**Project 1**

**Duration: 2 sprints (14 days)**

**Sprint planning meeting**

As decided by the Scrum Team whole prototype of the project will be delivered in 2 sprints.

**Schedule for Sprint 1**

Duration: 1 week

* Completion of all documentation part which includes whole description of the sprints, specifications of the project, description of algorithms used and all flowcharts etc.
* Setting up of GitHub Account.
* Setting up of Git.
* Communicating between Git and GitHub.
* Making algorithm and flowchart for the project.

**Schedule for Sprint 2**

Duration: 1 week

* Writing actual code for the given project.
* Testing the code with all possible user inputs.
* Checking for any kind of exception or error handling if it is occurring in the code.
* Final submission of the code

**Sprint 1:**

**Documentation:**

**Company Name:** Lockers Pvt. Ltd.

**Application Name:** LockedMe.com

**Specification:**

Application is capable of taking files from the user and making the directory of the file. Whenever required, you can access the directory in the ascending order of the file names. Provision is provided for deleting any file name, searching for a particular file name and adding more files to the directory.

**Setting Up Git and GitHub Account:**

Git and GitHub account need to be created as per the customer requirement for tracking and enhancement of the project.

**Java Concepts Used:**

Loops, Conditional Statements, Collections, Exception handling, Methods

**Data Structures Used:**

Searching and sorting algorithms

**Algorithm and Flowchart:**

1. One class named as Project is created with main function.
2. All required libraries are imported.
3. One global Set (TreeSet) is created with the name fileDirectory.
4. In main method scanner object and object of the main class Project is created.
5. Variables are initialized in main method.
6. Method initialDisplay is called for display of all required information to the user.
7. Infinite while loop is created to take continuous inputs from the user.
8. Initially 3 options are provided to the user:

* Press 1 for: To get the file names in ascending order
* Press 2 for: To perform other options
* Press 0 for: To exit or close the application

1. If user presses 1 then display the content of the TreeSet. It will be automatically in ascending order as TreeSet displays the content in ascending order.
2. If user press 2 then they will get following options:

* Press 3 for: To add a file to the user directory
* Press 4 for: To delete a file from user directory
* Press 5 for: To search for a file name in user directory
* Press 6 for: To go back to main context

1. If user presses 3 then method addFile() is called which will ask for the file name and then add the file to the existing directory. If the file name already exists then no change will occur.
2. If user presses 4 then method deleteFile() will be called which will ask for the file name that user want to delete. If typed file name is found in the directory then that file will be deleted otherwise message will be shown that file does not found.
3. If user presses 5 then method searchFile() will be called which will ask for the name of the file user want to search. If the typed name is found in the directory then message will be shown on the screen that file found otherwise message will be shown that file does not found.
4. If user presses 6 then they will be directed back to one step back.
5. If user presses 0 then application will be closed with the message on the user screen.
6. If user presses anything else then message will be shown on the screen that you have entered a wrong input.

**Sprint 2:**

**Code:**

**package** program.jav.lockers;

**import** java.util.Scanner;

**import** java.util.Set;

**import** java.util.TreeSet;

**public** **class** Project {

Set<String> fileDirectory= **new** TreeSet<String>();

**public** **static** **void** main(String[] args){

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

**int** input =7, input2;

Project p= **new** Project();

p.initialDisplay();

**while**(**true**){

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

System.***out***.println("Press 1 for: To get the file names in ascending order");

System.***out***.println("Press 2 for: To perform other options");

System.***out***.println("Press 0 for: To exit or close the application");

**try**{

input=sc.nextInt();

}

**catch**(Exception e){

System.***out***.println("Please enter valid input. Restart your code.");

**break**;

}

**if**(input==0){

System.***out***.println("You have chosen to close the program.");

**break**;

}

**switch**(input){

**case** 1:

p.display();

**break**;

**case** 2:

p.otherOptions();

**try**{

input2=sc.nextInt();

}

**catch**(Exception e){

System.***out***.println("Please enter valid input.");

**break**;

}

**switch**(input2){

**case** 3:

System.***out***.println("You have pressed 3");

p.addFile(sc);

**break**;

**case** 4:

System.***out***.println("You have pressed 4");

p.deleteFile(sc);

**break**;

**case** 5:

System.***out***.println("You have pressed 5");

p.searchFile(sc);

**break**;

**case** 6:

System.***out***.println("You have pressed 6");

**break**;

**default**:

System.***out***.println("You have entered wrong input.");

**break**;

}

**break**;

**default**:

System.***out***.println("You have entered wrong input.");

**break**;

}

}

}

**public** **void** initialDisplay(){

System.***out***.println("LockedMe.com");

System.***out***.println("Lockers Pvt. Ltd.");

System.***out***.println("You have following options to do operations: ");

}

**public** **void** otherOptions(){

System.***out***.println("Press 3 for: To add a file to the user directory");

System.***out***.println("Press 4 for: To delete a file from user directory");

System.***out***.println("Press 5 for: To search for a file name in user directory");

System.***out***.println("Press 6 for: To go back to main context");

}

**public** **void** addFile(Scanner sc){

String str;

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

str=sc.next();

fileDirectory.add(str);

}

**public** **void** deleteFile(Scanner sc){

String str;

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

str=sc.next();

**if**(fileDirectory.contains(str)){

fileDirectory.remove(str);

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

}

**else**{

System.***out***.println("File not Found");

}

}

**public** **void** searchFile(Scanner sc){

String str;

System.***out***.println("Enter the file name you want to search in directory: ");

str=sc.next();

**if**(fileDirectory.contains(str)){

System.***out***.println("File found in the directory.");

}

**else**{

System.***out***.println("File does not found in the directory");

}

}

**public** **void** display(){

**if**(fileDirectory.size()==0){

System.***out***.println("File directory is empty.");

}

**else**{

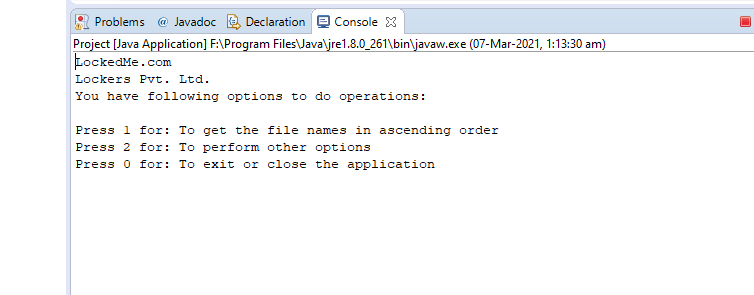
System.***out***.println(fileDirectory);

}

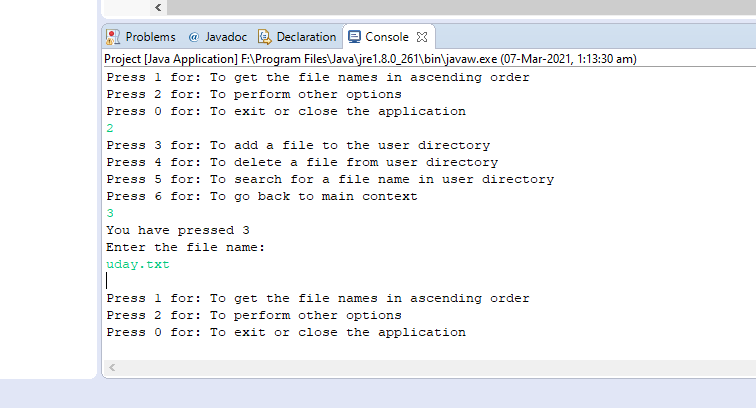
}

}

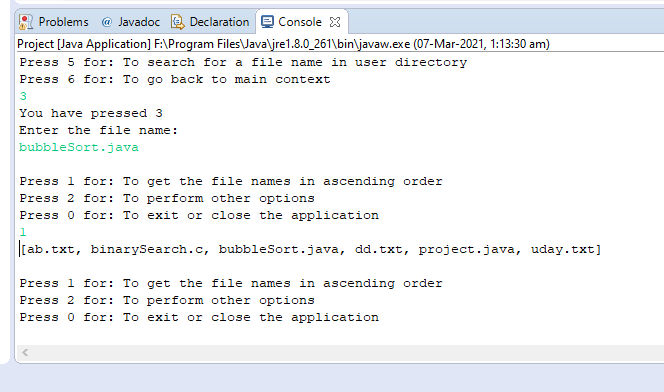
**Testing:**

****

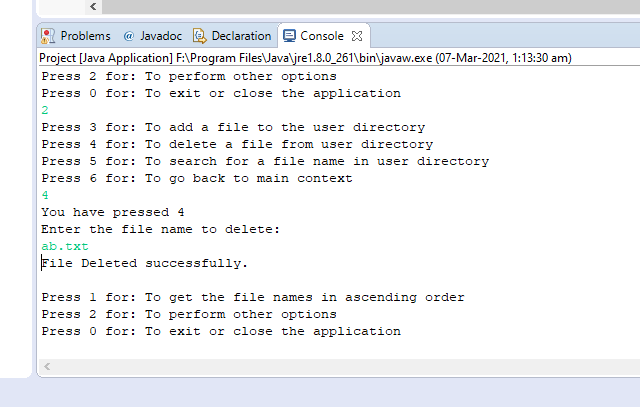
**Initial Page**

****

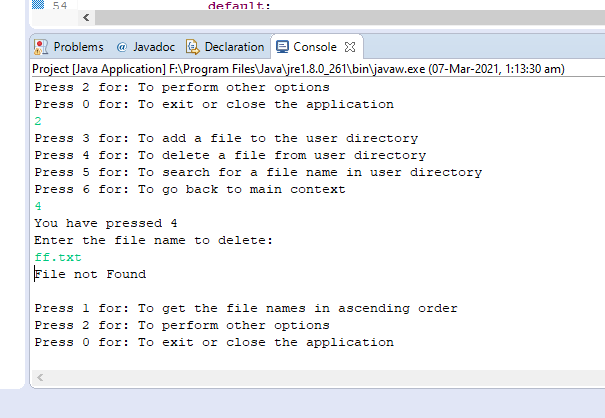
**Taking inputs**

****

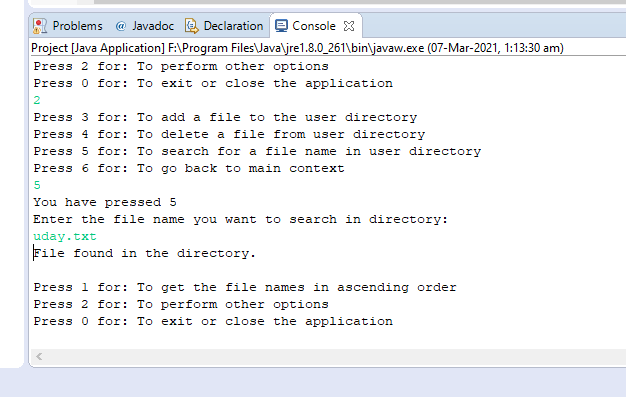
**Displaying file names in ascending order**

****

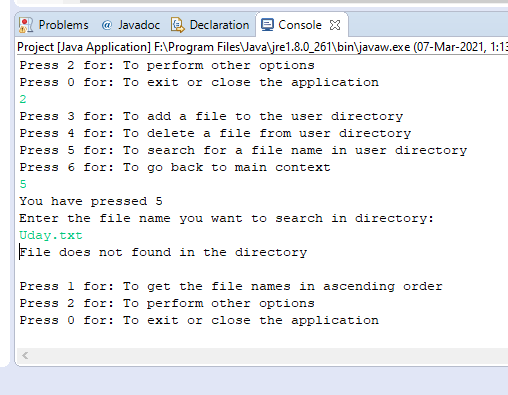
**Delete successfully**

****

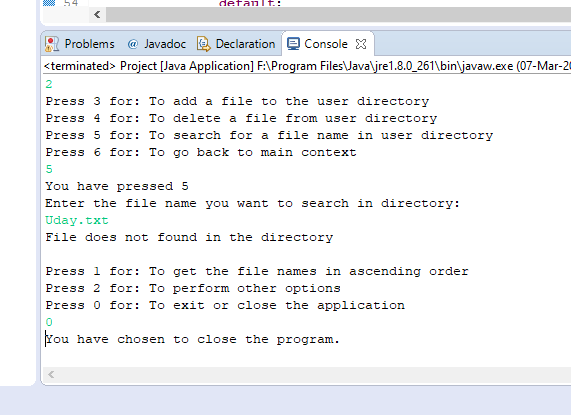
**Delete unsuccessful**

****

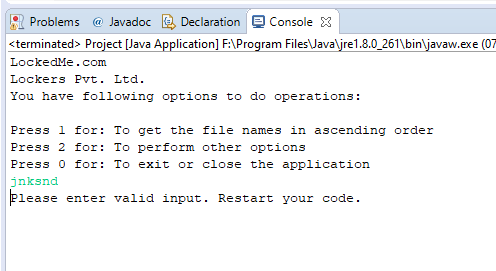
**Searching successful**

****

**Searching unsuccessful**

****

**Closing program**

****

**Wrong input**