Project Report

on

**FOLDER LOCK**

Submitted in the partial fulfilment of requirement

for the award of degree

Bachelor of Technology

in

Information Technology

Guides: Submitted by:

Mrs. Priyanka Vashisht Vinay Sehrawat [06210403112]

Mrs.Leena Singh Nikhil Dahiya [09010403112]

Mr. Ashok Yadav/Abhijeet Kumar Hitesh Ahuja [09910403112]

Vikas [10010403112]



Amity School of Engineering and Technology

[Affiliated to Guru Gobind Singh Indraprastha University, Delhi]

(June 2014- July 2014)

**TABLE OF CONTENTS**

* Certificate i
* Acknowledgement ii
* Abstract iii

Chapter 1 Introduction iv

Chapter 2.System Requirements vi

2.1. Software and Hardware Requirements

2.2. Software Requirement Specification (SRS) vii

2.2.1. Introduction

2.2.2 The Overall Description

2..2.3. Product Perspective

2.2.4 Specific Requirements

2.3. DFD ix

2.4 Flow Diagram x

Chapter 3 Environment Setup xi

3.1. About JAVA

3.2. Java Swing

Chapter 4 Implementation xvii

Chapter 5 Conclusion and Future Scope xix

* References xxi
* Appendix xxii
  + Appendix A – Gantt Chart
  + Appendix B – Source Code
  + Appendix C- Screen Shots

**CERTIFICATE**

This is to certify that this project report entitled **“*Folder Lock”*** comprehends the authentic work of in-house training accomplished by the team of listed students.

|  |  |  |
| --- | --- | --- |
| Name | | Roll No |
| 1. | Vinay Sehrawat | 06210403112 |
| 2. | Nikhil Dahiya | 09010403112 |
| 3. | Hitesh Ahuja | 09910403112 |

4. Vikas 10010403112

This training partially fulfils their Bachelor of Technology course and requirement after second year at Amity School of Engineering and Technology, an institution affiliated to Guru Gobind Singh Indraprastha University, Delhi.

................................... ........................... ......................................................

Mrs. Priyanka Vashisht Mrs. Leena Singh Mr. Ashok Yadav/Abhijeet kumar

i

**ACKNOWLEDGEMENT**

In the course development of this project many people have helped us on various levels. First of all we would like to thank our Director **Mrs. Rekha Aggarwal** and **Professor M.N Gupta**, Head, Department of IT and CSE, for his constant encouragement and guidance throughout the project, thus enabling us to perform our best .

We would also like to thank Mrs. Priyanka Vashisht, Mrs.Leena Singh and Mr. Ashok Yadav/Abhijeet Kumar for the unbound technical guidance and ideas that have helped us enrich and make this project better at each level. We would also like to thank our lab assistant Mr. Amit Kumar for his indefinite support in providing all the softwares which were needed throughout the training.

We would especially like to mention the role these teachers played in widening our horizons by offering advantage of their experience.

……………….. ………………. …………….. ..........................

Vinay Sehrawat Nikhil Dahiya Hitesh Ahuja Vikas

[06210403112] [09010403112] [09910403112] [10010403112]

ii

**ABSTRACT**

This project aims to implement a folder security software. This parses a subset of java language code. It possesses a subset of functionalities provided by Java. The parsing would result in actions on the given input.

Folder Lock is a multi-purpose program that allows different professionals and other persons like casual users and business oriented people to secure their data in the personal computer systems as well as in the portable storage devices.

The data that can be secured by this software can be of any type. For example, .mp3, .jpeg, .txt, .doc, .png, .apk, .docx, .mpeg and many more.

Folder Lock helps to protect data from any unauthorised data leak to the outside world. It keeps the valuable information protected from any misuse and also prevents data theft.

Folder lock also allows the owner of the information to access the data easily, as and when required by the user.

For the ease of the user, this software include all the functions to help the user easily lock and unlock the private and sensitive data as and when required by the user.

These functionalities include locking and unlocking of the data.

***iii***

***INTRODUCTION***

Folder Lock is a full suite solution for all your data security needs. It lets you keep your personal files protected from unauthorized access in both PC and portable drives. Folder Lock comes with a perfect combination of security tools, letting you protect your confidential data that is to be hidden to the outside world. Its purpose has always been to provide a simple and secure way to prevent unauthorized access to sensitive files.

Folder Lock  is an intuitive software solution whose main aim is to assist you in securing the contents of your directories and the directory itself, preventing unauthorized individuals from accessing them without your knowledge or consent.

Folder Lock is the most comprehensive data protection and backup solution in the market today. It provides users with enough flexibility to cater to ever-growing data protection needs in today’s complex environment. Whether it is for personal use or for business/trade purposes, there is a continuously increasing need for securing data. This need, and the unavailability of a comprehensive suite of data security products in the market,makes securing your data even more critical.

Folder Lock offers the following data protection and security features and benefits to cater to each specific user need:

***For casual users:***

For people who consider themselves as casual PC users, and who want to protect their personal and confidential files like their family photos, videos, pictures, financial information, passwords, personal letters, and information that would be inappropriate for your kids, friends or co-workers.

***For a business:***

For business people or IT professionals who need to protect their copyrighted materials, application development charters and workflows, project related documents, their source codes, company documents, trade secrets, and other sensitive intellectual property from accidental deletion and theft.

***For network protection:***

If you want to prevent Viruses, hacker tools, Trojans, Malware, Spyware, Malicious programs and harmful scripts accessing your important files.

***iv***

***For privacy:***

For anyone who does not wish to share their files and folders with people connected over a network or cable without their permission.

***For Intellectual Property:***

For people who need to protect and prevent illegal copying or duplication of their copyrighted material and secret stuff on your hard drive, partitions or portable devices.

***To Prevent Data-Theft:***

If you want to secure a notebook computer so that if it gets misplaced, lost or stolen, all your important files and personal folders are inaccessible.

***For Media:***

If you want to protect mp3s, videos, pictures, books or application files downloaded from KaZaa, Limwire, BearShare, Morpheus, iMesh or

other P2P programs.

***For Portable Data-Protection:***

If you want to protect data on a portable device such as a USB Flash Drive, Memory Stick, CD-RW, Zip Drive or Floppy Disk, so that you can transfer it from one computer to another or archive it without any fear of losing it.

The list is virtually endless. But the protection provided by Folder Lock is not!

**v**

**SYSTEM REQUIREMENTS**

**HARDWARE:**

* + **HARDWARE REQUIRED**
    - 1. INTEL Pentium-II or above
      2. 512 MB RAM, 10 GB hard disk

* + - **DEPLOYED**

1.) CORE 2 DUO processor 1.67GHz

2.) 1 GB RAM, 160 GB hard disk

* + **SOFTWARE REQUIRED**

**Environment** JAVA

**OS** Any OS.

**Development Kit** JDK 1.6

**vi**

**SRS For Folder Lock**

**1.Introduction**

Purpose:-

The purpose of this project is to make a software that will provide security to any folder or file by locking it with the help of different functions.

Scope:-

In today’s situation, data security is a great issue. So there is large scope of this software.

1. Lock:-

We set path of any directory to lock it. It automatically gets locked when lock button is clicked. It involves some functions.

1. Unlock:-

We give the path of any folder to unlock it. It automatically unlocks when unlock button is clicked. It involves some functions.

2) **The overall description:-**

This section describes the general factor that affect the product and its requirement.

2.1) Product perspective:-

2.1.1) *System interface* :- A computer system

2.1.2)*Interfaces* :- An operating system with graphical user interface i.e windows or unix. Here we are using XP/Windows 7.

2.1.3)*Software Interface* :- Software used is Java

Eclipse(Kepler).

a) Name:- Java Eclipe

b)Version:- java2->jdk1.7.

c)Specification:- Java2 from Sun Microsystem.

d) *Operations*:- Coding the Software functionality.

2.2) Product Functions:-

1) Lock

2)Unlock

2.3) Constraints:-

Java must be installed on the customer system as the software product is coded in java language, the software will run only on a system containing java.

2.4) Assumption:-

Java is installed on the computer.

3) **SPECIFIC REQUIREMENTS**:-

3.1) Function:- It performs two functions:-

a) Locking of folder or files.

b) Unlocking of folder or files.

3.2) Performance:-

a) It will be used by one user at a time.

b) It will handle all types of folders and files.

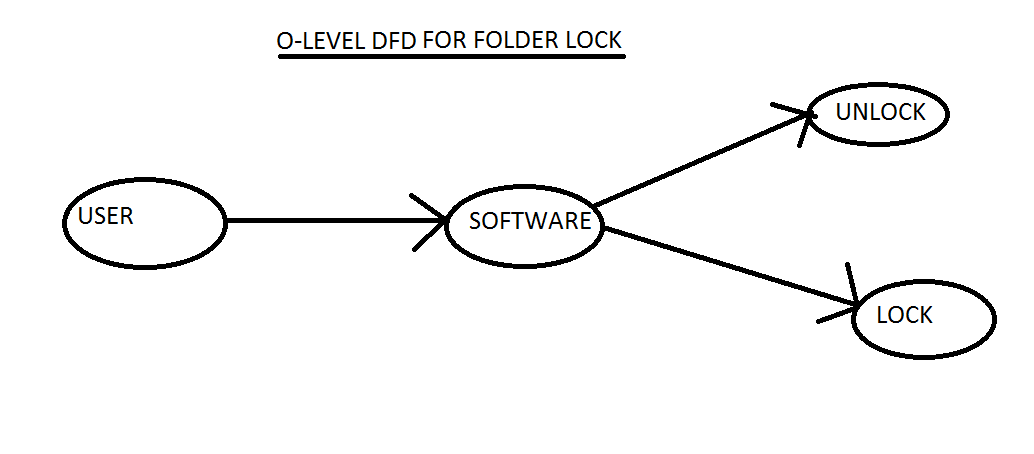
3.3) Software System Attributes:-

Reliability:- Software will be reliable in each condition.

Security:- It provides security for all types of files and folders.

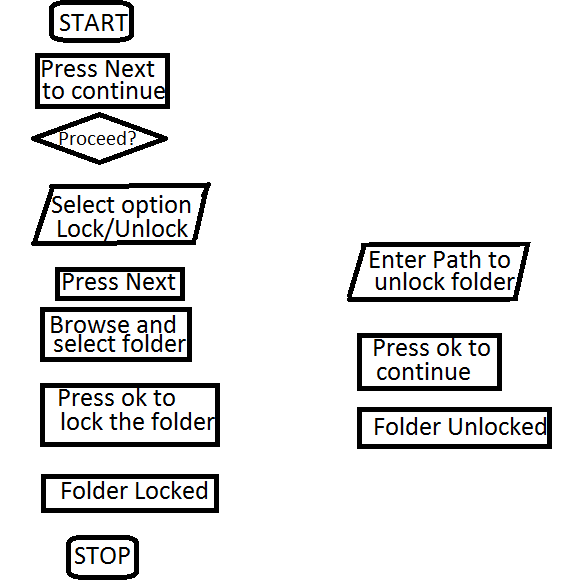
***vii***

***DFD FOR FOLDER LOCK***

******

***ix***

***FLOW DIAGRAM***

******

***x***

***ENVIRONMENT SETUP***

***JAVA***

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets. Because of its rich set of API's, similar to Macintosh and Windows, and its platform independence, Java can also be thought of as a platform in itself. Java also has standard libraries for doing mathematics. Much of the syntax of Java is the same as C and C++. One major difference is that Java does not have pointers. However, the biggest difference is that you must write object oriented code in Java. Procedural pieces of code can only be embedded in objects. In particular, some familiarity with the syntax of C/C++ is useful. In Java we distinguish between applications, which are programs that perform the same functions as those written in other programming languages, and applets, which are programs that can be embedded in a Web page and accessed over the Internet. Our initial focus will be on writing applications. When a program is compiled, a byte code is produced that can be read and executed by any platform that can run Java. What's most special about Java in relation to other programming languages is that it lets you write special programs called *applets* that can be downloaded from the Internet and played safely within a web browser. Traditional computer programs have far too much access to your system to be downloaded and executed willy-nilly. Although you generally trust the maintainers of various ftp archives and bulletin boards to do basic virus checking and not to post destructive software, a lot still slips through the cracks. Even more dangerous software would be promulgated if any web page you visited could run programs on your system. You have no way of checking these programs for bugs or for out-and-out malicious behavior before downloading and running them.

***xi***

***FEATURES OF JAVA***

* Simple
* Object-Oriented
* Robust
* Secure
* Portable
* Architecture-neutral
* High Performance
* Distributed
* Platform Independent
* **Object Oriented:** Object – Oriented Programming is the core of Java. The Java is designed to make clean, usable, pragmatic approach to objects. The object model in java is easy to extend and quite simple.
* **Security:** By using java-compatible Web browser, we can safely download Java applets without fear of viral infection or malicious content.
* **Portable:** We may carry the java byte code to any platform.
* **Robust:** Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.
* **Architecture-neutral:** There are no implementation dependent features e.g. size of primitive types is set.
* **High Performance:** Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)
* **Distributed:** We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet
* **Platform Independent:** Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into byte code. This byte code is a platform independent code because it can be run on multiple platforms i.e. **Write Once and Run Anywhere(WORA).**

**xii**

**ECLIPSE**

In [computer programming](http://en.wikipedia.org/wiki/Computer_programming), **Eclipse** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE). It contains a base [workspace](http://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Written mostly in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), Eclipse can be used to develop applications.

By means of various plug-ins, Eclipse may also be used to develop applications in other [languages](http://en.wikipedia.org/wiki/Programming_language): [Ada](http://en.wikipedia.org/wiki/Ada_(programming_language)), [ABAP](http://en.wikipedia.org/wiki/ABAP),[C](http://en.wikipedia.org/wiki/C_(programming_language)), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [COBOL](http://en.wikipedia.org/wiki/COBOL), [Fortran](http://en.wikipedia.org/wiki/Fortran), [Haskell](http://en.wikipedia.org/wiki/Haskell_(programming_language)), [JavaScript](http://en.wikipedia.org/wiki/JavaScript), [Lasso](http://en.wikipedia.org/wiki/Lasso_(programming_language)), [Natural](http://en.wikipedia.org/wiki/NATURAL), [Perl](http://en.wikipedia.org/wiki/Perl), [PHP](http://en.wikipedia.org/wiki/PHP), [Prolog](http://en.wikipedia.org/wiki/Prolog), [Python](http://en.wikipedia.org/wiki/Python_(programming_language)), [R](http://en.wikipedia.org/wiki/R_(programming_language)), [Ruby](http://en.wikipedia.org/wiki/Ruby_(programming_language)) (including [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails)framework), [Scala](http://en.wikipedia.org/wiki/Scala_(programming_language)" \o "Scala (programming language)), [Clojure](http://en.wikipedia.org/wiki/Clojure" \o "Clojure), [Groovy](http://en.wikipedia.org/wiki/Groovy_(programming_language)), [Scheme](http://en.wikipedia.org/wiki/Scheme_(programming_language)), and [Erlang](http://en.wikipedia.org/wiki/Erlang_(programming_language)" \o "Erlang (programming language)). It can also be used to develop packages for the software[Mathematica](http://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial [codebase](http://en.wikipedia.org/wiki/Codebase) originated from [IBM VisualAge](http://en.wikipedia.org/wiki/IBM_VisualAge).[[2]](http://en.wikipedia.org/wiki/Eclipse_(software)#cite_note-VisualAge-2) The Eclipse [software development kit](http://en.wikipedia.org/wiki/Software_development_kit) (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the [Eclipse Public License](http://en.wikipedia.org/wiki/Eclipse_Public_License), Eclipse [SDK](http://en.wikipedia.org/wiki/Software_development_kit) is [free and open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software).

Eclipse began as an [IBM Canada](http://en.wikipedia.org/wiki/IBM_Canada) project. [Object Technology International](http://en.wikipedia.org/wiki/Object_Technology_International) (OTI), which had previously marketed the [Smalltalk](http://en.wikipedia.org/wiki/Smalltalk)-based [VisualAge](http://en.wikipedia.org/wiki/VisualAge" \o "VisualAge) family of [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) products, developed the new product as a Java-based replacement. In November 2001, a consortium was formed with a board of stewards to further the development of Eclipse as open-source software. It is estimated that IBM had already invested close to $40 million by that time.

The original members were [Borland](http://en.wikipedia.org/wiki/Borland), [IBM](http://en.wikipedia.org/wiki/IBM), [Merant](http://en.wikipedia.org/wiki/Micro_Focus_International" \o "Micro Focus International),[QNX Software Systems](http://en.wikipedia.org/wiki/QNX_Software_Systems), [Rational Software](http://en.wikipedia.org/wiki/Rational_Software), [Red Hat](http://en.wikipedia.org/wiki/Red_Hat), [SuSE](http://en.wikipedia.org/wiki/SuSE" \o "SuSE), [TogetherSoft](http://en.wikipedia.org/wiki/Borland_Together" \o "Borland Together) and [WebGain](http://en.wikipedia.org/wiki/WebGain" \o "WebGain).[[6]](http://en.wikipedia.org/wiki/Eclipse_(software)#cite_note-6) The number of stewards increased to over 80 by the end of 2003. In January 2004, the[Eclipse Foundation](http://en.wikipedia.org/wiki/Eclipse_Foundation) was created.[[7]](http://en.wikipedia.org/wiki/Eclipse_(software)#cite_note-7)

Eclipse 3.0 (released on 21 June 2004) selected the [OSGi](http://en.wikipedia.org/wiki/OSGi" \o "OSGi) Service Platform specifications as the runtime architecture.

RELEASES:

Since 2006, the Foundation has coordinated an annual *Simultaneous Release*. Each release includes the Eclipse Platform as well as a number of other Eclipse projects.

xiii

**JAVA SWING**

**Swing** is the primary [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) [widget toolkit](http://en.wikipedia.org/wiki/Widget_toolkit). It is part of [Oracle](http://en.wikipedia.org/wiki/Oracle_Corporation)'s [Java Foundation Classes](http://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC) — an [API](http://en.wikipedia.org/wiki/Application_programming_interface) for providing a [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI [components](http://en.wikipedia.org/wiki/Software_component) than the earlier [Abstract Window Toolkit (AWT)](http://en.wikipedia.org/wiki/Abstract_Window_Toolkit). Swing provides a native [look and feel](http://en.wikipedia.org/wiki/Look_and_feel) that emulates the look and feel of several platforms, and also supports a [pluggable look and feel](http://en.wikipedia.org/wiki/Pluggable_look_and_feel) that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

**Swing features:**

* **Light Weight** - Swing component are independent of native Operating System's API as Swing API controls are rendered mostly using pure JAVA code instead of underlying operating system calls.
* **Rich controls** - Swing provides a rich set of advanced controls like Tree, Tabbed Pane , slider, color picker, table controls.
* **Highly Customizable** - Swing controls can be customized in very easy way as visual apperance is independent of internal representation.
* **Pluggable look-and-feel**- SWING based GUI Application look and feel can be changed at run time based on available values.

**Components of Swing**

* **JPanel:** JPanel is Swing’s version of the AWT class Panel and uses the same default layout, FlowLayout. JPanel is descended directly from JComponent.
* **JFrame:** JFrame is Swing’s version of Frame and is descended directly from that class. The components added to the frame are referred to as its contents; these are managed by the contentPane. To add a component to a JFrame, we must use its contentPane instead.
* **JDialog:** JDialog is Swing’s version of Dialog and is descended directly from that class. Like Dialog, it uses BorderLayout by default. Like JFrame and JWindow, JDialog contains a rootPane hierarchy including a contentPane, and it allows layered and glass panes. All dialogs are modal, which means the current thread is blocked until user interaction with it has been completed. JDialog class is intended as the basis for creating custom dialogs; however, some of the most common dialogs are provided through static methods in the class JOptionPane.
* [**JLabel**](http://www.javabeginner.com/java-swing/java-jlabel-class-example)**:** JLabel, descended from JComponent, is used to create text labels.
* [**JTextField**](http://www.javabeginner.com/java-swing/java-jtextfield-class-example)**:** JTextField allows editing of a single line of text. New features include the ability to justify the text left, right, or center, and to set the text’s font.
* [**JPasswordField**](http://www.javabeginner.com/java-swing/java-jpasswordfield-class-example)**:** JPasswordField (a direct subclass of JTextField) you can suppress the display of input. Each character entered can be replaced by an echo character.  
  This allows confidential input for passwords, for example. By default, the echo character is the asterisk, \*.
* [**JTextArea**](http://www.javabeginner.com/java-swing/java-jtextarea-class-example)**:** JTextAreaallows editing of multiple lines of text. JTextArea can be used in conjunction with class JScrollPane to achieve scrolling. The underlying JScrollPane can be forced to always or never have either the vertical or horizontal scrollbar;  
  JButton is a component the user clicks to trigger a specific action.
* [**JRadioButton**](http://www.javabeginner.com/java-swing/java-jbutton-class-example)**:** JRadioButton is similar to JCheckbox, except for the default icon for each class. A set of radio buttons can be associated as a group in which only one button at a time can be selected.
* [**JCheckBox**](http://www.javabeginner.com/java-swing/java-jcheckbox-class-example)**:** JCheckBox is not a member of a checkbox group. A checkbox can be selected and deselected, and it also displays its current state.
* [**JComboBox:** JComboBox](http://www.javabeginner.com/java-swing/java-jcombobox-class-example)is like a drop down box. You can click a drop-down arrow and select an option from a list. For example, when the component has focus, pressing a key that corresponds to the first character in some entry’s name selects that entry. A vertical scrollbar is used for longer lists.
* [**JList**](http://www.javabeginner.com/java-swing/java-jlist-class-example)**:** JListprovides a scrollable set of items from which one or more may be selected. JList can be populated from an Array or Vector. JList does notsupport scrolling directly, instead, the list must be associated with a scrollpane. The view port used by the scroll pane can also have a user-defined border. JList actions are handled using ListSelectionListener.
* [**JTabbedPane:** JTabbedpane](http://www.javabeginner.com/java-swing/java-jtabbedpane-class-example)contains a tab that can have a tool tip and a mnemonic, and it can display both text and an image.
* [**JToolbar:** JToolbar](http://www.javabeginner.com/java-swing/java-jtoolbar-class-example)contains a number of components whose type is usually some kind of button which can also include separators to group related components within the toolbar.
* [**FlowLayout**](http://www.javabeginner.com/java-swing/java-flowlayout-class-example)**:** FlowLayout when used arranges swing components from left to right until there’s no more space available. Then it begins a new row below it and moves from left to right again. Each component in a FlowLayout gets as much space as it needs and no more.

***xvi***

**IMPLEMENTATION**

**BASIC PROJECT DETAILS**

This software is built with the intuition of locking and unlocking the files and folders with the aim of securing the user’s sensitive data private. It lets you keep your personal files protected from unauthorized access in both PC and portable drives. Folder Lock comes with a perfect combination of security tools, letting you protect your confidential data that is to be hidden to the outside world. Its purpose has always been to provide a simple and secure way to prevent unauthorized access to sensitive files.

**PROJECT IMPLEMENTATION:-**

FIRST PAGE:-

This is the home page of the software. It displays the main objective of the software.

It also includes a button ‘NEXT’ which when clicked, the software advances to the second page

of the software.

SECOND PAGE: TASK SELECTION :-

This is the second page of the software which allows the user to select the task he wants to perform.

The two tasks included in this page are lock and unlock which are selected by clicking on any one of the radio buttons that are included in the page.

Clicking the next button advances the software to whatever the selection the user has made.

xvii

THIRD PAGE: LOCK:-

This page includes a browse button to select a particular directory which is to be locked.

The path is automatically seen in the text field after the directory has been chosen.

The lock button locks the folder and makes it unaccesable to the outside world.

FOURTH PAGE: UNLOCK:-

This page unlocks the directory.

The path of the folder that was locked has to be provided in order to unlock the folder.

The unlock button makes it available to the user.

ABOUT:

This page tells about the basic details of the software and includes the developer’s names.

xviii

**CONCLUSION**

* The Folder is successfully locked by browsing it in any directory of the computing device or by giving its path which is remembered for unlocking.
* The Folder is Successfully Unlocked by providing the path that is given at the time of locking it.
* This Software can be useful to lock a folder or file against getting accessed by any unauthorized person to prevent sensitive and confidential data access.
* This software can be used to lock a folder in the computing device as well as the portable drives.

Folder Lock is a software that allows different professionals and other persons like casual users and business oriented people to secure their data in the personal computer systems as well as in the portable storage devices.

It is an intuitive software solution whose main aim is to assist you in securing the contents of your directories and the directory itself, preventing unauthorized individuals from accessing them without your knowledge or consent.

This objective is successfully fulfilled by this software by locking and unlocking the directory.

xix

**FUTURE SCOPE**

Folder Lock is the most comprehensive data protection and backup solution in the market today.

It provides users with enough flexibility to cater to ever-growing data protection needs in today’s

complex environment. Whether it is for personal use or for business/trade purposes, there is a

continuously increasing need for securing data. This need, and the unavailability of a

comprehensive suite of data security products in the market,makes securing your data even more

critical.

There is a lot of future scope of this type of software as there is an increasing need to prevent

highly confidential data that is to be made unavailable to the outside world.

Folder Lock offers the following data protection and security features and benefits to cater to

each specific user need:

* Casual Users
* Business Oriented Protection
* Prevention of Data Theft
* Media Protection
* Portable Data Security

xx

**References**

**>** The Complete Reference JAVA2 by Herbert Schildt

> Programming With Java : A Primer by E.Balagurusamy

* + [www.java.net](http://www.java.net)
  + [www.sunmicrosystem.net](http://www.sunmicrosystem.net)
  + [www.sun.com](http://www.sun.com)

xxi

**APPENDIX**

**APPENDIX-A**

**Project Planning – Gantt Chart**

Week-1

Week- 2,3

Week-4

Week-5

Week-6

Project Planning

Requirement Modelling

Throwaway Prototyping

Software Design

Implementing Desing

Adding Functionality

Connecting Design

And functionality

Completion

And delivery

xxii

**APPENDIX-B**

**SOURCE CODE**

HOME PAGE

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.Font;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** javax.swing.Icon;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.Box;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JRadioButton;

**import** javax.swing.JSeparator;

**import** javax.swing.JTextField;

**import** javax.swing.JLabel;

**public** **class** A **extends** JFrame{

**private** JButton item4;

**private** JLabel item5;

**private** JLabel item6;

**private** JLabel lbl;

**private** ImageIcon image;

**private** JLabel item7;

**private** JButton item8;

**private** JSeparator horizontalseparator;

**public** A(){

**super**("FOLDER LOCK");

**this**.getContentPane().setBackground(Color.*YELLOW*);

Container c = getContentPane();

setLayout(**new** FlowLayout());

Icon a = **new** ImageIcon(getClass().getResource("rsz\_2arrow\_right.png"));

Icon b = **new** ImageIcon(getClass().getResource("rsz\_help.png"));

item4= **new** JButton("Next",a);

item5=**new** JLabel("WELCOME TO FOLDER LOCKER!");

item5.setFont(item5.getFont().deriveFont((**float**) 26.0));

item6=**new** JLabel("<html>This wizard will help you restrict access to a folder or a file that contain your sensitive private data.</html>");

item6.setFont(item6.getFont().deriveFont((**float**) 12.0));

item7=**new** JLabel("To Continue Click Next");

image = **new** ImageIcon(getClass().getResource("lock\_yellow.png"));

lbl=**new** JLabel(image);

item8 = **new** JButton("About",b);

horizontalseparator = **new** JSeparator();

Font myfont=**new** Font("Serif", Font.*ITALIC* | Font.*BOLD* |Font.*ROMAN\_BASELINE*,26);

Font newFont1= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*);

Font newFont2 = myfont.deriveFont(Font.*BOLD*,15);

Font newFont3= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*+Font.*BOLD*,14);

c.setLayout(**null**);

item4.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

D dd=**new** D();

dd.setDefaultCloseOperation(JFrame.*HIDE\_ON\_CLOSE*);

dd.setSize(500,400);

dd.setVisible(**true**);

item4.setBackground(Color.*RED*);

**try** {

**for** (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.*getInstalledLookAndFeels*()) {

**if** ("Windows".equals(info.getName())) {

javax.swing.UIManager.*setLookAndFeel*(info.getClassName());

**break**;

}

}

} **catch** (ClassNotFoundException ex) {

java.util.logging.Logger.*getLogger*(C.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (InstantiationException ex) {

java.util.logging.Logger.*getLogger*(C.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (IllegalAccessException ex) {

java.util.logging.Logger.*getLogger*(C.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.*getLogger*(C.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

}

}

});

item4.setBackground(Color.*WHITE*);

c.add(item4);

item4.setBounds(275,300,100,20);

c.add(item5);

item5.setBounds(40,20,500,30);

item5.setFont(newFont1);

c.add(item6);

item6.setBounds(80,40,200,200);

item6.setFont(newFont2);

c.add(lbl);

lbl.setBounds(200, 65, 300, 150);

c.add(item7);

item7.setBounds(155,230,150,30);

item7.setFont(newFont3);

item8.setBackground(Color.*WHITE*);

c.add(item8);

item8.setBounds(120, 300, 100, 20);

item8.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

F ff= **new** F();

ff.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

ff.setSize(500,400);

ff.setVisible(**true**);

item8.setBackground(Color.*RED*);

}

});

c.add(horizontalseparator);

horizontalseparator.setBounds(0,75,500,20);

horizontalseparator.setBackground(Color.*BLACK*);

}

**public** **static** **void** main(String args[]){

A a= **new** A();

a.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

a.setSize(500,400);

a.setVisible(**true**);

}

}

PAGE:2

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.Font;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ItemEvent;

**import** java.awt.event.ItemListener;

**import** javax.swing.ButtonGroup;

**import** javax.swing.Icon;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JRadioButton;

**import** javax.swing.JSeparator;

**import** javax.swing.JTextField;

**import** javax.swing.JLabel;

**import** java.awt.Container;

**import** javax.swing.JSeparator;

**public** **class** D **extends** JFrame {

**private** JLabel item1;

**private** JSeparator item2;

**private** JRadioButton item3;

**private** JLabel item4;

**private** JRadioButton RadioButtonUnlock;

**private** JLabel item6;

**private** JLabel item7;

**private** JButton item8;

**private** ButtonGroup group;

**private** ImageIcon image1;

**private** JLabel lbl1;

**private** ImageIcon image2;

**private** JLabel lbl2;

**public** D(){

**super**("TASK SELECTION");

**this**.getContentPane().setBackground(Color.*YELLOW*);

Container c = getContentPane();

setLayout(**new** FlowLayout());

Icon a = **new** ImageIcon(getClass().getResource("rsz\_2arrow\_right.png"));

item1=**new** JLabel("Select A Task To Perform");

item2 = **new** JSeparator();

item3 = **new** JRadioButton("LOCK",**false**);

item4 = **new** JLabel("Lets you lock the folder or file");

RadioButtonUnlock = **new** JRadioButton("UNLOCK",**false**);

item6 = **new** JLabel("Lets you unlock the folder or file");

item7= **new** JLabel("To Continue Click Next");

item8 = **new** JButton("Next",a);

image1 = **new** ImageIcon(getClass().getResource("rsz\_red\_lock.png"));

lbl1=**new** JLabel(image1);

image2 = **new** ImageIcon(getClass().getResource("rsz\_green\_unlock.png"));

lbl2=**new** JLabel(image2);

Font myfont=**new** Font("Serif", Font.*ITALIC* | Font.*BOLD* |Font.*ROMAN\_BASELINE*,26);

Font newFont1= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*);

Font newFont2 = myfont.deriveFont(Font.*BOLD*,11);

Font newFont3= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*+Font.*BOLD*,14);

c.setLayout(**null**);

group = **new** ButtonGroup();

group.add(item3);

group.add(RadioButtonUnlock);

c.add(item1);

item1.setBounds(100,20,500,30);

item1.setFont(newFont1);

c.add(item2);

item2.setBounds(0,75,500,30);

item2.setBackground(Color.*BLACK*);

c.add(item3);

item3.setBounds(40,125,90,20);

item3.setBackground(Color.*YELLOW*);

item3.addItemListener(**new** ItemListener(){

**public** **void** itemStateChanged(ItemEvent e) {

item8.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e) {

B bb=**new** B();

bb.setDefaultCloseOperation(JFrame.*HIDE\_ON\_CLOSE*);

bb.setSize(500,400);

bb.setVisible(**true**);

}

});

}

});

c.add(item4);

item4.setBounds(50,140,250,30);

item4.setFont(newFont2);

c.add(RadioButtonUnlock);

RadioButtonUnlock.setBounds(40,200,90,20);

RadioButtonUnlock.setBackground(Color.*YELLOW*);

RadioButtonUnlock.addItemListener(**new** ItemListener(){

**public** **void** itemStateChanged(ItemEvent ae) {

item8.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent ae) {

E ee=**new** E();

ee.setDefaultCloseOperation(JFrame.*HIDE\_ON\_CLOSE*);

ee.setSize(500,400);

ee.setVisible(**true**);

item8.setBackground(Color.*RED*);

}

});

}

});

c.add(item6);

item6.setBounds(50,215,250,30);

item6.setFont(newFont2);

c.add(item7);

item7.setBounds(150,285,150,30);

item7.setFont(newFont3);

c.add(item8);

item8.setBounds(175,325,100,20);

c.add(lbl1);

lbl1.setBounds(250,125,90,35);

c.add(lbl2);

lbl2.setBounds(248,200,90,35);

}

}

PAGE 3:

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.Font;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** java.io.File;

**import** javax.swing.Icon;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.Box;

**import** javax.swing.JCheckBox;

**import** javax.swing.JFileChooser;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JRadioButton;

**import** javax.swing.JSeparator;

**import** javax.swing.JTextField;

**import** javax.swing.JLabel;

**public** **class** B **extends** JFrame {

**private** JLabel item1;

**private** JSeparator item2;

**private** JButton item3;

**private** JTextField item4;

**private** JButton item5;

**private** JLabel item6;

**private** JLabel item7;

**private** JLabel item8;

**private** ImageIcon image;

**private** JLabel lbl;

JFileChooser fileChooser=**null**;

File selectedFile=**null**;

**public** B(){

**super**("LOCK");

**this**.getContentPane().setBackground(Color.*YELLOW*);

Container c = getContentPane();

Icon a = **new** ImageIcon(getClass().getResource("rsz\_browse.png"));

Icon b = **new** ImageIcon(getClass().getResource("rsz\_black\_lock.png"));

item1=**new** JLabel("Set Path To Lock");

item2=**new** JSeparator();

item3=**new** JButton("Browse",a);

item4 = **new** JTextField();

item5= **new** JButton("Lock",b);

item6= **new** JLabel("NOTE : Please Remember the Path");

item7= **new** JLabel("Click Browse to Select a Directory");

item8 = **new** JLabel("Path Selected:");

image = **new** ImageIcon(getClass().getResource("rsz\_2lock-2.png"));

lbl=**new** JLabel(image);

Font myfont=**new** Font("Serif", Font.*ITALIC* | Font.*BOLD* |Font.*ROMAN\_BASELINE*,26);

Font newFont1= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*);

Font newFont2 = myfont.deriveFont(Font.*BOLD*,15);

Font newFont3= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*+Font.*BOLD*,14);

Font newFont4 = myfont.deriveFont(Font.*BOLD*,12);

c.setLayout(**null**);

c.add(item1);

item1.setBounds(150,20,500,30);

item1.setFont(newFont1);

c.add(item2);

item2.setBounds(0,75,500,30);

item2.setBackground(Color.*BLACK*);

item3.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent evt) {

item4ActionPerformed(evt);

}

**private** **void** item4ActionPerformed(ActionEvent evt) {//GEN-FIRST:event\_FileSelectButtonActionPerformed

fileChooser=**new** JFileChooser();

fileChooser.setFileSelectionMode(JFileChooser.*FILES\_AND\_DIRECTORIES*);

**int** returnVal = fileChooser.showOpenDialog(**null**);

**if**(returnVal == JFileChooser.*APPROVE\_OPTION*) {

setSelectedFile(fileChooser.getSelectedFile());

setSelectFileText(getSelectedFile().getAbsolutePath());

item3.setBackground(Color.*RED*);

}

}

});

c.add(item3);

item3.setBounds(65, 150, 100, 20);

item3.setBackground(Color.*WHITE*);

c.add(item4);

item4.setBounds(300, 150, 150, 20);

c.add(item5);

item5.setBounds(185, 300, 100, 20);

item5.setBackground(Color.*WHITE*);

item5.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent evt) {

item5.setBackground(Color.*RED*);

item7ActionPerformed(evt);

}

**private** **void** item7ActionPerformed(ActionEvent evt) {

**try** {

String lockString="attrib +H +S "+getSelectedFile().getAbsolutePath();

Process runtimeProcess = Runtime.*getRuntime*().exec(**new** String[] { "cmd.exe", "/c", lockString });

runtimeProcess.waitFor();

**if**(runtimeProcess.exitValue()==0) {

JOptionPane.*showMessageDialog*(**null**, "Folder locked Successfully ");

}

} **catch** (Exception ex) {

JOptionPane.*showMessageDialog*(**null**, "Can not lock the file ");

}

}

});

c.add(item6);

item6.setBounds(150,330,200,20);

c.add(item7);

item7.setBounds(150,110,200,20);

item7.setFont(newFont3);

c.add(item8);

item8.setBounds(215, 150, 150, 20);

item8.setFont(newFont4);

c.add(lbl);

lbl.setBounds(135, 170, 200, 150);

}

**public** String getSelectFileTextUnhide() {

**return** item4.getText();

}

**public** **void** setSelectFileText(String SelectFileText) {

**this**.item4.setText(SelectFileText);

}

**public** File getSelectedFile() {

**return** selectedFile;

}

**public** **void** setSelectedFile(File selectedFile) {

**this**.selectedFile = selectedFile;

}

}

PAGE 4:

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.Font;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** java.io.File;

**import** javax.swing.Icon;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.Box;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JRadioButton;

**import** javax.swing.JSeparator;

**import** javax.swing.JTextField;

**import** javax.swing.JLabel;

**class** E **extends** JFrame{

**private** JLabel item1;

**private** JSeparator item2;

**private** JLabel item3;

**private** JTextField item4;

**private** JButton item5;

**private** ImageIcon image;

**private** JLabel lbl;

**public** E(){

**super**("UNHIDE");

**this**.getContentPane().setBackground(Color.*YELLOW*);

Container c = getContentPane();

setLayout(**new** FlowLayout());

Icon a = **new** ImageIcon(getClass().getResource("rsz\_unlock\_black.png"));

item1=**new** JLabel("Set Path To Unlock");

item2=**new** JSeparator();

item3=**new** JLabel("Enter The Path To Unhide:");

item4= **new** JTextField();

item5 = **new** JButton("Unlock",a);

image = **new** ImageIcon(getClass().getResource("rsz\_unlock.png"));

lbl=**new** JLabel(image);

Font myfont=**new** Font("Serif", Font.*ITALIC* | Font.*BOLD* |Font.*ROMAN\_BASELINE*,26);

Font newFont1= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*);

Font newFont2 = myfont.deriveFont(Font.*BOLD*,15);

Font newFont3= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*+Font.*BOLD*,14);

c.setLayout(**null**);

c.add(item1);

item1.setBounds(150,20,500,30);

item1.setFont(newFont1);

c.add(item2);

item2.setBounds(0,75,500,30);

item2.setBackground(Color.*BLACK*);

c.add(item3);

item3.setBounds(85,100,200,20);

item3.setFont(newFont2);

c.add(item4);

item4.setBounds(300,100,150,20);

item5.setBackground(Color.*WHITE*);

c.add(item5);

item5.setBounds(190,250,100,20);

item5.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent evt) {

item5.setBackground(Color.*RED*);

FolderLockButtonUnhideActionPerformed(evt);

}

**private** **void** FolderLockButtonUnhideActionPerformed(ActionEvent evt) {

**try** {

File fileToUnhide=**new** File(getSelectFileTextUnhide());

**if**(fileToUnhide.isHidden()){

String lockString="attrib -H -S "+fileToUnhide.getAbsolutePath();

Process runtimeProcess = Runtime.*getRuntime*().exec(**new** String[] { "cmd.exe", "/c", lockString });

runtimeProcess.waitFor();

**if**(runtimeProcess.exitValue()==0) {

JOptionPane.*showMessageDialog*(**null**, "File unlocked Successfully ");

}

}

} **catch** (Exception ex) {

JOptionPane.*showMessageDialog*(**null**, "Can not unlock the file");

}

}

});

c.add(lbl);

lbl.setBounds(165,120,150,150);

}

**public** String getSelectFileTextUnhide() {

**return** item4.getText();

}

}

PAGE 5:

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

**import** java.awt.Font;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** java.io.File;

**import** javax.swing.ImageIcon;

**import** javax.swing.JButton;

**import** javax.swing.Box;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JRadioButton;

**import** javax.swing.JScrollBar;

**import** javax.swing.JSeparator;

**import** javax.swing.JTextField;

**import** javax.swing.JLabel;

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**class** F **extends** JFrame{

**private** JLabel item1;

**private** JSeparator item2;

**private** JLabel item3;

**private** JLabel item4;

**private** JScrollPane scrollpane;

**private** ImageIcon image;

**private** JLabel lbl;

**public** F(){

**super**("ABOUT");

**this**.getContentPane().setBackground(Color.*YELLOW*);

Container c = getContentPane();

setLayout(**new** FlowLayout());

item1=**new** JLabel("About The Application");

item2=**new** JSeparator();

item3 = **new** JLabel("<html>Folder Lock is a full suite solution for all your data security needs. It lets you keep your personal files protected from unauthorized access in both PC and portable drives.</html>");

item4 = **new** JLabel("<html>Application Developed By:<br> Vinay Sehrawat<br>Nikhil Dahiya<br>Hitesh Ahuja<br>Vikas</html>");

image = **new** ImageIcon(getClass().getResource("rsz\_red\_lock (2).png"));

lbl=**new** JLabel(image);

Font myfont=**new** Font("Serif", Font.*ITALIC* | Font.*BOLD* |Font.*ROMAN\_BASELINE*,26);

Font newFont1= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*);

Font newFont2 = myfont.deriveFont(Font.*BOLD*,15);

Font newFont3= myfont.deriveFont(Font.*ITALIC*+Font.*ROMAN\_BASELINE*+Font.*BOLD*,15);

c.setLayout(**null**);

c.add(item1);

item1.setBounds(150,20,500,30);

item1.setFont(newFont1);

c.add(item2);

item2.setBounds(0,75,500,30);

item2.setBackground(Color.*BLACK*);

c.add(item3);

item3.setBounds(30,85,450,55);

item3.setFont(newFont3);

c.add(lbl);

lbl.setBounds(145,125,150,150);

c.add(item4);

item4.setBounds(30,235,450,100);

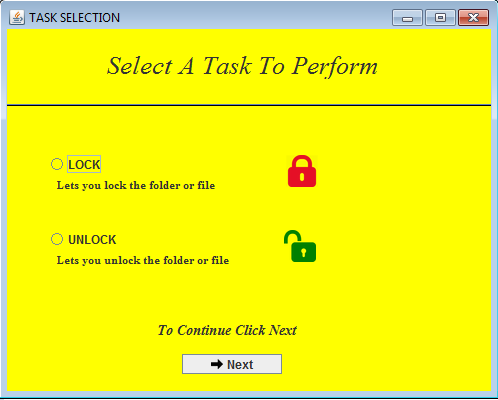
item4.setFont(newFont3);

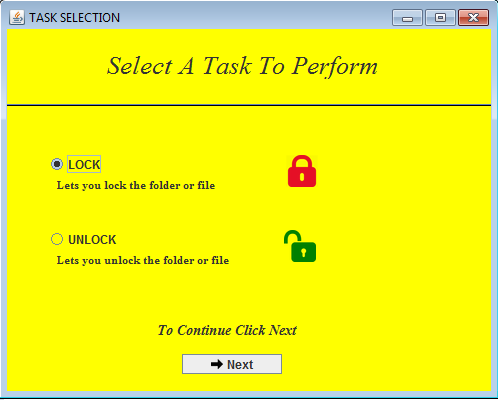
}

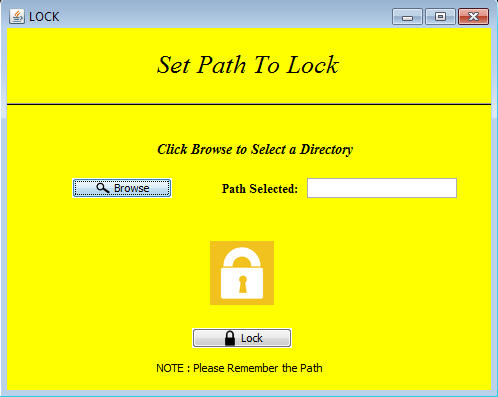
}

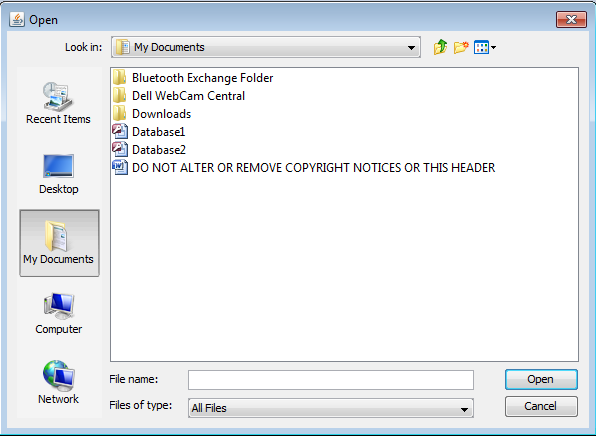
**SCREENSHOTS**

****

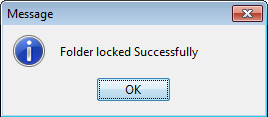
****

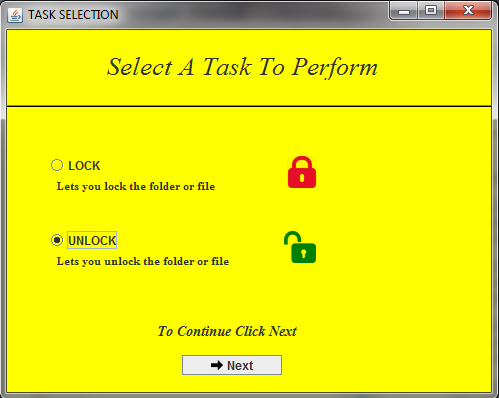
****

****

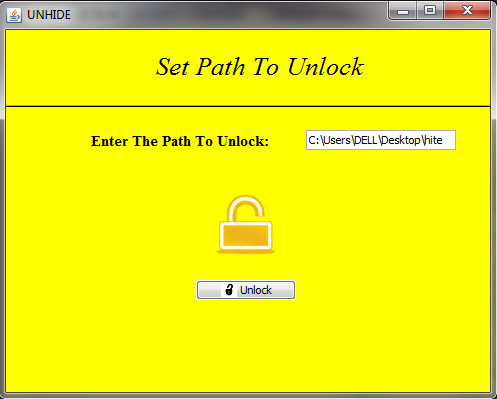
****

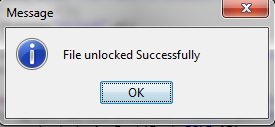
****

****

****

****

****

****