

LockedMe – Documentation

This document contains sections for:

- [Sprint planning and Task completion](#)
- [Core concepts used in project](#)
- [Flow of the Application.](#)
- [Demonstrating the product capabilities, appearance, and user interactions.](#)
- [Unique Selling Points of the Application](#)
- [Conclusions](#)

The code for this project is hosted at <https://github.com/sharafRiz/project.git>

The project is developed by SYED SHARAF ALI.

Sprints planning and Task completion

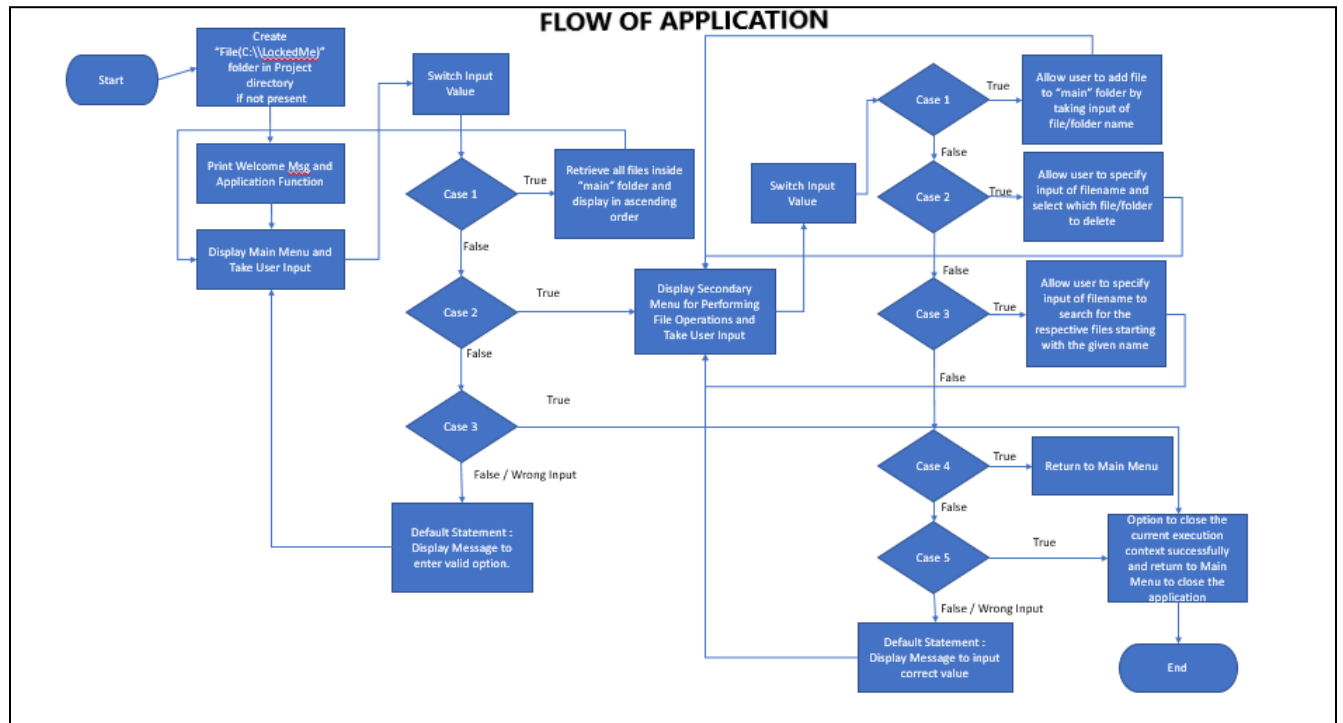
The project is planned to be completed in 1 sprint. Tasks assumed to be completed in the sprint are:

- Creating the flow of the application
- Initializing git repository to track changes as development progresses.
- Writing the Java program to fulfill the requirements of the project.
- Testing the Java program with different kinds of User input
- Pushing code to GitHub.
- Creating this specification document highlighting application capabilities, appearance, and user interactions.

Core concepts used in project

Collections framework, File Handling, Sorting, Flow Control, Recursion, Exception Handling, Streams API

Flow of the Application



Flow of
Application.pptx

Demonstrating the product capabilities, appearance, and user interactions

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

- 1 [Creating the project in Eclipse](#)
- 2 [Writing a program in Java to perform the File operations as specified by user \(LockerMe.java\)](#)
- 3 [Pushing the code to GitHub repository](#)

Step 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 **LockerMe** in any class name, check the checkbox "public static void main(String[] args)", and click on "Finish."

Step 2: Writing a program in Java to display Menu options available for the user (**MenuOptions.java**)

3.1. [Displaying Welcome Screen](#)

3.2. [Displaying Initial Menu](#)

3.3. [Displaying Secondary Menu for File Operations available](#)

Step 3.1: Writing method to display Welcome Screen

```
{
static final String welcomeMsg = ("\n ** Welcome to LockedMe.com **"
+ "\n\n=====Developer Details===== " +
"\nName-Sharaf Ali"
+ "\nProject Name: LOkkedMe.com");

static final String mainMenu = "\nMAIN MENU - \nPlease select any of the
following:" +
"\n 1. Display file names in ascending order from root directory"
+
"\n 2. More Business operations" +
"\n 3. Exit Program";

}
```

Output:

```

|  ** Welcome to LockedMe.com **
|
|  =====Developer Details=====
|  Name-Sharaf Ali
|  Project Name: LOkkedMe.com
|
|  MAIN MENU -
|  Please select any of the following:
|  1. Display file names in ascending order from root directory
|  2. More Business operations
|  3. Exit Program

```

Step 3.2: Writing method to display Main Menu

```

static final String mainMenu = "\nMAIN MENU - \nPlease select any of the following:"
+
+       "\n 1. Display file names in ascending order from root directory"
+
+       "\n 2. More Business operations" +
+       "\n 3. Exit Program";
}

```

Output:

```

|  ***** Select any option number from below and press Enter *****
|
|  1) Retrieve all files inside "main" folder
|  2) Display menu for File operations
|  3) Exit program

```

Step 3.3: Writing method to display Secondary Menu for File Operations

```

static final String secondaryMenu = "\n Select any of the following" +
+       "\n a. Add file" +
+       "\n b. Delete file" +
+       "\n c. Search file" +
+       "\n d. Go back to Main Menu";
+       System.out.println(fileMenu);
}

```

Output:

```
Select any of the following
a. Add file
b. Delete file
c. Search file
d. Go back to Main Menu
```

Step 4: Writing a program in Java to Execute Menu options selected by user

Step 4: Writing method to handle user input in Secondary Menu for File Operations

```
void showSecondaryMenu() {
    System.out.println(secondaryMenu);
    try {

        Scanner scanner = new Scanner(System.in);
        char[] input =
scanner.nextLine().toLowerCase().trim().toArray();
        char option = input[0];

        switch (option) {
            case 'a': {
                System.out.println("Enter file name: ");
                String filename = scanner.nextLine();
                addFile(filename);
            }
            case 'b': {
                System.out.println("Enter file name to delete: ");
                String filename = scanner.nextLine();
                deleteFile(filename);
            }
            case 'c': {
                System.out.println("Enter file name to Search: ");
                String filename = scanner.nextLine();
                searchFile(filename);
            }
            case 'd': {
                System.out.println("Going back to main menu...");
                showMainMenu();
                break;
            }
            default:
                System.out.println("Please Enter a, b, c or d");
                showSecondaryMenu();
        }
    } catch (Exception e) {
```

```

        System.out.println(e.toString());
        System.out.println("Please Enter a, b, c or d");
        showSecondaryMenu();
    }
}

```

Output:

```

Please select any of the following:
1. Display file names in ascending order from root directory
2. More Business operations
3. Exit Program
2

Select any of the following
a. Add file
b. Delete file
c. Search file
d. Go back to Main Menu
a
Enter file name:
txt
File Added txt

Select any of the following
a. Add file
b. Delete file
c. Search file
d. Go back to Main Menu
c
Enter file name to Search:
txt
File found in C:\LockedMe
Going back to main menu...

MAIN MENU -
Please select any of the following:
1. Display file names in ascending order from root directory
2. More Business operations
3. Exit Program

```

Step 5: Writing a program in Java to perform the Business Level operations as specified by user (**LockerMe.java**)

- Select your project and go to File -> New -> Class.
- Enter **LockerMe** in class name and click on "Finish."

- consists methods for -:

- 5.1. [Creating "new LockedMe" in directory if it's not already present](#)
- 5.2. [Displaying all files in "LockedMe" folder in ascending order.](#)
- 5.3. [Search files as specified by user input in "main" folder and it's subfolders.](#)
- 5.4. [Deleting a file/folder from "main" folder](#)

Step 5.1: Writing method to create new LockedMe folder in directory if it's not present

```
public void LockerMe() {
    directory = new File("C:\\LockedMe\\");
    if (!directory.exists())
        directory.mkdir();
```

Output:



Step 5.2: Writing method to display all files in "LockedMe" folder in ascending.

```
public void listFile() {
    LockerMe();

    if (directory.list().length == 0) {
        System.out.println("Folder is Empty");
    } else {
        String[] list = directory.list();
        System.out.println("The file in " + directory + " are:");
        Arrays.sort(list);
        for (String str : list) {
            System.out.println(str);
        }
    }
}
```

Output:

```
MAIN MENU -
Please select any of the following:
1. Display file names in ascending order from root directory
2. More Business operations
3. Exit Program
1
The file in C:\LockedMe are:
abc
bab.txt
cdc.txt
dvd.txt
sharaf
txt
LockedMeMain.java test_file.txt MenuOptions.java FileOperations.java HandleOptions.java
1 Checking if file content written in specified file.
```

Step 5.4: Writing method to search for files as specified by user input in “LockedMe” folder

```
public void searchFile(String filename) {
    LockerMe();
    File filepath = new File("C:\\LockedMe\\" + filename);
    String[] list = directory.list();
    for (String file : list) {
        if (filename.equals(file)) {
            System.out.println("File found in " + directory);
            return;
        }
    }
    System.out.println("FILE NOT FOUND");
    showSecondaryMenu();
}
```

Output:


```

** Welcome to LockedMe.com **

=====Developer Details=====
Name-Sharaf Ali
Project Name: LLockedMe.com

MAIN MENU -
Please select any of the following:
1. Display file names in ascending order from root directory
2. More Business operations
3. Exit Program
2

Select any of the following
a. Add file
b. Delete file
c. Search file
d. Go back to Main Menu
c
Enter file name to Search:
dvd.txt
File found in C:\LockedMe
Going back to main menu...

```

Step 5.5: Writing method to delete file/folder specified by user input in “LockedMe” folder.

```

public void deleteFile(String filename) {

    LockerMe();

    File filepath = new File("C:\\LockedMe\\" + filename);
    String[] list = directory.list();
    for (String file : list) {
        if (filename.equals(file) && filepath.delete()) {
            System.out.println("File " + filename + " deleted from " +
directory);
            showMainMenu();
            return;
        }
    }
}

```

Output:

```

** Welcome to LockedMe.com **

=====Developer Details=====
Name-Sharaf Ali
Project Name: LOkkedMe.com

MAIN MENU -
Please select any of the following:
  1. Display file names in ascending order from root directory
  2. More Business operations
  3. Exit Program
2

Select any of the following
a. Add file
b. Delete file
c. Search file
d. Go back to Main Menu
b
Enter file name to delete:
dvd.txt
File dvd.txt deleted from C:\LockedMe

MAIN MENU -
Please select any of the following:
  1. Display file names in ascending order from root directory
  2. More Business operations
  3. Exit Program

```

This PC > Local Disk (C:) > LockedMe

^	□ Name	^	Date modified	Type	Size
	abc		6/25/2022 3:06 PM	File	0 KB
	bab		6/25/2022 3:06 PM	Text Document	0 KB
	cdc		6/25/2022 3:06 PM	Text Document	0 KB
	sharaf		6/24/2022 11:12 PM	File	0 KB
	txt		6/25/2022 2:47 PM	File	0 KB

Step 6: Pushing the code to GitHub repository

- Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

- Initialize repository using the following command:

git init

- Add all the files to your git repository using the following command:

git add .

- Commit the changes using the following command:

git commit . -m <commit message>

- Push the files to the folder you initially created using the following command:

git push -u origin master

Unique Selling Points of the Application

1. The application is designed to keep on running and taking user inputs even after exceptions occur. To terminate the application, appropriate option needs to be selected.
2. The application can take any file/folder name as input.
3. User is also provided the option to write content if they want into the newly created file.
4. The application also allows user to delete folders which are not empty.
5. The user is able to seamlessly switch between options or return to previous menu even after any required operation like adding, searching, deleting or retrieving of files is performed.
6. When the option to retrieve files in ascending order is selected, user is displayed with option of viewing the files.

6.1. Ascending order of all files and folders inside the "LockedMe" folder.

7. The application is designed with modularity in mind. Even if one wants to update the path, they can change it through the source code. Application has been developed keeping in mind that there should be very less "hardcoding" of data.

Conclusions

Further enhancements to the application can be made which may include:

- Conditions to check if user is allowed to delete the file or add the file at the specific locations.
- Asking user to verify if they really want to delete the selected directory if it's not empty.
- Retrieving files/folders by different criteria like Last Modified, Type, etc.
- Allowing user to append data to the file.