**LockedMe.com Project Specification**

**Application Name:**

Lockedme.com

**Version History:**

|  |  |
| --- | --- |
| Author | Natalia Angarita |
| Purpose | Phase1 Project |
| Date | August 15 2021 |
| Version | 1.0 |

Contents

[1. Project GITHUB: 3](#_Toc80213694)

[2. Modules in the Project: 3](#_Toc80213695)

[3. Java concepts used in LockedMe.com Application: 6](#_Toc80213696)

[4. Sprint wise work: 7](#_Toc80213697)

[5. Project Code 7](#_Toc80213698)

# Project GITHUB:

* Project GITHUB link: https://github.com/nataliaangaritab/LockedMeProject.git
* Repository name: LockedMeProject

# Modules in the Project:

**Main Menu:**

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

* Display file names list: Method will return file names from the folder (folderpath).

A screenshot of a computer

Description automatically generated with medium confidence

* Add new file: Method will create or append content into the file specified.

A screenshot of a computer

Description automatically generated with medium confidence

* Delete a file: Method will delete the file name if it exists.

A screenshot of a computer

Description automatically generated with medium confidence

* Search a file: Method will search for a file specified by the user.

A screenshot of a computer

Description automatically generated with medium confidence

* Exit application.

A screenshot of a computer

Description automatically generated with medium confidence

# Java concepts used in LockedMe.com Application:

* Exception handling
* Working with files
* Naming Standards
* Modularity
* Object Oriented Programming
* Collections
* Control Structures
* Data Structures

# Sprint wise work:

|  |  |
| --- | --- |
| Sprint Number | Modules |
| Sprint 1 | Create application main directory  Display all files  Add new file |
| Sprint 2 | Delete a file  Search a file  Exit application |
| Sprint 3 | Testing  Deployment (Create jar file) |

# Project Code

1. Folder Structure

Graphical user interface, text, application

Description automatically generated

1. File: LockedMeProject.java

**package** lockedme;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** LockedMeProject

{

//Create path variable

**static** **final** String ***folderpath***="C:\\Users\\natal\\Desktop\\Simplilearn\\Phase1-LockedMeProject\\Files";

**static** Scanner *obj* = **new** Scanner(System.***in***);

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

{

**int** menuStatus = 1;

**do**

{

//Variable declaration

**int** userChoice;

//Menu

*displayMenu*();

System.***out***.println("Enter your choice:");

userChoice = Integer.*parseInt*(*obj*.nextLine());

**switch** (userChoice)

{

**case** 1:

*getFiles*();

**break**;

**case** 2:

*createFile*();

**break**;

**case** 3:

*deleteFile*();

**break**;

**case** 4:

*searchFile*();

**break**;

**case** 5:

System.*exit*(0);

**break**;

**default**: System.***out***.println("Invalid Option");

}

} **while**(menuStatus > 0);

}

/\*\*

\* Method will display menu so user can select an option

\*/

**public** **static** **void** displayMenu()

{

System.***out***.println("================================================================");

System.***out***.println("\tCompany Lockers Pvt. Ltd. - LockedMe.com Project");

System.***out***.println("================================================================");

System.***out***.println("1. Display file names list");

System.***out***.println("2. Add new file");

System.***out***.println("3. Delete a file");

System.***out***.println("4. Search a file");

System.***out***.println("5. Exit application");

System.***out***.println("================================================================");

}

/\*\*

\* Method will get file name in a list

\*/

**public** **static** **void** getFiles()

{

List<String> fileNames = FileManager.*getFiles*(***folderpath***);

**if** (fileNames.size()==0)

System.***out***.println("No files in directory");

**else**

{

System.***out***.println("FILES LIST IS BELLOW:");

**for**(String f:fileNames)

System.***out***.println(f);

}

}

/\*\*

\* Method will create a new file with specific number of lines

\*/

**public** **static** **void** createFile()

{

//Variable declaration

String fileName;

**int** linesCount;

List<String> content = **new** ArrayList<String>();

// Read file name from user

System.***out***.println("Enter file name: ");

fileName = *obj*.nextLine();

// Read number of lines from user

System.***out***.println("Enter number of lines in the file:");

linesCount = Integer.*parseInt*(*obj*.nextLine());

// Read lines from user

**for** (**int** i = 1; i <= linesCount; i++) {

System.***out***.println("Enter line " + i + ":");

content.add(*obj*.nextLine());

}

// Save content into the file

**boolean** isSaved = FileManager.*createFile*(***folderpath***, fileName, content);

**if** (isSaved)

System.***out***.println("File and data saved successfully!");

**else**

System.***out***.println("Some error ocurred. Please contact Support !!!");

}

/\*\*

\* Method will delete a file by name

\*/

**public** **static** **void** deleteFile()

{

//Delete a file

String fileName;

System.***out***.println("Enter file name to be deleted:");

fileName = *obj*.nextLine();

**boolean** isDeleted = FileManager.*deleteFile*(***folderpath***, fileName);

**if**(isDeleted)

System.***out***.println("File deleted successfully!!");

**else**

System.***out***.println("File not in folder or some access issue");

}

/\*\*

\* Method will search for a file by name

\*/

**public** **static** **void** searchFile()

{

//Search a file

String fileName;

System.***out***.println("Enter file name to be searched:");

fileName = *obj*.nextLine();

**boolean** isFound = FileManager.*searchFile*(***folderpath***, fileName);

**if**(isFound)

System.***out***.println("File was found on the folder!!");

**else**

System.***out***.println("File not in folder");

}

}

1. File: FileManager.java

**package** lockedme;

**import** java.io.File;

**import** java.io.FileWriter;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** FileManager

{

/\*\*

\* Method will return file names from the folder (folderpath).

\* **@param** folderpath

\* **@return** List<String>

\*/

**public** **static** List<String> getFiles(String folderpath)

{

//Create file object

File newfile = **new** File(folderpath);

//Get all the files into FileArray

File[] listOfFiles = newfile.listFiles();

//Declare list to store file names

List<String> fileNames = **new** ArrayList<String>();

Collections.*sort*(fileNames);

**for**(File f:listOfFiles)

fileNames.add(f.getName());

//Return the List

**return** fileNames;

}

/\*\*

\* Method will create or append content into the file specified.

\* **@param** folderpath

\* **@param** fileName

\* **@param** content

\* **@return** boolean

\*/

**public** **static** **boolean** createFile(String folderpath, String fileName, List<String> content)

{

**try**

{

File newfile = **new** File(folderpath, fileName);

FileWriter fw = **new** FileWriter(newfile);

**for**(String s:content)

{

fw.write(s+"\n");

}

fw.close();

**return** **true**;

}

**catch**(Exception Ex)

{

**return** **false**;

}

}

/\*\*

\* Method will delete the file name if it exists.

\* **@param** folderpath

\* **@param** fileName

\* **@return** boolean

\*/

**public** **static** **boolean** deleteFile(String folderpath, String fileName)

{

//Add folderpath with file name and create file object

File file = **new** File(folderpath + "\\" + fileName);

**try**

{

**if**(file.delete())

**return** **true**;

**else**

**return** **false**;

}

**catch**(Exception Ex)

{

**return** **false**;

}

}

/\*\*

\* Method will search for a file specified by the user.

\* **@param** folderpath

\* **@param** fileName

\* **@return** boolean

\*/

**public** **static** **boolean** searchFile(String folderpath, String fileName)

{

File file = **new** File(folderpath + "\\" + fileName);

**if**(file.exists())

**return** **true**;

**else**

**return** **false**;

}

}