LockedMe.com

prototyped application (User Interaction via command line) documentation

Submitted By: Arsha R Unnithan

arsharunnithan@gmail.com

GitHub repository link:
https://github.com/arsharunnithan/simpliLearnJan2023

1. Project and Developer Details

Company Lockers Pvt. Ltd. aims to digitize their products and chose LockedMe.com as their first project to start with. Developer is asked to develop a prototype of the application. The prototype of the application will be then presented to the relevant stakeholders for the budget approval.

- Specification document - Product's capabilities, appearance, and user interactions
- Number and duration of sprints required
- Setting up Git and GitHub account to store and track your enhancements of the prototype
- Java concepts being used in the project
- Data Structures where sorting and searching techniques are used.
- Generic features and three operations:
    - Retrieving the file names in an ascending order
    - Business-level operations:
        - Option to add a user specified file to the application
        - Option to delete a user specified file from the application
        - Option to search a user specified file from the application
        - Navigation option to close the current execution context and return to the main context
    - Option to close the application

The goal of the company is to deliver a high-end quality product as early as possible.

Project development assigned to: Arsha R Unnithan
Full Stack Developer, Company Lockers Pvt. Ltd.
arsharunnithan@gmail.com

Development cycle: 2 Sprints

Tech stack used:

- Java
- Eclipse
- Git
- GitHub
- Scrum
- Search and Sort techniques: Stream, Set, Files, Collections, Iterator, List
- Core concepts used: I/O Streams, Exception handling, OOPs, Recursion,Sorting
- Microsoft word used for documentation. Document shared in pdf format.

2. Sprints planned and the tasks achieved in them

Agile Project Management

Project Duration	2 Weeks (15 working days)
Number of Sprints	2
Total number of product backlog items	12

2.1 Project planning details

Sprint	Start date	End date	Product Backlog Item ID
Week 1	01/27/2023	02/02/2023	1.1, 1.1.2, 1.1.3, 1.2, 1.3, 1.3.1, 1.3.2
Week 2	02/03/2023	02/09/2023	1.3.2.1, 1.3.2.2, 1.3.2.3, 1.3.2.4, 1.3.3

2.2 Product Backlog/User Stories

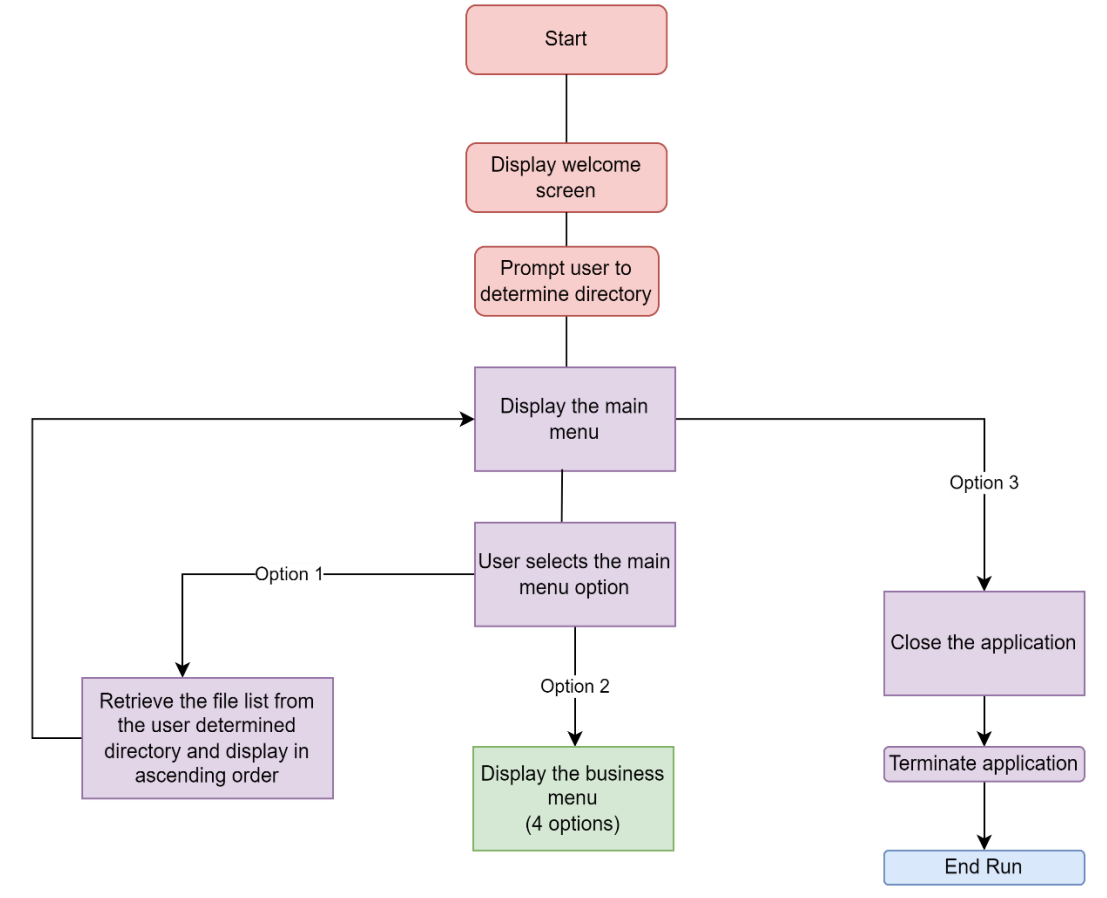
ID	User Story	Story Points	Assigned Sprint
1.1	The LockedMe.com application should display the welcome screen.	1	Sprint 1
1.1.2	Application name and developer details should be displayed.	-	Sprint 1
1.1.3	Features to accept the user input to select one of the options should be listed.	-	Sprint 1

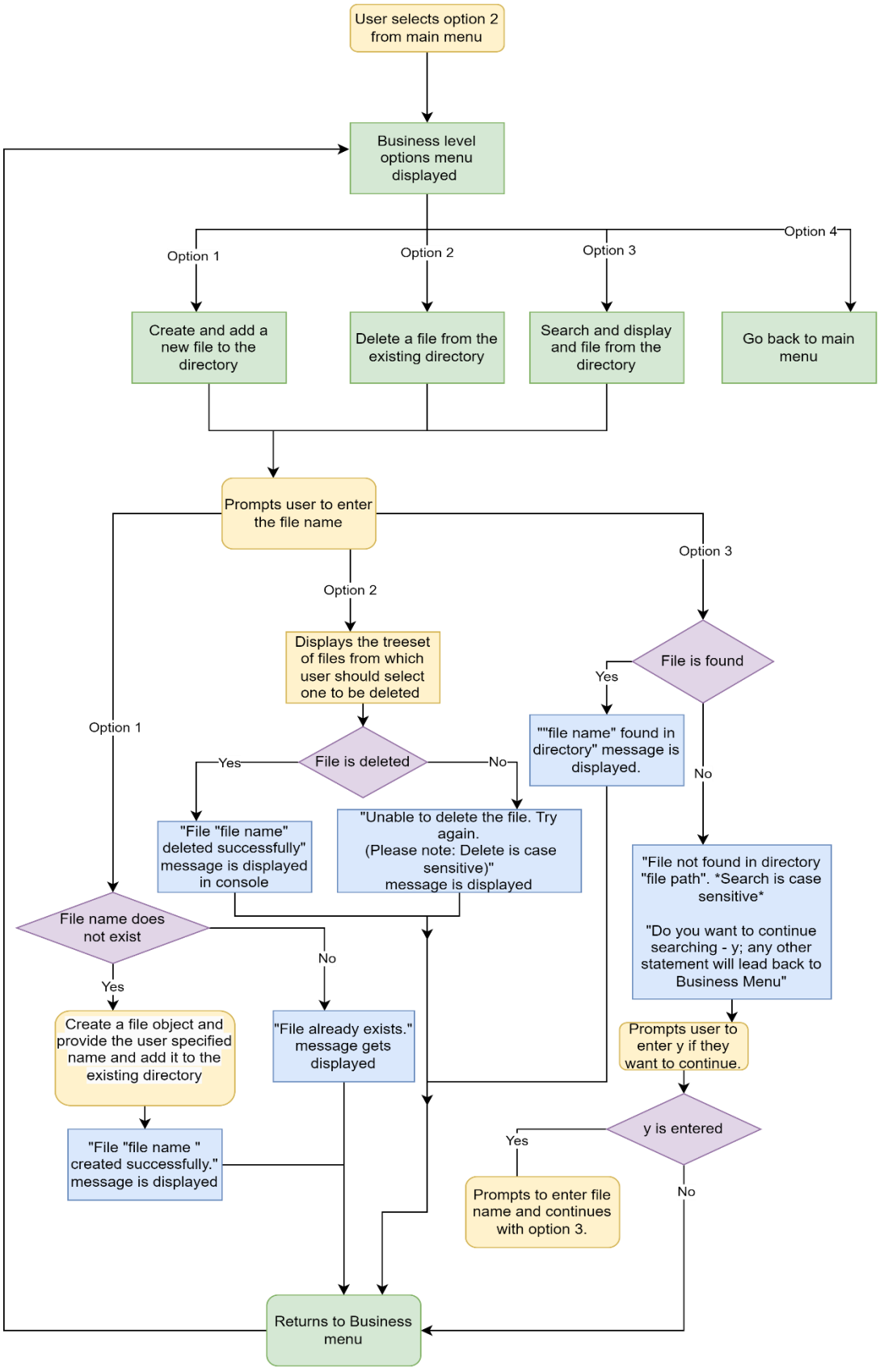
1.2	Ask user to specify the target directory for file operations. It cannot be changed further.	1	Sprint 1
1.3	A numerical menu of three operations should be considered as the main menu.	1	Sprint 1
1.3.1	First menu option should retrieve the file names and sort them in an ascending order.	2	Sprint 1
1.3.2	Second main menu operation is considered Business level operations with four option in it.	2	Sprint 1
1.3.2.1	First option to add a user specified file to the directory(Case sensitivity could be ignored)	2	Sprint 2
	• Ask user a file name and provide it to the new file being added.	-	
	• Return appropriate messages	-	
1.3.2.2	Second option of Business menu to	2	Sprint 2

	delete a user specified file from the existing directory.		
	• User should be displayed with a list of existing files in the directory from which they can select which one to delete.	-	
	• Delete the user specified file and Return appropriate messages. (Delete function is case sensitive)	-	
1.3.2.3	Third option of Business menu is to search and display a user specified file from the existing directory.	2	Sprint 2
	• Search and display the file and return appropriate messages.	-	
	• User should be given a second chance is the	-	

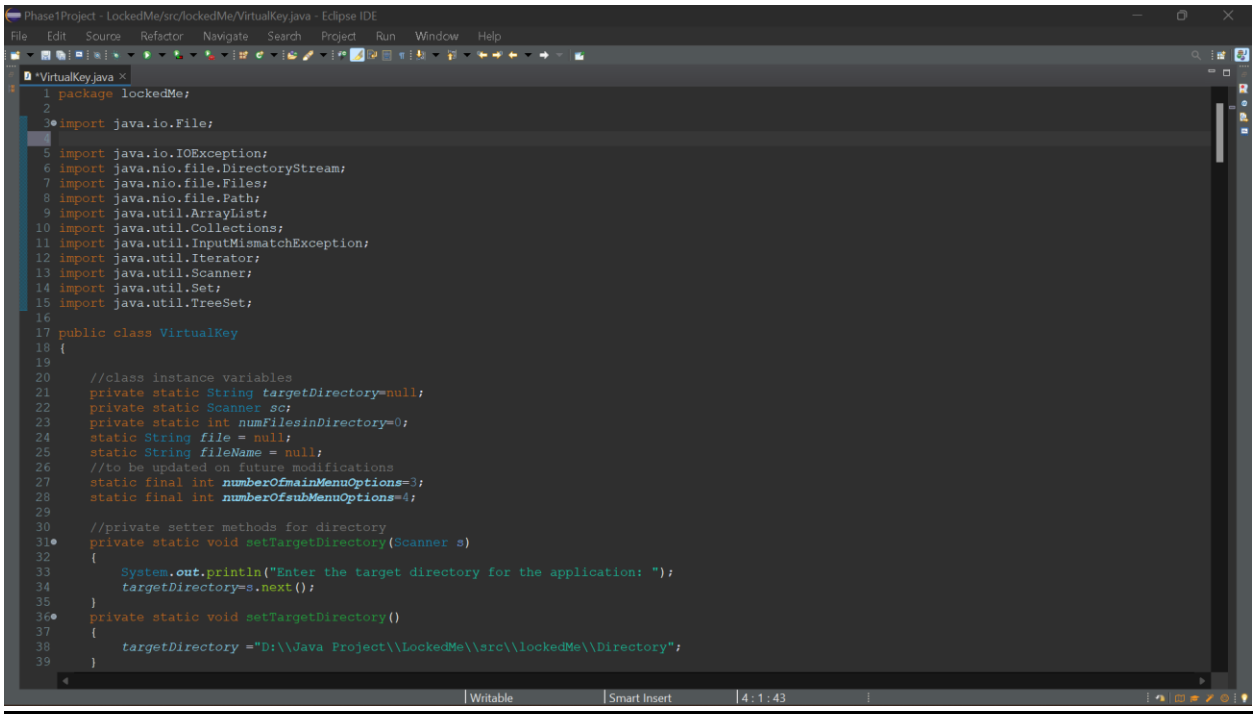
	file is not found.		
1.3.2.4	Fourth business menu option is to go back to the main menu.	1	Sprint 2
1.3.3	Third main menu option should close the application	1	Sprint 2

3. Algorithms/Flowchart of the application

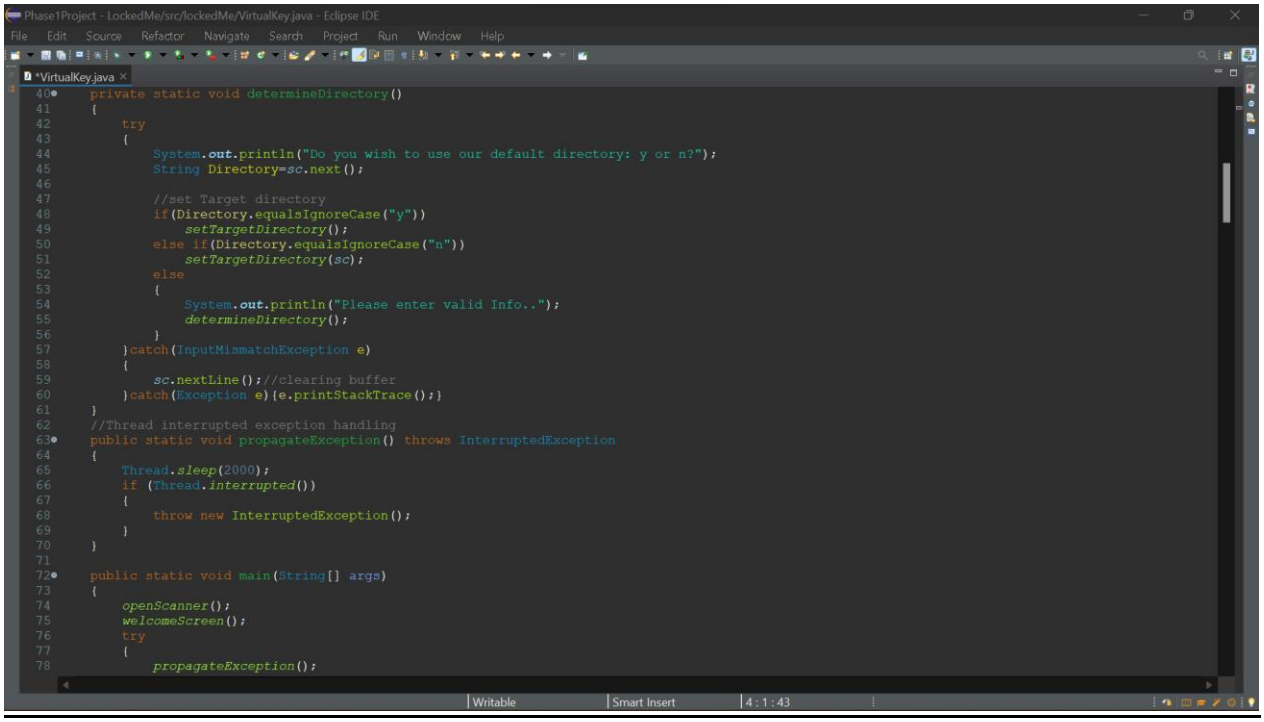



User selects option 2 from main menu
Business level options menu displayed
Option 1
Option 2
Option 3
Option 4
Create and add a new file to the directory
Delete a file from the existing directory
Search and display and file from the directory
Go back to main menu
Prompts user to enter the file name
Option 1
Option 2
Option 3
Displays the treeset of files from which user should select one to be deleted
File is deleted
Yes
No
"File "file name" deleted successfully" message is displayed in console
"Unable to delete the file. Try again. (Please note: Delete is case sensitive)" message is displayed
File name does not exist
Yes
No
Create a file object and provide the user specified name and add it to the existing directory
"File "file name " created successfully." message is displayed
"File already exists." message gets displayed
File is found
Yes
No
""file name" found in directory" message is displayed.
"File not found in directory "file path". *Search is case sensitive*
"Do you want to continue searching - y; any other statement will lead back to Business Menu"
Prompts user to enter y if they want to continue.
y is entered
Yes
No
Prompts to enter file name and continues with option 3.
Returns to Business menu

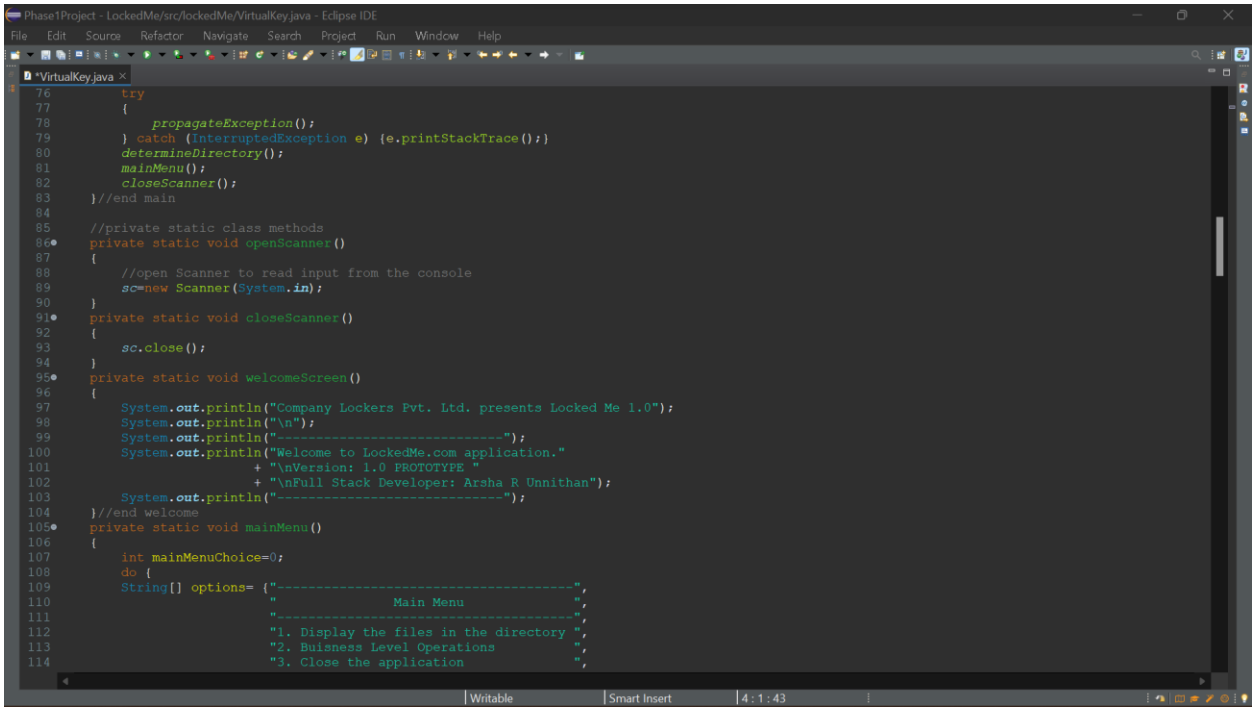
4. Source Code

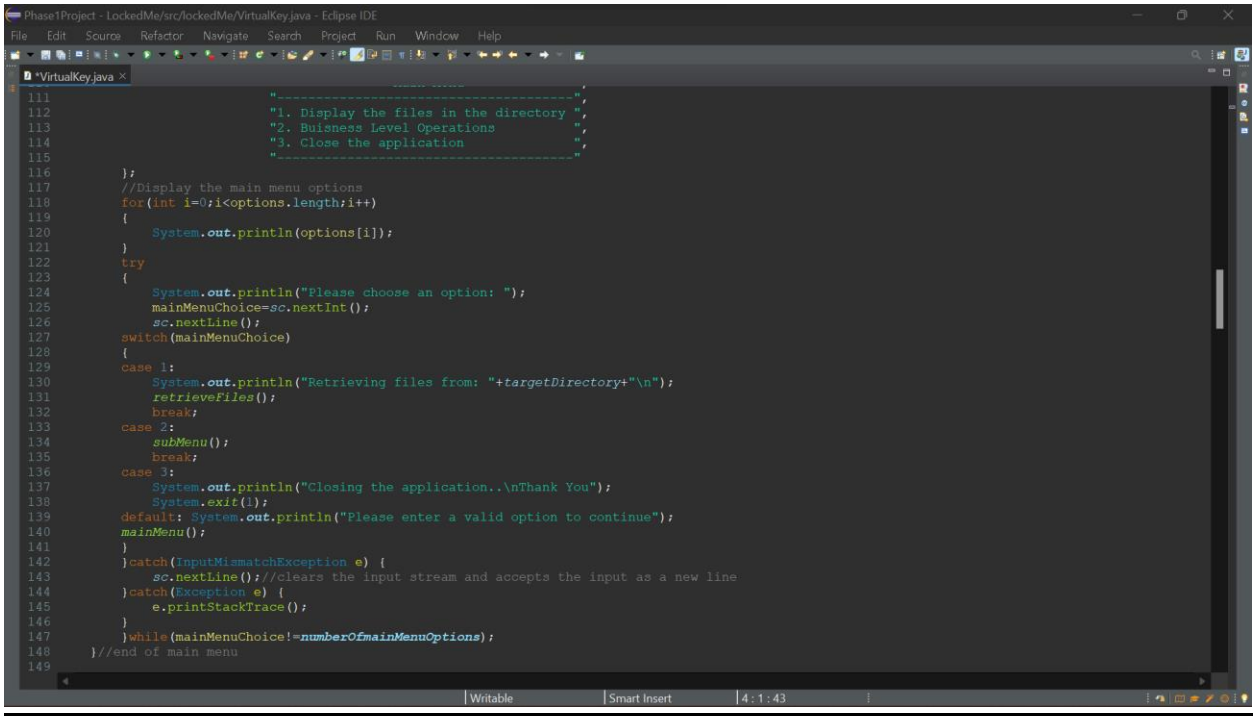


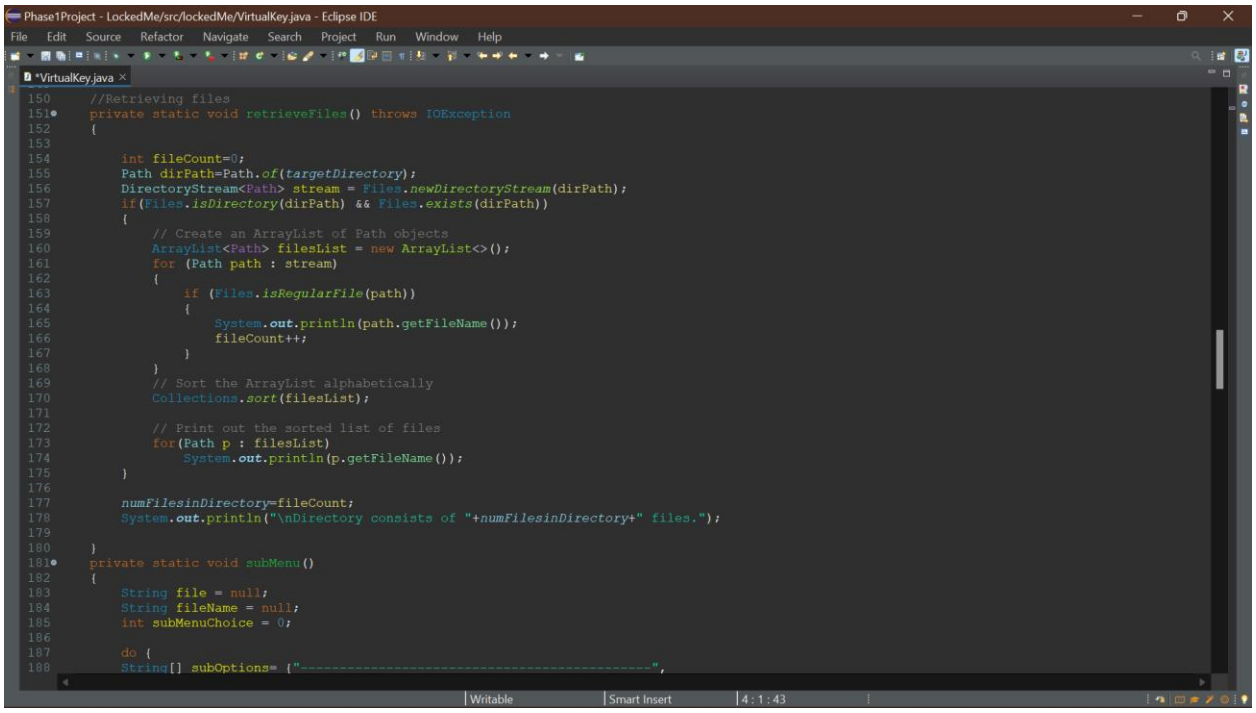

```java
package lockedMe;

import java.io.File;

import java.io.IOException;
import java.nio.file.DirectoryStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Collections;
import java.util.InputMismatchException;
import java.util.Iterator;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;

public class VirtualKey
{

    //class instance variables
    private static String targetDirectory=null;
    private static Scanner sc;
    private static int numFilesinDirectory=0;
    static String file = null;
    static String fileName = null;
    //to be updated on future modifications
    static final int numberOfmainMenuOptions=3;
    static final int numberOfsubMenuOptions=4;

    //private setter methods for directory
    private static void setTargetDirectory(Scanner s)
    {
        System.out.println("Enter the target directory for the application: ");
        targetDirectory=s.next();
    }
    private static void setTargetDirectory()
    {
        targetDirectory ="D:\\Java Project\\LockedMe\\src\\lockedMe\\Directory";
    }
```

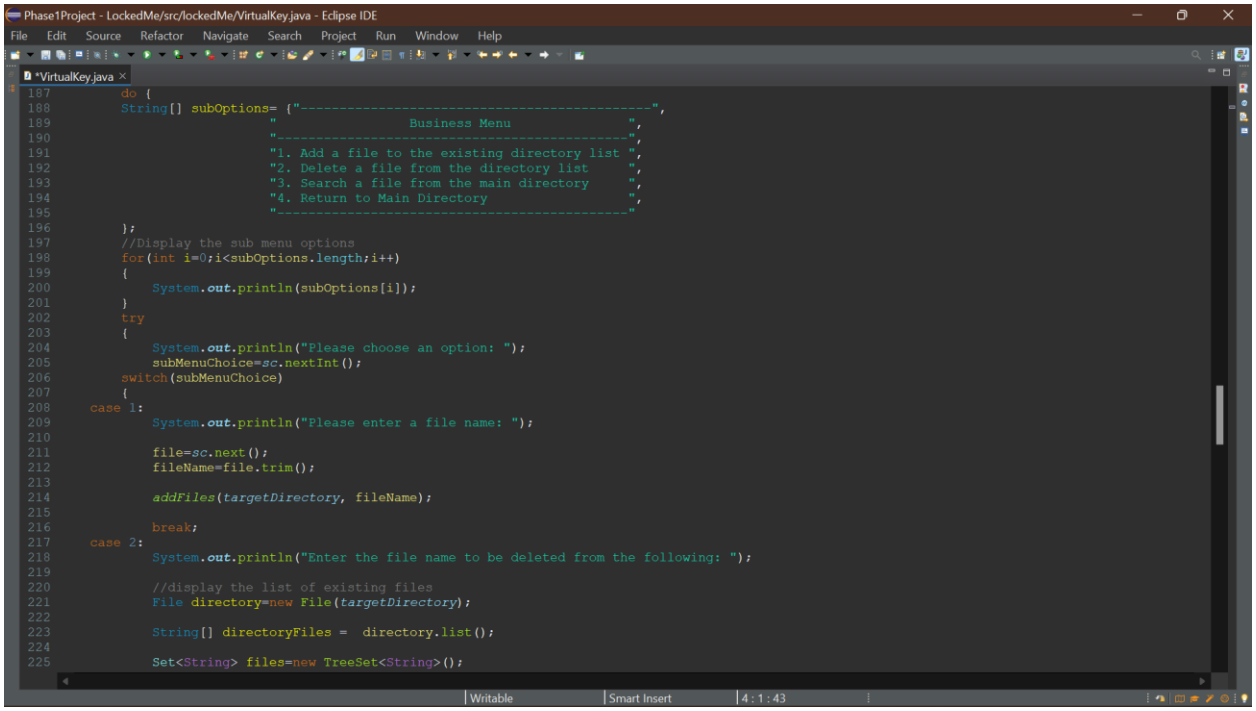


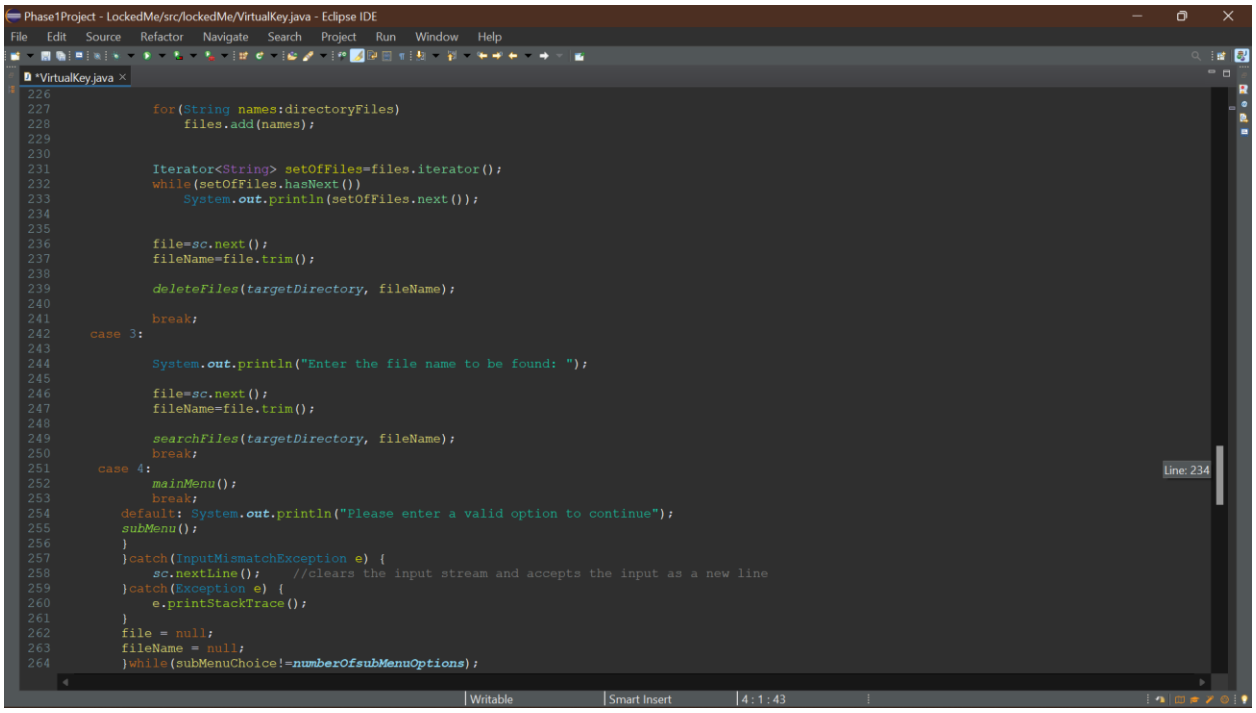

```java
    private static void determineDirectory()
    {
        try
        {
            System.out.println("Do you wish to use our default directory: y or n?");
            String Directory=sc.next();

            //set Target directory
            if(Directory.equalsIgnoreCase("y"))
                setTargetDirectory();
            else if(Directory.equalsIgnoreCase("n"))
                setTargetDirectory(sc);
            else
            {
                System.out.println("Please enter valid Info..");
                determineDirectory();
            }
        }catch(InputMismatchException e)
        {
            sc.nextLine();//clearing buffer
        }catch(Exception e){e.printStackTrace();}
    }
    //Thread interrupted exception handling
    public static void propagateException() throws InterruptedException
    {
        Thread.sleep(2000);
        if (Thread.interrupted())
        {
            throw new InterruptedException();
        }
    }

    public static void main(String[] args)
    {
        openScanner();
        welcomeScreen();
        try
        {
            propagateException();
```

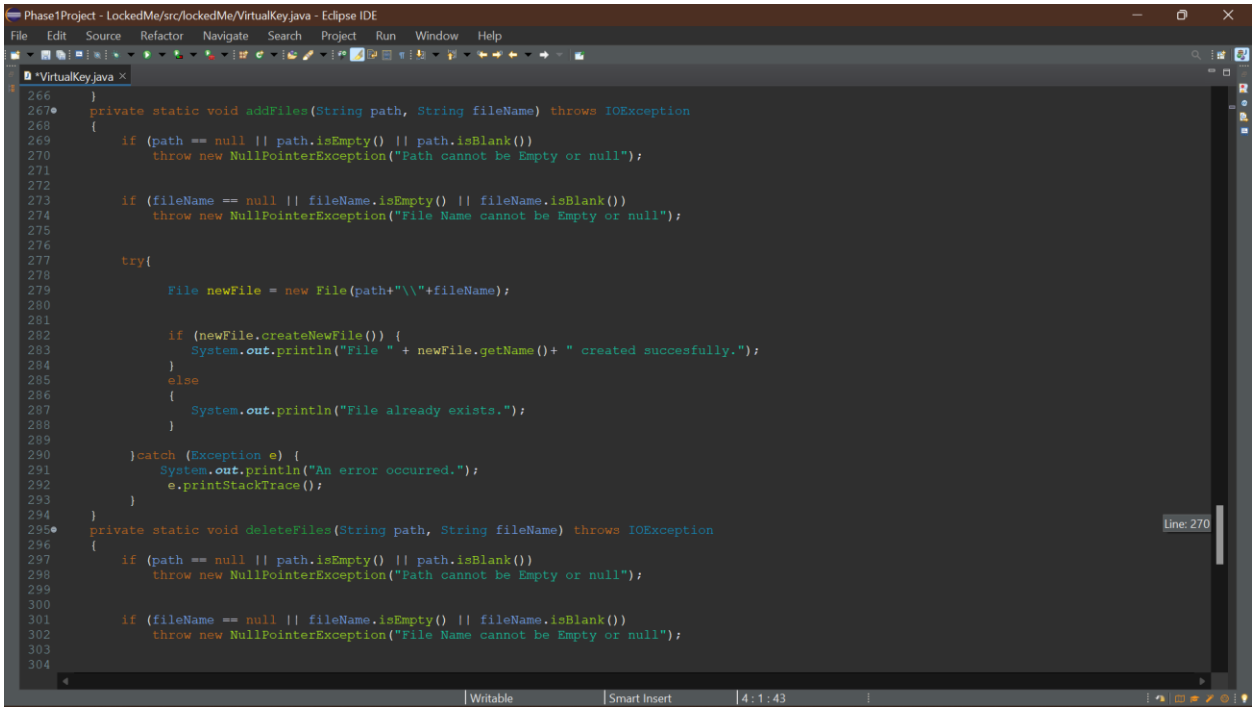

Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
try
{
propagateException();
} catch (InterruptedException e) {e.printStackTrace();}
determineDirectory();
mainMenu();
closeScanner();
}//end main
//private static class methods
private static void openScanner()
{
//open Scanner to read input from the console
sc=new Scanner(System.in);
}
private static void closeScanner()
{
sc.close();
}
private static void welcomeScreen()
{
System.out.println("Company Lockers Pvt. Ltd. presents Locked Me 1.0");
System.out.println("\n");
System.out.println("-----------------------------");
System.out.println("Welcome to LockedMe.com application."
+ "\nVersion: 1.0 PROTOTYPE "
+ "\nFull Stack Developer: Arsha R Unnithan");
System.out.println("-----------------------------");
}//end welcome
private static void mainMenu()
{
int mainMenuChoice=0;
do {
String[] options= {"--------------------------------------",
" Main Menu ",
"--------------------------------------",
"1. Display the files in the directory ",
"2. Buisness Level Operations ",
"3. Close the application ",
Writable Smart Insert 4 : 1 : 43

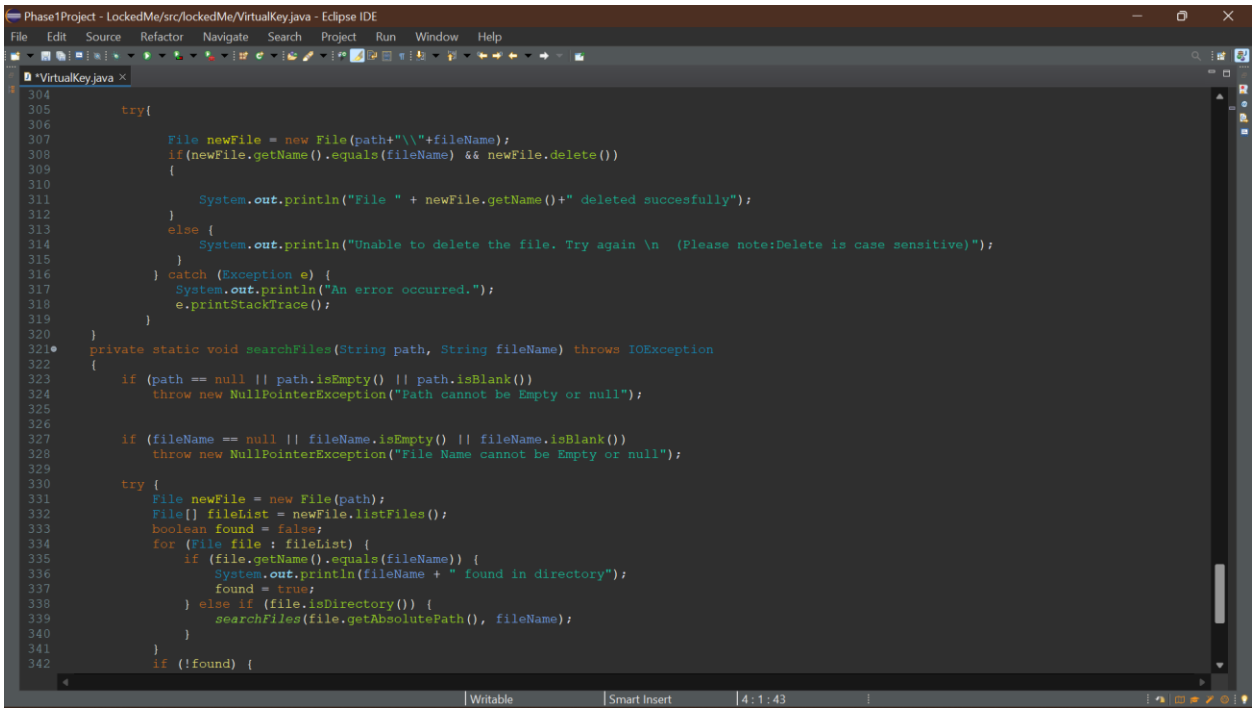

Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
"--------------------------------------",
"1. Display the files in the directory ",
"2. Buisness Level Operations ",
"3. Close the application ",
"--------------------------------------"
};
//Display the main menu options
for(int i=0;i<options.length;i++)
{
System.out.println(options[i]);
}
try
{
System.out.println("Please choose an option: ");
mainMenuChoice=sc.nextInt();
sc.nextLine();
switch(mainMenuChoice)
{
case 1:
System.out.println("Retrieving files from: "+targetDirectory+"\n");
retrieveFiles();
break;
case 2:
subMenu();
break;
case 3:
System.out.println("Closing the application..\nThank You");
System.exit(1);
default: System.out.println("Please enter a valid option to continue");
mainMenu();
}
}catch(InputMismatchException e) {
sc.nextLine();//clears the input stream and accepts the input as a new line
}catch(Exception e) {
e.printStackTrace();
}
}while(mainMenuChoice!=numberOfmainMenuOptions);
}//end of main menu
Writable Smart Insert 4 : 1 : 43

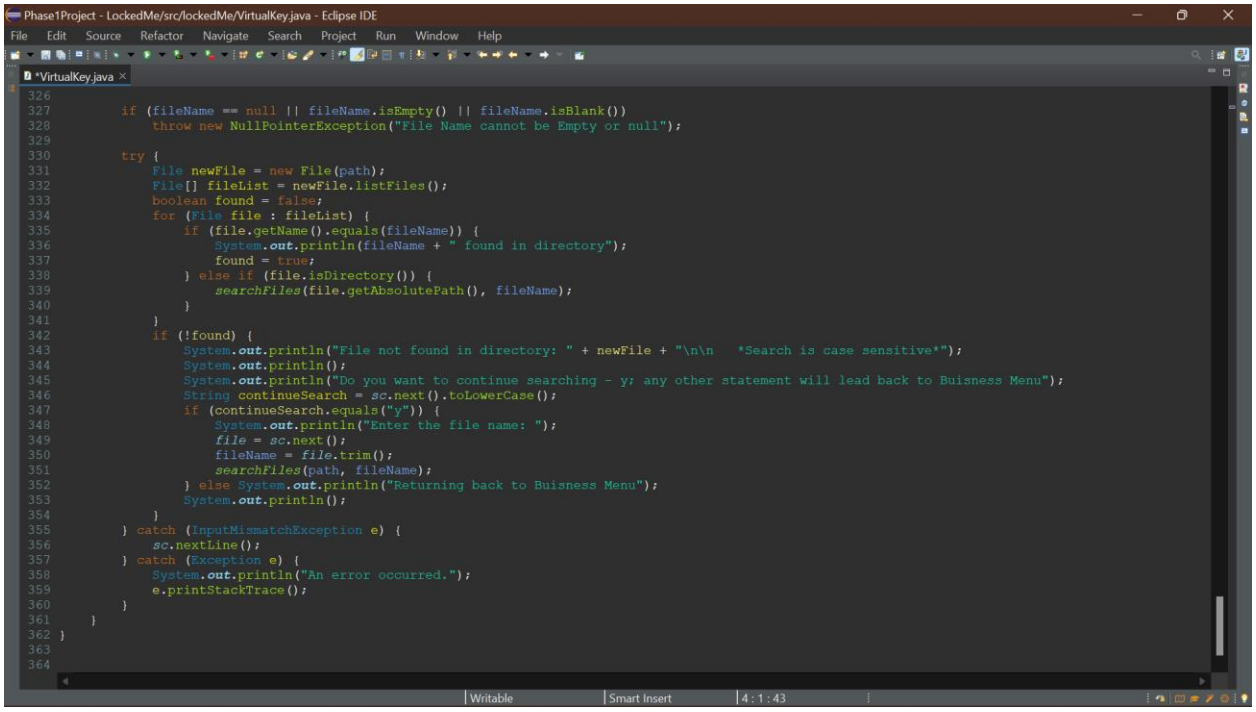

Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
//Retrieving files
private static void retrieveFiles() throws IOException
{
int fileCount=0;
Path dirPath=Path.of(targetDirectory);
DirectoryStream<Path> stream = Files.newDirectoryStream(dirPath);
if(Files.isDirectory(dirPath) && Files.exists(dirPath))
{
// Create an ArrayList of Path objects
ArrayList<Path> filesList = new ArrayList<>();
for (Path path : stream)
{
if (Files.isRegularFile(path))
{
System.out.println(path.getFileName());
fileCount++;
}
}
// Sort the ArrayList alphabetically
Collections.sort(filesList);
// Print out the sorted list of files
for(Path p : filesList)
System.out.println(p.getFileName());
}
numFilesinDirectory=fileCount;
System.out.println("\nDirectory consists of "+numFilesinDirectory+" files.");
}
private static void subMenu()
{
String file = null;
String fileName = null;
int subMenuChoice = 0;
do {
String[] subOptions= {"--",
Writable Smart Insert 4 : 1 : 43


Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
do {
String[] subOptions= {"--",
" Business Menu ",
"--",
"1. Add a file to the existing directory list ",
"2. Delete a file from the directory list ",
"3. Search a file from the main directory ",
"4. Return to Main Directory ",
"--"
};
//Display the sub menu options
for(int i=0;i<subOptions.length;i++)
{
System.out.println(subOptions[i]);
}
try
{
System.out.println("Please choose an option: ");
subMenuChoice=sc.nextInt();
switch(subMenuChoice)
{
case 1:
System.out.println("Please enter a file name: ");
file=sc.next();
fileName=file.trim();
addFiles(targetDirectory, fileName);
break;
case 2:
System.out.println("Enter the file name to be deleted from the following: ");
//display the list of existing files
File directory=new File(targetDirectory);
String[] directoryFiles = directory.list();
Set<String> files=new TreeSet<String>();
Writable Smart Insert 4 : 1 : 43


Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
for(String names:directoryFiles)
files.add(names);
Iterator<String> setOfFiles=files.iterator();
while(setOfFiles.hasNext())
System.out.println(setOfFiles.next());
file=sc.next();
fileName=file.trim();
deleteFiles(targetDirectory, fileName);
break;
case 3:
System.out.println("Enter the file name to be found: ");
file=sc.next();
fileName=file.trim();
searchFiles(targetDirectory, fileName);
break;
case 4:
mainMenu();
break;
default: System.out.println("Please enter a valid option to continue");
subMenu();
}
}catch(InputMismatchException e) {
sc.nextLine(); //clears the input stream and accepts the input as a new line
}catch(Exception e) {
e.printStackTrace();
}
file = null;
fileName = null;
}while(subMenuChoice!=numberOfsubMenuOptions);
Line: 234
Writable Smart Insert 4 : 1 : 43

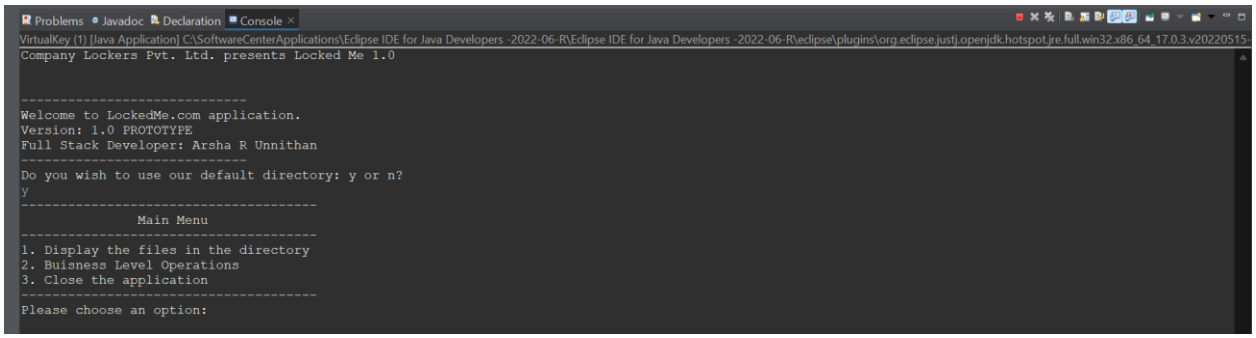

Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
}
private static void addFiles(String path, String fileName) throws IOException
{
if (path == null || path.isEmpty() || path.isBlank())
throw new NullPointerException("Path cannot be Empty or null");
if (fileName == null || fileName.isEmpty() || fileName.isBlank())
throw new NullPointerException("File Name cannot be Empty or null");
try{
File newFile = new File(path+"\\"+fileName);
if (newFile.createNewFile()) {
System.out.println("File " + newFile.getName()+ " created succesfully.");
}
else
{
System.out.println("File already exists.");
}
}catch (Exception e) {
System.out.println("An error occurred.");
e.printStackTrace();
}
}
private static void deleteFiles(String path, String fileName) throws IOException
{
if (path == null || path.isEmpty() || path.isBlank())
throw new NullPointerException("Path cannot be Empty or null");
if (fileName == null || fileName.isEmpty() || fileName.isBlank())
throw new NullPointerException("File Name cannot be Empty or null");
Line: 270
Writable Smart Insert 4 : 1 : 43


Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
try{
File newFile = new File(path+"\\"+fileName);
if(newFile.getName().equals(fileName) && newFile.delete())
{
System.out.println("File " + newFile.getName()+" deleted succesfully");
}
else {
System.out.println("Unable to delete the file. Try again \n (Please note:Delete is case sensitive)");
}
} catch (Exception e) {
System.out.println("An error occurred.");
e.printStackTrace();
}
}
private static void searchFiles(String path, String fileName) throws IOException
{
if (path == null || path.isEmpty() || path.isBlank())
throw new NullPointerException("Path cannot be Empty or null");
if (fileName == null || fileName.isEmpty() || fileName.isBlank())
throw new NullPointerException("File Name cannot be Empty or null");
try {
File newFile = new File(path);
File[] fileList = newFile.listFiles();
boolean found = false;
for (File file : fileList) {
if (file.getName().equals(fileName)) {
System.out.println(fileName + " found in directory");
found = true;
} else if (file.isDirectory()) {
searchFiles(file.getAbsolutePath(), fileName);
}
}
if (!found) {
Writable Smart Insert 4 : 1 : 43

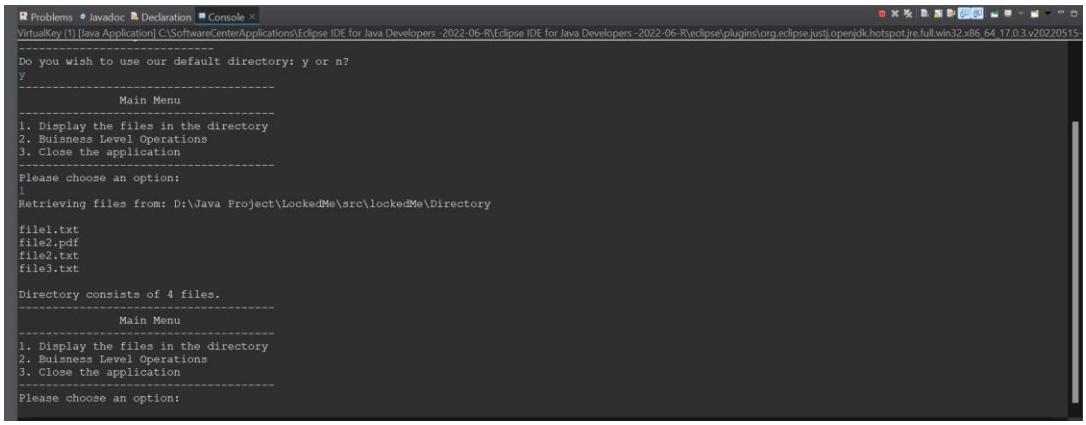

Phase1Project - LockedMe/src/lockedMe/VirtualKey.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*VirtualKey.java
if (fileName == null || fileName.isEmpty() || fileName.isBlank())
throw new NullPointerException("File Name cannot be Empty or null");
try {
File newFile = new File(path);
File[] fileList = newFile.listFiles();
boolean found = false;
for (File file : fileList) {
if (file.getName().equals(fileName)) {
System.out.println(fileName + " found in directory");
found = true;
} else if (file.isDirectory()) {
searchFiles(file.getAbsolutePath(), fileName);
}
}
if (!found) {
System.out.println("File not found in directory: " + newFile + "\n\n *Search is case sensitive*");
System.out.println();
System.out.println("Do you want to continue searching - y; any other statement will lead back to Buisness Menu");
String continueSearch = sc.next().toLowerCase();
if (continueSearch.equals("y")) {
System.out.println("Enter the file name: ");
file = sc.next();
fileName = file.trim();
searchFiles(path, fileName);
} else System.out.println("Returning back to Buisness Menu");
System.out.println();
}
} catch (InputMismatchException e) {
sc.nextLine();
} catch (Exception e) {
System.out.println("An error occurred.");
e.printStackTrace();
}
}
}
Writable Smart Insert 4 : 1 : 43

5. Outputs – screenshots

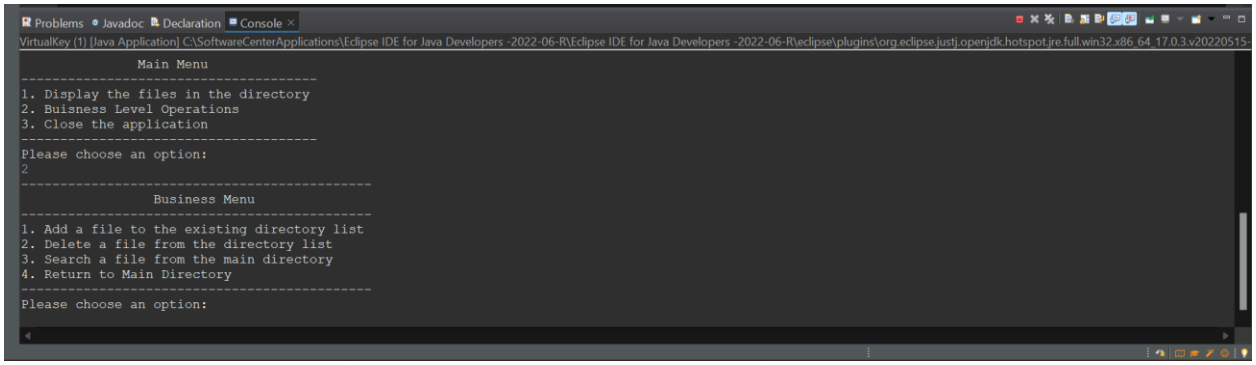
Upon Run, welcome screen:



Upon choosing option 1:



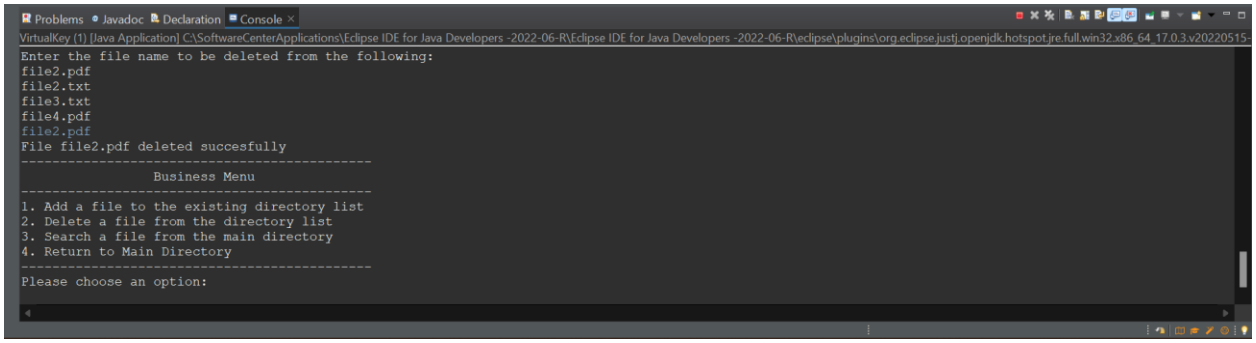
Upon choosing option 2:



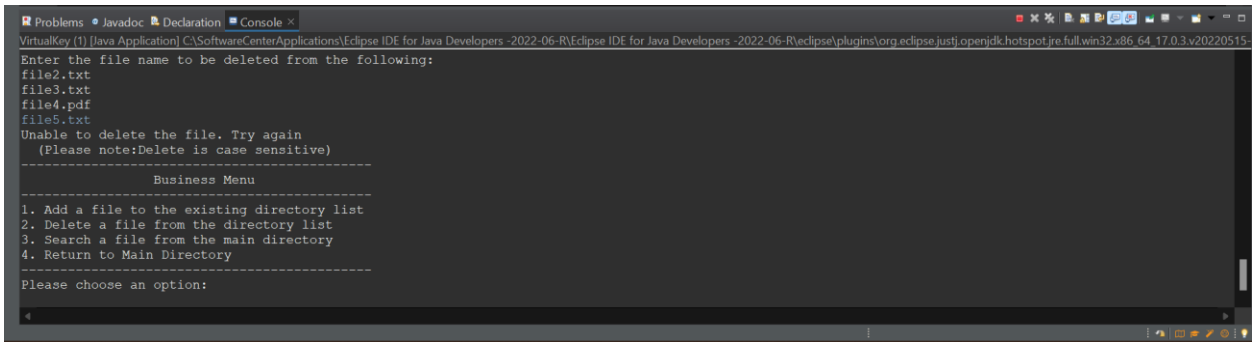
Upon choosing option 1 in business menu:

Upon choosing option 1 providing already existing file name:

Upon choosing option 2 in business menu:

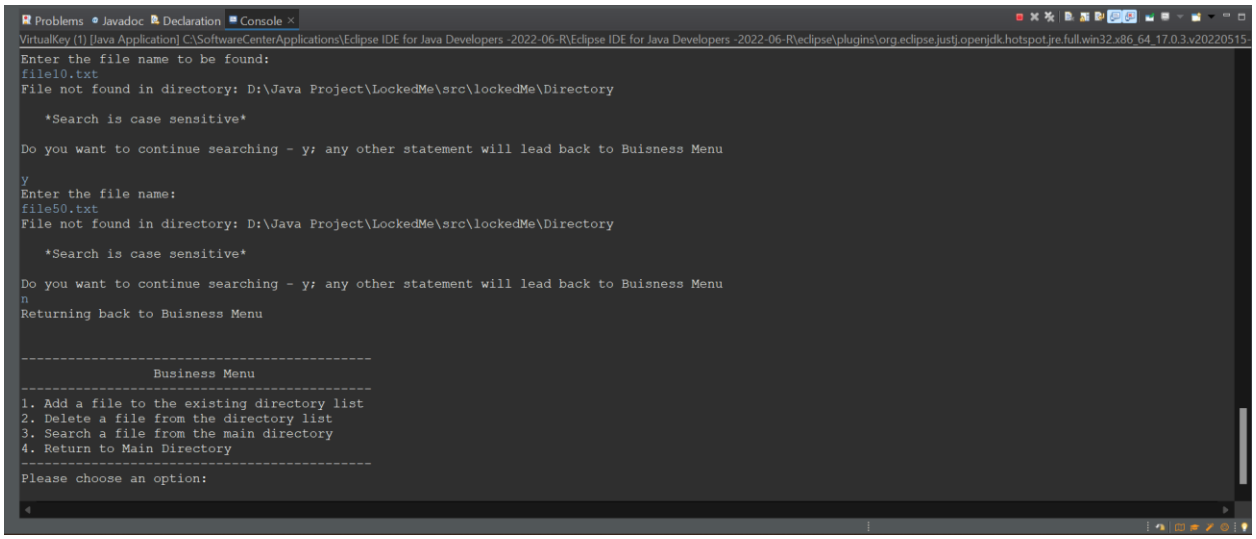


Upon choosing option 2 and giving invalid name:

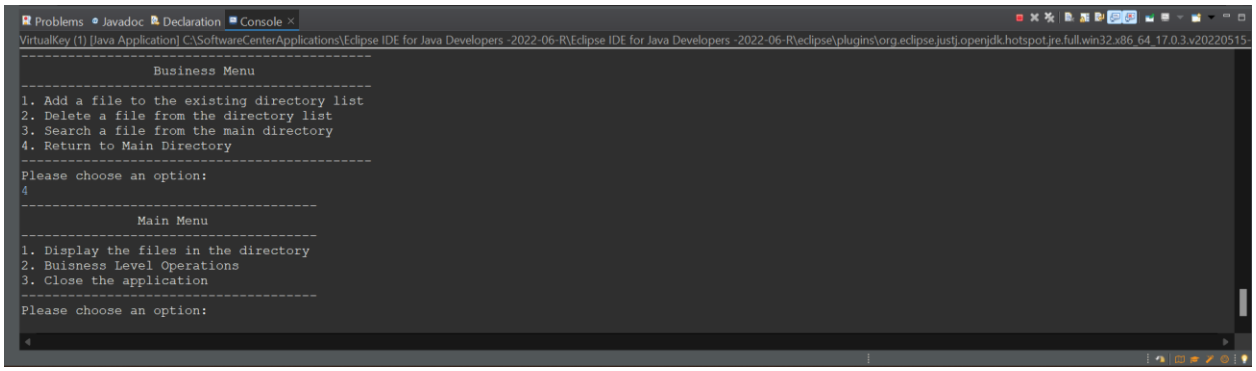


Upon choosing option 3 in business menu:

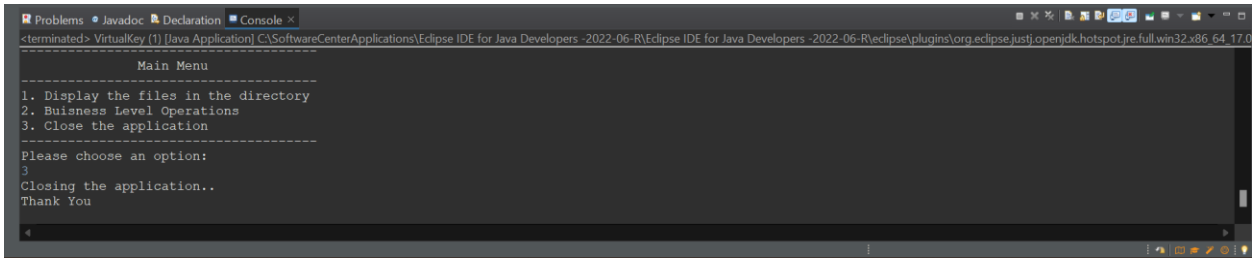
Upon choosing option 3 and giving invalid name:



Upon choosing option 4 business menu :



Upon choosing option 3 from main menu:



Conclusion:

Above application is just a prototype and it could be enhanced in various ways since lacks so much other features like advanced search filtering, robust file conversion tools, cloud integration, security features, automatic file conversion and versioning.

As for USPs, usability, robustness, performance, scalability, flexibility and reliability could be valid points.