ASSESMENT

Name: **Akash Chaurasiya**

Email: akashchaurasia888@gmail.com

Project name: **LockedMe.com**

Github Link :
https://github.com/AkashChaurasiya1996/Lockedm_Project

Table of Contents

ACKNOWLEDMENT

I would like to thank Simplilearn who gave me this opportunity to work on this project. I got to learn a lot from this project about implantation of object-oriented programming structure using data structure. At last, I would like to extend my heartfelt thanks to my parents because without their help this project would not have been successful. Finally, I would like to thank my dear friends who have been with me all the time. I would like to express my special thanks to our mentor Deepak Sir for his time and efforts he provided throughout the online session. Your useful advice and suggestions were helpful to me during the project's completion. In this aspect, I am eternally grateful to you. I would like to acknowledge that this project was completed entirely by me and not by someone else.

Signature

Akash Chaurasiya

PROBLEM STATEMENT

Virtual Key for Your Repositories

1 DESCRIPTION

Project objective:

As a Full Stack Developer, complete the features of the application by planning the development in terms of sprints and then push the source code to the GitHub repository. As this is a prototyped application, the user interaction will be via a command line.

Background of the problem statement:

Company Lockers Pvt. Ltd. hired you as a Full Stack Developer. They aim to digitize their products and chose LockedMe.com as their first project to start with. You're 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. Your manager has set up a meeting where you're asked to present the following in the next 15 working days (3 weeks):

- 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

You must use the following

- Eclipse/IntelliJ: An IDE to code for the application
- Java: A programming language to develop the prototype
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally
- Search and Sort techniques: Data structures used for the project
- Specification document: Any open-source document or Google Docs Following requirements should be met:
- The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.
- Document the step-by-step process starting from sprint planning to the product release.
- Application should not close, exit, or throw an exception if the user specifies an invalid input.
- You need to submit the final specification document which includes:
- Project and developer details
- Sprints planned and the tasks achieved in them
- Algorithms and flowcharts of the application
- Core concepts used in the project
- Links to the GitHub repository to verify the project completion
- Your conclusion on enhancing the application and defining the USPs (Unique Selling Points

Software Requirements

Sr no.	Requirements	Minimum Recomended
1	JAVA	VERSION 1.4.0 5 or Greater
	MEMORY	512 MB 1 GB or Greater
	Free Disk Space	300 MB 1 GB or More
	Processor	800 MHz 1.5 GHz
	Windows OS	Windows 7 or Greater

- System Requirement for GIT Desktop

- OS: Windows 7 or Greater(32 bit /64 bit)

PROPOSED METHOD

(a) The project will adopt the following components while implementing the methodology to achieve the expected results:

(b) We will use the do-while loop condition in our program.

(c) We will use the collection & array as well.

(d) We will have to use try and catch exception method otherwise during runtime we will get the error.

(e) We have to write the program in Java IDE i.e. Eclipse.

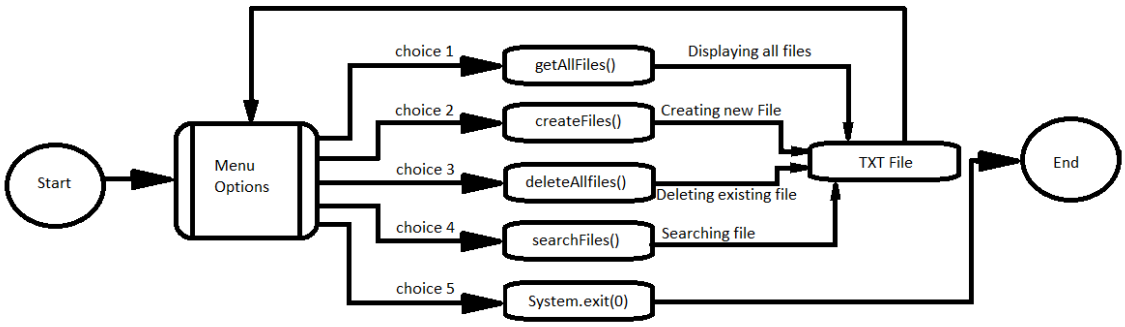
(f) This will help us to identify the error during the coding such as syntax error.

Flow Chart AND DFD

0 Level DFD

0 Level DFD

Flow Chart:



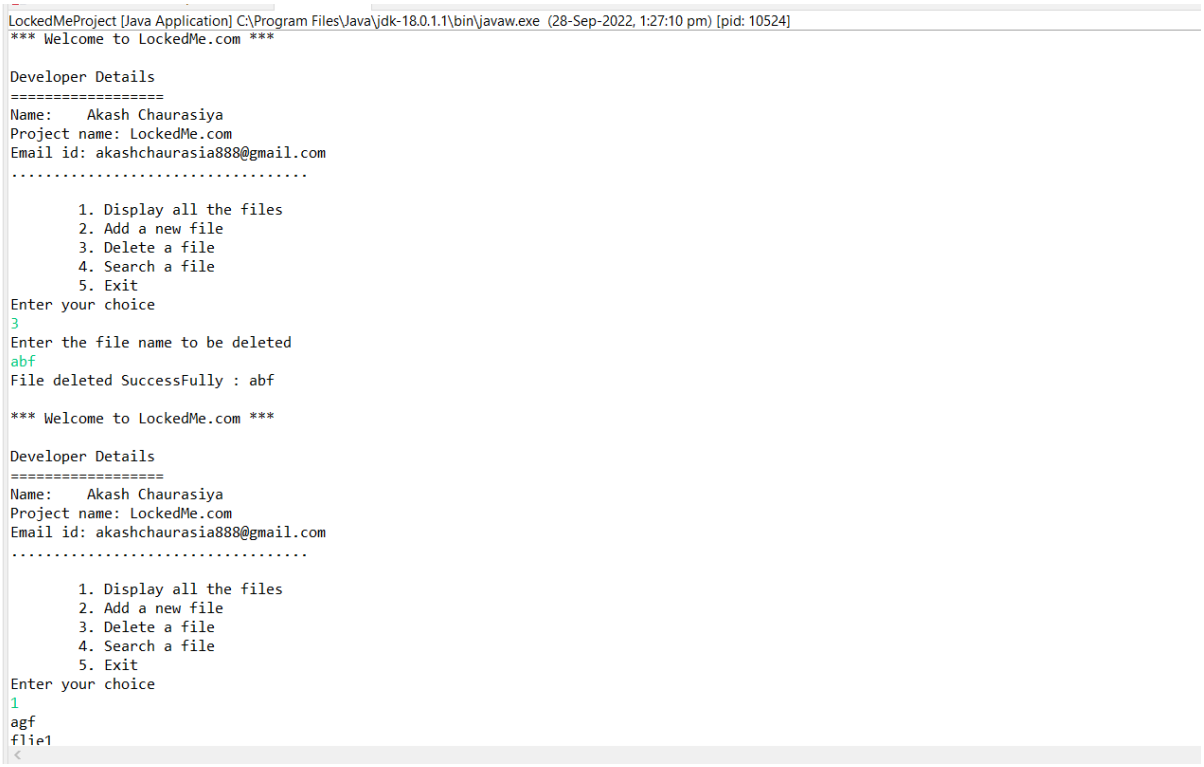
Flow Chart

Snapshots

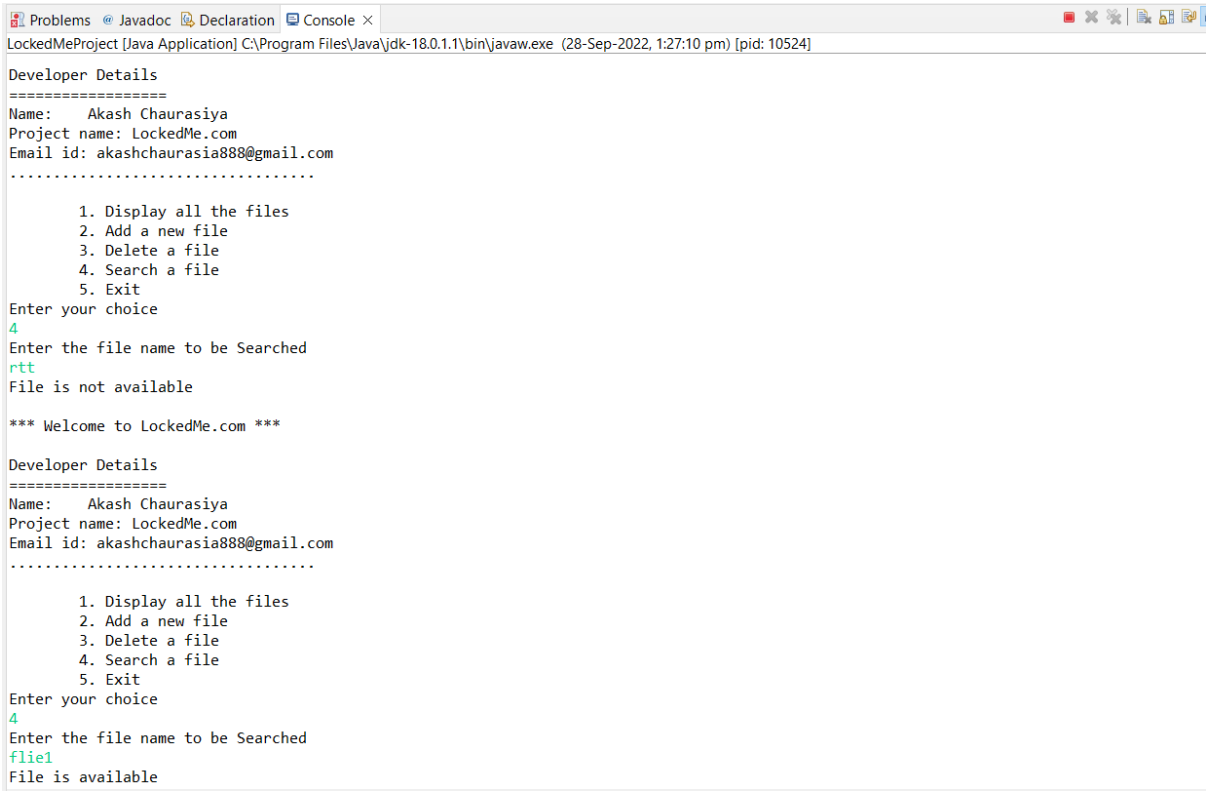
Display Files

Creating and editing file

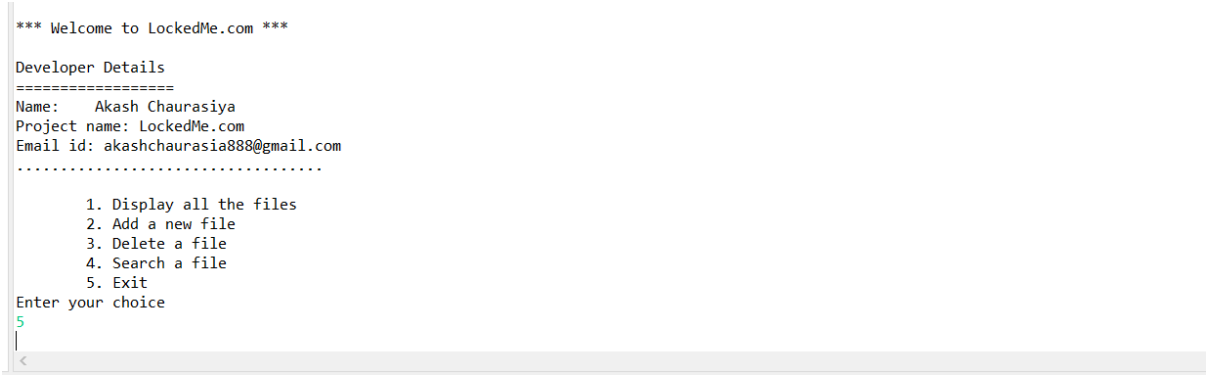
Before and after deleting file



Seraching File



Exit



Source Code

```
package com.Lockedm;

import java.util.Scanner;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.LinkedList;


public class LockedMeProject {

	static final String errorMessage = "Some error occured ";
	static final String projectFilesPath = "C:\\Users\\Lenovo\\Desktop\\java
eclipse\\jjjjj\\lockpro";
	public static void main(String args[])throws IOException {
		// TODO Auto-generated method stub
		{
			Scanner obj=new Scanner(System.in);
			int choice;

			do
			{
				displayMenuOptions();
				System.out.println("Enter your choice");
				choice=obj.nextInt();
				switch(choice)
				{
				case 1:
				getAllFiles();
				break;
				case 2:createFiles();
				break;
				case 3: deleteAllFiles();
				break;
				case 4: searchFiles();
				break;
				case 5: System.exit(0);
				break;
				default:System.out.println("Invalid option");
				}
			}while(option>0);
			}
				//obj.close();
		}

		public static void displayMenuOptions()
		{
			System.out.println("\n*** Welcome to LockedMe.com ***\n");
			System.out.println("Developer Details\n==================");
			System.out.println("Name:\t Akash Chaurasiya");
			System.out.println("Project name: LockedMe.com");
			System.out.println("Email id:
akashchaurasia888@gmail.com\n................................\n");

			System.out.println("\t1. Display all the files");
```

```
        System.out.println("\t2. Add a new file");
        System.out.println("\t3. Delete a file");
        System.out.println("\t4. Search a file");
        System.out.println("\t5. Exit");

}


/*This method will return all the files from the directory*/
public static void getAllFiles()
{
try {
File folder = new File(projectFilesPath);
File[] listOfFiles = folder.listFiles();
if(listOfFiles.length==0)
{
System.out.println("No Files exist");
}
else
{
for(var l:listOfFiles)
{
System.out.println(l.getName());
}
 }
 }
catch(Exception Ex)
{
System.out.println(errorMessage);
}
}


public static void createFiles() throws IOException
{
try
{
Scanner obj = new Scanner(System.in);
String fileName;
System.out.println("Enter the filename: ");
fileName = obj.nextLine();
int linesCount;
System.out.println("Enter how many lines in the file");
linesCount = Integer.parseInt(obj.nextLine());
FileWriter myWriter = new
FileWriter(projectFilesPath+"\\"+fileName);
for(int i=1;i<=linesCount;i++)
{
System.out.println("Enter the file line : ");
myWriter.write(obj.nextLine()+"\n");
}
System.out.println("File has been created successfully.");
myWriter.close();
//obj.close();
}
catch(Exception ex)
{
System.out.println("Some error has occcured");
}
```

```
                }


                /*This method will delete the file based on the user input if it
exists*/
                public static void deleteAllFiles()
                {
                Scanner obj = new Scanner(System.in);
                try
                {
                String fileName;
                System.out.println("Enter the file name to be deleted");
                fileName = obj.nextLine();
                File file = new File(projectFilesPath+"\\"+fileName);
                if(file.exists())
                {
                file.delete();
                System.out.println("File deleted SuccessFully : "+fileName);
                }
                else
                System.out.println("File do not exists");
                }
                catch(Exception ex)
                {
                System.out.println(errorMessage);
                }
                finally
                {
                //obj.close();
                }
                }
                /*This method will search the files from the directory*/
                public static void searchFiles()
                {
                Scanner obj = new Scanner(System.in);
                try
                {
                String fileName;
                System.out.println("Enter the file name to be Searched");
                fileName = obj.nextLine();
                File folder = new File(projectFilesPath);
                File[] listOfFiles = folder.listFiles();
                LinkedList<String> filenames = new LinkedList<String>();
                for(var l:listOfFiles)
                filenames.add(l.getName());
                if(filenames.contains(fileName))
                System.out.println("File is available");
                else
                System.out.println("File is not available");
                }
                catch(Exception ex)
                {
                System.out.println(errorMessage);
                } finally
                {
                //obj.close();
                }
                }
        }
```