LockedMe.com
(Sprint work and Project Specifications)

Author	Pooja Vishwakarma
Purpose	Sprint work and specifications
Date	13th August 2021
Version	1.0

Table of Contents

1. Modules in the Project

1.1 Display all files

Description : This funcntionality will list down all the files that are present in the locker folder. User have to select option 1 and application will display the file count and files in ascending order.

1.2 Add a new file

Description : Using this module, user will be able to add a new file in the locker folder. User will have to select option 2 and application will ask for file name and the file content. User will enter file name and number of lines to be added. Then user will be prompted to add the content line by line. Application will save the file and display the message to the user if file is successfully saved. If any error occurs and application is not able to save the new file so in that case , user will be notified about the same.

1.3. Delete a file

Description : This funcntionality will delete the user specified file from the locker folder. User will have to select option 3 and application will prompt user to enter the file name that is to be deleted. After the user has entered the file name, application will delete that file and display the message to the user if succussfully deleted. User will also be notified if due to some issue file was deleted.

1.4 Search a file

Description : Using this functionality user can search if a file is present in the locker folder or not. User will have to select option 4 and enter the file name. Application will search the specified fi le in the locker folder. If found, application will display that specified file is present in the folder. If file is not present, user will be notified for same.

1.5. Exit.

Description : User can exit the application after the desired action are done by entering 5.

2. Technology used

- IDE – Eclipse
- Language – Java
- Ubuntu Terminal
- Github

Java Concept used-

- Data Structures
- File Handling
- Sorting
- Modularity
- Collections framework
- Flow Control structures
- Naming standards
- Object oriented programming

3. Sprint planning and task completion

This Project was completed in 2 sprints. Each sprint was of 2 weeks. Tasks divided and completed in each sprints are -

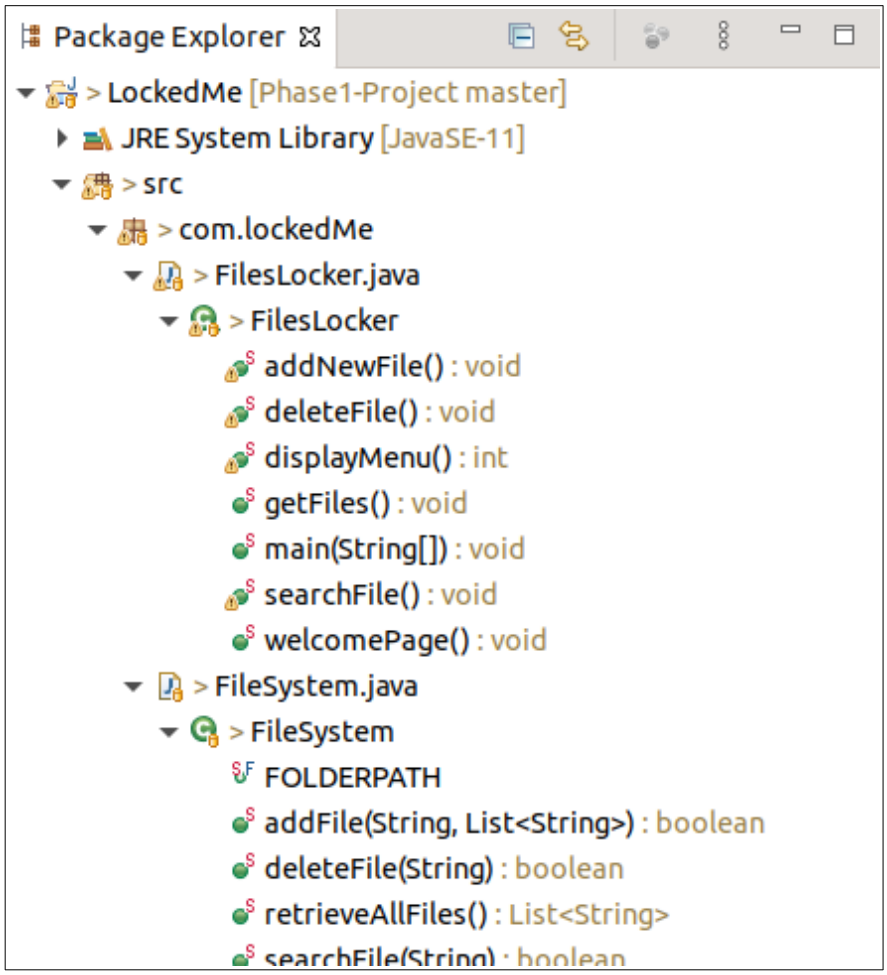
Sprint Number	Tasks
Sprint 1	- Creating application flow - Designing and coding the first module ‘Display all files’. -Designing and coding the second module ‘Add a new file’. - Testing both the modules - Pushing the code to Github.
Sprint 2	- Designing and coding the third module ‘Delete a file’. - Designing and coding the fourth module ‘Search a file’. - Designing and coding the ‘Exit’ functionality. - Unit and integration testing of all modules. - Deployment - Documentation

4. Project Github link

Repository Name	Phase1-Project
Github link	https://github.com/Pooja-vish06/Phase1-Project

5. Project code and folder structure

5.1. Folder Structure



5.2. Project Code

FilesLocker.java

```
package com.lockedMe;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;

public class FilesLocker {

	public static void main(String[] args) {

		// Variable declaration
		int option = 1;

		// loop till user exits the system
		do {
			welcomePage();
			int choice = displayMenu();
			switch (choice) {
			case 1:
				getFiles();
				break;
			case 2:
				addNewFile();
				break;
			case 3:
				deleteFile();
				break;
			case 4:
				searchFile();
				break;
			case 5:
				option = 0;
				System.out.println("Thank you for using our
services.");
				System.out.println("Developed by - Pooja
Vishwakarma.");
				System.exit(0);
				break;
			default:
				System.out.println("Invalid input. Please enter
valid option. ");
			}

		} while (option > 0);

	}

	// =====WELCOME PAGE=====
	public static void welcomePage() {
```

```
System.out.println("================================================");
            System.out.println("$\t\tLOCKER PVT LTD.\t\t\t$");
            System.out.println("\tDeveloped by - Pooja Vishwakarma");


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

      }

      // ======DISPLAY MENU======
      public static int displayMenu() {

            //
            Scanner sc = new Scanner(System.in);

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


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

            // take choice from user.

            try {
                  int choice = Integer.parseInt(sc.nextLine());
                  return choice;
            } catch (NumberFormatException e) {
                  System.out.println("You've entered non-integer number.
Please enter valid option from below.");


System.out.println("================================================");
                  return displayMenu();
            }

      }

      // ======GET ALL FILES======
      public static void getFiles() {
            int count = 0;
            List<String> files = FileSystem.retrieveAllFiles();
            count = files.size();
            System.out.println("Total " + count + " files are present.");
            System.out.println("Files are : ");
            Collections.sort(files);
            for (String s : files) {
                  System.out.println(s);
            }
      }

      // ======Add A FILE======
      public static void addNewFile() {
```

```
            Scanner sc = new Scanner(System.in);
            List<String> fileContent = new ArrayList<String>();

            // User will input the file name
            System.out.println("Enter your file name :");
            String fileName = sc.nextLine();

            // User will enter number of lines here
            System.out.println("Enter number of lines you want to add :");
            int linesCount = Integer.parseInt(sc.nextLine());

            // Loop through number of lines
            for (int i = 1; i <= linesCount; i++) {
                  System.out.println("Enter line " + i + ":");
                  String lines = sc.nextLine();
                  fileContent.add(lines);
            }

            boolean fileAdded = FileSystem.addFile(fileName, fileContent);
            if (fileAdded)
                  System.out.println("File saved successfully.");
            else {
                  System.out.println("File could not be saved. Try again
later.");
                  System.out.println("if probelm persists, contact admin
admin@lockerme.com.");
            }

      }

      // ======DELETE A FILE======
      public static void deleteFile() {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the file name to be deleted :");
            String fileName = sc.nextLine();
            boolean isDeleted = FileSystem.deleteFile(fileName);
            if (isDeleted) {
                  System.out.println(fileName + " deleted successfully.");
            } else {
                  System.out.println(fileName + " couldn't be deleted. Try
again later.");
                  System.out.println("if probelm persists, contact admin
admin@lockerme.com.");
            }

      }

      // ======SEARCH A FILE======
      public static void searchFile() {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter file name to be searched");
            String fileName = sc.nextLine();
            boolean isFound = FileSystem.searchFile(fileName);
            if (isFound)
                  System.out.println(fileName + " is present in the
locker.");
            else
```

```
                  System.out.println(fileName + " is not present in the
locker.");

      }

}
```

FileSystem.java

```
package com.lockedMe;

import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;

public class FileSystem {

      // variable declaration
      public static final String FOLDERPATH =
"/home/pooja/Documents/Simplilern/Phase1-Project/Files";

      /**
       * This method provides the name of all files present in the defined
folder.
       *
       * @return List of files
       * @String
       */
      public static List<String> retrieveAllFiles() {

            File file = new File(FOLDERPATH);

            File[] fileNames = file.listFiles();

            List<String> allFiles = new ArrayList<String>();

            for (File f : fileNames)
                  allFiles.add(f.getName());

            return allFiles;

      }

      /**
       * This method creates a new file and adds the content provided by the
user,
       * returns true if file is added.
```

```
	 *
	 * @param fileName
	 * @param fileContent
	 * @return boolean
	 */
	public static boolean addFile(String fileName, List<String>
fileContent) {

		File file = new File(FOLDERPATH + "/" + fileName);

		try {
			FileWriter fw = new FileWriter(file);
			for (String content : fileContent)
				fw.write(content + "\n");

			fw.close();
			return true;
		} catch (Exception ex) {

			return false;
		}

	}

	/**
	 * This method deletes the file specified by user, returns true if
deleted.
	 *
	 * @param fileName
	 * @return boolean
	 */
	public static boolean deleteFile(String fileName) {

		File file = new File(FOLDERPATH + "/" + fileName);
		if (file.delete())
			return true;
		else
			return false;

	}

	/**
	 * This method searches the user specified file in the locker folder,
returns
	 * true if found.
	 *
	 * @param fileName
	 * @return boolean
	 */
	public static boolean searchFile(String fileName) {

		File file = new File(FOLDERPATH + '/' + fileName);
		if (file.exists())
			return true;
		else
			return false;
```

```
        }
    }
```