Project: LockedMe.com

Developed By: Lockers Pvt Ltd.

Table Of Contents:

User Interactions and Project Capabilities:

"LockedMe.com" project includes all the below listed functionality.

- ## Menu based user interface:

The UI of the project is console based and displays an interactive menu upon execution. The menu displays useful options which provides the user to choose among the capabilities of the program such as file name retrieval, file addition, file deletion, file search, return to main context (if user is in subcontext and wants to choose another option from the previous context), Exit.

Each of the functionalities of the program are elaborated below.

- ➔ File Name Retrieval: When the user selects this option the names of all the files that are present in the "Files" directory of the project folder are listed out in an ascending order.
- ➔ File Addition: This option allows the user to add a file to the "Files" directory of the project.
- ➔ File Deletion: This option allows the user to delete a pre-existing file from the "Files" directory.
- ➔ File Search: This option notifies the user on whether or not the specified file exists in the "Files" directory.
- ➔ Return to main: This option displays the main context options to the user again and waits for him to enter his choice.
- ➔ Exit: This option exits the program.

- ## Continuous contextual execution:

The program runs mainly on the contextual looping system where the context change is based on the user choice**.**

The execution contexts used in this program are as listed below:

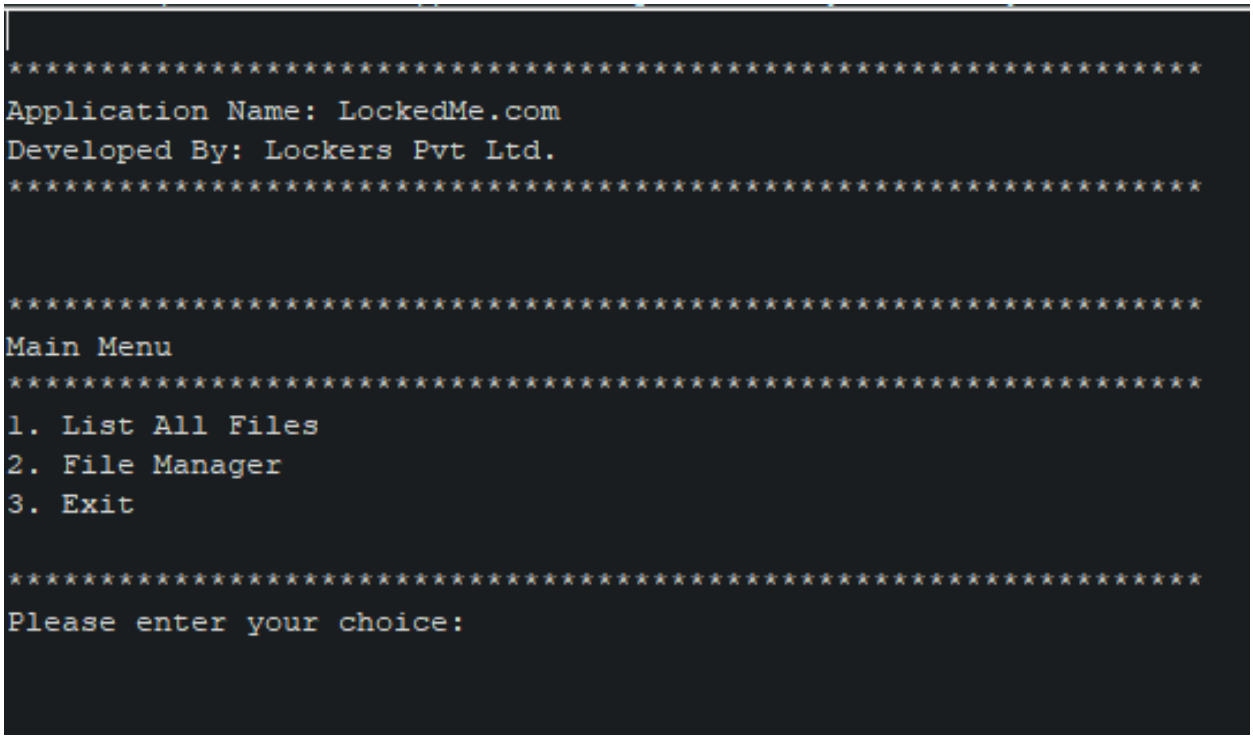
- ➔ MAIN_MENU context.
- ➔ FILE_MANAGER context
- ➔ LIST_ALL_FILES context.
- ➔ ADD_FILE context.
- ➔ DELETE_FILE context.
- ➔ SEARCH_FILE context.

The execution of the program is highly dependent on which context is currently active. It is the active context that decides the options provided to the user and the actions he/she is allowed to perform.

Appearance:

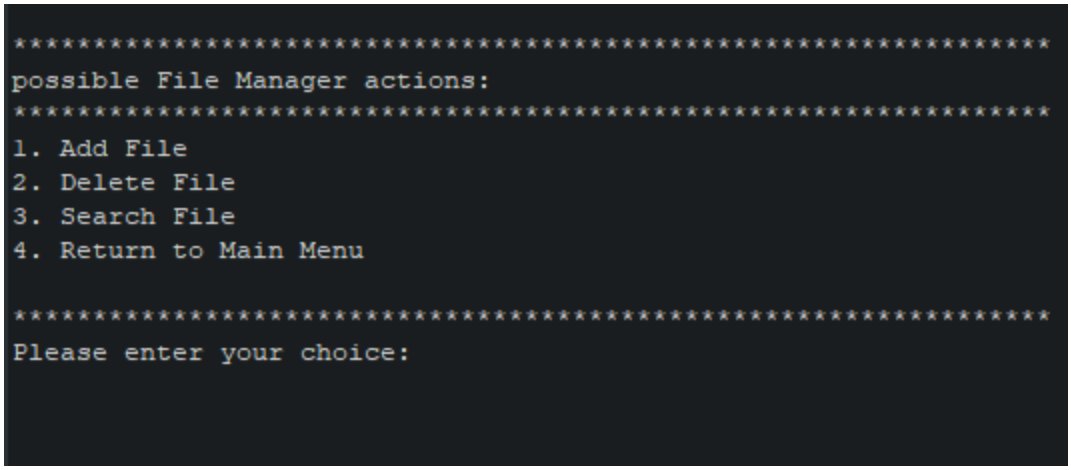
Since this is a prototype of the application, all the user interaction and navigation is purely console based. Hence all the user inputs and program outputs are logged on the console.

Sample screenshots of the execution are attached below:



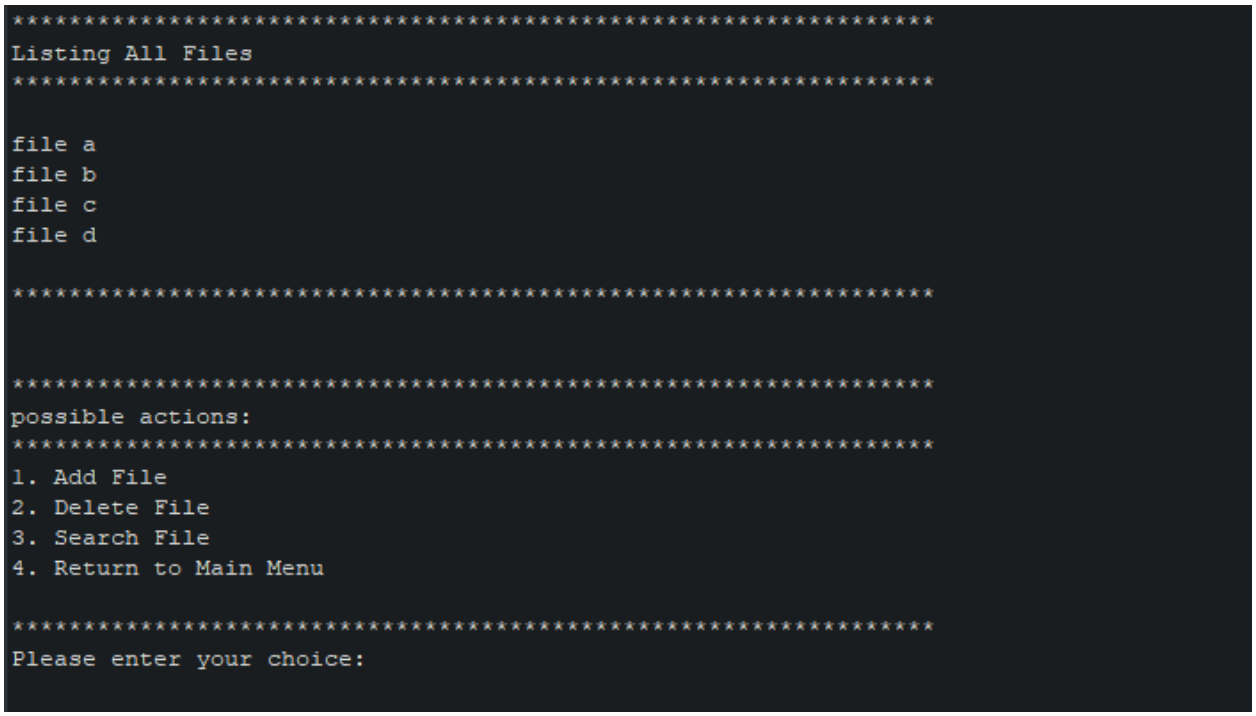
Main Menu context:

This is the Main Menu context of the program where the user is shown the developer details, application name and also provided with various actions to choose from.



File Manager context:

This is the File Manager context which provides the actual file managing capabilities to the user.



List All Files Context:

This is the List all files context where the user can list all the files in the FILES directory of the project and also choose to perform an action from the displayed context options.

```
**************************************************************************
Adding files to the directory.

Existing files:
**************************************************************************

file a
file b
file c
file d

**************************************************************************

Please enter a file name:
file e

**************************************************************************
file e added successfully.
Updated files:
**************************************************************************

file a
file b
file c
file d
file e

**************************************************************************


**************************************************************************
possible actions:
**************************************************************************
1. List All Files
2. Add File
3. Delete File
4. Search File
5. Return to Main Menu

**************************************************************************
Please enter your choice:
```

Add File context:

This is the Add File context where the user can see if the specified file is added successfully and the updated list of files in the directory.

```
**********************************************************************
Deleting files from the directory.

Existing files:
**********************************************************************

file a
file b
file c
file d
file e

**********************************************************************

Please enter a file name:
file e

**********************************************************************
file e deleted successfully.
Updated files:
**********************************************************************

file a
file b
file c
file d

**********************************************************************


**********************************************************************
possible actions:
**********************************************************************
1. List All Files
2. Add File
3. Delete File
4. Search File
5. Return to Main Menu

**********************************************************************
Please enter your choice:
```

Delete File context:

This is the delete file context where the user can see if the specified file has been deleted successfully and can choose to perform an action from the context options.

```
**********************************************************************
possible actions:
**********************************************************************
1. List All Files
2. Add File
3. Delete File
4. Search File
5. Return to Main Menu

**********************************************************************
Please enter your choice:
4

**********************************************************************

Please enter a file name:
file b
file b exists at: C:\Users\rohit\git\LockedMe.com\LockedMe.com\Files\file b

**********************************************************************
possible actions:
**********************************************************************
1. List All Files
2. Add File
3. Delete File
4. Search File
5. Return to Main Menu

**********************************************************************
Please enter your choice:
```

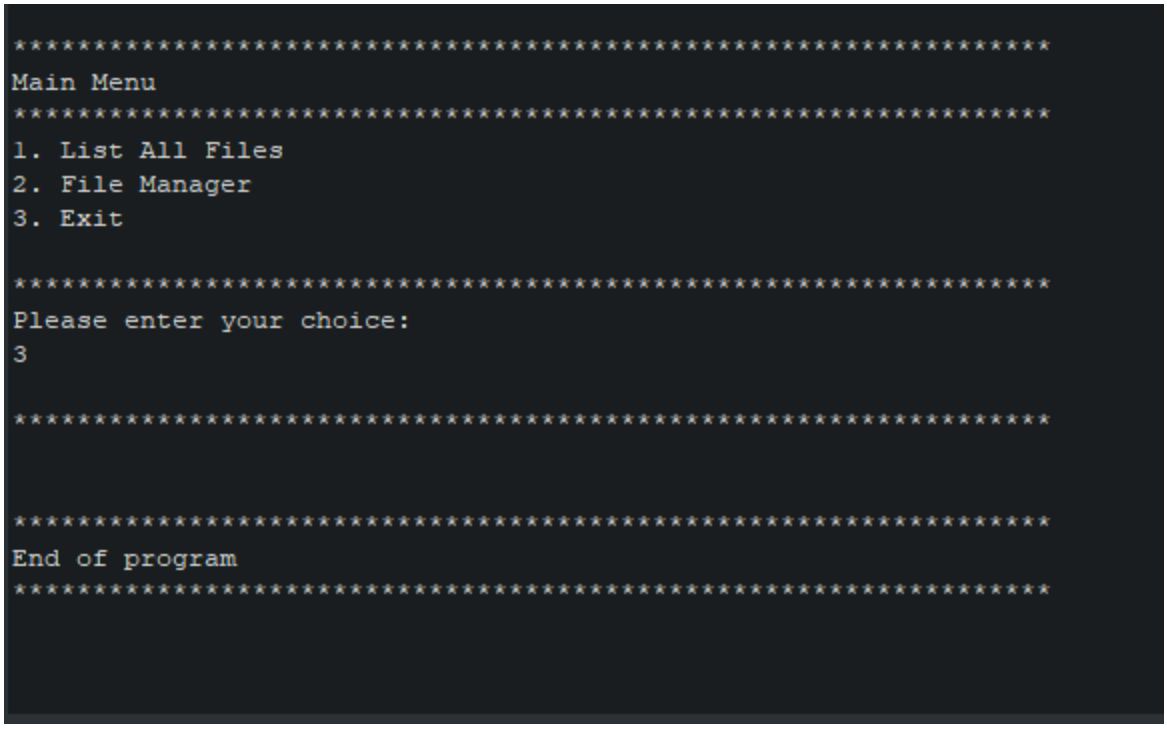
Search File Context:

This is the search file context where the user can see if the specified exists in the directory and also perform certain actions based on the context options.

```
**************************************************************************
possible actions:
**************************************************************************
1. List All Files
2. Add File
3. Delete File
4. Search File
5. Return to Main Menu

**************************************************************************
Please enter your choice:
5

**************************************************************************


**************************************************************************
Returning to Main Menu
**************************************************************************


**************************************************************************
Application Name: LockedMe.com
Developed By: Lockers Pvt Ltd.
**************************************************************************


**************************************************************************
Main Menu
**************************************************************************
1. List All Files
2. File Manager
3. Exit

**************************************************************************
Please enter your choice:
```

Return To Menu option:

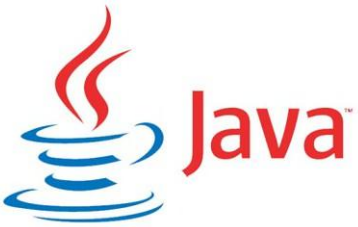
This is the return to menu option which is provided to the user in the sub contexts to be able to return to the main menu context.

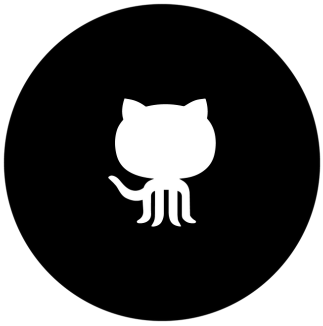


Exit Option:

Finally this is the exit option provided to the user on all the contexts to exit the program from any active context.

Software used:





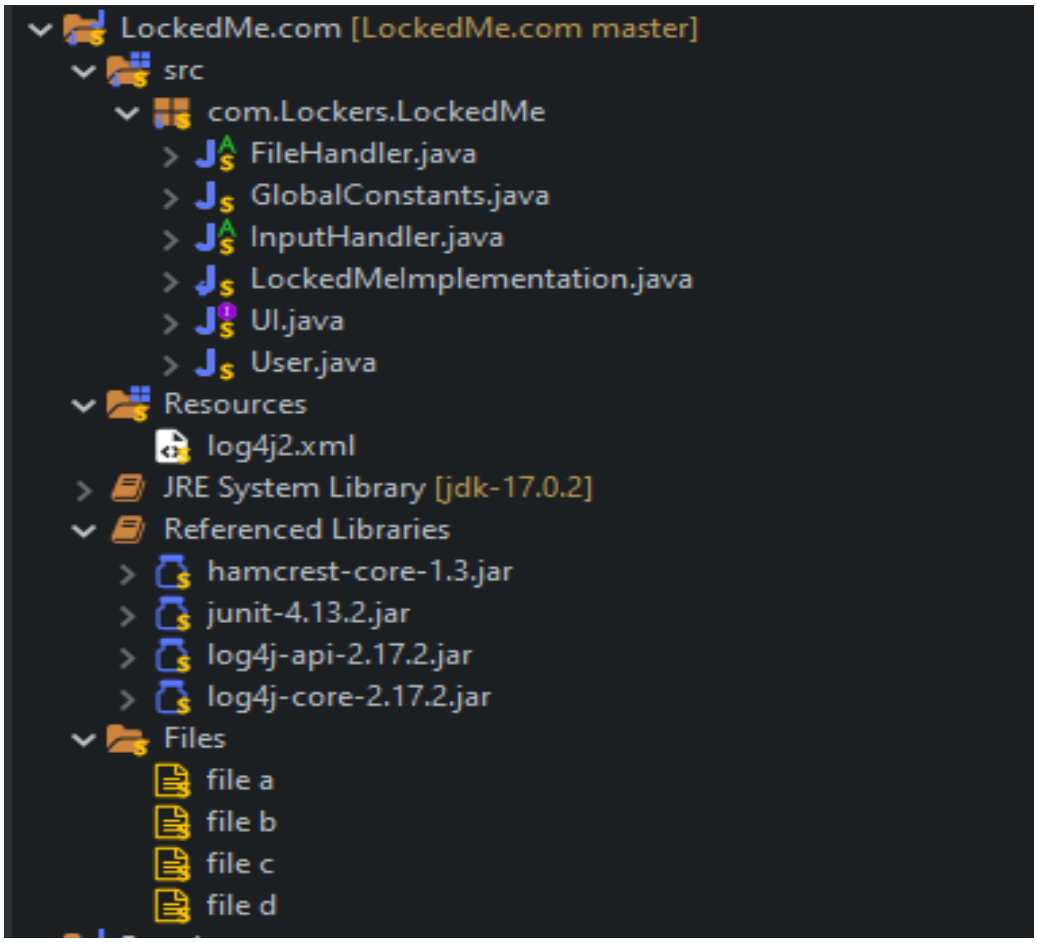
Github

- ➔ Log4j
- ➔ Junit (Included but not used in the prototype)
- ➔ Agile Scrum Methodology

Java concepts used in the project:

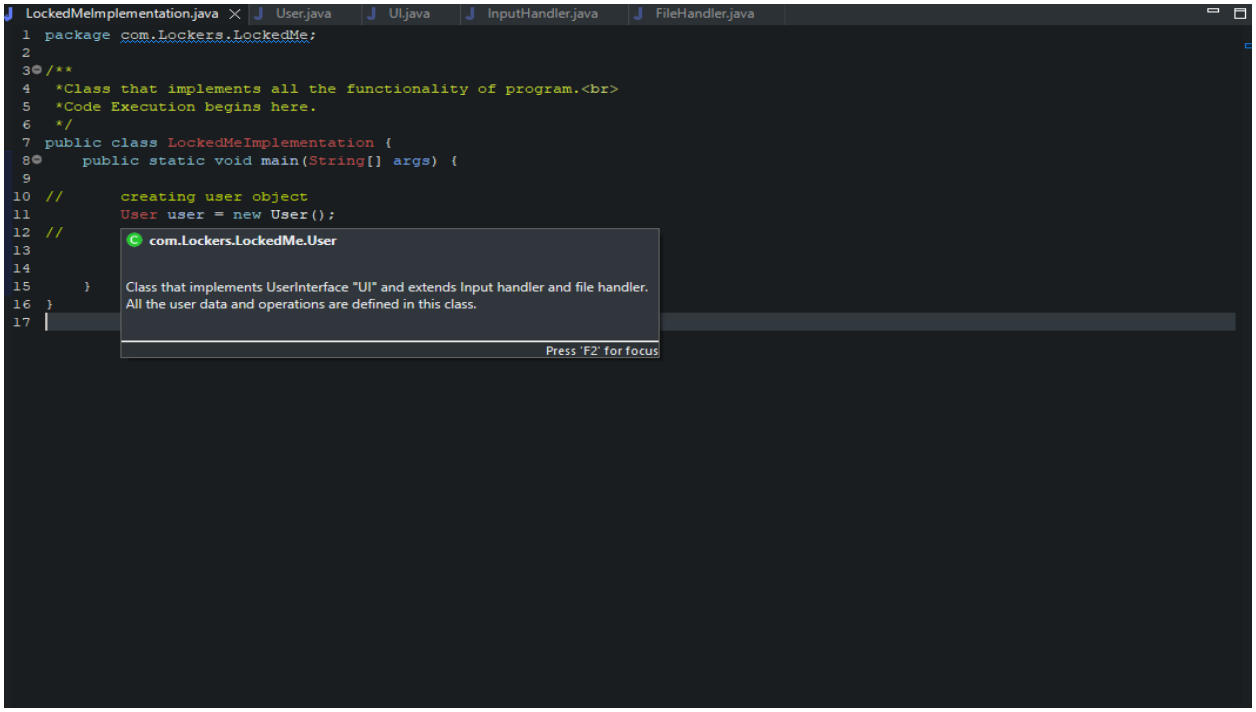
- ➔ Most of the oops concepts have been used in this project.
- ➔ Proper coding conventions and rules have been followed.
- ➔ Proper documentation for every custom class and method has been provided in javadoc.
- ➔ The core concepts used in the project are listed below with attached code snapshots from the eclipse ide.

Code snapshots and eclipse screen captures:



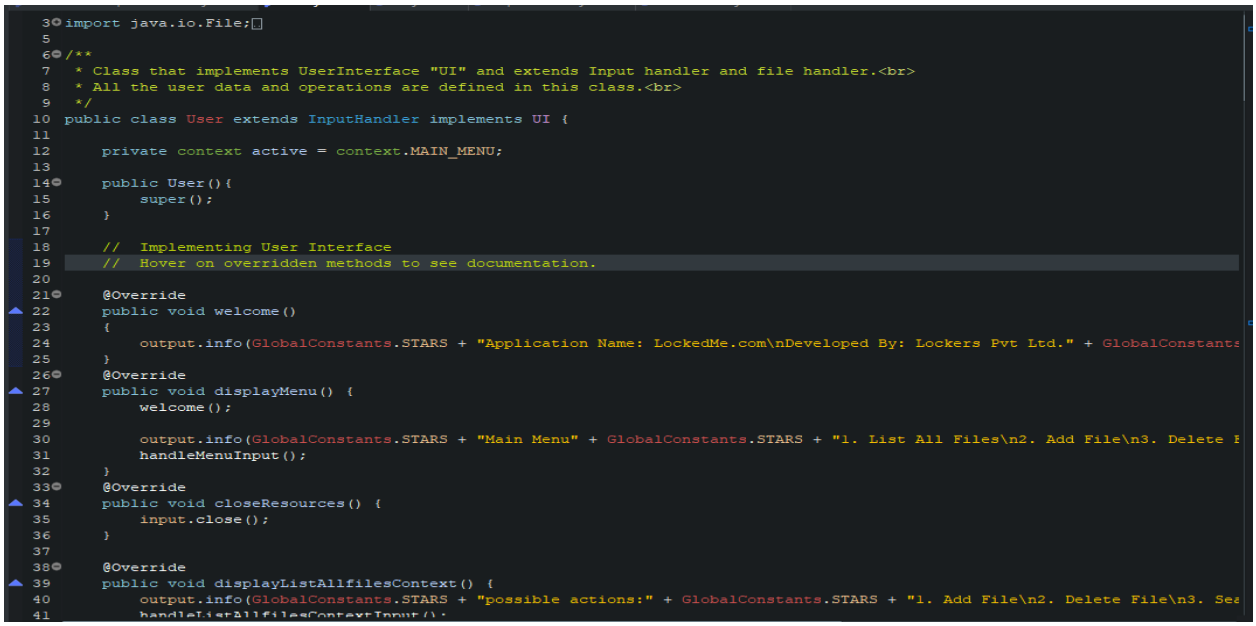
Naming convention and folder structure:

Proper naming convention and folder structure has been followed while developing the code.



Documentation comments:

Documentation for user defined methods and classes has been provided with javadoc.



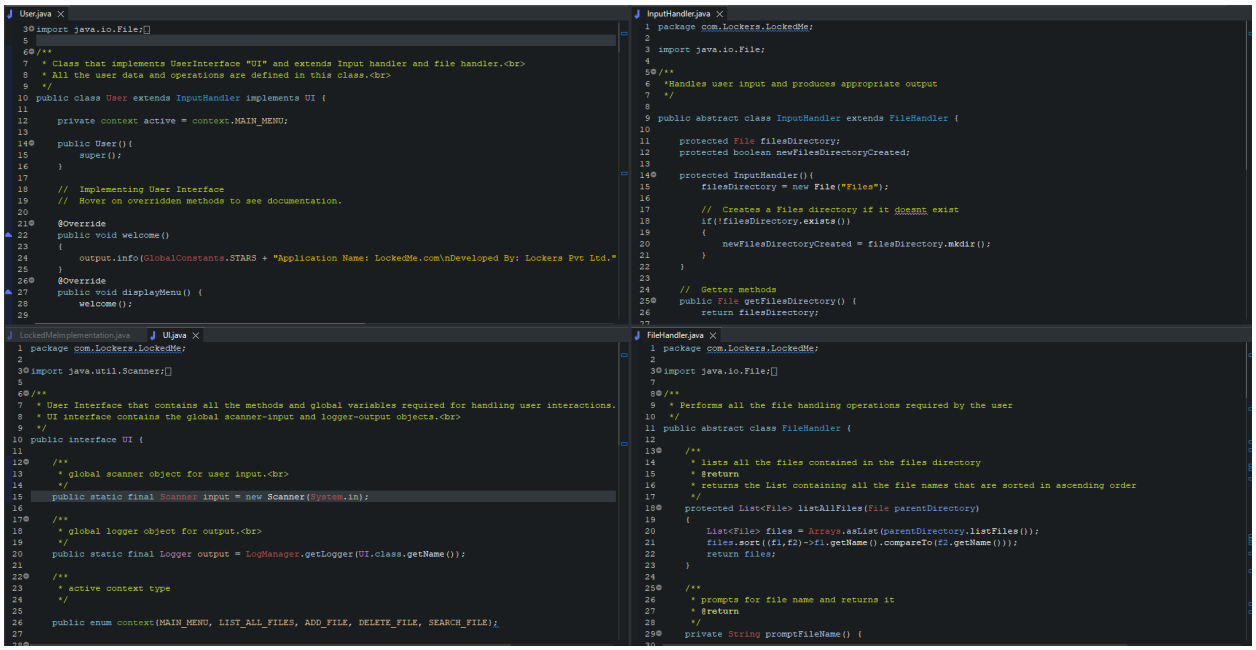
Improved readability:

Proper commenting and annotations have been used to ensure a code that is bug and error free.

```
public class User extends InputHandler implements UI {
```

Core Java and OOPS concepts:

Java OOPS features like abstraction, encapsulation, inheritance and interfaces have been used.



Modularity and reusability:

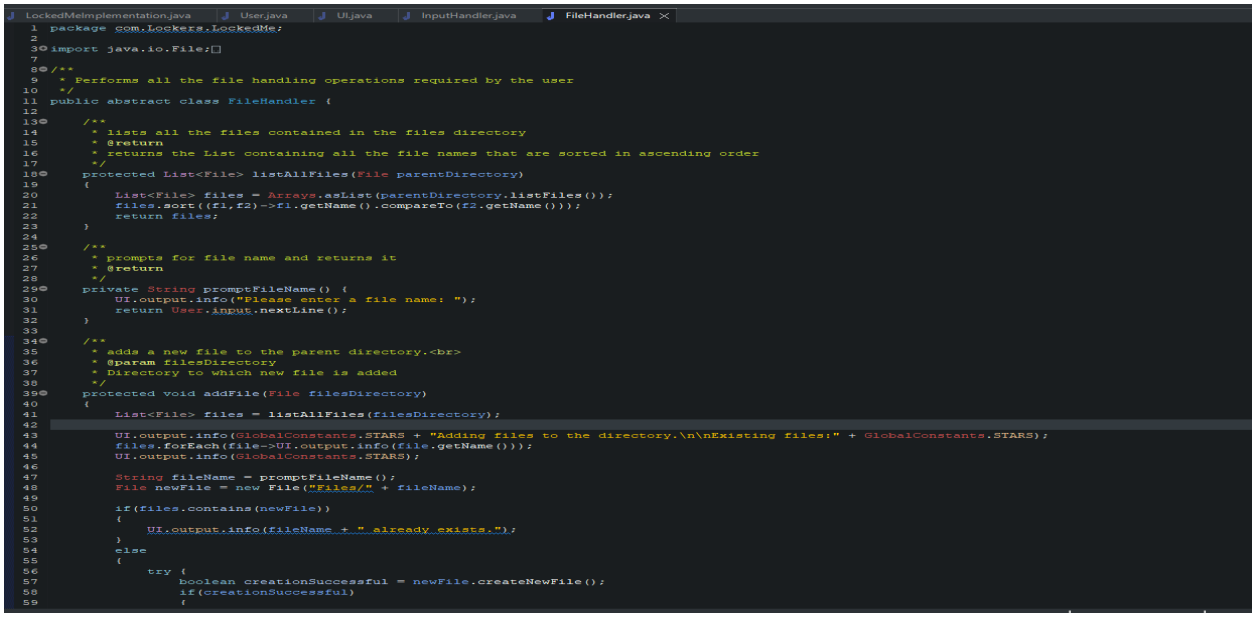
Code has been divided into proper reusable blocks and proper security has been provided by using the concepts of abstraction and access modification.

LockedMeImplementation.java | User.java | UI.java | InputHandler.java | FileHandler.java

```java
package com.Lockers.LockedMe;

import java.io.File;

/**
 *Handles user input and produces appropriate output
 */

public abstract class InputHandler extends FileHandler {

    protected File filesDirectory;
    protected boolean newFilesDirectoryCreated;

    protected InputHandler(){
        filesDirectory = new File("Files");

        //  Creates a Files directory if it doesnt exist
        if(!filesDirectory.exists())
        {
            newFilesDirectoryCreated = filesDirectory.mkdir();
        }
    }

    //  Getter methods
    public File getFilesDirectory() {
        return filesDirectory;
    }

    public boolean isNewFilesDirectoryCreated() {
        return newFilesDirectoryCreated;
    }


    /**
     * prompts for and handles the user input for main context.
     */
    protected abstract void handleMenuInput();

    /**
     * prompts for and handles the user input for list all files context.
     */
    protected abstract void handleListAllfilesContextInput();

    /**
     * prompts for and handles the user input for add file context.
     */
    protected abstract void handleSubContext();



}
```

Input handling:

Input handling has been done separately and proper implementation has been provided by the user class.



File Handling:

File Handling has been done separately to avoid any conflicts and keep the code clear, which also promotes reusability.

Exception Handling:

All the exceptions have been handled properly so that the code won't stop running abruptly.

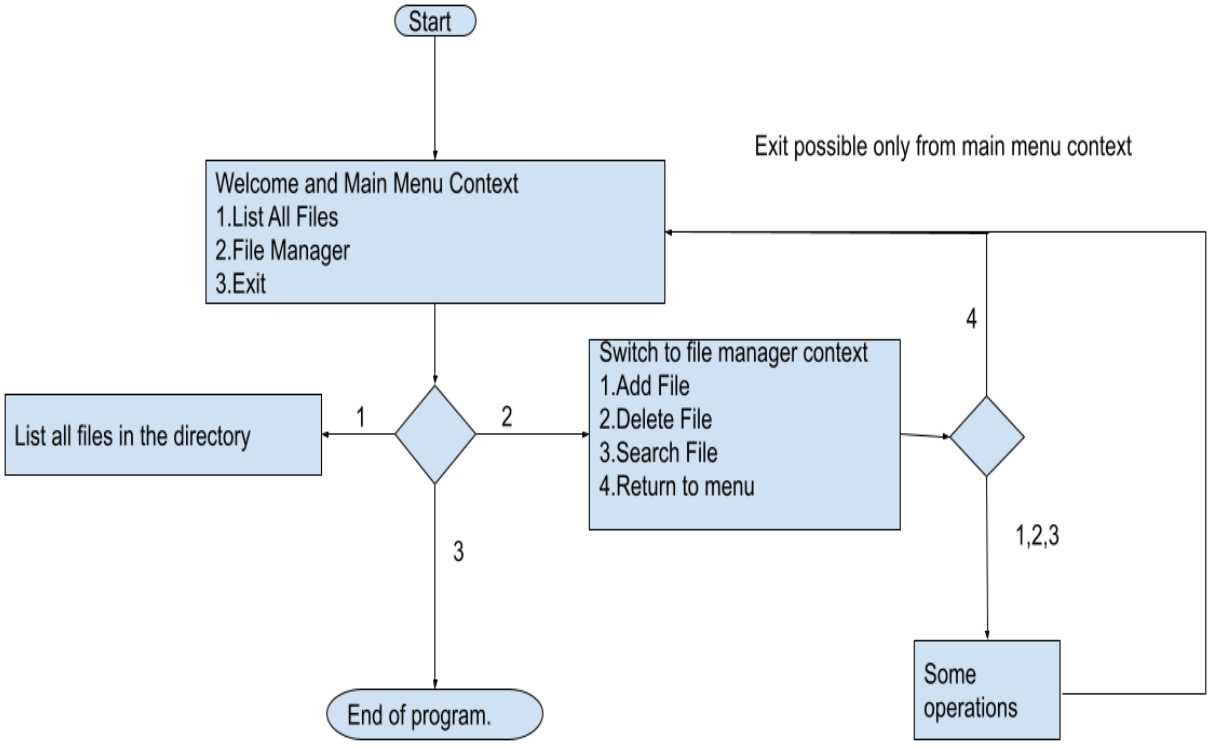
LockedMeImplementation.java | User.java | UI.java | InputHandler.java | FileHandler.java | log4j2.xml

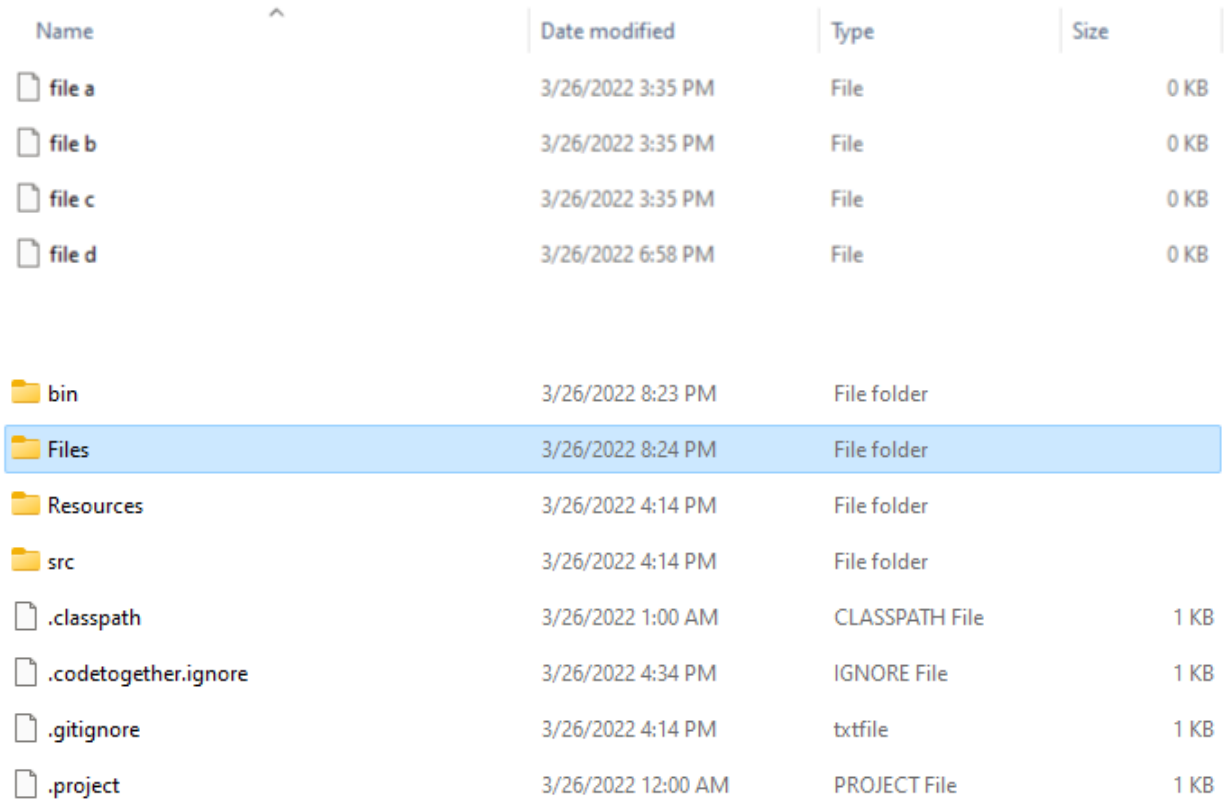
```
<?xml version="1.1" encoding="UTF-8"?>
<!--Logging only onto the console for this project-->
<Configuration status = "WARN">
    <Appenders>
        <Console name = "ConsoleAppender" target = "SYSTEM_OUT">
        <!--Making it so that only the message gets printed along with the new line-->
            <PatternLayout pattern = "%msg%n"/>
        </Console>
    </Appenders>
    <Loggers>
        <Root level = "info">
            <AppenderRef ref = "ConsoleAppender"/>
        </Root>
    </Loggers>
</Configuration>
```

Logging:

Logging has been done to the console using the Log4j2 logger which has been configured to log only the message on the console logger and the log level has been set to info.

Algorithm Flow chart depicting flow of the program:





Name	Date modified	Type	Size
file a	3/26/2022 3:35 PM	File	0 KB
file b	3/26/2022 3:35 PM	File	0 KB
file c	3/26/2022 3:35 PM	File	0 KB
file d	3/26/2022 6:58 PM	File	0 KB
bin	3/26/2022 8:23 PM	File folder	
Files	3/26/2022 8:24 PM	File folder	
Resources	3/26/2022 4:14 PM	File folder	
src	3/26/2022 4:14 PM	File folder	
.classpath	3/26/2022 1:00 AM	CLASSPATH File	1 KB
.codetogether.ignore	3/26/2022 4:34 PM	IGNORE File	1 KB
.gitignore	3/26/2022 4:14 PM	txtfile	1 KB
.project	3/26/2022 12:00 AM	PROJECT File	1 KB

Output File creation:

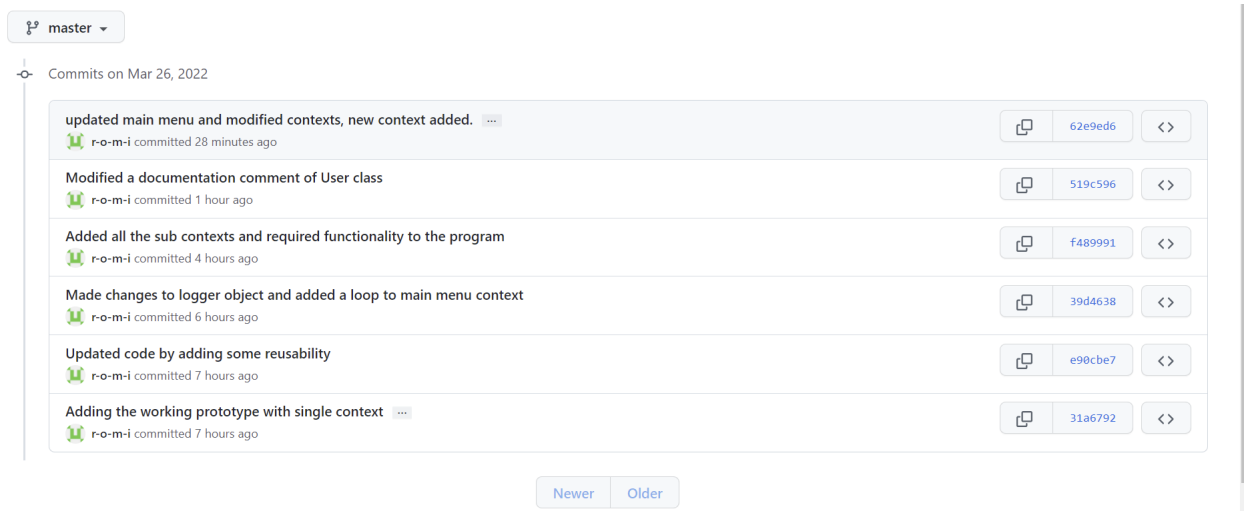
Files created by the program are in the directory “Files” located in the project folder.

Agile Scrum Methodology:

The project was time boxed to be of 2 Sprint length , but due to overtime working and better availability of resources it was completed within one sprint.

Git and Github:

```
rohit@DESKTOP-R3FCOHI MINGW64 ~/git/LockedMe.com (master)
$ git log --oneline --decorate --graph
* 62e9ed6 (HEAD -> master, origin/master) updated main menu and modified contexts, new context added.
* 519c596 Modified a documentation comment of User class
* f489991 Added all the sub contexts and required functionality to the program
* 39d4638 Made changes to logger object and added a loop to main menu context
* e90cbe7 Updated code by adding some reusability
* 31a6792 Adding the working prototype with single context
```



The project is up to date and all the versions have been committed and tracked on both local and remote repositories.

Github Repository Link:

https://github.com/r-o-m-i/LockedMe.com.git

Conclusion and USPs:

To conclude the program LockedMe.com enables user to manage files in the Files directory of the project folder as it is a prototype, USPs of the project is that apart from the required functionality it also provides the user with options to manage files in almost all the contexts except for the main menu context.