**Lockedme.com**

**Application specification**

|  |  |
| --- | --- |
| Author | Abdulaziz Alhadlag |
| Purpose | Application specification |
| Date | 10/08/2021 |
| Version | 1 |

Contents

[Modules in the project 3](#_Toc79485547)

[Application Menu 3](#_Toc79485548)

[Display all files 3](#_Toc79485549)

[Add file 3](#_Toc79485550)

[Delete file 3](#_Toc79485551)

[Search file 3](#_Toc79485552)

[Sprints planning 3](#_Toc79485553)

[Project Github link 4](#_Toc79485554)

[Repository name 4](#_Toc79485555)

[Github Link 4](#_Toc79485556)

[Project Code 5](#_Toc79485557)

[Folder structure 5](#_Toc79485558)

[Prototype1 5](#_Toc79485559)

# Modules in the project

## Application Menu

* 1. This Module will show the Landing Page of the Application

## Display all files

* 1. This Module will print out the files in the folder path in ascending order

## Add file

* 1. This module will allow the user to add a file.

## Delete file

* 1. This module will allow the user to delete a file.

## Search file

* 1. This module will allow the user to add a file.

# Sprints planning

|  |  |
| --- | --- |
| **Sprint Number** | **Modules** |
| 1 | Application Menu |
| 2 | Display all Files |
| 3 | Add File  Delete File  Search File  Testing  Deployment |

# Project Github link

|  |
| --- |
| Repository name |
| Java Project |
| Github Link |
| https://github.com/akhadlag/JavaProject |

# Java Technologies used

* Exception Handling
* Working with files
* Naming Standards
* Modularity
* Object Oriented Programming
* Collections
* Control Structures
* Data Structures

# Project Code

|  |
| --- |
| Folder structure |
|  |
| Prototype1 |
| package phase1Project;  import java.util.Scanner;  import java.io.File;  import java.io.FileWriter;  import java.util.ArrayList;  import java.util.Arrays;  import java.util.Collections;  import java.util.List;  public class PrototypeV1 {  public static void main(String[] args) {    // We Start the prototype calling the landing page and from there we take the users input for what he wants to do    landingPage();    }    // The Methods to be used in the prototype    /\*\*  \* this method will show the landing page and ask the user to choose from the landing page  \*/  static void landingPage() {    // Printing the Prototype Landing Page and asking the user for an input    System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println();  System.out.println();  System.out.println();  System.out.println(" lockedme.com");  System.out.println();  System.out.println();  System.out.println(" Please choose from the options below");  System.out.println(" 1-To view the files");  System.out.println(" 2-To Control the files");  System.out.println(" 3-To Close the application");  System.out.println();  System.out.println();  System.out.println(" Devolped by : Lockers Pvt. Ltd.");  System.out.println();  System.out.println();  System.out.println();  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");        // Identifying the variable to be used in choosing the option and reading input from user  Scanner obj = new Scanner(System.in);  int mainch = Integer.parseInt(obj.nextLine());    // switch case to choose from the landing page    switch (mainch) {  // case 1 will print out all of the files in the folderpath  case 1:  viewFiles();  return;  // case 2 will show you the control files menu to choose form  case 2:  controlFiles();  return;  // case 3 will exit the prototype  case 3:  System.exit(0);  return;  // if the input is invalid the system will show the below message  default:  System.out.println("Please Enter a Valid Number");  landingPage();  }      }      /\*\*  \* this method prints out the control files menu and asks the user to choose what item he wants to do  \*/  static void controlFiles () {    // printing out the control files menu and asking the user for an input    System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println();  System.out.println();  System.out.println();  System.out.println(" Lockers Pvt. Ltd.");  System.out.println();  System.out.println();  System.out.println(" Please choose from the options below");  System.out.println(" 1-To Add a New File");  System.out.println(" 2-To Delete an exiciting file");  System.out.println(" 3-To Search for a File");  System.out.println(" 4-to get back to the main menu");  System.out.println();  System.out.println();  System.out.println(" Devolped by : Abdulaziz AlHadlag");  System.out.println();  System.out.println();  System.out.println();  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");    // Reading input from user  Scanner obj = new Scanner(System.in);  int contch = Integer.parseInt(obj.nextLine());    // switch case to control the files    switch (contch) {    // case 1 will let the user add file  case 1 :  // variable decleration  String fn;  int linescount;  List<String> content = new ArrayList<String>();  String folderpath ="C:\\Users\\hadlagak\\Desktop\\Test";    // read file name from the user  System.out.println("Enter the File name ");  fn=obj.nextLine();  //read number of lines from user  System.out.println("Enter how many lines in the file");  linescount=Integer.parseInt(obj.nextLine());  // read content from the user  for (int i=1;i<=linescount;i++) {  System.out.println("Enter line"+i+":");  content.add(obj.nextLine());  }  // save the content into the file  boolean isSaved = addFile(folderpath, fn, content);  if (isSaved) {  System.out.println("file and data is saved");  controlFiles();  }  else {  System.out.println("some error occured");  controlFiles();  }  return;  // case 2 will let the user delete a file  case 2:  // variable decleration and assigning the folderpath  String filename;  folderpath ="C:\\Users\\hadlagak\\Desktop\\Test";  // ask the user to enter file name  System.out.println("enter filename to be deleted");  filename=obj.nextLine();    // to check if the file exict in the folder it will delete it  boolean isDeleted = deleteFile(folderpath, filename);  if(isDeleted) {  System.out.println("File is Deleted");  controlFiles();  }  else {  System.out.println("Either file is not deleted or does not exict");  controlFiles();  }    return;  // case 3 allows the user to search for a file  case 3:  // assign the folder path  folderpath ="C:\\Users\\hadlagak\\Desktop\\Test";  // ask the user to enter the file name  System.out.println("enter filename to be searched for");  filename=obj.nextLine();    boolean isFound = searchFile(folderpath, filename);  if(isFound) {  System.out.println("File is present");  controlFiles();  }  else {  System.out.println("Either file is not present");  controlFiles();  }  return;  // returning to the landing page  case 4:  landingPage();  return;    default :  System.out.println("Please Enter a Valid Number");  controlFiles();  }  }  /\*\*  \* This method print out the file in the assigned path in ascending order  \*/  static void viewFiles() {    // Declaring the files path  File fileDir = new File("C:\\Users\\hadlagak\\Desktop\\Test");  // seraching the files and sorting them in ascending order  if(fileDir.isDirectory()){  // declare a list to store file names  List<String> listFile = Arrays.asList(fileDir.list());  Collections.sort(listFile);  // printing out the results  System.out.println("---------------------------------------");  System.out.println("Here Are Your Files");  for(String s:listFile){  System.out.println(s);}}  else{  System.out.println(fileDir.getAbsolutePath() + " is not a directory");  }  //return the user to the landing page after showing the results    System.out.println();  System.out.println();  System.out.println("---------------------------------------");  System.out.println();  System.out.println();  landingPage();    }      /\*\*  \* this method allows the user to add new file  \*/  static boolean addFile (String folderpath, String fn, List<String> content) {  try {  // creating new file  File fl = new File (folderpath, fn);  FileWriter fw = new FileWriter(fl);  // creating the content in the file  for (String s:content)  {  fw.write(s+"\n");  }  fw.close();  return true;  }  catch(Exception Ex) {  return false;  }  }  /\*\*  \* this method allows the user to delete an exiting file  \*/  static boolean deleteFile (String folderpath, String filename) {  // to look for the file needed  File file = new File(folderpath+"\\"+filename);  try {  // to delete the file if exict  if (file.delete())  return true;  else  return false;  }  catch (Exception Ex) {  return false;  }    }  /\*\*  \* this method allows the user to search for an exiting file  \*/  static boolean searchFile (String folderpath, String filename) {  // to search for the file  File file = new File(folderpath+"\\"+filename);  if (file.exists())  return true;  else  return false;  }    } |