Virtual key for
your repository

Project Contains:

Project Details:

Project Objectives:

As a Full Stack Developer, complete the features of the application by planning the development in terms of sprints and then push the source code to the GitHub repository. As this is a prototyped application, the user interaction will be via a command line.

The flow and features of the application:

- Plan more than two sprints to complete the application
- Document the flow of the application and prepare a flow chart
- List the core concepts and algorithms being used to complete this application
- Code to display the welcome screen. It should display:
    - Application name and the developer details
    - The details of the user interface such as options displaying the user interaction information
    - Features to accept the user input to select one of the options listed
- The first option should return the current file names in ascending order. The root directory can be either empty or contain few files or folders in it
- The second option should return the details of the user interface such as options displaying the following:
    - Add a file to the existing directory list
        - You can ignore the case sensitivity of the file names
    - Delete a user specified file from the existing directory list
        - You can add the case sensitivity on the file name in order to ensure that the right file is deleted from the directory list
        - Return a message if FNF (File not found)
    - Search a user specified file from the main directory
        - You can add the case sensitivity on the file name to retrieve the correct file
        - Display the result upon successful operation
        - Display the result upon unsuccessful operation
    - Option to navigate back to the main context
- There should be a third option to close the application
- Implement the appropriate concepts such as exceptions, collections, and sorting techniques for source code optimization and increased performance

Developer Details:

- Hanuman Prasad (https://github.com/hanumanprasadvishwa/Assignment-1)
- hanumanprasad0108@gmail.com

Sprints Planning:

The scope of project has been divided into 3 sprints.

Sprint 1:

Understanding the requirement of the project and creating a flowchart & algorithm for the same.

Sprint 2:

Creating the Welcome screen, Main-Menu & Sub-Menu for the Application. Creating java program for the various tasks as per the flow chart created.

Sprint 3:

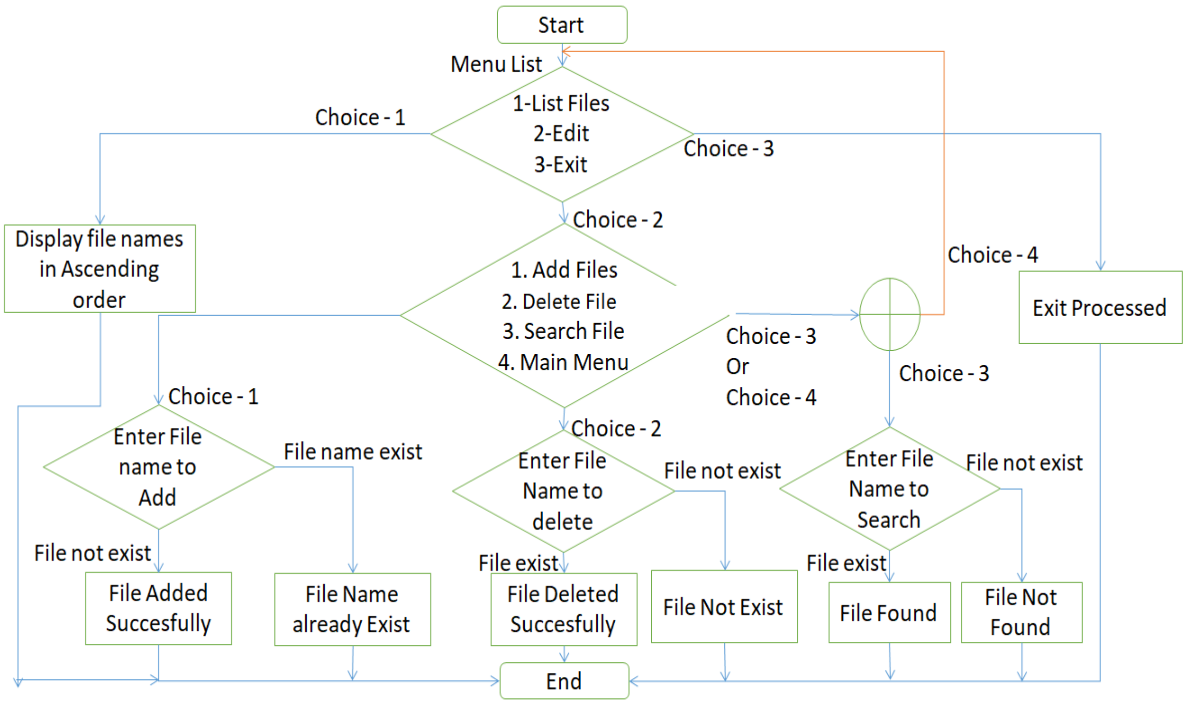
Implementing the different methods & writing the code to list the existing files in the specified directory,creating the new files,deleting & searching the files from the directory to create the fully working application.

Sprint 4:

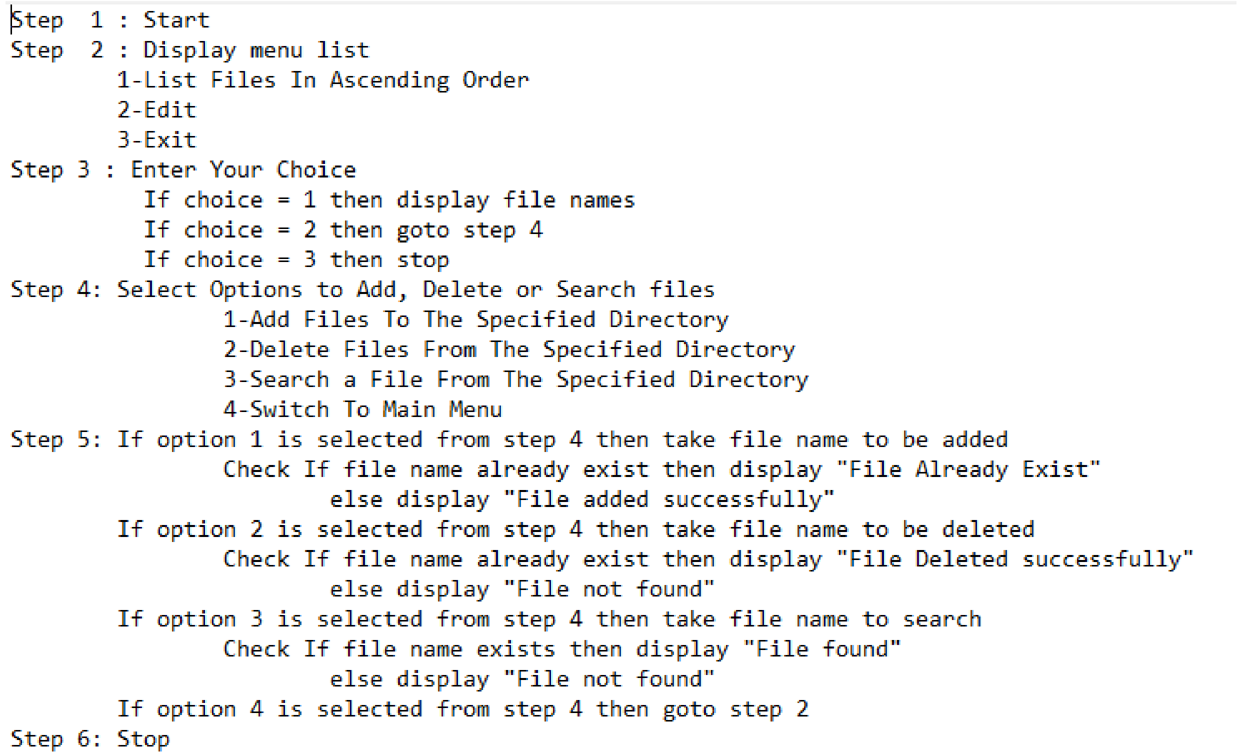
Testing and fixing the issues occurred in the application and validating with various inputs & Pushing the program to Github.

Flow Charts & Algorithms:

Flow chart:



Algorithm:



```
Step  1 : Start
Step  2 : Display menu list
        1-List Files In Ascending Order
        2-Edit
        3-Exit
Step 3 : Enter Your Choice
          If choice = 1 then display file names
          If choice = 2 then goto step 4
          If choice = 3 then stop
Step 4: Select Options to Add, Delete or Search files
                1-Add Files To The Specified Directory
                2-Delete Files From The Specified Directory
                3-Search a File From The Specified Directory
                4-Switch To Main Menu
Step 5: If option 1 is selected from step 4 then take file name to be added
                Check If file name already exist then display "File Already Exist"
                        else display "File added successfully"
        If option 2 is selected from step 4 then take file name to be deleted
                Check If file name already exist then display "File Deleted successfully"
                        else display "File not found"
        If option 3 is selected from step 4 then take file name to search
                Check If file name exists then display "File found"
                        else display "File not found"
        If option 4 is selected from step 4 then goto step 2
Step 6: Stop
```

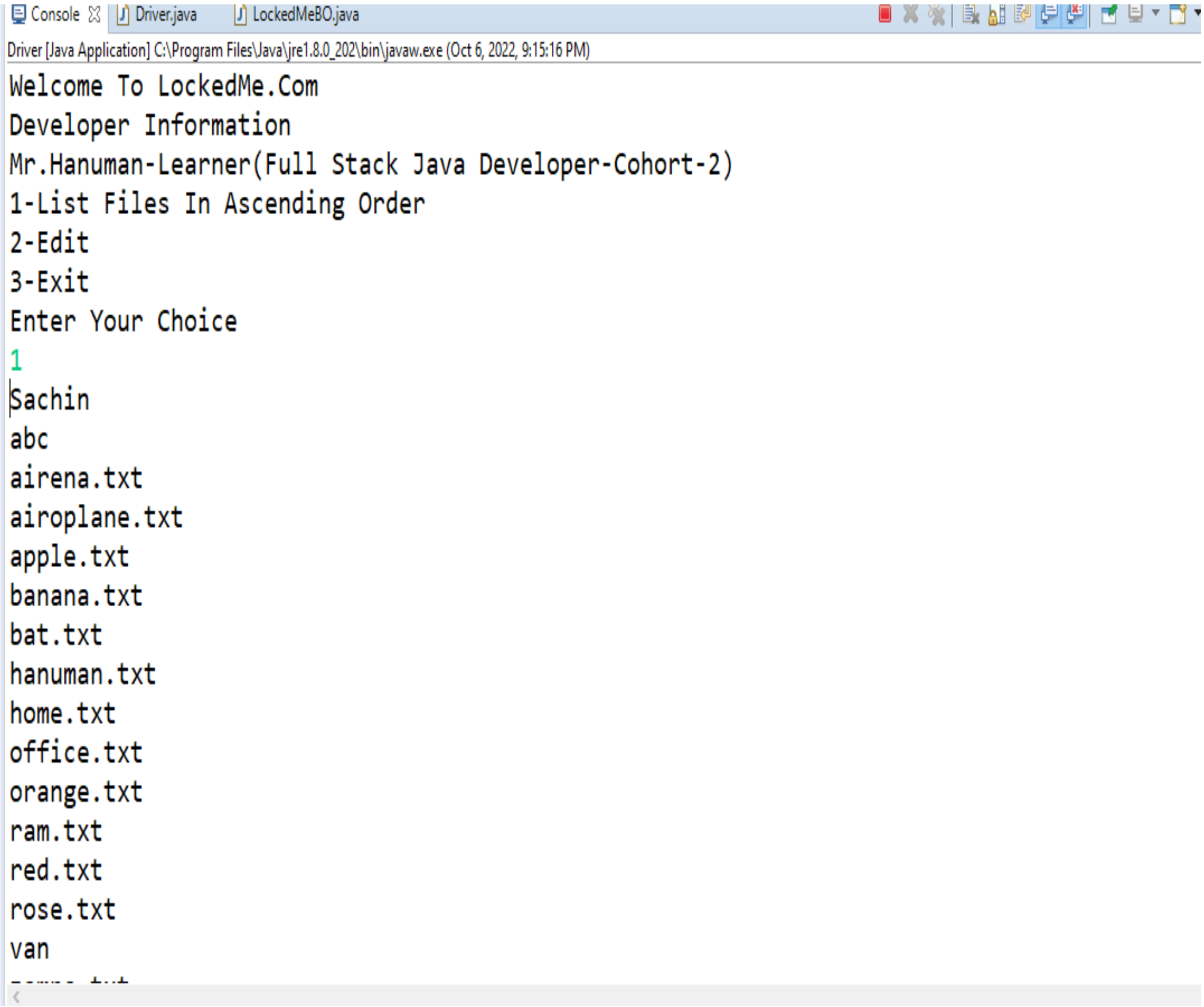
```
package Com.lockedme;

import java.io.File;

public class LockedMeBO {
    public void displayFilesInAscendingOrder(String directoryPath) {
        try {
            File f = new File(directoryPath);
            File files[] = f.listFiles();
            Sort(files, files.length);
            for (File fileEntry : files) {
                if (fileEntry.isFile())
                    System.out.println(fileEntry.getName());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void Sort(File[] files, int n) {
        int x, j, swaps;
        File temp;
        for (x = n - 2; x >= 0; x--) {
            swaps = 0;
            for (j = 0; j <= x; j++) {
                if (files[j].getName().compareTo(files[j +
1].getName()) > 0) {
                    temp = files[j];
                    files[j] = files[j + 1];
                    files[j + 1] = temp;
                    swaps++;
                }
            }
            if (swaps == 0)
                break;
        }
    }

    public void addFile(String fileName, String directoryPath) {
        try {
            File directory = new File(directoryPath);
            if (directory.isDirectory()) {
                File f = new File(directoryPath + "/" + fileName);
                if (!f.exists()) {
                    f.createNewFile();
                    System.out.println("File Added Successfully");
                }

                else {
                    System.out.println("File Already Existed");
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void deleteFile(String fileName, String directoryPath) {
        boolean isDeleted = false;
```

```
            boolean isExisted = false;
            try {
                  File f = new File(directoryPath);
                  File files[] = f.listFiles();
                  for (File fileEntry : files) {
                        if (fileEntry.getName().equals(fileName)) {
                              isExisted = true;
                              if (fileEntry.delete())
                                    isDeleted = true;
                        }
                  }
                  if (isExisted && isDeleted)
                        System.out.println("File Deleted Successfully");
                  else
                        System.out.println("File Is Not Existed To Delete");

            } catch (Exception e) {
                  e.printStackTrace();
            }
      }

      public void searchFile(String filename, String directorypath) {
            boolean isExisted = false;
            try {
                  File f = new File(directorypath);
                  File files[] = f.listFiles();
                  for (File fileEntry : files) {
                        if (fileEntry.isFile()) {
                              if (fileEntry.getName().equals(filename))
                                    isExisted = true;
                        }
                  }
                  if (isExisted)
                        System.out.println("File Found");
                  else
                        System.out.println("File Is Not Existed In The 
Directory");
            } catch (Exception e) {
                  e.printStackTrace();
            }

      }
}
```

```
//Main Method

package Com.lockedme;

import java.util.Scanner;

public class Driver {

    public static void main(String[] args) {
        final String ROOT_DIRECTORY_PATH = "D:\\Files";
        String fileName;
        LockedMeBO bo = new LockedMeBO();
        Scanner sc = new Scanner(System.in);
        System.out.println("Welcome To LockedMe.Com");
        System.out.println("Developer Information");
        System.out.println("Mr.Hanuman-Learner(Full Stack Java 
Developer-Cohort-2)");
        int ch, ch1;
        do {
            while (true) {
                System.out.println("1-List Files In Ascending 
Order");

                System.out.println("2-Edit");
                System.out.println("3-Exit");
                System.out.println("Enter Your Choice");
                ch = sc.nextInt();
                if (ch == 1 || ch == 2 || ch == 3)
                    break;
                System.err.println("Please Enter Correct Choice");
            }
            switch (ch) {
            case 1:

    bo.displayFilesInAscendingOrder(ROOT_DIRECTORY_PATH);
                break;
            case 2:
                do {
                    while (true) {
                        System.out.println("1-Add Files To The 
Specified Directory");

                        System.out.println("2-Delete Files From 
The Specified Directory");

                        System.out.println("3-Search a File From 
The Specified Directory");

                        System.out.println("4-Switch To Main 
Menu");

                        ch1 = sc.nextInt();
                        if (ch1 == 1 || ch1 == 2 || ch1 == 3 || 
ch1 == 4)

                            break;
                        System.err.println("Please Enter Correct 
Choice");

                    }
                    switch (ch1) {
                    case 1:
                        System.out.println("Enter The File Name 
To Add");

                        fileName = readFileName(sc);
```

```
                                    bo.addFile(fileName,
ROOT_DIRECTORY_PATH);
                                    break;
                              case 2:
                                    System.out.println("Enter The File Name
To Remove");
                                    fileName = readFileName(sc);
                                    bo.deleteFile(fileName,
ROOT_DIRECTORY_PATH);
                                    break;
                              case 3:
                                    System.out.println("Enter The File Name
To Search");
                                    fileName = readFileName(sc);
                                    bo.searchFile(fileName,
ROOT_DIRECTORY_PATH);
                                    break;
                              }
                        } while (ch1 <= 3);
                        break;
                  /*
                   * default: System.err.println("Please Enter Correct
Choice");
                   */
                  }
            } while (ch < 3);
      }

      private static String readFileName(Scanner sc) {
            sc.nextLine();
            String fileName = sc.nextLine();
            return fileName;
      }
}
```

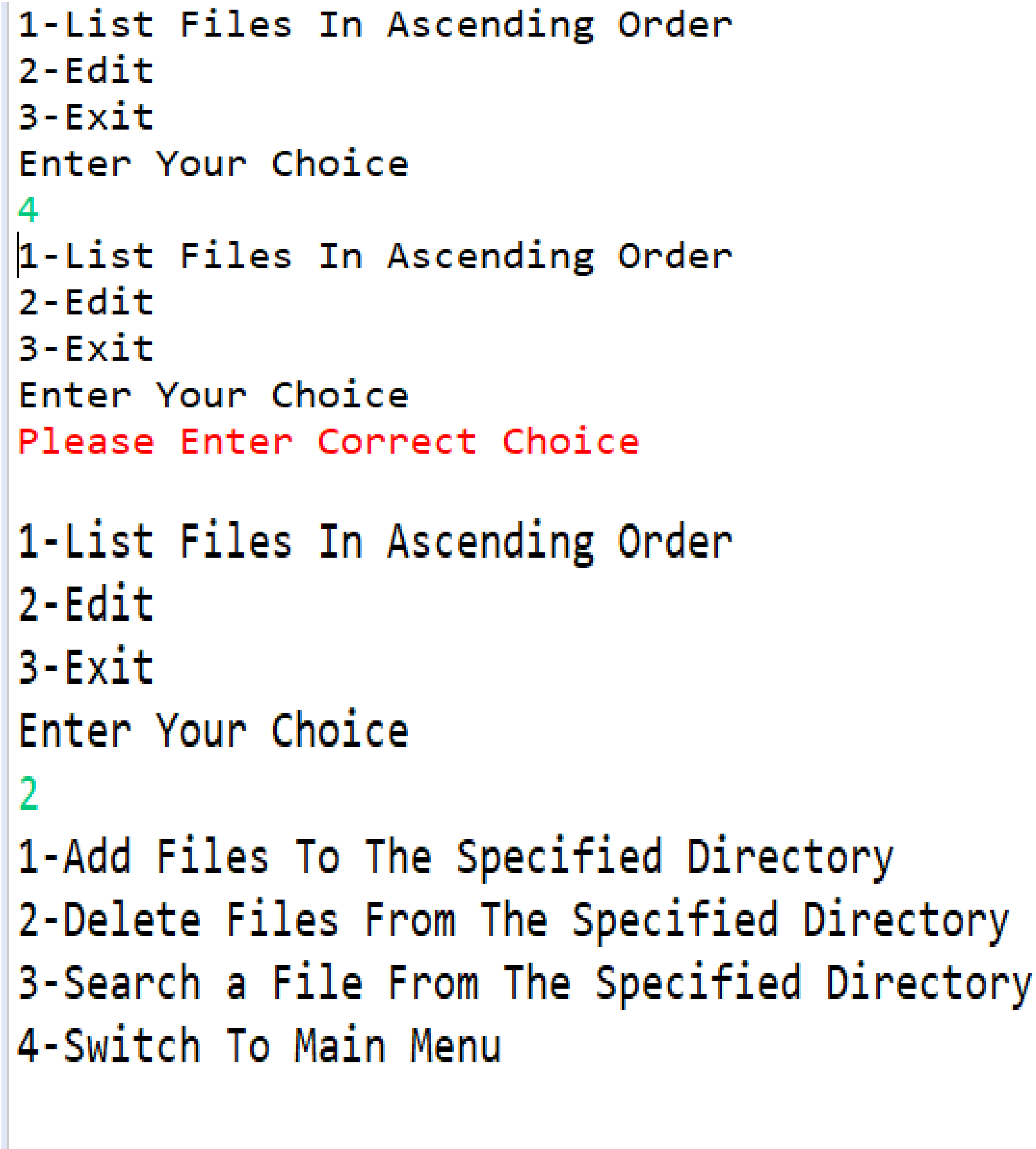
//Output shown on the Console after getting i/p from the User.
(1st display the welcome screen & then get the i/p from user)



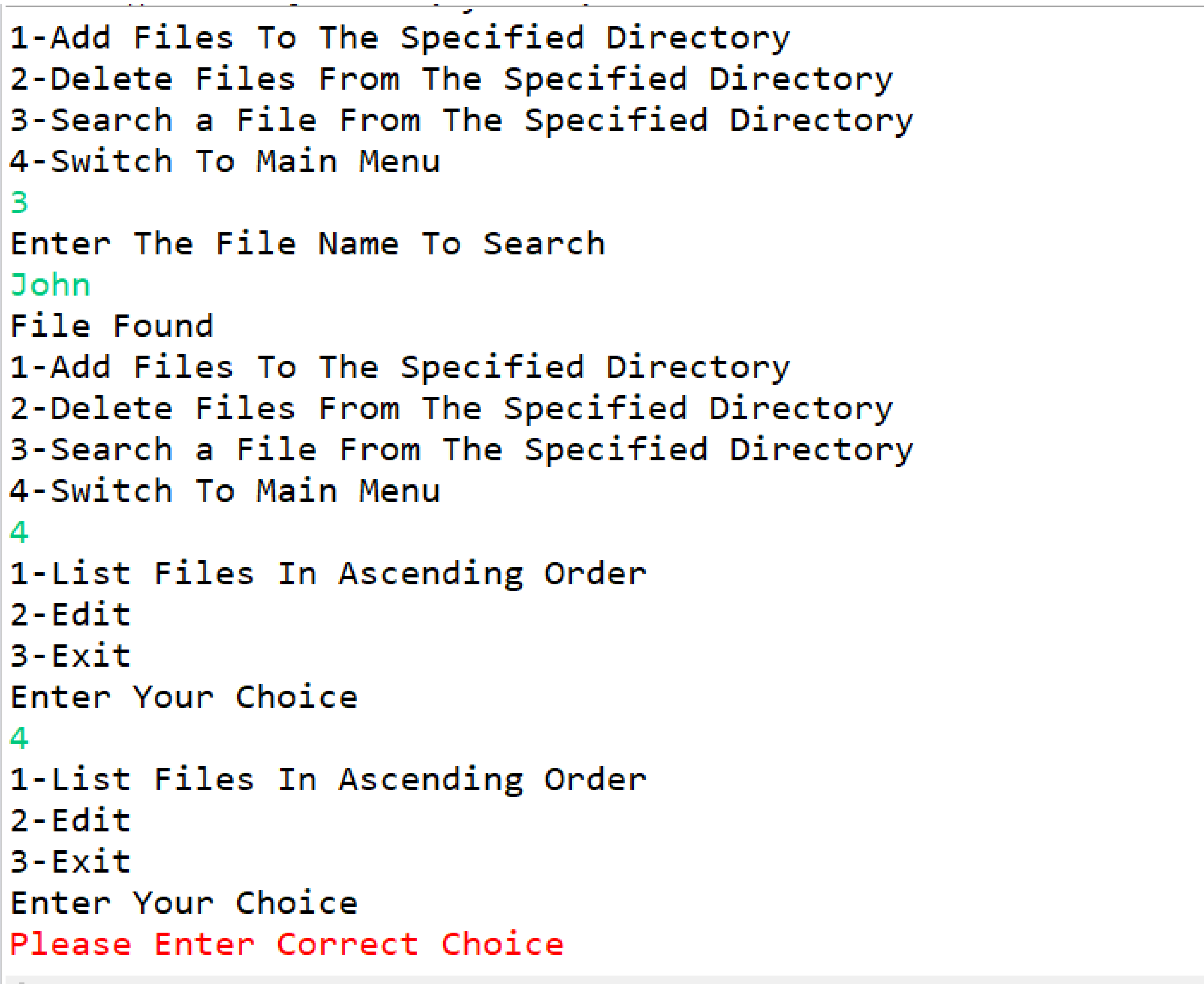
Driver [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (Oct 6, 2022, 9:15:16 PM)

```
Welcome To LockedMe.Com
Developer Information
Mr.Hanuman-Learner(Full Stack Java Developer-Cohort-2)
1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
1
Sachin
abc
airena.txt
airoplane.txt
apple.txt
banana.txt
bat.txt
hanuman.txt
home.txt
office.txt
orange.txt
ram.txt
red.txt
rose.txt
van
[illegible].txt
```

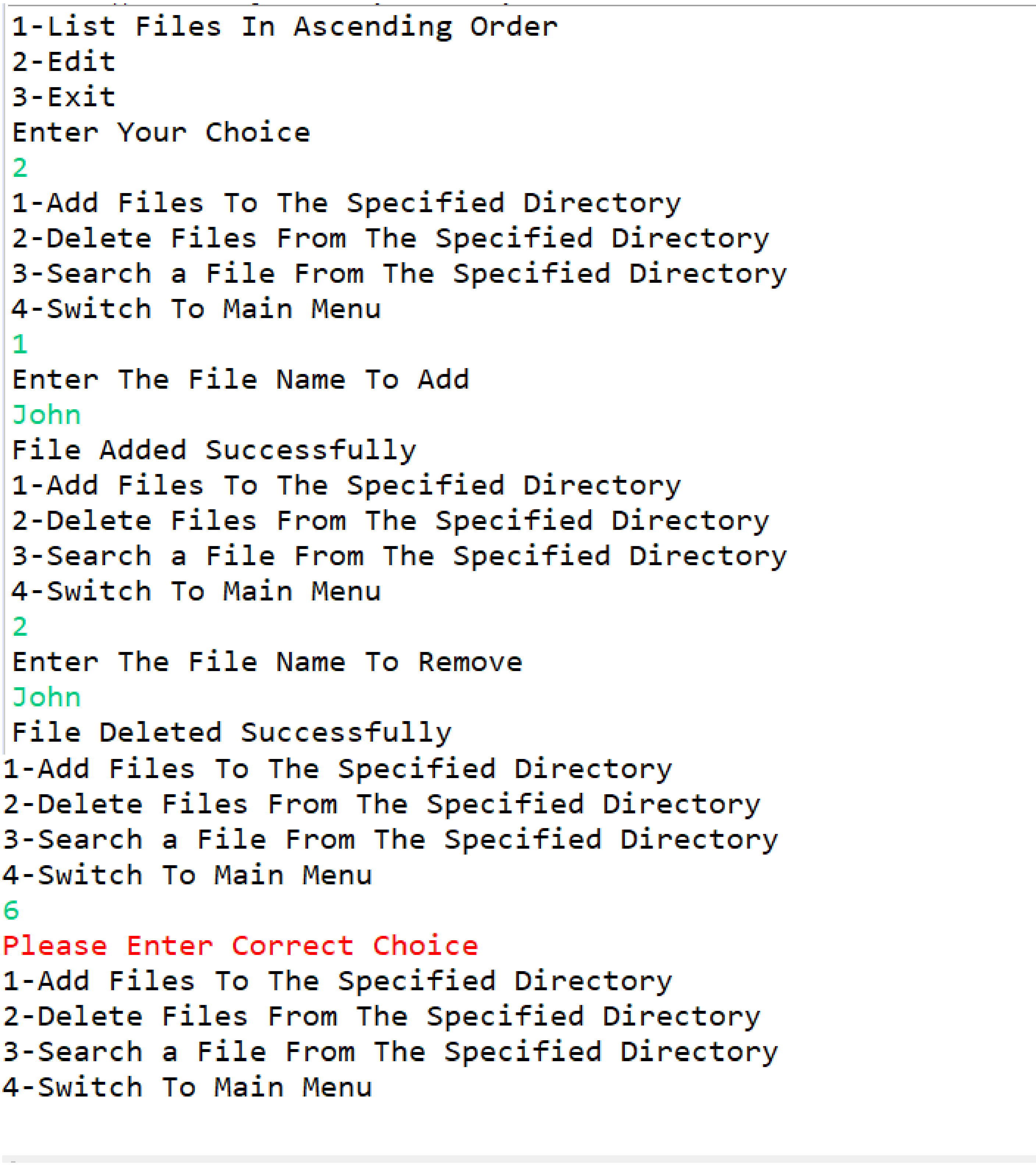
//Display the Message on console if user inputs the wrong choice.



```
1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
4
1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
Please Enter Correct Choice

1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
2
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
```



```
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
3
Enter The File Name To Search
John
File Found
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
4
1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
4
1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
Please Enter Correct Choice
```



```
1-List Files In Ascending Order
2-Edit
3-Exit
Enter Your Choice
2
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
1
Enter The File Name To Add
John
File Added Successfully
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
2
Enter The File Name To Remove
John
File Deleted Successfully
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
6
Please Enter Correct Choice
1-Add Files To The Specified Directory
2-Delete Files From The Specified Directory
3-Search a File From The Specified Directory
4-Switch To Main Menu
```

Core concepts used:

The following concepts are used in the project,

- File Management
- Exception Handling
- Scanner Class Object
- Regular Expression
- Bubble Sorting & Linear Search method

Link to Github:

https://github.com/hanumanprasadvishwa/Assignment-1

Unique selling points:

1. Application is simple & easy to handle.
2. It is user friendly.
3. Different operations like listing,inserting,Deleting & Searching of File in the specified directory can be done easily.
4. Application is easy to use and requires as minimal inputs as possible, yet providing great accessibility for all the necessary actions.

Conclusion:

Application is prepared according to required features. Also, considering tediousness for some of the asked features they are simplified in order to increase the productivity.