**Project: Lockers Private Limited (LockedMe.com)**

|  |  |
| --- | --- |
| **Developer** | **JJ** |
| **Version** | **1.0** |

Table of Contents

**Sprints2**

Sprint 1 2

Sprint 22

Sprint 32

**Algorithm2**

**Algorithm Description** 4

**Technologies Used** 4

**Git Repository details** 4

**Code Walkthrough 5**

RunApp.java5

MainMenu.java6

MainMenu.java7

MainMenuImpl.java8

Subenu.java9

SubMenuImpl.java10

**1. Sprint**

* There will be 3 sprints and Member team will be 2.
* Total duration of the project will be of 80 hours i.e. 2 weeks (business days)
  + Adi ( Scrum Master )
  + JJ ( Software engineer )

1. Sprint 1 will cover
   * Creating project flow diagrams i.e. Algorithms, Create GIT Repository, Setting up access to Machine and Installation of required software.
   * Start the application with valid input otherwise error message.
   * User is able to exit the application without any issue.
   * Create prototype of an application in Java. ( 15 % )
   * Update the work in GIT
   * Presentation of finished work to Customer
2. Sprint 2 will cover
   * Create prototype of an application in Java (85 % )
   * User should be able to interact with file system and should be able to perform Add file, Delete file and Navigate to Main menu.
   * User should be able to search the file in dedicated path.
   * Update work in GIT
   * Presentation of finished work to Customer
3. Sprint 3 will cover
   * Final proto type for an application (100 %)
   * Perform Unit test/Integration Test
   * Update work in the GIT repository
   * Present finished work to Customer and final signoff.
   * Pizza Party

**2. Algorithm:**

Start App

Return File

User Interaction

Close the App.

Search File

Add File

Delete File

Navigate to Main

1. **Algorithm description :**
2. Click on the Bat file to start an application
3. User prompted with list of options. Select one of the listed options.
4. If Option
   1. 1 : List out the files
   2. 2: User Interaction e.g. Add file, Delete file, Search file or Go to Main Menu.
   3. 3: exit from the application
5. In User Interaction
   1. 1: Add file option, Enter the name of the file with format that needs to be added. If file already exists, message with “file already exits”. If no existing file with same name and format, it will create new file with a success message. After success message, enter ‘e’ to exit from add file operation or ‘c’ to continue the operation again.
   2. 2: Delete file option, it will ask for file name and format. If file is found, it will delete the file and return success message. If file is not found, it will message out file not found. After message, enter ‘e’ to exit Delete file option, or ‘c’ for continue the operation again.
   3. 3: Search files option, to enter a file name with format. After file name with format is entered. It will search the file in path and return success message if the file found or it will enter file not found. After success message enter ‘e’ to exit from search file option or ‘c’ for continue search operation again.
   4. 4: Navigate to main page.
6. **Technologies Used:**
   1. Java Data Structures.
   2. Java Exception Handling
   3. Java Input/Output operations for Files.
   4. Java Utilities classes like Scanner
7. **Git Repository details.**

[**https://github.com/shailyjeet/fsd/tree/main/com.fsd1.Typecast/src/com/fsd1/LockerPvtLtd**](https://github.com/shailyjeet/fsd/tree/main/com.fsd1.Typecast/src/com/fsd1/LockerPvtLtd)

1. **Code Walkthrough** 
   1. RunApp.java

|  |
| --- |
| /\*\*  \*  \*/  **package** com.fsd1.LockerPvtLtd;  /\*\*  \* **@author** JJ  \*  \*/  **public** **class** RunApp {  /\*\*  \* runApp is the first Java program that is called.  \* It calls mainMenu.java  \*/  **static** mainMenu *obj* = **new** mainMenu();  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  System.***out***.println(":::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::");  System.***out***.println(":::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::");  System.***out***.println(":::::::::::::::::::::LockedMe.com :::::::::::::::::::::::::::::");  System.***out***.println(":::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::");  System.***out***.println(":::::::::::::::Application created by: JJ::::::::::::::::::::::");  System.***out***.println(":::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::");  System.***out***.println(":::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::");  System.***out***.println("please select from the below options: ");  System.***out***.println(" ");  System.***out***.println("Enter 1. to get the file name in the directory ");  System.***out***.println("Enter 2. to search, add or delete files in the directory");  System.***out***.println("Enter 3. to Exit the Application");  **do** {    *obj*.mainMenuimplementation();    } **while** (**true**);  }  } |

* 1. **MainMenu.java**

|  |
| --- |
| **package** com.fsd1.LockerPvtLtd;  /\*\*  \* **@author** JJ  \*  \*/  **import** java.util.Scanner;  **public** **class** MainMenu {  /\*\*  \* mainMenu Java application program provides menu options  \* 1 to get the files in directory  \* 2 to search, add file in the directory  \* 3 Exit maain Application.  \*/  **static** Scanner *sc* = **new** Scanner(System.***in***);  **static** mainMenuImpl *mmi* = **new** mainMenuImpl();  **static** subMenu *sm* = **new** subMenu();    **public** **void** mainMenuimplementation()  {  **do**  {  String user\_input = *sc*.next();    **switch**(user\_input)  {  // case to Retrieve file name in the directory  **case** "1":  *mmi*.getFileNames();  **break**;    // case to manage the directory files  **case** "2":  *sm*.subMenuImplementation();  **break**;    // case to exit application  **case** "3":  *mmi*.exitApplication();  **break**;    **default**:  System.***out***.println("Invalid input: Please Eenter from the following options:");  System.***out***.println("Enter 1. to get the file name in the directory ");  System.***out***.println("Enter 2. to search, add or add files in the directory");  System.***out***.println("Enter 3. to Exit the Application");    }    }  **while** (**true**);  }  } |

* 1. **MainMenuImpl.java**

|  |
| --- |
| **package** com.fsd1.LockerPvtLtd;  /\*\*  \* **@author** JJ  \*  \*/  **import** java.io.File;  **import** java.util.Scanner;  **public** **class** MainMenuImpl  {    **static** String *path* = "C:\\Users\\shail\\eclipse-workspace\\com.fsd1.Typecast\\src\\com\\fsd1\\LockerPvtLtd\\Documents";  **static** mainMenu *obj* = **new** mainMenu();    /\*\*  \* Method to Retrieve file names in the directory (ASC order)  \* The method does not return anything.  \* The method displays list of files in the current Documents directory.  \*/  **public** **void** getFileNames()  {    System.***out***.println("The directory contain following files:");  File dir = **new** File(*path*);  File[] listofFiles = dir.listFiles();  String[]flist = dir.list();    **if** (flist == **null**)  {  System.***out***.println("The directory is empty");    }    **for**(File file : listofFiles)  {  **if** (file.isFile())  {  System.***out***.println(file.getName());    }    }  }    /\*\*  \* Method to exit an application. Before exit, It will prompt a question  \* Y/Y is Yes and it exit an application  \* N/n is NO and it will take to Main Menu.  \*/    **public** **void** exitApplication()  {  Scanner sc = **new** Scanner(System.***in***);  System.***out***.println("Are you sure you want to exit the Application?");  System.***out***.println("\n Enter (Y/y)->YES To exit the Application?");  System.***out***.println("\n Enter (N/n)-> NO To return Main Menu ");  String exit = sc.next();    **if**(exit.equals("y") || exit.equals("Y"))  {  System.***out***.println("thank you for using the Application");  System.*exit*(0);  }  **else** **if** (exit.equals("n") || exit.equals("N"))  {  System.***out***.println("1. to search the file name");  System.***out***.println("1. to get the file name in the directory ");  System.***out***.println("2. to search, add or add files in the directory");  System.***out***.println("3. to Exit the Application");  *obj*.mainMenuimplementation();    }  **else** **if**(!(((exit.equals("n")||exit.equals("N"))||(exit.equals("y")||exit.equals("Y")))))  {  System.***out***.println("Please enter valid input");  }    sc.close();  }    } |

* 1. **Submenu.java**

|  |
| --- |
| **package** com.fsd1.LockerPvtLtd;  /\*\*  \* **@author** JJ  \*  \*/  **import** java.util.Scanner;  **public** **class** SubMenu {  **static** mainMenu *mm* = **new** mainMenu();  **static** mainMenuImpl *mmi* = **new** mainMenuImpl();  **static** subMenu *sm* = **new** subMenu();  **static** subMenuImpl *smi* = **new** subMenuImpl();  **static** Scanner *sc* = **new** Scanner(System.***in***);    /\*\*  \* The method subMenuImplementation provides menu options to Add file, delete file,  \* search file or Go back to main menu. It also prints default options is any  \* parameter other than 1,2 or 3 is entered.  \*/  **public** **void** subMenuImplementation() {  System.***out***.println("Enter 1. to search the file name");  System.***out***.println("Enter 2. to add the file ");  System.***out***.println("Enter 3. to delete the file");  System.***out***.println("Enter 4. to go Back to the Main menu");  **do** {  String user\_input = *sc*.next();  **switch** (user\_input) {  **case** "1":  *smi*.searchFile();  **break**;  **case** "2":  *smi*.addFile();  **break**;  **case** "3":  *smi*.deleteFile();  **break**;  **case** "4":  System.***out***.println("Enter 1. to get the file name in the directory ");  System.***out***.println("Enter 2. to search, add or delete files in the directory");  System.***out***.println("Enter 3. to Exit the Application");  *mm*.mainMenuimplementation();  **break**;  **default**:  System.***out***.println(" Invalid input, please Enter: ");  System.***out***.println("Enter 1. to search the file name");  System.***out***.println("Enter 2. to add the file ");  System.***out***.println("Enter 3. to delete the file");  System.***out***.println("Enter 4. to go Back to the Main menu");  **break**;  }  } **while** (**true**);  }  **}** |

* 1. **SubMenuImpl.java**

|  |
| --- |
| **package** com.fsd1.LockerPvtLtd;  **import** java.io.File;  **import** java.io.IOException;  **import** java.util.Scanner;  /\*\*  \* **@author** JJ  \*  \*/  **public** **class** SubMenuImpl {  **static** String *path* = mainMenuImpl.*path*;  **static** Scanner *sc1* = **new** Scanner(System.***in***);  **static** Scanner *sc2* = **new** Scanner(System.***in***);  **static** Scanner *sc3* = **new** Scanner(System.***in***);  /\*\*  \* The Method searchfile() does not return anything.  \* The method just prints Found if the specified filename exists in the directory.  \*/    **public** **void** searchFile() {  System.***out***.println("Please enter the file name");  String filename = *sc1*.next();  File searchfile = **new** File(*path* + "/" + filename);  **if** (searchfile.exists()) {  System.***out***.println(" Found : File \"" + filename + "\" at location" + *path* + "\n");  } **else** {  System.***out***.println("File not found \n");  }  }  /\*\*  \* The Method addfile() does not return anything  \* The addfile() simply creates the filename of extenstion provided.  \* The method also checks if mentioned file already exists in the directory.  \* If filename specified is not created it prints out error message.  \*/  **public** **void** addFile() {  System.***out***.println("Please enter the file name to be added in the directory" + " " + *path*);  String filename = *sc2*.next();  File addfile = **new** File(*path* + "/" + filename);  **try** {  **if** (addfile.createNewFile())  System.***out***.println(" File \"" + filename + "\" added to " + *path* + "\n");  **else** **if** (addfile.exists())  System.***out***.println(" File \"" + filename + "\" already exists at " + *path* + "\n");  **else**  System.***out***.println("Something went wrong. File NOT added\n");  } **catch** (IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  **if** (*sc1*.equals("2") || *sc1*.equals("3") || *sc1*.equals("4")) {  **return**;  }  }    /\*\*  \* The method deletefile() does not return anything.  \* The method simply deletes the filename specified.  \* Error message is printed if spcified filename is not deleted.  \*/  **public** **void** deleteFile() {    System.***out***.println("Please enter the file name to be deleted from the directory:" + " " + *path*);  String filename = *sc3*.next();  File file = **new** File(*path*+ "/" + filename);    **if**(file.delete())  {  System.***out***.println(" File \""+filename +"\" deleted from " +*path* + "\n");  }  **else**  {  System.***out***.println(" Delete operation failed. File count not be found \n");  }    }  } |