**LOCKED ME**

**VIRTUAL KEY FOR YOUR REPOSITORIES**

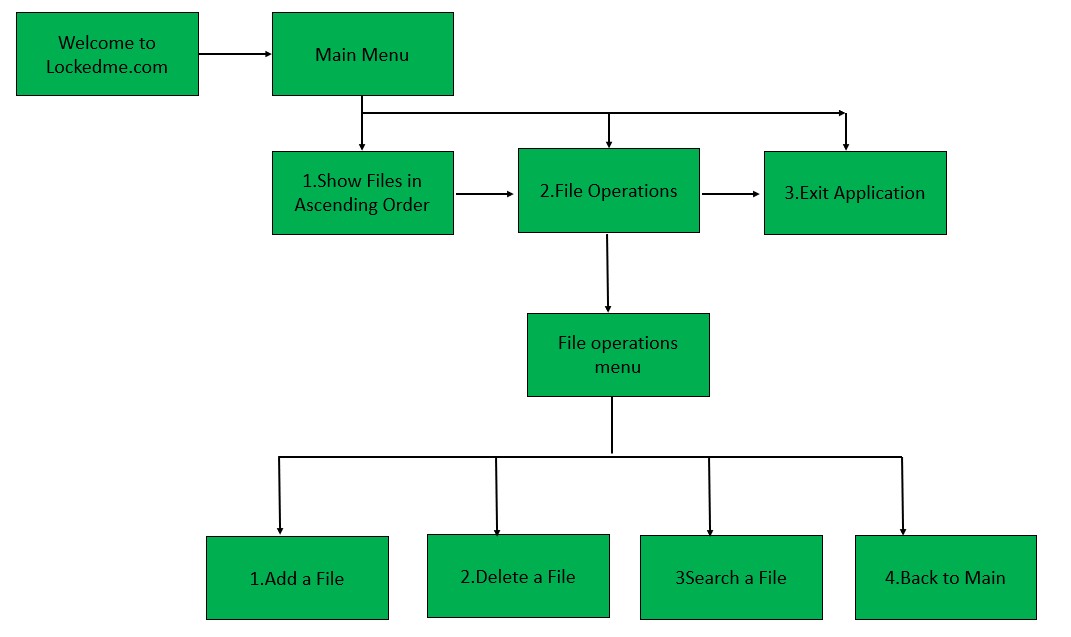
**Phase 1 assessment project**

## 

## **Core concepts used in project:**

Collections framework, File Handling, Sorting, Flow Control, Recursion, Exception Handling.

## **Flow of the Application:**



**Generic features and operations available:**

1) Retrieving the file names in an ascending order

2) Business-level operations:

* Option to add a user specified file to the application
* Option to delete a user specified file from the application
* Option to search a user specified file from the application
* Navigation option to close the current execution context and return to the main context

3) Option to close the application

## **Demonstrating the product capabilities, appearance, and user interactions:**

## **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 **LockedMe** in any class name, check the checkbox “public static void main (String[] args)”, and click on “Finish.”

## 

## **Step 2: Writing a program in Java for the application (LockedMe.java)**

**2.1: Writing method to listing a file and creating a directory**

package project1;

import java.io.File;

import java.util.Arrays;

import java.util.Collections;

import java.util.List;

import java.util.Scanner;

public class LockedMe {

public static void listingFile() {

File fileDir = new File("C:\\locker");

//Creating the directory

fileDir.mkdir();

if(fileDir.isDirectory()){

List <String>listFile = Arrays.asList(fileDir.list());

Collections.sort(listFile);

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

System.out.println("Sorting by filename in ascending order");

for(String s:listFile){

System.out.println(s);

}

}

else{

System.out.println(fileDir.getAbsolutePath() + " is not a directory");

}LockedMe.mainMenu();

}

**2.2: Writing method to creating a file:**

public static void createFile() {

File file = new File("C:\\locker");

//Creating the directory

file.mkdir();

Scanner sc = new Scanner(System. ***in*** ); //object of Scanner class

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

String name = sc.nextLine(); //variable name to store the file name

File file1 = new File("C:\\locker\\"+name);

try {

if (file1.createNewFile()) {

System.***out***.println("File Created! :)");

LockedMe.*fileMenuOperations*();

} else {

System.***out***.println("File already exists :(");

LockedMe.*fileMenuOperations*();

}

} catch (Exception e) {

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

}

}

}

**2.3: Writing method to deleting a file:**

public static void deleteMethod() {

String filename;

Scanner scan = new Scanner(System.***in***);

System.***out***.println("Enter the Name of File to Delete: ");

filename = scan.nextLine();

File file= new File("C:\\locker\\"+filename);

if (file.delete()) {

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

LockedMe.*fileMenuOperations*();

}

else {

System.***out***.println("Oops !! File not found.Please try again ");

LockedMe.*fileMenuOperations*();

}

}

}

**2.3: Writing method to searching a file:**

public static void searchMethod() {

System.***out***.println("Enter the name of the file to Search");

Scanner s1 = new Scanner(System.***in***);

String folderName = s1.next();

File file = new File("C:\\locker\\"+folderName);

if (file.exists()) {

System.***out***.println("Yep! File found!! :)"+"\n"+"This is the file location");

System.***out***.println(file.getAbsolutePath());

LockedMe.*fileMenuOperations*();

} else {

System.***out***.println("Sorry, File Not Found :(");

LockedMe.*fileMenuOperations*();

}

}

**2.4: Writing method for showing the Main menu and it’s operations**

public static void mainMenu() {

System.***out***.println("Please select any of the below options...?");

System.***out***.println("Enter \"1\" for showing current file names in ascending order ");

System.***out***.println("Enter \"2\" for your Business Operation");

System.***out***.println("Enter \"3\" for exit the application");

}

public static void fileMenuOperations() {

System.***out***.println("Please choose one of the following options :");

System.***out***.println("1. Create a file");

System.***out***.println("2. Delete a File");

System.***out***.println("3. Search for a File");

System.***out***.println("4. for main menu");

Scanner sc2=new Scanner(System.***in***);

try {

int choice=sc2.nextInt();

switch(choice) {

case 1:

System.***out***.println("You have selected to create file");

*createFile*();

break;

case 2:

System.***out***.println("You have selected to Delete file");

*deleteMethod*();

break;

case 3:

System.***out***.println("You have Selected to search file");

*searchMethod*();

break;

case 4:

System.***out***.println("Going back to MainMenu");

LockedMe.*mainMenu*();

break;

}

}catch(Exception e) {

System.***out***.println("Oops!! Please Enter valid number");

LockedMe.*fileMenuOperations*();

}

}

**2.4: Writing method for showing the Welcome Screen and file menu operations**

public static void main(String[] args) {

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

System.***out***.println("Welcome to LOCKEDME :) ");

System.***out***.println("This project is Developed by K S MURALI");

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

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

LockedMe.*mainMenu*();

while(true) {

try {

int option=sc.nextInt();

switch(option) {

case 1 :

System.***out***.println("These are the files in ascending");

*listingFile*();

break;

case 2 :

LockedMe.*fileMenuOperations*();

break;

case 3 :

System.***out***.println("Have a nice day!! Thanks for using locker ");

System.***out***.println("Application closed");

System.*exit*(option);

break;

default:

System.***out***.println("Oops!! please enter the valid number");

break;

}

}catch(Exception e){

System.***out***.println("Oops!! Please Enter valid number");

LockedMe.*main*(args);

}

}

}

}

## **Step 3: 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. Even if the user wants to create nested folder structure, user can specify the relative path, and the application takes care of creating the required folder structure.
3. User is also provided the option to write content if they want into the newly created file.
4. The application doesn’t restrict user to specify the exact filename to search/delete file/folder. They can specify the starting input, and the program searches all files/folder starting with the value and displays it. The user is then provided the option to select all files or to select a specific index to delete.
5. The application also allows user to delete folders which are not empty.
6. 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.
7. When the option to retrieve files in ascending order is selected, user is displayed with two options of viewing the files.
   1. Ascending order of folders first which have files sorted in them,
   2. Ascending order of all files and folders inside the “main” folder.

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.