Virtual Key for Repositories

This document contains sections for:

The code for this project is hosted at https://github.com/pritamkundu2000/Phase1Project-Virtual-Key.git
The project is developed by Pritam Kundu.

Sprints planning and Task completion

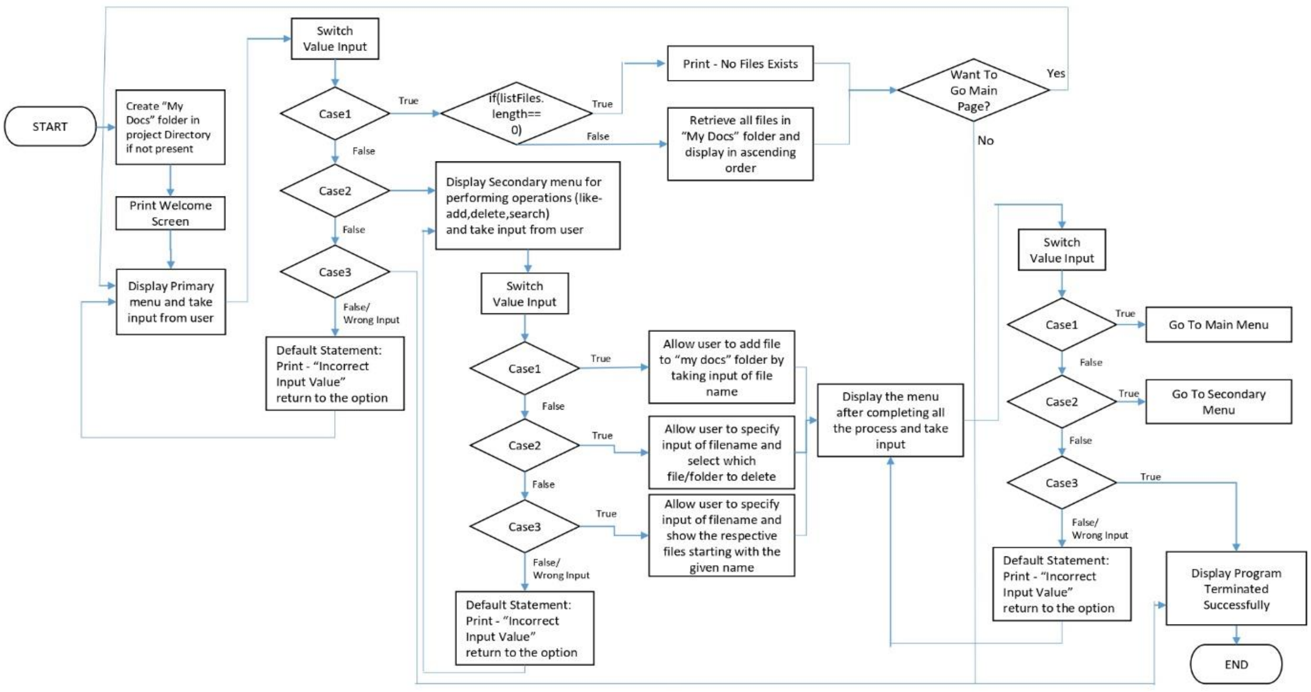
The project is planned to be completed in 1 sprint. Tasks assumed to be completed in the sprint are:

- 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



START
Create "My Docs" folder in project Directory if not present
Print Welcome Screen
Display Primary menu and take input from user
Switch Value Input
Case1
True
False
Case2
False
Case3
False/ Wrong Input
Default Statement: Print - "Incorrect Input Value" return to the option
if(listFiles. length== 0)
True
False
Print - No Files Exists
Retrieve all files in "My Docs" folder and display in ascending order
Want To Go Main Page?
Yes
No
Display Secondary menu for performing operations (like- add,delete,search) and take input from user
Switch Value Input
Case1
True
False
Case2
True
False
Case3
True
False/ Wrong Input
Allow user to add file to "my docs" folder by taking input of file name
Allow user to specify input of filename and select which file/folder to delete
Allow user to specify input of filename and show the respective files starting with the given name
Default Statement: Print - "Incorrect Input Value" return to the option
Display the menu after completing all the process and take input
Switch Value Input
Case1
True
Go To Main Menu
False
Case2
True
Go To Secondary Menu
False
Case3
True
False/ Wrong Input
Default Statement: Print - "Incorrect Input Value" return to the option
Display Program Terminated Successfully
END

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 for the entry point of the application (**Main.java**)
3. Writing a program in Java to display Primary Menu options available for the user (**Welcome.java**)
4. Writing a program in Java to handle First Option of Primary Menu selected by user (**FirstOpt.java**)
5. Writing a program in Java to handle Secondary Menu options selected by user (**SecondOpt.java**)
6. Writing a program in Java to perform the File operations as specified by user (**Operations.java**)
7. 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 **Main** 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 entry point of the application (**Main.java**)

```
public class Main {
        public static void main(String[] args) {
                Welcome.welcome();
                System.out.println("Thanks for Using our application");
        }
}
```

Step 3: Writing a program in Java to display Menu options available for the user (**Welcome.java**)

- Select your project and go to File -> New -> Class.
- Enter **Welcome** in class name and click on “Finish.”
- **Welcome** consists methods for -:

3.1. Displaying Welcome Screen

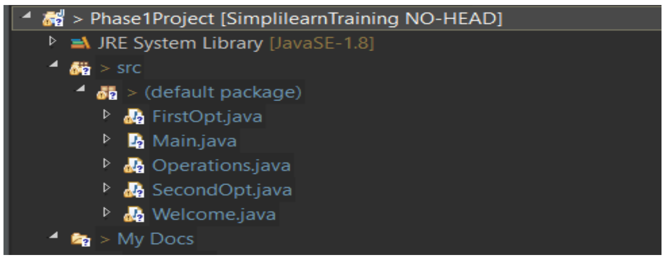
3.2. Displaying First Menu

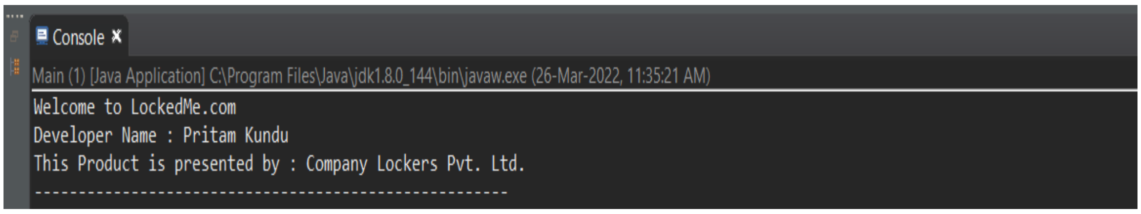
Step 3.1: Writing method to display Welcome Screen

```
        public static void welcome(){

                File dirPath=new File("My Docs");
                if(!dirPath.exists()) {
                        dirPath.mkdir();
                }

                System.out.println("Welcome to LockedMe.com");
                System.out.println("Developer Name : Pritam Kundu");
                System.out.println("This Product is presented by : Company Lockers Pvt. Ltd.");

                welcomePage(dirPath);


        }
```

Output:

It will make new "My Docs" Directory if not present


```
◢ > Phase1Project [SimplilearnTraining NO-HEAD]
  ▷ JRE System Library [JavaSE-1.8]
  ◢ > src
    ◢ > (default package)
      ▷ FirstOpt.java
      ▷ Main.java
      ▷ Operations.java
      ▷ SecondOpt.java
      ▷ Welcome.java
  ◢ > My Docs
```

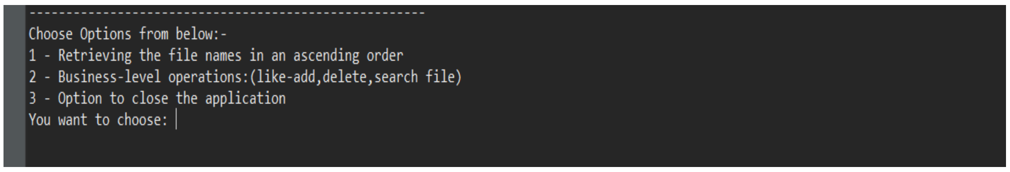


```
Main (1) [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (26-Mar-2022, 11:35:21 AM)
Welcome to LockedMe.com
Developer Name : Pritam Kundu
This Product is presented by : Company Lockers Pvt. Ltd.
----------------------------------------------------
```

3.2 Displaying Primary Menu:-

```
public static void welcomePage(File dirpath) {
        try {
            System.out.println("----------------------------------------------------");

            System.out.println("Choose Options from below:-");
            System.out.println("1 - Retrieving the file names in an ascending order");
            System.out.println("2 - Business-level operations:(like-add,delete,search file)");
            System.out.println("3 - Option to close the application");

            Scanner sc=new Scanner(System.in);

            System.out.print("You want to choose: ");
            int input=sc.nextInt();
            System.out.println("----------------------------------------------------");

            switch (input) {
            case 1:
                FirstOpt.firstOption(dirpath);
                break;

            case 2:
                SecondOpt.secondOption(dirpath);
                break;
            case 3:
                System.out.println("Application Closed-----");
                break;
            default:
                System.out.println("----------------------------------------------------");
```

```
                                        System.out.println("Input type Error---- Enter only Integer as input");
                                        welcomePage(dirpath);
                                }

                        } catch (InputMismatchException e) {
                                System.out.println("------------------------------------------------------");
                                System.out.println("Input type Error---- Enter only Integer as input");
                                welcomePage(dirpath);
                        }
                }
        }
```



```
-----------------------------------------------------
Choose Options from below:-
1 - Retrieving the file names in an ascending order
2 - Business-level operations:(like-add,delete,search file)
3 - Option to close the application
You want to choose: |
```

STEP 4:-

Writing a program in Java to handle First Option of Primary Menu selected by user (**FirstOpt.java**)

- Select your project and go to File -> New -> Class.
- Enter **FirstOpt** in class name and click on "Finish."
- **FirstOpt** consists methods for -:

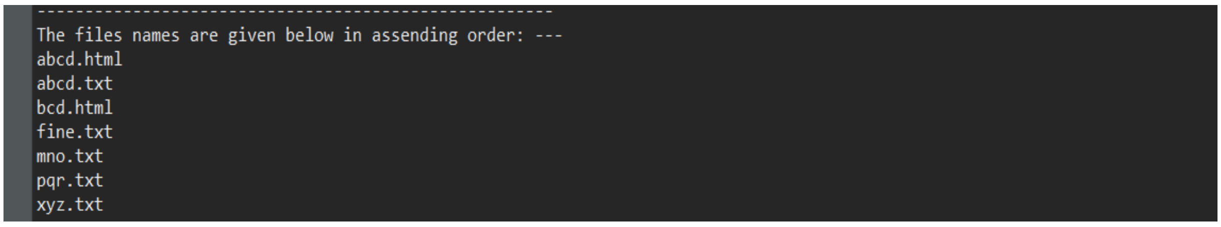
4.1. Retriving all the files in ascending order.

4.2. Handling input selected by user To go to Primary menu or not.

4.1. Retriving all the files in ascending order:-

```
public static void firstOption(File dirpath) {


                File[] contents=dirpath.listFiles();
                if(contents.length==0) {
                        System.out.println("No files Exists");
                }
                else {
                        System.out.println("The files names are given below in 
assending order: ---");
                        for(File each: contents) {
                                System.out.println(each.getName());
                        }
                }
                firstoptionOperation(dirpath);
        }
```

OUTPUT:-

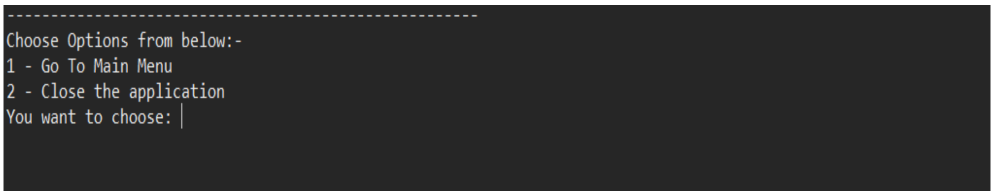


```
-----------------------------------------------------
The files names are given below in assending order: ---
abcd.html
abcd.txt
bcd.html
fine.txt
mno.txt
pqr.txt
xyz.txt
```

4.2. Handling input selected by user To go to Primary menu or not

```
public static void firstoptionOperation(File dirpath) {
		try {
			System.out.println("-----------------------------------------------------");

			System.out.println("Choose Options from below:-");
			System.out.println("1 - Go To Main Menu");
			System.out.println("2 - Close the application");

			Scanner sc=new Scanner(System.in);

			System.out.print("You want to choose: ");
			int input=sc.nextInt();
			System.out.println("-----------------------------------------------------");

			switch (input) {
			case 1:
				Welcome.welcomePage(dirpath);
				break;

			case 2:
				System.out.println("Application Closed-----");
				break;

			default:
				System.out.println("Wrong Input given");
				firstOption(dirpath);
				break;
			}

		} catch (InputMismatchException e) {
			System.out.println("-----------------------------------------------------");
			System.out.println("Input type Error---- Enter only Integer as input");
			firstOption(dirpath);
		}
		}
```

OUTPUT:-



```
-----------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Close the application
You want to choose: 
```

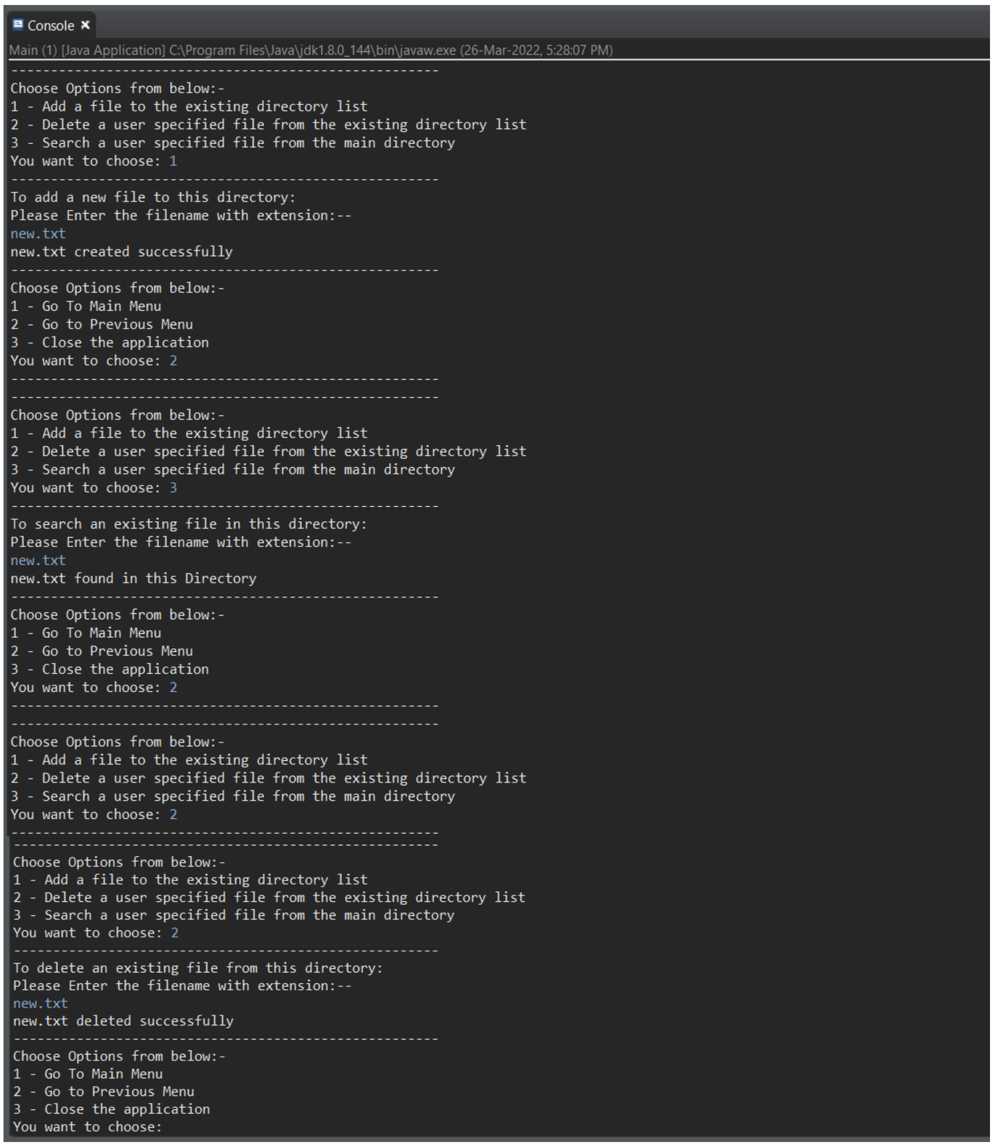
STEP-5: Writing a program in Java to handle Secondary Menu options selected by user (**SecondOpt.java**)

- Select your project and go to File -> New -> Class.
- Enter **SecondOpt** in class name and click on "Finish."
- **SecondOpt** consists methods for -:

- ### 5.1. Handling input selected by user in secondary Menu for File Operations

```
public static void secondOption(File dirpath) {

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

                    System.out.println("Choose Options from below:-");
                    System.out.println("1 - Add a file to the existing directory list");
                    System.out.println("2 - Delete a user specified file from the existing directory list");
                    System.out.println("3 - Search a user specified file from the main directory");

                    Scanner sc=new Scanner(System.in);

                    System.out.print("You want to choose: ");
                    int input=sc.nextInt();
                    System.out.println("-----------------------------------------------------");

                    switch (input) {
                    case 1:
                            Operations.add(dirpath);
                            break;

                    case 2:
                            Operations.delete(dirpath);
                            break;
                    case 3:
                            Operations.search(dirpath);
                            break;
                    default:
                            System.out.println("Wrong Option Choosen");
                            secondOption(dirpath);
                            break;
                    }

            } catch (InputMismatchException e) {
                    System.out.println("-----------------------------------------------------");
                    System.out.println("Input type Error---- Enter only Integer as input");
                    secondOption(dirpath);
            }
        }
```

OUTPUT:-



```
Console ✖
Main (1) [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (26-Mar-2022, 5:28:07 PM)
------------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 1
------------------------------------------------------
To add a new file to this directory:
Please Enter the filename with extension:--
new.txt
new.txt created successfully
------------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Go to Previous Menu
3 - Close the application
You want to choose: 2
------------------------------------------------------
------------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 3
------------------------------------------------------
To search an existing file in this directory:
Please Enter the filename with extension:--
new.txt
new.txt found in this Directory
------------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Go to Previous Menu
3 - Close the application
You want to choose: 2
------------------------------------------------------
------------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 2
------------------------------------------------------
------------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 2
------------------------------------------------------
To delete an existing file from this directory:
Please Enter the filename with extension:--
new.txt
new.txt deleted successfully
------------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Go to Previous Menu
3 - Close the application
You want to choose:
```

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

- Select your project and go to File -> New -> Class.
- Enter **Operations** in class name and click on "Finish."
- **Operations** consists methods for -:

5.1.Creating a file as specified by user input.

5.2.Deleting a file/folder from "My Docs" folder

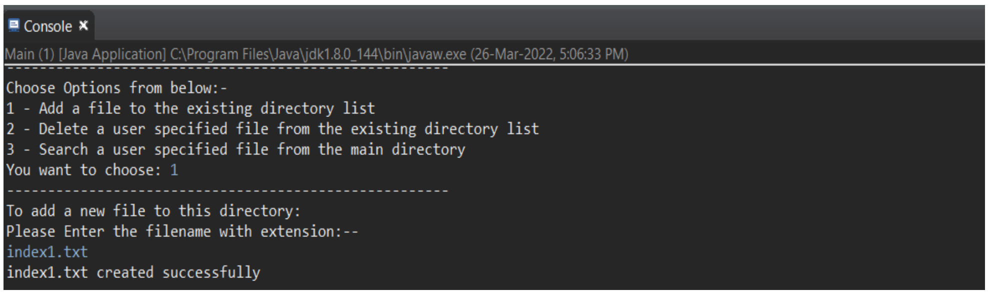
5.3.Search files as specified by user input and print that it is present in the directory.

5.4. Handling input selected by user To go to Primary menu or not

5.1.Creating a file/folder as specified by user input.-

```
public static void add(File dirpath) {

                    System.out.println("To add a new file to this directory:");
                    System.out.println("Please Enter the filename with extension:--");

                    Scanner scanner=new Scanner(System.in);
                    String file_name=scanner.next();


                    File newFile=new File(dirpath,file_name);

                    if(!newFile.exists()) {
                              newFile.createNewFile();
                              System.out.println(file_name+" created successfully");
                    }else {
                              System.out.println(file_name+" already exists");
                    }
                    secondOptionOperation(dirpath);

          }
```

OUTPUT:-

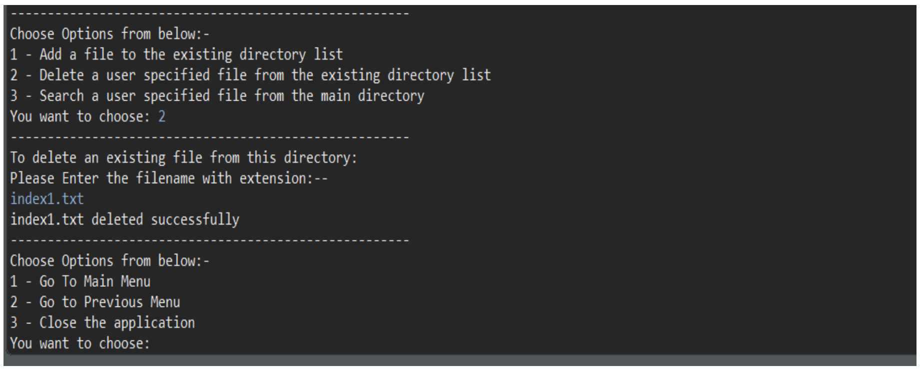


```
Main (1) [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (26-Mar-2022, 5:06:33 PM)
------------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 1
------------------------------------------------------
To add a new file to this directory:
Please Enter the filename with extension:--
index1.txt
index1.txt created successfully
```

5.2.Deleting a file/folder from “My Docs” folder:-

```
public static void delete(File dirpath) {

        System.out.println("To delete an existing file from this directory:");
        System.out.println("Please Enter the filename with extension:--");

        Scanner scanner=new Scanner(System.in);
        String file_name=scanner.next();


        File dFile=new File(dirpath,file_name);

        if(dFile.exists()) {
                dFile.delete();
                System.out.println(file_name+" deleted successfully");
        }else {
                System.out.println(file_name+" doesn't exists");
        }
        secondOptionOperation(dirpath);

    }
```

OUTPUT:-

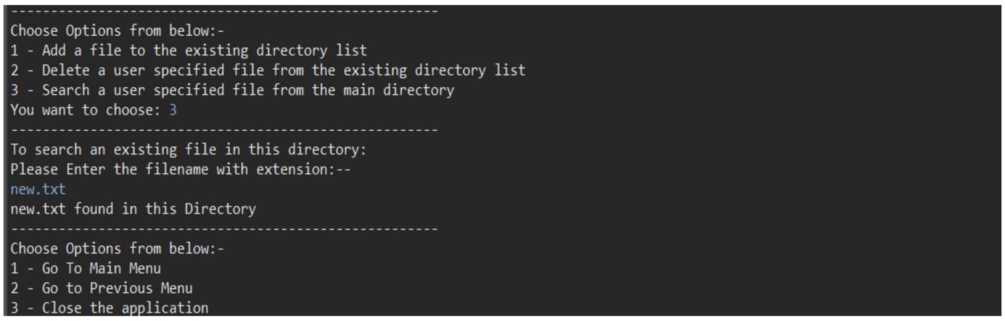


```
----------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 2
----------------------------------------------------
To delete an existing file from this directory:
Please Enter the filename with extension:--
index1.txt
index1.txt deleted successfully
----------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Go to Previous Menu
3 - Close the application
You want to choose:
```

5.3.Search files as specified by user input and print that it is present in the directory:-

```
public static void search(File dirpath) {

            System.out.println("To search an existing file in this directory:");
            System.out.println("Please Enter the filename with extension:--");

            Scanner scanner=new Scanner(System.in);
            String file_name=scanner.next();

            File[] allFileNames=dirpath.listFiles();

            int flag=0;
            for(File file:allFileNames) {
                    if(file.getName().equalsIgnoreCase(file_name)) {

                            flag=1;
                            break;
                    }
            }
            if(flag==0) {
                    System.out.println(file_name+" not found in this directory");
            }
            else {
                    System.out.println(file_name+" found in this Directory");
            }
            secondOptionOperation(dirpath);
    }
```

OUTPUT:-

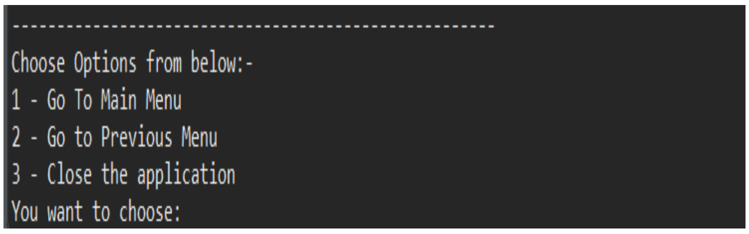


```
----------------------------------------------------
Choose Options from below:-
1 - Add a file to the existing directory list
2 - Delete a user specified file from the existing directory list
3 - Search a user specified file from the main directory
You want to choose: 3
----------------------------------------------------
To search an existing file in this directory:
Please Enter the filename with extension:--
new.txt
new.txt found in this Directory
----------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Go to Previous Menu
3 - Close the application
```

5.4. Handling input selected by user To go to Primary menu or not:

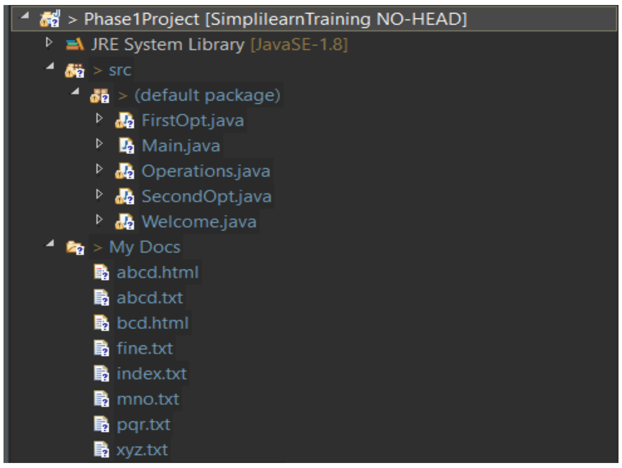
```
public static void secondOptionOperation(File dirpath) {
		try {
			System.out.println("----------------------------------------------------");

			System.out.println("Choose Options from below:-");
			System.out.println("1 - Go To Main Menu");
			System.out.println("2 - Go to Previous Menu");
			System.out.println("3 - Close the application");

			Scanner sc=new Scanner(System.in);

			System.out.print("You want to choose: ");
			int input=sc.nextInt();
			System.out.println("----------------------------------------------------");

			switch (input) {
			case 1:
				Welcome.welcomePage(dirpath);
				break;

			case 2:
				SecondOpt.secondOption(dirpath);
				break;

			case 3:
				System.out.println("Application Closed-----");
				break;

			default:
				System.out.println("Wrong Input given");
				secondOptionOperation(dirpath);
				break;
			}

		} catch (InputMismatchException e) {
			System.out.println("----------------------------------------------------");
			System.out.println("Input type Error---- Enter only Integer as input");
			secondOptionOperation(dirpath);
		}
	}
```

OUTPUT:-



```
----------------------------------------------------
Choose Options from below:-
1 - Go To Main Menu
2 - Go to Previous Menu
3 - Close the application
You want to choose:
```

To see List of the files on Eclipse, right click on Project and click "Refresh"-



```
> Phase1Project [SimplilearnTraining NO-HEAD]
  JRE System Library [JavaSE-1.8]
  > src
    > (default package)
      FirstOpt.java
      Main.java
      Operations.java
      SecondOpt.java
      Welcome.java
  > My Docs
    abcd.html
    abcd.txt
    bcd.html
    fine.txt
    index.txt
    mno.txt
    pqr.txt
    xyz.txt
```

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 name as input.

3. 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.

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 two options of viewing the files.

   6.1. Ascending order of folders first which have files sorted in them,
   6.2. Ascending order of all files and folders inside the “My Docs” 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.
- Allowing user to open a file and add the item in the file.