

**WIN4TECH**  
**A FORENSIC TOOLKIT**



By

NC Chanda Gul

NC Ayesha Aslam

NC Saneeha Khalid

PC Umer Khan

Submitted to the Faculty of Computer Science

National University of Sciences and Technology, Rawalpindi in partial fulfillment

for the requirements of a B.E Degree In Computer Software Engineering

AUGUST 2009

## APPROVAL CERTIFICATE

I, Lec. Ahmad Raza Cheema declares that the project has been completed and report has been reviewed. The report is found to be correct in all respects and does not contain any copied/plagiarized material.

-----

Signature (Supervisor)

-----

Date

## **ABSTRACT**

Computer forensics deals with the ways and methodologies for exploring a computer system for the purpose of relating evidence to a particular event. The goal of our project is to develop a toolkit-Win4tech for investigating a compromised/ misused computer system. Win4Tech uses File system, System Registry, System Processes, Opened Files, User Accounts and USB Records for extracting evidence. The tool will be cost-effective and usable as compared to existing forensic toolkits.

## **DECLARATION**

No portion of the work presented in this dissertation has been submitted in support of any other award or qualification either at this institution or elsewhere.

## **DEDICATION**

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation, a work of  
this magnitude would have been impossible.

## ACKNOWLEDGEMENTS

*We bow our heads to offer our humblest thanks to Allah, Almighty, the Most Gracious, the Most Compassionate, Who bestowed us with the vigor and zeal to achieve this important task of our life, despite of losing heart at many turns.*

Firstly, we are greatly indebted to convey our heart-felt gratitude to our supervisors, Col Naveed Sarfraz Khattak(CS Dept) , Lec Ahmad Raza Cheema (IS Dept) and Asst Prof Fauzan Mirza (SEECs, NUST) under whose kind supervision this toolkit is completed. We are thankful and appreciative of their full co-operation, valuable guidance, and constant encouragement due to which we are able to complete our task in the given duration of time.

We give the credit of the completion of this work to our parents, seniors and teachers who gave us support each time we lost hope and felt dejected.

We must acknowledge and admire the CS (Computer Science) Department's all staff members' for providing us the best possible facilities due to which we are able to achieve our goals.

A word of thanks to Military College of Signals as it had been our foundation and made us capable to undertake the project.

And finally to conclude, we must utter thanks to ourselves for "at last", achieving an important milestone of our lives.

# TABLE OF CONTENTS

## List of Illustrations

## List of Tables

<b>1. Introduction.....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 Preface.....	1
1.3 Motivation .....	2
1.4 Problem Statement .....	2
1.5 Project Scope .....	3
1.6 Objectives .....	3
1.7 Project Beneficiaries .....	3
1.8 Project Title.....	3
1.9 Project Logo.....	3
1.10 Work breakdown Structure .....	4
1.11 Deliverables .....	4
1.12 Conclusion .....	5
<b>2. Literature Review .....</b>	<b>6</b>
2.1 Introduction.....	6
2.2 Background.....	6
2.3 File System (NTFS) .....	6
2.4 System Registry .....	9
2.5 Running Processes .....	10
<b>3. System Requirements Specification (SRS) .....</b>	<b>12</b>
3.1 Introduction.....	12
3.2 Intended Audience and Reading Suggestions.....	12
3.3 Business Context.....	12
3.4 Product Perspective.....	12
3.5 Product Features.....	13
3.6 User Classes and Characteristics .....	13
3.7 Operating Environment.....	13
3.8 Design and Implementation Constraints.....	13
3.9 User Documentation .....	14
3.10 Assumptions and Dependencies .....	14
3.11 Functional Requirements .....	14
3.12 User Interface Requirements.....	24
3.13 Other Nonfunctional Requirement.....	25

<b>4. Project Design.....</b>	<b>27</b>
4.1 Introduction.....	27
4.2 System Overview .....	27
4.3 System Architectural Design .....	28
4.4 Decomposition Design.....	29
4.5 Data Design .....	33
4.6 Data Description .....	38
4.7 Component Design .....	39
4.8 Human Interface Design.....	40
4.9 Graphical Model.....	48
4.10 UML Based Design.....	50
<b>5. Implementation Details .....</b>	<b>57</b>
5.1 Introduction.....	57
5.2 Component Implementation Details .....	57
<b>6. Testing and Results.....</b>	<b>66</b>
6.1 Introduction.....	66
6.2 Testing Schedule.....	67
6.3 Test Cases .....	68
6.4 Test Summary Report .....	72
<b>7. Conclusion and Future Work .....</b>	<b>73</b>
7.1 Conclusion .....	73
7.2 Future Work.....	73
<b>Annexure A Software Development Plan.....</b>	<b>74</b>
<b>Annexure B User Manual .....</b>	<b>87</b>
<b>Annexure C Bibliography.....</b>	<b>103</b>



## LIST OF ILLUSTRATIONS

Figure #	Caption	Page #
1.1	Graph showing Increasing Cyber Crime Incidents	2
1.2	Project Logo	3
1.3	Work Breakdown Structure of Win4Tech	4
4.1	Architectural Design of Win4Tech	28
4.2	Architectural Design of File Tracker	29
4.3	Architectural Design of Reg Tracker	30
4.4	Architectural Design of Activity Viewer	30
4.5	Architectural Design of User Accounts History	31
4.6	Architectural Design of Current Process Viewer	31
4.7	Architectural Design of USB Tracker	32
4.8	Architectural Design of <i>Rens</i> Snapshotter	32
4.9	Data Flow Diagram of Win4Tech	33
4.10	Data Flow Diagram of File Tracker	34
4.11	Data Flow Diagram of Reg Tracker	35
4.12	Data Flow Diagram of User Accounts History	35

4.13	Data Flow Diagram of Activity Viewer	36
4.14	Data Flow Diagram of Current Process Viewer	37
4.15	Data Flow Diagram of <i>Rens</i> Snapshoter	37
4.16	Data Flow Diagram of USB Tracker	38
4.17	Component Design of Win4Tech	39
4.18	Structural Model of Win4Tech	48
4.19	Object Model of Win4Tech	49
4.20	Use Case Diagram of Win4Tech	50
4.21	Class Diagram of Win4Tech	51
4.22	Collaboration Diagram of Win4Tech	52
4.23	Sequence Diagram of Win4Tech	53
A-1	Evolutionary Software Process Model	79
A-2	Organization Chart of the Executive Staff	80
A-3	Project Charter	83

## LIST OF TABLES

<b>Table #</b>	<b>Caption</b>	<b>Page #</b>
6.1	Team Members' Responsibilities	67
6.2	Testing Schedule	67
6.3	Test case # 1	68
6.4	Test case # 2	69
6.5	Test case # 3	69
6.6	Test case # 4	70
6.7	Test case # 5	71
6.8	Test case # 6	71
6.9	Test case # 7	72
A-1	List of Project Deliverables	77
A-2	Project Responsibilities Chart	81
A-3	Staffing Plan Chart	85

## LIST OF ABBREVIATIONS

Abbreviation	Description
Win4Tech	Windows Forensic Technology
NTFS	Windows NT File System
MFT	Master File Table
FAT	File Allocation Table
MAC	Modified Access Creation Time
DLL	Dynamic Link Libraries
URL	Universal Resource Locator
SID	Security Identifier
API	Application Programming Interface

# MANUAL

# WIN4TECH








### **Purpose of Win4Tech:**

Computer Forensics is the application of methods and techniques for investigating a computer system to reveal criminal activities in a court of law. Criminal activities like hacking, information theft, virus attacks and malicious software productions are growing rapidly. To deal with these threats, it is highly important to secure information and perform collection and analysis of evidence from victimized systems.

### **Overview of Win4Tech:**



Major functionality of our product will be inspecting users' activity.

Key features are:

-  Examining files which have been viewed previously by any user on the system.
-  Traversing Windows Registry and locating sub keys for every root key and returning the Last Write Time of any key value.
-  Searching the recent activities on a system which includes recent documents, windows searched items, programs that execute on system startup and run commands.
-  Providing record of Created User Accounts on the system.
-  Providing list of currently running processes and associated DLLs (Dynamic Link Library). It will also provide list of files opened by these processes.
-  Recording details of previously and currently connected USBs
-  Providing detailed information about the objects (e.g. text boxes, list boxes, status bars, etc) residing in currently opened windows.

### **Intended Audience:**

The *Win4tech manual* is written for law enforcement and corporate security professionals with the following competencies:

-  Basic knowledge of and training in forensic policies and procedures
-  Basic knowledge of and experience with personal computers
- Familiarity with the fundamentals of collecting digital evidence

- 🔍 Experience with case studies and reports
- 🔍 Familiarity with the Microsoft Windows environment.

### **System Requirements:**

The underlying platform for Win4Tech will be *Microsoft Windows XP* with 32 bit architecture. Our product will function on a live system. *Dot Net framework 2.0* will be required to make it operative.

### **Handling Evidence:**

Computer forensics involves the acquisition, preservation, analysis, and presentation of computer evidence. This type of evidence is fragile and can easily, even inadvertently, altered, destroyed, or rendered inadmissible as evidence. Computer evidence must be properly obtained, preserved, and analyzed to be accepted as reliable and valid in a court of law. To preserve the integrity of case evidence, forensic investigators do not work on the original files themselves. Instead, they create an exact replica of the files and work on this image to ensure that the original files remain intact.

### **Installation Process:**

#### **MAIN INTERFACE OF WIN4TECH:**

The following is the main graphical user interface of Win4Tech,



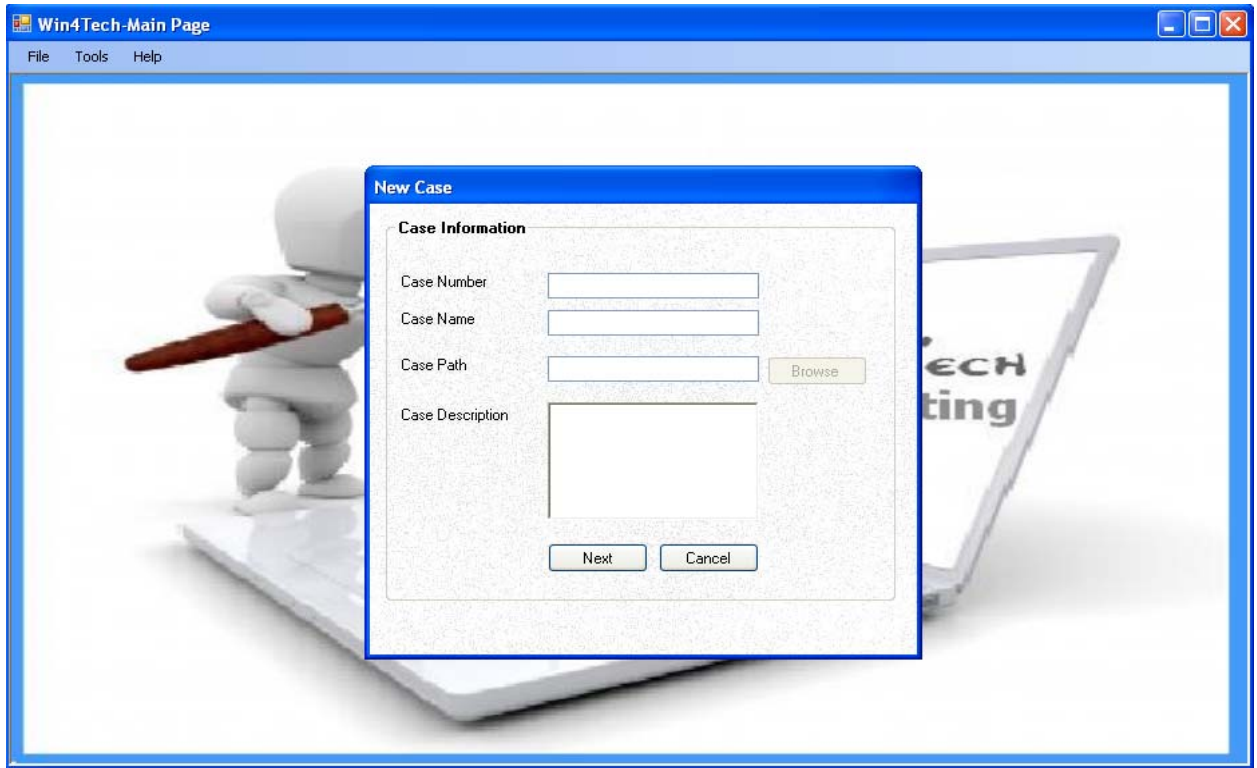
### **STARTING A CASE:**

You access the New Case Wizard by selecting **File > New Case**. If this is your first time opening Win4Tech or if you have chosen to always display the FTK Startup screen, select **New Case**



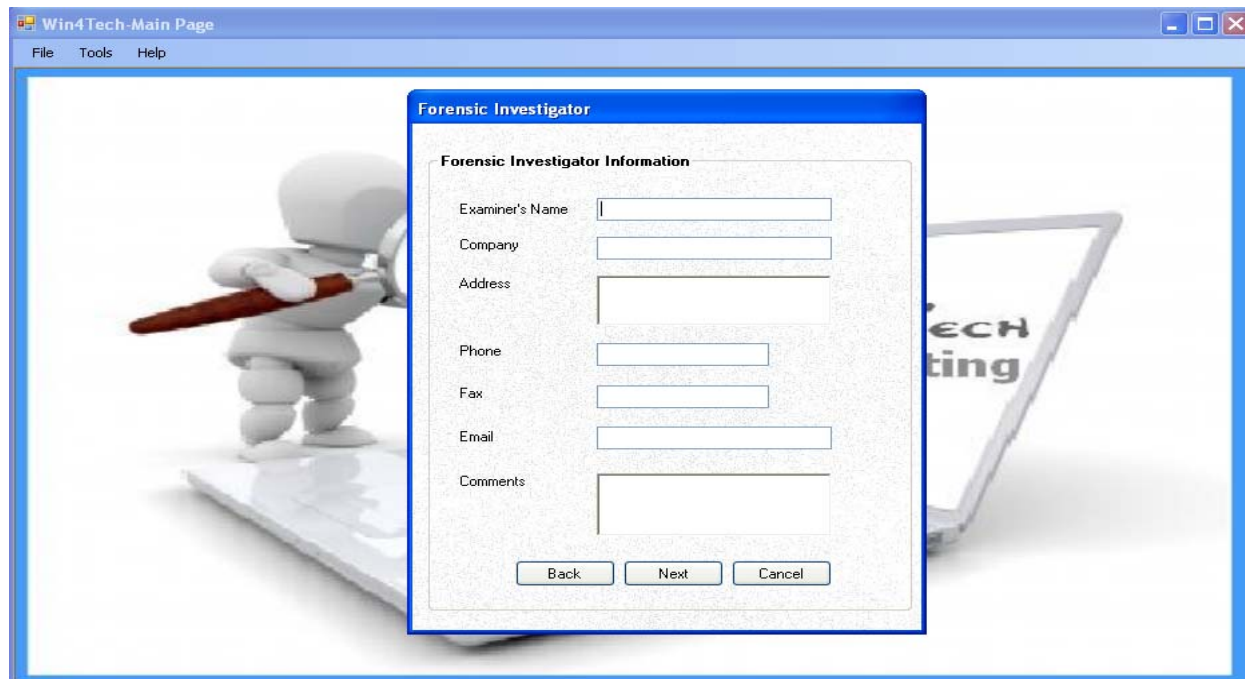
To start a new case, you must complete the following steps.

- 1) Enter the case information.
- 2) Enter the case name.
- 3) Select the Case Path by clicking on the browse button.
- 4) Enter the case description.



5) Click the **Next** button so, that you can go to the next form.

### **FORENSIC INVESTIGATOR'S DETAILS:**



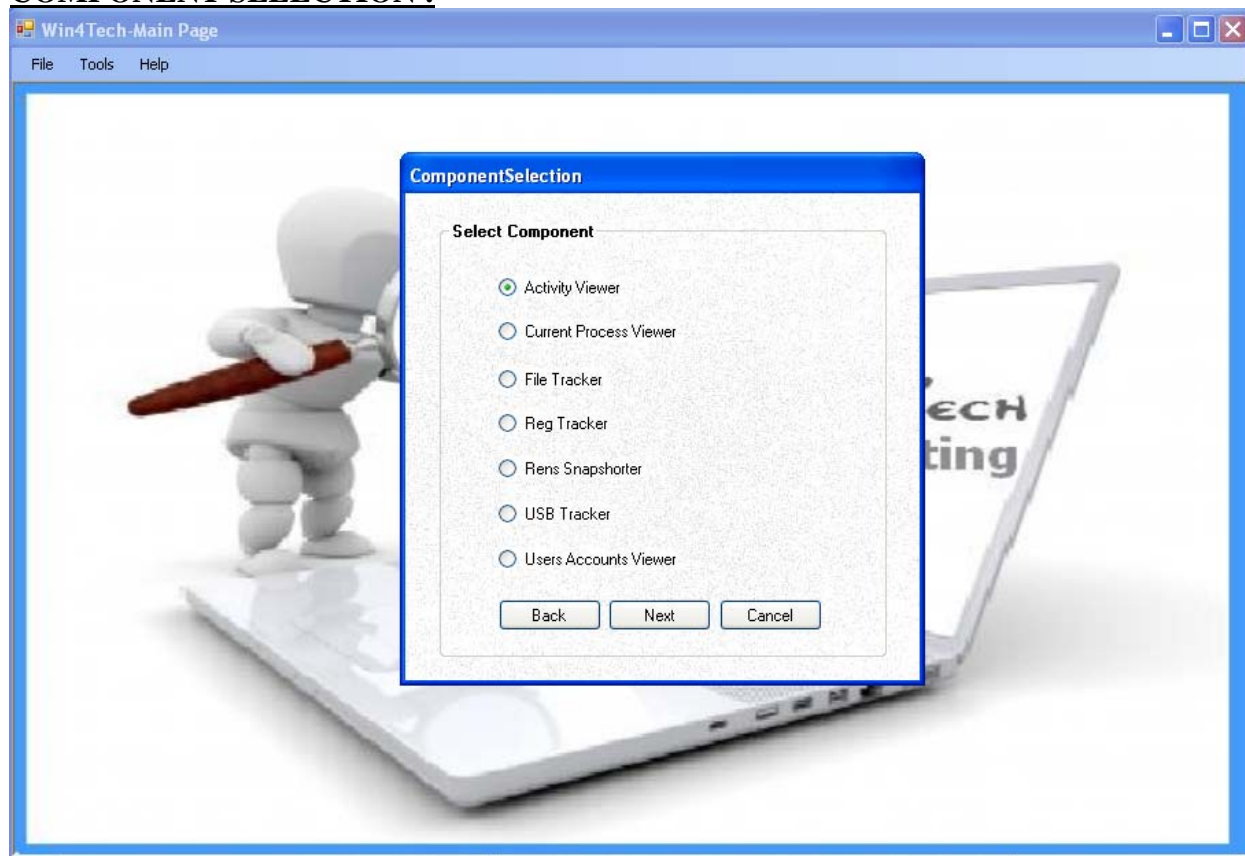
1. Enter your name in the **Examiner's field** as it is mandatory otherwise you would not be able to proceed next.



2. ( **Optional** ) Enter the Company's name that you work for in the **Compnay's field**.
3. ( **Optional** ) Enter the address in the **Address field**.
4. ( **Optional** ) Enter the Phone Number in the **Phone's field**.
5. ( **Optional** ) Enter the fax Number in the **Fax field**.
6. ( **Optional** ) Enter the email address.
7. ( **Optional** ) Write the comments.

Click the NEXT button.

**COMPONENT SELECTION :**



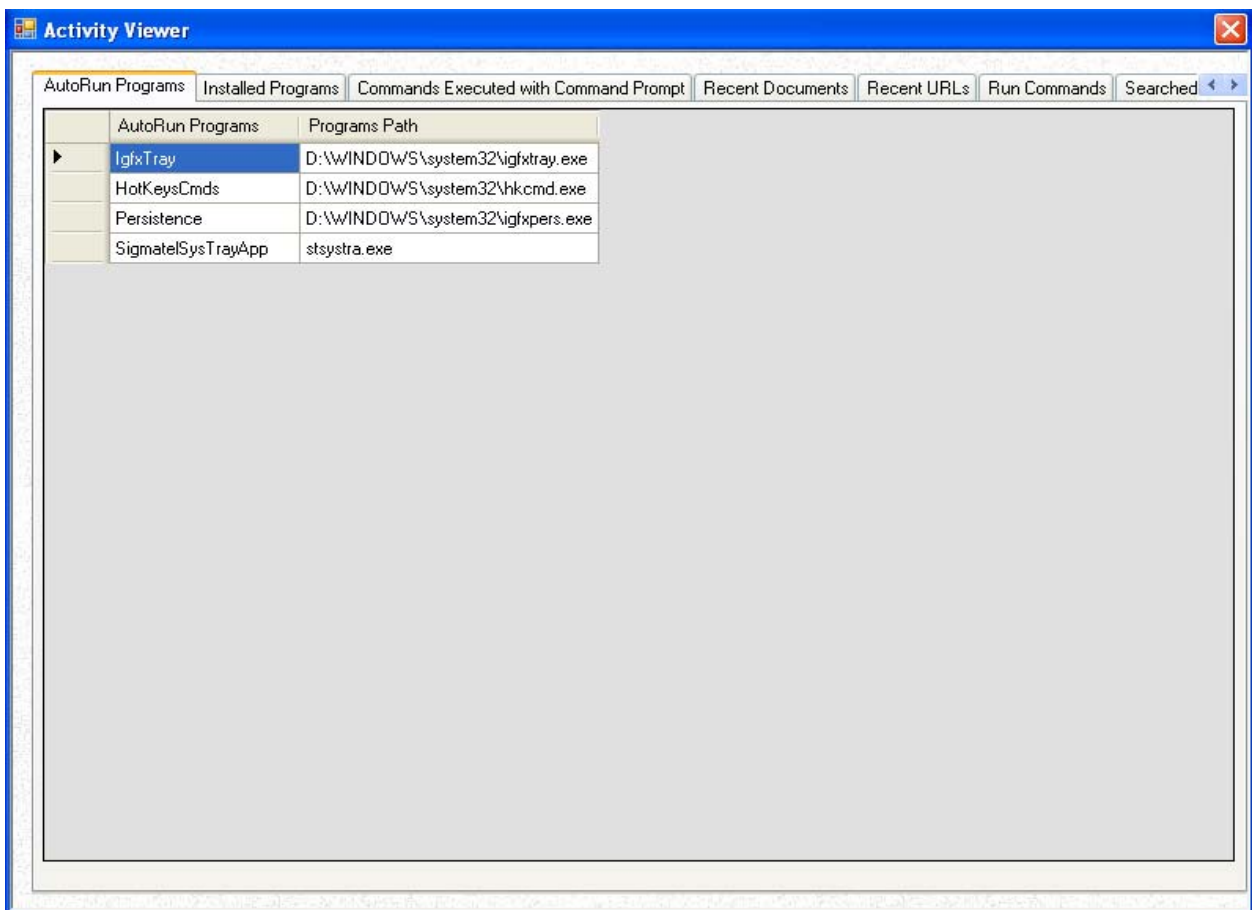
Select the component you want to chose from the mentioned components.Lets suppose you have chosen **Activity Viewer**.Click Next.

**ACTIVTY VIEWER:**

This component will tell you a quite alot of evidences that are :

1. AutoRun Programs
2. Installed Programs

3. Commands executed with command prompt
4. Recent Documents
5. Recent URLs.
6. Run Commands.
7. Search Commands.
8. Search Words and Phrases.
9. Start-Up Programs.

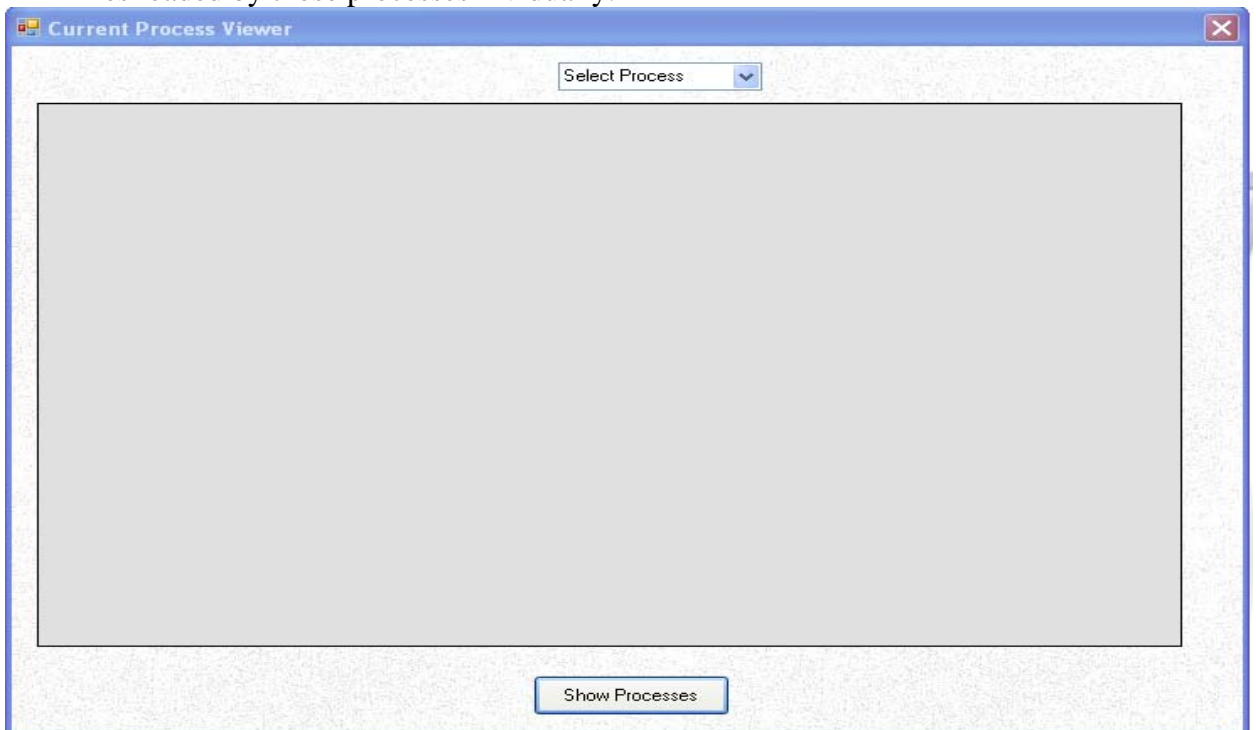


If you click on the tabs, you can see the generated reports that you require. You can close this module and simply select another component .

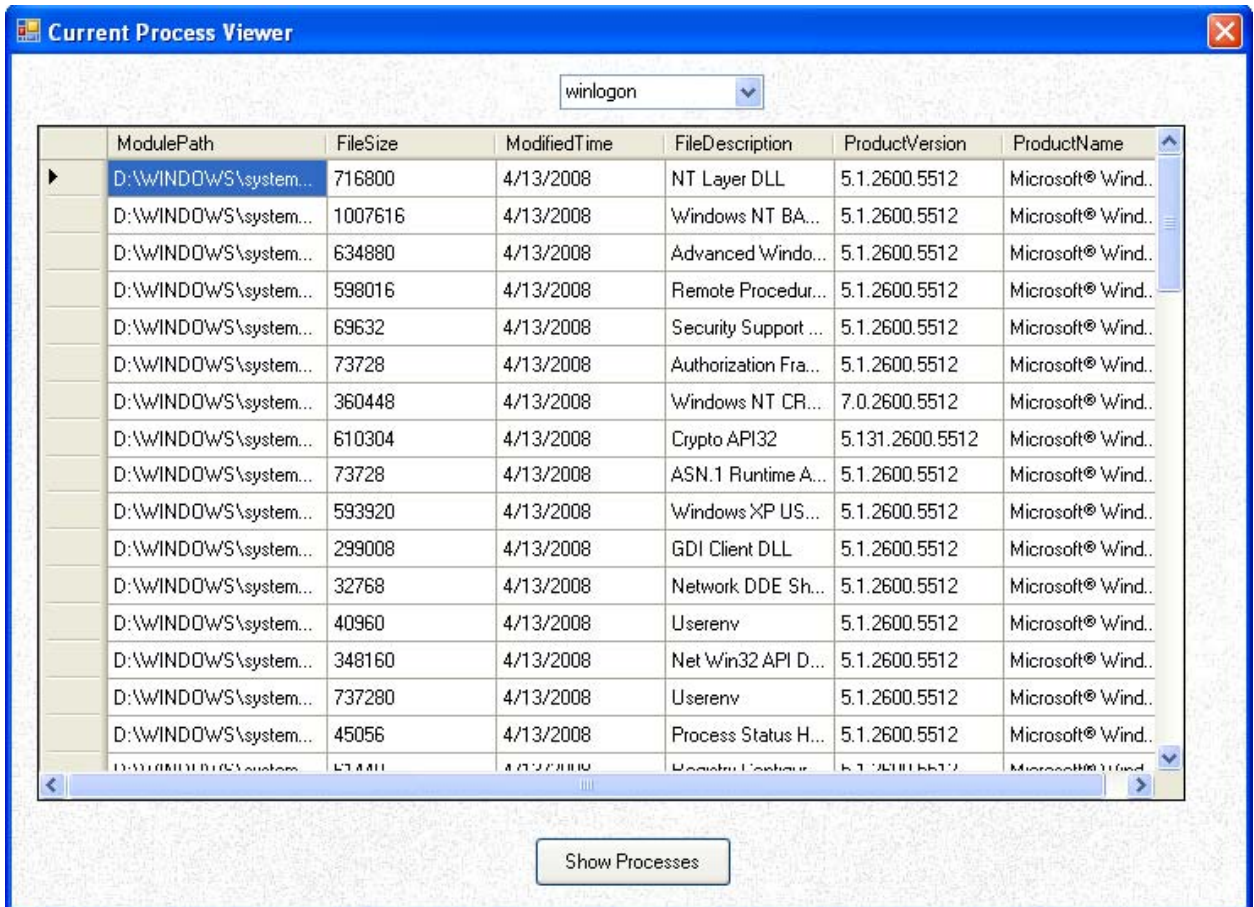


**CURRENT PROCESS VIEWER:**

This component will show you the current processes running on the system and all the DLL files loaded by those processes invidually.



Click the **Show Processes** button and it will show you the list of processes in the combobox and the DLL files of each process along with the File Size, Modified Time, File Description, Product Version and Product Name.



Again If you want to chose another component, you can go to the tools tab and select another tool.

**USB TRACKER:**

This component will list the information of all the USBs that were ever plugged into the system after operating system installation. The component will display the name, friendly name, serial number, first and last plug date, vendor id, product id, class and subclasses and names that were given to them.

Name	Friendly Name	Type	Serial Number	First Plug Date	Last Plug Date	Vendor ID	Product ID	Class	SubCl
DataTraveler 2.0	Kingston Dat...	USB Mass S...	5B82160006...	5/27/2009 7...	Not Available	13fe	1f00	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	5B84110001...	7/20/2009 1...	Not Available	13fe	1f00	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	000FEAFAA...	6/1/2009 11...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	5B6A1A835...	6/3/2009 1...	Not Available	13fe	1a00	08	06
Storage Media	Sony Stora...	USB Mass S...	2A08041423...	5/28/2009 1...	Not Available	054c	0243	08	06
Flash Disk	USB 2.0 Flas...	USB Mass S...	5b16fbd239...	7/21/2009 2...	Not Available	1307	0163	08	06
DataTraveler G2	Kingston Dat...	USB Mass S...	0014780F99...	7/14/2009 1...	Not Available	0951	1624	08	06
Cruzer Micro	SanDisk Cru...	USB Mass S...	2004351452...	5/27/2009 7...	Not Available	0781	5151	08	06
COBY MP3 Player	COBY MP3 ...	USB Mass S...	0010100010...	3/24/2009 3...	Not Available	0402	5661	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	0C11516150...	6/3/2009 12...	Not Available	08ec	0016	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	0010000000...	7/14/2009 9...	Not Available	0951	1607	08	06
DataTraveler 2...	Kingston Dat...	USB Mass S...	0000000797	6/3/2009 12...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	0014780EC5...	3/13/2009 1...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	8990000000...	6/3/2009 12...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	8990000000...	3/24/2009 1...	Not Available	0951	1603	08	06

### User's Account History Viewer

This component provides detailed information about all the user accounts present on the system. The interface of this module is shown below:

	Account Name	SID	Account Creation Time	Last Login Time	Registry Last Modification Time	Profile Folder Last Modification Time	Profile Path	Size Of Registry
▶	LocalService	S-1-5-19	3/13/2009 11:49...	7/27/2009 8:27...	7/27/2009 8:04...	3/13/2009 11:49...	D:\Documents a...	262144
	NetworkService	S-1-5-20	3/13/2009 11:48...	7/27/2009 8:27...	7/27/2009 8:04...	3/13/2009 11:48...	D:\Documents a...	262144
	umer	S-1-5-21-746137...	3/13/2009 11:50...	7/27/2009 8:25...	7/27/2009 8:03...	7/14/2009 10:49...	D:\Documents a...	2883584

This component will also give us the the following details:

1. The creation time of each user account will be displayed

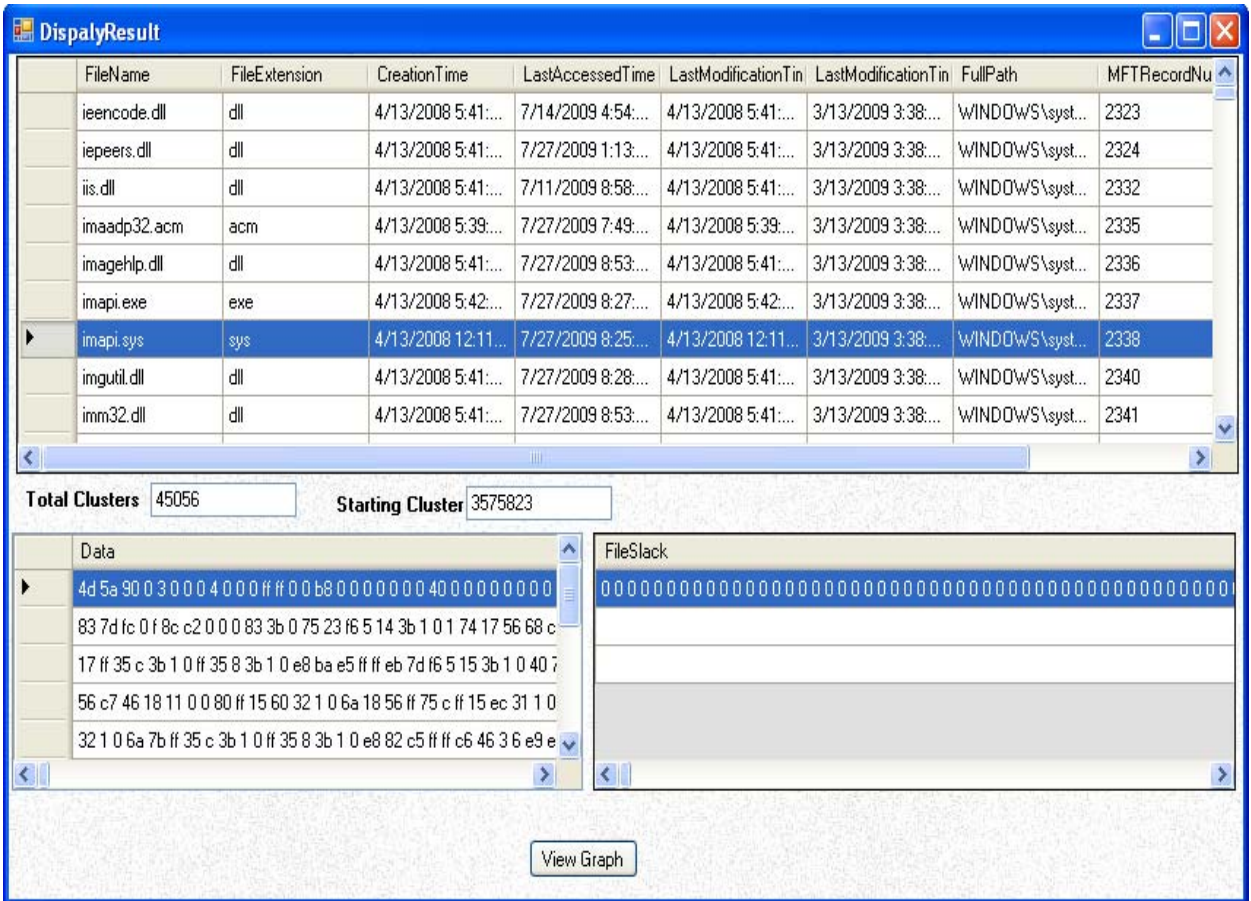
2. Last date of the changes made to the registry and the profile folder will be displayed
3. It gives information about the Security-ID of an account which is used widely for referring to a particular account in the windows registry.

### **FILE TRACKER:**

This component will generate a report of the files in a particular drive. The component will provide information about all the files (also the hidden ones) in the drive satisfying the search criteria. The component will be initiated by selecting options of date and drive name. As a result the system will generate a report of files satisfying the search criteria. We can also select the time which will help us to see the desired results.

The screenshot shows a window titled "File Tracker-Main" with a close button in the top right corner. The main content area has the title "FILE TRACKER" in blue. Below the title, there are two date selection fields: "Start Date Of Analysis" and "Last date of Analysis", both showing "07, 27, 2009". Below these is a checked checkbox labeled "Select The Time Range". Underneath, there are two rows of time selection controls: "FROM" and "TO". Each row has "Hour", "Min", and "Sec" dropdown menus, all currently set to "0". At the bottom, there are two buttons: "Scan Disk" and "Cancel".

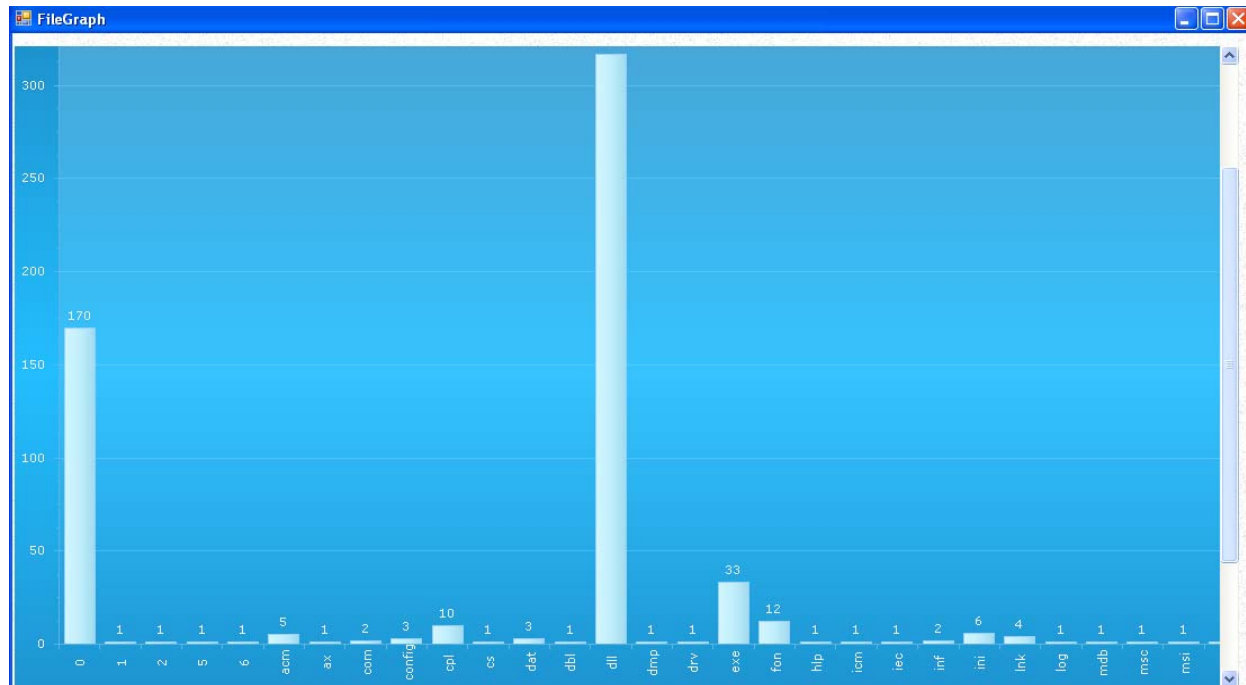
Select the date and time and then click **Scan Disk** which will display the result.



This component will give us the following information :

1. Modified, Accessed , Creation date and time and path for all the files satisfying the search criterion will be displayed.
2. The starting cluster number and total number of clusters used, for all the files found.
3. The data storage space that exists from the end of the file to the end of the last cluster assigned to the file is called "file slack". File slack for all the files found according to the search criteria should be displayed. The slack data will be displayed in the format as it is stored on disk.

We can also view the graph of how much a particular type of file has been accessed. The horizontal axis will show us the file extension and the vertical axis will show us how many times it has been accessed.



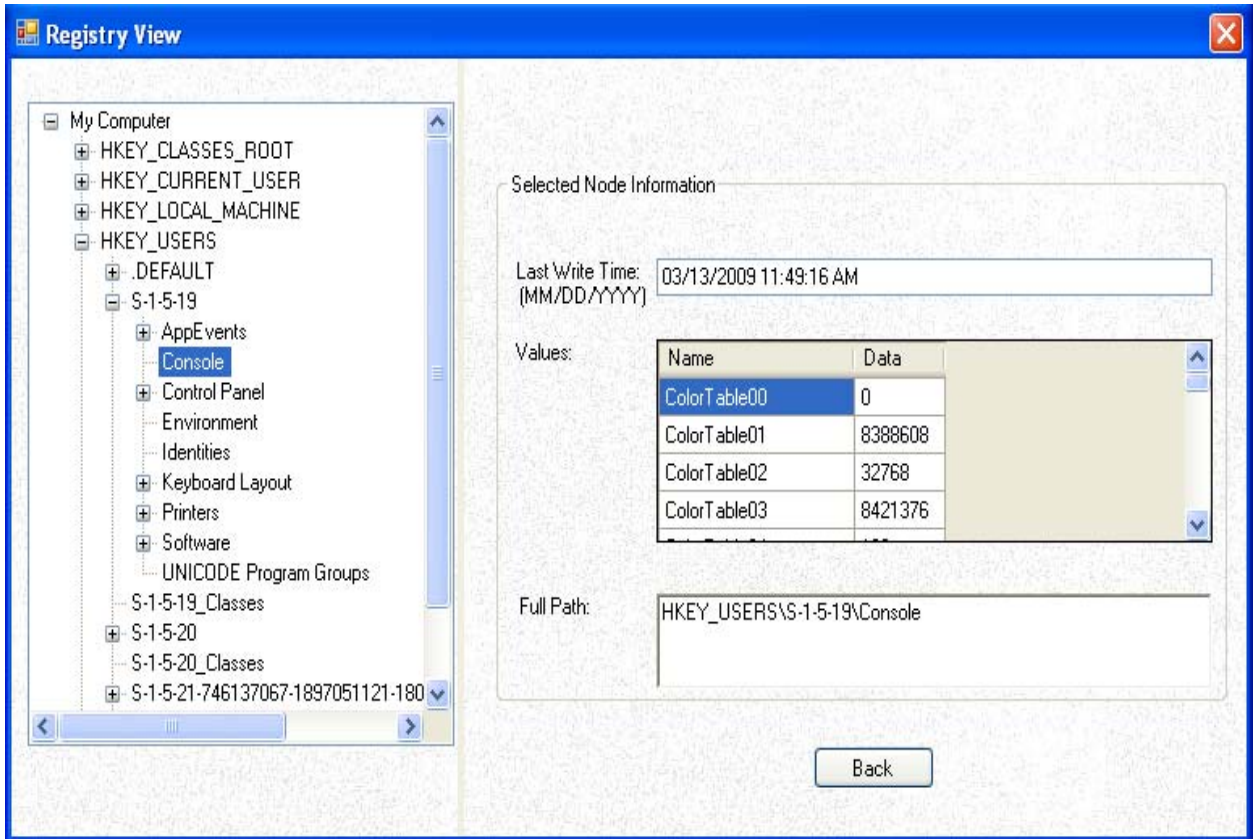
## Reg Tracker:

This component will generate a report of all the registry keys satisfying the search criteria given by the user. It will also act as a registry viewer showing last write time as additional information with each registry key. It will provide two options, first to view the registry in tree form and second to specify the search criteria in order to view certain keys. After selecting first option a registry view in tree form will be displayed. But if the second option is selected, then the user needs to select different options of dates and root key to make the search specific. As a result the component will show information of all the registry keys satisfying the search criteria.



By choosing the option of loading the entire Registry, you will get an interface which shows all the registry keys along with their last write time.





**RENS SNAPSHOTER:**

This component will grab the data stored in standard list views, tree views, list boxes, combo boxes, text boxes and edit boxes of the opened windows. This requirement is very important because it gives detailed information about all the opened windows in a system. The provided information can be sorted in a much convenient way as compared to capturing screenshots of opened windows and then analyzing them from investigation point of view.

The component will display the title of the opened window and the type, handle and window class of each control of an opened window.

The screenshot shows the RensSnapshoter application window. The top part is a table with columns: Title, Type, Handle, Items, Visible, Window Class, ProcessID, and Process Name. The bottom part is a list of application components with columns: ListViewItem, ListViewSubItem, and ListViewSubItem.

Title	Type	Handle	Items	Visible	Window Class	ProcessID	Process Name
Start Menu	SysListView32	327746	7	False	SysListView32	304	explorer
Start Menu	Button	327736	0	False	Button	304	explorer
Start Menu	SysListView32	262290	13	False	SysListView32	304	explorer
ComponentSele...	WindowsForms1...	198424	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	198428	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	198412	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	198408	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	198440	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	198434	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	461064	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	461032	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSele...	WindowsForms1...	657616	0	False	WindowsForms1...	1508	Win4Tech.vshos

ListViewItem	ListViewSubItem	ListViewSubItem
{E-mail}	{E-mail}	{Outlook Express}
{Internet Explorer}	{Internet Explorer}	{}
{Windows Messenger}	{Windows Messenger}	{}
{Media Player Classic}	{Media Player Classic}	{}
{Adobe Reader 8}	{Adobe Reader 8}	{}
{VLC media player}	{VLC media player}	{}
{Microsoft Office Word 2007}	{Microsoft Office Word 2007}	{}

## Chapter 1

### Introduction

#### 1.1 Introduction

This Chapter highlights the significance and necessity of Computer Forensics in the current era. Motivation behind taking the project, project scope, objectives, work break down structure (WBS) and other project specifications are also discussed in this chapter.

## **1.2 Preface**

The meaning of the word "forensics" is "to bring to the court". Forensics is the process which deals in finding evidence and recovering the data. The evidence includes many forms such as finger prints or complete files on computer hard drives etc. [1]

Since past few years, technical evidence has become more important in proving criminal and civil cases as compared to other physical evidence, this is due to a rapid development in the field of information technology and science.

It is necessary for network administrator and security staff of networked organizations to practice computer forensics and should have knowledge of laws because rate of cyber crimes is increasing greatly. It is very interesting for managers and personnel who want to know how computer forensics can become a strategic element of their organization security. Personnel, security staff and network administrator should know all the issues related to computer forensics. Computer experts use advanced tools and techniques to recover deleted, damaged or corrupt data and evidence against attacks and intrusions. These evidences are collected to follow cases in criminal and civil courts against those culprits who committed computer crimes. [1]

The survivability and integrity of network infrastructure of any organization depends on the application of computer forensics. In the current situations computer forensics should be taken as the basic element of computer and network security. [1]

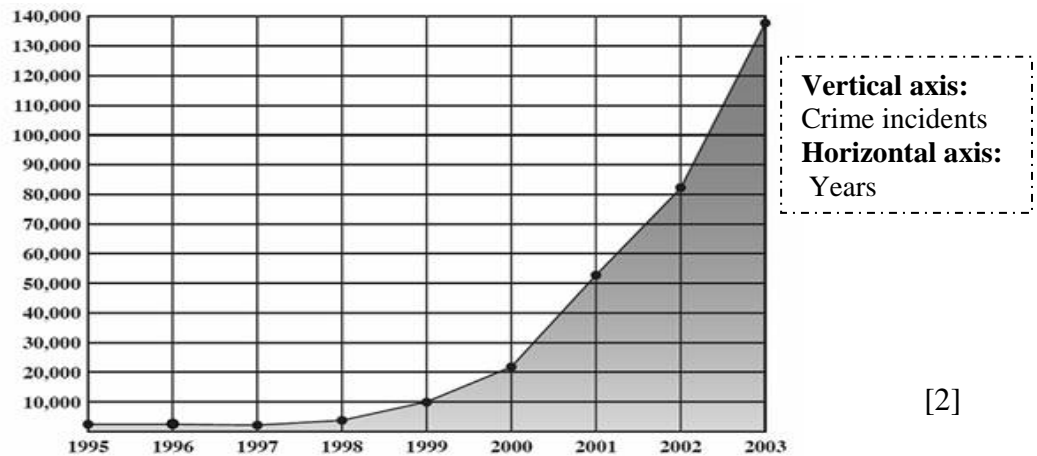
## **1.3 Motivation**

It was an earnest desire of our group to perform some work related to Digital Forensics. During the quest of right project to be chosen, our desire was enhanced when we were endeavored to solve a challenge 2008 provided on the website: [www.dfrws.org](http://www.dfrws.org). The motivation behind doing a project in forensics field, was to utilize our degree courses in implementing a real time application and correspondingly, to perform a research in this field too. This assisted us in developing a software application which is cost effective as

compared to existing forensic toolkits and useful and functional for Law Enforcement Agencies like **Federal Investigation Agency (FIA)**.

## 1.4 Problem Statement

Within past few years usage of computer technology has become very frequent in every field of life. It has benefited humans in a huge range i.e.; from everyday common tasks to huge technical and complex projects. But along with this positive side of picture, negative usage of computers is also increasing day by day. Cyber Crimes like hacking, information theft, virus attacks and malicious software productions have become a common practice these days. To deal with these threats it is highly important to secure information and perform analysis and collection of evidences from victimized systems.



*Figure 1.1 Graph showing Increasing Cyber Crime Incidents*

## 1.5 Project Scope

The scope of our project is to investigate users' activity which includes gathering evidence from

1. File system
2. System registry
3. System processes
4. Opened files

5. User accounts
6. USB records

This tool is developed for Windows XP operating system

## **1.6 Objectives**

The objective of our project is to develop software for gathering evidence from Microsoft Windows XP Operating System. This is a new area for both research and implementation.

## **1.7 Project Beneficiaries**

This project is not only useful for real time forensic investigation in Law Enforcement Agencies but also finds utility in examining standalone home pcs. This project proves to be beneficiary to those who want to take up the field of Computer Forensics and can bring improvements in this implemented software.

## **1.8 Project Title**

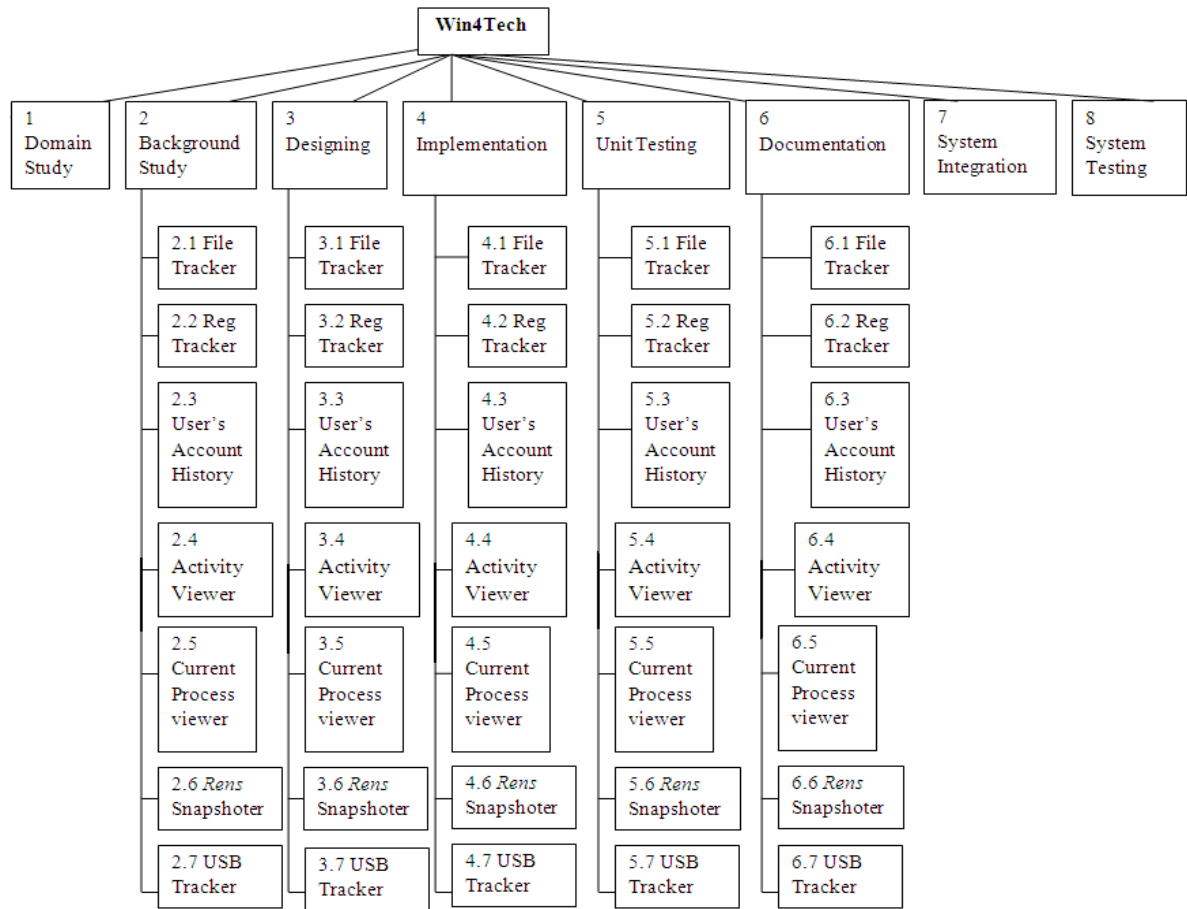
Win4Tech

## **1.9 Project Logo**



*Figure 1.2 Project Logo*

## 1.10 Work Breakdown Structure



*Figure 1.3 Work Breakdown Structure of Win4Tech*

## 1.11 Deliverables

Our project is a tool oriented project which aims to develop software. If any system (Windows XP installed) would be used for criminal activities then this tool will help in knowing about user's crime and will be a proof for law enforcement agencies to punish for his/her offense.

Deliverables of the project are:

- a. Running exe of the Software
- b. Code of the Project
- c. User Manual
- d. Complete Technical Documentation
- e. Test Cases and Results

## **1.12 Conclusion**

Hence, there is a requirement of developing a cost effective and efficient forensic toolkit which can prove to be helpful in preventing cyber crimes enhancements. The project specifications, objectives and deliverables are precisely discussed in this chapter.

## **Literature Review**

### **2.1 Introduction**

This chapter describes the literature studied for implementing the project. It describes the sources of evidence which are exploited by the project for converting the raw data into useful evidence. In this section various sources of evidence are discussed. It is also described that how these evidences can prove useful in a crime investigation.

### **2.2 Background**

The project comes under the domain of computer forensics. Computer Forensics deals with the preservation, identification, extraction and documentation of computer evidence [3]. Criminal activities like hacking, information theft, virus attacks and malicious software productions are growing rapidly. To deal with these threats, it is highly important to secure information and perform collection and analysis of evidence from victimized systems. The toolkit WIN4TECH performs tracking of files created, accessed or modified previously on the system, tracks the registry keys accessed, provides information of recent activities and current processes along with associated DLLs, provides USB and user account records and grabs data from the standard system controls from all the opened windows. All these tasks belong to different domains. Some require knowledge of file system, others require information about registry and still others require knowledge about the processes. Therefore the literature review is divided into three sections. First section deals with the file system (NTFS), second deals with Windows Registry and the third deals with the open processes.

### **2.3 File System (NTFS)**

Large amount of evidence related to computer usage can be extracted from analyzing the File System. As the structure of different file systems vary widely, so different types of evidence can be extracted from different file systems. WIN4TECH extracts evidence from NTFS file system.



## ***MFT***

NTFS is fundamentally organized as a set of files. These files include System files and Master File Table (MFT). MFT contains information of all files and directories present on a particular disk partition. Each file and directory has at least one record in the table. If the information about a file or directory does not fit in a single record, then multiple records may exist. These records are known as *extension records* and their information is present in the base record of each file [4].

### ***File Attributes***

Each MFT record is divided into header and a set of attributes. The header contains information like offset to the attributes and MFT number, assigned to the record. The header information is used for reading file attributes, which contain information about the associated file. Each attribute is also preceded by a header, which contains information about that attribute. The significant information in attribute header includes size of attributes, its compression status and whether the attribute is resident or non resident. Resident attributes fit within the MFT record and non resident attributes' data exists on the disk outside the MFT table. In case of non resident attributes, the MFT record contains offsets to the attribute data. Important attributes include Standard Attribute, File Name Attribute and Data Attribute. [5]

The *\$Standard Attribute* contains information regarding modification, last access and creation (MAC) times of a particular file or a directory. The MAC times associated with a file can prove to be useful evidence in tracing files accessed within a specific time interval for forensic analysis. It also includes attributes associated with a file like read only, system, hidden, sparse, temporary, reparse point, not-content indexed, device, archive, compressed, encrypted etc.[5]

The *\$FileName* Attribute contains information regarding filename, reference to the parent directory record and MAC times. The MAC times included in this attribute are updated only when the file name is changed; otherwise the Standard Attribute has updated values for the MAC times [5]. The reference to the parent directory of any file can be used to determine its parent directory and following the same procedure, full path of a file or directory can be resolved. Retrieving full path of a file can be helpful in acquiring complete information about the files.

The *\$Data* Attribute's header contains important information regarding file data. This information includes the size of file, disk space occupied by the file and compression status of file, etc. In order to determine file's compression status, data attributes contains a compression flag; if its value is non-zero then the data of the file is compressed and the size of compressed data is also present in the header [5].

The attribute itself contains file data. Data can exist in two forms; resident or non resident. In case of resident attributes, data exists in MFT record otherwise; offsets to the data are present in data runs [5]. Data runs contain information about the starting Cluster number of the file data and total number of clusters occupied by the file after the starting cluster [6]. If the file is fragmented then multiple values for starting cluster number and total clusters occupied will be present in the file's data runs. For retrieving file data the complete data run needs to be read.

### ***File Slack***

Criminals may exploit features of NTFS to hide suspicious data in certain disk locations. This hidden data is not directly shown by OS but it can be extracted by writing specialized programs. One of the locations where data can be hid is File Slack. File slack is the disk space left at the end of the last cluster of the file if the file size is not multiple of cluster size. File slack is further divided into ***RAM slack*** and ***drive slack***. RAM slack is the disk space existing from the end of file data to the end of last partially filled sector [7]. RAM slack handling is dependant on the OS. In Windows XP RAM slack is filled with zeroes. If it contains some data other than zeroes, then it indicates that RAM slack is manipulated for some suspicious activity [8].The drive slack contains data which was previously present on the disk. It can be the character pattern placed at formatting time or contents of some deleted file [8]. For finding the file slack, last cluster number assigned to a file needs to be located from the data runs. The size of **file slack** can be found by subtracting the file size (in terms of clusters) from the actual size of file data [8].

All the attributes and File Slack provide information about the storage and the usage of files present on the disk. Therefore analyzing them provides useful evidence which may become useful in solving a crime.

## 2.4 System Registry

System Registry is a system-defined database used by the Windows operating system to store hardware and software configuration information [9].

Registry contains a rich amount of evidence which can be useful in forensic investigation of a system. This section contains registry keys of forensic value which are catered in the project.

The commands fed in to **Run** window can be extracted from the registry key *HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU*. The value name, **MRUList** present in this registry key depicts the order of the commands entered in the Run Window. The data of **MRUList** is a string having value names, with the most recent being placed first and the oldest last.

The registry key *HKCU\Software\Microsoft\Search Assistant\ACMr\5603* provides names of the files and folders searched through **Windows Default Search**. Furthermore, the phrases and words searched within the files, which are entered in Windows Default Search can be extracted from registry key *HKCU\Software\Microsoft\Search Assistant\ACMr\5604*.

Recent files can be determined by the values of subkeys of the registry key *HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Recent Docs*. The subkeys are the file extensions and each extension contains the list of files recently accessed, accordingly.

The URLs typed in the Address field of Internet Explorer can be retrieved from the registry key *HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\TypedURLs*. 25 most recent URLs are stored in this key. The value names are named from **url1** to **url25**, showing the most recent URL being **url1**.

The programs executed at system start up can be viewed from the registry key *HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run*.

The information is helpful in detecting Trojans or viruses which run at system boot up.

Installed programs information can be retrieved from the registry key *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*.

The names of the programs which run with command prompt execution can be retrieved from the registry key *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Command Processor*.

The names of the files which are opened/saved using Open Save Dialog box can be retrieved from the registry key *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*.\*

Registry contains a lot of evidence regarding the USBs which are ever plugged into the system after operating system installation. The Registry key *HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Enum\USB* contains series of subkeys, each of a similar format (i.e., *USB\VID\_v(4)&PID\_d(4)&REV\_r(4)*). Each of these device ID subkeys also has one or more subkeys, which are the instance IDs for the devices that have been connected to the system. The subkeys of Registry key *HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\EnumUSBStor* have format like that of the device ID. Below them are instance ID subkeys, representing each of the devices that have been connected to the system. Registry keys have a value associated with them referred to as their “Last Write” time. This value is analogous to the last modification time usually associated with files, as it represents the last time the Registry key was modified in some way. The Last Write time of the keys can be used to make a partial timeline with respect to user activity involving USB storage devices. Additionally, USB storage devices that are assigned drive letters can be viewed through examination of the Registry values found under the key *HKEY\_LOCAL\_MACHINE\SYSTEM\MountedDevices*. To determine which drive letter a certain USB storage device was assigned, the value from “**ParentIDPrefix**” of the device’s key under USBSTOR and HKLM\System\MountedDevices Registry key can be helpful. [10]

## 2.5 Running Processes

Volatile data should be considered as evidence of interest to an investigation. This potentially exculpatory information may also simply “go away” when the system is turned off or loses power. The *processes running* at the time of computer seizure

constitute volatile data. Complete information about all visible and hidden processes must be extracted in order to analyze the computer system completely. The information includes process names, process IDs, dll files loaded by processes, information about opened windows like the title of window, handle of window, window visibility information, window classes, product name, product version, company name, executable file name, items inside the window like buttons, textboxes, combo boxes, list boxes, list views and edit boxes.

## System Requirements Specification (SRS)

### 3.1 Introduction

This chapter deals with the precise and explicit functionality and capabilities which a software system (i.e., a software application) must provide, as well as states any required constraints by which the system must abide. SRS contains software product's various features, functional and nonfunctional requirements

### 3.2 Intended Audience and Reading Suggestions.....

This Document is intended for developers, users, testers, and project team. This document comprises of project scope, functional and non functional requirements, components and different types of interfaces involved in the project.

SRS firstly describes the project scope, then overall description of the final product, followed by System Components and finally the functional and non functional requirements.

### 3.3 Business Context

Final outcome of our project will be helpful for the Law enforcement agencies. The product will be very cost effective as compared to the available forensic tool kits.

### 3.4 Product Perspective

Our product "Win4Tech" comes under the domain of Computer Forensics. Computer forensics involves gathering evidence from a computer system when it has been misused. Our product will examine a compromised machine for the purpose of gathering evidence from it and will provide a record of activity performed on that machine.

Already available forensic tools include *ENCASE* and *FTK* (Forensic Toolkit). These products are not cost effective. Our product will be useful for Law Enforcement Agencies and it will enable them to carry out investigation in a much convenient and cost effective manner.

### **3.5 Product Features**

Major functionality of our product will be inspecting users' activity.

Key features are:

1. Examining files which have been viewed previously by any user on the system.
2. Traversing Windows Registry and locating sub keys for every root key and returning the Last Write Time of any key value.
3. Searching the recent activities on a system which includes recent documents, windows searched items, programs that execute on system startup and run commands.
4. Providing record of Created User Accounts on the system.
5. Providing list of currently running processes and associated DLLs (Dynamic Link Library). It will also provide list of files opened by these processes.
6. Recording details of previously and currently connected USBs.
7. Providing detailed information about the objects (e.g. text boxes, list boxes, status bars, etc) residing in currently opened windows.

### **3.6 User Classes and Characteristics**

User of our tool will be a forensic investigator. He must be acquainted with domain knowledge of forensic field in order to use this tool effectively.

### **3.7 Operating Environment**

The underlying platform for Win4Tech will be *Microsoft Windows XP* with 32 bit architecture. Our product will function on a live system. *Dot Net framework 2.0* will be required to make it operative.

### **3.8 Design and Implementation Constraints**

We will follow *Evolutionary Development Model*. *Visual C#* will be the language used for implementation.

### **3.9 User Documentation**

For user assistance, we will provide help feature, embedded within the application and a complete technical documentation comprising of technical and implementation details.

### **3.10 Assumptions and Dependencies**

One of the assumptions is availability of live computer system instead of the digital image of hard disk. We also need to install Dot Net framework 2.0 to make our tool workable.

### **3.11.....Functional Requirements**

Functional requirements of Win4Tech include tracking of files created, accessed or modified previously on the system, tracking registry keys, providing information of recent activities and current processes along with associated DLLs, providing USB and user accounts records and grabbing the data from the standard system controls from all the opened windows. Win4Tech is composed of seven components and each component has its own functionality.

#### **I. Activity Viewer**

**Description:** This component will generate information regarding users' activities on a system such as recent files viewed, recent accessed URLs, the commands run from start->run, the search items previously entered by the user into *Windows Default Search*, the programs present at the system start up and the program list which runs with the execution of command processor.

The information yielded by this program can be very useful for performing forensics on any system.

### **Functional Requirements**



### **Recent Documents**

**Description:** There should be a provision that the user can view information of recent files accessed previously on the system. The information of recent files includes file pathname, file MAC times and account name through which the file was accessed.

**Criticality:** This requirement is highly important as it displays the previously viewed files. From investigation point of view, list of recent files can be a good evidence to detect users' activities on any system.

**Constraint:** This requirement needs administrator rights on the system in order to retrieve record of recent files from different user accounts on that system.

### **Recent URLs**

**Description:** This requirement is to obtain a collection of recent URLs which have been browsed by the user on any system.

**Criticality:** This requirement is significant enough as it can help in discovering users viewed URLs. It can be important evidence in a situation where a criminal activity is performed through internet.

### **Search Commands Entered Into Windows Default Search**

**Description:** This requirement is to obtain the search commands written in *windows Default Search*, including the searched filenames, words or phrases contained in a file.

### **Programs Present At the System Start Up**

**Description:** This requirement is to obtain a collection of Programs which execute on the system start up.

**Criticality:** This requirement is considerably important because it retrieves the programs run when the system gets started. If there is some Trojan present in the system which gets activated at the system boot up then through this requirement the Trojan can be detected.

**Constraint:** This requirement needs administrator rights on the system in order to retrieve the list of StartUp programs.

**List of Programs Run With the Execution of Command Prompt**

**Description:** This requirement is to obtain a set of programs which run with command prompt execution.

**List of AutoRun programs**

**Description:** This requirement is to obtain a set of programs, configured to run during system bootup or login.

**List of Programs Installed On a System**

**Description:** This requirement is to obtain a set of programs that have been installed on the system.

## **II. Current Process Viewer**

**Description:** The component will generate a report about all the processes currently running on the system, along with the DLLs loaded by these processes. It also returns list of files which are being accessed by the currently active processes.

### **Functional Requirements**

**Current Processes**

**Description:** All the processes currently running on the system will be displayed.

**Criticality:** This requirement is important as it can give information about any suspicious process running invisibly on the system.

**DLLs for Each Process**

**Description:** The DLLs loaded by each process will be displayed.

**Criticality:** This requirement is important as it can point to a malicious dll associated with a valid process.

### **Files Accessed By Each Process**

**Description:** The files being accessed by each process will be displayed.

**Criticality:** This requirement is important as it can point to an important file being accessed by an infected (but valid) process.

### **III. File Tracker**

**Description:** This component will generate a report of the files in a particular drive. The component will provide information about all the files (also the hidden ones) in the drive satisfying the search criteria.

**Stimulus/Response Sequences:** The component will be initiated by selecting options of date and drive name. As a result the system will generate a report of files satisfying the search criteria.

### **Functional Requirements**

#### **MAC Times**

**Description:** Modified, Accessed and Creation date and time for all the files satisfying the search criterion will be displayed.

**Criticality:** This requirement is most important as it serves as the main evidence for relating date and times to a particular crime.

**Issues:** Underlying file system can be NTFS or FAT; therefore they need to be handled separately.

**Dependencies with Other Requirements:** This requirement is dependent on Search Criteria requirement.

### **Search Criteria**

**Description:** User will be provided various choices such as selecting start and end date, drive name and the option whether to search by modification, accessed or creation date and time of files.

**Criticality:** The requirement is very important as it allows users to achieve brief and compact record of files of his/her interest.

**Dependencies with Other Requirements:** All other requirements are dependent upon this requirement.

### **Filename and File pathname**

**Description:** The file name and file path for all the files found, according to the search criteria will be displayed.

**Technical Issues:** Underlying file system can be NTFS or FAT; therefore they need to be handled separately.

**Dependencies with Other Requirements:** This requirement is dependent on Search Criteria requirement.

### **Cluster Number and Total Number Of Clusters**

**Description:** The starting cluster number and total number of clusters used, for all the files found, according to the search criteria will be displayed.

**Technical Issues:** Underlying file system can be NTFS or FAT; therefore they need to be handled separately.

**Dependencies with Other Requirements:** This requirement is dependent on Search Criteria requirement.

### **Data of File**

**Description:** The data of each file found according to the search criteria will be displayed. The data will be displayed in the format as it is stored on disk.

**Technical Issues:** Underlying file system can be NTFS or FAT; therefore they need to be handled separately.

**Dependencies with Other Requirements:** This requirement is dependent on Search Criteria requirement.

### **File Slack**

**Description:** The data storage space that exists from the end of the file to the end of the last cluster assigned to the file is called "file slack". File slack for all the files found according to the search criteria should be displayed. The slack data will be displayed in the format as it is stored on disk.

**Technical Issues:** Underlying file system can be NTFS or FAT; therefore they need to be handled separately.

**Dependencies with Other Requirements:** This requirement is dependent on Search Criteria requirement.

## **IV. Reg Tracker**

**Description:** This component will generate a report of all the registry keys satisfying the search criteria given by the user. It will also act as a registry viewer showing last write time as additional information with each registry key.

**Stimulus/Response Sequences:** The component will provide two options, first to view the registry in tree form and second to specify the search criteria in order to view certain keys. After selecting first option a registry view in tree form will be displayed. But if the second option is selected, then the user needs to select different options of dates

and root key to make the search specific. As a result the component will show information of all the registry keys satisfying the search criteria.

## **Functional Requirements**

### **Registry view option**

**Description:** The component will display registry in tree form.

**Criticality:** This requirement is very important because it gives flexibility to the user if he exactly doesn't know the key name which he is looking for.

### **Search Capability**

**Description:** Search capability will be provided to the user so that he can view the results of his interest. User will be given option to select start date and end date. There will be an option for the user to select a root key or to write the key name himself if he wants to make the search more specific. After specifying the search criteria the component will display information about the registry keys satisfying the search criteria.

**Criticality:** The requirement is very important because it will allow the user to do a quick search instead of traversing the registry tree again and again.

**Constraint:** The information of the registry keys which are protected by the system will not be displayed.

### **Last Write Time**

**Description:** Last write time will be displayed with each registry key in tree form and also with the keys in the search specific records

**Criticality:** This requirement is most important because without it evidence is of no use because it can't be related to any crime.

**Constraint:** Last write time will not be displayed for the system protected keys.

**Dependencies with Other Requirements:** This requirement is dependent on the above two requirements because without their fulfillment it is not possible to achieve this requirement.

**Registry key path and registry key values**

**Description:** The component will display registry key path and values for the selected key in the tree view. The registry key path and values will also be displayed in the records returned in case of specific search.

**Criticality:** The requirement is very important because the values of the registry keys serve as evidence in many cases.

**Constraint:** The requirement will not be fulfilled for the system protected keys and sub keys.

## **V. Rens Snapshoter**

**Description:** This component will grab the data stored in standard list views, list boxes, combo boxes, text boxes and edit boxes of the opened windows.

### **Functional Requirements**

**Title, Type, Handle and Window Class**

**Description:** The component will display the title of the opened window and the type, handle and window class of each control of an opened window.

**Criticality:** This requirement is very important because it gives detailed information about all the opened windows in a system. The provided information can be sorted in a much convenient way as compared to capturing screenshots of opened windows and then analyzing them from investigation point of view.

**Visibility Information**

**Description:** The component will tell whether the window is visible or not.

**Criticality:** The requirement is very important because it helps in tracing out any suspicious windows and processes that are running.

**Process ID, Process Name**

**Description:** The component will display the process ID and process name of the process which opened a particular window.

**Criticality:** This requirement is most important because it helps in figuring out whether the window is opened by a malicious process or not.

## **VI. USB Tracker**

**Description:** This component will list the information of all the USBs that were ever plugged into the system after operating system installation.

### **Functional Requirements**

**Name and Friendly Name**

**Description:** The component will display the name and friendly name of each USB.

**Criticality:** This requirement is very important because it can shorten the time to trace the USB used for any crime.

**First Plug Time**

**Description:** The component will display first plug date and time of each USB

**Criticality:** This requirement is most important because it serves as a proof that a particular USB was plugged at a particular time.

**Last Plug Time**

**Description:** The component will display the last plug time of each USB.



**Criticality:** The requirement is very important because it can be helpful for deciding whether the USB was used in a particular crime or not.

**Constraint:** Last Plug Time will be displayed only for the USBs that were plugged before the recent system start.

### **Status**

**Description:** The component will display the status information of each USB. The status information can be: Connected or Not Connected.

### **Drive Letter, Vendor ID and Product ID**

**Description:** The component will display drive letter assigned to the USB and its vendor and product ID.

**Constraint:** If two USBs are assigned same drive letters at different times then only the drive letter of the USB which was plugged afterwards can be retrieved.

### **Serial number, Class and Sub Class**

**Description:** The component will display serial number, class and sub class of each USB.

**Criticality:** The requirement is very important because serial number serves as a unique identifier of a USB.

## **VII. User's Account History Viewer**

**Description:** This component provides detailed information about all the user accounts present on the system.

### **Functional Requirements**

#### **Last Login time of Each Account**

**Description:** The last login time for each user account will be displayed in the report.

**Criticality:** This requirement is very important as it serves as evidence while determining the activities of different account owners present on the computer.

**Creation Time of Each Account**

**Description:** The creation time of each user account will be displayed in the product.

**Criticality:** The requirement is important as it allows the investigator to link the activities occurred between certain dates to a user account.

**Registry and Profile Folder Modification Times**

**Description:** Last date of the changes made to the registry and the profile folder will be displayed in the report.

**Criticality:** This requirement is important as it gives information about the dates on which user has incorporated any change to his personal settings or has made any changes to his profile folder.

**Account Name and SIDs**

**Description:** All accounts present on the system will be displayed along with their SIDs.

**Criticality:** This requirement is important as it gives information about the SID of an account which is used widely for referring to a particular account in the windows registry.

**3.12..... User Interface Requirements**

We will follow HCI (Human Computer Interaction) standards for designing the interface of Win4Tech. Following are the user interface requirements specific to the project:

1. The interface should be designed in such a way that the tool can be used without any formal training.

2. In the designing of interface elements, care must be taken to minimize the cognitive overload, by logically linking and designing the interface units and elements.
3. There should be minimal typing overhead while taking input from the user such as providing an option to select dates from a calendar, instead of typing by hand.
4. All the reports generated by the tool should follow the same format.
5. All the reports should be sortable so that user can sort the results according to his requirements.
6. The content of the reports should be aligned to the left.
7. Multiple views should be available for viewing the results.
8. Option for saving reports in *csv* format should be incorporated.
9. The interface should be usable and familiar to the user. Colors and styles of buttons, toolbars, textboxes and other interface elements should be the same as in the Microsoft tools such as displaying the close button in the top right corner instead of displaying it somewhere else.
10. Execution progress should be made visible to the user. This can be achieved by displaying the progress bar.
11. Error messages should be very descriptive and self explanatory.
12. Elaborate help should be provided which should be component specific.
13. Help should be displayed in a separate window.
14. Help material should be written in clear, familiar language avoiding jargon as much as possible. It should also be logically organized.

### **3.13 Other Nonfunctional Requirements**

#### **Performance**

The execution time for each component of the tool will not exceed 7 minutes.

#### **Correctness**

All the results obtained will be 100% accurate. Any incorrect result can lead to wrong perception of a particular crime.

**Maintainability**

Proper design and programming standards will be followed to provide maintainability.

**Reliability**

The system will not crash in unexpected situations, but it should generate suitable errors.

**Testability**

The system will be tested in all possible scenarios.

**Usability**

The system will be usable enough to be used without formal training.

## **System Requirements Specification (SRS)**

### **4.1 Introduction**

The purpose of this chapter is to provide all details of the System Architectural Design, Decomposition Design, Data Design, Component Design, Human Interface Design, Graphical Model, UML Based Design for Win4Tech. The Architectural Design part focuses on the high-level project decomposition and Component Design and UML Based Design focus on the low level description of the implementation classes and all their attributes and methods.

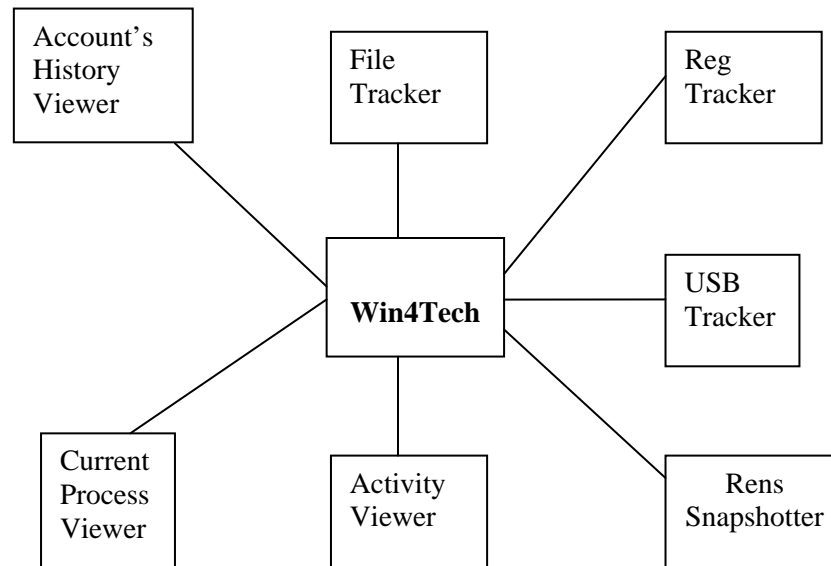
### **4.2 .....System Overview**

This product “Win4Tech” comes under the domain of Computer Forensics. Computer forensics involves gathering evidence from a computer system when it has been misused. Our product will examine a compromised machine for the purpose of gathering evidence from it and will provide a record of activity performed on that machine.

Already available forensic tools include **ENCASE** and **FTK** (Forensic Toolkit). These products are not cost effective. Our product will be useful for Law Enforcement Agencies and will enable them to carry out investigation in a much convenient and cost effective manner.

### 4.3 System Architectural Design

The architectural design of Win4Tech can be illustrated as:



*Figure 4.1 Architectural Design of Win4Tech*

### Modular Structure

Win4Tech is further divided into seven modules. This division is based on functionality difference.

These components are listed as follows:

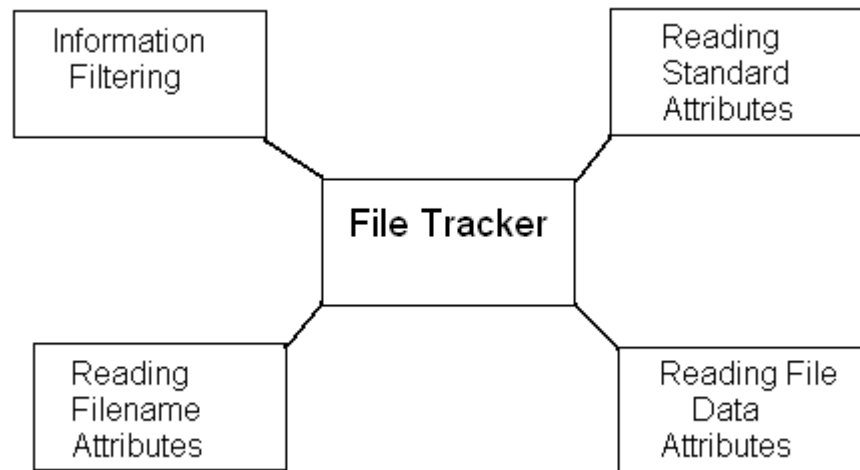
1. Activity Viewer
2. Current Process Viewer
3. File Tracker
4. Reg tracker
5. *Rens* Snapshotter
6. USB Tracker
7. User Accounts History

All these components have different functionalities and are therefore independent of each other. These all components will be assembled in one basic interface and they can be accessed through that interface.

### Functionality of Each Subsystem

1. **Activity Viewer** - Searching the recent activities on a system which includes recent documents, windows searched items, programs that execute on system startup and run commands.
2. **Current Process Viewer** - Providing list of currently running processes and associated DLLs (Dynamic Link Library). It will also provide list of files opened by these processes.
3. **File Tracker** - Examining files which have been viewed previously by any user on the system.
4. **Reg Tracker** - Traversing Windows Registry and locating sub keys for every root key and returning the Last Write Time and values of any key.
5. **Rens Snapshotter** - Providing detailed information about the objects (e.g. text boxes, list boxes etc) residing in currently opened windows.
6. **USB Tracker** - Recording details of previously and currently connected USBs.
7. **User Accounts History** - Providing record of Created User Accounts on the system.

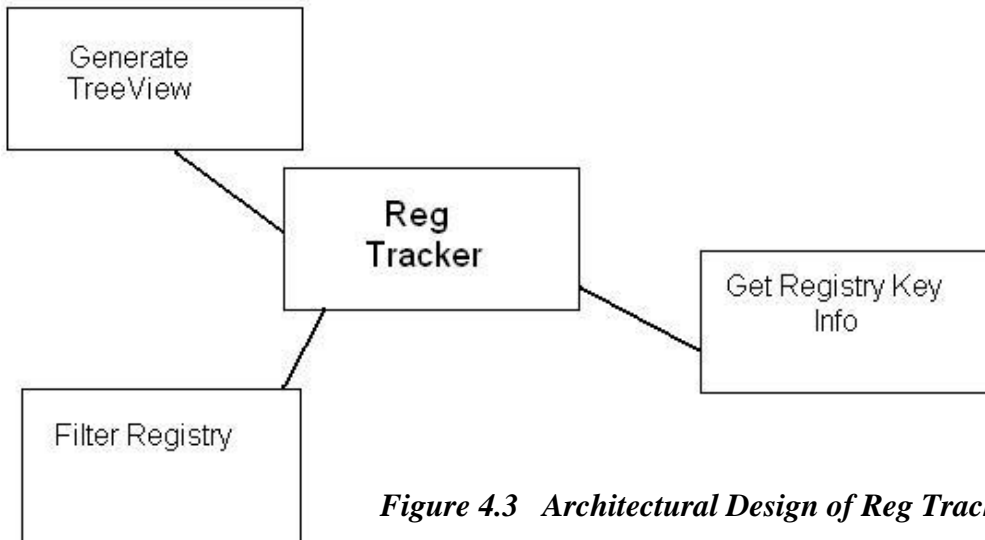
#### 4.4 Decomposition Design



*Figure 4.2 Architectural Design of File Tracker*

#### Architectural Design of Reg tracker

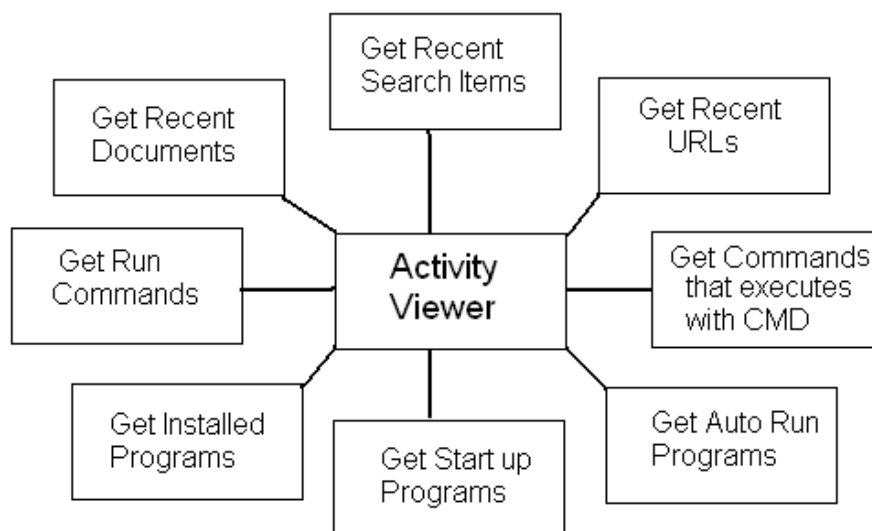
RegTracker is divided into two processing units. First is responsible for generating the tree structure of registry and second is responsible for the extraction of important information from the registry key.



**Figure 4.3 Architectural Design of Reg Tracker**

**Architectural Design of Activity Viewer**

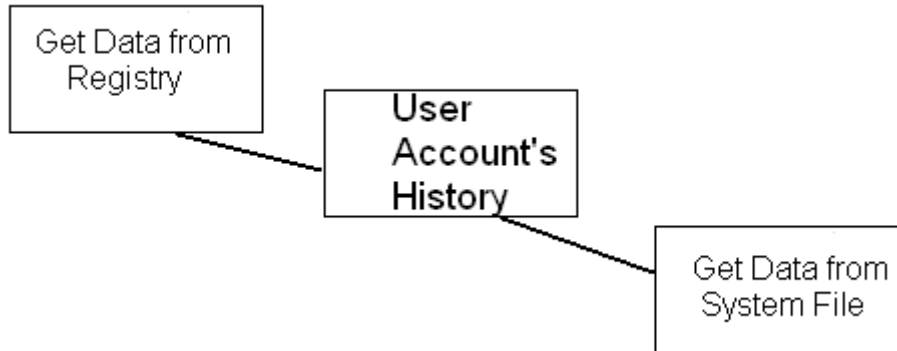
Activity viewer is divided into seven processing units. The information extracted by these units includes Recent documents accessed, Recent URLs accessed, Installed Programs, Autorun programs, Programs that run with system startup and CMD command, Commands entered into RUN and Windows Search.



**Architectural Des** *Figure 4.4 Architectural Design of Activity Viewer*



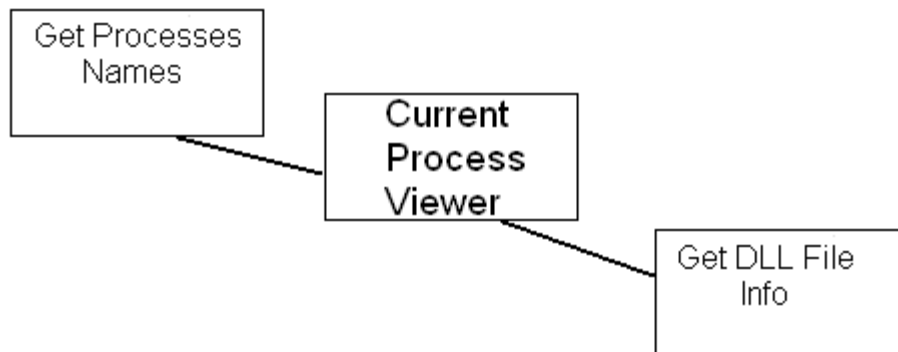
Users' account history viewer is divided into two processing units. One is responsible for extracting information from Registry and the other retrieves information from System Files.



*Figure 4.5 Architectural Design of User Accounts History*

### **Architectural Design of Current Process Viewer**

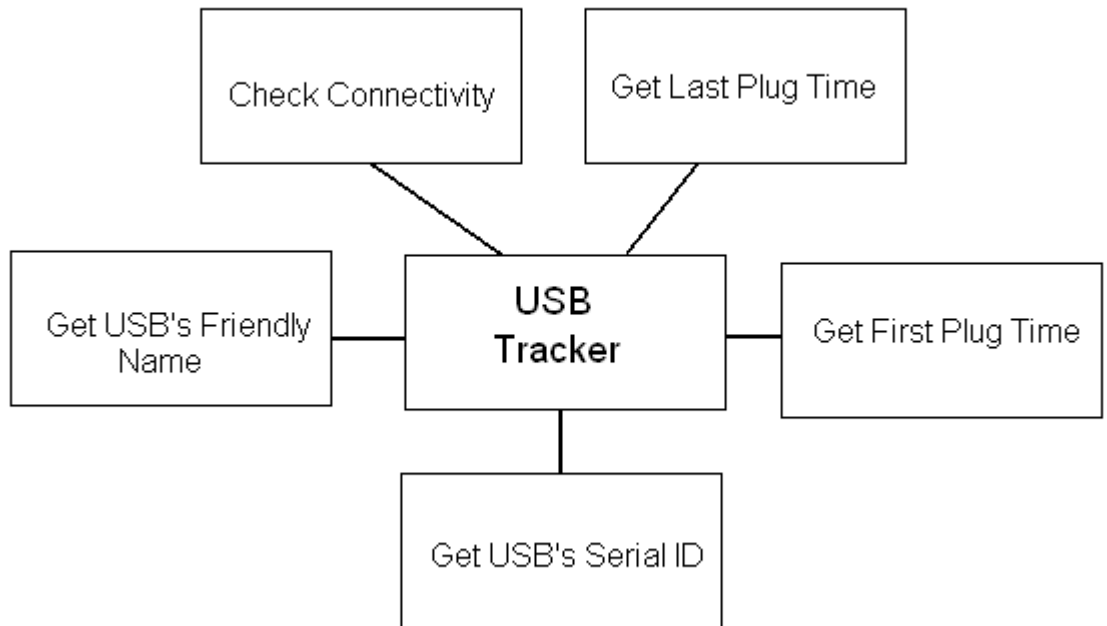
Current Process Viewer is divided into two processing units. One is responsible for extracting information from running processes and the other retrieves information from the DLL files.



*Figure 4.6 Architectural Design of Current Process Viewer*

### **Architectural Design of USB Tracker**

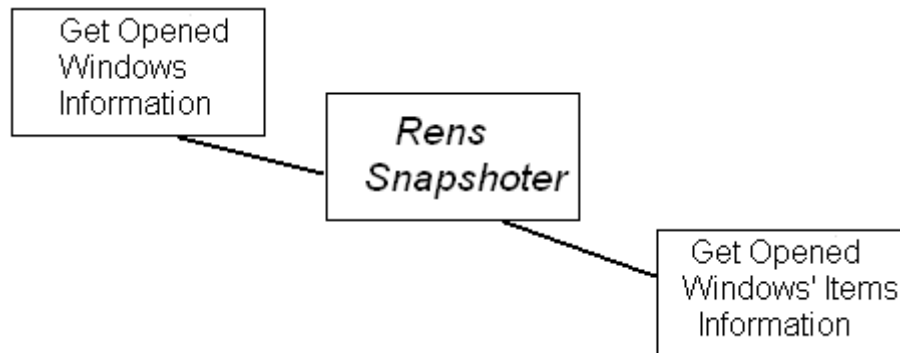
USB Tracker is divided into five processing units. Each module extracts information from System Registry. Extracted Information includes Last Plug Time, First Plug Time, USB name and USB Serial Number.



**Figure 4.7 Architectural Design of USB Tracker**

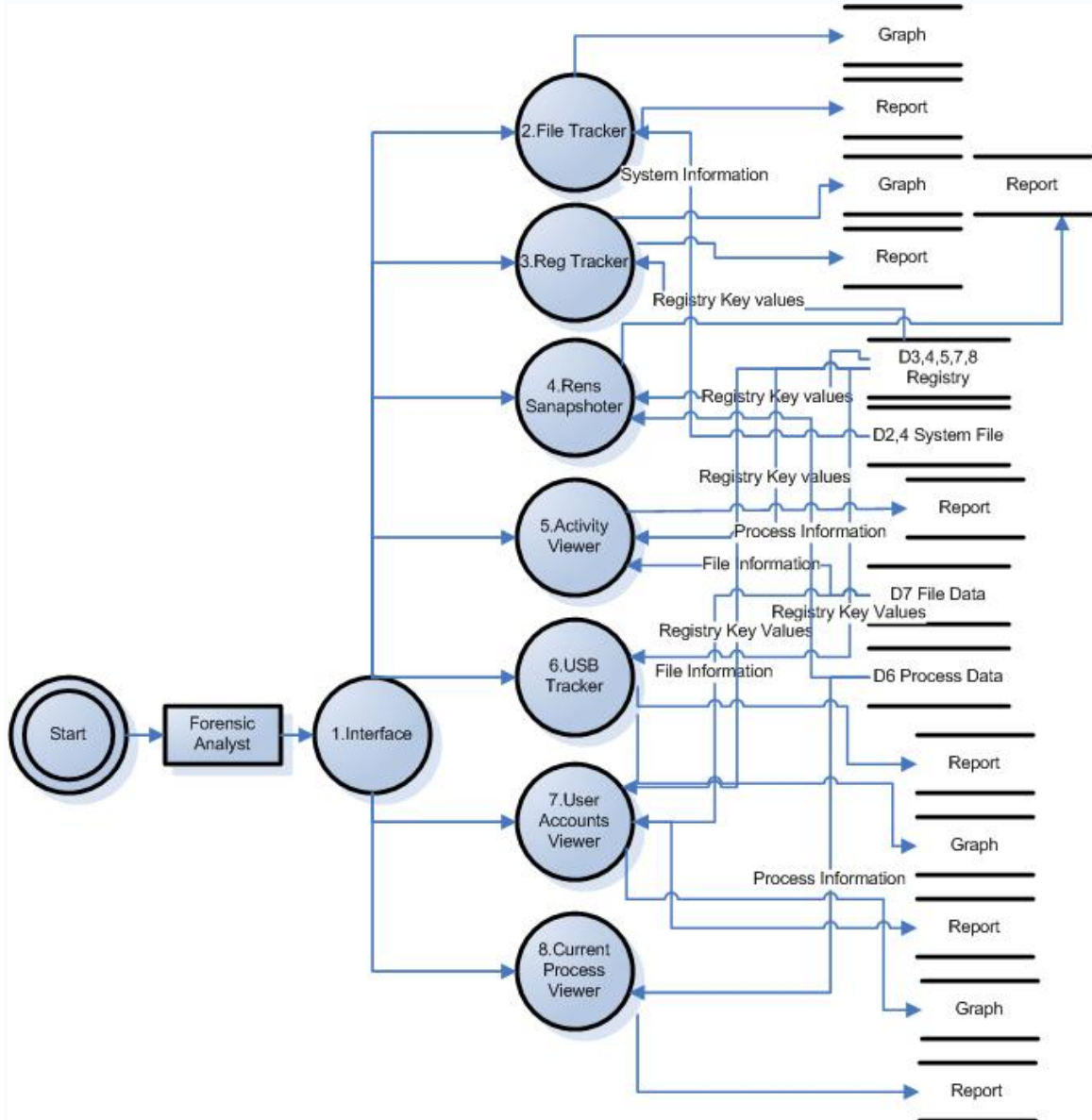
**Architectural Design of Rens Snapshoter**

Rens Snapshoter is divided into two processing units. One is responsible for extracting information about opened windows and the other retrieves information about the windows items.



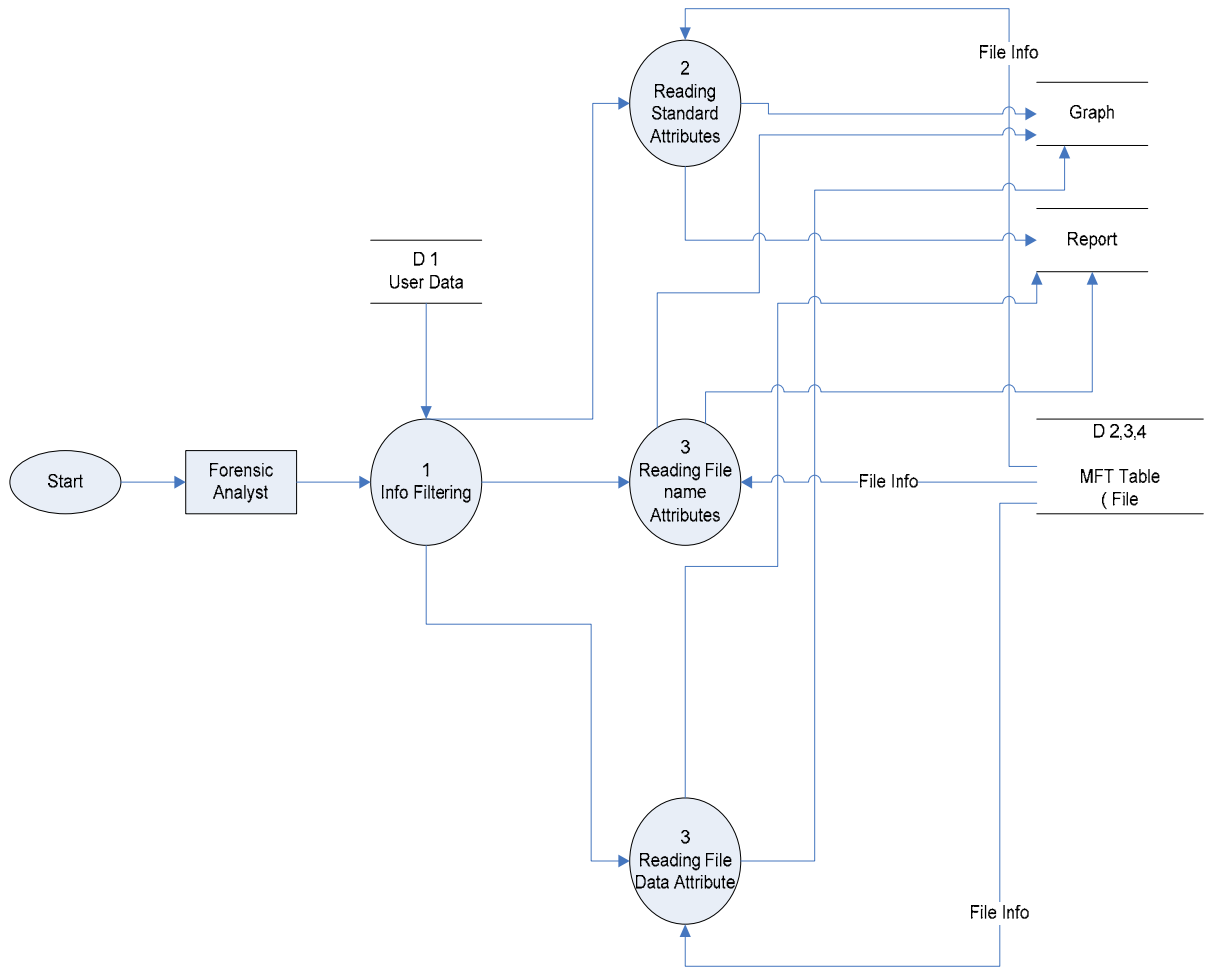
**Figure 4.8 Architectural Design of Rens Snapshoter**

Data flow diagram of Win4Tech is the graphical representation of the "flow" of data through the [system](#). This DFD can also be used for the [visualization](#) of [data processing](#) (structured design) of the whole system.

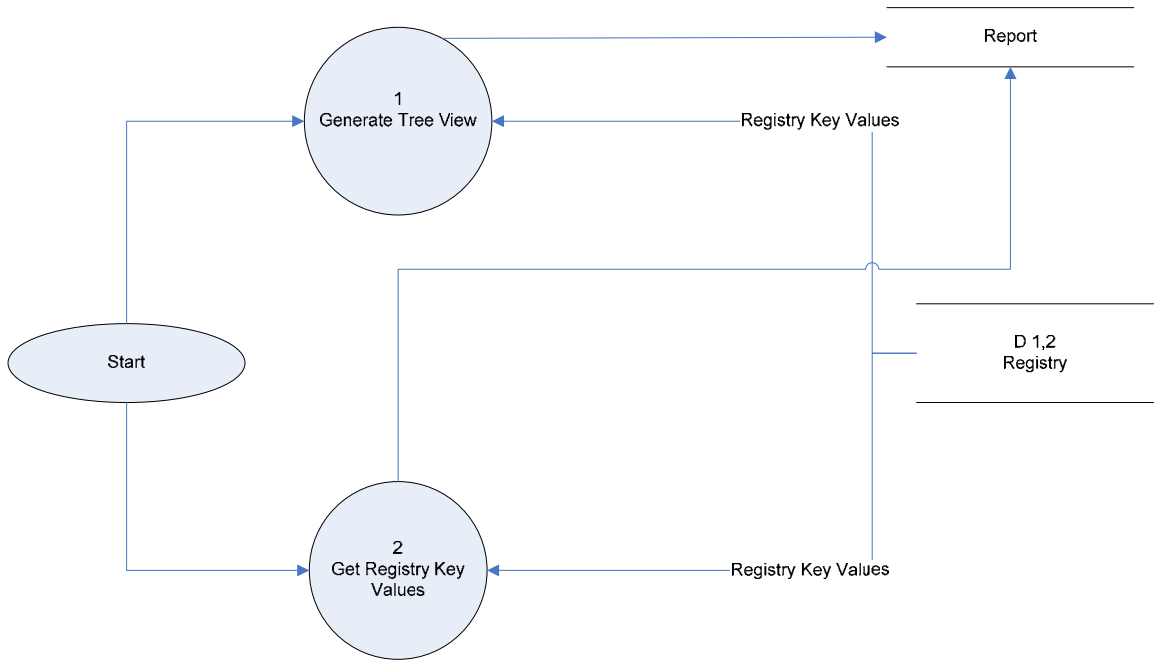


*Figure 4.9 Data Flow Diagram of Win4Tech*

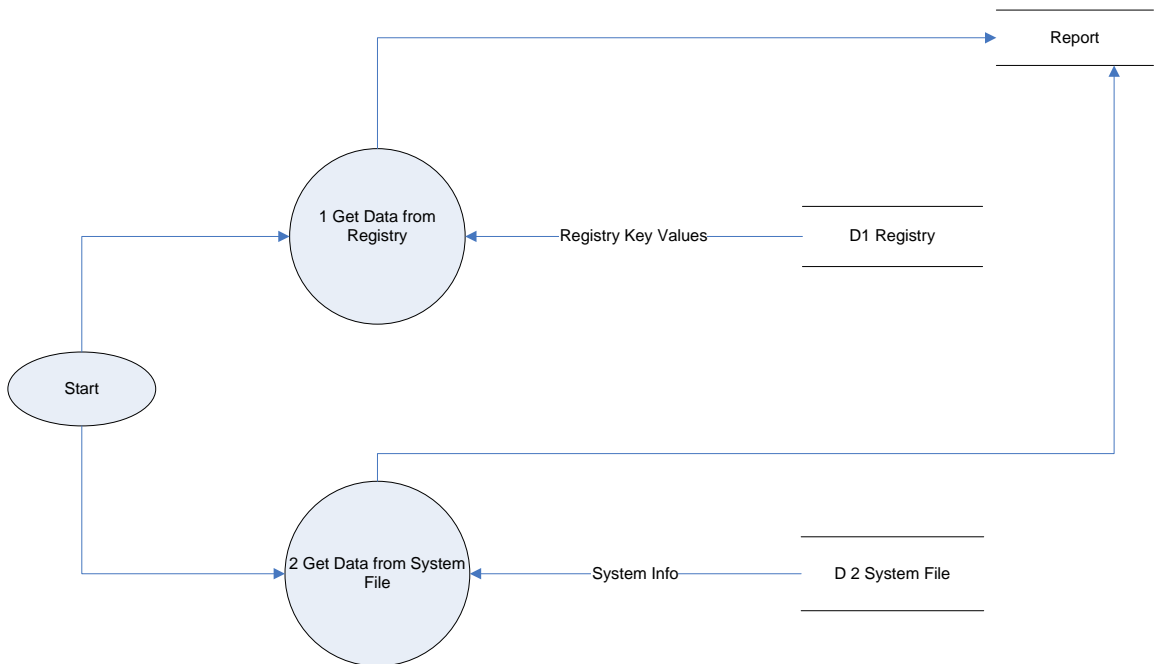
### Data Flow Diagrams (DFD's) of Subsystems



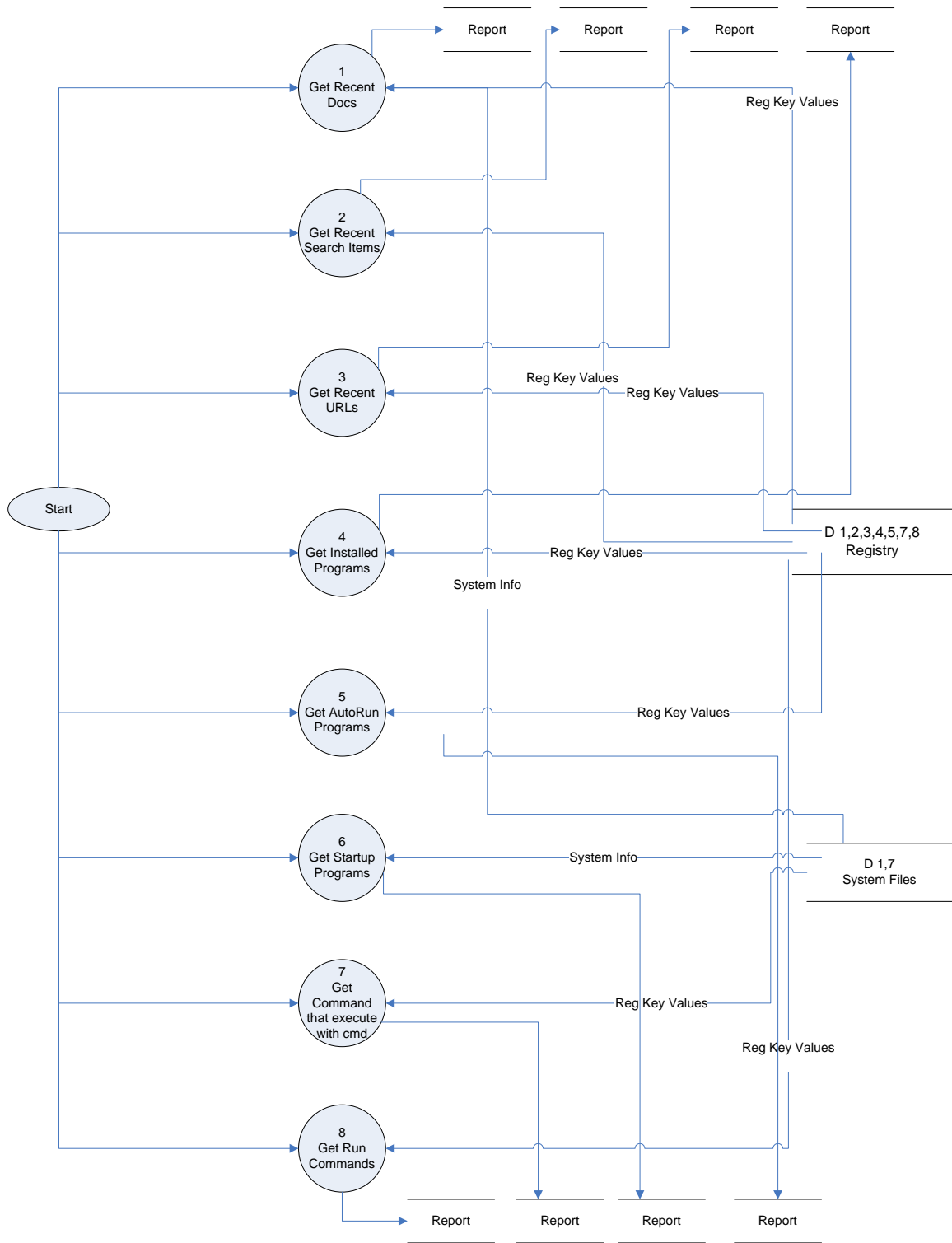
**Figure 4.10 Data Flow Diagram of File Tracker**



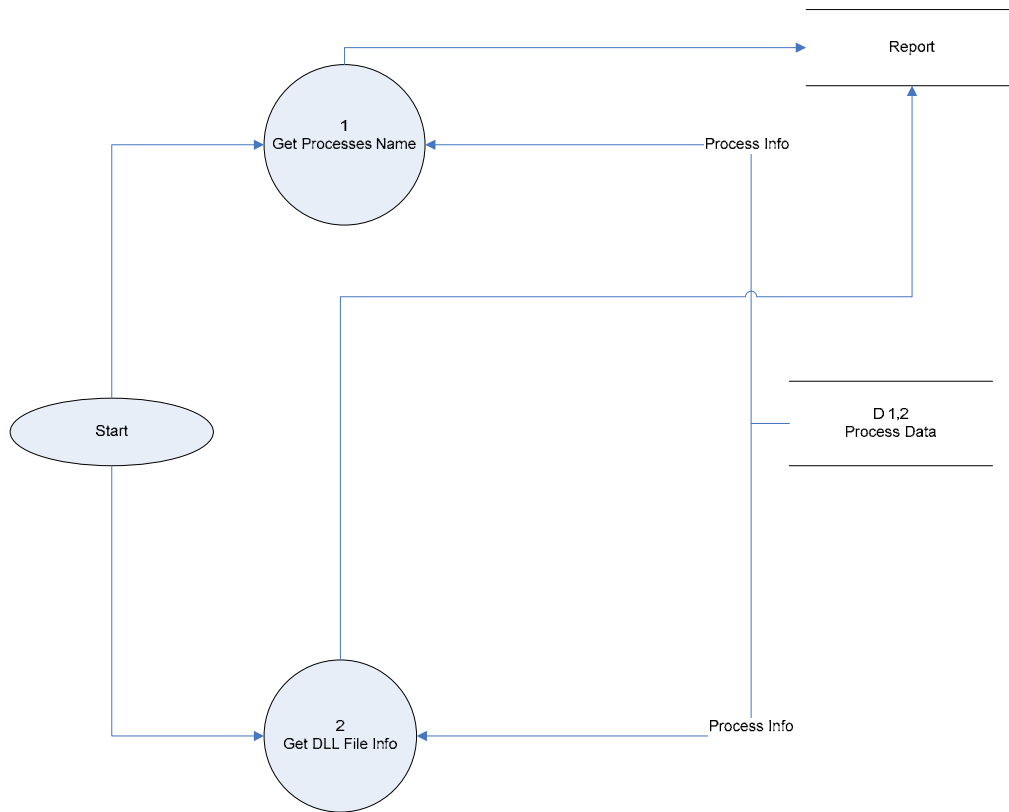
**Figure 4.11 Data Flow Diagram of Reg Tracker**



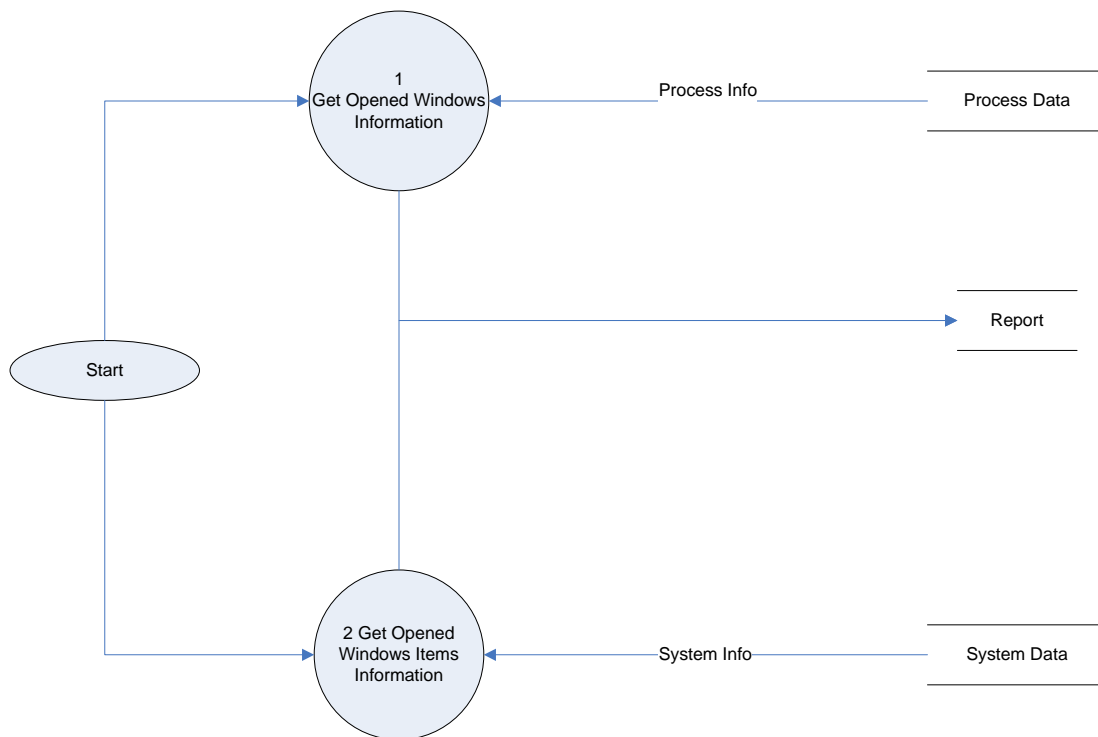
**Figure 4.12 Data Flow Diagram of User Accounts History**



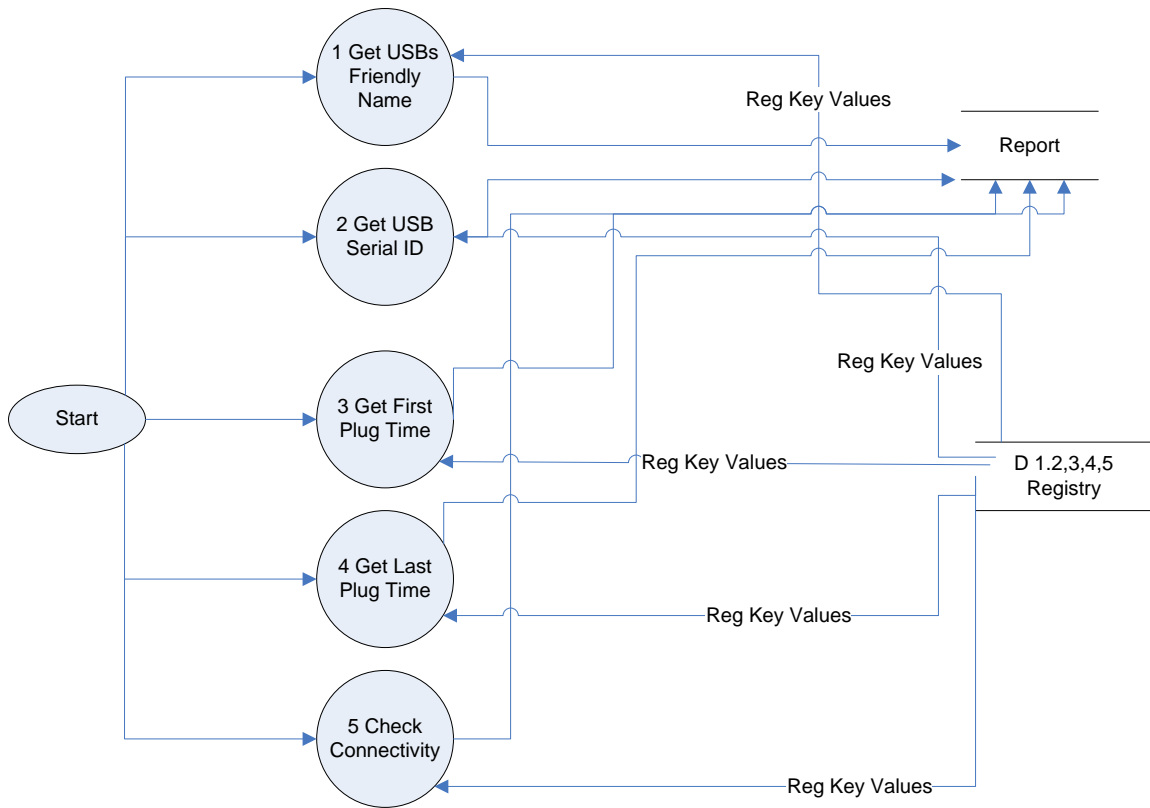
**Figure 4.13 Data Flow Diagram of Activity Viewer**



**Figure 4.14 Data Flow Diagram of Current Process Viewer**



**Figure 4.15 Data Flow Diagram of Rens Snapshotter**



**Figure 4.16 Data Flow Diagram of USB Tracker**

## 4.6 Data Description

### Information Domain

The places from where Win4Tech retrieves information are System Data (different System files), File System Data (e.g. NTFS and FAT Data), Running Process Data and System Registry Data.

All these information domains play vital role in tracing any user's activities. In order to capture all such activities, Win4Tech components search these locations to bring out useful information for forensic analysis.

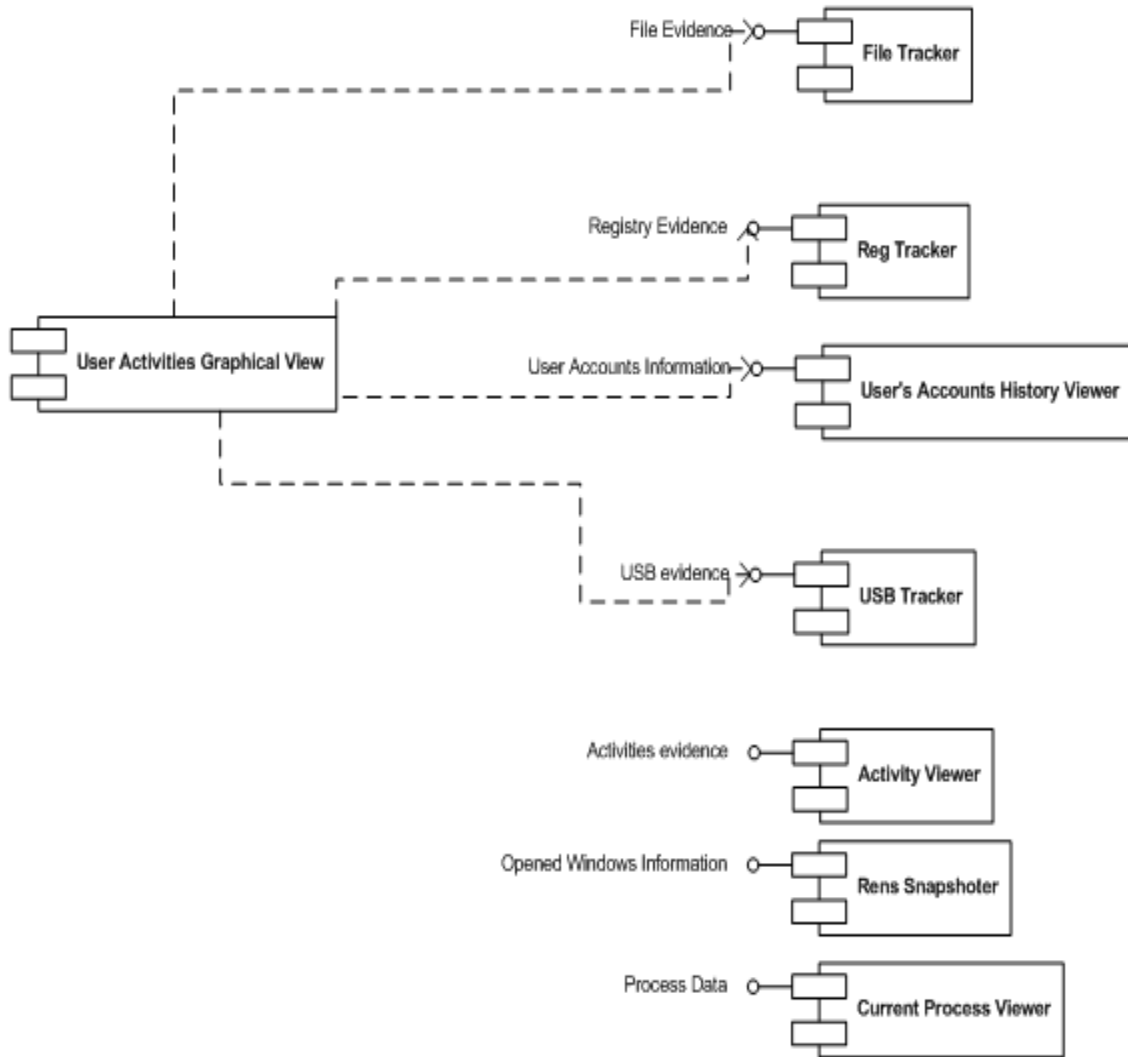
This information can be extracted from these information domains (locations) through different Windows APIs or by performing System level programming to fetch data from File System.

This extracted information can be stored in different data storage items like File or Registry Objects. The major data of all subcomponents of Win4Tech is stored in Data Tables.



#### 4.7 Component Design.....

Component diagram of Win4Tech depicts how its [components](#) are wired together to form the whole software system.



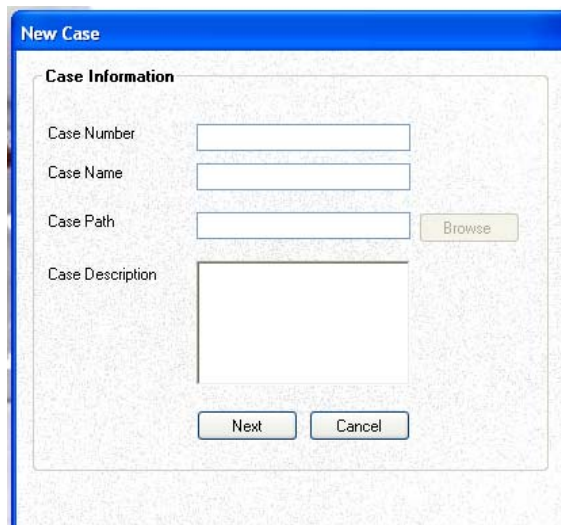
*Figure 4.17 Component Design of Win4Tech*

#### 4.8 Human Interface Design.....

## Main Interface



## Case Information



The "New Case" dialog box contains a "Case Information" section with the following fields and controls:

- Case Number:
- Case Name:
- Case Path:
- Case Description:

At the bottom of the dialog, there are two buttons:  and .

## Forensic Investigator Information

**Forensic Investigator**

**Forensic Investigator Information**

Examiner's Name

Company

Address

Phone

Fax

Email

Comments

## Filtration of Components

**ComponentSelection**

**Select Component**

Activity Viewer

Current Process Viewer

File Tracker

Reg Tracker

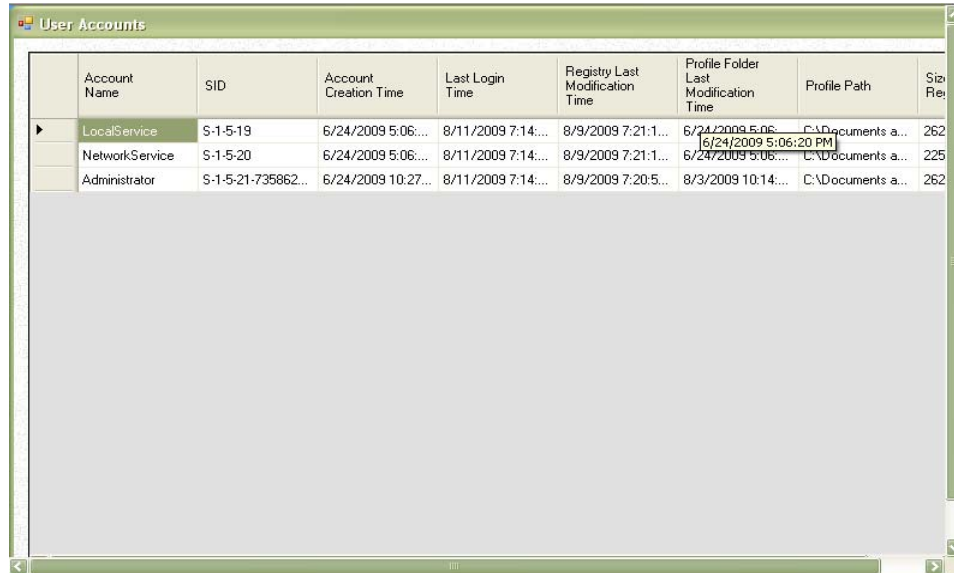
Rens Snapshorter

USB Tracker

Users Accounts Viewer

## Users' Account History Viewer

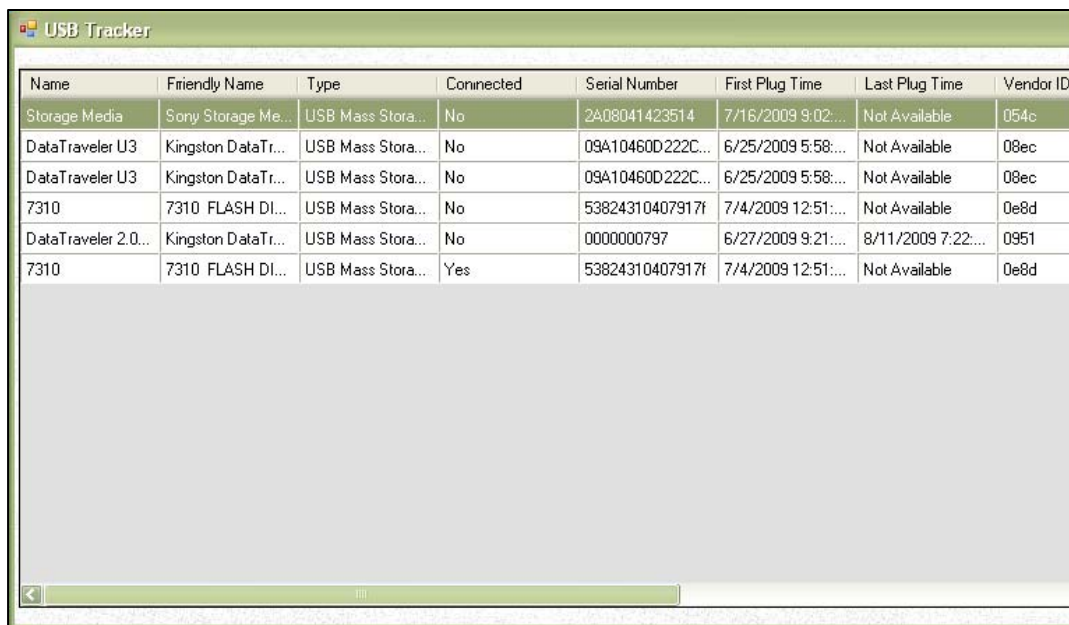
Users' account history viewer generates a report showing all the account present on the system.



Account Name	SID	Account Creation Time	Last Login Time	Registry Last Modification Time	Profile Folder Last Modification Time	Profile Path	Size
LocalService	S-1-5-19	6/24/2009 5:06:...	8/11/2009 7:14:...	8/9/2009 7:21:1...	6/24/2009 5:06:...	C:\Documents a...	262
NetworkService	S-1-5-20	6/24/2009 5:06:...	8/11/2009 7:14:...	8/9/2009 7:21:1...	6/24/2009 5:06:20 PM	C:\Documents a...	225
Administrator	S-1-5-21-735862...	6/24/2009 10:27...	8/11/2009 7:14:...	8/9/2009 7:20:5...	8/3/2009 10:14:...	C:\Documents a...	262

## USB Tracker

USB Tracker generates a report showing information about all the plugged USBs on the system, after operating system installation.



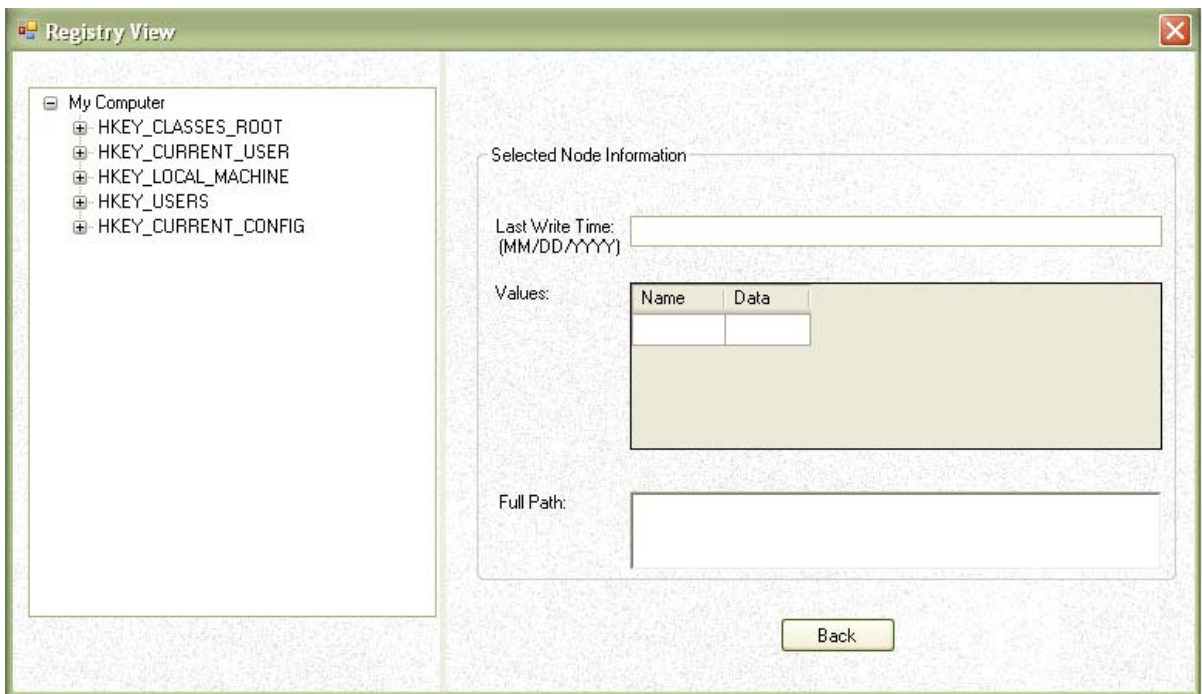
Name	Friendly Name	Type	Connected	Serial Number	First Plug Time	Last Plug Time	Vendor ID
Storage Media	Sony Storage Me...	USB Mass Stora...	No	2A08041423514	7/16/2009 9:02:...	Not Available	054c
DataTraveler U3	Kingston DataTr...	USB Mass Stora...	No	09A10460D222C...	6/25/2009 5:58:...	Not Available	08ec
DataTraveler U3	Kingston DataTr...	USB Mass Stora...	No	09A10460D222C...	6/25/2009 5:58:...	Not Available	08ec
7310	7310 FLASH DI...	USB Mass Stora...	No	53824310407917f	7/4/2009 12:51:...	Not Available	0e8d
DataTraveler 2.0...	Kingston DataTr...	USB Mass Stora...	No	0000000797	6/27/2009 9:21:...	8/11/2009 7:22:...	0951
7310	7310 FLASH DI...	USB Mass Stora...	Yes	53824310407917f	7/4/2009 12:51:...	Not Available	0e8d

## Reg Tracker

It generates overall tree structure of the registry, provides information regarding subkeys, last write time of all registry keys and registry key values. It also provides user an option to acquire information of his interest via entering the particular registry hive.



If user selects to specify parameters then the following window will appear:



Input\_Search\_Parameters

Select one of the following input options

Enter the key name by yourself

Select from one of the root keys

Enter the key name:

Select the root key:

Select the start date (MM/DD/YYYY):

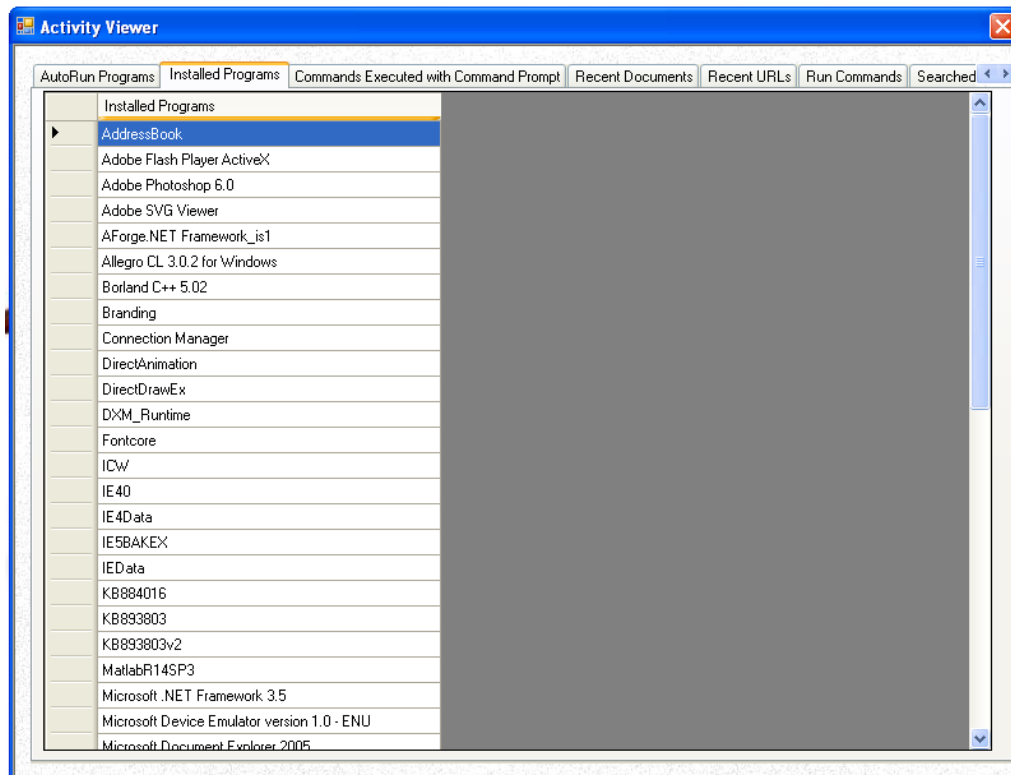
Select the end date (MM/DD/YYYY):

Track

After tracking according to the specified criteria a report is generated, containing all registry keys which satisfy the proposed criteria.

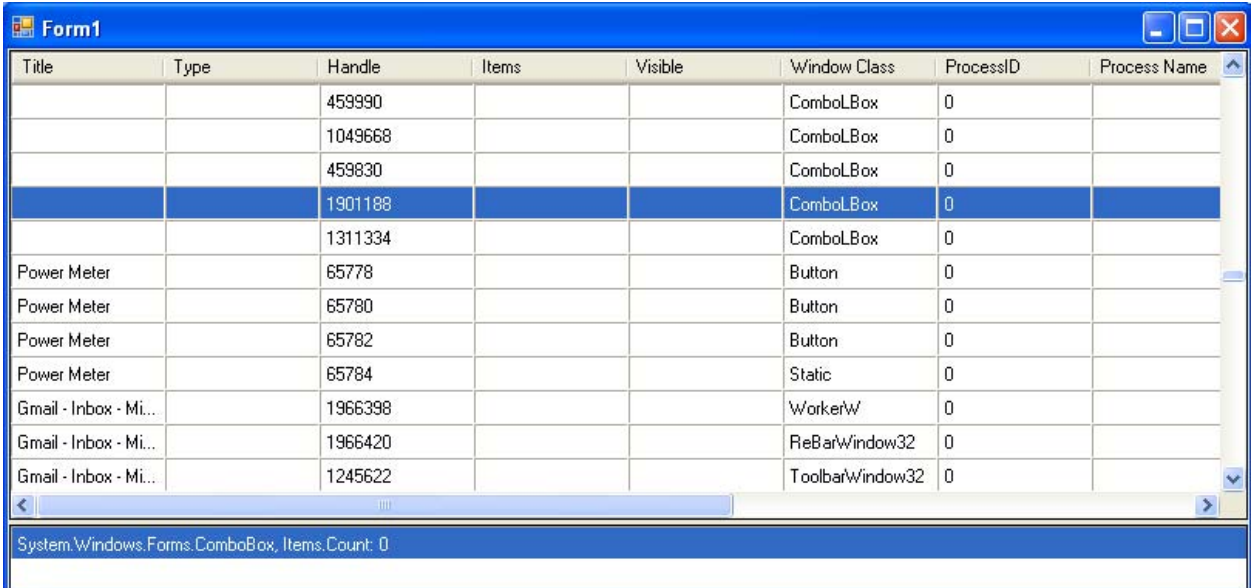
## Activity Viewer

This component will generate a report showing information regarding users' activities on a system such as recent files viewed, recent accessed URLs, the commands run from start->run, the search items previously entered by the user into Windows Default Search, the programs present at the system start up and the program list which runs with the execution of command processor.



## Rens Snapshoter

This component will generate a report displaying the data associated with opened windows e-g standard list views, text boxes etc.

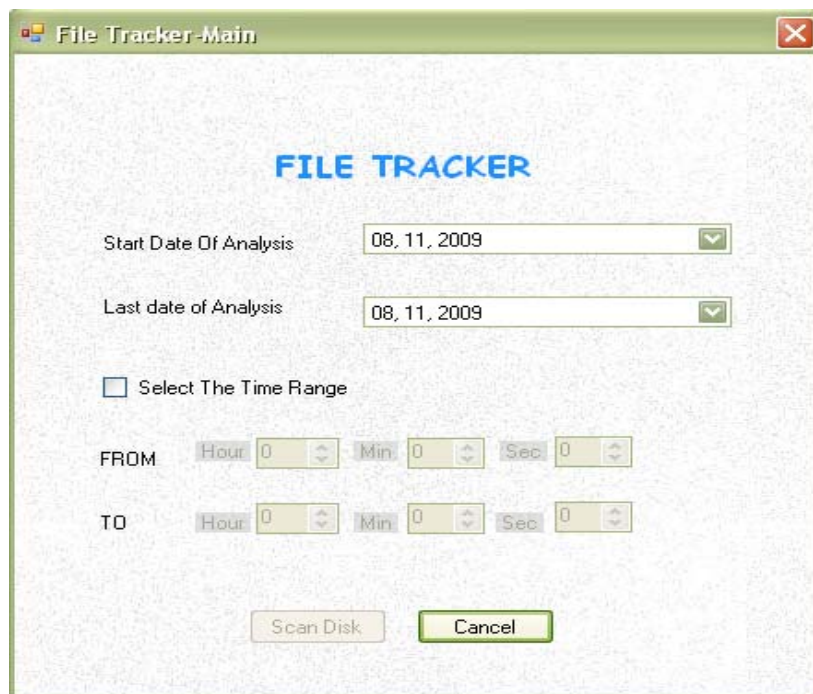


The screenshot shows a window titled 'Form1' containing a table with the following columns: Title, Type, Handle, Items, Visible, Window Class, ProcessID, and Process Name. The table lists various system components and applications, with the row for 'Gmail - Inbox - Mi...' selected. Below the table, a status bar displays the text 'System.Windows.Forms.ComboBox, Items.Count: 0'.

Title	Type	Handle	Items	Visible	Window Class	ProcessID	Process Name
		459990			ComboBox	0	
		1049668			ComboBox	0	
		459830			ComboBox	0	
		1901188			ComboBox	0	
		1311334			ComboBox	0	
Power Meter		65778			Button	0	
Power Meter		65780			Button	0	
Power Meter		65782			Button	0	
Power Meter		65784			Static	0	
Gmail - Inbox - Mi...		1966398			WorkerW	0	
Gmail - Inbox - Mi...		1966420			ReBarWindow32	0	
Gmail - Inbox - Mi...		1245622			ToolBarWindow32	0	

## File Tracker

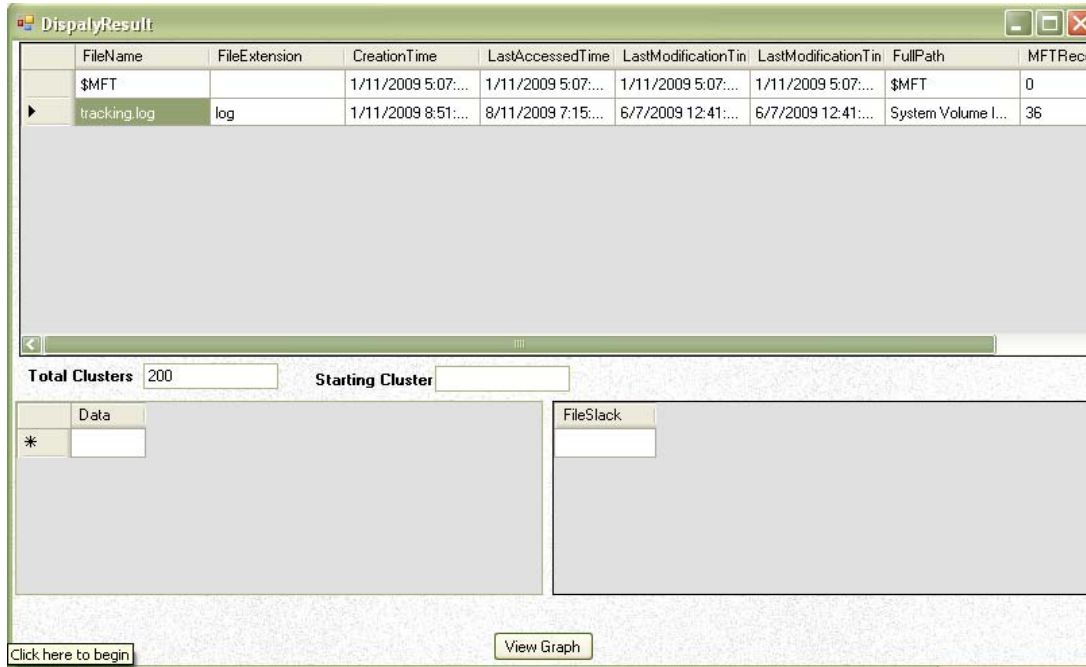
This Component generates a report providing information about files on a particular drive. The files can be filtered according to date and time.



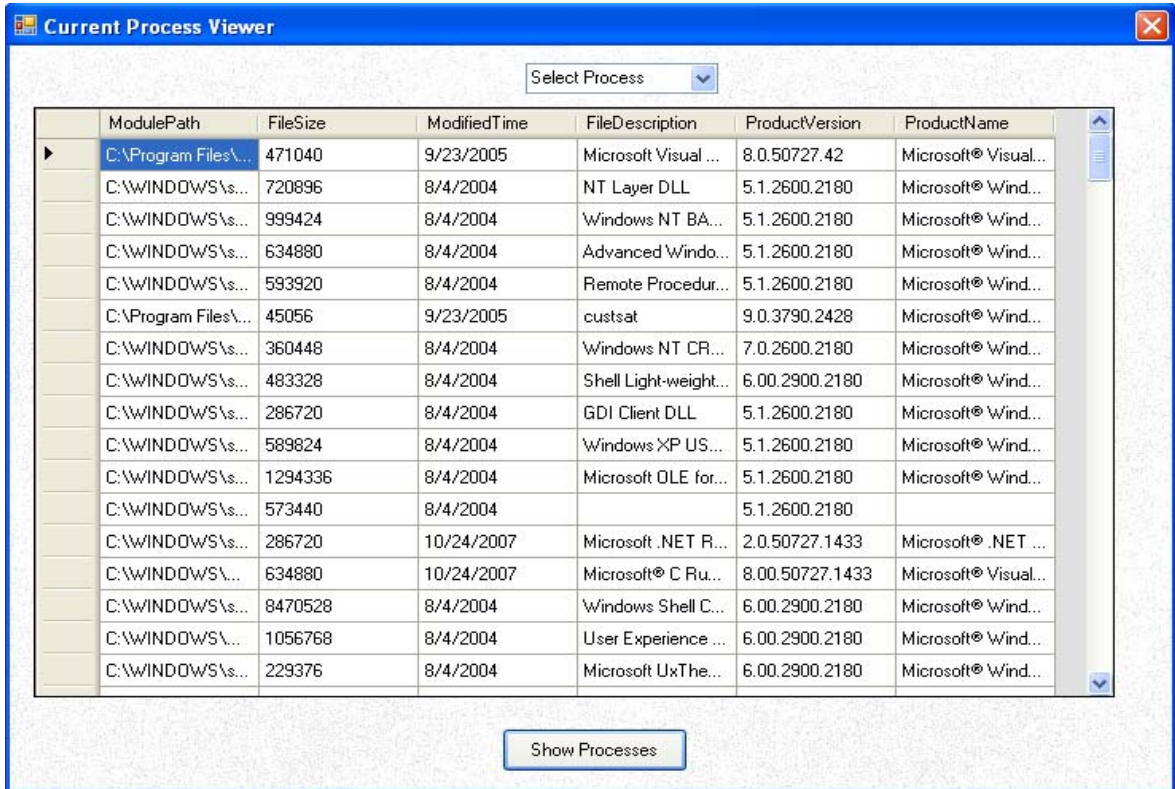
The screenshot shows the 'File Tracker-Main' window. It features a title bar with the text 'File Tracker-Main' and a close button. The main area has a light blue background with the text 'FILE TRACKER' in blue. Below this, there are two date selection fields: 'Start Date Of Analysis' and 'Last date of Analysis', both set to '08, 11, 2009'. A checkbox labeled 'Select The Time Range' is currently unchecked. Below the checkbox, there are two rows of time selection fields: 'FROM' and 'TO'. Each row has three spinners for 'Hour', 'Min', and 'Sec', all set to '0'. At the bottom of the window, there are two buttons: 'Scan Disk' and 'Cancel'.



The data and file slack of any particular selected file can be viewed.

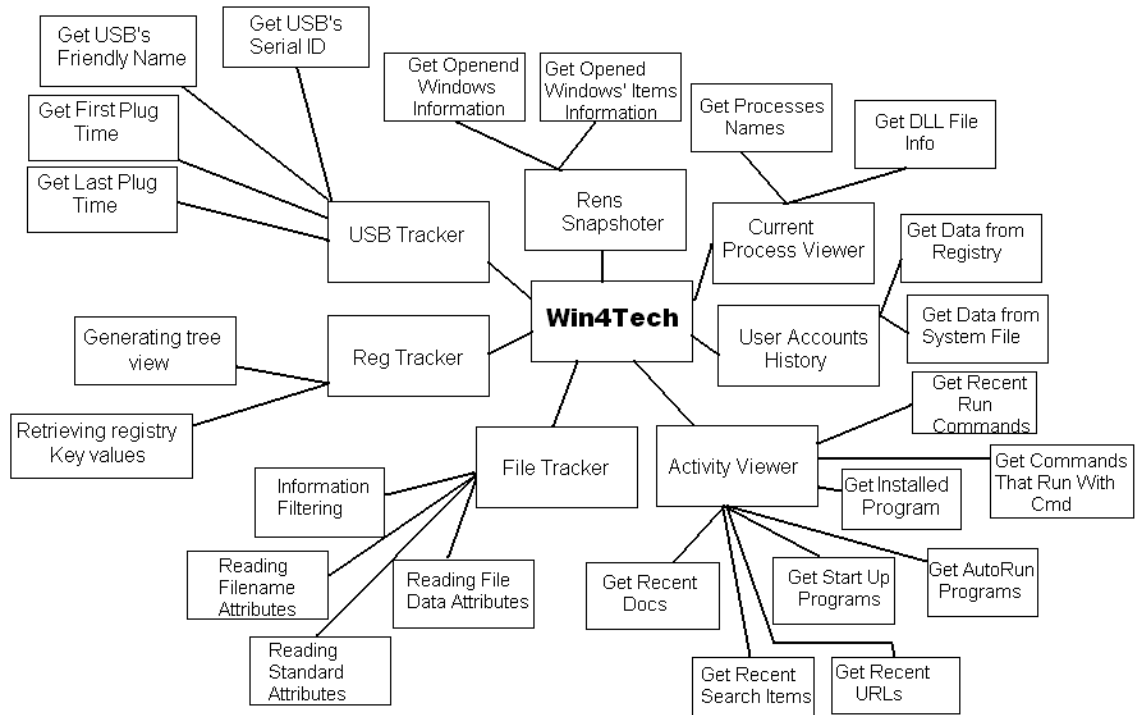


## Current Process Viewer



## 4.9 Graphical Model

### Structural Model



*Figure 4.18 Structural Model of Win4Tech*

# Object Models

Object model is developed in **RATIONAL ROSE** software.

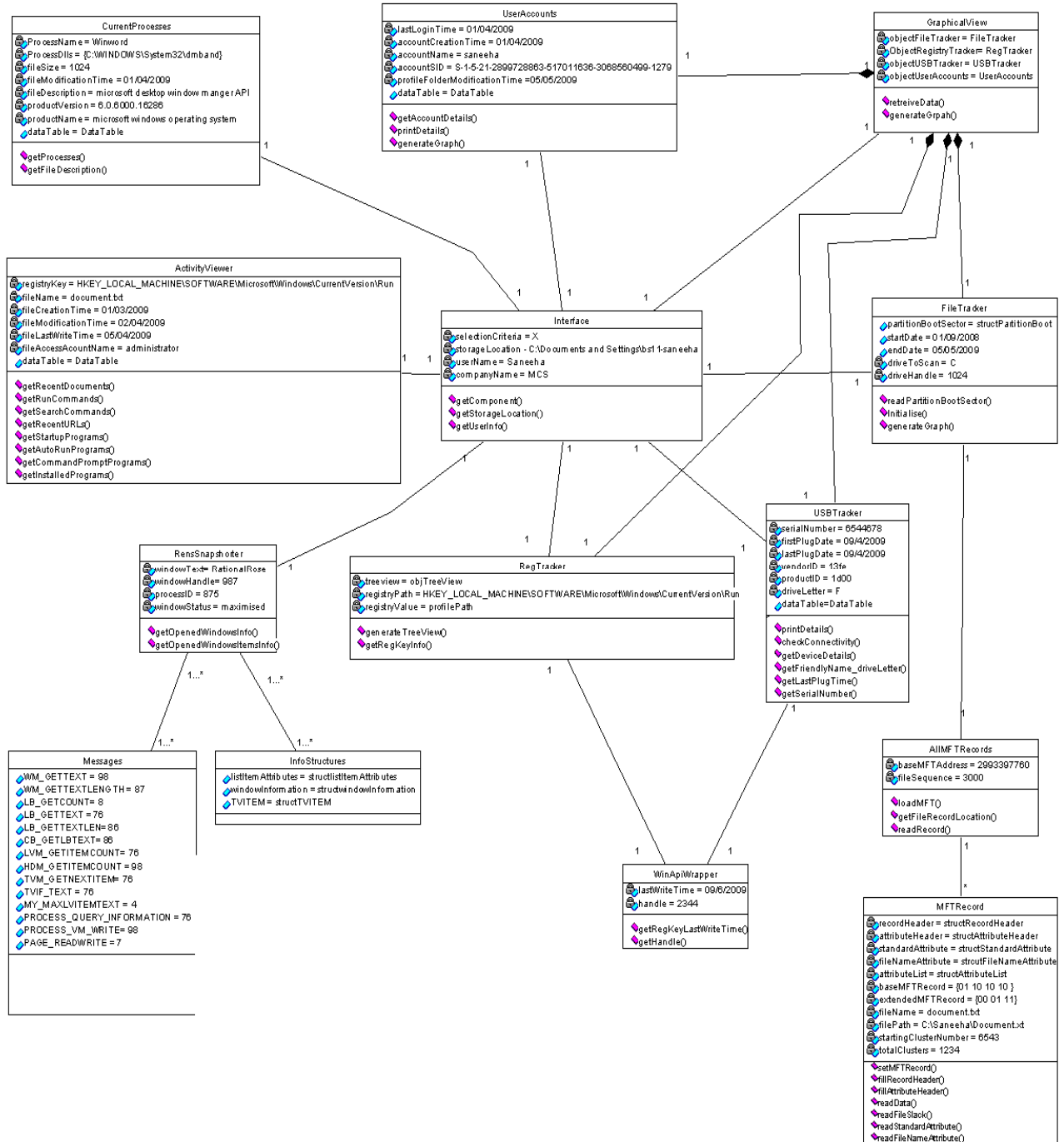
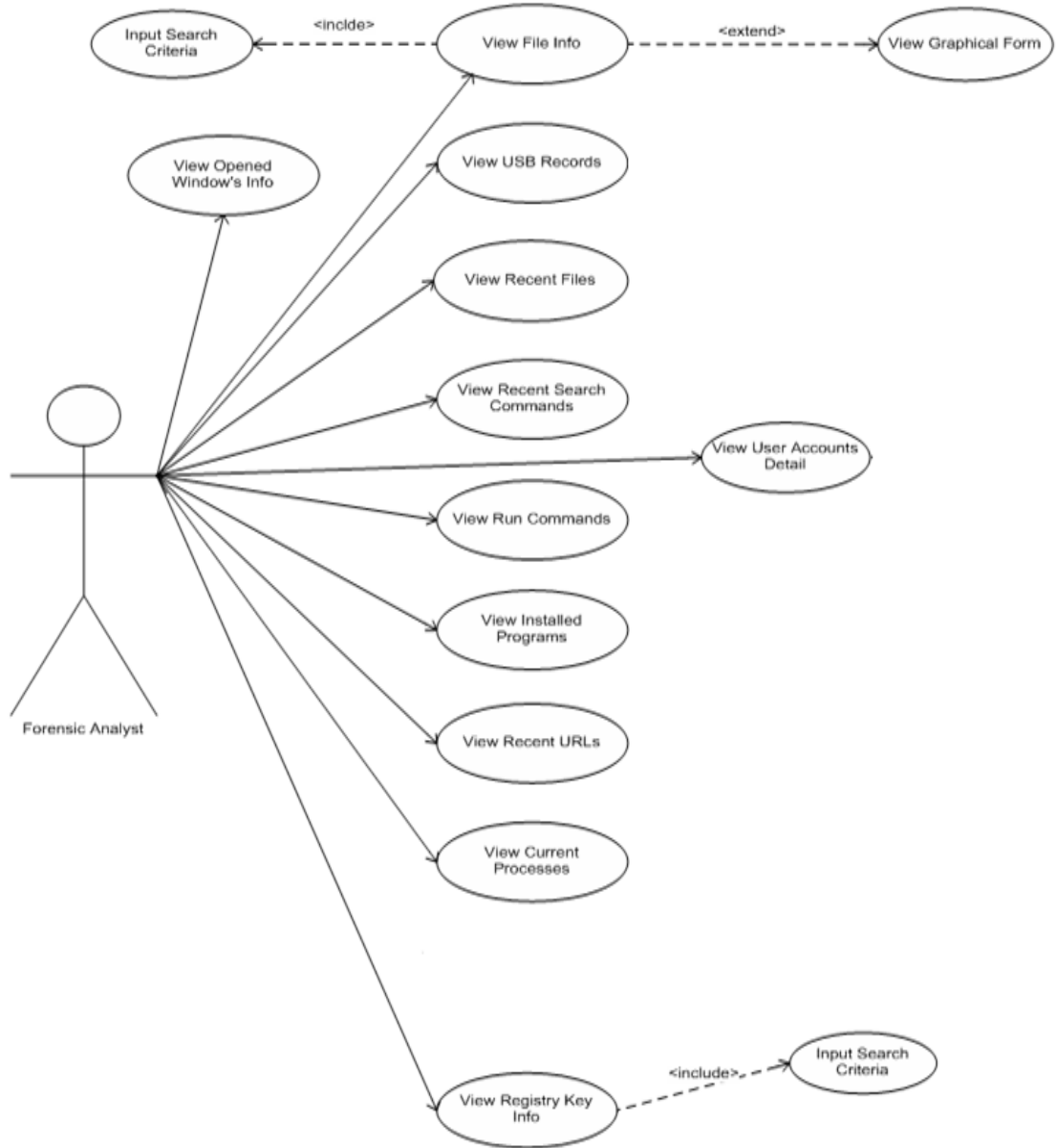


Figure 4.19 Object Model of Win4Tech

## 4.10 UML Based Design

### Use Case Diagram



*Figure 4.20 Use Case Diagram of Win4Tech*

# Class Diagram

Class diagram is developed in **RATIONAL ROSE** software.

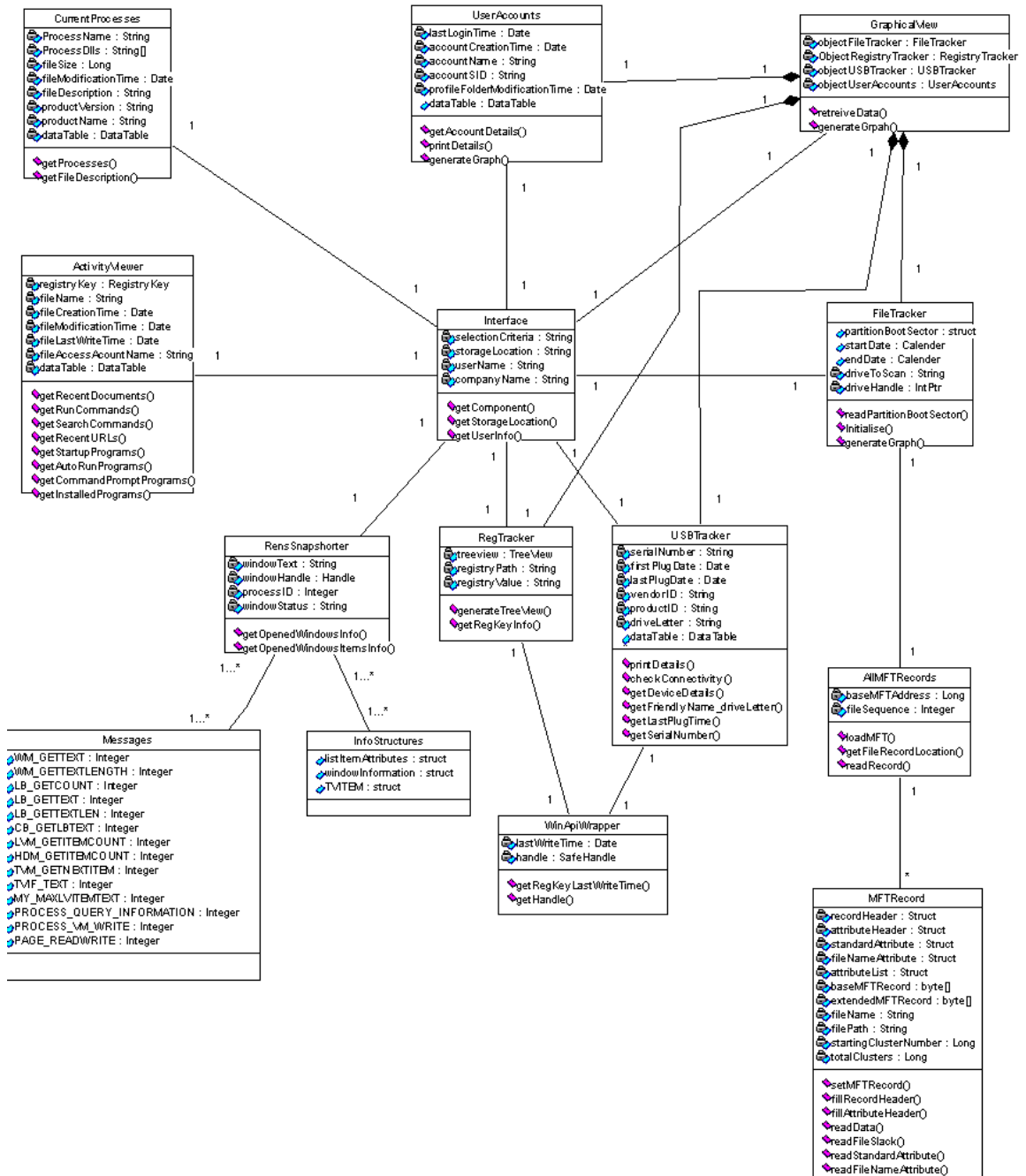


Figure 4.21 Class Diagram of Win4Tech

# Interaction Diagram

## Collaboration Diagram

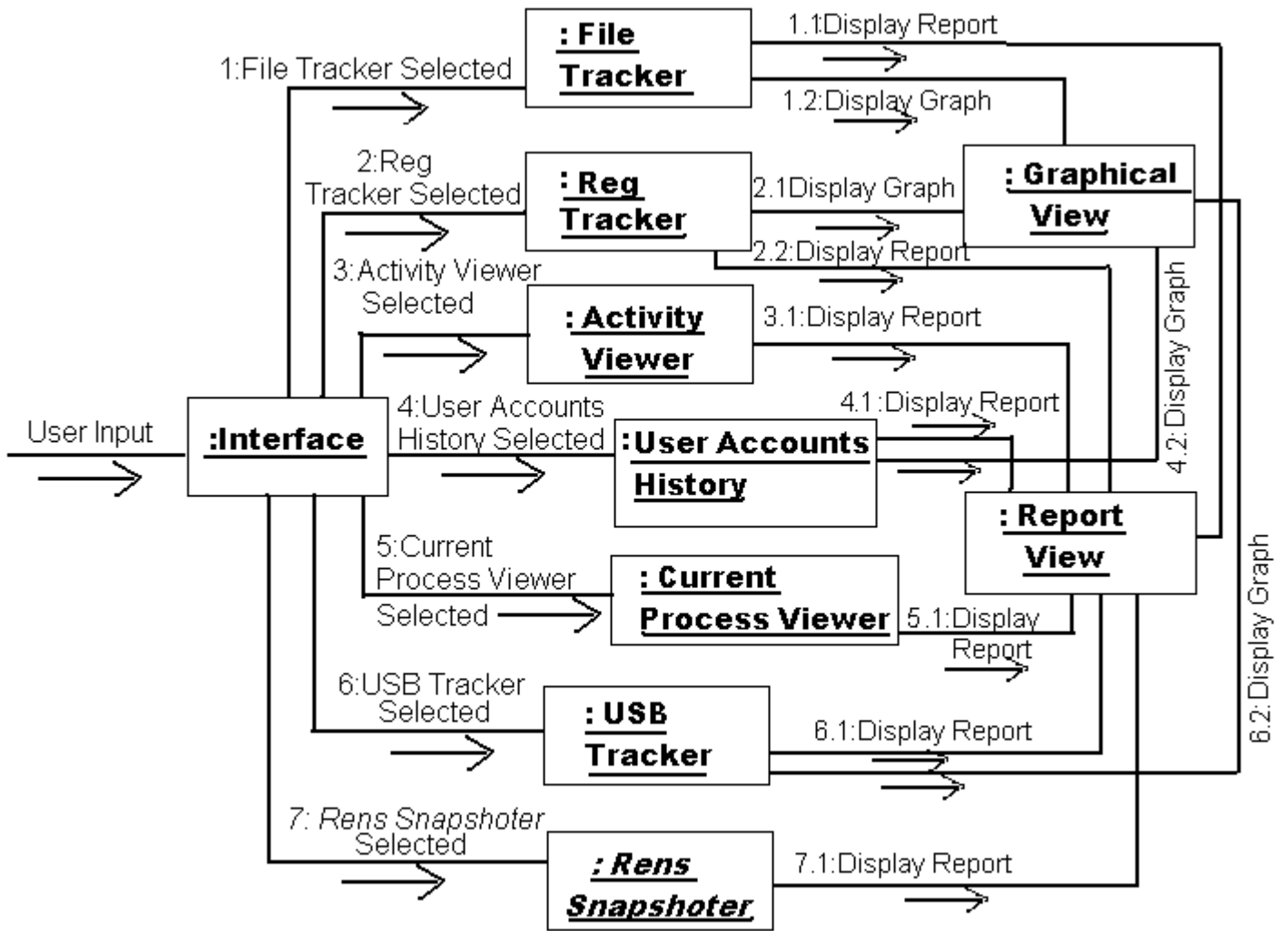


Figure 4.22 Collaboration Diagram of Win4Tech

## Sequence Diagram

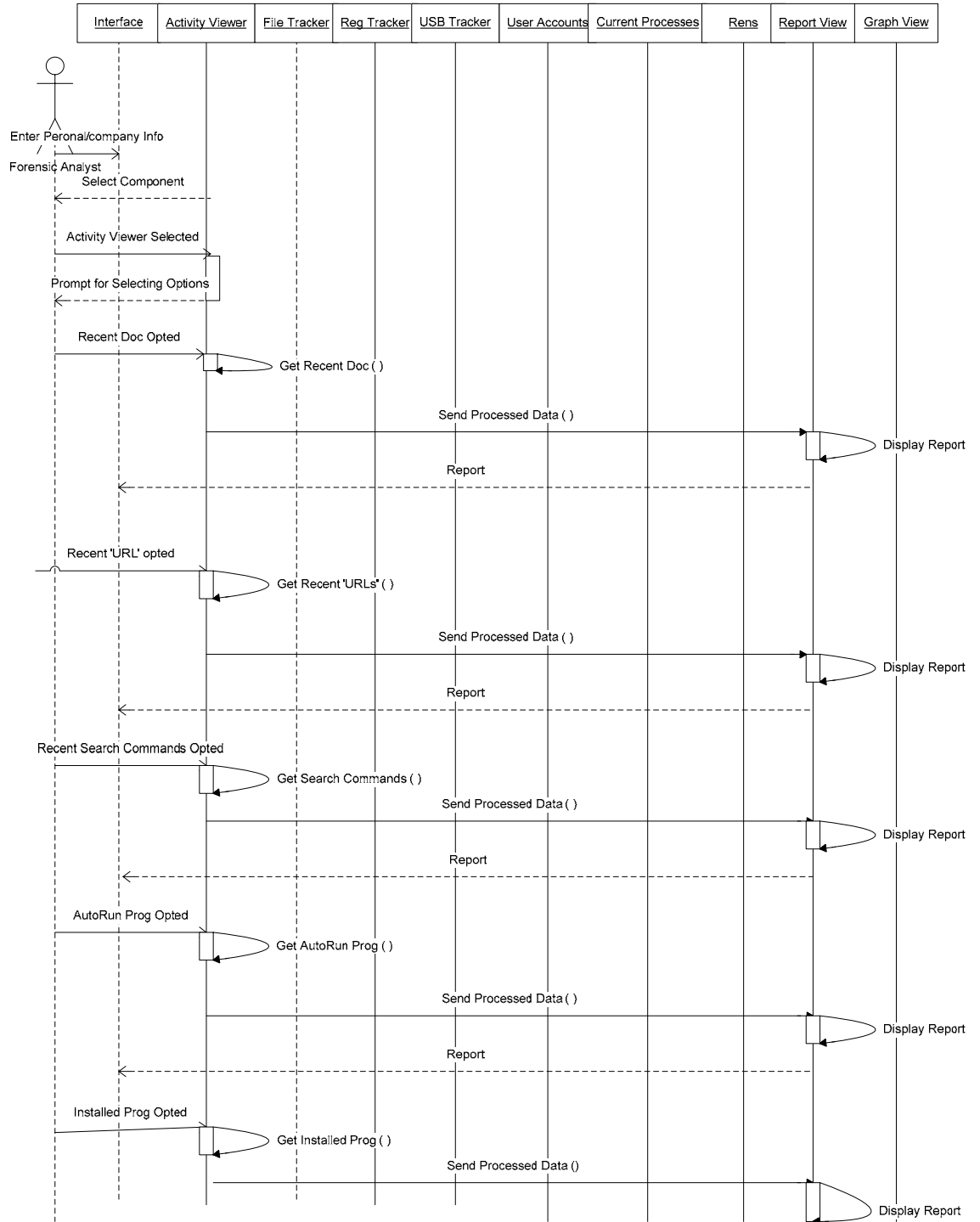
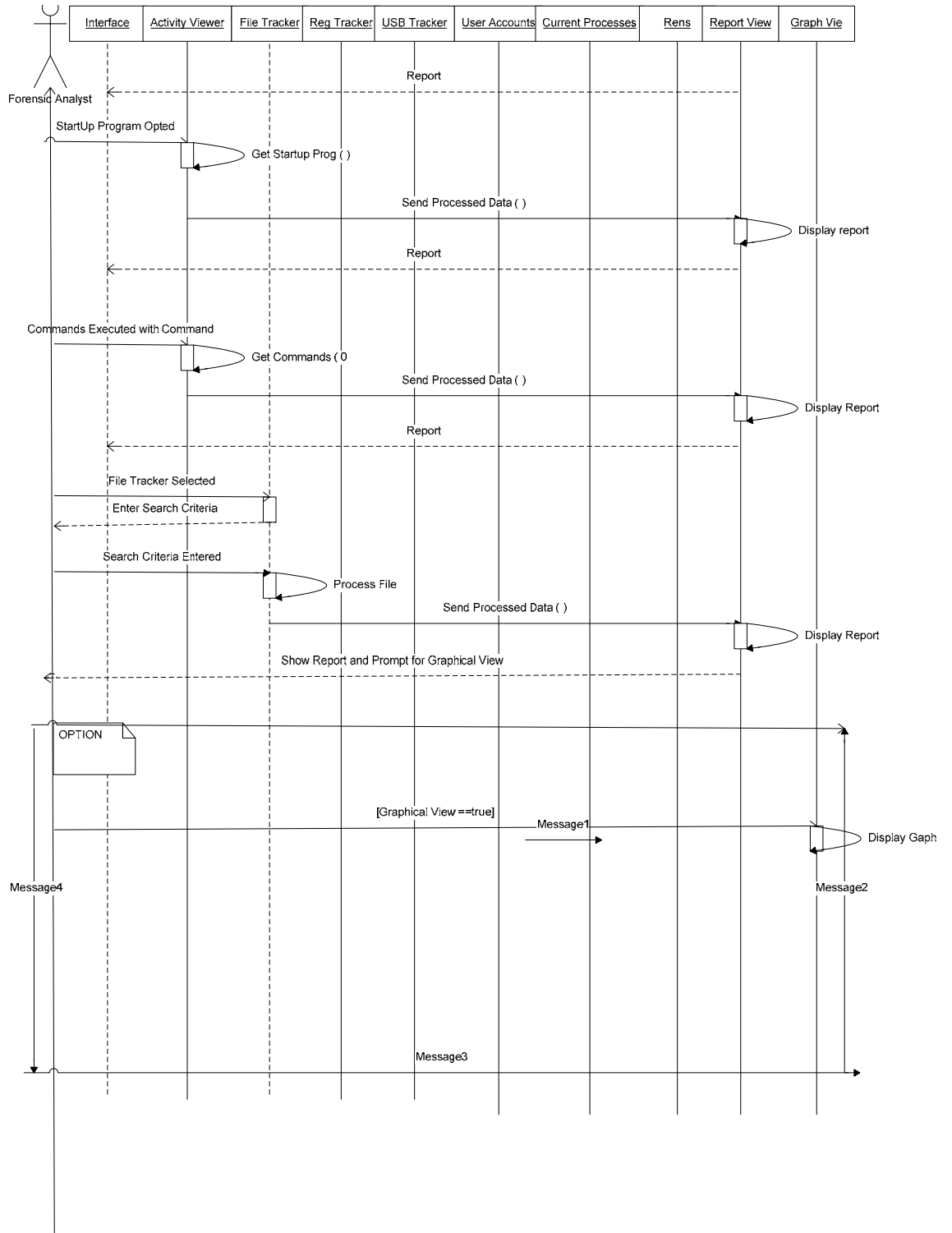
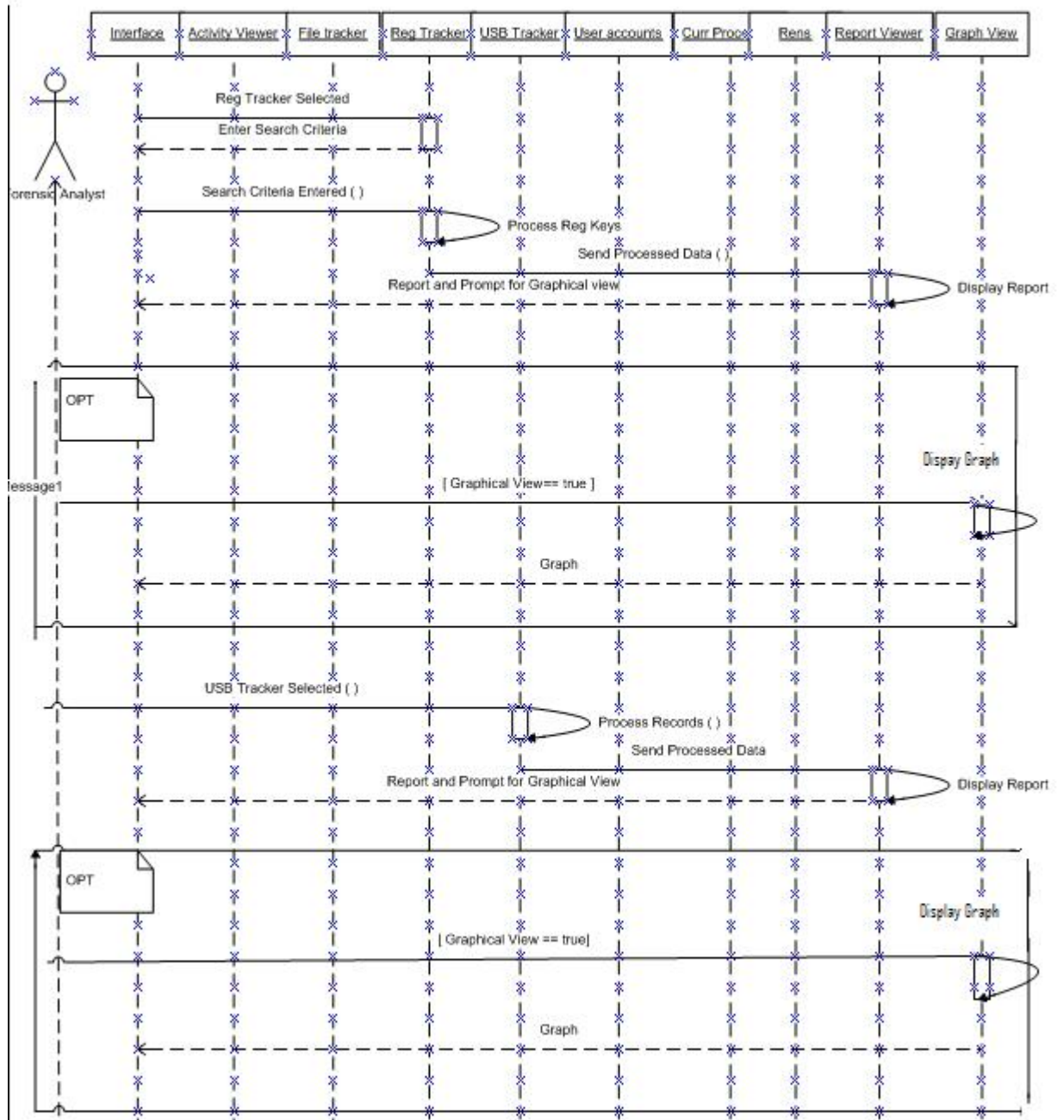
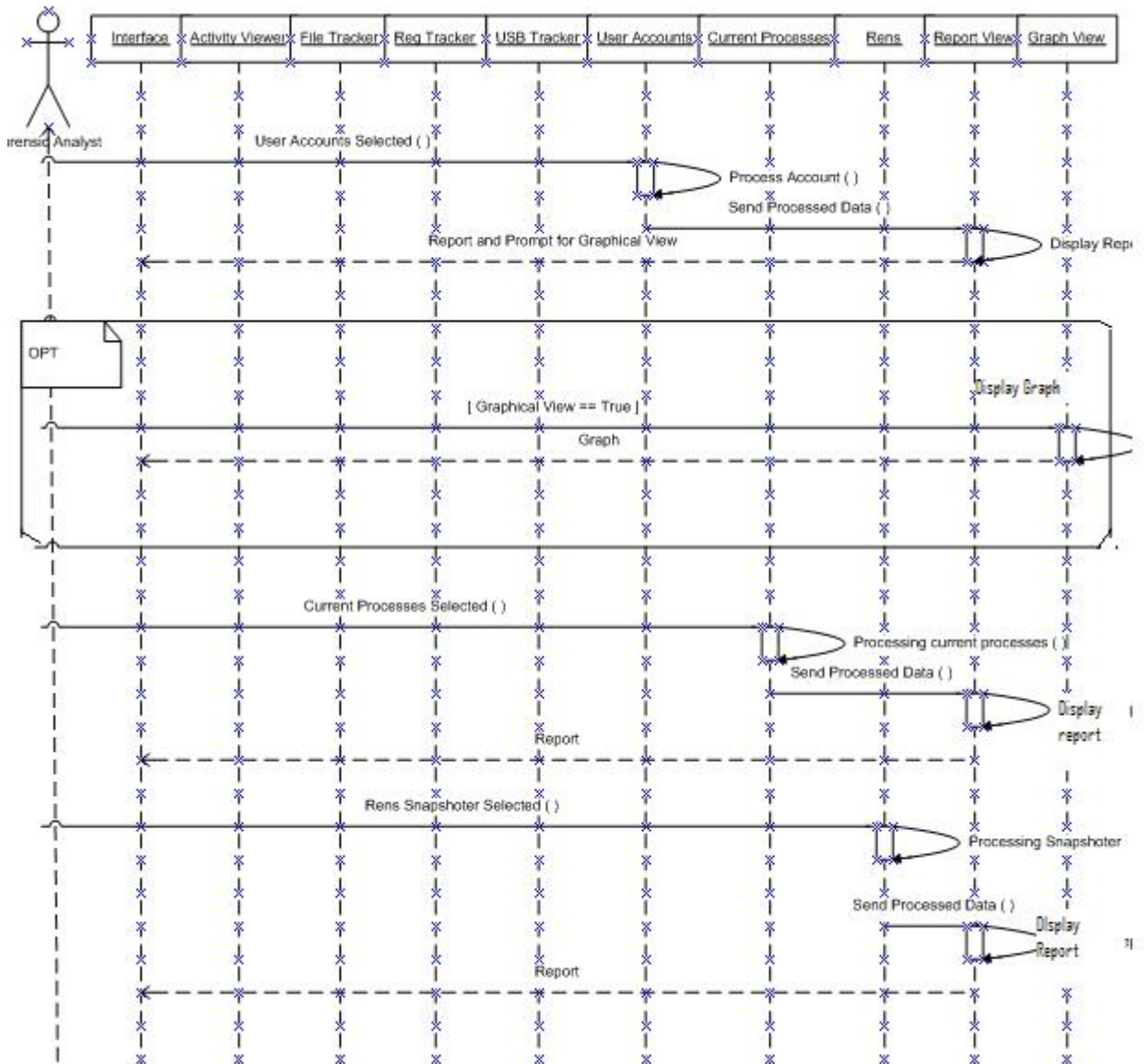


Figure 4.23 Sequence Diagram of Win4Tech









## Implementation Details

### 5.1 Introduction

This chapter deals with implementation details of Win4Tech. Win4tech is further divided into seven subcomponents. Therefore, coding specifications of each component are discussed separately in this chapter.

### 5.2 Components' Implementation Details

#### Activity Viewer

##### Implementation Details of Recent Documents

Recent documents are those files which are recently viewed or accessed in a system. They are retrieved from two data sources; one of these data sources is a particular registry key's name and data and the second source is system's folder "**Recent**". It can be found out inside "<Operating system drive letter>:\\Documents and settings". This folder contains all recent files, their pathname, MAC times and the account information through which they were accessed.

From registry, these files can be retrieved using registry's built-in functions *OpenSubKey* (*key\_name*) and *GetValueNames()* and from the **Recent** folder, recent files can be retrieved using the function *GetDirectories(path)* in order to reach to the **Recent** folder of every login account in a system and then via using the function *GetFiles(complete\_path)*, recent files along with their information can be extracted. The MAC times of the files can be taken out by making an instance of type *FileInfo* and access its attributes of *CreationTime*, *LastAccessTime*, *LastWriteTime*.

All of these information can be stored in datatable's (rows and columns) and then for the purpose of information display, datagridviews are used to pick the data stored in the datatables.

### **Implementation Details of Recent URLs**

Recent URLs typed in the Address field of Internet Explorer can be retrieved from a registry key ***HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\TypedURLs***. This particular registry key contains a list of 25 recent URLs. The most recent URLs are shown at the top of the list. This arrangement of URLs can be useful in forensic examination.

These URLs can be extracted using two registry functions ***OpenSubKey(key\_name)*** and ***GetValueNames()***. These URLs are stored in datatables and then displayed in a datagridview.

### **Implementation Details of Search Commands entered into Windows Default Search**

Searched files and folders fed into Windows Default Search can be obtained from the registry key ***HKCU\Software\Microsoft\Search Assistant\ACMr\5603***. Furthermore, the phrases and words searched within the files, entered in Windows Default Search can be extracted from registry key ***HKCU\Software\Microsoft\Search Assistant\ACMr\5604***.

Registry functions ***OpenSubKey(key\_name)*** and ***GetValueNames()*** are used to retrieve this information. This information is stored in datatables and then displayed in a datagridview.

### **Implementation Details of Programs Present At the System Start Up**

Programs at system startup can be obtained from system folder “***Startup***”. This folder is present in “<Operating system drive letter>:\Documents and settings\All Users\Start Menu\Programs”.

The information extracted from this folder includes startup programs’ names, their paths and last access times.

This information can be extracted from this folder by using the function ***GetFiles(complete\_path)***. The last access time of the program can be taken out by making an instance of type ***FileInfo*** and access its attribute of ***LastAccessTime***.

This information is stored in datatable’s (rows and columns) and then displayed in datagridviews.

### **Implementation Details of AutoRun Programs**

AutoRun programs can be obtained from the registry key *HKLM\Software\Microsoft\Windows\CurrentVersion\Run*.

Registry functions *OpenSubKey(key\_name)* and *GetValueNames()* are used to retrieve this information. This information is stored in datatables and then displayed in a datagridview.

### **Implementation Details of Command Prompt Execution Programs**

Programs that run with command prompt execution can be obtained from the registry key *HKLM\Software\Microsoft\Command Processor*.

Registry functions *OpenSubKey(key\_name)* and *GetValueNames()* are used to retrieve this information. This information is stored in datatables and then displayed in a datagridview.

### **Implementation Details Of List of Programs Installed On a System**

Installed programs on any system can be obtained from the registry key *HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall*

Registry functions *OpenSubKey(key\_name)* and *GetValueNames()* are used to retrieve this information. This information is stored in datatables and then displayed in a datagridview.

### **Implementation Details of Current Process Viewer**

The method *Process.GetProcesses* retrieves the list of the current processes running on the system. *ObjModulesList.Modules* loads the dynamic link library files being used by the running processes. *foreach (ProcessModule objModule in ObjModules)* iterates through the modules.

*String strFileSize = GetValidString (objModule.ModuleMemorySize.ToString ( ));*

This function will return the size of files in string format.

*String strFileModificationDate =GetValidString (objFileInfo.LastWriteTime.ToShortDateString ( ));*

This function will return the modification date of the DLL files in string format.

```
String strFileDescription = GetValidString  
(objModule.FileVersionInfo.FileDescription.ToString ( ));
```

This function will return the file description of the DLL files also in string format.

```
String strProductName = GetValidString (objModule.FileVersionInfo.ProductName.ToString  
( ));
```

This function will return the product name that is associated with the DLL file also in string format.

```
String strProductVersion = GetValidString  
(objModule.FileVersionInfo.ProductVersion.ToString ( ));
```

This function will return the product version of the product also in string form.

### **Implementation Details of File Tracker**

File Tracker retrieves information about the files present on the system. It retrieves information from the drives formatted with NTFS file system. For accessing the disk partition, the API “CreateFile()” is used. The handle returned by the API is used to open a file stream into the drive. The first sector of the drive consists of “Partition Boot Sector”. The information extracted from the “Partition Boot Sector” includes

- Cluster size
- Number of sectors in one cluster
- Base address of MFT

In NTFS information about all files and directories is present in MFT (Master File Table), therefore the base address is very important for accessing the MFT. To extract information about the files we need to read the MFT. One boot sector is read, the file pointer is moved to the MFT table.

### ***Reading MFT***

As MFT is also a file, therefore its record is also present in MFT table. In the MFT table first record belongs to itself. It contains information about the MFT table. The

information about all files and directories is present in the \$DATA attribute of this record. Therefore the program reads all the offsets present in the \$DATA attribute. If the MFT table is fragmented then multiple offsets will be present. All the offsets are read and stored in an array.

### ***Reading All Records***

Now when all the offsets are read, pointer is moved to all fragments one by one and all records present in that fragment are read. The size of one record is fixed; therefore that number of bytes is stored in a byte array. Then the byte array is sorted into different structures in order to retrieve the relevant information. The structures made are:

### ***\$Standard Attribute***

The Standard Attribute contains information regarding modification, last access and creation (MAC) times of a particular file or a directory. The API `DateTime.FromFileTime()` is used for converting the system time to standard datetime format. The attribute also includes information about the property associated with a file like read only, system, hidden, sparse, temporary, reparse point, not-content indexed, device, archive, compressed, encrypted etc. Different codes are assigned to each property; the code is read and matched to assess the property of the file.

### ***\$Filename Attribute***

The File Name Attribute contains information regarding filename, reference to the parent directory record and MAC times. The file name is encoded using Unicode encoding. The reference to the parent directory is used recursively for establishing the full path of the file. The references may be present of different MFT fragments, therefore the MFT reference number is used to calculate the fragment first, and then the record in that fragment.

### ***\$Data attribute***

The header of \$DATA attribute is read for retrieving the size of file, disk space occupied by the file and compression status of file, etc. The attribute itself contains file

data. If the data is resident then the attribute itself contains the data, no offsets are present. But if data is non resident, offsets to the data are present in data runs. Data runs contain information about the starting Cluster number of the file data and total number of clusters occupied by the file after the starting cluster. As the length of the clusters assigned to a file may vary, therefore NTFS does not allocate fixed bytes for the storage of total cluster number. Also the size for storing starting cluster number is not fixed. In the data run the first byte contains information about the number of bytes allocated for the storage of starting cluster and total clusters. Therefore after reading this value, respective number of bytes is read. If the file is fragmented then multiple values for starting cluster number and total clusters occupied will be present in the file's data runs. The last value of the run is indicated when the byte contains all zeroes. For retrieving file data the complete data run needs to be read. For each value in the data run, pointer is moved to that location on the disk and data is read cluster by cluster.

#### ***\$AttributeList Attribute***

If multiple records are associated with one file, then the information about the extended records is present in the \$AttributeList attribute. It contains information about the MFT record number which is allocated to the file. It also describes which attribute is present in the extended record. Once the new MFT number is read, pointer is moved to the new record and attributes are read from this record.

#### ***File Slack***

If file size is not multiple of cluster size, then some disk space is left at the end of the last cluster, which is not occupied by the file's data. This disk space is called ***file slack***. In order to read file slack, pointer is moved to the last cluster of the file. In the header of \$DATA attribute, information is stored about the logical and physical size of the file. Subtracting the two values gives the size of file slack. From the last cluster bytes equal to the file slack size are read. File slack may contain data hidden by some criminal or content of some deleted file.



## **Implementation Details of Reg Tracker**

Implementation details are written sub module wise:

### ***Registry View***

A tree view of the registry is displayed by adding the root keys of the registry as tree nodes in the tree view. An event is generated when the user clicks on the + (expand) in the tree view. The code which is triggered when this event occurs works by further getting the sub key names of the key which is clicked (for expanding) and adding them as the nodes of the clicked key. This enhances the performance of the program because the whole tree view is not generated in the start. When the user clicks on a particular key, the program displays last write time, values (if any) and full path of the key. The program obtains the full path using the Name attribute of the registry key. Last write time is obtained by using the API "RegQueryInfoKey" which exists in "advapi32.dll". Similarly values (if any) are obtained by first finding out the kind of the registry values using ***RegistryKey.GetValueKind*** function and then using ***RegistryKey.GetValue*** function. The values are then converted into hex format for display.

### ***Search Specific View***

The registry key entered/selected by user is traversed recursively and last write time of each key is checked against the time range selected by the user. The key satisfies the criteria if the last write time falls in the time range.

## **Implementation Details of Rens Snapshoter**

Opened windows are enumerated using EnumWindows api which is defined in "user32.dll". This api provides handles of the opened windows. Opened window items (buttons, list boxes etc) are found by using EnumChildWindows api, defined in "user32.dll". Process id of the process which has opened a particular window is obtained by using GetWindowThreadProcessId which is defined in "user32.dll". GetProcessById function of System.Diagnostics.Process class is used to access the process by passing the process ID. MainModule.FileName attribute of the process provides Process Module File Name. GetVersionInfo function of System.Diagnostics.FileVersionInfo class is used to

obtained a FileVersionInfo object. The attributes of the object which provide company, description, productName and version information are CompanyName, FileDescription, ProductName and FileVersion respectively. Windows visibility information is found by using IsWindowVisible api which is defined in user32.dll. GetClassName API, defined in user32.dll is used to obtain the class name of the windows. The content of the window items is found by sending relevant messages to the window using *SendMessage* API, defined in "user32.dll". GetWindowTextLength API defined in user32.dll is used to get the length of text of button and edit window. GetWindowText API defined in user32.dll is used to further get the text of button and edit window. Similarly to retrieve the text of comboboxes, SendMessage API is used. CB\_GETCOUNT message is first sent to window using SendMessage API to get the number of items in the combobox. CB\_GETLBTEXTLEN and CB\_GETLBTEXT messages are sent to the combobox window for retrieving the length and text of each item respectively. To retrieve the number of items in a listview window, LVM\_GETITEMCOUNT message is sent to the listview window. LVM\_GETITEMTEXT message is sent to listview window to retrieve the text of each item in the listview. LVM\_GETHEADER message is sent to the listview window to get the header of the listview. HDM\_GETITEMCOUNT is sent to the listview window to count the number of columns in the header of the listview.

### **Implementation Details of USB Tracker**

Serial number of the USB is found by retrieving the name of the second level subkeys of HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR. Class, Sub Class, ParentIdPrefix and Friendly Name of the USB is obtained by getting the values of the subkeys of above key using RegistryKey.GetValue function. First Plug Time of the USB is found by obtaining the last write time of the subkey LogConf of the subkey having serial number. Drive letter is found by matching the ParentIdPrefix by the values of the key: HKEY\_LOCAL\_MACHINE\SYSTEM\MountedDevices. If a registry value contains ParentIdPrefix then the corresponding Key Value Name provides the drive letter of the USB. Last plug time of the USB is obtained by matching the subkeys of "SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}" with the serial number of the USB and then further finding the last write

time of the “Control” Subkey. Name,Description,protocol,vendor id,revision and product id are obtained by getting the values of the second level subkeys of HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Enum\USB. The value names which provide these information are LocationInformation, DeviceDesc, CompatibleIDs and HardwareID(gives vendorid,productid and revision) respectively.

### **Implementation Details of User’s Account History Viewer**

Users’ account history viewer retrieves information from windows registry and system files. Information from the registry is extracted from the following key:

*"SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\"*

This key has subkeys belonging to all accounts present on the system. The subkey name is the SID assigned to a particular user account. The value “ProfileImagePath” is present in each subkey. It contains the path of each user’s profile folder. The program extracts the SID and profile path from the registry. Using the profile path the folder of each account is accessed.

For each user account there is a NTUSER.DAT file created by the operating system. This file contains user settings and preferences. The HKCU registry hive is loaded from the NTUSER.DAT file. The NTUSER.DAT file is created when the user account is created and the user first logs in. Therefore the creation time of this file is the creation time of the user account. Every time the user logs in, the file is accessed for loading the registry hive. Therefore the last access time of the file is the last login time of the user. If any changes are made in the settings they are stored in the NTUSER.DAT file, therefore the last modification time of the file is the time of the last changes made to the registry. The size of NTUSER.DAT file is the size of the registry occupied by the settings of a particular user.

Similarly the modification time of the profile folder indicate the time at which the user last made changes to his documents folder.

## Testing and Results

### 6.1 Introduction

This chapter discusses the testing performed on the subcomponents of Win4ech in order to validate that its output results fulfill the initial requirement document.

Various test cases are designed and executed on the final implemented toolkit for this purpose of obtaining efficiency, reliability and accuracy in the final executable software.

#### Test Personnel

Parties	Contact Person	Role and Responsibilities
DS	Lec Ahmed Raza Cheema	<ul style="list-style-type: none"> <li>• Overall Supervision</li> <li>• Work stream Management</li> <li>• Review of Test Plan and Test Cases</li> <li>• Monitor testing schedule and procedure.</li> </ul>
Testing Team Lead	Chanda Gul (chandagull@yahoo.com)	<ul style="list-style-type: none"> <li>• QA Team Lead</li> <li>• Development of Test Plan and Test Cases.</li> <li>• Managing and directing testing activity</li> <li>• To Ensure Testing activity comply with project and test plan</li> <li>• Conduct testing</li> </ul>
Test Engineer	Ayesha Aslam (ayesha.nust@gmail.com)	<ul style="list-style-type: none"> <li>• Develop test cases and conduct testing</li> <li>• Validate output results against</li> </ul>

		requirements
Test Engineer	Saneeha Khalid (saneeha.nust@gmail.com)	<ul style="list-style-type: none"> <li>• Develop test cases and conduct testing</li> <li>• Validate output results against requirements</li> <li>• Submit error reports</li> </ul>
Test Engineer	Umer Khan (ummer26@hotmail.com)	<ul style="list-style-type: none"> <li>• Develop test cases and conduct testing</li> <li>• Validate output results against requirements</li> </ul>

*Table 6.1 Team Members' Responsibilities*

## 6.2 Testing Schedule

Task	Start Date	End Date	Days
Document Requirement	15-03-2009	21-03-2009	6
Document Design	9-05-2009	14-05-2009	5
Create Test Cases	20-07-2009	22-07-2009	2
Conduct walk-through of Test Plan and Test Cases	26-07-2009	28-07-2009	2
Test Case Execution	28-07-2009	30-07-2009	2
Sign off on Test	30-07-2009	31-07-2009	1

*Table 6.2 Testing Schedule*

### 6.3 Test Cases

Test Case Name / Title : Testing USB Tracker	
Test Case Number	1
Purpose	To verify USBs name, friendly name, type, serial number, first plug time, last plug time, vendor id, product id, class, sub class, protocol, drive letter against the relevant registry values
Precondition	At least one usb should ever be plugged after the recent operating system installation
Procedure	Click on USB Trakcer from Tools menu or in initial selection wizard
Expected Results	All USBs plugged which are visible in registry
Actual Results	All USBs plugged which are visible in registry
Status	Pass
Tester	Chanda Gul
Date	28-07-2009

**Table 6.3 Test case # 1**

Test Case Name / Title : Testing Reg Tracker	
Test Case Number	2
Purpose	To verify registry keys values, full path and last write time
Precondition	NIL
Procedure	Click on Reg Tracker from Tools menu or in initial selection wizard. Compare values of one or

	more keys with their expected values. Expected values can be found by using the Export option in the right click menu of a particular registry key in registry editor
Expected Results	As given by the export option
Actual Results	As given by the export option
Status	Pass
Tester	Chanda Gul
Date	28-07-2009

**Table 6.4 Test case # 2**

Test Case Name / Title : Testing Rens Snapshoter	
Test Case Number	3
Purpose	To verify that all currently opened windows are listed by the component and title and text of the windows is correctly displayed
Precondition	NIL
Procedure	Click on Rens Snapshoter from Tools menu or in initial selection wizard
Expected Results	Title of all currently opened windows and the text (if present) in the window items should be displayed
Actual Results	Title of all currently opened windows and the text (if present) in the window items are displayed
Status	Pass
Tester	Chanda Gul
Date	28-07-2009

**Table 6.5 Test case # 3**

Test Case Name / Title : Testing File Tracker	
Test Case Number	4
Purpose	To verify MAC times, compression status, path, data and slack of files
Precondition	NIL
Procedure	Click on File Trakcer from Tools menu or in initial selection wizard. A new file is created on a low level formatted disk
Expected Results	MAC times of the newly created file should match and the slack should be zero.
Actual Results	MAC times of the newly created file matched and the slack was zero.
Status	Pass
Tester	Saneeha Khalid
Date	28-07-2009

***Table 6.6 Test case # 4***

Test Case Name / Title : Testing Current Process Viewer	
Test Case Number	5
Purpose	To verify that
Precondition	At least one usb should ever be plugged after the recent operating system installation
Procedure	Click on USB Trakcer from Tools menu or in initial selection wizard
Expected Results	All USBs plugged which are visible in registry
Actual Results	All USBs plugged which are visible in



	registry
Status	Pass
Tester	Umer khan
Date	28-07-2009

**Table 6.7 Test case # 5**

Test Case Name / Title : Testing User Accounts Viewer	
Test Case Number	6
Purpose	To verify SID, Account Name, Account Creation Time, Last Login Time, Registry Last Modification Time, Profile folder's Last Modification Time, Profile Path, Size of Registry of each user account that has ever been created since the recent operating system installation
Precondition	NIL
Procedure	Click on User Accounts Viewer from Tools menu or in initial selection wizard. All the attributes are checked by creating a user Account and the dates are noted manually.
Expected Results	As noted manually
Actual Results	As noted manually
Status	Pass
Tester	Saneeha Khalid
Date	30-07-2009

**Table 6.8 Test case # 6**

Test Case Name / Title : Testing Activity Viewer	
Test Case Number	7
Purpose	To verify Recent Files, URLs Accessed, Searched Items, Auto run and Startup programs, Run Commands and Commands that run with command prompt
Precondition	NIL
Procedure	Click on Activity Viewer from Tools menu or in initial selection wizard
Expected Results	Recent Files, URLs Accessed, Searched Items, Auto run and Startup programs, Run Commands and Commands that run with command prompt as shown by the related registry keys and their values
Actual Results	Recent Files, URLs Accessed, Searched Items, Auto run and Startup programs, Run Commands and Commands that run with command prompt as shown by the related registry keys and their values
Status	Pass
Tester	Ayesha Aslam
Date	30-07-2009

*Table 6.9 Test case # 7*

## 6.4 Test Summary Report

The tests have shown the following results

- All software modules preliminary execution shows desired results and fulfill the initial requirements gathered.
- All modules integration testing illustrate no errors and exceptions.

## **Conclusion and Future Work**

This chapter concludes the overall system giving its advantages and limitations if any.

### **7.1 Conclusion**

The implemented project Win4Tech suggests various methods of investigating a computer. This forensic toolkit helps in extracting information from different evidence sources existing inside a computer system. These sources provide important record about the history of activities performed by a user on a system. Using this information, a forensic analyst can examine users' tasks and activities on a particular system.

This toolkit is useful for making the cyber world secure and crime free. It can be a prolific forensic toolkit for Law Enforcement Agencies for the purpose of confining the criminals.

### **7.2 Future Work**

The developed toolkit has the functionality to display NTFS (file system) file slack but further work on this particular area can be performed to retrieve hidden data from hard disk.

This toolkit can also be modified for analyzing internet activities on any system. Furthermore, this can be altered to be used as an Enterprise Software so that it can monitor users' activities on a live system.

This toolkit can be improved by making extensive research on other forensically significant registry keys and then extracting data from them. Likewise more system files, folders and processes can be explored for gathering important evidence out of them.

**ANNEXURE A**  
**SOFTWARE DEVELOPMENT PLAN**

## **Preface**

In this document the comprehensive development and management plan for forensic toolkit “Win4Tech” is described. This plan will be firmly followed for the completion of project. Any amendment applied to this plan will be incorporated in the next versions of this document.

The document includes the details of the software to be delivered; major activities, major milestones and required resources.

## **Introduction**

This section contains the details of the project and the software toolkit to be built. In this section a brief overview of the project is given.

## **Product Functions**

Win4Tech provides a wide range of functionalities, which includes evidence collection from various important places inside a computer system. It is capable to divulge information from system's registry, folders and open processes running on a system.

## **Minimum Requirements**

The minimum requirement for the software to be operational on any system requires Microsoft WINDOWS XP, *Dot Net framework 2.0 and* software publisher **Telerik** of ASP.NET AJAX to be installed on it.

## **Major Milestones**

The key milestones of the project are:

- Completion of Requirement Analysis and Project Specifications phase.
- Completion of Design phase
- Implementation and Integration of components.
- Product testing and delivery

## **Project Deliverables**

This section delineates the major items to be delivered. List of project deliverable is as follows:

<b>Deliverable Name</b>	<b>Due Date</b>
Project Definition and List of team Members	November 15, 2008
Project Defense	February 10, 2009
Requirements Description and Analysis	March 21, 2009
Detail Design Document	14 <sup>th</sup> May, 2009
Fully Functional Product Model	July 22, 2009
Testing Plan	July 29, 2009
Project Report and Review	July 31, 2009

*Table A-1 List of Project Deliverables*

### **Details of the Deliverables**

Following are brief details of the project deliverables:

#### **Project Definition and List of Team Members**

This includes defining the project and list of team members.

#### **Project Defense**

This presentation describes how the team is to be structured and administered and what are the goals and objectives which have to be achieved for the completion of the project.

## **Requirements Description and Analysis**

The requirements of the proposed software system, on which design and coding is based, is outlined in this document.

## **Detail Design Document**

This document builds a high-level design of the project, incorporates the Design Model made in Rational Rose Software and other dataflow diagrams and figures which describe the project's overall design.

## **Fully Functional Product Model**

A fully functional product model will be demonstrated. The full source code will be given on demand.

## **Testing Plan**

A complete testing plan including test cases for every subcomponent of Win4Tech will be carried out.

## **Project Report and Review**

This will include complete project report and documentation explaining achievements of the project. Enhancements in this project will also be suggested in this report.

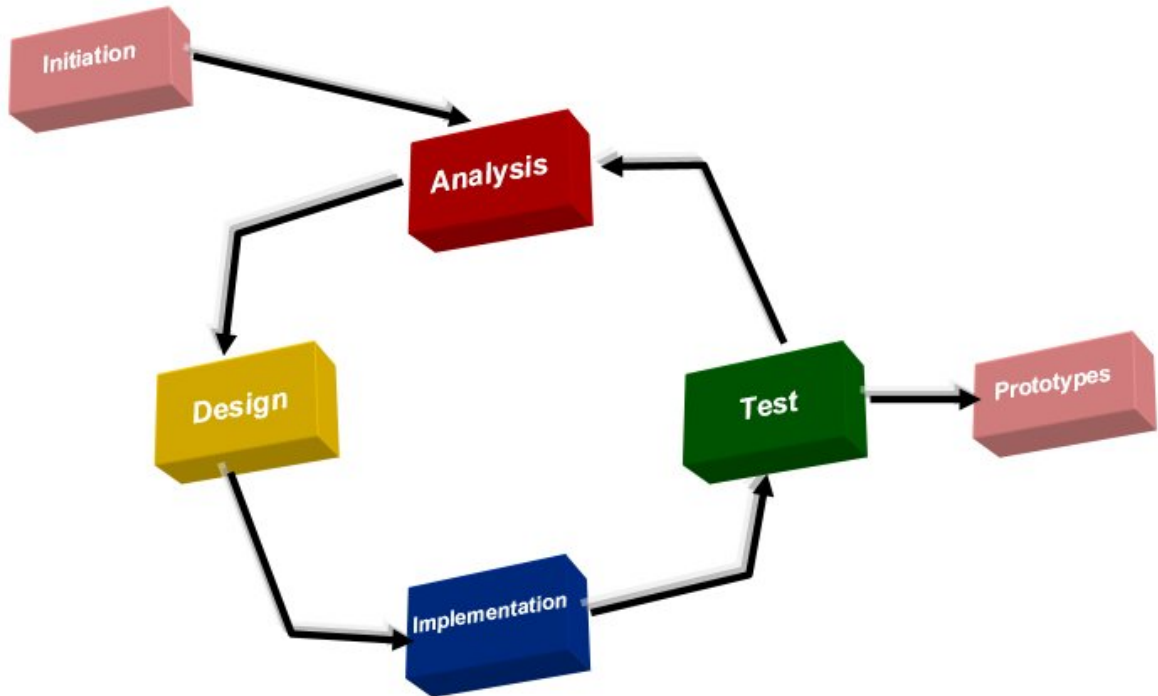
## **Project Organization**

This section describes the development structure of the project which includes the process model, the organizational structure and responsibilities of individuals on the project.



## Software Process Model

The system will be developed using evolutionary software process model. The model is shown in the following figure.



[11] *Figure A-1 Evolutionary Software Process Model*

### Significance

1. Can be implemented with fewer team members.
2. Each iteration delivers a functionally operational product and thus customers can get to see the working version of the product at each stage.

[12]

3. Can be evolved over time to make required changes in a complex system, thus making it much more efficient
4. Evolutionary software process models are well suited for the development of object oriented systems.

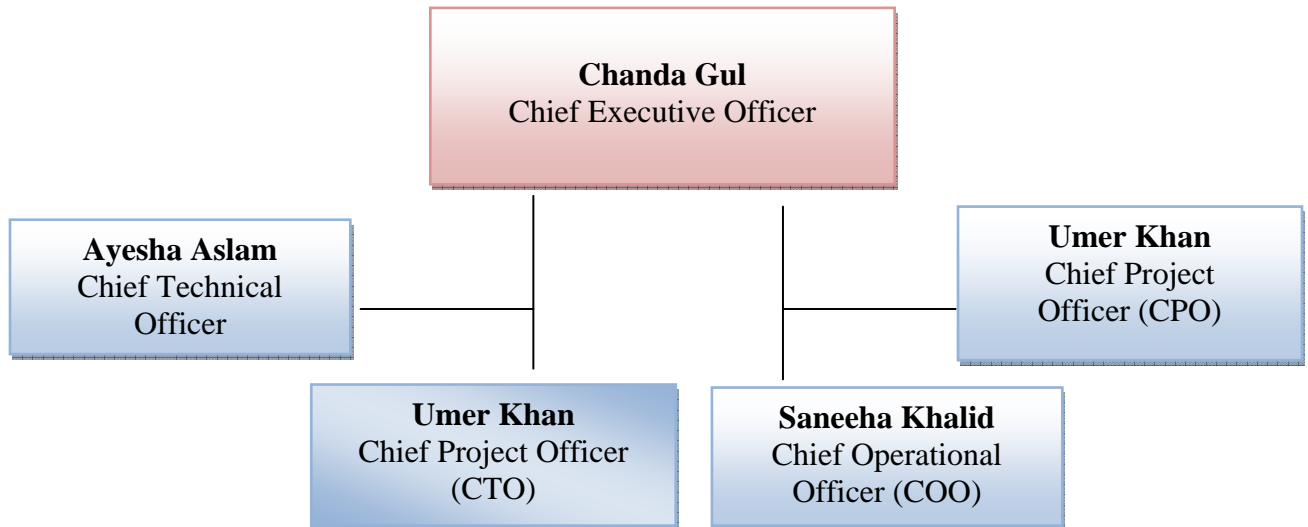
[13]

## Organizational Structure

### Parent Organization

The personnel involved in this project are as follows:

1. Chanda Gul (Chief Executive Officer, Quality Assurance Manager)
2. Ayesha Aslam (Chief Technical Officer, User Interface Prototyper)
3. Saneeha Khalid (Chief Operational Officer, Requirements Engineer)
4. Umer Khan (Chief Project Officer, Designer)



**Figure A-2 Organization Chart of the Executive Staff**

Each member is equipped with significant knowledge of Computer Science, Mathematics and other skills acquired during education tenure to accomplish the project.

All syndicate members are expected to contribute equal amounts of work to the project. This workload needs to be balanced with other courses team members are undertaking. It is expected that a weekly workload for this project should be a minimum of approximately 30 hours/week.

## Organizational Boundaries and Interfaces

1. Regular meetings with client will be held. Approval of the project progress is taken by the client at least once a week.
2. The project documentation will not be revealed to any party other than the client by the team members.
3. Informal Communication will also be used extensively in this project due to its quick and efficient information transferable properties. The team makes use of personal communication and email communication as much as possible. However, given the academic restrictions, email communication is used as the primary form of interacting with other team members.
4. Project Leader will be responsible for communicating with the upper management.
5. There are no subcontracting organization(s) associated with project.

## Project Responsibilities

The following chart illustrates the persons and their respective responsibilities concerning this project. Any alterations in the given structure will be incorporated in the future versions of the document.

<i>Responsibility</i>	<i>Persons Responsible</i>
Overall Project Manager	Chanda Gul
Quality Assurance Manager	Chanda Gul
End-User Documentation Manager	Ayesha Aslam
Requirements Engineer	Saneeha Khalid
Software Architecture	Umer Khan
Technical Self-Reviews	Ayesha Aslam
Software Design	Umer Khan
Software Testing	Saneeha Khalid

*Table A-2: Project Responsibilities Chart*

## **Managerial Process**

In this section we describe strategy that will be adopted to manage the project. This includes management objectives, priorities, project assumptions, dependencies, constraints, monitoring and controlling mechanisms, and the staffing plan.

## **Management Objectives and Priorities**

The major objective is to ensure that time constraints are strictly followed. The quality of work is prioritized. During the earlier stages of the project main concern will be following schedule.

An efficient and functional running forensic toolkit is at the highest priority. The main goal of the project is to achieve the maximum possible efficient functions of the project in the minimum possible time. The project team will attempt to strike maximum possible balance between time and functional priority.

## **Assumptions, Dependencies and Constraints**

It is assumed that the schedule will not clash with the academic schedule of the team members. The team will be able to setup the project to test the product in real time.

## **Major Risks**

The major risks that impinge on the project are

1. Clash of project schedule with academic responsibilities of team members
2. Shortage of time
3. Lack of software engineering experience and unfamiliarity with tools or programming language.

## Project Charter

**Project Title:** WIN4TECH

**Project Start date:** Aug 01, 2008

**Project End date:** July 30, 2008

**Project Leader:** NC Chanda Gul

**Project Objectives:** The objective of the project is to develop software for gathering evidence from a computer system. If any system would be used for criminal purpose then this tool will help in investigating user's crime and will enable law enforcement agencies to punish for his/her offense.

**Business Objectives:** To develop a cost effective forensic toolkit that will be helpful for Law enforcement agencies.

### Stakeholders

The stakeholders of our project are:

1. Project Supervisors
  - Lt. Col. Naveed Sarfraz Khattak (Head of CS Department, MCS, NUST)
  - Lec. Ahmed Raza Cheema (MCS)
  - Dr. Fauzan Mirza (Asst. Professor NIIT)
2. Project Team
  - NC Chanda Gul (Group Leader)
  - NC Ayesha Aslam (Group Member)
  - NC Saneeha Khalid (Group Member)
  - PC Umer Khan (Group Member)
3. CS Department MCS
4. Computer Forensic Investigator (User)

### Assumptions and Dependencies

The tool will gather evidence from live computer system. It will not work on the digital image file directly.

---

*Figure A-3 Project Charter*

## **Progress Monitoring and Controlling Mechanism**

### **Monitoring and Controlling Mechanisms**

Project schedule, quality, and functionality will be tracked throughout the project. In a weekly meeting the progress will be reviewed and analyzed by Project Team Members. Problems encountered by any team member will be discussed and resolved accordingly.

### **Report Contents**

Separate reports shall be primed for monitoring the functional, technical, quality and cost scrutinize of the project. Progress reports will also be maintained in due course of time. The status reports will contain the following details:

1. Status of the current phase activities
2. Estimated time of completion of the current phase
3. Milestone deliverables at current phase

The status reports will be accompanied by Gantt Charts. Various other figures like Activity diagram and pert diagram will also be presented to support the facts laid down in the status reports.

### **Staffing Plan**

Considering the complexity of project at hand and fixed number of personnel, the minimum skill levels are not defined at this stage. The entire duration of the project will be headed and dealt with the same time, each individual acquire different role during different phases.

There will be no extra personnel acquired during the course of the project besides those that constitute the project syndicate before the initiation of the project.

Type of Personnel's	Number of personnel	Required Skill Level / Qualification(s)
Requirement Engineer (s)	1	Experience in Requirements Engineering
Software Designer (s)	2	Software Engineering
Project Leader	1	Software Engineering
Coder (s)	3	Excellent Coding in Visual C#.
QA Manager	1	Experience as QA Officer
Test Officer (s)	2	Experience as Test Officer

*Table A-3 Staffing Plan Chart*

## Technical Process

This section explains the top-level technical processes used on the project including the technical methods, tools, and techniques; major software documents; and supporting activities.

## Methods, Tools and Techniques

### Operating Environment

The operating environment will be that of Microsoft WINDOWS XP. *Dot Net framework 2.0 and* software publisher **Telerik** of ASP.NET AJAX installed on the system is mandatory.

### Hardware

Stand alone IBM PCs.

## **Software tools**

1. Compiler or IDE : Microsoft Visual Studio 2005
2. Programming language : Visual C#

## **Remarks**

Software will have object-oriented reusable structure. Documentation will be clear and explanatory. Microsoft Project software will be used to aid in management and in fixing strict timelines.

## **Project Support Functions**

The project is supported by following other documents. These documents describe the plans for functions that support the software development effort.

1. Detailed Documentation
2. End - user documentation (USER MANUAL)

The plans for these supporting functions will be developed in due time and should be referred to as the need arises. Any other supporting document that needs to be included on the list of the above documents will be added in the later versions of this document.



**ANNEXURE B**  
**USER MANUAL**

# MANUAL

WIN4TECH

## **Purpose of Win4Tech**

Computer Forensics is the application of methods and techniques for investigating a computer system to reveal criminal activities in a court of law. Criminal activities like hacking, information theft, virus attacks and malicious software productions are growing rapidly. To deal with these threats, it is highly important to secure information and perform collection and analysis of evidence from victimized systems.

## **Overview of Win4Tech**

Major functionality of our product will be inspecting users' activity.

Key features are:

1. Examining files which have been viewed previously by any user on the system.
2. Traversing Windows Registry and locating sub keys for every root key and returning the Last Write Time of any key value.
3. Searching the recent activities on a system which includes recent documents, windows searched items, programs that execute on system startup and run commands.
4. Providing record of Created User Accounts on the system.
5. Providing list of currently running processes and associated DLLs (Dynamic Link Library). It will also provide list of files opened by these processes.
6. Recording details of previously and currently connected USBs
7. Providing detailed information about the objects (e.g. text boxes, list boxes, status bars, etc) residing in currently opened windows.

## **Intended Audience**

The *Win4tech manual* is written for law enforcement and corporate security professionals with the following competencies:

1. Basic knowledge of and training in forensic policies and procedures
2. Basic knowledge of and experience with personal computers  
Familiarity with the fundamentals of collecting digital evidence
3. Experience with case studies and reports
4. Familiarity with the Microsoft Windows environment.

## **System Requirements**

The underlying platform for Win4Tech will be *Microsoft Windows XP* with 32 bit architecture. Our product will function on a live system. *Dot Net framework 2.0* will be required to make it operative.

## **Handling Evidence**

Computer forensics involves the acquisition, preservation, analysis, and presentation of computer evidence. This type of evidence is fragile and can easily, even inadvertently, altered, destroyed, or rendered inadmissible as evidence. Computer evidence must be properly obtained, preserved, and analyzed to be accepted as reliable and valid in a court of law. To preserve the integrity of case evidence, forensic investigators do not work on

the original files themselves. Instead, they create an exact replica of the files and work on this image to ensure that the original files remain intact.

## Installation Process

### MAIN INTERFACE OF WIN4TECH

The following is the main graphical user interface of Win4Tech,



### STARTING A CASE

You access the New Case Wizard by selecting **File > New Case**. If this is your first time opening Win4Tech or if you have chosen to always display the FTK Startup screen, select **New Case**



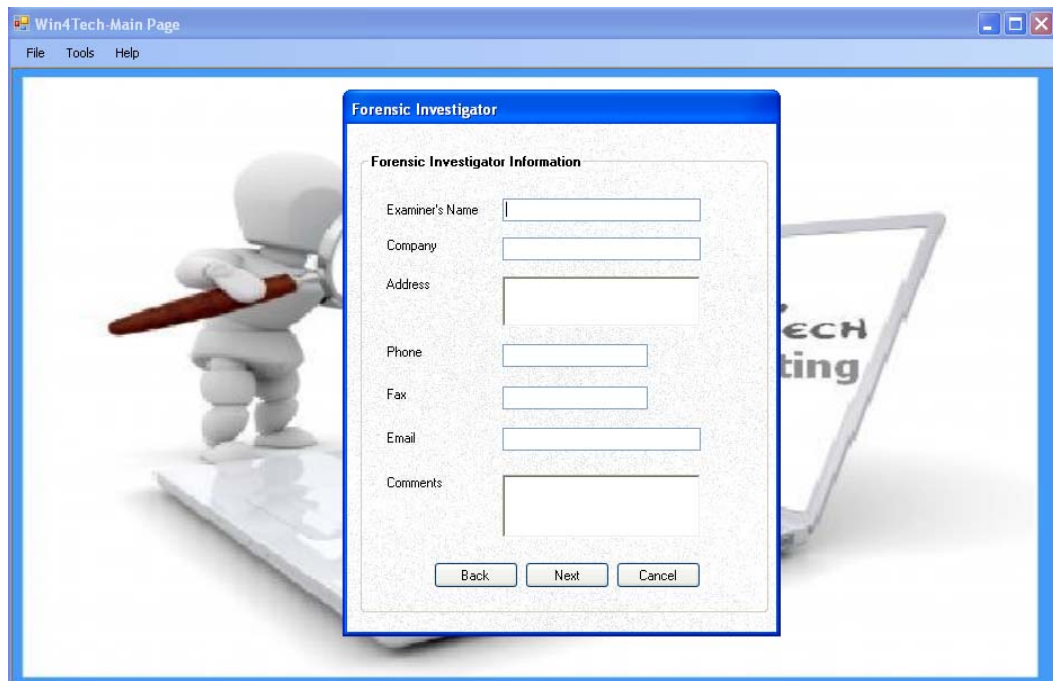
To start a new case, you must complete the following steps.

- 6) Enter the case information.
- 7) Enter the case name.
- 8) Select the Case Path by clicking on the browse button.
- 9) Enter the case description.



10) Click the **Next** button so, that you can go to the next form.

### **FORENSIC INVESTIGATOR'S DETAILS**

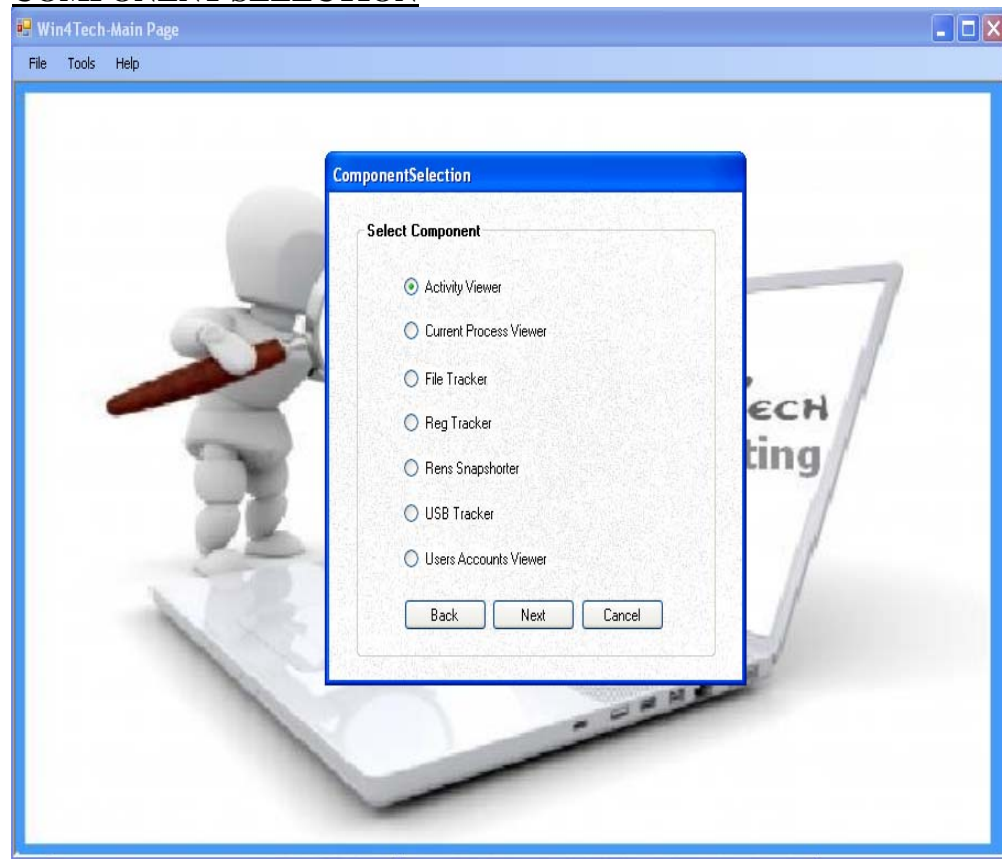


8. Enter your name in the **Examiner's field** as it is mandatory otherwise you would not be able to proceed next.

9. ( **Optional** ) Enter the Company's name that you work for in the **Compnay's field**.
10. ( **Optional** ) Enter the address in the **Address field**.
11. ( **Optional** ) Enter the Phone Number in the **Phone's field**.
12. ( **Optional** ) Enter the fax Number in the **Fax field**.
13. ( **Optional** ) Enter the email address.
14. ( **Optional** ) Write the comments.

Click the NEXT button.

### **COMPONENT SELECTION**

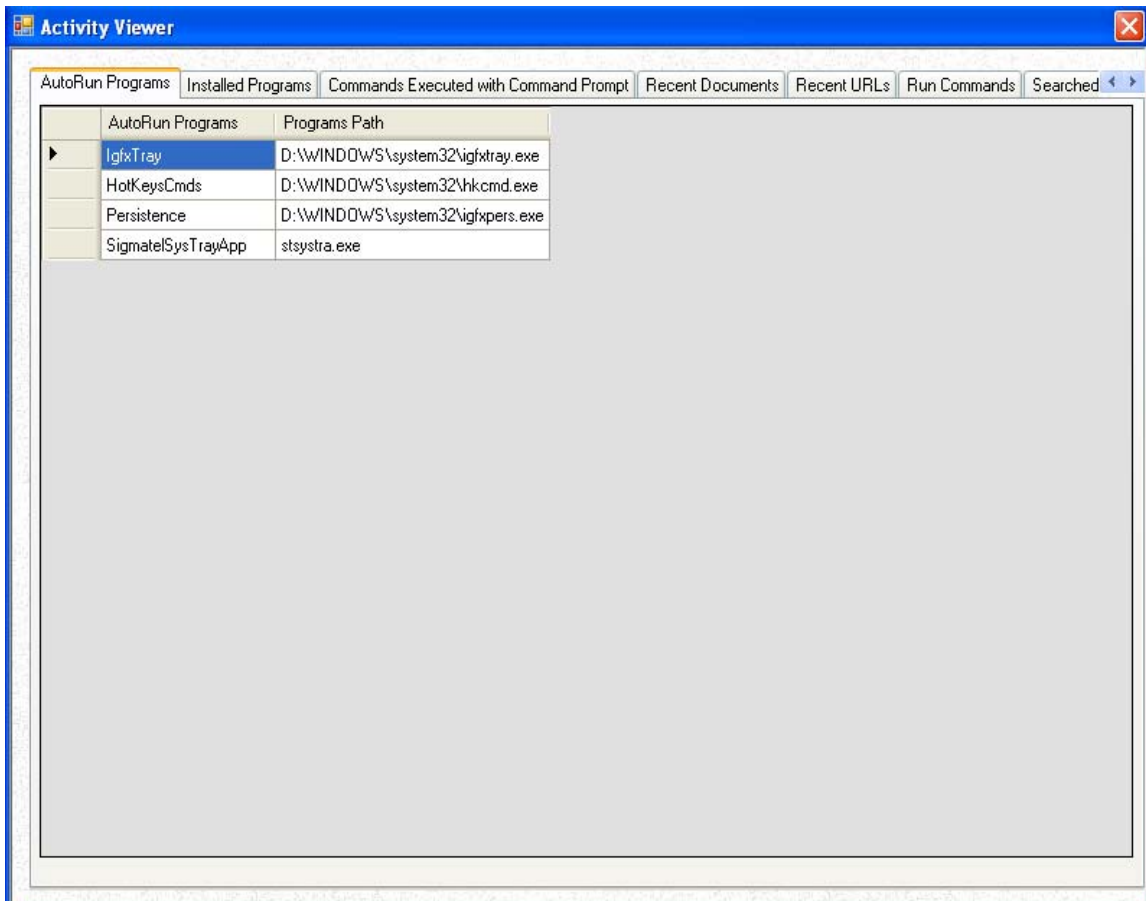


Select the component you want to choose from the mentioned components. Lets suppose you have chosen **Activity Viewer**. Click Next.

## **ACTIVITY VIEWER**

This component will tell you a quite a lot of evidences that are:

10. AutoRun Programs
11. Installed Programs
12. Commands executed with command prompt
13. Recent Documents
14. Recent URLs.
15. Run Commands.
16. Search Commands.
17. Search Words and Phrases.
18. Start-Up Programs.



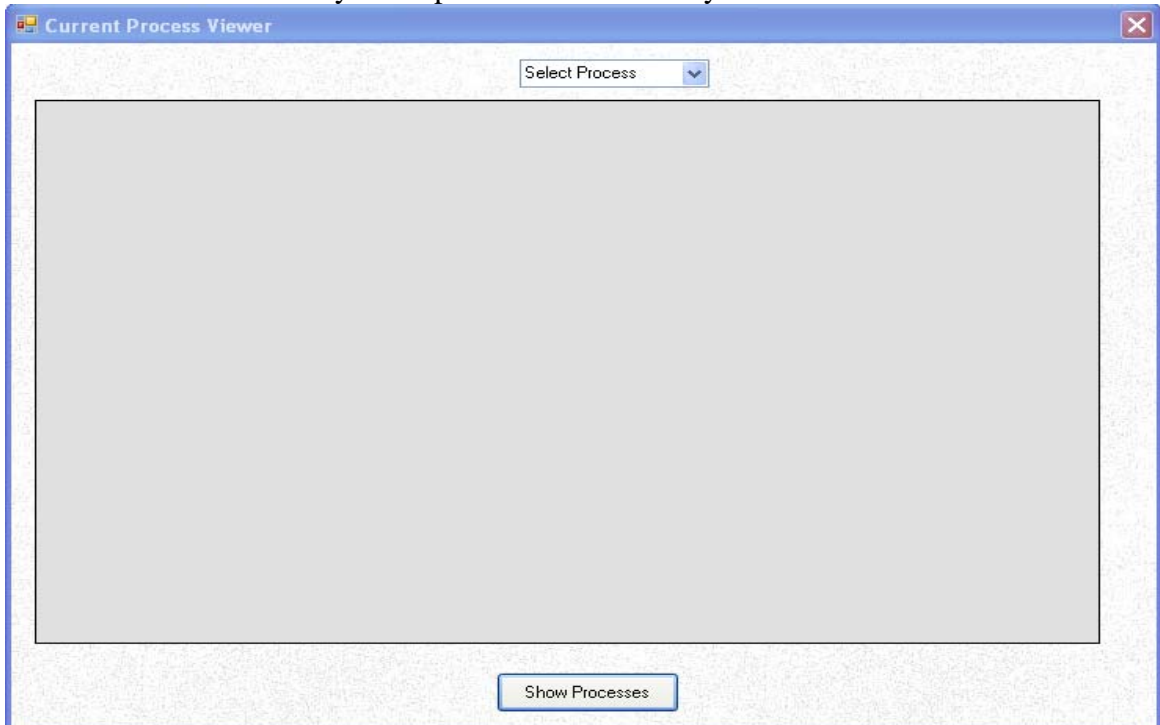


If you click on the tabs, you can see the generated reports that you require. You can close this module and simply select another component .

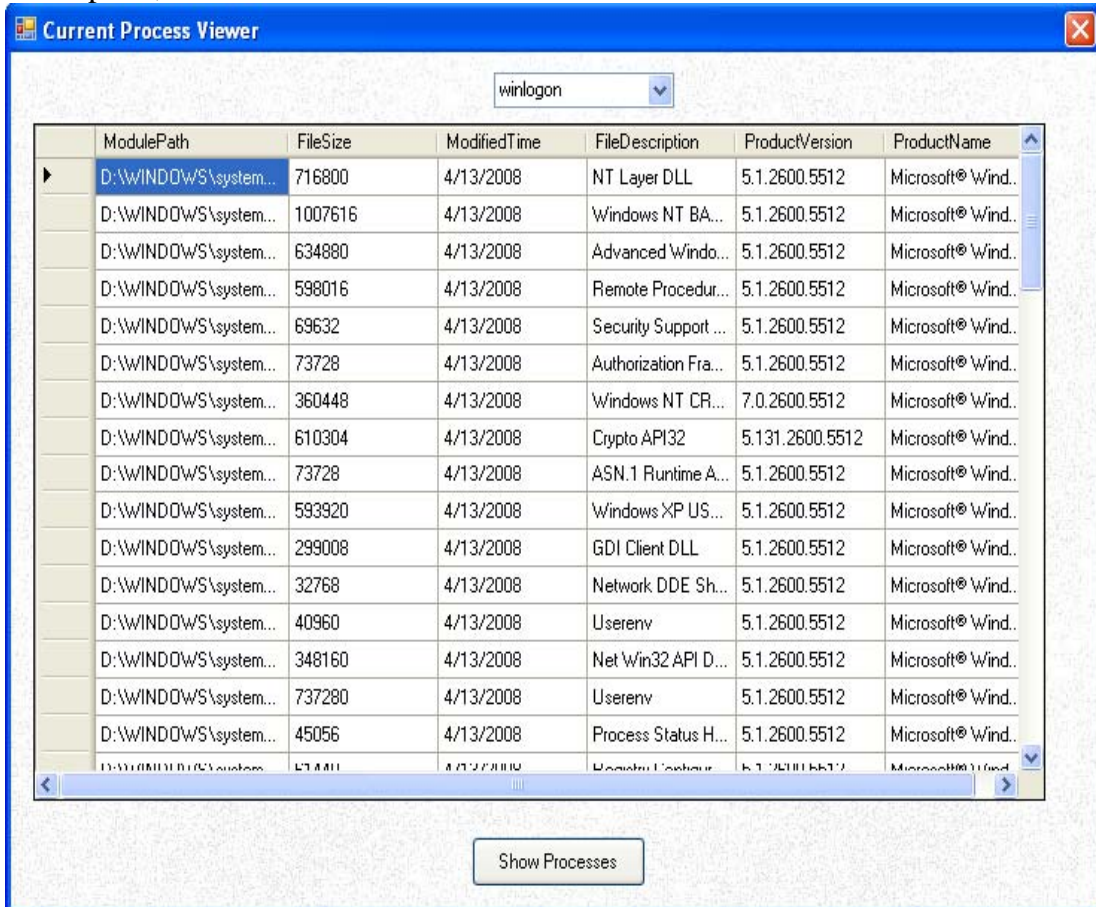


## **CURRENT PROCESS VIEWER**

This component will show you the current processes running on the system and all the DLL files loaded by those processes individually.



Click the **Show Processes** button and it will show you the list of processes in the combo box and the DLL files of each process along with the File Size, Modified Time, File Description, Product Version and Product Name.



Again if you want to choose another component, you can go to the tools tab and select another tool.

## **USB TRACKER**

This component will list the information of all the USBs that were ever plugged into the system after operating system installation. The component will display the name, friendly name, serial number, first and last plug date, vendor id, product id, class and subclasses and names that were given to them.

The screenshot shows the 'USB Tracker' application window. It contains a table with the following columns: Name, Friendly Name, Type, Serial Number, First Plug Date, Last Plug Date, Vendor ID, Product ID, Class, and SubCl. The table lists various USB devices, including Kingston DataTraveler 2.0 drives, a Sony Storage Media device, a USB 2.0 Flash Disk, a Kingston DataTraveler G2, a SanDisk Cruzer Micro, a COBY MP3 Player, and several other Kingston DataTraveler 2.0 drives.

Name	Friendly Name	Type	Serial Number	First Plug Date	Last Plug Date	Vendor ID	Product ID	Class	SubCl
DataTraveler 2.0	Kingston Dat...	USB Mass S...	5B82160006...	5/27/2009 7...	Not Available	13fe	1100	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	5B84110001...	7/20/2009 1...	Not Available	13fe	1100	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	000FEAFAA...	6/1/2009 11...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	5B6A1A835...	6/3/2009 1...	Not Available	13fe	1a00	08	06
Storage Media	Sony Stora...	USB Mass S...	2A08041423...	5/28/2009 1...	Not Available	054c	0243	08	06
Flash Disk	USB 2.0 Flas...	USB Mass S...	5b16fbd239...	7/21/2009 2...	Not Available	1307	0163	08	06
DataTraveler G2	Kingston Dat...	USB Mass S...	0014780F99...	7/14/2009 1...	Not Available	0951	1624	08	06
Cruzer Micro	SanDisk Cru...	USB Mass S...	2004351452...	5/27/2009 7...	Not Available	0781	5151	08	06
COBY MP3 Player	COBY MP3 ...	USB Mass S...	0010100010...	3/24/2009 3...	Not Available	0402	5661	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	0C11516150...	6/3/2009 12...	Not Available	08ec	0016	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	0010000000...	7/14/2009 9...	Not Available	0951	1607	08	06
DataTraveler 2.0...	Kingston Dat...	USB Mass S...	0000000797	6/3/2009 12...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	0014780EC5...	3/13/2009 1...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	8990000000...	6/3/2009 12...	Not Available	0951	1603	08	06
DataTraveler 2.0	Kingston Dat...	USB Mass S...	8990000000...	3/24/2009 1...	Not Available	0951	1603	08	06

## USER'S ACCOUNT HISTORY VIEWER

This component provides detailed information about all the user accounts present on the system. The interface of this module is shown below:

The screenshot shows the 'User Accounts' application window. It contains a table with the following columns: Account Name, SID, Account Creation Time, Last Login Time, Registry Last Modification Time, Profile Folder Last Modification Time, Profile Path, and Size Of Registry. The table lists three accounts: LocalService, NetworkService, and umer.

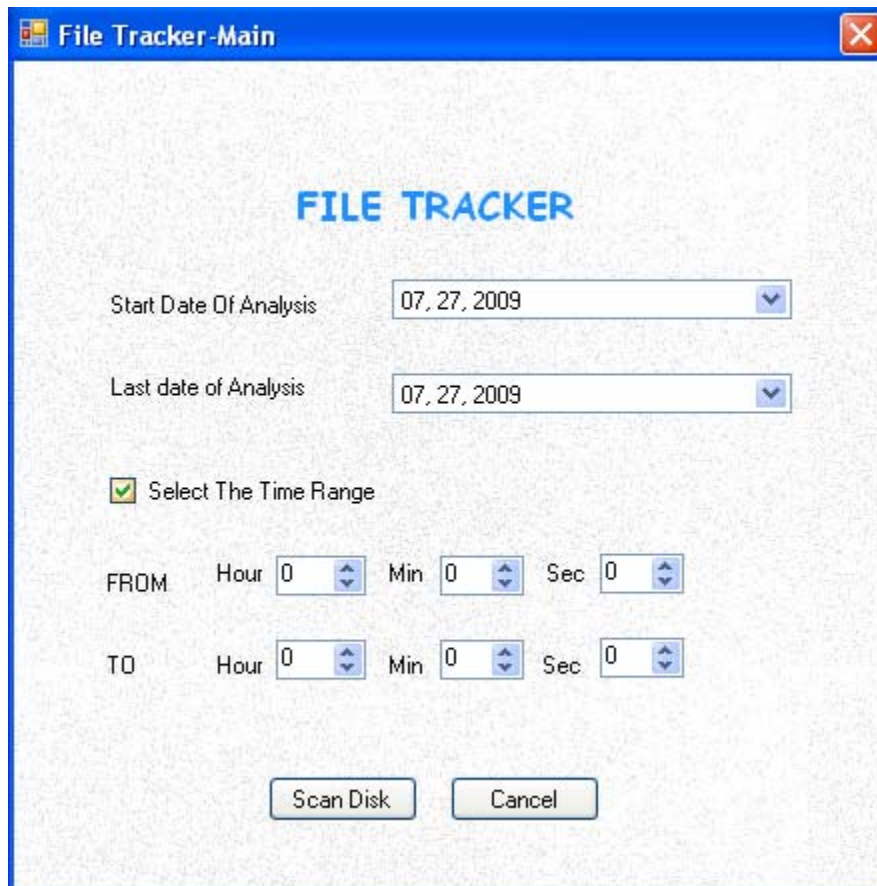
Account Name	SID	Account Creation Time	Last Login Time	Registry Last Modification Time	Profile Folder Last Modification Time	Profile Path	Size Of Registry
LocalService	S-1-5-19	3/13/2009 11:49...	7/27/2009 8:27:...	7/27/2009 8:04:...	3/13/2009 11:49...	D:\Documents a...	262144
NetworkService	S-1-5-20	3/13/2009 11:48...	7/27/2009 8:27:...	7/27/2009 8:04:...	3/13/2009 11:48...	D:\Documents a...	262144
umer	S-1-5-21-746137...	3/13/2009 11:50...	7/27/2009 8:25:...	7/27/2009 8:03:...	7/14/2009 10:49...	D:\Documents a...	2883584

This component will also give us the following details:

4. The creation time of each user account will be displayed
5. Last date of the changes made to the registry and the profile folder will be displayed
6. It gives information about the Security-ID of an account which is used widely for referring to a particular account in the windows registry.

## **FILE TRACKER**

This component will generate a report of the files in a particular drive. The component will provide information about all the files (also the hidden ones) in the drive satisfying the search criteria. The component will be initiated by selecting options of date and drive name. As a result the system will generate a report of files satisfying the search criteria. We can also select the time which will help us to see the desired results.



File Tracker-Main

**FILE TRACKER**

Start Date Of Analysis: 07, 27, 2009

Last date of Analysis: 07, 27, 2009

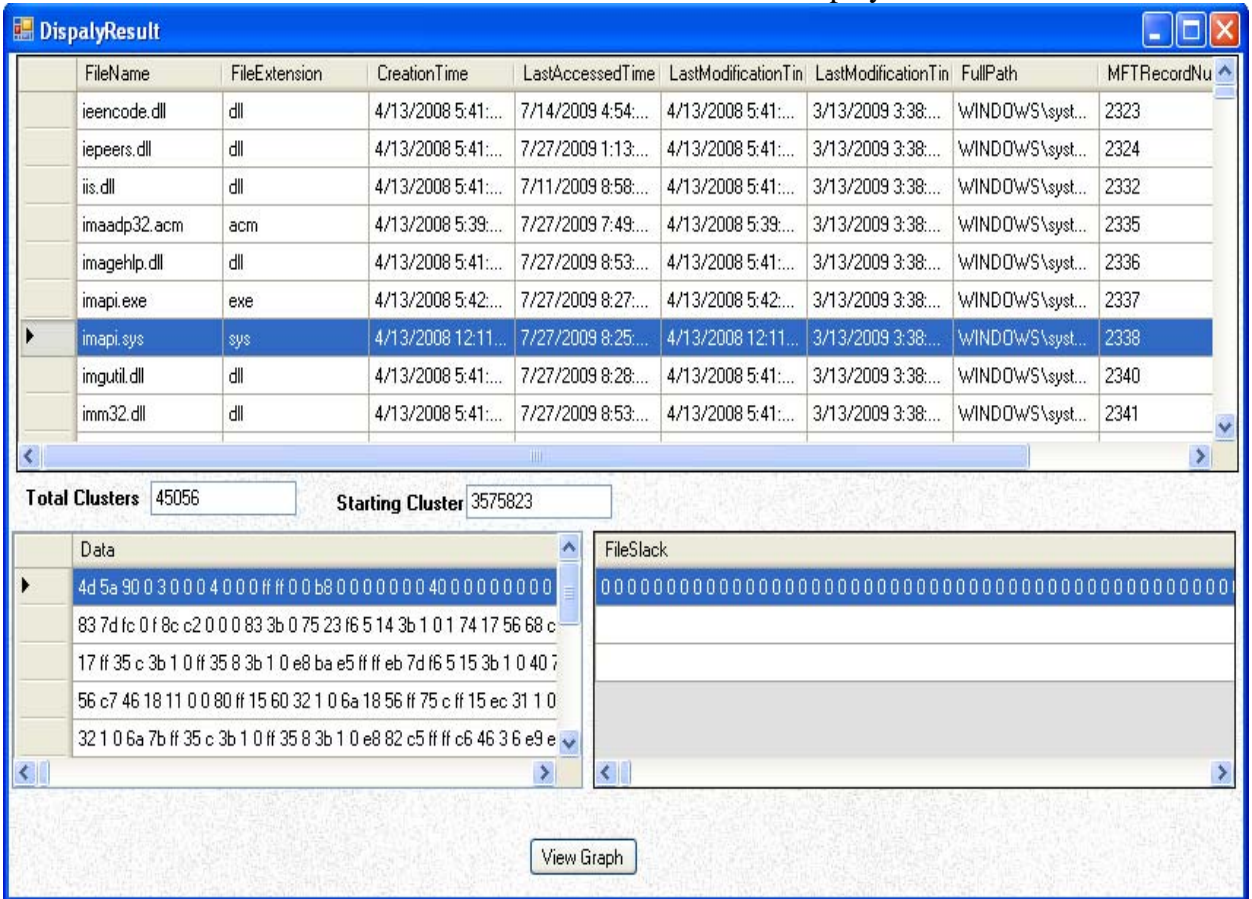
Select The Time Range

FROM Hour: 0 Min: 0 Sec: 0

TO Hour: 0 Min: 0 Sec: 0

Scan Disk Cancel

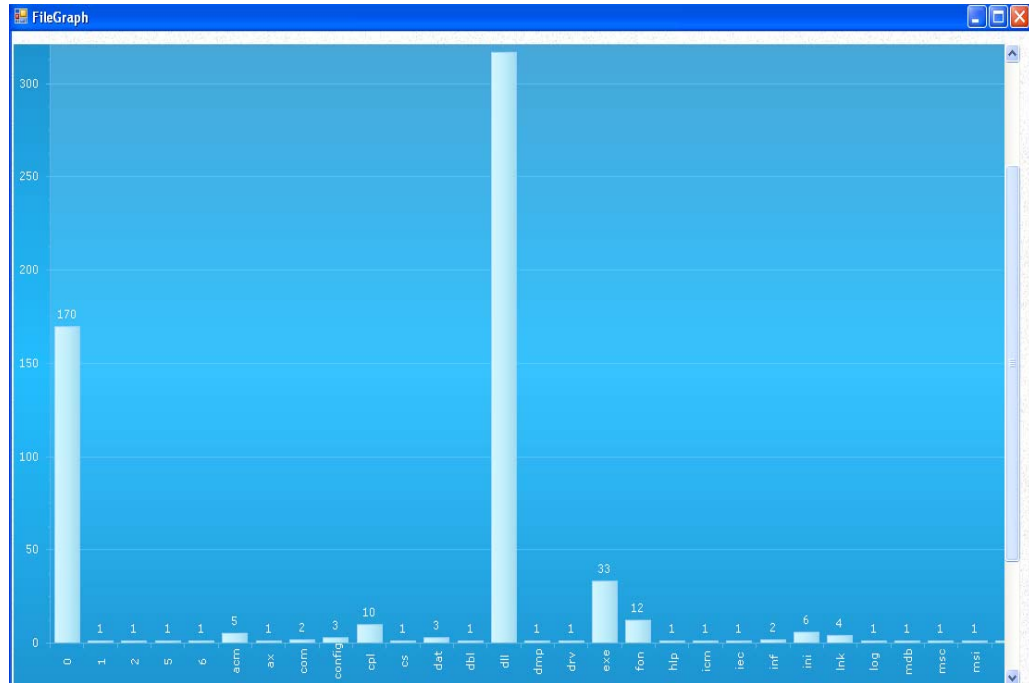
Select the date and time and then click **Scan Disk** which will display the result.



This component will give us the following information:

4. Modified, Accessed, Creation date and time and path for all the files satisfying the search criterion will be displayed.
5. The starting cluster number and total number of clusters used, for all the files found.
6. The data storage space that exists from the end of the file to the end of the last cluster assigned to the file is called "file slack". File slack for all the files found according to the search criteria should be displayed. The slack data will be displayed in the format as it is stored on disk.

We can also view the graph of how much a particular type of file has been accessed. The horizontal axis will show us the file extension and the vertical axis will show us how many times it has been accessed.

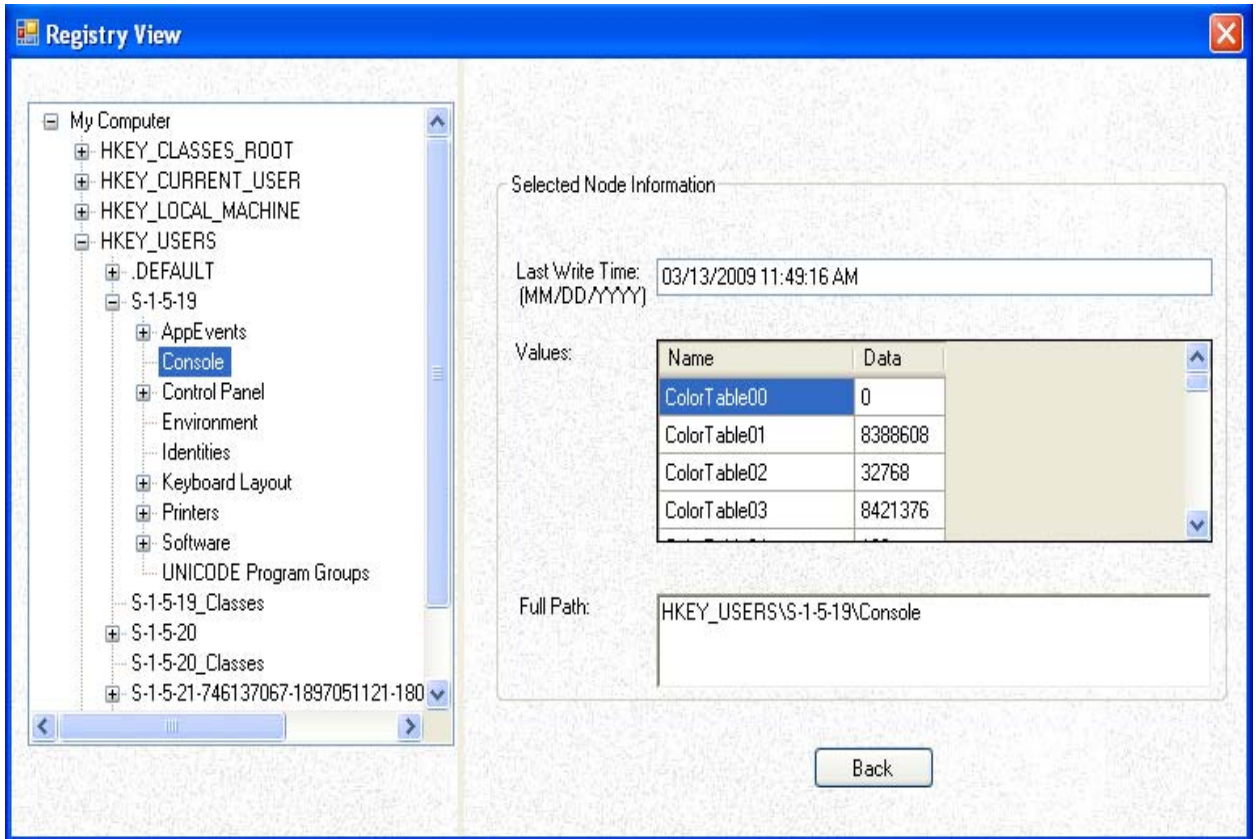


## **REG TRACKER**

This component will generate a report of all the registry keys satisfying the search criteria given by the user. It will also act as a registry viewer showing last write time as additional information with each registry key. It will provide two options, first to view the registry in tree form and second to specify the search criteria in order to view certain keys. After selecting first option a registry view in tree form will be displayed. But if the second option is selected, then the user needs to select different options of dates and root key to make the search specific. As a result the component will show information of all the registry keys satisfying the search criteria.



By choosing the option of loading the entire Registry , you will get an interface which shows all the registry keys along with their last write time .



## **RENS SNAPSHOTER**

This component will grab the data stored in standard list views, tree views, list boxes, combo boxes, text boxes and edit boxes of the opened windows. This requirement is very important because it gives detailed information about all the opened windows in a system. The provided information can be sorted in a much convenient way as compared to capturing screenshots of opened windows and then analyzing them from investigation point of view.

The component will display the title of the opened window and the type, handle and window class of each control of an opened window.

Title	Type	Handle	Items	Visible	Window Class	ProcessID	Process Name
Start Menu	SysListView32	327746	7	False	SysListView32	304	explorer
Start Menu	Button	327736	0	False	Button	304	explorer
Start Menu	SysListView32	262290	13	False	SysListView32	304	explorer
ComponentSelec...	WindowsForms1...	198424	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	198428	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	198412	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	198408	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	198440	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	198434	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	461064	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	461032	0	False	WindowsForms1...	1508	Win4Tech.vshos
ComponentSelec...	WindowsForms1...	657616	0	False	WindowsForms1...	1508	Win4Tech.vshos

ListViewItem: {E-mail}    ListViewSubItem: {E-mail}    ListViewSubItem: {Outlook Express}
ListViewItem: {Internet Explorer}    ListViewSubItem: {Internet Explorer}    ListViewSubItem: {}
ListViewItem: {Windows Messenger}    ListViewSubItem: {Windows Messenger}    ListViewSubItem: {}
ListViewItem: {Media Player Classic}    ListViewSubItem: {Media Player Classic}    ListViewSubItem: {}
ListViewItem: {Adobe Reader 8}    ListViewSubItem: {Adobe Reader 8}    ListViewSubItem: {}
ListViewItem: {VLC media player}    ListViewSubItem: {VLC media player}    ListViewSubItem: {}
ListViewItem: {Microsoft Office Word 2007}    ListViewSubItem: {Microsoft Office Word 2007}    ListViewSubItem: {}



**ANNEXURE C**  
**BIBLIOGRAPHY**

- [1] <http://ezinearticles.com/?Importance-of-Computer-Forensics&id=1116636>
- [2] [http://www.osc.edu/education/si/projects/forensics/cyber\\_crimes.jpg](http://www.osc.edu/education/si/projects/forensics/cyber_crimes.jpg)
- [3] [http://www.us-cert.gov/reading\\_room/forensics.pdf](http://www.us-cert.gov/reading_room/forensics.pdf)
- [4] “How NTFS Works”,  
Available:<http://technet.microsoft.com/enus/library/cc781134.aspx>
- [5] “*NTFS Documentation*”, By Richard Russon, Yuval Fiedel,  
Available:<http://data.linux-ntfs.org/ntfsdoc.pdf.aspx>
- [6] “NTFS Files and Data Storage”  
Available:<http://www.pcguide.com/ref/hdd/file/ntfs/filesFiles-c.html>
- [7] “*Security Risk Assessment, Classified Data Identification & Data Elimination Guidelines*”, Available: <http://www.forensicsintl.com/riskscan.html>
- [8] “*Analysis of hidden data in NTFS files system*”, Cheong Kai Wee, Edith Cowan University
- [9] “*The Windows System Registry*”,  
Available:[http://www.webopedia.com/DidYouKnow/Hardware\\_Software/2005/windows\\_system\\_registry.asp](http://www.webopedia.com/DidYouKnow/Hardware_Software/2005/windows_system_registry.asp)
- [10] Tracking USB storage: Analysis of windows artifacts generated by USB storage devices Harlan Carvey \*, Cory Altheide
- [11] [http://www.wittmannclan.de/ptr/cs/evolutionary\\_model.jpg](http://www.wittmannclan.de/ptr/cs/evolutionary_model.jpg)
- [12] [http://it.toolbox.com/wiki/index.php/Evolutionary\\_Software\\_Process\\_Model](http://it.toolbox.com/wiki/index.php/Evolutionary_Software_Process_Model)
- [13] <http://www.cyberarmy.net/library/article/52>

