LockedMe.com

A prototype documentation.

by Harsh Dedhia

run LockedMeMain

https://github.com/dedhiah10/LockedMe.com/tree/master

1) Project Statement:

GitHub Repository: https://github.com/dedhiah10/LockedMe.com/tree/master

Create an application which uses command line interface for user interaction. Plan the development in terms of sprints and then push the source code to the GitHub repository. The program is developed using JAVA and eclipse IDE, it takes user input and adds, deletes, searches, sorts, displays files.

2) Sprints Planning:

A. Product Backlog:

i. Plan Algorithms.
  a. Algorithm for User Interface.
  b. File handling algorithm.
  c. Sorting and searching algorithm.
  d. File path changing method.

ii. Create navigation algorithm using **SwitchingAlgo** and **LockedMeMain** classes.
  a. Code **main menu**.
  b. Code **first menu**.
  c. Code **second menu**.
  d. Code **loops** and **conditional statements** for proper navigation.
  e. Code **scanner** and **print** statements.

iii. Create **FileHandler** class.
  a. Select file handling service.
  b. Code **addNewFile** method.
  c. Code **deleteFile** method.
  d. Code **searchFile** method.
  e. Code **changePath** method.
  f. Code **sortedFileOutput** method.
  g. Close all services before exit

iv. Test run the application.
  a. **Run** and **Debug** the application.

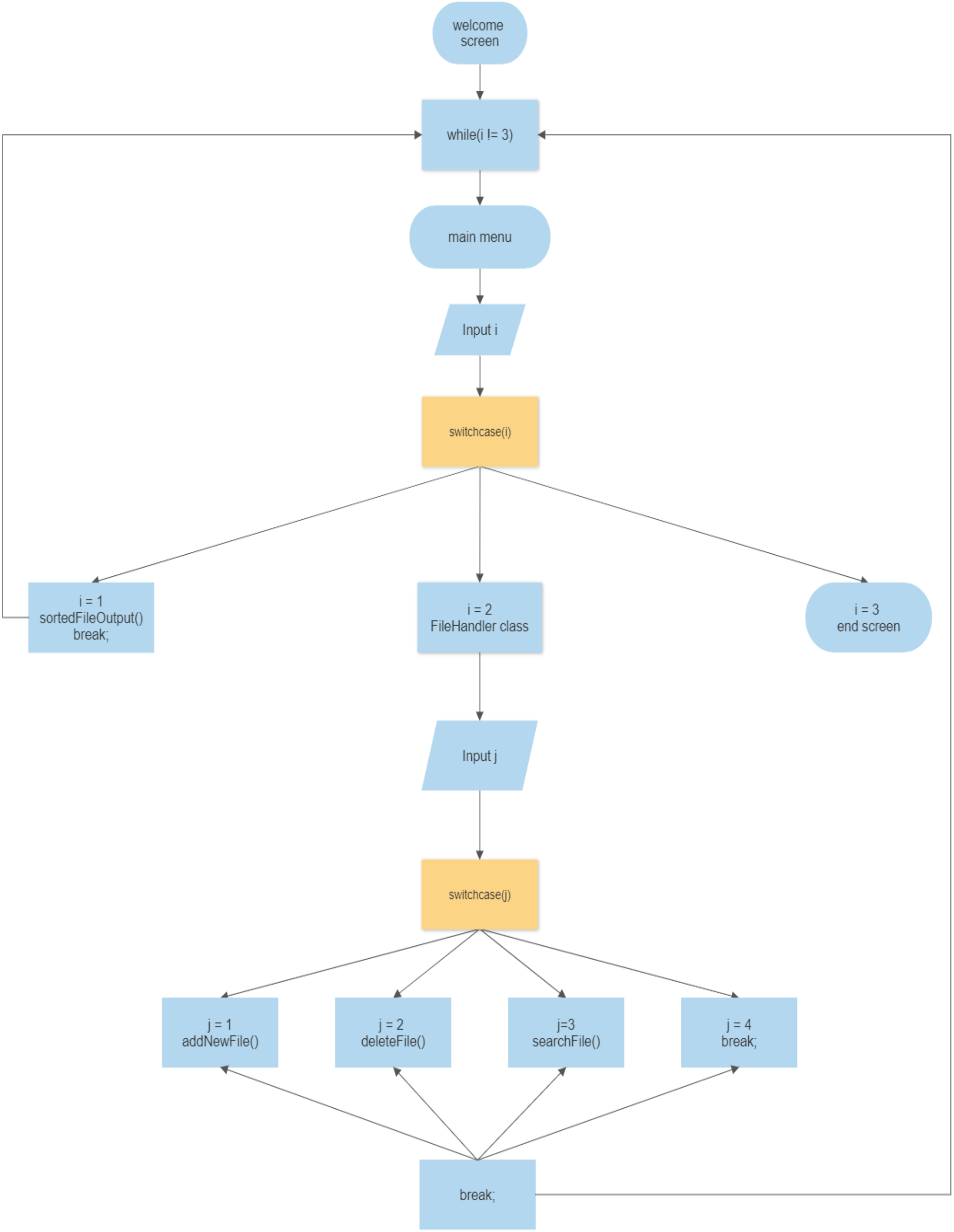
B. Sprint Table:

Sprint No.	Tasks	Estimation	Status
1.	**Plan** Algorithms.	4 hours	Pending
2.	Create **SwitchingAlgo, LockedMeMain** classes.	4 hours	Pending
3.	Create **FileHandler** class.	6 hours	Pending
4.	**Test** run the application.	4 hours	Pending

C. Sprints:

i. Plan Algorithms:

a. Algorithm for User Interface:



The given figure is an algorithm for the user interface implemented through console. The whole logic is surrounded by while loop that breaks only when option: “3. Exit application.” is selected and the program finishes executing. The application takes `int` input i and puts it through a switchcase. For the first option: “1. List all files in ascending order.” The sortedFileOputput method using a sorting algorithm kicks in and prints all the files in file path after sorting them. And for option: “2. add/delete/search files.” The FileHandler kicks in.

The FileHandler class asks for another `int` input j. This input is put through another switchcase. For values of j, options selected are: The first option: “1. Add a new File.” Which asks for a String input for the file name uses addNewFile method and creates the file if it doesn’t already exist. The second option: “2. Delete an existing file.” This asks for a String input for the file name then, utilizes deleteFile method to delete a file if it exists. The fourth option: “4. Go back to the Main Menu.” Breaks this switch case and continues to the main while loop so we end up in the main menu. The fifth option: “5. change filePath” is an option in development which uses changePath method by passing String input as argument and changing the directory where files are read and written to.

Now, after executing any one of these methods, the switchcase breaks and the control is shifted to the main while loop. Hence, we return to main menu. So the code now resets. And that is how we will code SwitchingAlgo and LockedMeMain class.

b. File Handling algorithm:

In this, we use the java File class by using `import java.io.File;` It has many methods that are useful to us in file operations of add, delete and search. We will use String path and concatenate it using ‘+’ and then use it as argument to create File class instance. After creating the instance, we can use the addNewFile method or the deleteFile method using “.createNewFile(path+name)” or “.delete(path+name)”. The String name is user input after selecting a method addNewFile or deleteFile.

Now, for searching a file from the path, we use “.list(path)” so we get String[] return type. Then we can use Iterator or for loop, but since searching is through ever file, we use for-each loop for convenience. Now, for each String: String[], we use “.contains(name)” method where name is taken as user input. And print the values using printstream class by using if statement of “.contains(name)”. This gives us the three operations of add, delete and search. And that is how we will code FileHandler class.

c. Sorting and searching Algorithm:

The list of files is acquired using the ".list(path)" this gives us a return type of String[]. A sorting algorithm is now required to sort the String[]. We use bubble sort technique on the unsorted String[] array and swap neighboring array entries until the list is sorted. This sorting is done using the compareTo method , if statement and a String temp. After bubble sorting the list is now sorted and is printed using a simple for-each method.

The option: "3. Search for a file." asks for a String input for the file name, further invoking searchFile method and returns all the search results. TreeSet is a type of sorted collection, It sorts data as they are added. So when we do a ".toString()" of each String Object in TreeSet, we get a pre-sorted list. Why? Due to the properties of TreeSet, which is, it sorts the added Strings as we add it. Hence the sorting algorithm we use is TreeSet's internal sorting.

d. File path changing method:

The extra feature of changePath method. We use simple setter method and change the directory where all the operations take place. Now the ".createNewFile(path+name)", ".delete(path+name)" and the ".list(path)" use the new path set by the user. And hence we achieve the path change.

Sprint No.	Tasks	Estimation	Status
1.	**Plan** Algorithms.	4 hours	Done
2.	Create **SwitchingAlgo, LockedMeMain** classes.	4 hours	Pending
3.	Create **FileHandler** class.	6 hours	Pending
4.	**Test** run the application.	4 hours	Pending

ii. Create navigation algorithm using SwitchingAlgo and LockedMeMain classes:

   a. Code main menu:
   b. Code first menu:
   c. Code second menu:
   d. Code loops and conditional statements for proper navigation:
   e. Code scanner and print statements:

The coding for class SwitchingAlgo and LockedMain is done in Eclipse IDE using jdk 16 and JSE 1.8, the codes are uploaded to GitHub. GitHub link is at the start and end of document and on footer of everypage.

iii. Create FileHandler class:

   a. Select file handling service:
   b. Code addNewFile method:
   c. Code deleteFile method:
   d. Code searchFile method:
   e. Code changePath method:
   f. Code sortedFileOutput method:
   g. Close all services before exit:

The coding for class FileHandler is done in Eclipse IDE using jdk 16 and JSE 1.8, the codes are uploaded to GitHub. GitHub link is at the start and end of document and footer of everypage.

Sprint No.	Tasks	Estimation	Status
1.	**Plan** Algorithms.	4 hours	Done
2.	Create **SwitchingAlgo, LockedMeMain** classes.	4 hours	Done
3.	Create **FileHandler** class.	6 hours	Done
4.	**Test** run the application.	4 hours	Pending

iv. Test run the application:

a. Run and Debug the application:

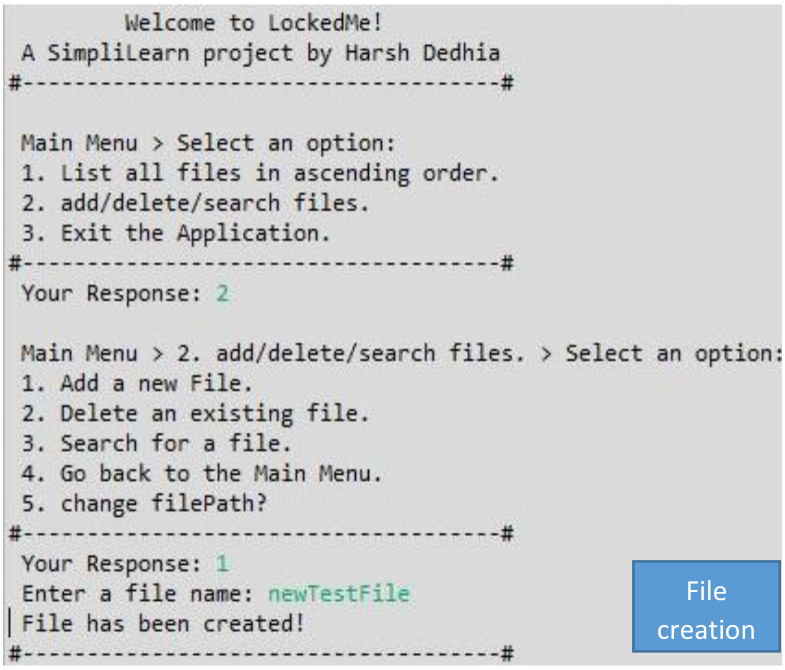
The application was run and debugged several times. It was tweaked until all the grammatical errors and logical errors were sorted. A new feature was added where the application would reprint the entire menu options list if you entered wrong value twice. This was done for convenience as the console tends to get messy.

Sprint No.	Tasks	Estimation	Status
1.	**Plan** Algorithms.	4 hours	Done
2.	Create **SwitchingAlgo, LockedMeMain** classes.	4 hours	Done
3.	Create **FileHandler** class.	6 hours	Done
4.	**Test** run the application.	4 hours	Done

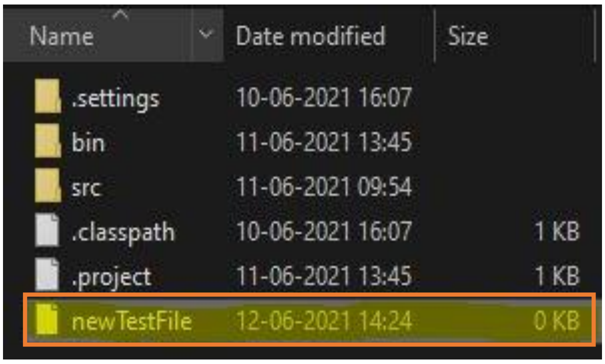
3) Working of the application:

A. Adding a file:

Adding a file in the directory with user input for file name.



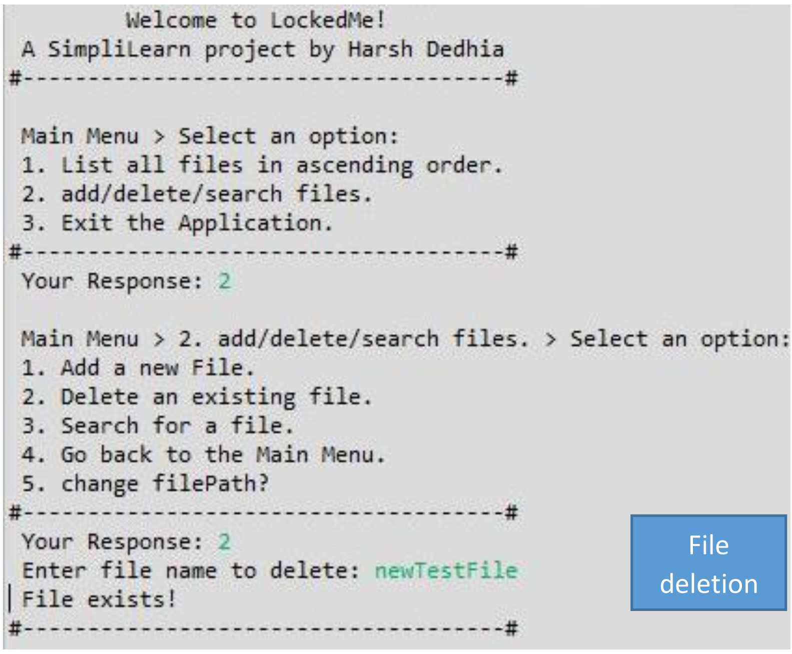
```
         Welcome to LockedMe!
 A SimpliLearn project by Harsh Dedhia
#-------------------------------------#

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#-------------------------------------#
 Your Response: 2

 Main Menu > 2. add/delete/search files. > Select an option:
 1. Add a new File.
 2. Delete an existing file.
 3. Search for a file.
 4. Go back to the Main Menu.
 5. change filePath?
#-------------------------------------#
 Your Response: 1
 Enter a file name: newTestFile
 File has been created!
#-------------------------------------#
```



Name	Date modified	Size
.settings	10-06-2021 16:07	
bin	11-06-2021 13:45	
src	11-06-2021 09:54	
.classpath	10-06-2021 16:07	1 KB
.project	11-06-2021 13:45	1 KB
newTestFile	12-06-2021 14:24	0 KB

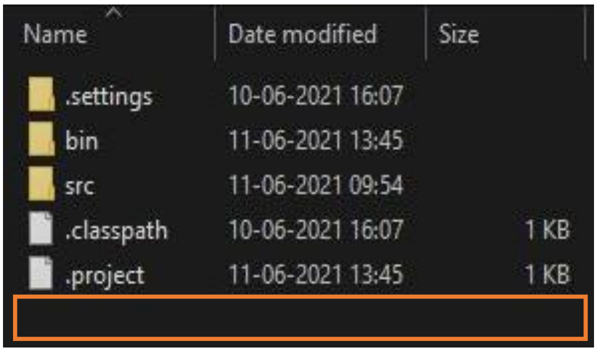
B. Deleting a file:

Deleting a file in the directory with user input for file name.



```
        Welcome to LockedMe!
 A SimpliLearn project by Harsh Dedhia
#-------------------------------------#

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#-------------------------------------#
 Your Response: 2

 Main Menu > 2. add/delete/search files. > Select an option:
 1. Add a new File.
 2. Delete an existing file.
 3. Search for a file.
 4. Go back to the Main Menu.
 5. change filePath?
#-------------------------------------#
 Your Response: 2
 Enter file name to delete: newTestFile
 File exists!
#-------------------------------------#
```

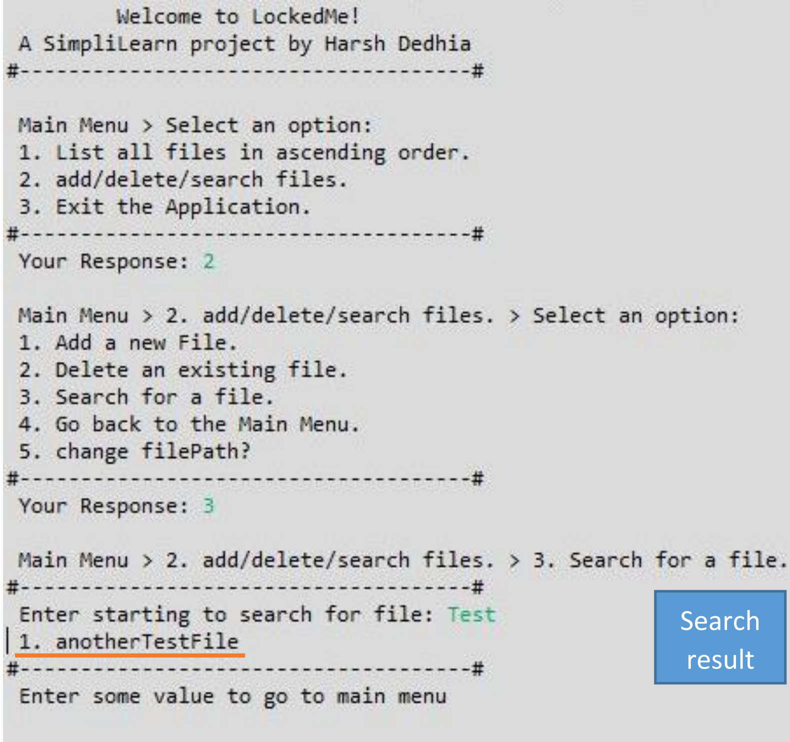
File deletion



Name	Date modified	Size
.settings	10-06-2021 16:07	
bin	11-06-2021 13:45	
src	11-06-2021 09:54	
.classpath	10-06-2021 16:07	1 KB
.project	11-06-2021 13:45	1 KB

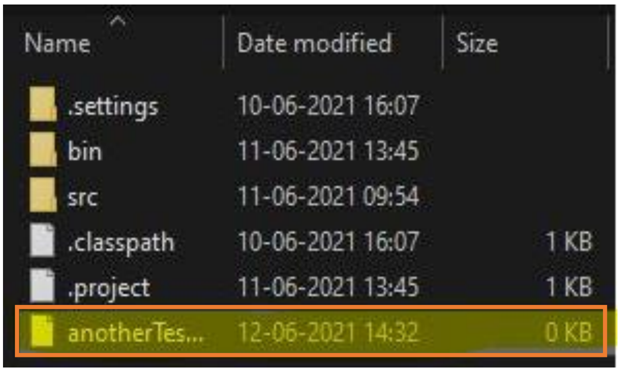
C. Searching a file:

Searching a file in the directory with user input for search criteria.



```
        Welcome to LockedMe!
 A SimpliLearn project by Harsh Dedhia
#-------------------------------------#

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#-------------------------------------#
 Your Response: 2

 Main Menu > 2. add/delete/search files. > Select an option:
 1. Add a new File.
 2. Delete an existing file.
 3. Search for a file.
 4. Go back to the Main Menu.
 5. change filePath?
#-------------------------------------#
 Your Response: 3

 Main Menu > 2. add/delete/search files. > 3. Search for a file.
#-------------------------------------#
 Enter starting to search for file: Test
 1. anotherTestFile
#-------------------------------------#
 Enter some value to go to main menu
```

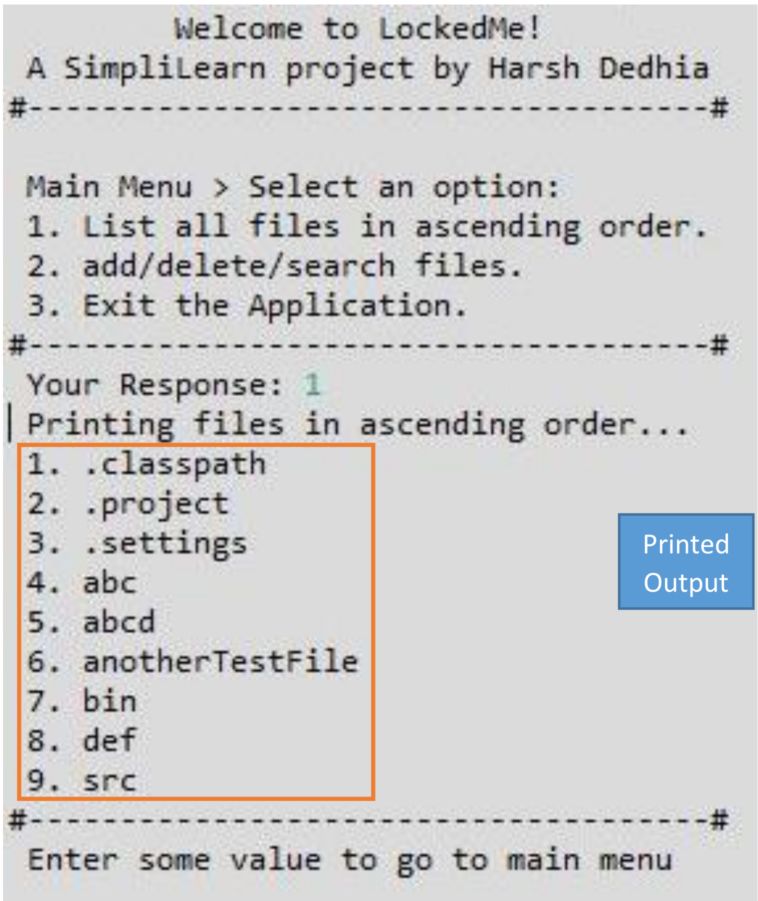
Search result



Name	Date modified	Size
.settings	10-06-2021 16:07	
bin	11-06-2021 13:45	
src	11-06-2021 09:54	
.classpath	10-06-2021 16:07	1 KB
.project	11-06-2021 13:45	1 KB
anotherTes...	12-06-2021 14:32	0 KB

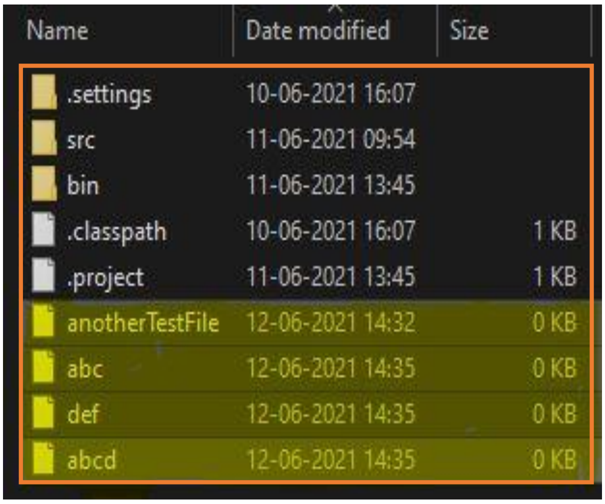
D. Printing sorted files:

Printing all files in the directory after sorting in an ascending order.



```
        Welcome to LockedMe!
 A SimpliLearn project by Harsh Dedhia
#-----------------------------------#

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#-----------------------------------#
 Your Response: 1
 Printing files in ascending order...
 1. .classpath
 2. .project
 3. .settings
 4. abc
 5. abcd
 6. anotherTestFile
 7. bin
 8. def
 9. src
#-----------------------------------#
 Enter some value to go to main menu
```

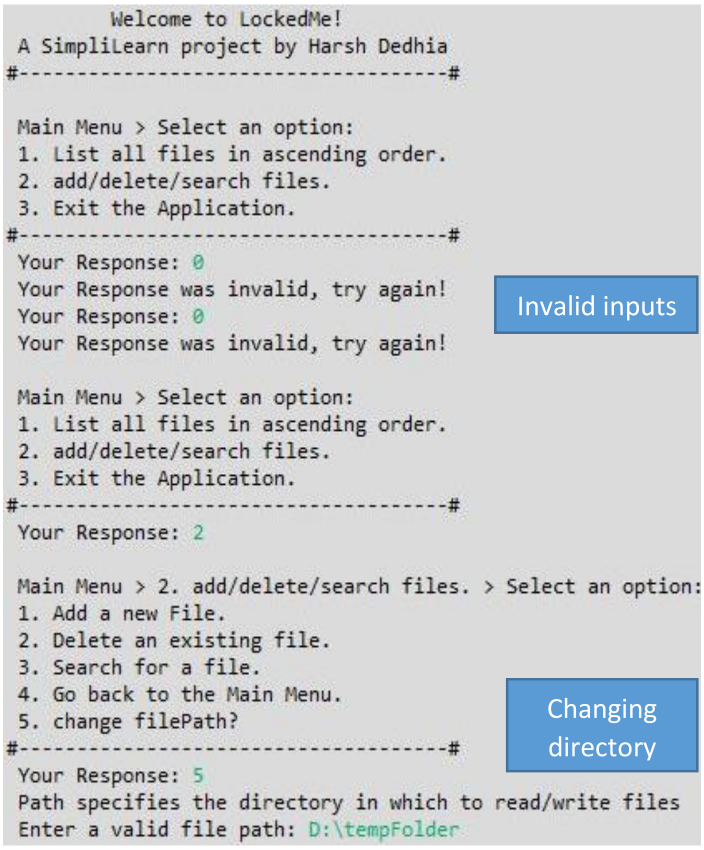
Printed Output



Name	Date modified	Size
.settings	10-06-2021 16:07	
src	11-06-2021 09:54	
bin	11-06-2021 13:45	
.classpath	10-06-2021 16:07	1 KB
.project	11-06-2021 13:45	1 KB
anotherTestFile	12-06-2021 14:32	0 KB
abc	12-06-2021 14:35	0 KB
def	12-06-2021 14:35	0 KB
abcd	12-06-2021 14:35	0 KB

E. Complex operation:

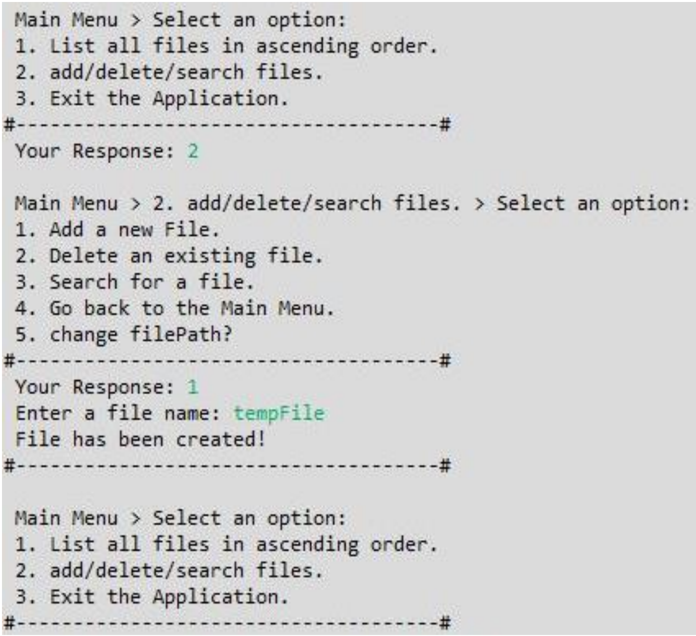
Trying out wrong inputs so the application does not crash unexpectedly, also file directory is taken with user input and file operations are done there.



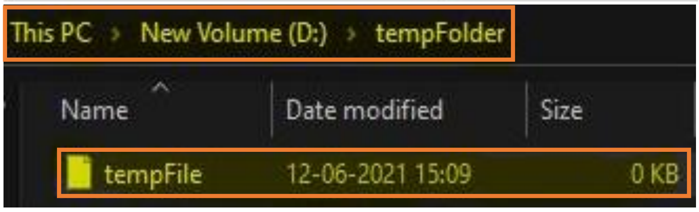
```
        Welcome to LockedMe!
 A SimpliLearn project by Harsh Dedhia
#------------------------------------#

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#------------------------------------#
 Your Response: 0
 Your Response was invalid, try again!
 Your Response: 0
 Your Response was invalid, try again!

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#------------------------------------#
 Your Response: 2

 Main Menu > 2. add/delete/search files. > Select an option:
 1. Add a new File.
 2. Delete an existing file.
 3. Search for a file.
 4. Go back to the Main Menu.
 5. change filePath?
#------------------------------------#
 Your Response: 5
 Path specifies the directory in which to read/write files
 Enter a valid file path: D:\tempFolder
```

Invalid inputs

Changing directory



```
 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#------------------------------------#
 Your Response: 2

 Main Menu > 2. add/delete/search files. > Select an option:
 1. Add a new File.
 2. Delete an existing file.
 3. Search for a file.
 4. Go back to the Main Menu.
 5. change filePath?
#------------------------------------#
 Your Response: 1
 Enter a file name: tempFile
 File has been created!
#------------------------------------#

 Main Menu > Select an option:
 1. List all files in ascending order.
 2. add/delete/search files.
 3. Exit the Application.
#------------------------------------#
```



This PC › New Volume (D:) › tempFolder

Name	Date modified	Size
tempFile	12-06-2021 15:09	0 KB

These screenshots of working application are also uploaded.

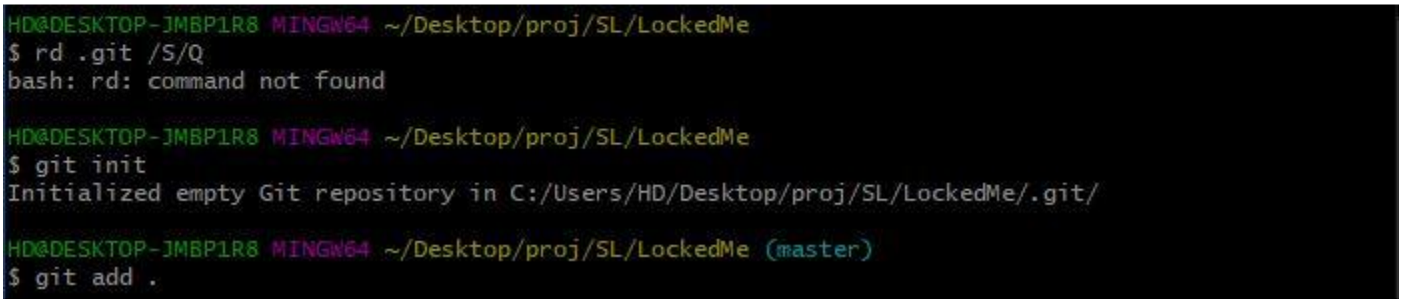
4) Git and GitHub:

A. Setting up git and github:

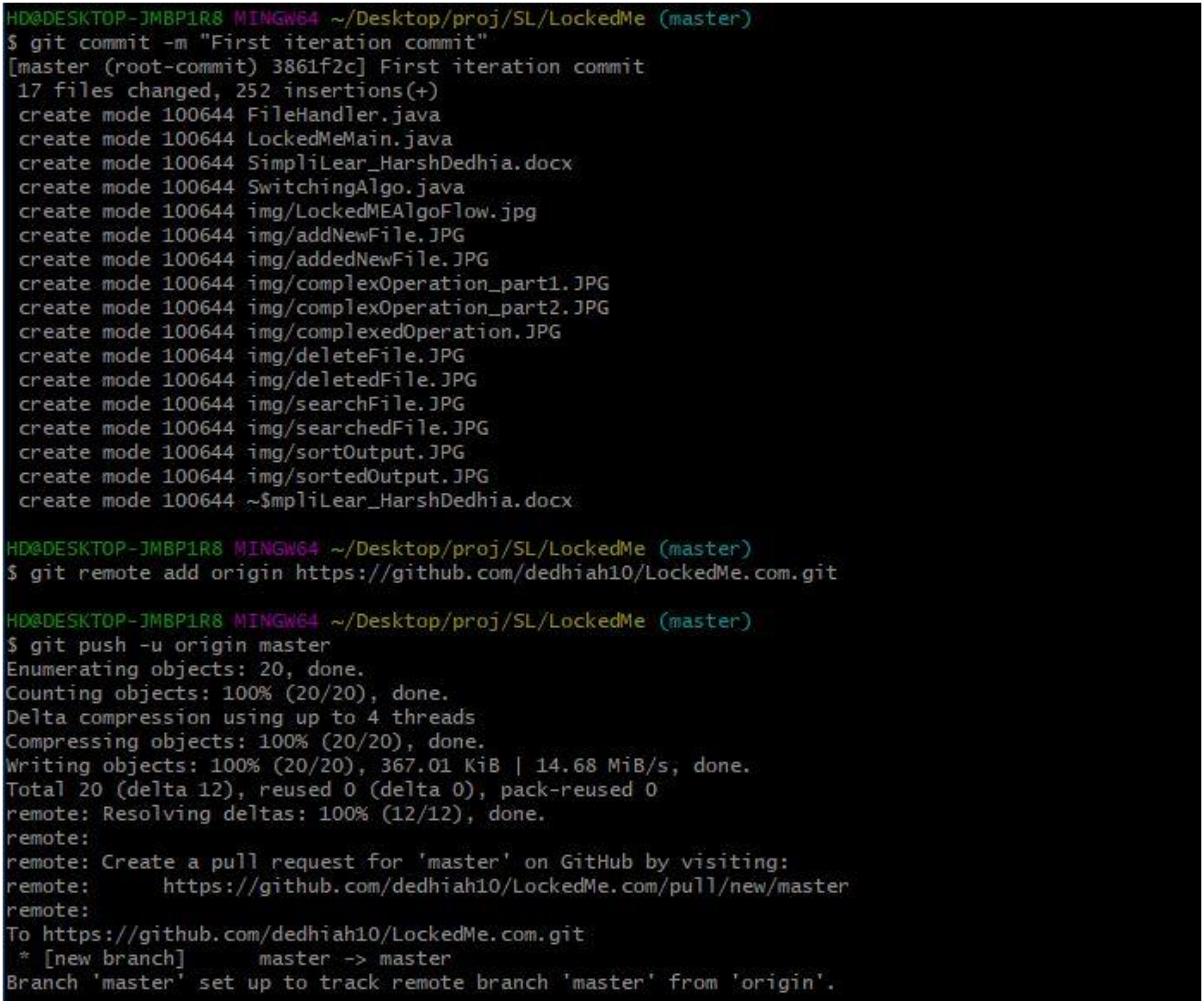
We set up git on PC, we make a git repository on GitHub.

B. Pushing to a git repository:

Now we link git on PC and git repository on git hub using git bash. Then we use git commit and git push commands to upload to repository and give a commit message.

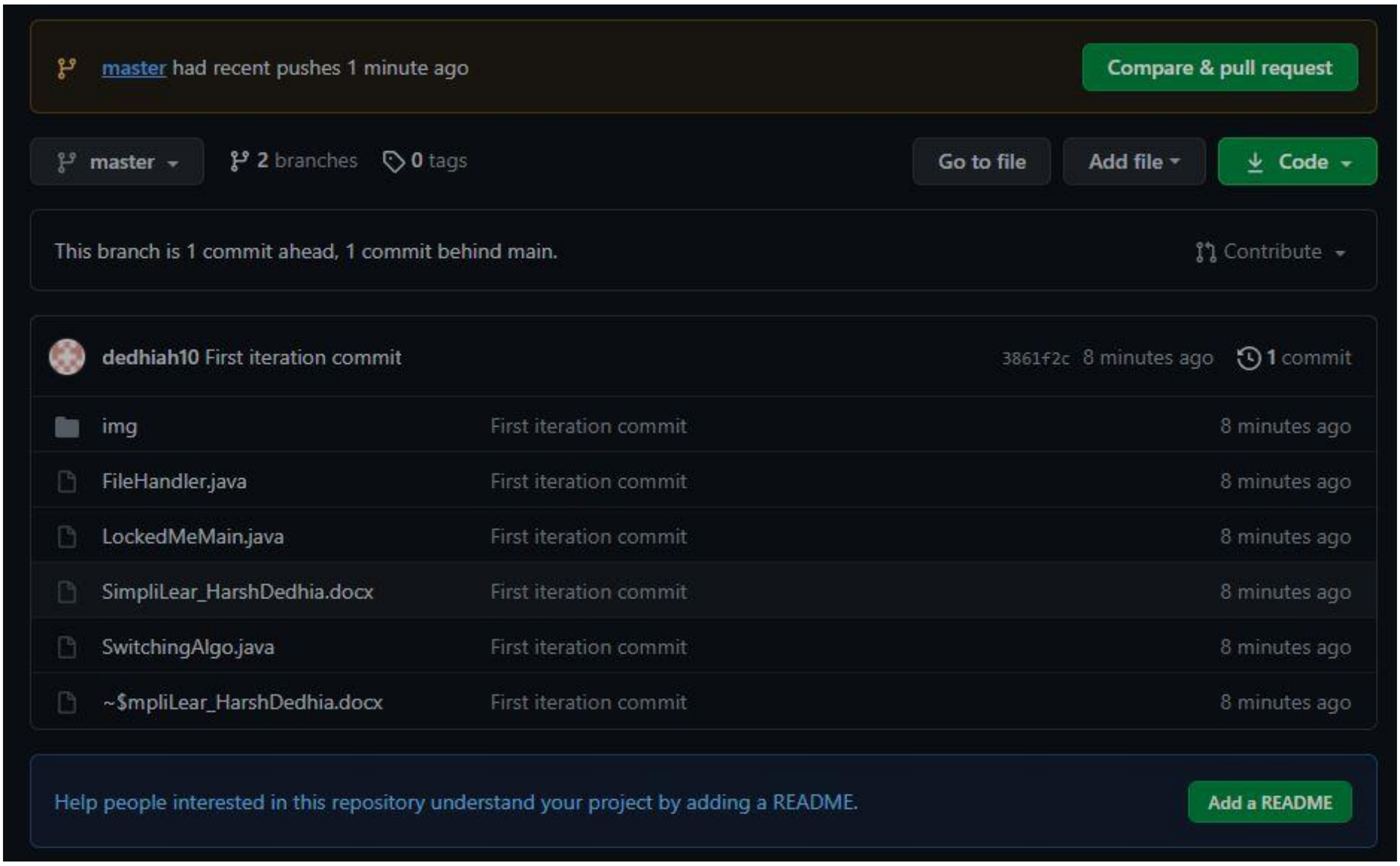


```
HD@DESKTOP-JMBP1R8 MINGW64 ~/Desktop/proj/SL/LockedMe
$ rd .git /S/Q
bash: rd: command not found

HD@DESKTOP-JMBP1R8 MINGW64 ~/Desktop/proj/SL/LockedMe
$ git init
Initialized empty Git repository in C:/Users/HD/Desktop/proj/SL/LockedMe/.git/

HD@DESKTOP-JMBP1R8 MINGW64 ~/Desktop/proj/SL/LockedMe (master)
$ git add .
```



```
HD@DESKTOP-JMBP1R8 MINGW64 ~/Desktop/proj/SL/LockedMe (master)
$ git commit -m "First iteration commit"
[master (root-commit) 3861f2c] First iteration commit
 17 files changed, 252 insertions(+)
 create mode 100644 FileHandler.java
 create mode 100644 LockedMeMain.java
 create mode 100644 SimpliLear_HarshDedhia.docx
 create mode 100644 SwitchingAlgo.java
 create mode 100644 img/LockedMEAlgoFlow.jpg
 create mode 100644 img/addNewFile.JPG
 create mode 100644 img/addedNewFile.JPG
 create mode 100644 img/complexOperation_part1.JPG
 create mode 100644 img/complexOperation_part2.JPG
 create mode 100644 img/complexedOperation.JPG
 create mode 100644 img/deleteFile.JPG
 create mode 100644 img/deletedFile.JPG
 create mode 100644 img/searchFile.JPG
 create mode 100644 img/searchedFile.JPG
 create mode 100644 img/sortOutput.JPG
 create mode 100644 img/sortedOutput.JPG
 create mode 100644 ~$mpliLear_HarshDedhia.docx

HD@DESKTOP-JMBP1R8 MINGW64 ~/Desktop/proj/SL/LockedMe (master)
$ git remote add origin https://github.com/dedhiah10/LockedMe.com.git

HD@DESKTOP-JMBP1R8 MINGW64 ~/Desktop/proj/SL/LockedMe (master)
$ git push -u origin master
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 4 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (20/20), 367.01 KiB | 14.68 MiB/s, done.
Total 20 (delta 12), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (12/12), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/dedhiah10/LockedMe.com/pull/new/master
remote:
To https://github.com/dedhiah10/LockedMe.com.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

GitHub Repository: https://github.com/dedhiah10/LockedMe.com/tree/master



master had recent pushes 1 minute ago

This branch is 1 commit ahead, 1 commit behind main.

dedhiah10 First iteration commit 3861f2c 8 minutes ago 1 commit

img	First iteration commit	8 minutes ago
FileHandler.java	First iteration commit	8 minutes ago
LockedMeMain.java	First iteration commit	8 minutes ago
SimpliLear_HarshDedhia.docx	First iteration commit	8 minutes ago
SwitchingAlgo.java	First iteration commit	8 minutes ago
~$mpliLear_HarshDedhia.docx	First iteration commit	8 minutes ago

Help people interested in this repository understand your project by adding a README.

The end

GitHub Repository: https://github.com/dedhiah10/LockedMe.com/tree/master