of the main requirements for our system.

However, this method does have its disadvantages. Imagine if there are a lot of subdirectories and files, the tag cloud would be too crowded and messy. As a result, searching for a single file would be too time-consuming as users will have to browse through many lines to locate the intended file. Besides that, being too crowded defeats the ability to be visually appealing. Kaser and Lemire (n.d.) states that the tag cloud's width and height has an upper bound as to reduce scrolling. This implies that the increase in the number of tags will also reduce the minimum font size, as the maximum font is not variable, making it hard to read certain tags. In summary, this method is not suitable for large directories. Figure 2 shows how an overcrowded tag cloud might look like.

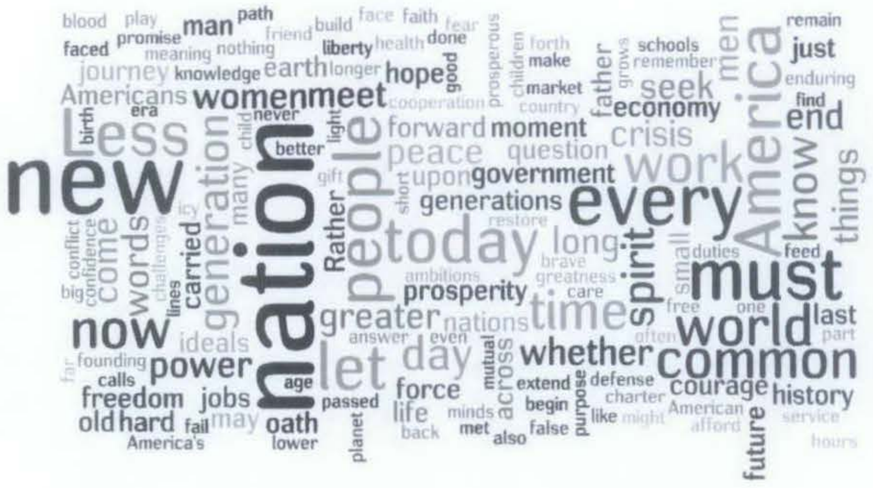
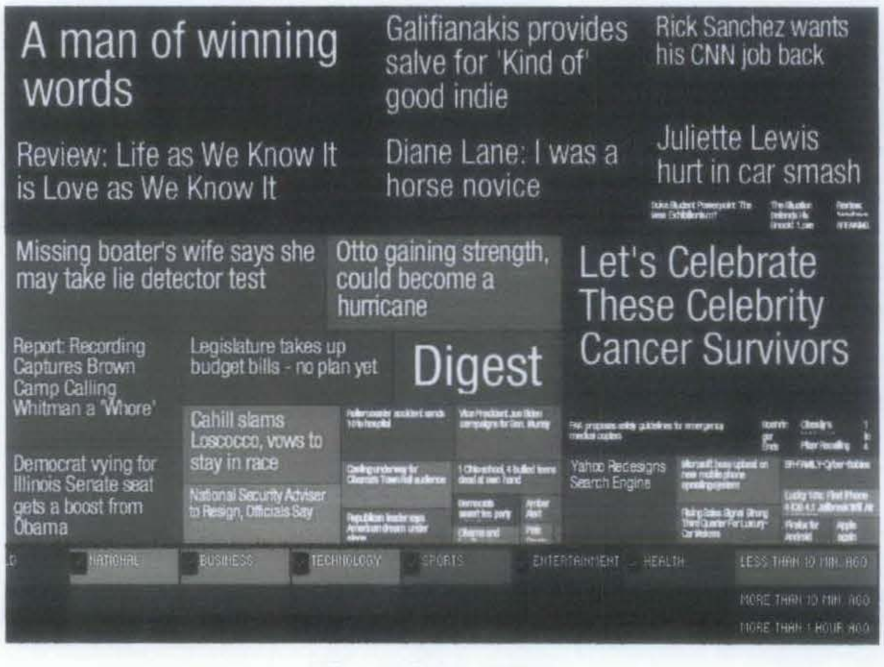


Figure 2: Overcrowded Tag Cloud

An example of current systems that uses this pattern is ***http://newsmap.jp/*** Newsmap is a web application that "visually reflects the constantly changing landscape of the Google News news aggregator" (Marumushi, n.d.). What it does is it represents news headlines with tags. The more popular the news, the bigger the tag and as it becomes older, it turns darker. News tags are updated in real time. Figure 3 is a snapshot of Newsmap.



A man of winning words

Galifianakis provides salve for 'Kind of' good indie

Rick Sanchez wants his CNN job back

Review: Life as We Know It is Love as We Know It

Diane Lane: I was a horse novice

Juliette Lewis hurt in car smash

Missing boater's wife says she may take lie detector test

Otto gaining strength, could become a hurricane

Let's Celebrate These Celebrity Cancer Survivors

Report: Recording Captures Brown Camp Calling Whitman a 'Whore'

Legislature takes up budget bills - no plan yet

Digest

Cahill slams Loscocco, vows to stay in race

Democrat vying for Illinois Senate seat gets a boost from Obama

National Security Adviser to Resign, Officials Say

Yahoo Redesigns Search Engine

NATIONAL BUSINESS TECHNOLOGY SPORTS ENTERTAINMENT HEALTH

LESS THAN 10 MIN. AGO

MORE THAN 10 MIN. AGO

MORE THAN 1 HOUR AGO

Figure 3: Newsmap

2.2 Hierarchical Diagrams

Here, we will be looking at one of the most common visual diagram which is the hierarchy diagram. This diagram is best represented by an organizational chart. Figure 4 shows an example of a simple organization chart.

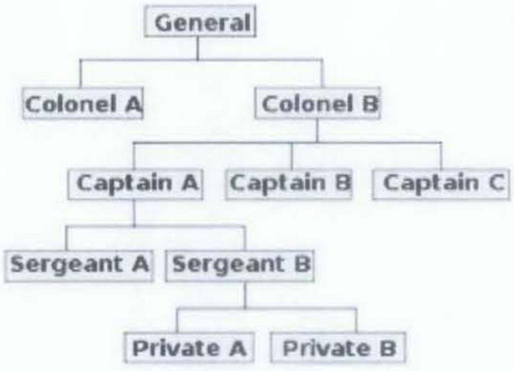


Figure 4: Example of a Simple Organization Chart

This type of diagram offers a clear view of the hierarchy of contents. Users can quickly understand the organization of contents. In the case of an excessive amount of subdirectories and files being present, the diagram can just group the contents of a subdirectory into one entity. The users can expand this group by clicking on it. To explain further, take for example, "Colonel A" from Figure 4. "Colonel A" might be the aggregation of all its contents. In other words, its contents are hidden as to not crowd the overall diagram. Users can click on "Colonel A" to expand and see its contents. The system could work in such a way that, when clicked, "Colonel A" will take the place of "General" (becomes the root) as to maintain the simplicity of the chart. By looking at the advantages, this method seems feasible for our project.

Now, the question lies on whether this conventional method can provide the visual appeal to users. Besides that, is it efficient in terms of minimizing the amount of clicks that the user would have to perform before the intended file or folder is found? Excessive grouping would cause the number of clicks to rise. In other words, this method lacks the ability to work efficiently and does not make full use of the space around it. Next, we will look at mind mapping.

The use of hierarchy diagrams are demonstrated in numerous organizational charts. Almost every organization has a chart that displays the top-down control and relative position of their personnel. These charts are a form of hierarchy diagrams. Figure 5 is an example of this, taken from the Coca-Cola Company.

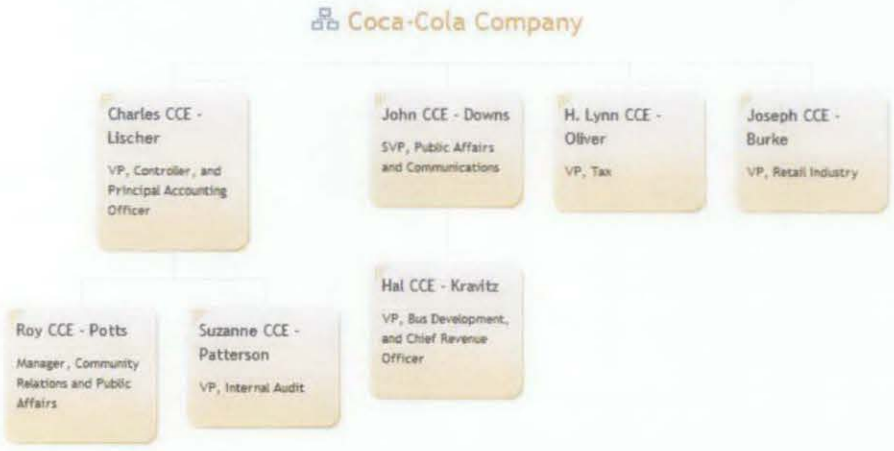


Figure 5: Coca-Cola Company's Organizational Chart (Cogmap, 2010)

2.3 Mind Map

Serrat (2009) describes a mind map as a circular, non-linear way of organizing information. The mind map shows the connection between a central topic and concepts, themes or tasks that relates to it including their levels of relativity. Figure 6 is an example of a simple mind map.

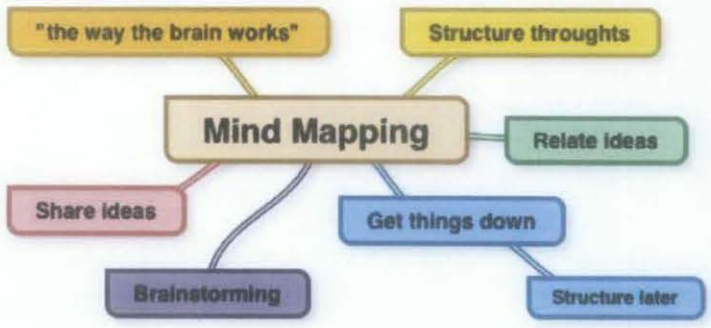


Figure 6: Example of Mind Map

A mind map makes use of the full range of our cortical skills – word, image, number, logic, rhythm, color and spatial awareness – in a single powerful technique which enables us to utilize the infinite expanse of our brain (Buzan, 2000). In other words, a mind map can be both visually appealing as well as a powerful tool to stimulate the user's memory abilities. After viewing the mind map, the users might be able to remember the location of files and folders better. The belief is supported by The Facilitators (1998) as they had stated that "The resultant Mind Map can be 'seen' by the eye and memorized by your visual memory that has been shown to be almost perfect". Lau (1998) also supports this by stating that graphic stimuli can enhance memory because viewers are able to associate keywords with corresponding colors and icons, producing a visual field of reference.

An example of an application that uses the mind map tool is Thinkmap Visual Thesaurus (***http://www.visualthesaurus.com/***). This web application allows you to enter any word and it will generate a mind map that links related words to the word you entered. Figure 7 is example of a mind map generated by this visual thesaurus.

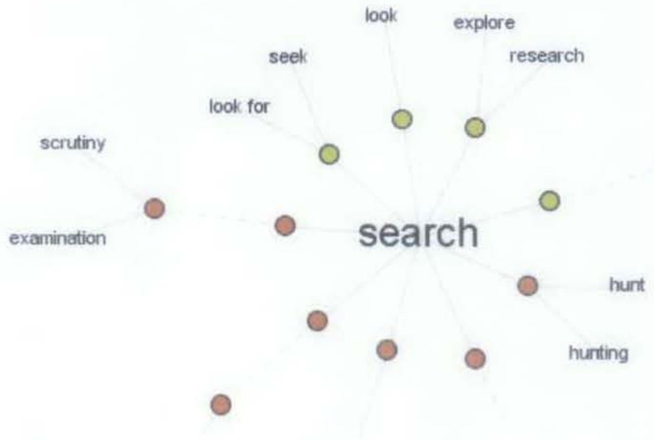


Figure 7: Example of Mind Map from Visual Thesaurus

Without a doubt, by just looking at a mind map, we can easily see the hierarchy and organization of files and folders. Like in the case of the organization chart, the technique of 'grouping' can be utilized here to avoid clutter. This technique is one that groups together an entity with its descendents. As there might be a large amount of files and folders, 'grouping' can prove to be useful. However, we cannot avoid the fact that excessive grouping can eventually backfire and render the search impractical. Just imagine if the file someone is looking for is stored within levels and levels of subdirectories. Users would have to perform numerous clicks before finding the file. Also, the advantage of providing a hierarchical view of the organization of directories and subdirectories will also be rendered void as excessive groupings will confuse the users. Thus, although the mind mapping technique can be seen as a revolutionary way to search for files, it has many potential risks and disadvantages. Sometimes, a simple method is enough to meet the needs of users. We will discuss more on the suitability of this technique in Chapter 4 and 5.

CHAPTER 3
METHODOLOGY

3. METHODOLOGY

3.1 Methodology

The development of the proposed system will follow the Extreme Programming (XP) technique of Agile Development. Figure 8 basically shows the flow between phases in this development technique.

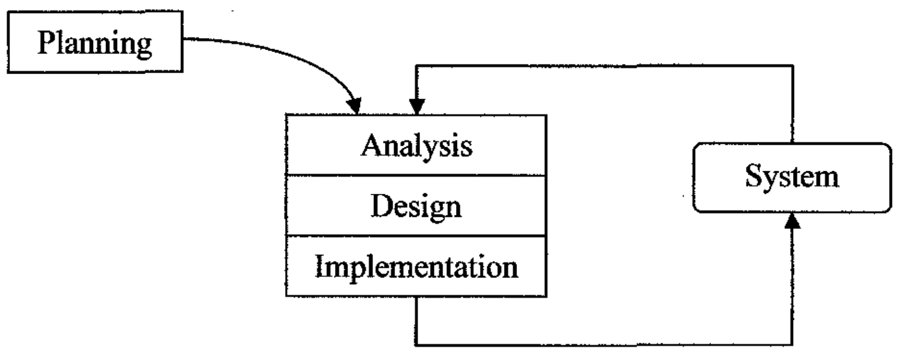


Figure 8: Extreme Programming Methodology

Dennis, Wixom and Tegarden (2005) explain that XP is based on four core values which are communication, simplicity, feedback and courage. However, since this is a small and individual project where requirements for the system are set by none other than myself, values like communication and feedback has limited applications here. Despite that, the nature of this development method makes it highly suitable for the project at hand. I will now proceed to explain this method and its advantages.

Firstly, this technique involves incremental change. After the planning phase, the analysis, design and implementation phases will be repeated if the developed system has not yet been proven acceptable in terms of fulfilling the requirements set by developers and users. In this project, the requirements are roughly determined by my supervisor and me. Throughout the project, there might be sudden changes to those requirements. Since XP is highly adaptable to changes, choosing it as my development method is definitely right. Basically, the system is quickly developed, tested by the developers and then sent for refinement and redevelopment. This quick develop and test cycle is very suitable for a comparatively small project like this. Change for the better is easily implemented using this method.

XP follows the KISS principle which is the practice of simplicity. As this project is an individual project and is fairly small, simplicity is best here. Besides that, XP begins with understanding the basic requirements and then coding will begin in small, simple modules which will be tested upon completion. This allows a quick delivery of a working system and reduces the time needed for requirements gathering. This is advantageous in this project as only basic requirements are identified in the beginning and more specific requirements might unfold as we code the program. The following will describe the main activities that will be undertaken at each phase in the development of the Directory Mind Mapping System.

<u>Project Planning</u>

Purpose: To establish an overview of the intended software product, and uses that to create the basic project structure, evaluate feasibility and risks associated with the project. Besides that, it involves describing appropriate management and technical approaches (Elucidata, n.d.).

Main Activities:

- Define project objectives
- Prepare project proposal
- Create project development schedule

Project Analysis

Purpose: Define high-level goals and refine those goals into a set of one or more requirements (Elucidata, n.d.).

Main Activities:

- Identify project requirements
- Requirements study
- Define functional and technical requirements

Project Design

Purpose: To transform the detailed, defined requirements into complete thorough specifications for the system to direct the work of the Implementation Phase (Maryland, 2008).

Main Activities:

- Design the user interface
- Design the resulting mind map (results display)
- Design the flow of processes by means of a simple sketchbook

<u>Project Implementation</u>

Purpose: Converts the deliverables of the Design Phase into a complete system. Puts in place the hardware, software and other important elements of the system for its successful execution (Maryland, 2008).

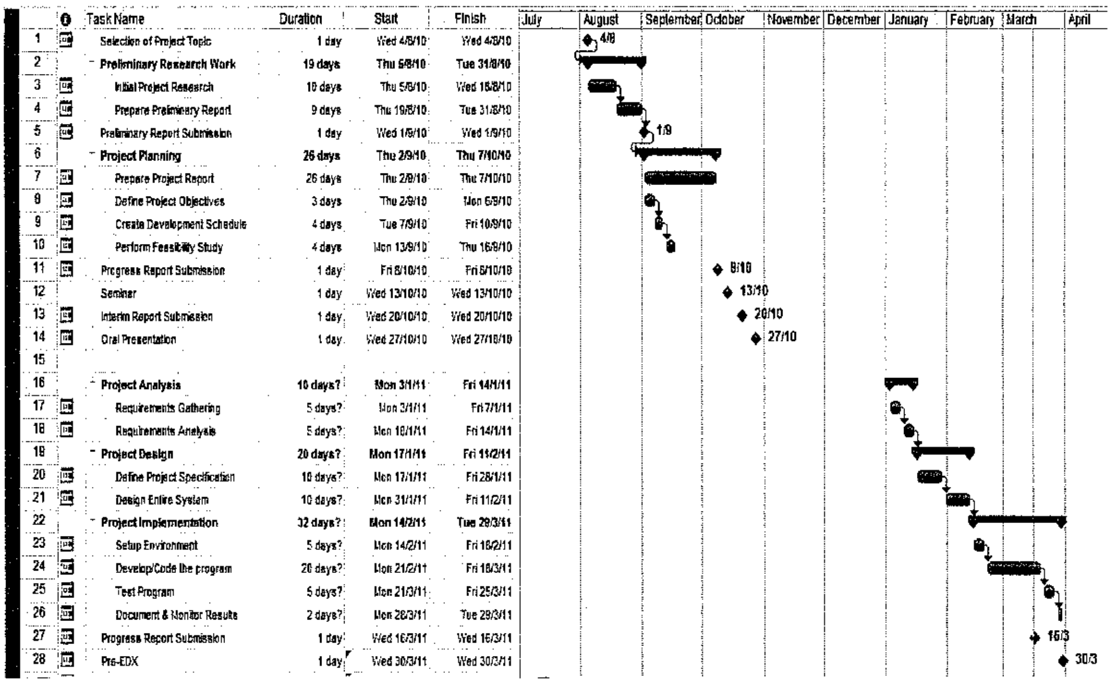
Main Activities:

- Code the program
- Setup environment for system execution
- Test the system and implement changes
- Document the system

NOTE: The analysis, design and implementation phases are iterated in the effort to perfect the system. Testing is done after every iteration to ensure that the system continues to return acceptable results and to reduce bugs in the system.

3.2 Key Milestones and Gantt Chart

All software development projects involve balancing three important aspects which are time, cost and quality. Ideally, the project is completed and delivered on time, within budget and is of good quality (fulfill requirements). However, in real world situations, it is usually hard to achieve that. Thus, in the effort to be as close to the ideal situation as possible, project management should start early and also carefully carried out throughout all the phases in the project methodology. With that in mind, a Gantt chart with all the tasks and key milestones are plotted and planned out. It is important that the datelines in the Gantt chart are strictly adhered to. Only then can the project development go efficiently. Figure 9 shows a Gantt chart containing the tasks and key milestones of my Final Year Project. A more comprehensive view of the Gantt chart is attached as Appendix B at the end of this report.



	Task Name	Duration	Start	Finish
1	Selection of Project Topic	1 day	Wed 4/8/10	Wed 4/8/10
2	- Preliminary Research Work	19 days	Thu 5/8/10	Tue 31/8/10
3	Initial Project Research	10 days	Thu 5/8/10	Wed 18/8/10
4	Prepare Preliminary Report	9 days	Thu 19/8/10	Tue 31/8/10
5	Preliminary Report Submission	1 day	Wed 1/9/10	Wed 1/9/10
6	- Project Planning	26 days	Thu 2/9/10	Thu 7/10/10
7	Prepare Project Report	26 days	Thu 2/9/10	Thu 7/10/10
8	Define Project Objectives	3 days	Thu 2/9/10	Mon 6/9/10
9	Create Development Schedule	4 days	Tue 7/9/10	Fri 10/9/10
10	Perform Feasibility Study	4 days	Mon 13/9/10	Thu 16/9/10
11	Progress Report Submission	1 day	Fri 8/10/10	Fri 8/10/10
12	Seminar	1 day	Wed 13/10/10	Wed 13/10/10
13	Interim Report Submission	1 day	Wed 20/10/10	Wed 20/10/10
14	Oral Presentation	1 day	Wed 27/10/10	Wed 27/10/10
15				
16	- Project Analysis	10 days?	Mon 3/1/11	Fri 14/1/11
17	Requirements Gathering	5 days?	Mon 3/1/11	Fri 7/1/11
18	Requirements Analysis	5 days?	Mon 10/1/11	Fri 14/1/11
19	- Project Design	20 days?	Mon 17/1/11	Fri 11/2/11
20	Define Project Specification	10 days?	Mon 17/1/11	Fri 28/1/11
21	Design Entire System	10 days?	Mon 31/1/11	Fri 11/2/11
22	- Project Implementation	32 days?	Mon 14/2/11	Tue 29/3/11
23	Setup Environment	5 days?	Mon 14/2/11	Fri 18/2/11
24	Develop/Code the program	20 days?	Mon 21/2/11	Fri 18/3/11
25	Test Program	5 days?	Mon 21/3/11	Fri 25/3/11
26	Document & Monitor Results	2 days?	Mon 28/3/11	Tue 29/3/11
27	Progress Report Submission	1 day	Wed 16/3/11	Wed 16/3/11
28	Pre-EDX	1 day	Wed 30/3/11	Wed 30/3/11

Figure 9: Snapshot of Project Tasks Gantt Chart

NOTE: The diamond icon in the Gantt chart (Figure 9) indicates a milestone.

3.3 Tools Required

1. Netbeans IDE

   An integrated development environment or simply, a compiler is needed to compile, build and run the Java codes. This IDE provides other useful functionalities like easy creation of a graphical user interface and the transformation of codes into a standalone program (.jar).

2. Adobe Photoshop or Paint

   An image editing software is needed to design or modify icons or images to be used in this report. There will also be minor editing of icons to be displayed on the program.

CHAPTER 4
RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

4.1 Overview

As stated in the *Background* section of this report, the question of whether a better search facility exists or whether there is an alternative method to search for files and folders.

In the effort to find the answers to these questions, surveys and researches were carried out to identify and create a better, more effective search method. Of course, this search method would be compared to the Windows 'Search' function at the end. The survey was distributed through email and the answers were solicited and reviewed. Information related to this subject and results of findings were mostly derived from reference books, journals and the internet. This section discusses the findings of both my survey and personal researches. Besides that, this section also includes the initial graphical user interface design of the system, a system flowchart and the detailed explanation of the proposed search program/system.

4.2 Survey

Before going into the results of the survey, here is a brief overview of the survey. This survey was distributed through email to various students of University Teknology PETRONAS (UTP) from different years, programs, race and origin. 20 students replied with their answers and the results in section 4.2.1 are actually the aggregation and summation of the answers given by those 20 students, each with diverse backgrounds. These students are of course between the age of 20 and 25 years old. This age group is chosen solely because they are frequently in need to search for files or folders in their personal computer. Their input might be vital to this research. The survey questions are attached as Appendix A.

4.2.1 Survey Results

1. How would you normally search for files stored in your computer?

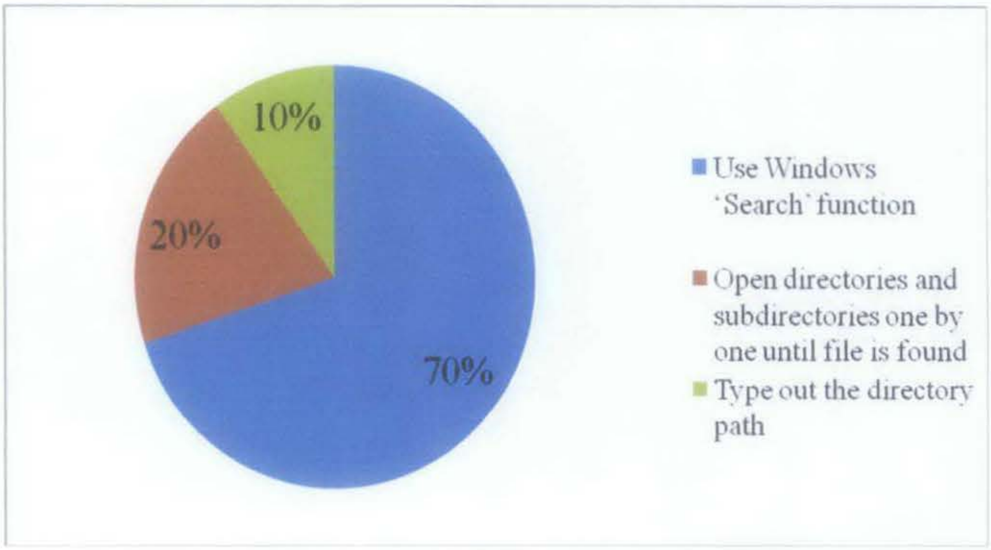


Figure 10 (a): Results of Survey Question 1

70% answered '*Use Windows 'Search' function*'

20% answered '*Open directories and subdirectories one by one until file is found*'

10% answered '*Type out the directory path*'

2. How would you search for files if you have totally forgotten its name and location?

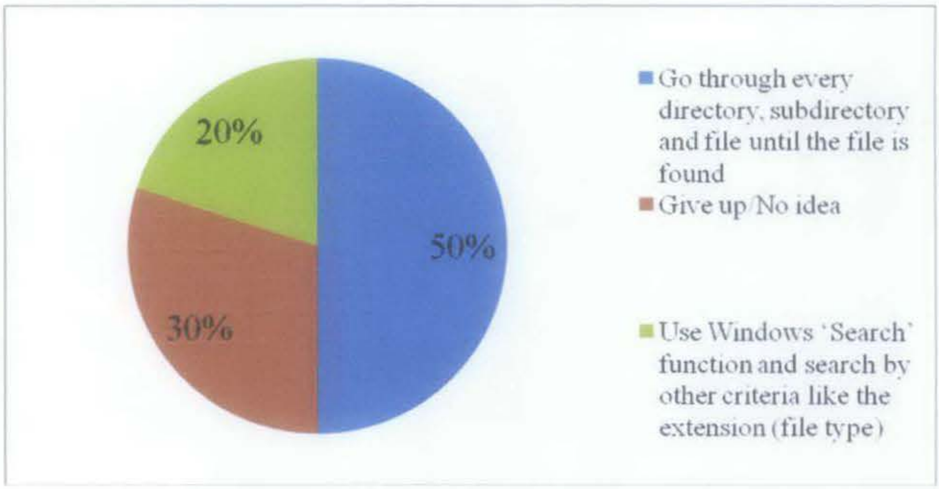


Figure 10 (b): Results of Survey Question 2

50% answered '*Go through every directory, subdirectory and file until the file is found*'

30% answered '*Give up/No idea*'

20% answered '*Use Windows 'Search' function and search by other criteria like the extension (file type)*'

3. Which search method would you prefer to use? (Please choose either a, b or c)

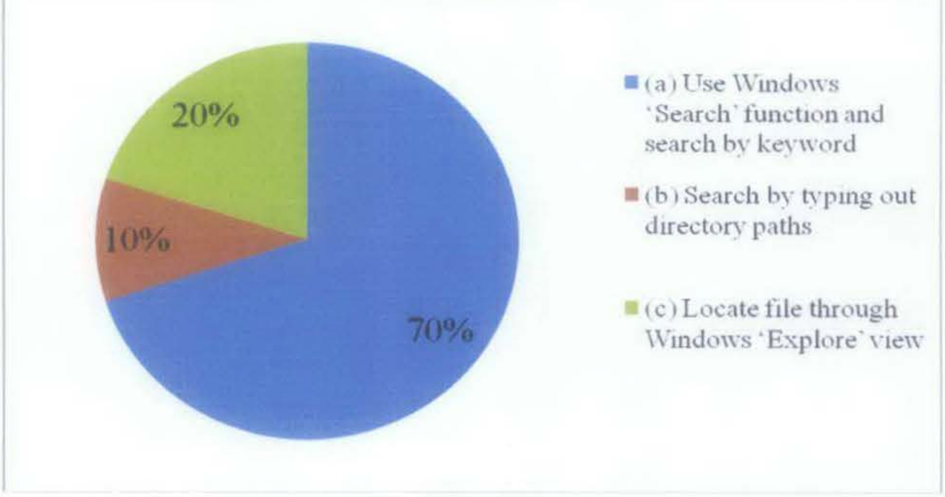


Figure 10 (c): Results of Survey Question 3

70% answered:	*a). Use Windows 'Search' function and search by keyword*
10% answered:	*b). Search by typing out directory paths (Eg: Typing out 'C:\Program Files\Microsoft Office')*
20% answered:	*c). Locate file through Windows' 'Explore' view (View hierarchy of folders/files and click on directories to expand them in order to view their contents)*

4.2.2 Survey Results Review

From the results above, we can see that the majority of people are very reliant on the Windows 'Search' function. Of course, this does not come as a surprise because this is the only search facility available in Windows. Most people are used to it and have the opinion that it is the fastest method to search for files. However, interestingly enough, when users find themselves in a situation where they cannot remember both the location and name of the file they are looking for, they abandon this 'Search' function. From *Question 2* of the survey, we notice that a total of 80% gave up on the idea of using the 'Search' function when initially 70% of them actually stated that they prefer searching for files using this function (from *Question 3*).

This brings us to the question: Is the 'Search' function really effective enough to search for files and folders during all occasions? How about during extraordinary cases like in *Question 2*? This survey clearly informs us that a more effective or an alternative search method can be invented. To create the perfect search method is a bit far-reaching so we should focus on creating an alternative search method to cater for unusual circumstances. These circumstances can be like what was described earlier, when users forget the name and location of a file or even when users vaguely remember the file contents but forget its name or location. To avoid cases like in Question 2, which is forgetting the file name and location, we should visually represent the hierarchy of directories, subdirectories and files in a way that can be easily remembered. As we have discussed in the *Literature Review* section of this report, the mind map is a very effective tool in achieving this objective. The mind map can be both visually appealing as well as a powerful tool to stimulate the user's memory abilities. By building our new search program upon the principles and structure of the mind map, we can obtain its advantages. If the hierarchy of files and folders are displayed in the structure of a mind map, users can easily remember and recall the locations of files later. This is because they also utilize their visual memory.

Besides that, the ability to quickly grasp the organization of files and folders allow users to quickly identify its mistakes or flaws. Once they can see the weaknesses of the current organization, they can quickly work to reorganize files in a way that is easily accessible, understood and remembered. This shortens future search-time of files and folders. All these are the potential benefits and advantages of the mind mapping search method.

However, while designing and researching about the application of mind mapping technique in search programs, many flaws and disadvantages were also found. Firstly, there is the problem of excessive groupings. We must consider the fact that there are C and D drives in Windows that stores hundreds of directories and subdirectories. Also, we cannot deny the fact that users would define, let us say, the C drive as the input directory path. When this happens, a mind map too large to be displayed on one screen will be produced. This means that excessive groupings will happen to adjust the size of the mind map to fit the viewing pane. This has implications like causing the organization of files and folders to look confusing and too many clicks required before reaching a desired file. We have to admit that if this is the case, users would just fall back to using the Windows 'Search' facility. That way, we will fail to achieve our objective. Thus, an alternative should be proposed. After studying the ways to search files and folders, an interesting method arises. This alternative method is to search for file contents. This method still fulfills our objective of developing an alternative search system that would prove to be useful.

4.3 Graphical User Interface (GUI) Design

The system that is to be developed here is a standalone application. Thus, it really does not need to have interfaces to communicate with third party applications. With this in mind, the main design here would be the graphical user interface that the system must have in order to be able to 'converse' with the users. This section will provide an insight of the initial graphical user interface. Note though, that since we are following XP programming methodology, the graphical user interface might change as user requirements change. The XP programming methodology allows for a fast working increment of the system to be developed. Once that increment is tested and observed, additional requirements might be added and a new increment of the system would eventually be underway.

As mentioned in the previous section, I am proposing the search-by-content method instead of the mind mapping technique as an alternative search program. To further support this decision, this section will cover a comparison between the GUI of a search-by-content search program and the mind mapping search program. Figure 11 shows the initial design of the graphical user interface for the Windows Directory Mind Mapping application.

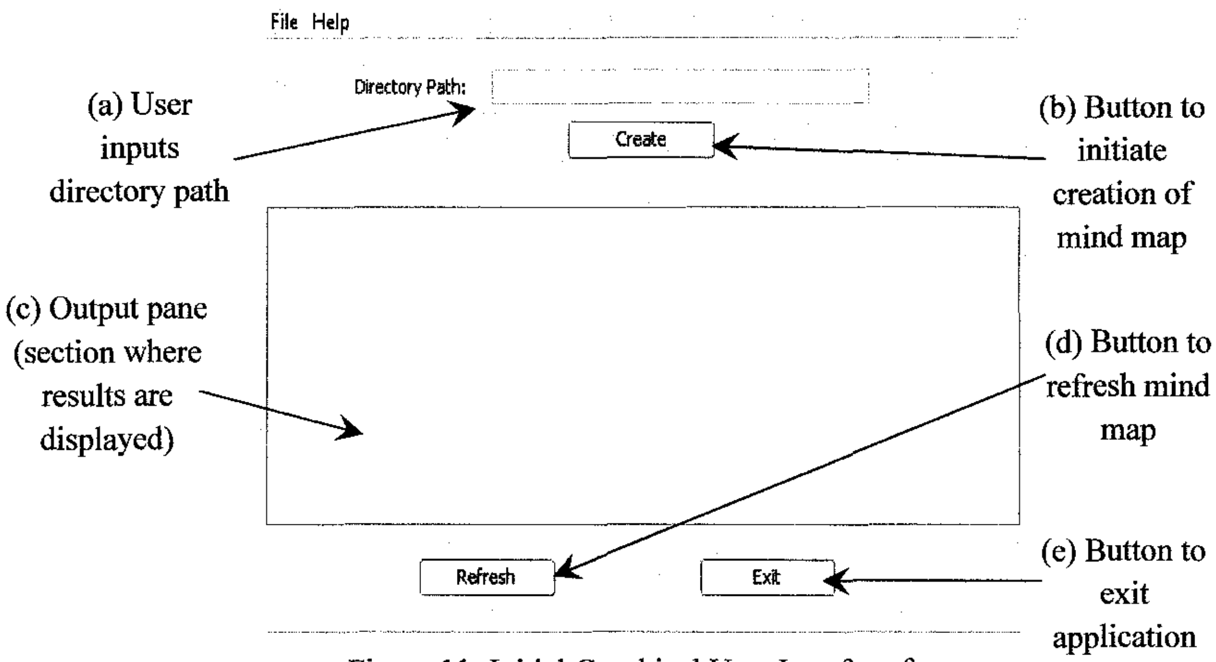

File Help
Directory Path:
Create
Refresh
Exit
(a) User inputs directory path
(b) Button to initiate creation of mind map
(c) Output pane (section where results are displayed)
(d) Button to refresh mind map
(e) Button to exit application

Figure 11: Initial Graphical User Interface for Windows Directory Mind Mapping System

As for the output, the results will be shown in a mind map format. The application will get the directory path from the user and when the user clicks the 'Create' button, the application will generate a mind map with the user specified directory as the root of the map. A more detailed flow of events is explained in section 4.4. The mind map should look something like the mind map in Figure 12. There should be icons to differentiate a file and a folder.

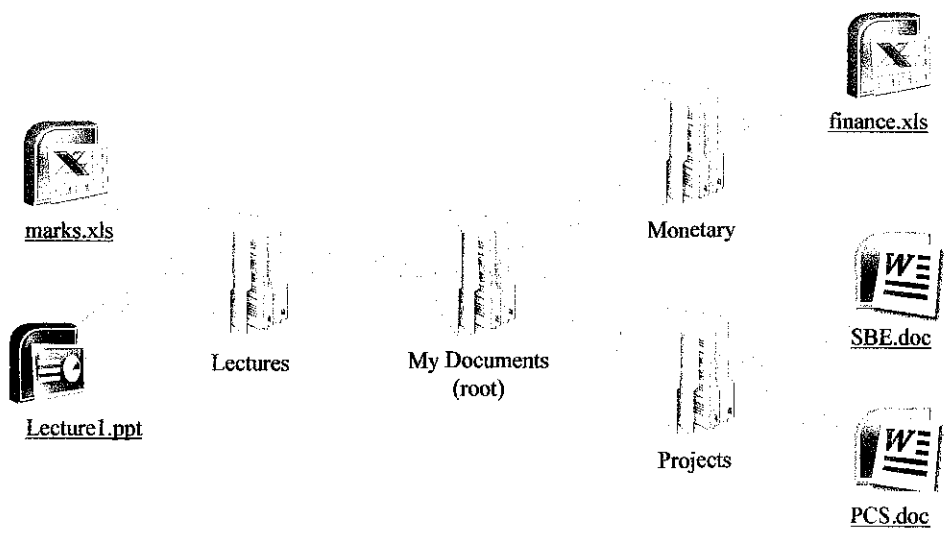
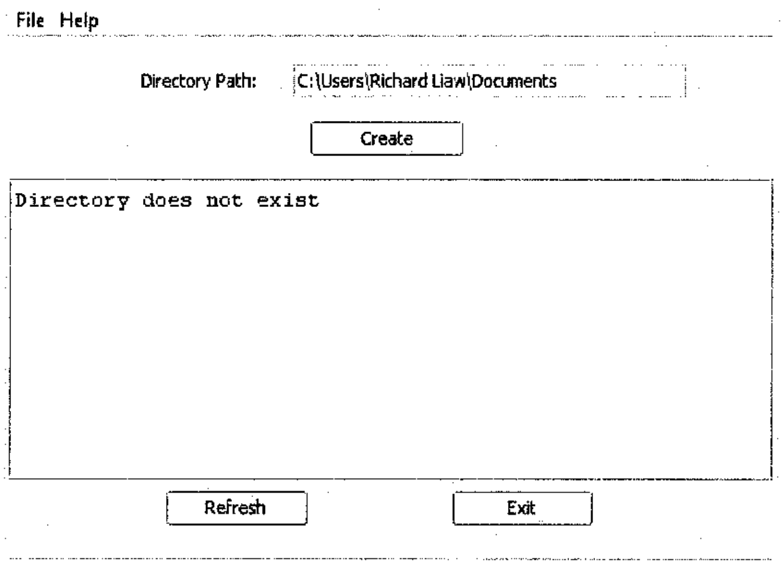


Figure 12: Example of Resulting Mind Map

If the users were to enter an invalid input or directory path, a simple error message will be displayed on the output pane. There is no need for a pop-up here. Figure 13 gives an example of this.



```
Directory does not exist
```

Figure 13: Error Message Design for Mind Mapping System

However, the mind mapping method has a lot of disadvantages when it comes to file search. Displaying a mind map in the output pane is a new technique, one that is unfamiliar to the users. What if a tree is displayed instead? This tree still utilizes the benefits of a mind map: enhances memory of file location and provides a hierarchical view of files and folders. Not to mention, this view is familiar to users. Sometimes, creating something new is not always the best solution. Creating something familiar and usable might turn out to be the best choice. Thus, an alternative method is proposed: the search-by-content method. This program will first obtain the directory path and a keyword or key-phrase from the user. Then, the program will access the directory specified and search for all files that contain the keyword or key-phrase. This means that the program will iteratively open files/folders and inspect their contents for the specified keyword/key-phrase. Finally, the program will display all files that contain the keyword/key-phrase neatly in a tree. The structure of this tree will be further explained in the coming sections. Another advantage of this is that the potential size of the tree is reduced. This reduces the problem of excessive clicking.

Figure 14 shows the initial design of the graphical user interface for the Search-By-Content application.

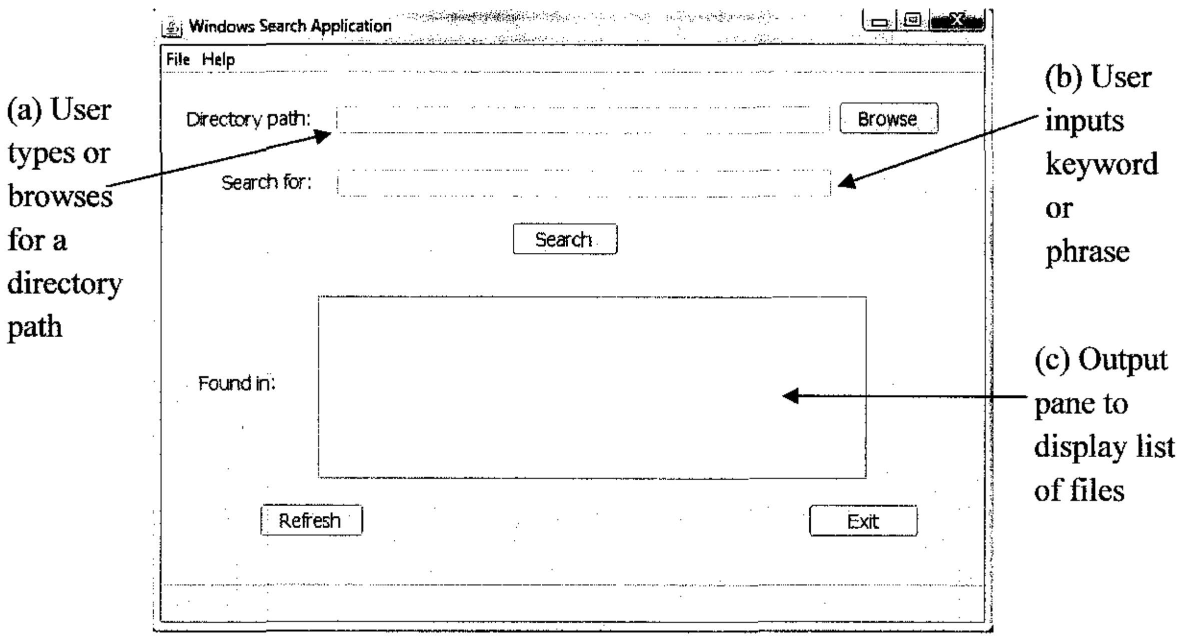
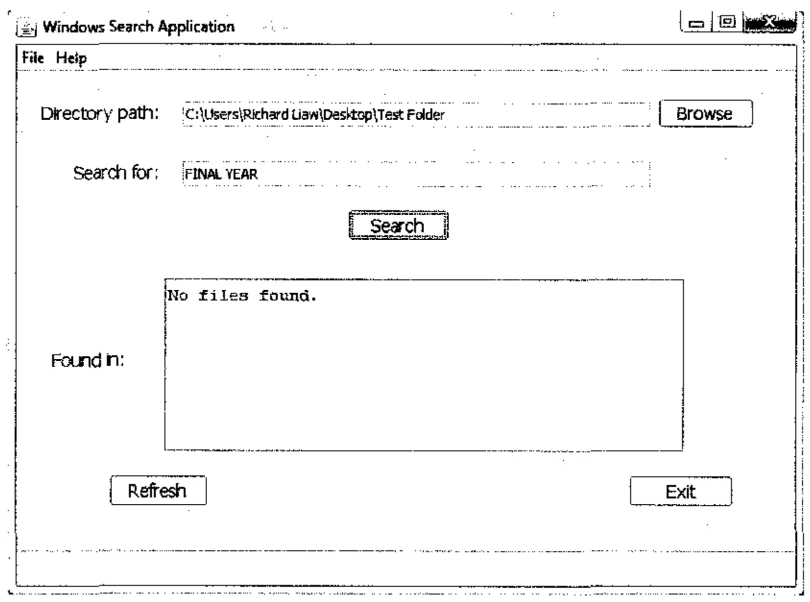

Windows Search Application
File Help
Directory path:
Browse
Search for:
Search
Found in:
Refresh
Exit
(a) User types or browses for a directory path
(b) User inputs keyword or phrase
(c) Output pane to display list of files

Figure 14: Initial Graphical User Interface for Search-By-Content System

The rest of the elements are similar to the Mind Mapping Search System. Besides that, the error message, if generated, are also displayed in the output pane. Figure 15 below shows an example of an error message for the Search-By-Content System if no files were found.



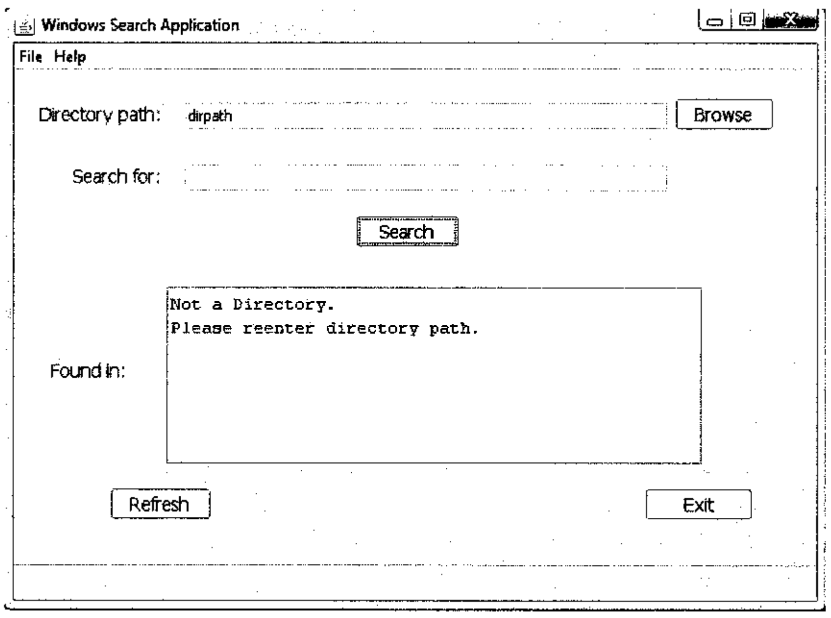
Directory path: C:\Users\Richard Liaw\Desktop\Test Folder

Search for: FINAL YEAR

```
No files found.
```

Figure 15: 'No files found' Error Message for Search-By-Content System

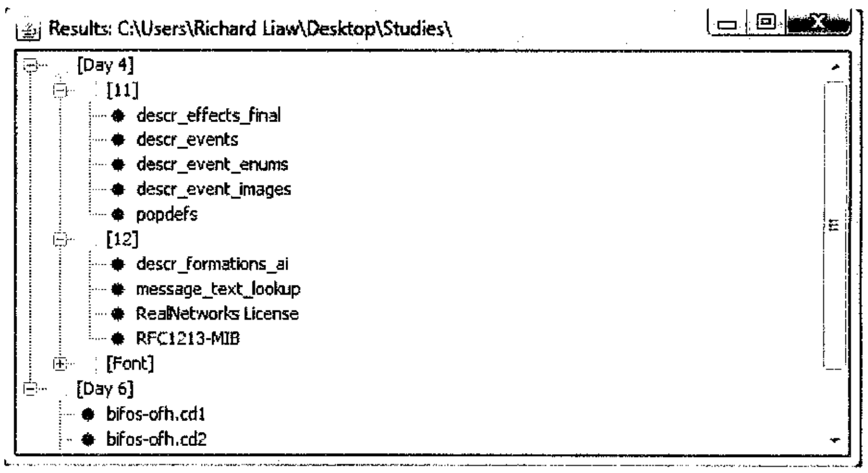
Figure 16 below shows an example of an error message for the Search-By-Content System when a user enters an invalid directory path.



```
Not a Directory.
Please reenter directory path.
```

Figure 16: Invalid Directory Path Error Message for Search-By-Content System

If we refer back to Figure 14 in page 26, a section (c) is labeled and identified as the output pane. The output pane only displays the results in a list and NOT in a tree diagram. The tree will appear in a pop-up window instead. Figure 17 shows an example of a tree generated after a search has been initiated.



```
Results: C:\Users\Richard Liaw\Desktop\Studies\
[Day 4]
    [11]
        descr_effects_final
        descr_events
        descr_event_enums
        descr_event_images
        popdefs
    [12]
        descr_formations_ai
        message_text_lookup
        RealNetworks License
        RFC1213-MIB
    [Font]
[Day 6]
    bifos-ofh.cd1
    bifos-ofh.cd2
```

Figure 17: Pop-up Window Displaying Resulting Tree

NOTE: Figure 17 above has folders like ‘Day 4’, ‘11’, etc. They are of no significance but rather, random folders created just for this example.

4.4 System Flow Chart

This section displays the flowchart of each method: mind mapping and search-by-content with the intention of providing a better view of the differences between the two and their potential as a search program.

4.4.1 Flow Chart for Windows Directory Mind Mapping System

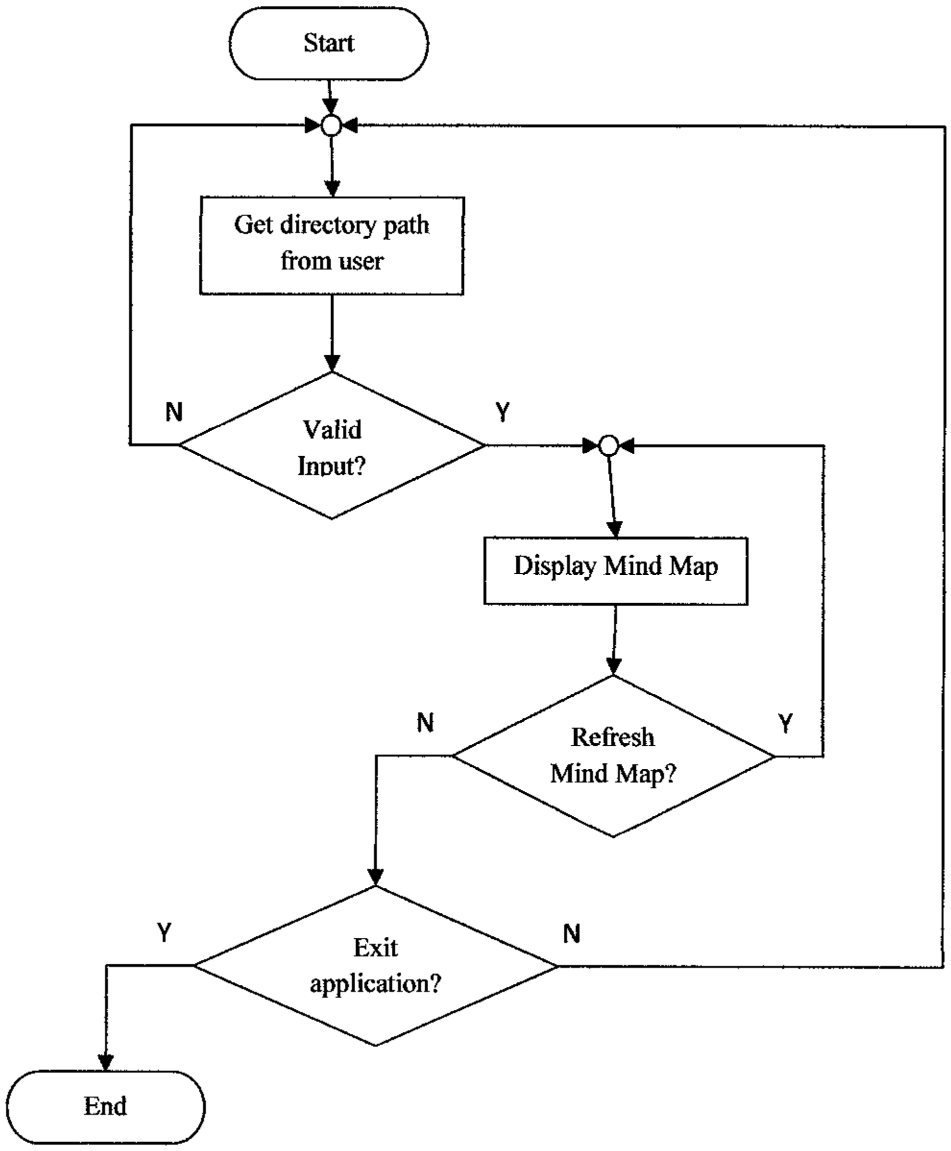


Figure 18: Windows Directory Mind Mapping System Flow Chart

4.4.2 Flow Chart for Windows Search-By-Content System

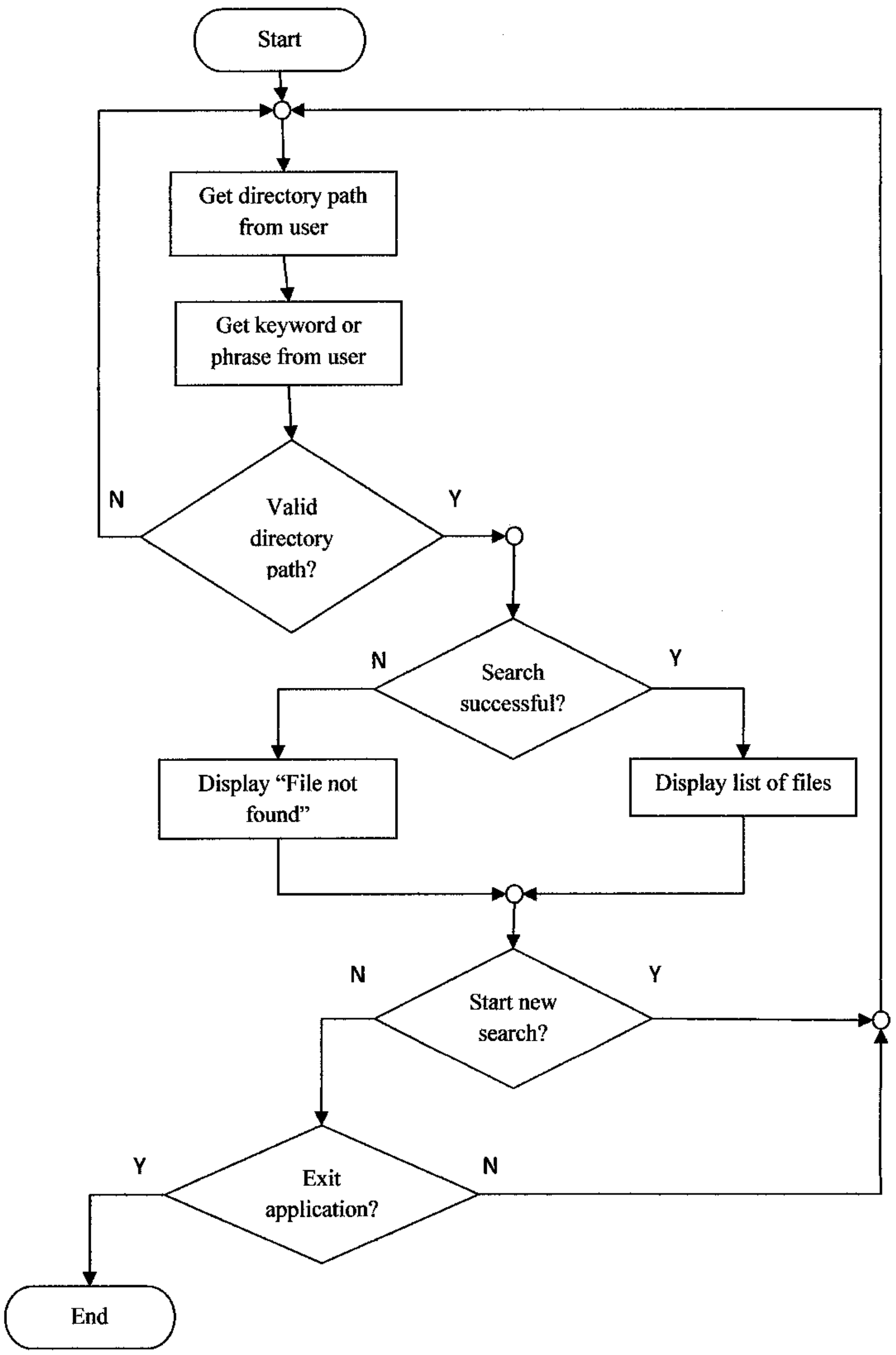


Figure 19: Windows Search-By-Content System Flow Chart

4.5 Detailed Explanation of the Search-By-Content Program

4.5.1 Directory Path

A directory path is required as an input from the user. Of course, the program will check for the validity of the path. If the directory path does not exist or the input from user is not a directory path at all, an error message is displayed. An example of this error message is already given in section 4.3. To further aid the users, this program includes a 'Browse' button. By clicking this button, a GUI file chooser will appear in a pop-up window. This means that users do not need to remember a directory path or even copy and paste it from another window or file. By browsing through the computer, users can select any folder they wish to set as the target directory. By setting a target directory and clicking 'Open', the directory path will automatically be transferred into the 'Directory Path' text field of the program. To illustrate the process above, a series of screenshots are provided below.

Figure 20 below shows the directory path text field and the 'Browse' button at the end of it.

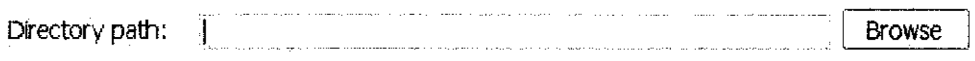
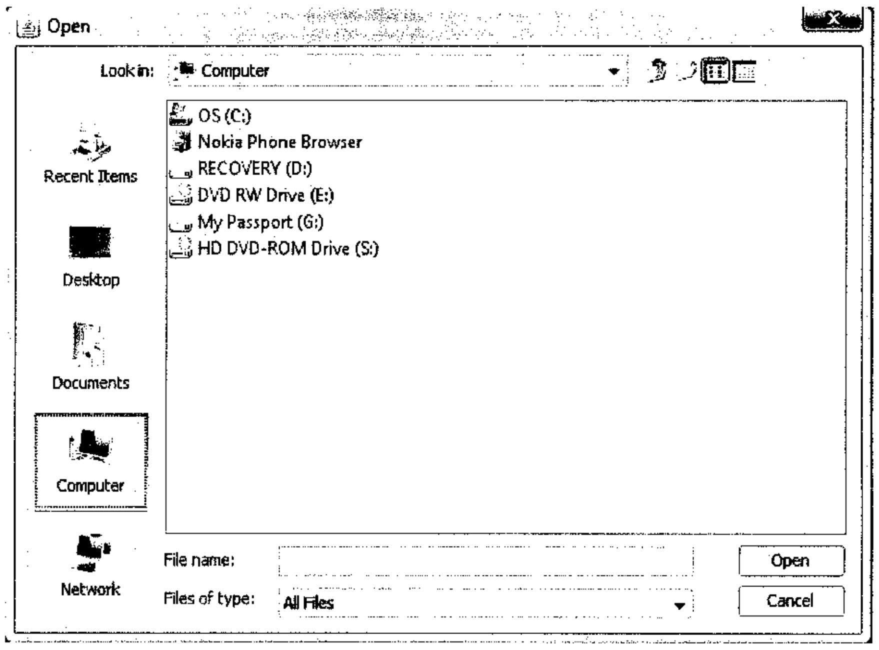


Figure 20: Directory Path Text Field

As mentioned, users can manually type out a directory path or click on the 'Browse' button. Figure 21 shows the pop-up window containing the file chooser.


Open

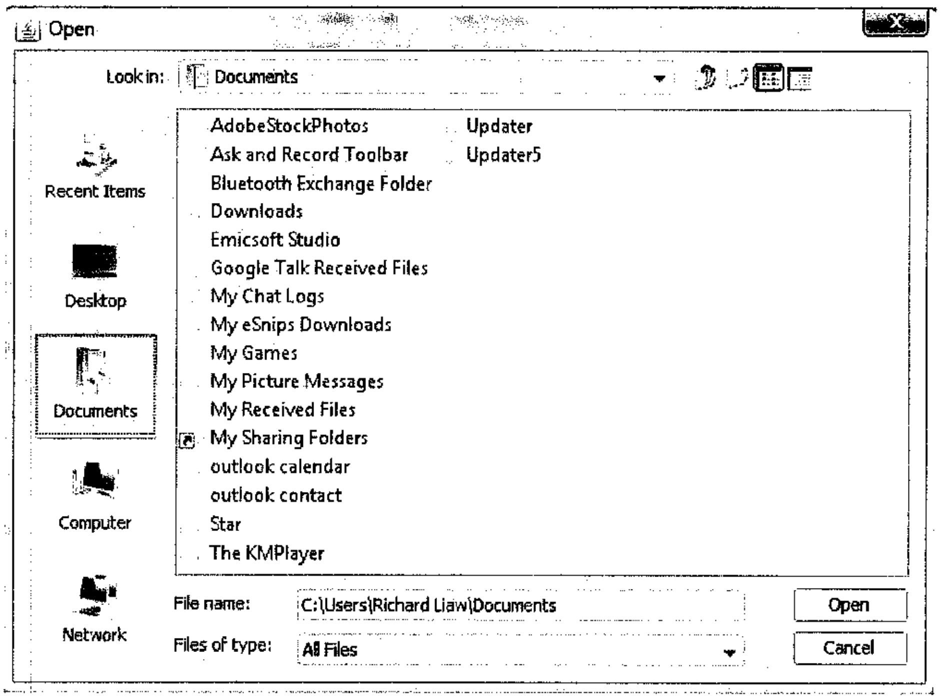
Look in: Computer

OS (C:)
Nokia Phone Browser
RECOVERY (D:)
DVD RW Drive (E:)
My Passport (G:)
HD DVD-ROM Drive (S:)

File name:
Files of type: All Files

Figure 21: Pop-up File Chooser

Users can browse through the directories in the computer and can select any folder they wish to set as the target directory. Figure 22 will show an example when a user selects the directory of 'My Documents' as the target directory.


Open

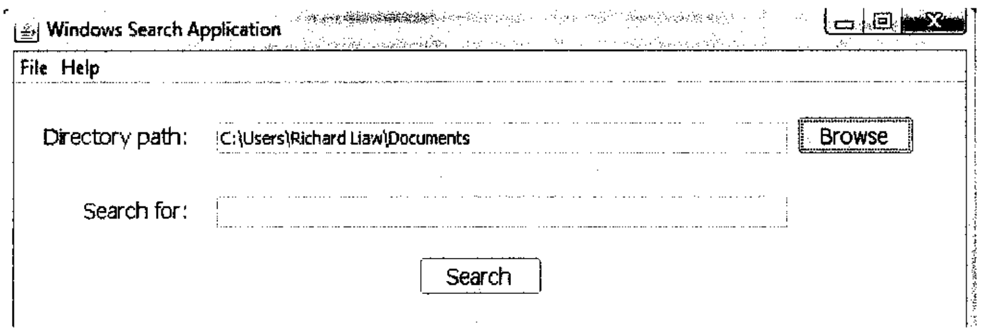
Look in: Documents

AdobeStockPhotos
Ask and Record Toolbar
Bluetooth Exchange Folder
Downloads
Emicsoft Studio
Google Talk Received Files
My Chat Logs
My eSnips Downloads
My Games
My Picture Messages
My Received Files
My Sharing Folders
outlook calendar
outlook contact
Star
The KMPlayer
Updater
Updater5

File name: C:\Users\Richard Liaw\Documents
Files of type: All Files

Figure 22: Setting 'My Documents' as Target Directory

By simply clicking the 'Open' button in Figure 22, the directory path of 'My Documents' will be transferred into the directory path text field of the program. Figure 23 will show the end result - after clicking the 'Open' button.



C:\Users\Richard Liaw\Documents

Figure 23: After Opening a File in File Chooser

4.5.2 Keyword/Key Phrase Search

Next, the user is required to enter a keyword or a phrase. The purpose of this is to provide the program with an argument to search for. For now, the program will search for the keyword or phrase as a whole. This means that the search will only return true on files that contain the exact match to the keyword or phrase entered by the user. The only exception is the upper or lower case of the keyword or phrase. It does not matter in what case the user enters the keyword or phrase. For example, if the user were to enter 'Final' as the keyword or phrase, the program will search for the word 'final' in lower and upper case including a combination of both. The search is not limited to only alphabetic input. A combination of any kind of character is also accepted (eg: numbers and punctuation marks). To illustrate, Figure 24 shows the text field to enter keyword or phrase.

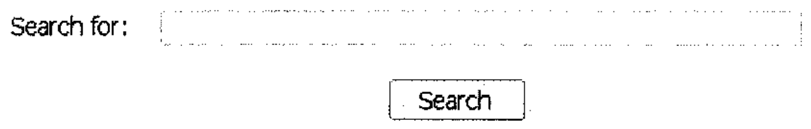


Figure 24: Keyword/Key Phrase Text Field

Say for example, the user searches for the word 'study'. Figure 25 shows the word 'study' being entered into the text field.

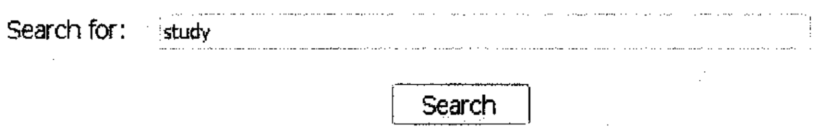
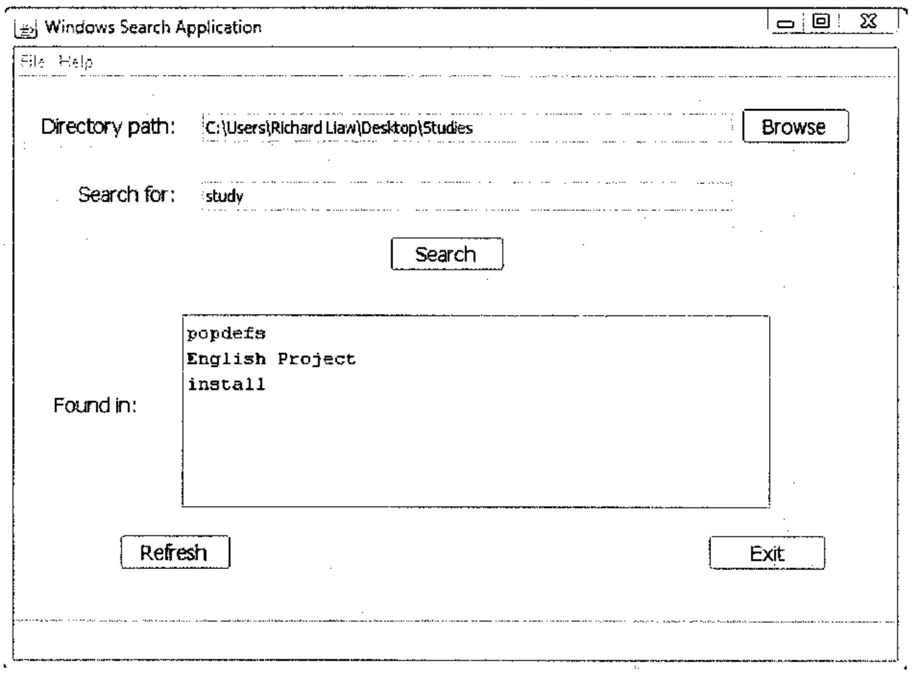


Figure 25: Search for 'study'

When the 'Search' button is clicked, the program will then go into what is similar to a loop of opening files one after another in search of the word. If there are subdirectories in the specified directory, the program will open the subdirectory and the same 'loop' will occur. The same goes for subdirectories of subdirectories. The search will only end when all the subdirectories under the user specified directory were searched. Immediate after the search ends, a list of files will appear in the output text area. The files displayed in the list are the files that contain the designated keyword or phrase, or in our case, the word 'study'. Figure 26 shows the list generated when the word 'study' is being searched.



Directory path: C:\Users\Richard Liaw\Desktop\Studies

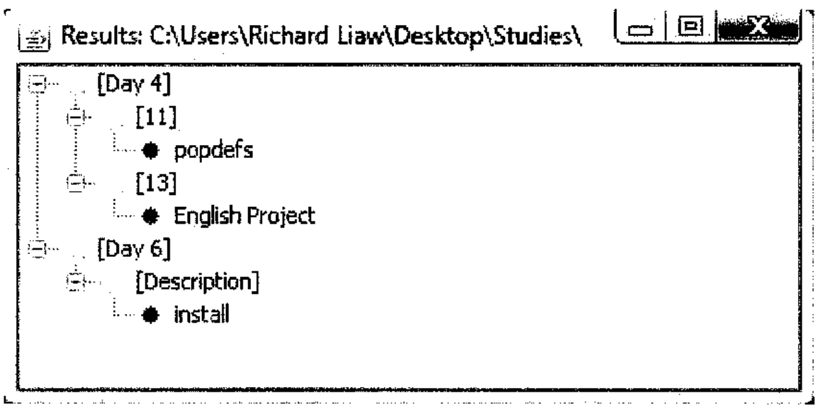
Search for: study

```
popdefs
English Project
install
```

Figure 26: List of Files Containing the Word 'study'

Of course, this list does not provide a hierarchical view of the location of the files. Like mentioned, a tree will be generated as it provides the benefits that a mind map can offer on top of being familiar to users (not familiar to users if in the case of a mind map). Familiarity goes a long way. It can even affect users' acceptance of the program. More in-depth analysis will be discussed in the *Conclusion & Recommendation* section.

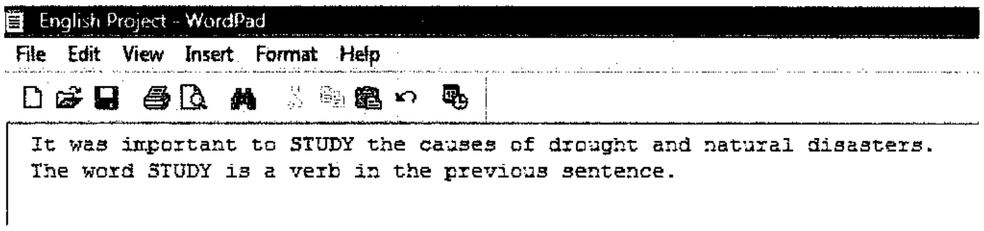
Figure 27 below shows the expanded tree generated when the word 'study' is being searched. This tree shows up in a pop-up window immediately after the search is complete.



```
Results: C:\Users\Richard Liaw\Desktop\Studies\
[Day 4]
    [11]
        ● popdefs
    [13]
        ● English Project
[Day 6]
    [Description]
        ● install
```

Figure 27: Tree Containing Files with the Word 'study'

If we were to look at the contents of the files, we can see that the file 'English Project' contains the word STUDY and still passes the search criteria. Figure 28 shows an extract from the contents of 'English Project'.

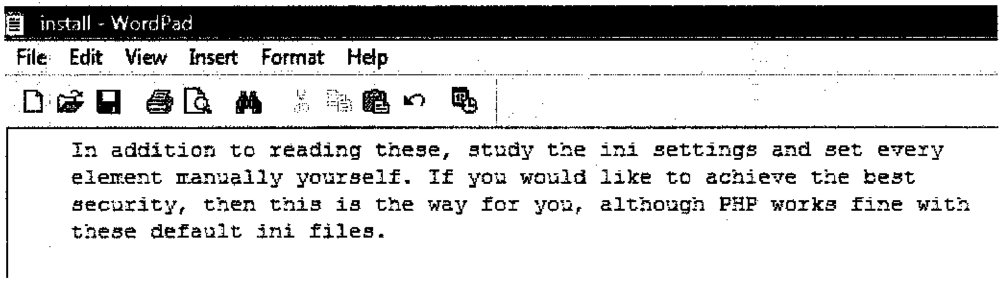


```
It was important to STUDY the causes of drought and natural disasters.
The word STUDY is a verb in the previous sentence.
```

Figure 28: Extract from the Contents of 'English Project'

NOTE: For the simplicity of our example, the file extracts shown are all from simple text files.

Figure 29 shows the contents of the file 'install'. This file contains the word 'study' with all letters in the lower case.



```
In addition to reading these, study the ini settings and set every
element manually yourself. If you would like to achieve the best
security, then this is the way for you, although PHP works fine with
these default ini files.
```

Figure 29: Extract from the Contents of 'install'

The example above proves that the search is NOT case sensitive but it does only return true for files containing words that are the exact match to the keyword in all other aspects.

4.5.3 Internal Implementation

The 'Search' button does deserve a bit more attention and explanation. This button is the main functionality of the whole program and most of the background operations and processes are triggered by this button. To explain, this button listens for an action performed by the user. In this case, the action is a button click. When clicked, the program will first get the text from both the directory path and keyword/key phrase text fields. Then, the program will check whether the directory path entered is valid or not. After which the program will enter into a recursion. A recursion or more specifically, a recursive method is basically one that works very much like a loop. The difference is, instead of running a counter and looping as long as the counter is within range, the recursive method is one that repeats by invoking/calling itself. The recursive method in this program basically goes through all the files stored under the specified directory, including the files of all subdirectories (or subdirectories of subdirectories and so on so forth).

At every file, the program will open a file input stream and read the contents of that file line after line. The purpose is to search for the intended keyword or phrase. At the successful discovery of the first instance of the keyword or phrase in a file, the line-by-line reading of that file will end and the program will proceed to search in the next file. All files that are found containing the keyword or phrase will be added into a Vector. A Vector in Java (object oriented programming) is basically a collection of objects. Since a file is an object, Vector is definitely suitable to be used here. The main advantage of using a Vector in this program is its ability to contain other Vectors. Now, if Vectors can contain other Vectors, this means that the hierarchy of files and folders can be recorded and maintained. The Vectors are created and objects are added according to the relative hierarchy and location of files and folders. To explain it in clearer words, every time a subdirectory is encountered, a new Vector will be created and added into its parent Vector. Every time a file is encountered, a new file object is initialized and added into its parent Vector. This continues until all files and folders are recorded as a hierarchy of Vectors and objects. Take for example the hierarchy of files and folders in Figure 30.

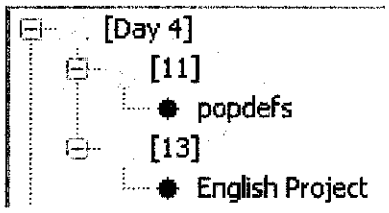
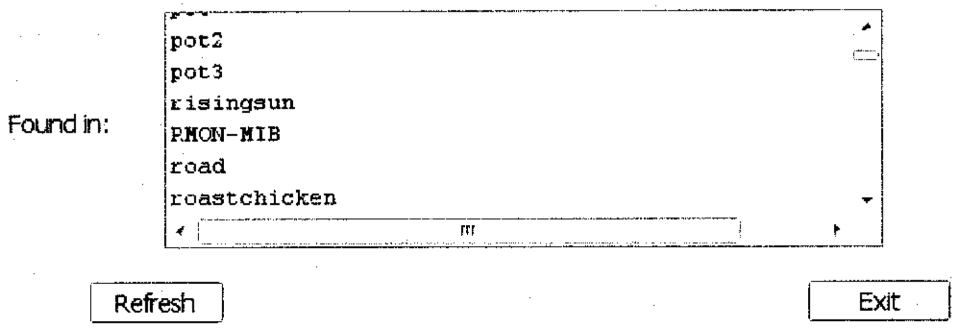


Figure 30: Example of a Hierarchy of Files and Folders

The directory 'Day 4' is basically the root Vector. This root Vector contains Vector '11' and Vector '13'. Vector '11' contains the file object 'popdefs' and Vector '13' contains the file object 'English Project'. That is an illustration of how the use of Vectors can record/mimic the hierarchy of files and folders. Once the search is done, the program will have a complete hierarchy of Vectors and objects. Next, the program just has to display all the Vectors, sub-Vectors and objects as folders, sub-folders and files while preserving their relative position in the hierarchy. This will result in a tree. This tree will appear in a pop-up window. The reason for a pop-up window is that it allows the users to easily resize the window to view the tree at their preferred size. Another advantage is that, they can perform another search and still keep a previously generated tree for reference. The pop-up will not disappear until it is manually closed by the user. It would not be as convenient if the tree were to be displayed in the main window. If that is the case, then generating a new tree will get rid of a previously generated tree. This system does not have that disadvantage.

4.5.4 The List and Tree of Files

Once a search is completed, a list of files containing the keyword or phrase will be displayed in the output text area. The output text area is also utilized as an area to display error messages as was portrayed in section 4.3. The purpose of the list is to provide users with a simple view of all the files that they would be interested in. Sometimes, a user might not be interested in the hierarchy of files or their relative location. They just want to look for a file that contains a keyword or phrase. This simple list is thus sufficient in helping a user to identify a target file. Figure 31 shows an example of this simple list.



```
pot2
pot3
risingsun
RMON-MIB
road
roastchicken
```

Figure 31: Example of a Generated List

If a simple list is not enough to fulfill the needs of a user, they can refer to the generated tree. As mentioned, this tree appears in a pop-up window and displays the files that contain the keyword or phrase along with their relative position in a hierarchy. The program will initially create an unexpanded tree. Users are then free to click on the '+' sign and expand the tree to view its lower level contents. Figure 32 shows an example of a generated tree before a user expands it.

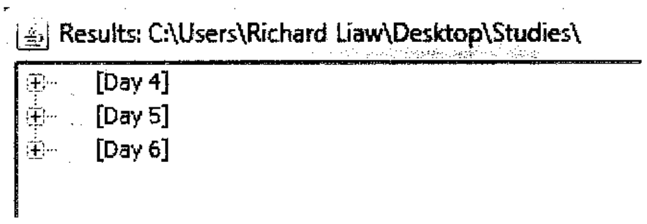
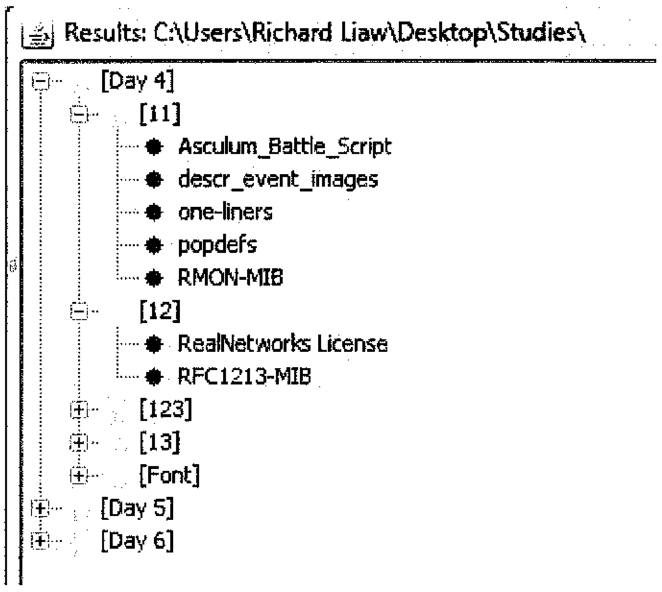


Figure 32: Example of Unexpanded Tree

Figure 33 shows how the tree would look like when users expand a few folders to better view the hierarchy of files and folders.

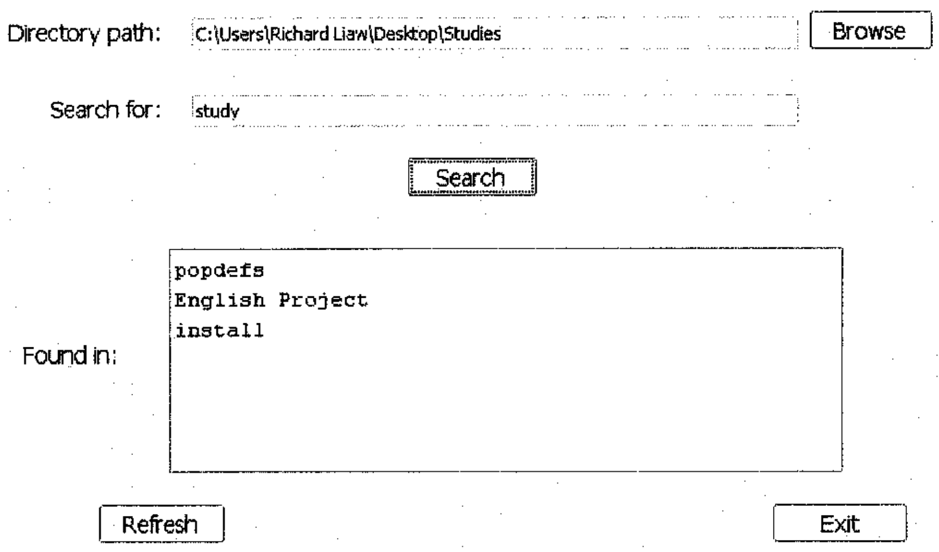

Results: C:\Users\Richard Liaw\Desktop\Studies\

```
[Day 4]
    [11]
        Asculum_Battle_Script
        descr_event_images
        one-liners
        popdefs
        RMON-MIB
    [12]
        RealNetworks License
        RFC1213-MIB
    [123]
    [13]
    [Font]
[Day 5]
[Day 6]
```

Figure 33: Example of Expanded Tree

4.5.5 Refresh and Exit

The next two components of the program are rather straightforward. The 'Refresh' button does not refresh all text fields like what most would expect. When the 'Refresh' button is clicked, the keyword/key phrase text field along with the output text area will be cleared. The directory path text field however, will be left unaltered. The reason for this is that it can be considered to be more practical. Most of the time, a user's next search will be at a similar, if not the same, location. This means that if the directory path's text field is cleared, the user would have to type or browse for the path again. Thus, it is more practical that the directory path is left intact, voiding the need of retyping part of the path or its entirety. To illustrate the workings of the 'Refresh' button, Figure 34 will show how it would look like before refreshing and Figure 35 will show how it would look like after a refresh.

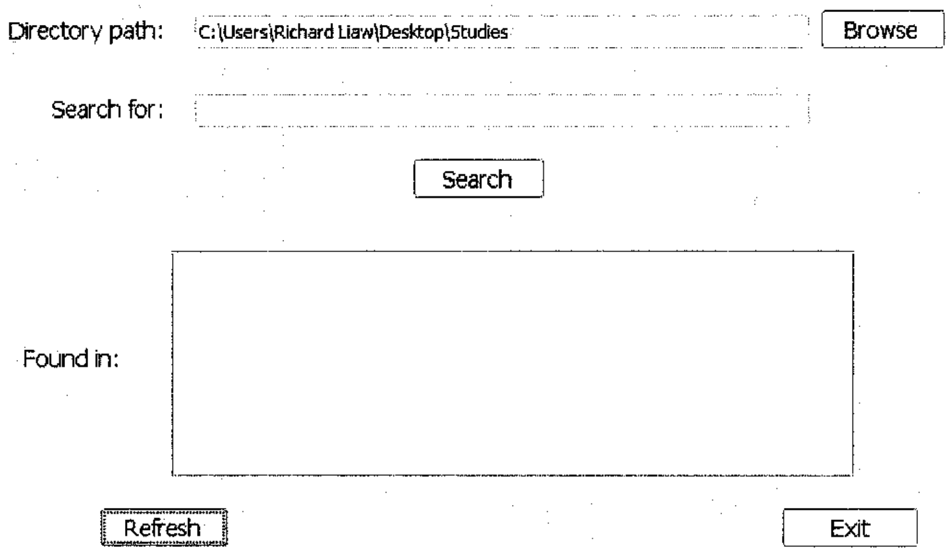

Directory path: C:\Users\Richard Liaw\Desktop\Studies

Search for: study

Found in:

```
popdefs
English Project
install
```

Figure 34: Before Refresh


Directory path: C:\Users\Richard Liaw\Desktop\Studies

Search for:

Found in:

Figure 35: After Refresh

The 'Exit' button simply closes the program. It works in a very similar fashion to the Close ('X') button on the top right corner of a window.

4.5.6 Help: About

The final component is the 'Help' menu under the toolbar. By clicking on 'Help' > 'About', a pop-up window will appear explaining briefly the purpose of the program. This is more of a formality as there are rarely users who would not know what the program is about (its purpose) by just looking at it. Figure 36 shows the location of the 'Help' menu and Figure 37 will show the 'About' window.

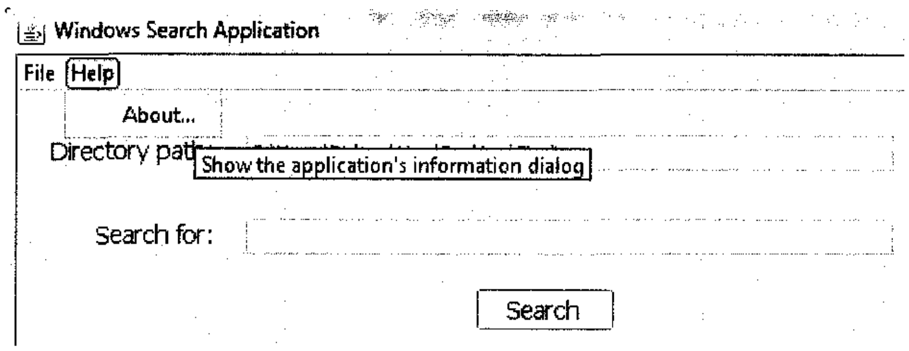
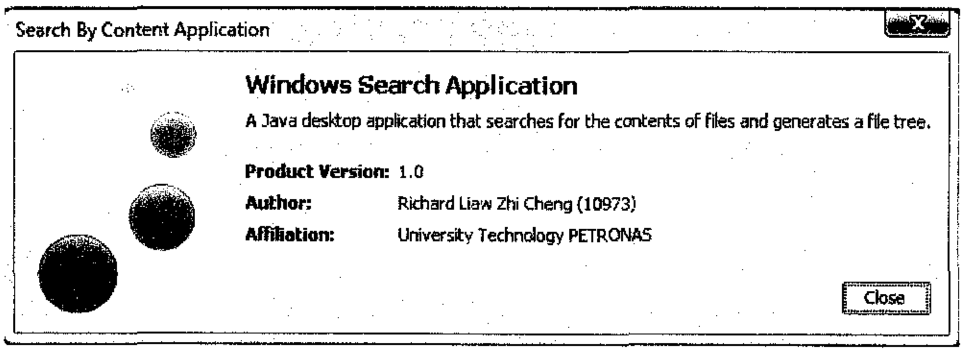

Windows Search Application
File
Help
About...
Directory pat
Show the application's information dialog
Search for:
Search

Figure 36: Help Menu on Toolbar


Search By Content Application

Windows Search Application

A Java desktop application that searches for the contents of files and generates a file tree.

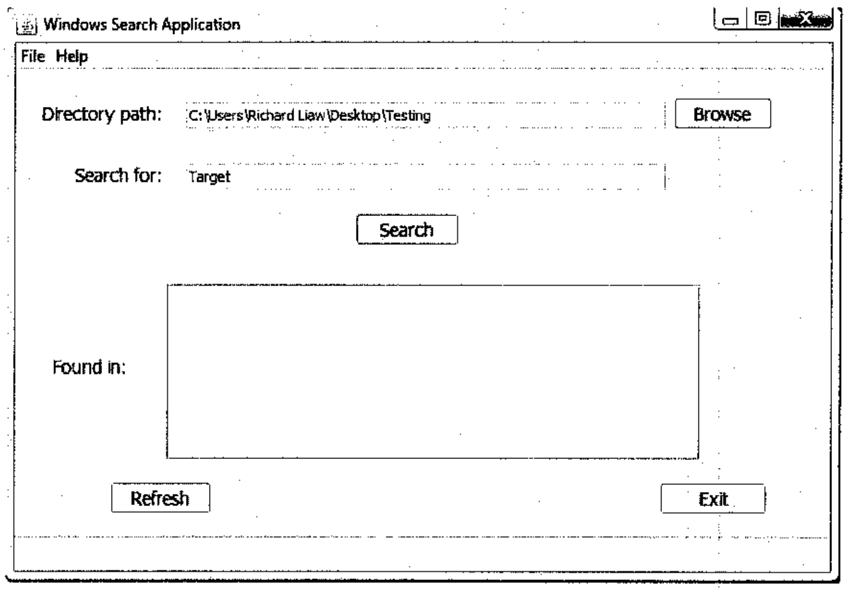
Product Version: 1.0
Author: Richard Liaw Zhi Cheng (10973)
Affiliation: University Technology PETRONAS

Figure 37: 'About' Window

4.6 User Acceptance Testing (UAT)

At the completion of the development of a working system, a BETA test was carried out. Five end users were involved in this test and were instructed to run the search program on their personal computer. Their feedback was obtained.

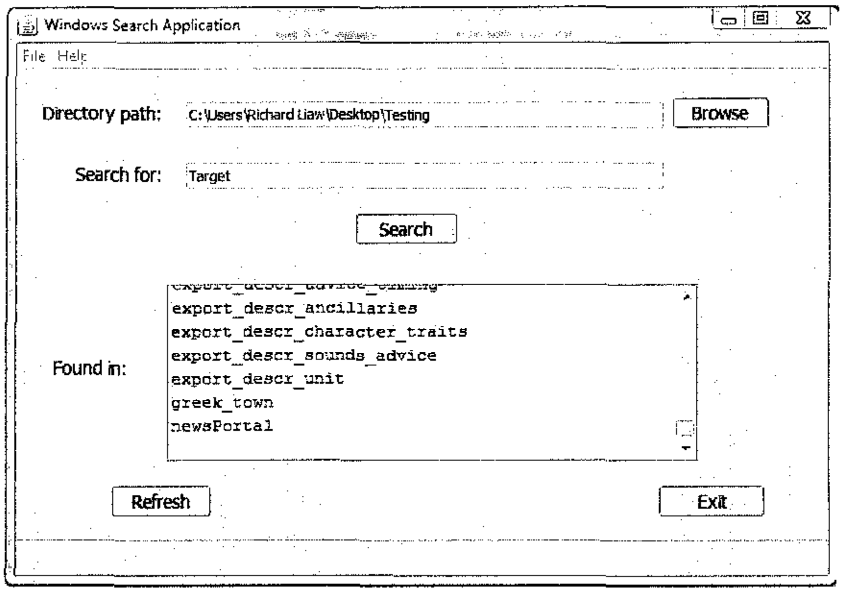
Test case: The keyword to search for is standardized. The keyword is "Target". The directory path that is to be entered to start the search is also standardized. The root/target directory is a folder named "Testing". This is a directory created solely for this test and distributed to all end users. This root directory is populated with levels after levels of subdirectories (maximum 6 levels). There are approximately 187 subdirectories contained within this directory. Figure 38 – 40 below shows an example of how the search will be done and the expected results. The time it takes for the system to complete the search is measured and recorded. 'Time' here is defined as the duration (in seconds) from when the user clicks the 'Search' button to when the Tree Diagram window pops up.



Directory path: C:\Users\Richard Liaw\Desktop\Testing

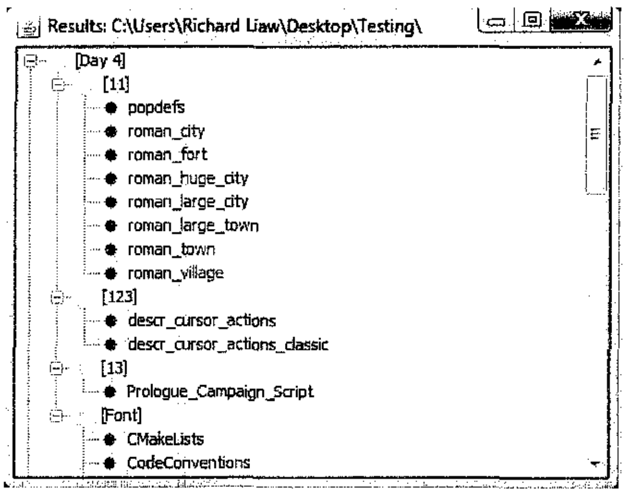
Search for: Target

Figure 38: User Acceptance Test (Initial Stage)



```
export_descr_ancillaries
export_descr_character_traits
export_descr_sounds_advice
export_descr_unit
greek_town
newsPortal
```

Figure 39: User Acceptance Test (Results Generated on Main GUI)



```
[Day 4]
  [11]
    ● popdefs
    ● roman_city
    ● roman_fort
    ● roman_huge_city
    ● roman_large_city
    ● roman_large_town
    ● roman_town
    ● roman_village
  [123]
    ● descr_cursor_actions
    ● descr_cursor_actions_classic
  [13]
    ● Prologue_Campaign_Script
  [Font]
    ● CMakeLists
    ● CodeConventions
```

Figure 40: User Acceptance Test (Tree Displayed on Pop-up Window)

End User 1:

Computer Specifications:

Operating System:	Windows Vista™ Home Premium (6.0, Build 6002) Service Pack 2 (6002.vistasp2_gdr.100608-0458)
System Manufacturer:	Dell Inc.
Processor:	Intel(R) Core(TM) 2 CPU T7200 @ 2.00GHz (2 CPUs) ~2.0GHz
Memory:	2046MB RAM

Search Time: 9 seconds

User feedback:
"A useful program. The tree diagram helps me in navigating to the folder that I was looking for. Maybe you would want to include an 'Advanced Search' where users can enter more criteria for the search? Just an opinion."

End User 2:
Computer Specifications:

Operating System:	Windows XP Professional (5.1, Build 2600) Service Pack 3 (2600.xpsp_sp3_gdr.101209-1647)
System Manufacturer:	n/a
Processor:	Intel(R) Pentium(R) 4 CPU 2.40GHz
Memory:	1016MB RAM

Search Time: 10 seconds

User feedback:
"I don't think I will use a program like this much. Unless of course, the list and tree diagram are more interactive. I want to double click the list or tree and directly open the files."

End User 3:
Computer Specifications:

Operating System:	Windows XP Professional (5.1, Build 2600) Service Pack 3 (2600.xpsp_sp3_gdr.090804-1435)
System Manufacturer:	n/a
Processor:	Intel(R) Pentium(R) 4 CPU 2.40GHz
Memory:	504MB RAM

Search Time: 19 seconds

User feedback:

"Kind of slow for me. Maybe an option to choose whether I want the tree diagram to be generated or not. If I disable it, wouldn't the search be faster?"

End User 4:

Computer Specifications:

Operating System:	Windows XP Professional (5.1, Build 2600) Service Pack 3 (2600.xpsp_sp3_gdr.080814-1236)
System Manufacturer:	n/a
Processor:	Intel(R) Core(TM) 2 Duo CPU E8400 @ 3.00GHz (2 CPUs)
Memory:	1980MB RAM

Search Time: 8 seconds

User feedback:

"A decent program. I run it from my USB drive and it still searches as fast. How about designing one for the Mac OS?"

End User 5:

Computer Specifications:

Operating System:	Windows 7 Professional 32-bit (6.1, Build 7600) (7600.win7_rtm.090713-1255)
System Manufacturer:	Hewlett-Packard
Processor:	Intel(R) Atom(TM) CPU N450 @ 1.66GHz (2 CPUs), ~1.7GHz
Memory:	1024MB RAM

Search Time: 11 seconds

User feedback:

"Can be better if the tree is interactive."

Test Findings:

The average search time is: <u>11.4 seconds</u>

The system is found to be overall satisfactory. However, there are improvements that can be made especially in terms of user interaction with the results. Both the resulting list and tree diagram should be more interactive. Users should be able to double click on files to open them or even drag and drop to move and rearrange files.

4.7 The Problem of Mind Mapping

In this chapter and Chapter 2, we have already discussed the potential problems that would arise if the mind mapping technique were to be used as a search method. To summarize the points discussed, excessive grouping would cause the confusion of the organization of files and folders and too many clicks required by users to reach an intended file. Mind mapping technique might be a powerful visual representation method but when applied to files and folders search in Windows, it might actually backfire. If an overly cluttered and lengthy mind map is produced, users might eventually be fed up with the system and refuse to use it. When this happens, the initial objectives will never be fulfilled. Besides that, generating a large mind map might require too much processing power or, in a more measurable unit: processing time. When using a search method, users would prefer a shorter response time. The Windows 'Search' function does a considerably good job in this aspect. Thus, to match the current system, the newly proposed search system should also incorporate shorter search durations. However, like I mentioned, generating a large mind map would require more processing power. The long wait, again, will leave negative perceptions and impressions on the users. If users do not like the system, they will never use it. There are cases when small mind maps are generated. This is when users specify a directory that only contains a few descendents. Although it might work well for this case, we cannot ignore the fact that this system is impractical about half the time.

There is also a work-around that can be proposed. If the search program is limited to work with only small directories, the problem of generating large mind maps can be avoided. This is to say, there are a limited amount of descendent nodes that the mind map can generate. When this amount is exceeded, an error message would be displayed and a new search can be initiated. This method can be an alternative search method but its range of applications is rather limited. There are but only a narrow selection of circumstances that would prove this work-around useful.

Also, another work-around would be the idea to combine the search-by-content method and the mind mapping technique. In other words, instead of displaying the tree, a mind map is displayed in its place. Of course, this can be a probable new research topic and an improvement on my currently proposed search program but for now, let us look at an obvious implication. The mind mapping technique is definitely a new idea to be incorporated into a search program. When introducing something new, it is always accompanied by certain risks. These risks include the problem of unfamiliarity. If a user is unfamiliar with a new program or application, there is a chance that he/she might refuse to continue using the program. The mind map can be an interesting new program but users might be confused on how the generated mind map actually represents the files and folders. In this case, a familiar tree would suffice in meeting the users' needs. Almost everyone who uses the Windows operating system would know what a tree depicts and represents. This shows how a tree can actually out-compete the mind mapping technique.

In other words, the benefits offered by a mind mapping search system are insufficient to justify the time and effort that is to be put in to develop it. Thus, an alternative method of Search-By-Content is proposed.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5. CONCLUSION AND RECOMMENDATION

5.1 Conclusion and Recommendation

The Search-By-Content method has the ability to produce a faster search time. In other words, it has a shorter response time. This is because the program only searches for files containing the keyword or phrase. The system will also generate just a simple list of file names. The winning point of this search method is that it provides a comprehensive 'tree view', which the Windows Search function does not provide. There might be sharewares or open-source applications out there that have similar functionalities but none are as lightweight or have been made an official search application in Windows. My recommendation is that a system like this is ultimately integrated into the Windows 'Search' function. Not only will this new system work as an alternative search method, it will also become a complementary function to what is available in Windows now.

As already discussed, this method is concern with generating a tree instead of a mind map. However, they really are not all that different. Both allow users to better remember the location of files and provide a hierarchical view of files and folders. We have also discussed in the previous section that a tree can even be more suitable than a mind map as it prevails in terms of user familiarity.

The Search-By-Content system is useful in various situations. There might be the case when users want to search for files relating to a certain topic but could not remember their file names. Also, there are times when users would want to search for certain contents but there are too many files with similar names stored arbitrary in a folder. Besides that, there are times when users would want to know which document contains certain contents. Another advantage of the developed system is that it is lightweight. Users can just store this search program on a USB drive and plug into any computer that has the Java Runtime Environment (JRE) installed to run/use it to search for files. It is that simple. All these situations would call for the need of an alternative search method: the Search-By-Content system. There are just numerous more situations that the Search-By-Content system would prove useful.

Although I have started off favoring and researching the application of the mind mapping technique in search programs, my resulting recommendation is on the search-by-content method. This is because, throughout the study, the impracticality of a mind mapping search system has become clearer and in the midst of searching for an alternative, one method emerged as a probable and capable alternative.

Besides that, the recommendation is still in line with the prior objectives of this study. To recap, the objectives were:

1. To develop a search program that could serve as an alternative or even a complement to the Windows 'Search' program.
2. To develop a system that provides the basic functionalities of a search program and a consistent interface with Windows applications to improve user familiarity.
3. Ultimately, study on the potential of the mind mapping technique in search programs and determine whether it is best suited as an alternative search program. If not, propose a better alternative.

The Search-By-Content system is able to serve as an alternative search program. This fulfills the first objective. The Search-By-Content system will be generated as a Windows application that incorporates the general graphical user interface of other Windows application. This fulfills the second objective. Finally, we have been discussing intensively about the mind mapping search technique in Chapter 2, 4 and 5. The results were not in favor of the initially proposed technique so a new method is proposed: the Search-By-Content system. This fulfills the third objective.

In conclusion, this endeavor is more of a study on a valuable alternative search method. A basic prototype of the Search-By-Content system is developed to complement the findings of this study as well as to demonstrate its basic features.

REFERENCES

Buzan, T. (2000). *The Mind Map Book: Millennium Edition.* Bath, Great Britain: BBC Worldwide Limited.

Cogmap. (2010). Coca-Cola Company. Retrieved from http://www.cogmap.com/chart/coca-cola-company

Dennis, A., Wixom, B. H. & Tegarden, D. (2005). *Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach* (2nd ed.). Hoboken, NJ, USA: Wiley.

Elucidata. (n.d.). *The Software Development Life Cycle (SDLC)* (Version 1.0d). Retrieved from http://www.elucidata.com/refs/sdlc.pdf

Hearst, M.A. & Rosner, D. (n.d.). Tag Clouds: Data Analysis Tool or Social Signaller? Retrieved from School of Information, University of California, Berkeley website: http://flamenco.berkeley.edu/papers/tagclouds.pdf

Kaser, O. & Lemire, D. (n.d.). *Tag-Cloud Drawing: Algorithms for Cloud Visualization.* Retrieved from http://www2007.org/workshops/paper_12.pdf

Lau, B. (1998). *Mind Mapping.* Retrieved from Learning Strategies Corporation, Minneapolis, Minnesota, USA website: http://www.caribbeantheology.net/Study%20Habits/Mine%20mapping.PDF

Marumushi. (n.d.). *Projects / Newsmap.* Retrieved from http://marumushi.com/projects/newsmap

Maryland. (2008). *Systems Development Life Cycle (SDLC).* Retrieved from Department of Information Technology, Maryland website: http://doit.maryland.gov/policies/Documents/sdlc/sdlcvol2.pdf

Phelps, J. (2006). *Folksonomy and Web 2.0 - Power to the People.* Retrieved from Division of Information Technology, University of Wisconsin, Madison: http://net.educause.edu/ir/library/pdf/MWR0638.pdf

Serrat, O. (2009). *Drawing Mind Maps.* Retrieved from Knowledge Solutions, Asian Development Bank website: http://www.adb.org/documents/information/knowledge-solutions/drawing-mind-maps.pdf

The Facilitators (1998). *The Facilitator: Mind Mapping* (Issue 22). Retrieved from The Facilitator website: http://www.thefacilitator.com/htdocs/Mind%20Mapping.pdf

APPENDIX A: Survey on File Content Search System

FYP Survey on File Content Search System

1. How would you normally search for files stored in your computer?

____

____

2. How would you search for files if you have totally forgotten its name and location?

____

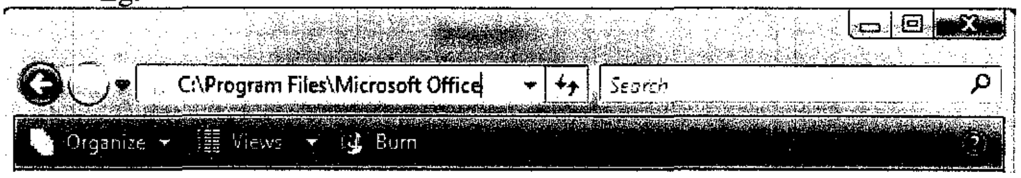
____

3. Which search method would you prefer to use? *(Please tick only one circle)*

○ 1. Use Windows 'Search' function and search by keyword

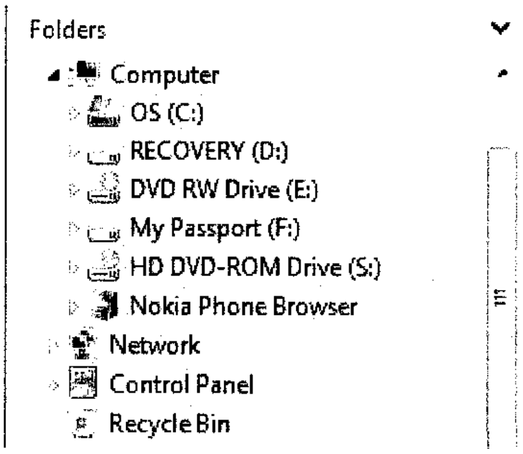
○ 2. Search by typing out directory paths

Eg:



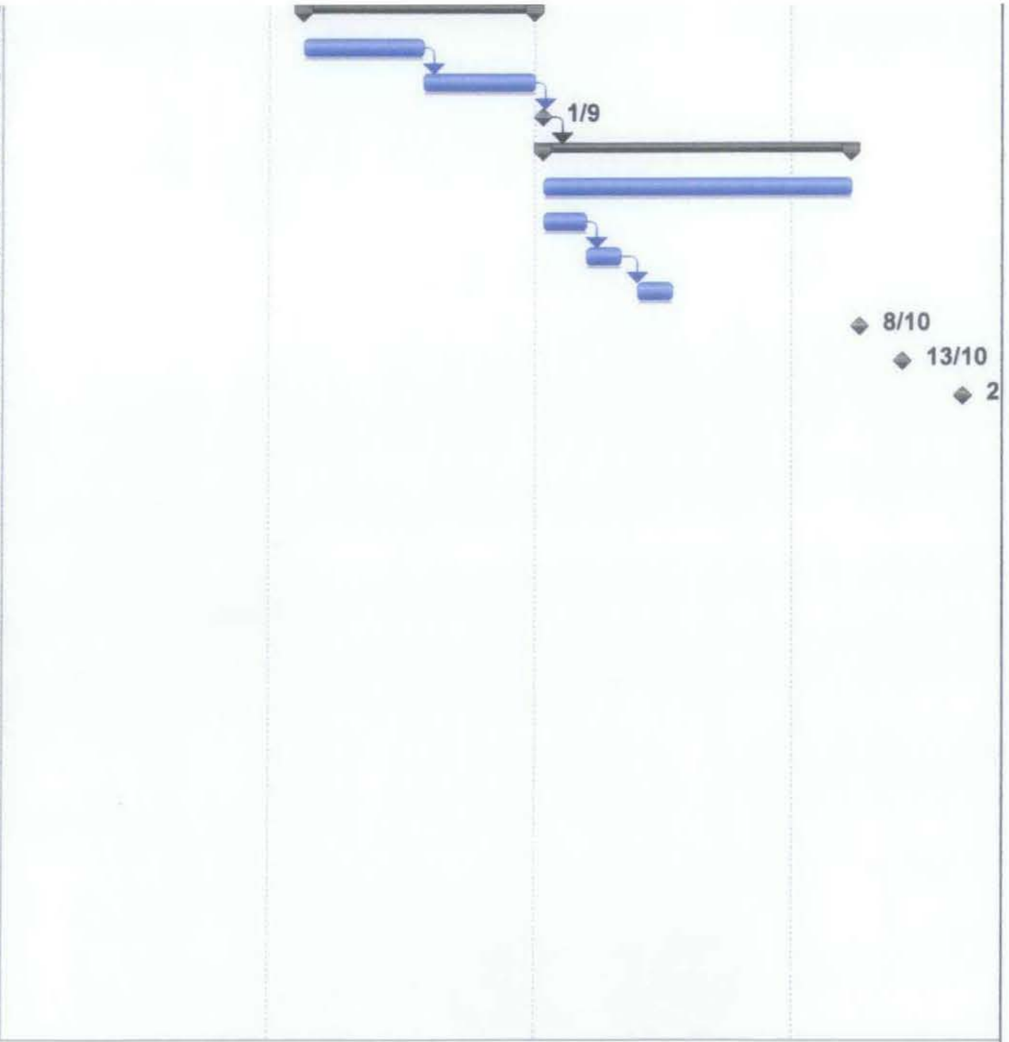
○ 3. Locate file through Windows 'Explore' view

Eg:



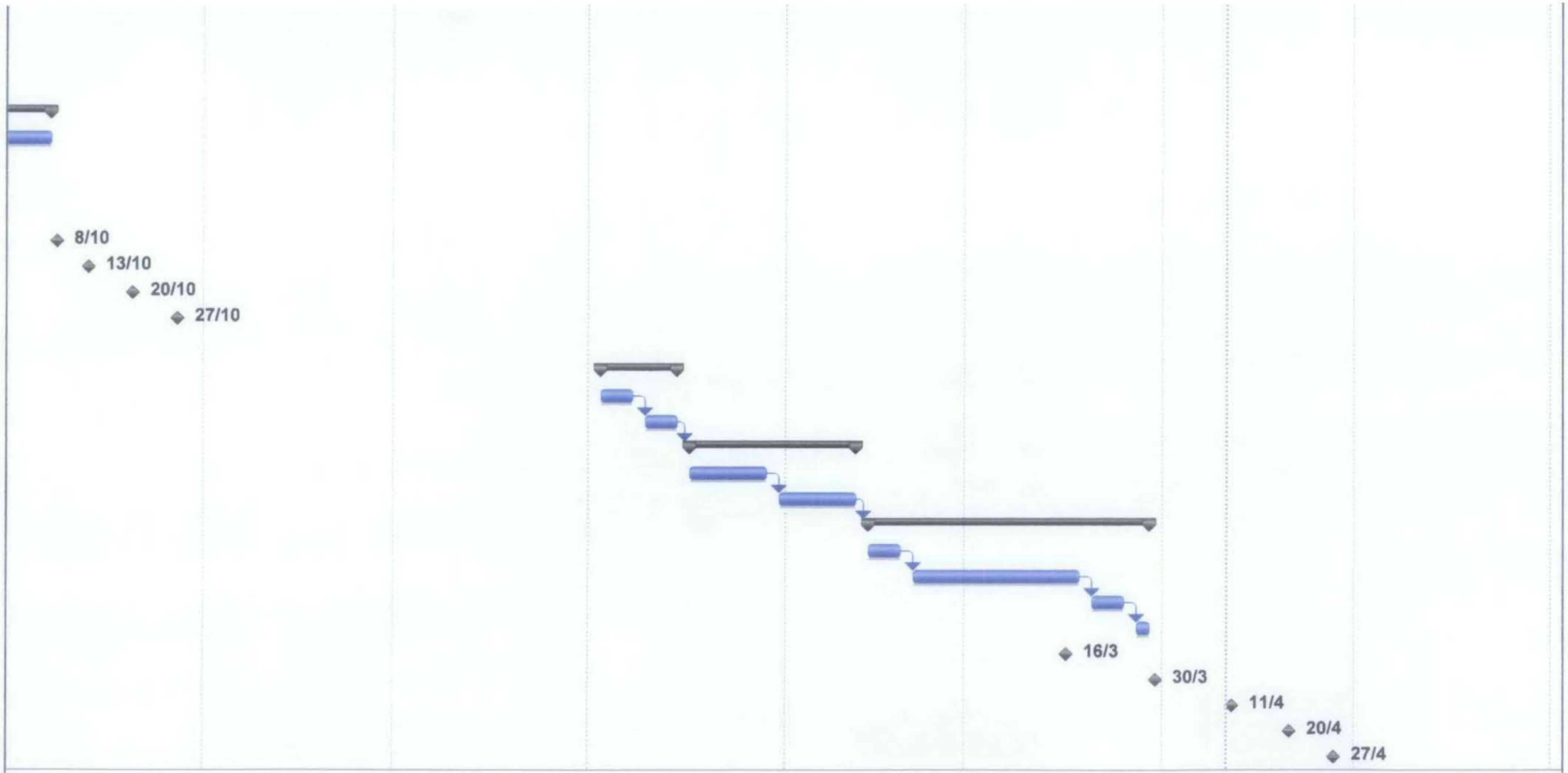
APPENDIX B: File Content Search System Project Gantt Chart

2		**Preliminary Research Work**	**19 days**	**Thu 5/8/10**	**Tue 31/8/10**
3	▦	Initial Project Research	10 days	Thu 5/8/10	Wed 18/8/10
4	▦	Prepare Preliminary Report	9 days	Thu 19/8/10	Tue 31/8/10
5	▦	Preliminary Report Submission	1 day	Wed 1/9/10	Wed 1/9/10
6		**Project Planning**	**26 days**	**Thu 2/9/10**	**Thu 7/10/10**
7	▦	Prepare Project Report	26 days	Thu 2/9/10	Thu 7/10/10
8	▦	Define Project Objectives	3 days	Thu 2/9/10	Mon 6/9/10
9	▦	Create Development Schedule	4 days	Tue 7/9/10	Fri 10/9/10
10	▦	Perform Feasibility Study	4 days	Mon 13/9/10	Thu 16/9/10
11	▦	Progress Report Submission	1 day	Fri 8/10/10	Fri 8/10/10
12		Seminar	1 day	Wed 13/10/10	Wed 13/10/10
13	▦	Interim Report Submission	1 day	Wed 20/10/10	Wed 20/10/10
14	▦	Oral Presentation	1 day	Wed 27/10/10	Wed 27/10/10
15					
16		**Project Analysis**	**10 days?**	**Mon 3/1/11**	**Fri 14/1/11**
17	▦	Requirements Gathering	5 days?	Mon 3/1/11	Fri 7/1/11
18	▦	Requirements Analysis	5 days?	Mon 10/1/11	Fri 14/1/11
19		**Project Design**	**20 days?**	**Mon 17/1/11**	**Fri 11/2/11**
20	▦	Define Project Specification	10 days?	Mon 17/1/11	Fri 28/1/11
21	▦	Design Entire System	10 days?	Mon 31/1/11	Fri 11/2/11
22		**Project Implementation**	**32 days?**	**Mon 14/2/11**	**Tue 29/3/11**
23	▦	Setup Environment	5 days?	Mon 14/2/11	Fri 18/2/11
24	▦	Develop/Code the program	20 days?	Mon 21/2/11	Fri 18/3/11
25	▦	Test Program	5 days?	Mon 21/3/11	Fri 25/3/11
26	▦	Document & Monitor Results	2 days?	Mon 28/3/11	Tue 29/3/11
27	▦	Progress Report Submission	1 day	Wed 16/3/11	Wed 16/3/11
28	▦	Pre-EDX	1 day	Wed 30/3/11	Wed 30/3/11
29	▦	Dissertation Submission	1 day	Mon 11/4/11	Mon 11/4/11
30	▦	Viva	1 day	Wed 20/4/11	Wed 20/4/11
31	▦	Final Dissertation Submission	1 day	Wed 27/4/11	Wed 27/4/11



Project: FYP Gantt Chart
Date: Mon 11/4/11

Task	▬	Milestone	◆	External Tasks	▬
Split	┈┈┈	Summary	▬	External Milestone	◇
Progress	▬	Project Summary	▬	Deadline	⇩



Project: FYP Gantt Chart Date: Mon 11/4/11	Task		Milestone	◆	External Tasks	
	Split		Summary		External Milestone	◇
	Progress		Project Summary		Deadline	⇩