LockedMe – Virtual Key for Repositories

(Write Up Specification Document)

Version History

Author	Rabin Kumar Pati
Doc purpose	Specification Document
Date	15-Aug -2021
Version	01

Table of Contents

Project Information

This section intends to provide a high-level picture of the project as well as document details of the project.

Include the following in this section:

Project Title: Virtual Key repository

Project version history: Document version detail with editor and date.

Project Summary: This application designed with java core concepts includes the functionalities for users to create, view, update, and delete file from folder through command line.

Project Timeframe: 2 Weeks

Prepared By: Rabin Kumar Pati

Attached Documentation: The Projects Screen shots as well as Code Screen shots in pdf format with writeup specification document.

Project Contacts: Rabin Kumar Pati

Project Details

- Sprint planning and Task completion
- Technologies used in project
- Flow of the Application.
- Demonstrating the user interactions through Command Line.
- Project package structure.
- Pushing the code to GitHub repository
- Discussion Points of the Application
- Conclusions
- Appendix

The code for this project is hosted at

https://github.com/RABINPATI/LockedMeAssessment

The project is developed by **RABIN KUMAR PATI.**

1. Sprints planning and Task completion

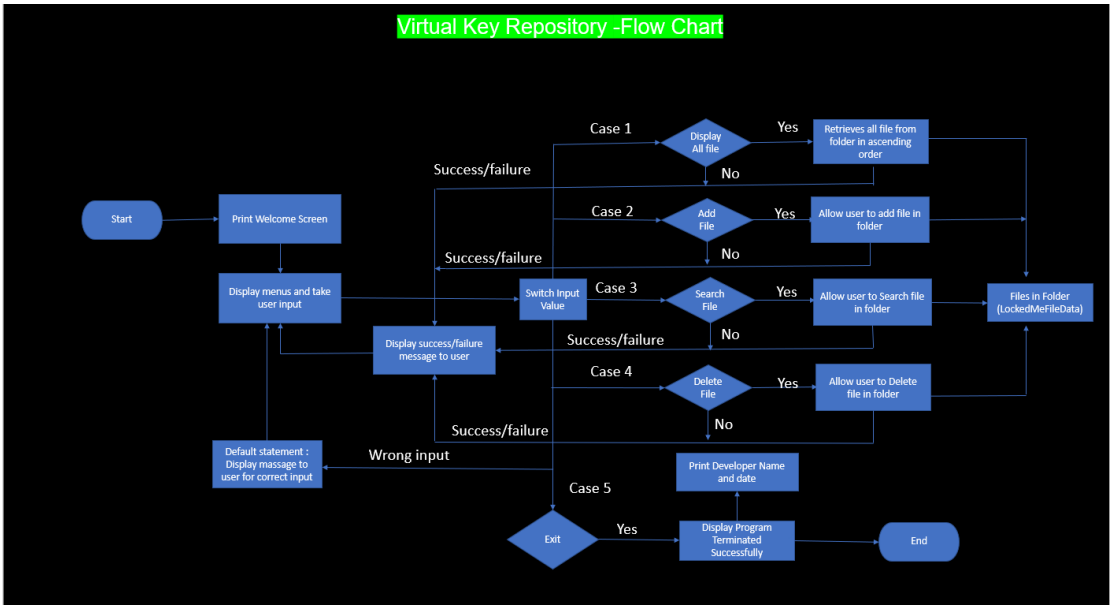
The project is planned to be completed in 3 sprints. Tasks assumed to be completed in the sprint are:

Sprint 1	Gathering the Requirements. Install related Software. Modularized the project. Design the process Flow.
Sprint 2	Development Module 1: Create Menus Module 2: Display all files. Module 3: Add file Testing all 3 modules Git Commit to repository
Sprint 3	Module 4: Search a file Module 5: Delete file Testing all 5 modules Git Commit to repository Create Documentation

2. Technologies used in project

Java Concepts	OOPS Concept File Handling Exception Handling Collections framework java 8 API
IDE	Eclipse
Software Tool	GitHub Desktop Snipping Tool
Documentation	MS Office

3. Flow Of Application



Virtual Key Repository -Flow Chai

4. Demonstrating user interactions

- Creating the project in Eclipse.
- Writing a program in Java for the entry point of the application (**LockedMeMain.java**).
- Writing a program in Java to display Menu options available for the user (**MenuOptions.java**).
- Writing a program in Java to perform the File operations as specified by user (**FileDataOperation.java**).
- Writing a program in Java to perform the File operations logic to perform file operation (**FileDataManager.java**).

4.1 Creating a new project in Eclipse

- Open Eclipse
- Go to File -> New -> Project -> Java Project -> Next.
- Type in any project name and click on “Finish.”
- Select your project and go to File -> New -> Class.
- Enter **LockedMeMain** in any class name, check the checkbox “public static void main(String[] args)”, and click on “Finish.”

4.2 Entry Point of Assessment Application

```java
package com.file.data;

public class LockedMeMain {

    // Folder Path for file to access data

    static final String FOLDER_PATH = "D:\\Project\\FSD\\Phase 1\\LockedMeAssessment\\LockedMeFileD

    public static void main(String[] args) {

        // Display Application name

        MenuOptions.displayWelcomeHeaderNote();

        // Display file processing operations

        MenuOptions.menuProcessOperations(FOLDER_PATH);

    }

}
```

https://github.com/RABINPATI/LockedMeAssessment/blob/main/LockedMe/src/com/fileData/LockedMeMain.java

4.3 Display Menu options

MenuOptions consists below methods for -:

- **displayWelcomeHeaderNote** for displaying Application name.
- **displayWelcomeFooterNote** for displaying developer name and current date.
- **displayFileMenuOptions** for displaying menus.
- **menuProcessOperatins** for file process operation**.**

```java
package com.file.data;

import java.text.SimpleDateFormat;

public class MenuOptions {

    /**
     * Add the details for application Name in header part.
     */

    public static void displayWelcomeHeaderNote() {

    /**
     * Add the details about developer and date in below part.
     */
    public static void displayWelcomeFooterNote() {

    /**
     * Display the menu options
     */

    public static void displayFileMenuOptions() {

    /**
     *
     * @param folderPath
     * File processing operation
     * Retrieve files
     * Add files,Search file, Delete file
     */

    public static void menuProcessOperations(String folderPath) {

}
```

output:

```
_____________________________________________________________________________

                    Application Name: LockedMe.com

_____________________________________________________________________________


****** Select any number from below for file operation and press Enter ******

1) Retrieve all files
2) Add a file
3) Search a file
4) Delete a file
5) Exit program

Enter an option to proceed
5
Program exited successfully.
_____________________________________________________________________________
                                                        Developed By RABIN

                                                        Date:09-Aug-2021

_____________________________________________________________________________
```

https://github.com/RABINPATI/LockedMeAssessment/blob/main/LockedMe/src/com/fileData/MenuOptions.java

4.4 User input process

- It consists below methods for user input process.

  - **createFiles** for process the data to FileManager class for add operation.
  - **displayFileNames** for process the data to FileManager class for retrieving all files in ascending order.
  - **searchFile** for process the data to FileManager class for searching a file from folder with case sensitive, if exists display success message if file not exist display file not found.
  - **deleteFile** for process the data to FileManager class delete a file from folder with case sensitive, if deleted display success message if file not exist display file not found.

```
 1 package com.file.data;
 2
 3⊕import java.util.ArrayList;
 7
 8 public class FileDataOperation {
 9
10⊖    /**
11      * @param folderPath
12      * Add file
13      */
14
15⊕    public static void createFiles(String folderPath) {
53⊖    /**
54      * @param folderPath
55      * Display Allfiles
56      */
57
58⊕    public static void displayFileNames(String folderPath) {
74⊖   /**
75      * @param folderPath
76      * Search File
77      */
78⊕    public static void searchFile(String folderPath) {
103⊖   /**
104     * @param folderPath
105     * Delete File
106     */
107⊕    public static void deleteFile(String folderPath) {
132
133 }
134
```

https://github.com/RABINPATI/LockedMeAssessment/blob/main/LockedMe/src/com/fileData/FileDataOperation.java

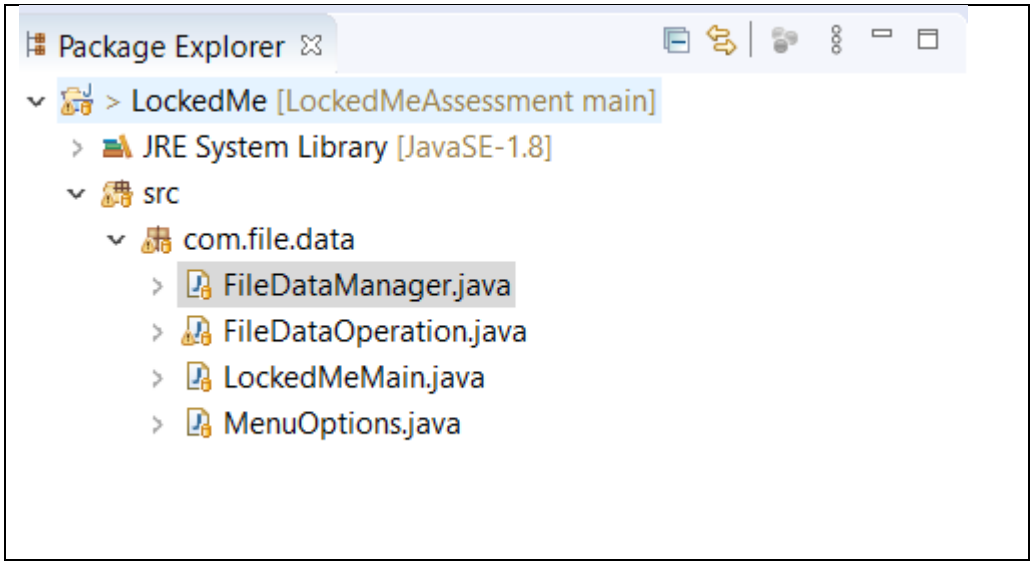
4.5 User input logical operation

- It consists below methods for user input process
  - **createFiles** for add into folder.
  - **displayFileNames** for retrieving all files.
  - **searchFile** for searching a file from folder with case sensitive, if exists display success message if file not exist display file not found.
  - **deleteFile** for delete a file from folder with case sensitive, if deleted display success message if file not exist display file not found.

```
1  package com.file.data;
2
3  import java.io.File;
8
9  public class FileDataManager {
10
11     /**
12      *
13      * @param folderPath
14      * @param fileName
15      * @param dataContent
16      * @return file in directory
17      */
18
19     public static boolean createFiles(String folderPath, String fileName, List<String> dataCont
42
43     /**
44      *
45      * @param folderPath
46      * @return all files
47      */
48     public static List<String> getAllFiles(String folderPath) {
63     /**
64      *
65      * @param folderPath
66      * @param file_Name
67      * @return true/false based on search results.
68      */
69     public static boolean searchFile(String folderPath, String file_Name) {
91     /**
92      *
93      * @param folderPath
94      * @param file_Name
95      * @return true/false based on delete results.
96      */
97
98     public static boolean deleteFile(String folderPath, String file_Name) {
124 }
```

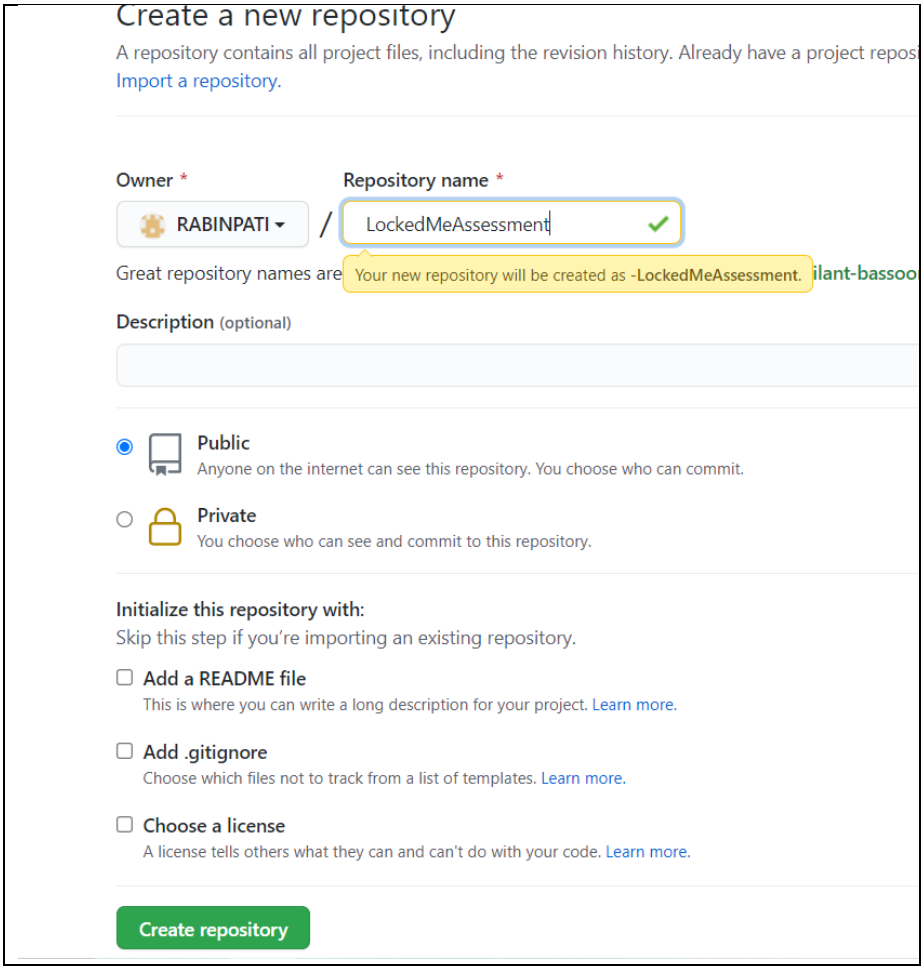
https://github.com/RABINPATI/LockedMeAssessment/blob/main/LockedMe/src/com/fileData/FileDataManager.java

5. Project package structure

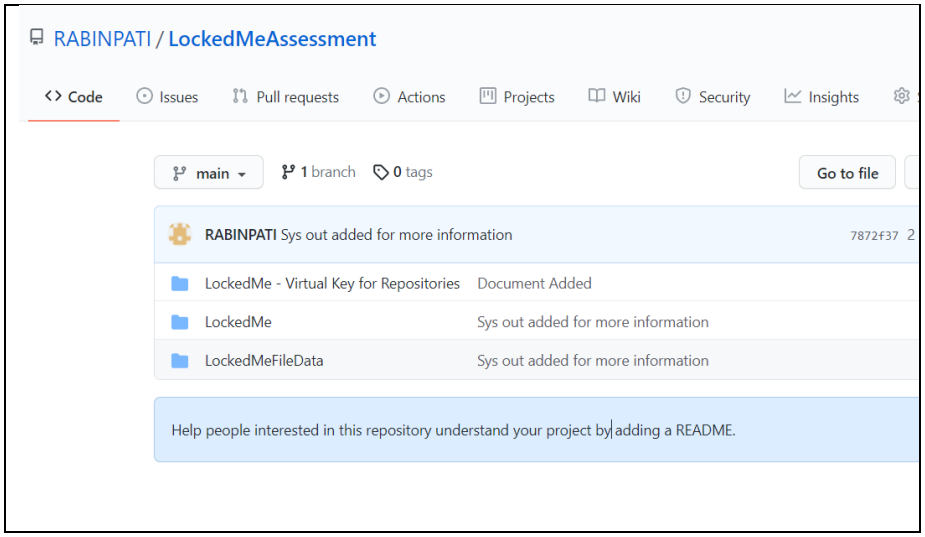


6. GitHub Process for Commit to Repository

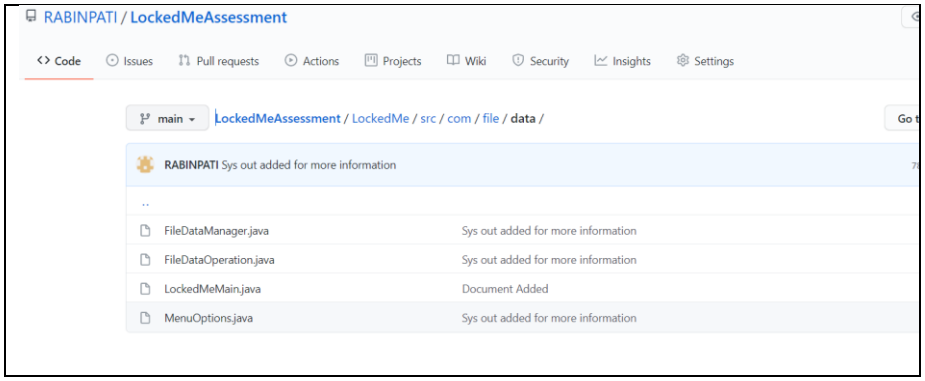
- Create a new repository



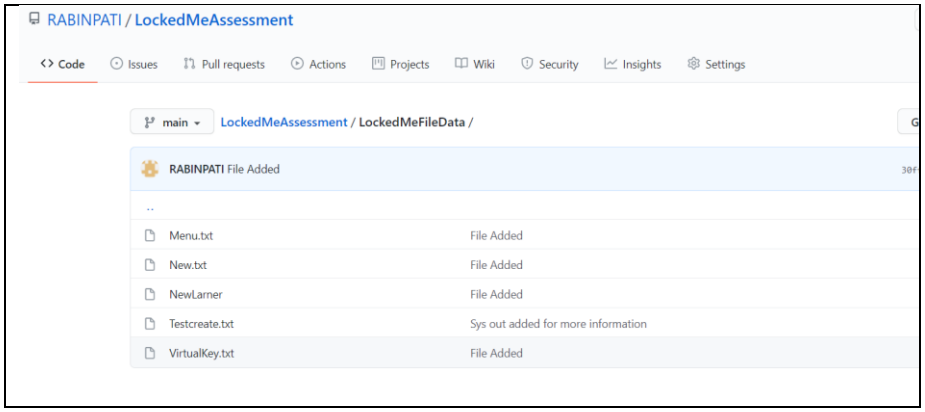
- Code and files committed in GitHub Repository



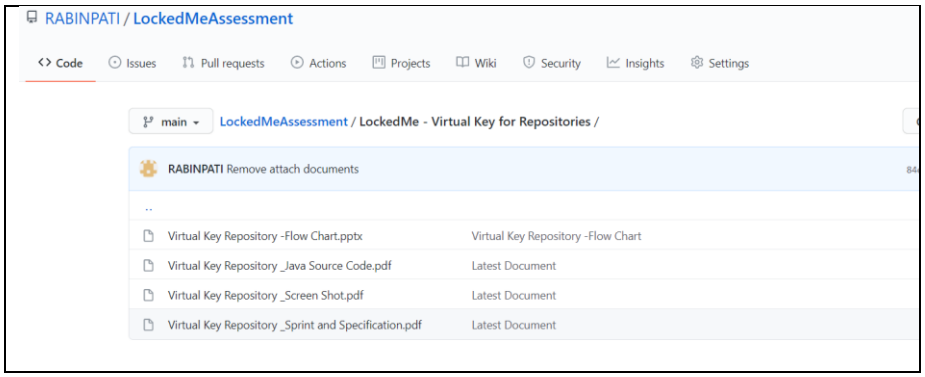
- Java Files in repository



- Text files in repository



- Documents in repository



7. Discussion Points of the Application

This application covers below point

- ✓ Simple display menus for user interaction.
- ✓ Appropriate module to describes operation.
- ✓ Follows java naming standard.
- ✓ Declared appropriate comments for future reference.
- ✓ Standard success/failure message to user.
- ✓ Documented with attached screenshot for reference.

8. Conclusion

It's given me pleasure and more confidence after successfully learn concepts of oops and other core concept of java with the help of Simplelearn trainer.
I done all the local set up in my machine after downloading the required software and practice as well as done one small assessment as prescribed in current doc.

9. Appendix

Referenced below material used for assessment

https://tw2021.lms.simplilearn.com/courses/3850/Implement-OOPS-using-JAVA-with-Data-Structures-and-Beyond/syllabus

https://stackoverflow.com
https://www.javatpoint.com/java-tutorial