Project Name

LockedMe.com: Virtual Key for Your Repositories

Descriptions: Project Includes the Functionality Like

- Display Existing files from directory in sorting order.
- List of file operations
  - File Creation
  - Get Files by Name
  - Update Files
  - Delete File
  - Exit from the current context
  - Close the application

Generic features and three operations:

- Retrieving the file names in an ascending order

Business-level operations:

- Option to add a user specified file to the application
- Option to delete a user specified file from the application
- Option to search a user specified file from the application
- Navigation option to close the current execution context and return to the main context
- Option to close the application

Java concepts used in the project

Java Core concepts Like

- Class and Object.
- Java Class Methods (Static and Instance)
- Java Variables (Static and Instance)
- Java Class Attributes
- Java Arrays
- Java Operators
- Java Condition class
- Java Packages

- String Concepts.
- Scanner Class
- Oops Pillars:
    - Polymorphisms
    - Encapsulations
    - Interfaces
- Data Structure and Collection Interfaces:
- List Interface and Array List Class
- File Concepts:
    - File class
    - FileInputStreams
    - BufferdInputStream
    - FileWriter
    - BufferWrited
- Exception Handling Class

Project Requirements:

- An IDE's Used For coding: Eclipse
- Programing Language: core Java
- Version Control system: Git and GitHub
- For Sorting and Searching: List and Array List used

Git and GitHub Commands used for Code Push:

Step 1: git status

Step 2: git add .

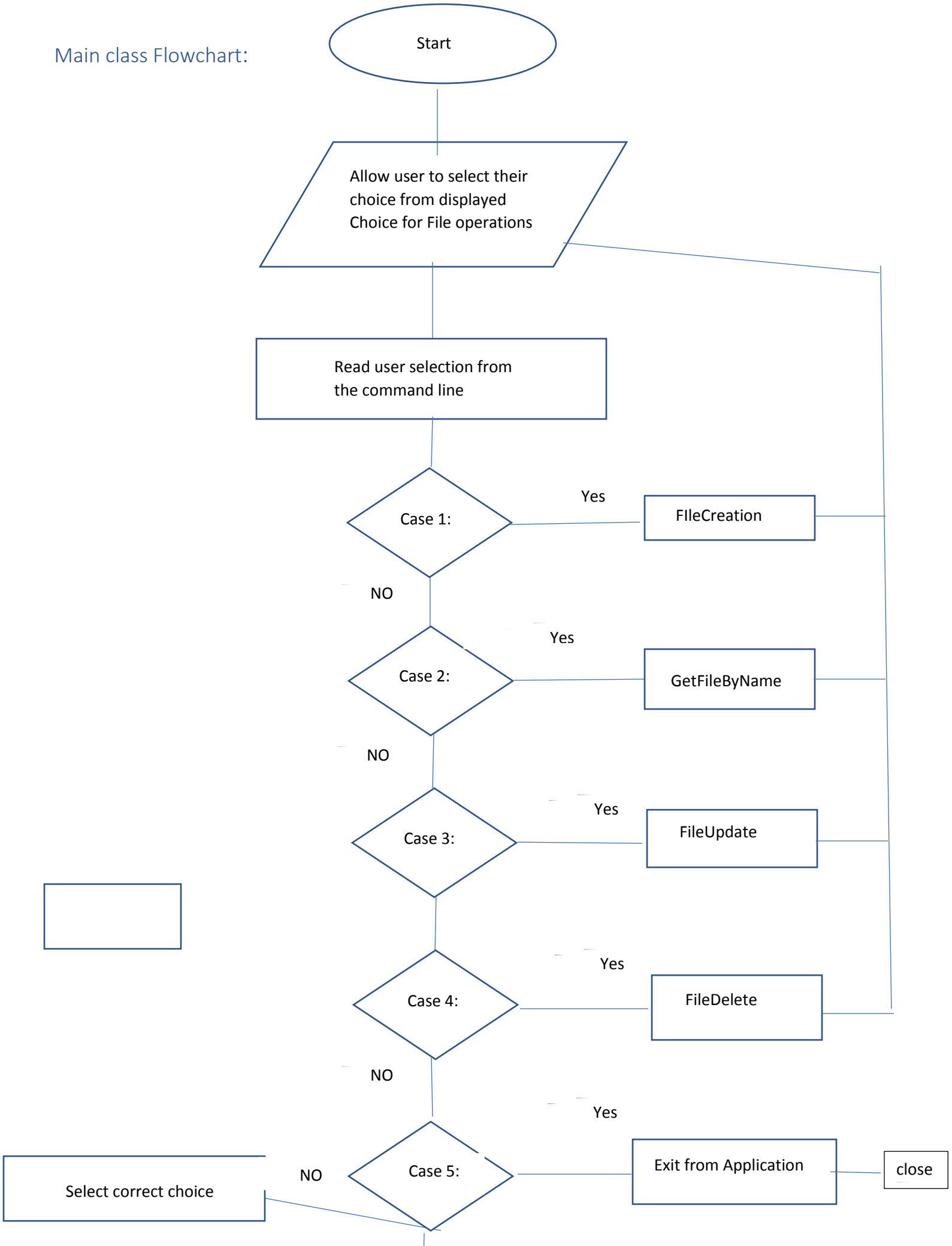
Step 3: git commit -m "commit message"

Step 4: git push origin master

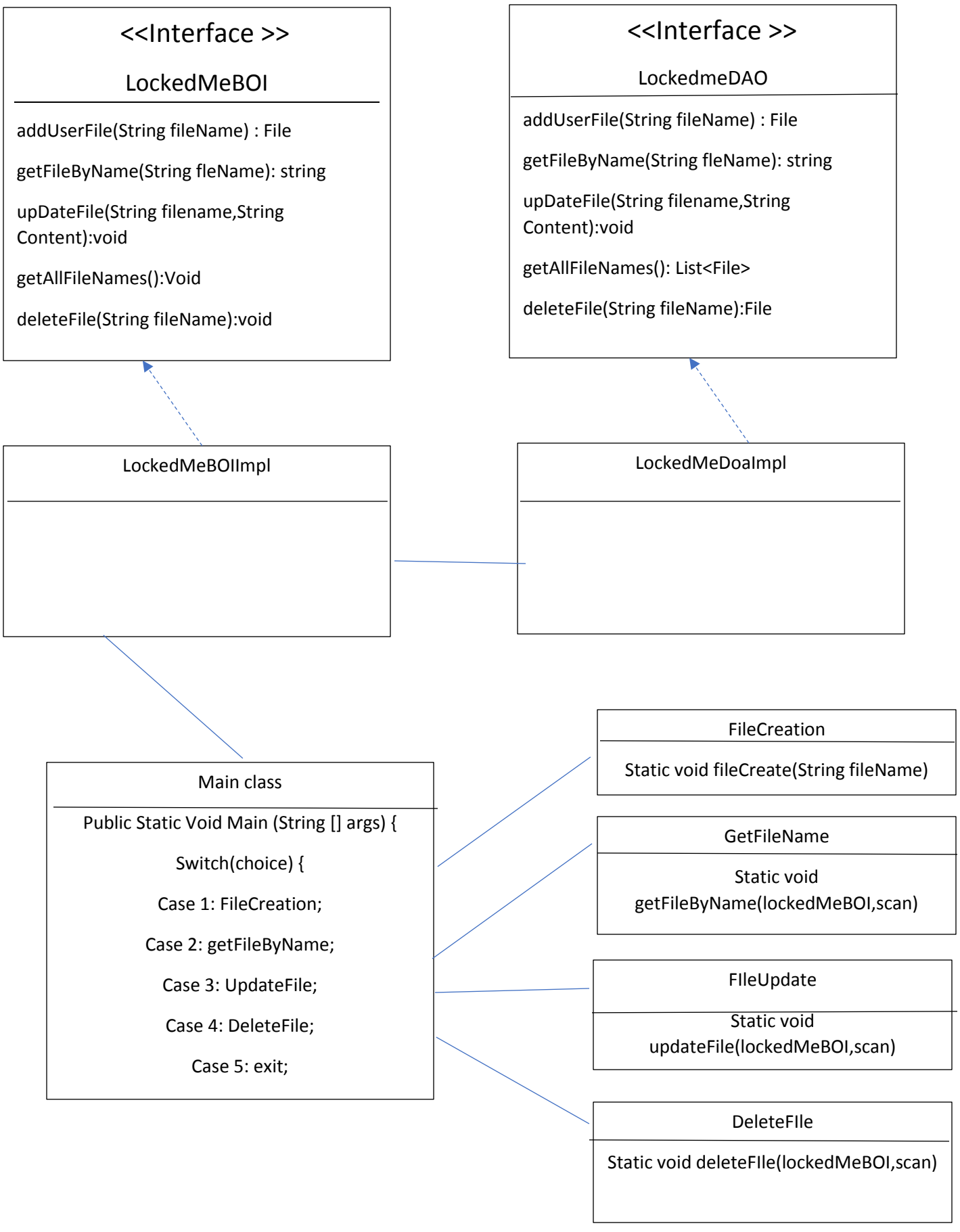
My Project Code you can find in below path:

>>git@github.com:kiranc2210/virtualFileManagementApp.git

Main class Flowchart:



LockedMeDotCom UML Diagram:



Algorithm for LockedMe Application

Step 1: Display All File name in Ascending order

Step 2; Display Choices for user.

Choice 1: Add a file to the existing directory with File Extension

Choice 2. Get File by providing the Filename

Choice 3. Update the File by Entering filename and Contents of file

Choice 4. Delete the file by filename whit Extension

Choice 5. exit context

Step 3: if User Selects a Choice 1 then

Enter the File name

If filename <3 then

System.out.println("File name should be at least 3 character long")

else

Enter File name

if File Exit with File name then:

System .out.println("File already exist with filename ")

Else

Create a file with filename

Return file

Step 4: if user select choice 2 then

Enter File name to get the file

If file exist with filename in directory

Display: file name to in console

else

throws file Does not found exception in the console

step 5: if user selects choice 3 then

enter the file and Contents to be update to the file

if file exist then:

update he file with content

else

throws file not found exception6

step 6: if user choice is 4 then

display: enter File name to be deleted from Directory

if file exist in a directory then:

Display file name and display, are you sure want to delete file?

If yes, the

File will be deleted

else

return to main context

else

throws file not found exception

step 7: user choice is 5 then exit the Application

Application exit from the execution.

LockedMe Application Souce Codes:

Main.java

```
/**
 *
 */
package com.LockedMeDotCom.presentationView;

import java.io.IOException;

/**
 *
 */

import java.util.Scanner;

import com.LockedMeDotCom.bussinessObject.implimentations.LockedMeBoImpl;
import com.LockedMeDotCom.bussinessob.LockedMeBOI;

/**
 * @author kiran
 *
```

```
 */
public class Main {

	public static void main(String[] args) {
		System.out.println("\t\t\t\tWelcome to LockedMe.com V1.3");
		System.out.println();

		System.out.print("ProjectName" + " " + ":" + "   " + "LOCKEDME.COM\t\t\t");
		System.out.println("\t\tAuthorName" + "  " + ":" + "   " + "Kiran Kumar C");
		System.out.println();

		LockedMeBOI lockedMeBOI;


			lockedMeBOI = new LockedMeBoImpl();
			System.out.println("\n\n");

			System.out.println("****File Available in the Current Directory****\n");
			try {
				lockedMeBOI.getAllFileNames();
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}

			System.out.println();

			Scanner scan = new Scanner(System.in);

			int choice = 0;

			System.out.println("Main Manu operations");
			System.out.println("**********************\n");
			do {
				System.out.println("1. Add a file to the existing directory with File
Extension\r\n");
				System.out.println("2. Get File by providing the FileName\r\n");
				System.out.println("3. Update the File by Entering fileName and Contents of
file\r\n");
				System.out.println("4. Delete the file by fileName whit Extension \r\n");
				System.out.println("5. exite context\r\n");
				System.out.println("Enter your choice");

				try {

					choice = Integer.parseInt(scan.nextLine());

				} catch (NumberFormatException e) {

				}
```

```
                        switch (choice) {

                        case 1:
                                FileCreation.fileCreate(lockedMeBOI, scan);
                                break;

                        case 2:
                                GetFileName.getFileByName(lockedMeBOI, scan);
                                break;
                        case 3:
                                FileUpdate.updateFile(lockedMeBOI, scan);
                                break;
                        case 4:

                                FileDelete.deleteFile(lockedMeBOI, scan);
                                break;
                        case 5:
                                System.out.println("Exit from the Application");
                                System.exit(0);
                                break;
                        default:
                                System.out.println("****File Available in the Current Directory****");
                                try {
                                        lockedMeBOI.getAllFileNames();
                                } catch (Exception e) {
                                        // TODO Auto-generated catch block
                                        e.printStackTrace();
                                }
                                System.out.println("Entered choice is invalid. Choice should be
between 1 to 6");
                                break;
                        }

                        try {
                                Thread.sleep(3000);
                        } catch (InterruptedException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }

                } while (choice != 5);
                scan.close();

        }

}
```

FileCreation.java

```
/**
 *
 */
package com.LockedMeDotCom.presentationView;

import java.util.Scanner;

import com.LockedMeDotCom.bussinessob.LockedMeBOI;

/**
 * @author kiran
 *
 */
public class FileCreation {

	@SuppressWarnings("resource")
	public static void fileCreate(LockedMeBOI lockedMeBOI,Scanner scan) {
		 scan = new Scanner(System.in);

		System.out.println("***Enter the File Name to create new file(FileName Should Contain Atleast 3 character):***");

		try {
			String fileName = scan.nextLine();

			if (fileName.length() > 3) {

				lockedMeBOI.addUserFile(fileName);
				System.out.println("**********************");
				System.out.println("\n");
			} else {
				System.out.println("File Name Should contain Atleast 3 characters ");
			}

		} catch (Exception e) {

			e.printStackTrace();
		} /*finally {
			scan.close();
		}*/
	}

}
```

GetFileByName.java

```
/**
 *
 */
package com.LockedMeDotCom.presentationView;
```

```
import java.awt.Desktop;
import java.io.File;
import java.util.Scanner;

import com.LockedMeDotCom.bussinessob.LockedMeBOI;

/**
 * @author kiran
 *
 */
public class GetFileName {

	@SuppressWarnings("resource")
	public static void getFileByName(LockedMeBOI lockedMeBOI,Scanner scan) {
		 scan = new Scanner(System.in);
		System.out.println("***Get file by providing specified File Name:***");
		try {


			String fileName = scan.nextLine();
			String flnm =lockedMeBOI.getFileByName(fileName);

			if(flnm.equals(fileName)) {
				System.out.println("***Do you want to open the file? Yes or No***");
			if ((scan.nextLine()).equalsIgnoreCase("Yes")) {

				if(!Desktop.isDesktopSupported()){
			   System.out.println("Desktop is not supported on opening the File");
			   return;
		    }
			 Desktop desktop = Desktop.getDesktop();
		    //let's try to open PDF file
		   File file = new File(System.getProperty("user.dir") + "/LockedMeDotCom/" +fileName);
		    if(file.exists()) desktop.open(file);

			}else {
				System.out.println("***There is some problem in opening the File***");
			}
			}
			System.out.println("**********************");
			System.out.println("\n");

		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();

		}
		/*finally {
			scan.close();
		}*/
```

```
	}

}
```

FileUpdate.java

```
/**
 *
 */
package com.LockedMeDotCom.presentationView;

import java.util.Scanner;

import com.LockedMeDotCom.bussinessob.LockedMeBOI;

/**
 * @author kiran
 *
 */
public class FileUpdate {

	@SuppressWarnings("resource")
	public static void updateFile(LockedMeBOI lockedMeBOI, Scanner scan) {
	 scan = new Scanner(System.in);
		System.out.println("****Update the Existing File with new contents*****");
		System.out.println("**Enter the FileName to update**");
		String fileName = scan.nextLine();
		try {
			if (lockedMeBOI.getFileByName(fileName).equals(fileName)) {
				System.out.println("**Enter contents to be updated**");
				String contents = scan.nextLine();
				try {
					lockedMeBOI.updateFile(fileName, contents);
					System.out.println("\n");
				} catch (Exception e2) {
					// TODO Auto-generated catch block
					e2.printStackTrace();
				}
			} else {
				System.out.println("Please, Check the File Name");
				System.out.println("\n");
			}
		} catch (Exception e2) {
			// TODO Auto-generated catch block
			e2.printStackTrace();
		}
		/*finally {
			scan.close();
		}*/
```

```
	}


}
```

FileDelete.java

```
/**
 *
 */
package com.LockedMeDotCom.presentationView;

import java.util.Scanner;

import com.LockedMeDotCom.bussinessob.LockedMeBOI;

/**
 * @author kiran
 *
 */
public class FileDelete {
	@SuppressWarnings("resource")
	public static void deleteFile(LockedMeBOI lockedMeBOI,Scanner scan) {
		 scan = new Scanner(System.in);
		System.out.println("***Enter the file name to delete the file from directory:***");

		String filename = scan.nextLine();
		try {

			if (lockedMeBOI.getFileByName(filename).equals(filename)) {

				System.out.println("**Are you sure want to delete? Yes Or No**");
				if ((scan.nextLine()).equalsIgnoreCase("Yes")) {
					try {
						lockedMeBOI.deleteUserFile(filename);
						System.out.println("**********************");
						System.out.println("\n");

					} catch (Exception e) {

						e.printStackTrace();
					}
				} else {
					System.out.println("**********************");
					System.out.println("\n");

				}
			} else {
				System.out.println("Use Right File Name to delete:");

			}
```

```
                } catch (Exception e1) {

                }

                /*finally {
                        scan.close();
                }       */
        }

}
```

LockedMeBOI.java (Interface class)

```
/**
 *
 */
package com.LockedMeDotCom.bussinessob;

import java.io.File;
import java.io.IOException;

/**
 * @author kiran
 *
 */
public interface LockedMeBOI {
        public File addUserFile(String files) throws IOException;

        public String getFileByName(String fileNeme) throws IOException;

        public void updateFile(String string, String fileContents) throws IOException;

        public void getAllFileNames() throws IOException;

        public void deleteUserFile(String fileName) throws IOException;

}
```

LockedMeBOImpl.java (implemented class)

```
/**
 *
 */
package com.LockedMeDotCom.bussinessObject.implimentations;

import java.io.File;
import java.io.IOException;

import com.LockedMeDotCom.dataAccessObject.LockedMeDAO;
import com.LockedMeDotCom.dataAccessObject.implimentaions.LockedMeDaoImpl;
import com.LockedMeDotCom.bussinessob.LockedMeBOI;

/**
 * @author kiran
 *
 */
public class LockedMeBoImpl implements LockedMeBOI {

	private LockedMeDAO lmd;

	private LockedMeDAO getLMD() {
		if (lmd == null) {
			lmd = new LockedMeDaoImpl();
		}
		return lmd;
	}

	public File addUserFile(String fileName) throws IOException {

		String userFileName = "";
		String[] filename = fileName.split(" ");
		for (String name : filename) {
			userFileName = name.substring(0, 1).toUpperCase() +
name.substring(1).toLowerCase();
		}

		File fn = getLMD().addUserFile(userFileName);

		return fn;
	}

	public String getFileByName(String fileName) throws IOException {

		String file = "";
		String fname = getLMD().getFileByName(fileName);

		if ((fname).equals(fileName)) {
```

```
                file = fname;
                System.out.println("File Found is: " + file);

        } else {
                System.out.println(fileName + " " + "does not exist");
        }

        return file;
}

public void updateFile(String fileName, String fileContents) throws IOException {
        boolean updatesuccess = false;
        getLMD().updateFile(fileName, fileContents);
        updatesuccess = true;

        if (updatesuccess) {
                System.out.println("Update Success full");
                System.out.println("\n");
        } else {
                System.out.println("Update Fail");
                System.out.println("\n");
        }

}

public void deleteUserFile(String fileName) throws IOException {

        File file = getLMD().deleteUserFile(fileName);
        if (file.exists()) {
                file.delete();
        } else {
                System.out.println("File Does not exit in the directory");
        }

        System.out.println("File has been Deleted");

}

public void getAllFileNames() throws IOException {

        // java.util.List<File>
        // fileName=(getLMD().getAllFileNames()).stream().sorted().collect(Collectors.toList());
        // fileName.stream().forEach(System.out::println);
        java.util.List<File> fileName = getLMD().getAllFileNames();

        int listSize = fileName.size();
        if (listSize == 0) {
                System.out.println("Directory is Empty");

        } else {
                for (File file : fileName) {
```

```
                        String name = file.getName();

                        System.out.print(name + "\n");

                    }

                }
                System.out.println("Totoal number of file Available in the Directory:" + listSize);

        }
}
```

LockedMeDAO.java (Interface class)

```
/**
 *
 */
package com.LockedMeDotCom.dataAccessObject;

import java.io.File;
import java.io.IOException;
import java.util.List;

/**
 * @author kiran
 *
 */
public interface LockedMeDAO {

        public File addUserFile(String files) throws IOException;

        public String getFileByName(String fileNeme) throws IOException;

        public List<File> getAllFileNames() throws IOException;

        public void updateFile(String fileName, String fileContents) throws IOException;

        public File deleteUserFile(String fileName) throws IOException;

}
```

LockedMeDoaImpl (implemented class)

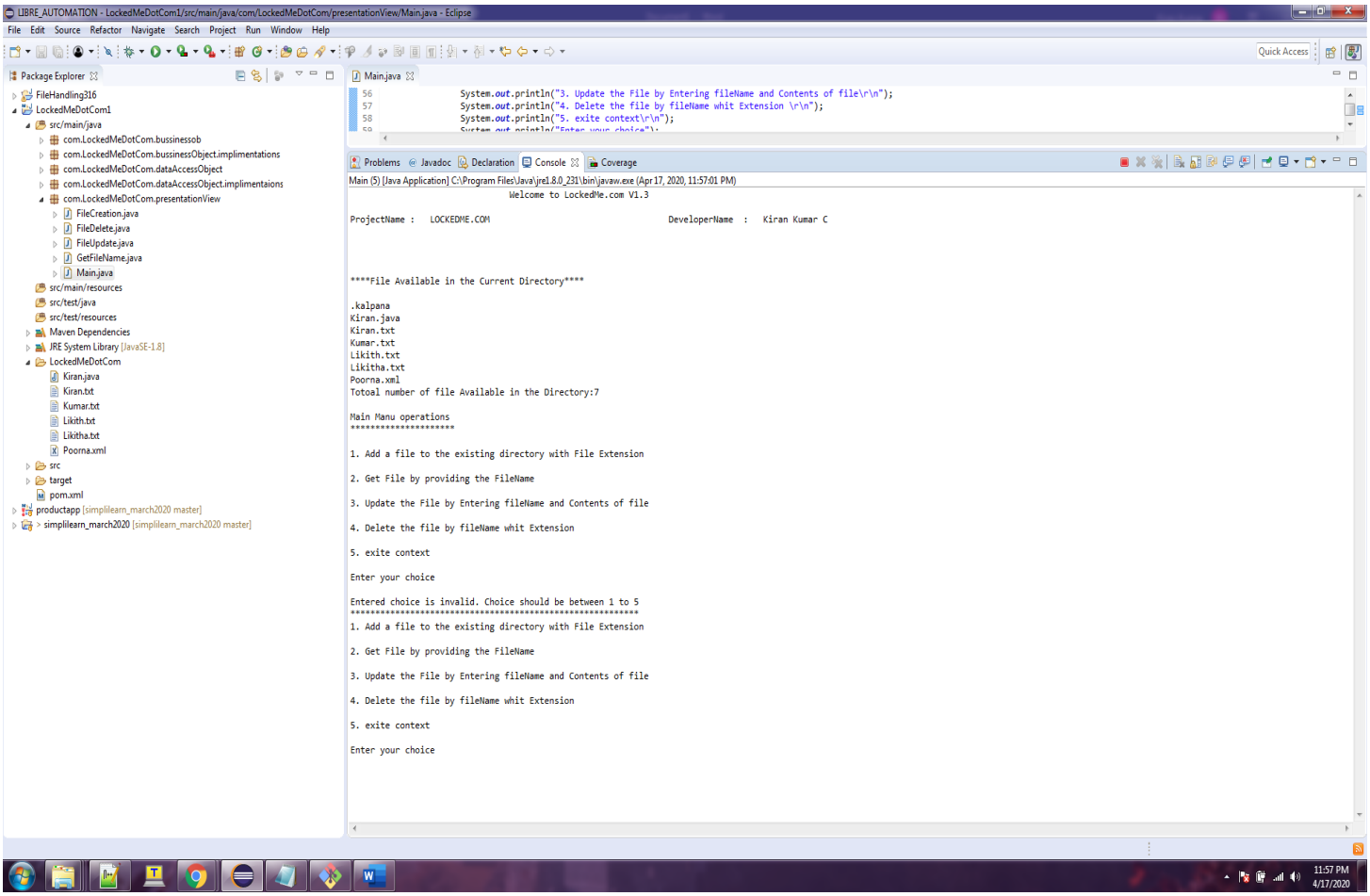
```
/**
 *
 */
package com.LockedMeDotCom.dataAccessObject.implimentaions;

import java.io.BufferedWriter;
```

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

import java.util.List;

import com.LockedMeDotCom.dataAccessObject.LockedMeDAO;

/**
 * @author kiran
 *
 */
public class LockedMeDaoImpl implements LockedMeDAO {

	/*
	 * private static String getFileExtension(String filename) { String fileName =
	 * filename; if(fileName.lastIndexOf(".") != -1 && fileName.lastIndexOf(".") !=
	 * 0) return fileName.substring(fileName.lastIndexOf(".")+1); else return ""; }
	 */

	public File addUserFile(String userFileName) throws IOException {

		boolean success = false;

		File file = new File(System.getProperty("user.dir") + "/LockedMeDotCom/" +
userFileName);

		if (file.exists()) {
			System.out.println("File already exists");
		} else {
			System.out.println("No such file exists, creating now");
			success = file.createNewFile();

			if (success) {
				System.out.println("Successfully created new file:" + userFileName);
				try {
					Thread.sleep(1000);
				} catch (InterruptedException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}

				Runtime.getRuntime()
						.exec("notepad" + " " + System.getProperty("user.dir") +
"/LockedMeDotCom/" + userFileName);

			} else {
				System.out.println("Failed to create new file:" + userFileName);
			}
```

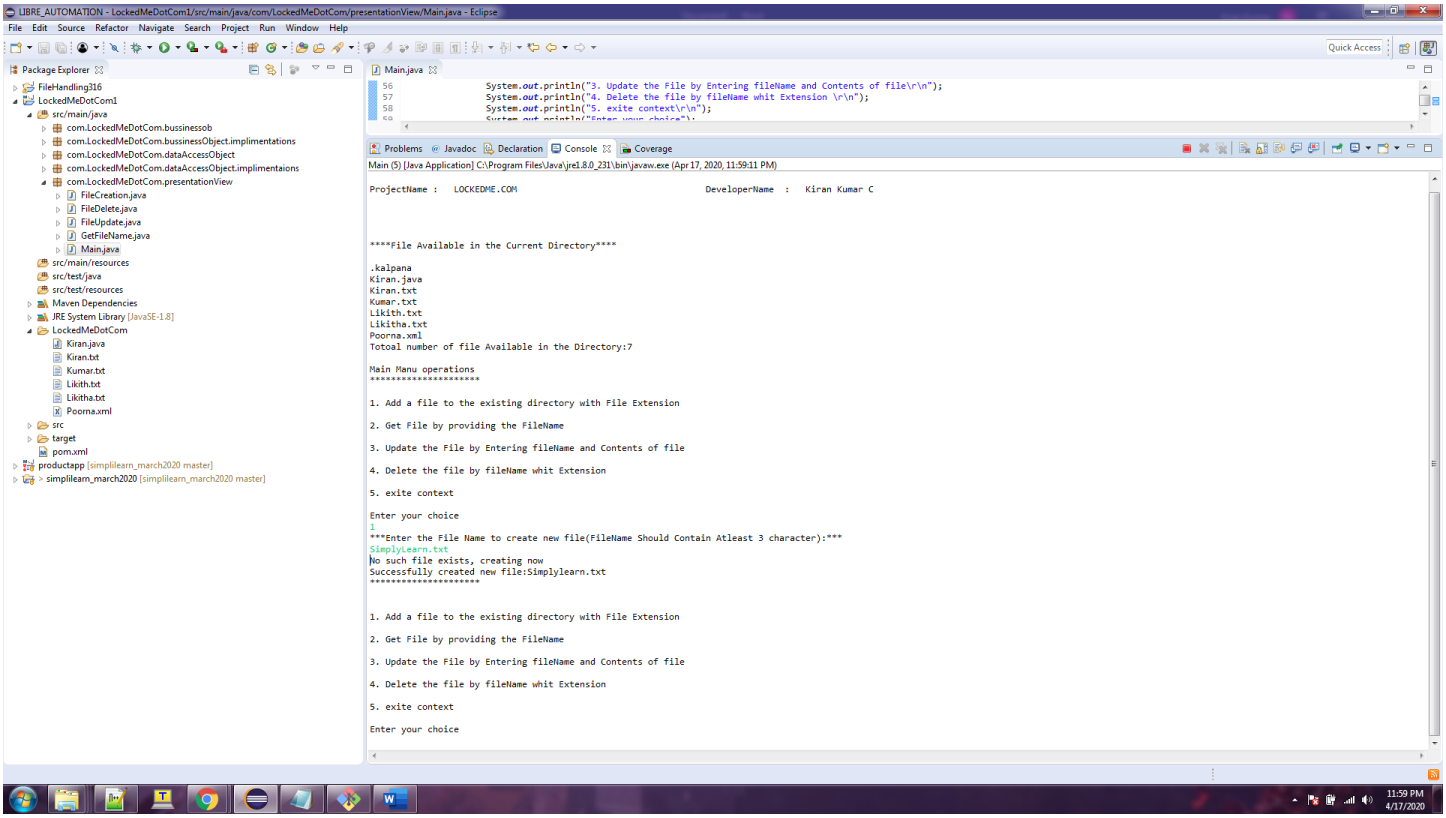
```
		}

		return file;

	}

	public String getFileByName(String fileName) throws IOException {
		List<File> listFiles = this.getAllFileNames();
		String fi ="";
		for (int i = 0; i <= listFiles.size() - 1; i++) {
			if ((listFiles.get(i).getName()).equals(fileName)) {
				fi = listFiles.get(i).getName();
			}
		}
		return fi;

	}

	public List<File> getAllFileNames() throws IOException {
		List<File> fileList = new ArrayList<File>();
		File dir = new File(System.getProperty("user.dir") + "/LockedMeDotCom/");

		File[] files = dir.listFiles();
		for (File file : files) {

			fileList.add(file);
		}

		return fileList;
	}

	public void updateFile(String fileName, String fileContents) throws IOException {

		boolean success = false;

		BufferedWriter bw = null;
		FileWriter fw = null;

		try {
			File file = new File(System.getProperty("user.dir") + "/LockedMeDotCom/" +
fileName);
			if (file.exists()) {
				success = true;
			} else {
				success = false;
			}
			if (success) {

				// true = append file
				fw = new FileWriter(file.getAbsoluteFile(), true);
				bw = new BufferedWriter(fw);
```

```
                    bw.write(fileContents);

                }
                else {
                    System.out.println("FIle Does not Found");
                }

            } catch (IOException e) {

                e.printStackTrace();

            } finally {

                try {

                    if (bw != null)
                        bw.close();

                    if (fw != null)
                        fw.close();

                } catch (IOException ex) {

                    ex.printStackTrace();

                }
            }
        }

        public File deleteUserFile(String fileName) throws IOException {

            File file = new File(System.getProperty("user.dir") + "/LockedMeDotCom/" + fileName);



            return file;

        }

}
```

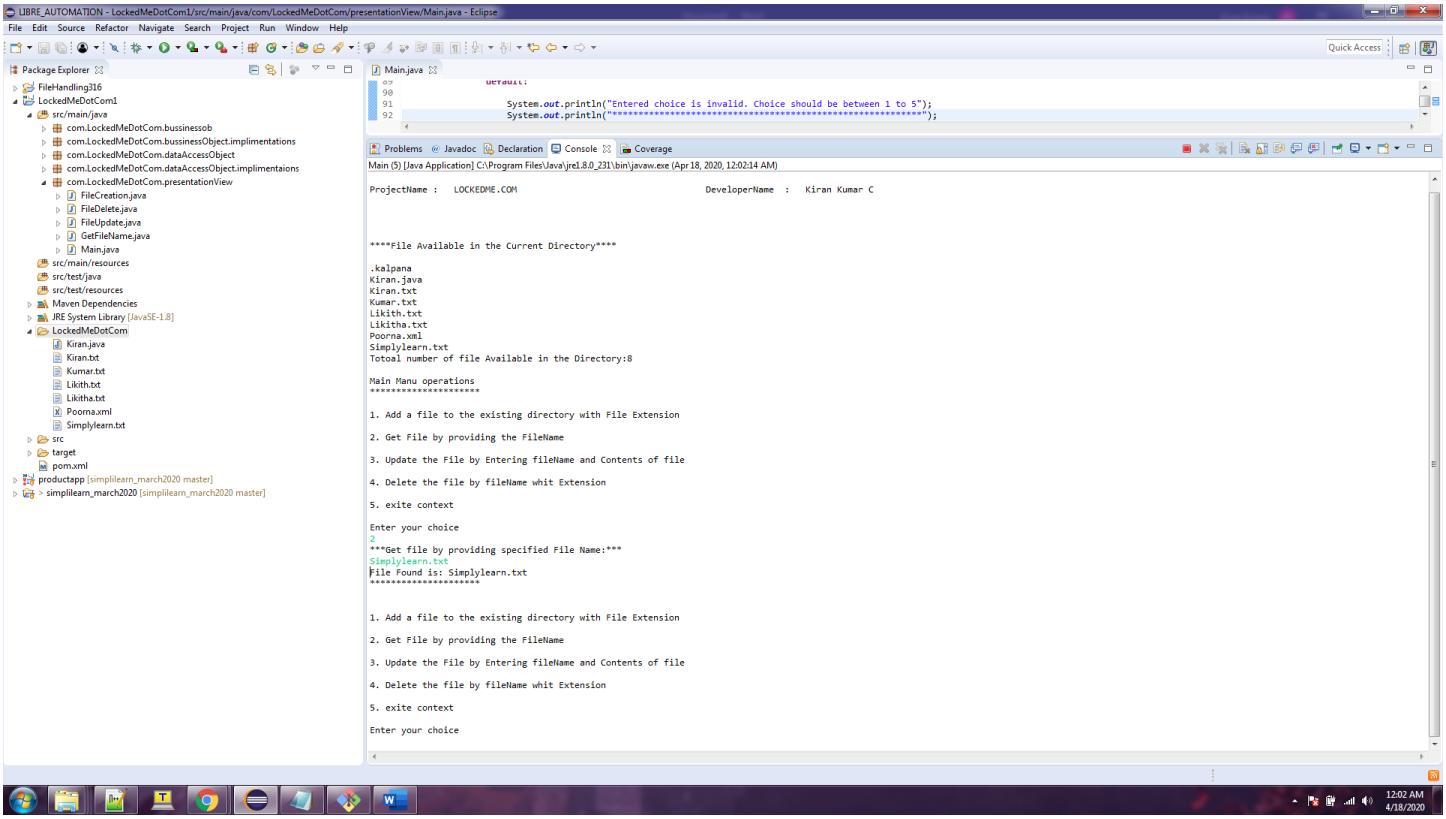
LockedMeDotCom Screen Shots positvie testcases:



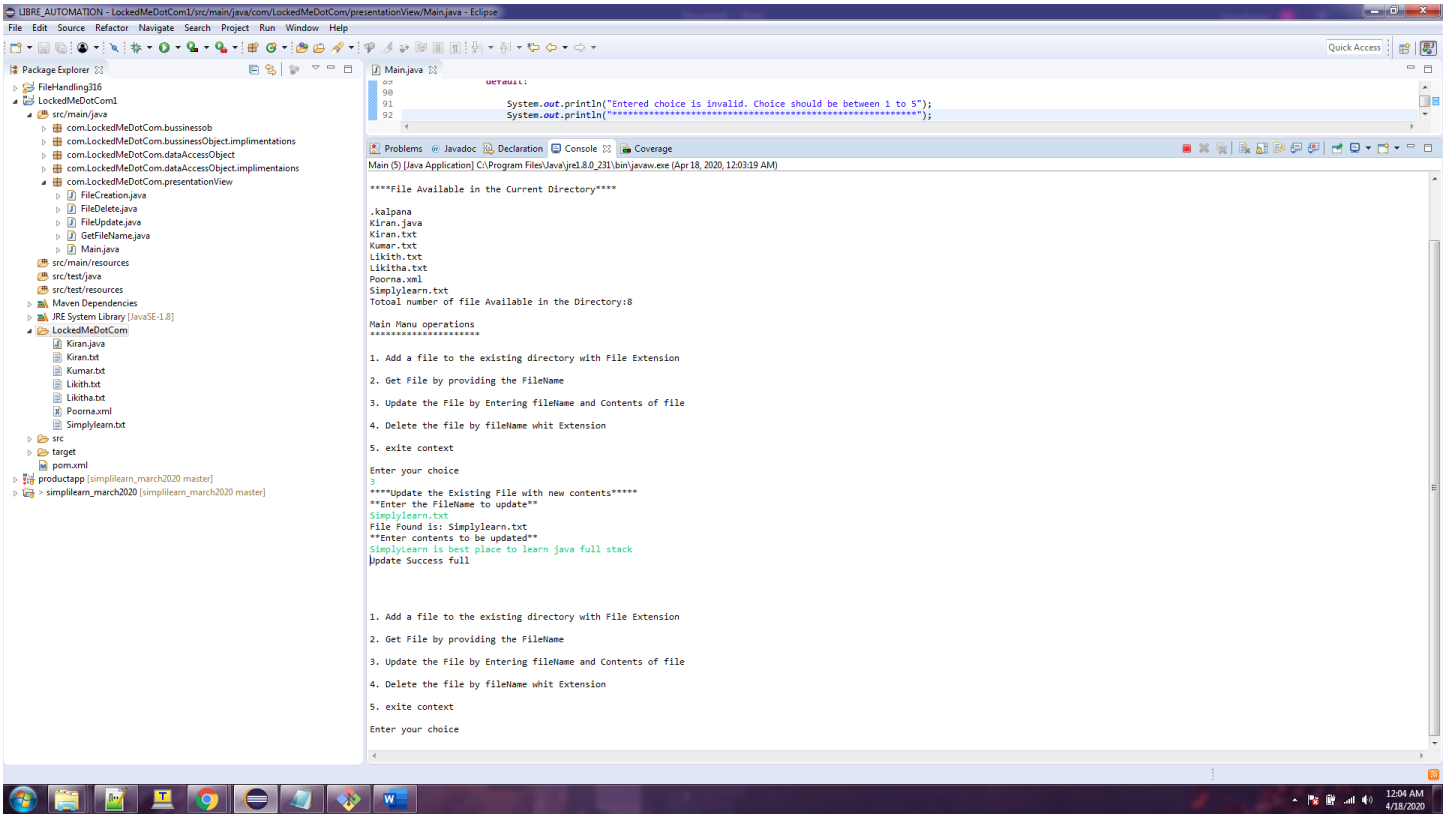
Screenshot 1.0: returns Invalid choice when user Selects invalid Option



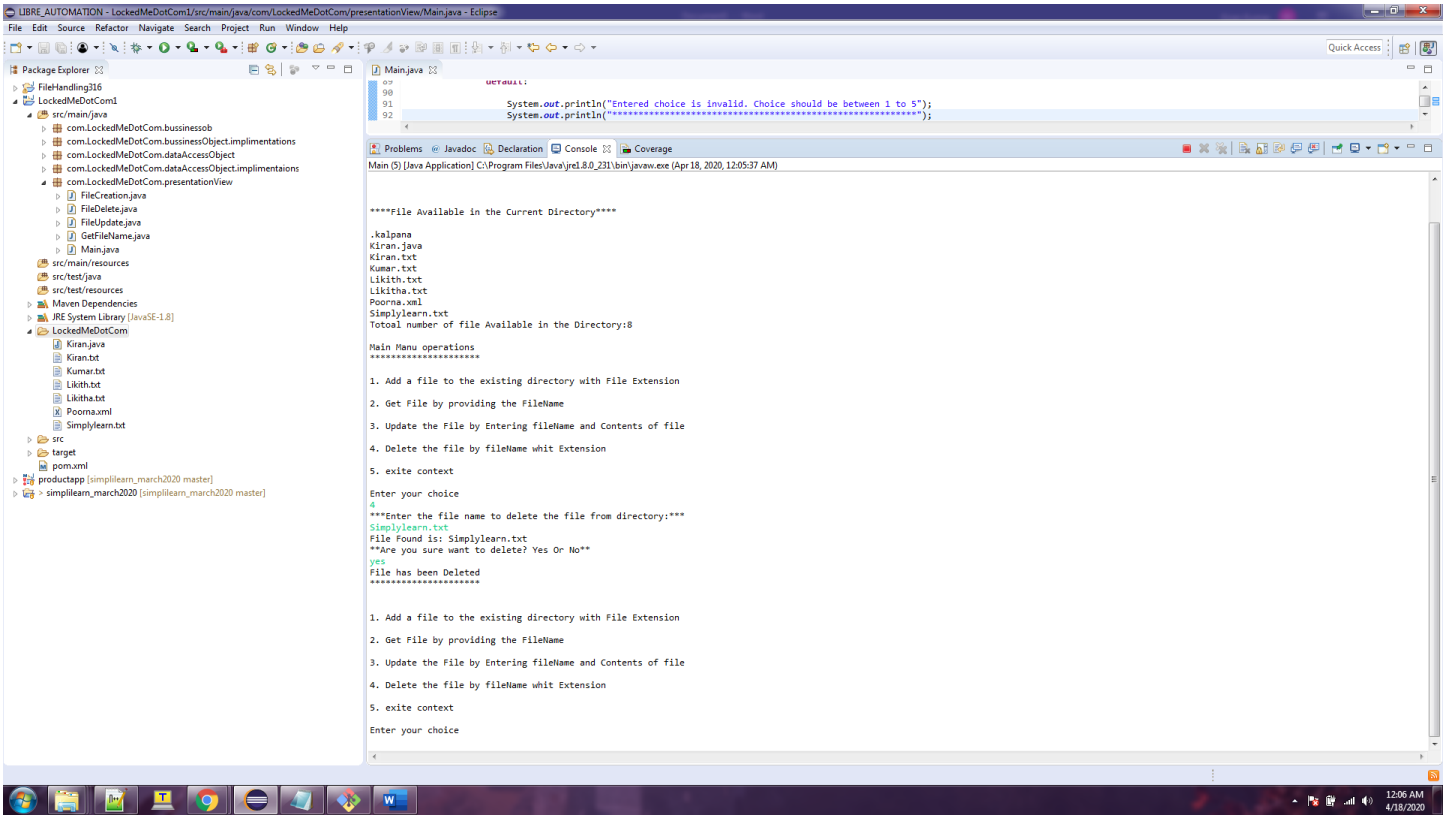
Screenshot 1.1: File Adding Successful



Screenshot1.2: Get File By Name Success ful

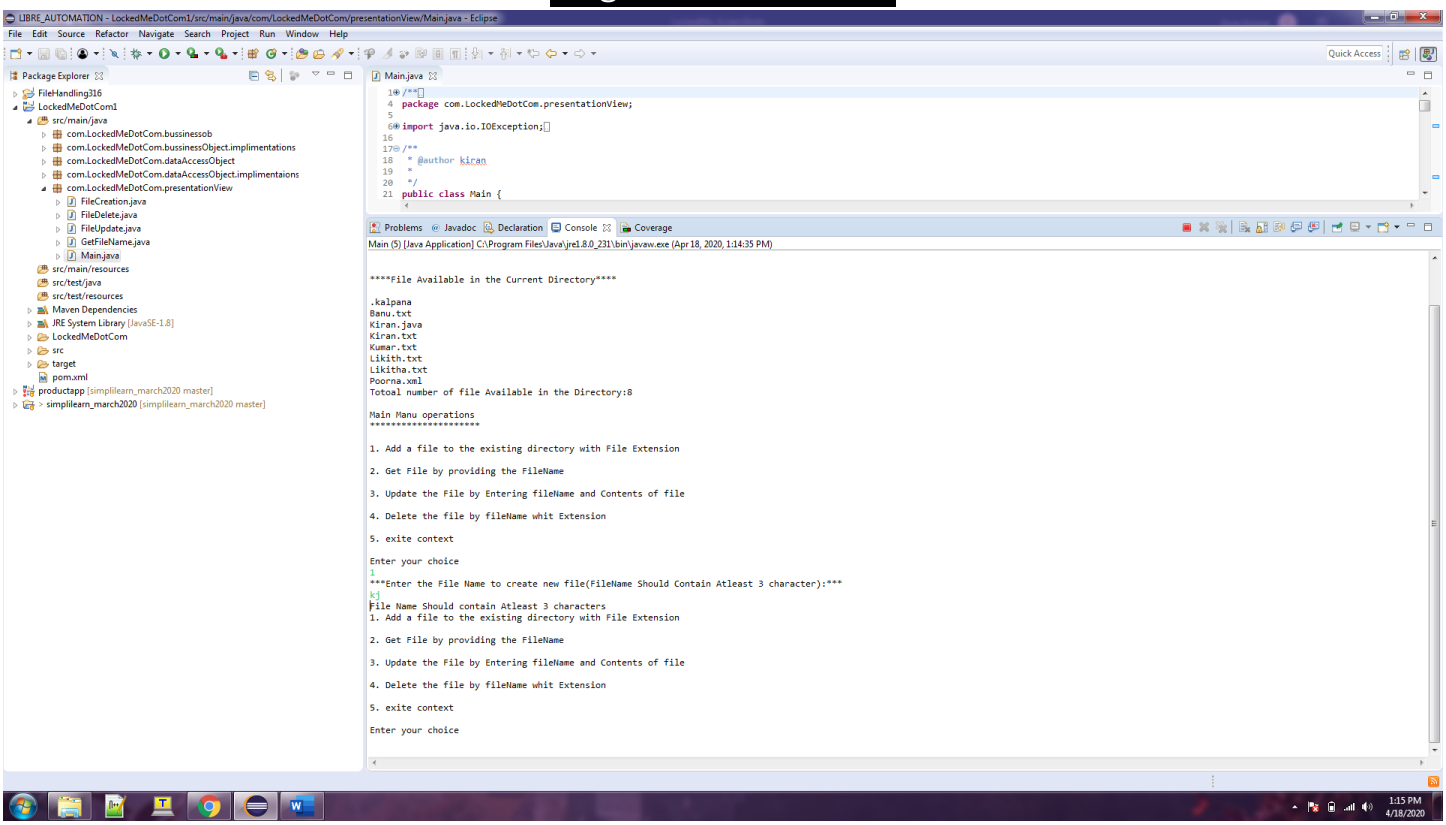


Screeshot1.3: File update Successful

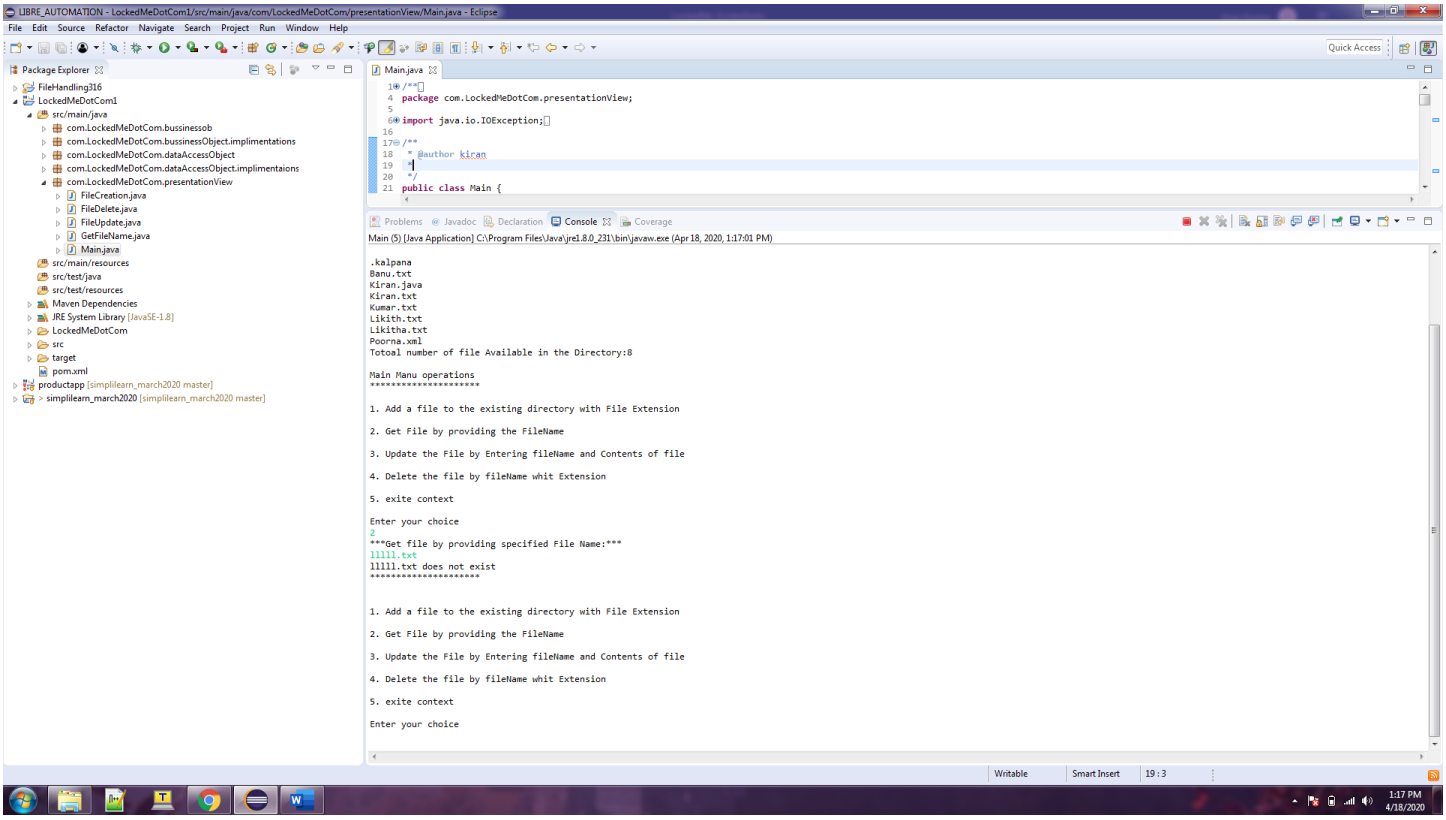


Screenshot1.4; File Deletion Success full

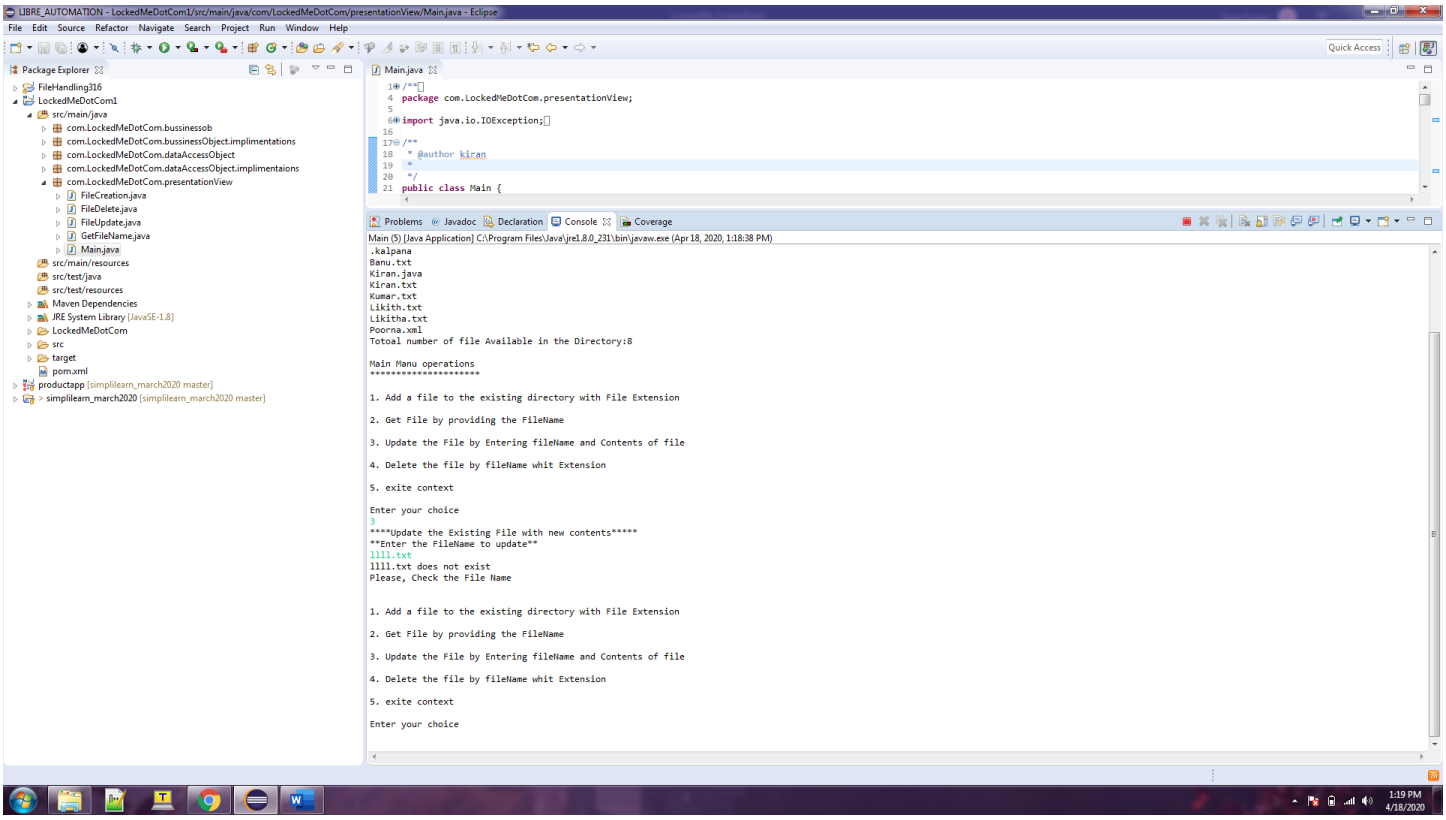
Negative test Cases:



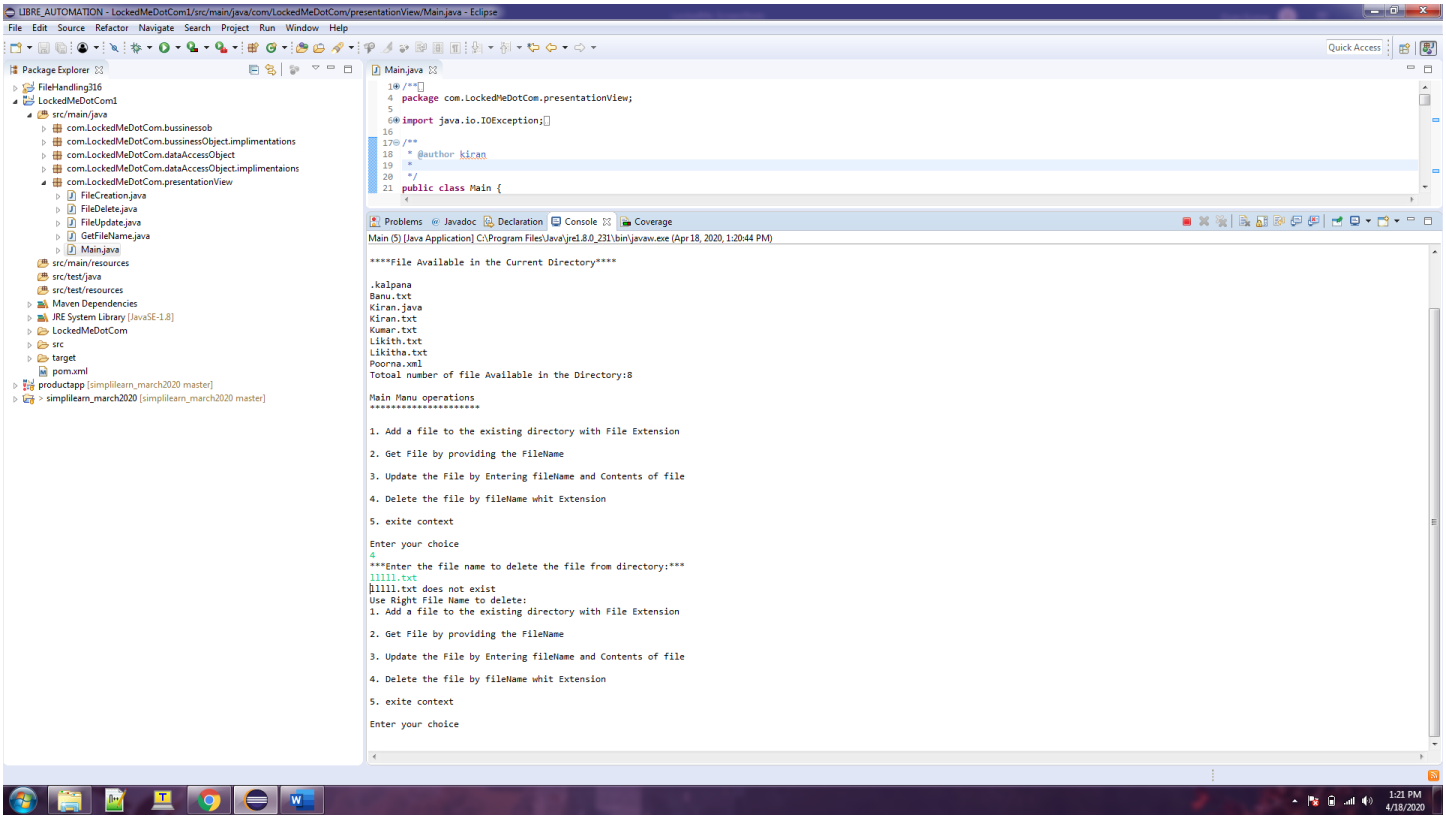
Screen shots 2.0: throws file name should be at least 3 character long when enter below 3 characters



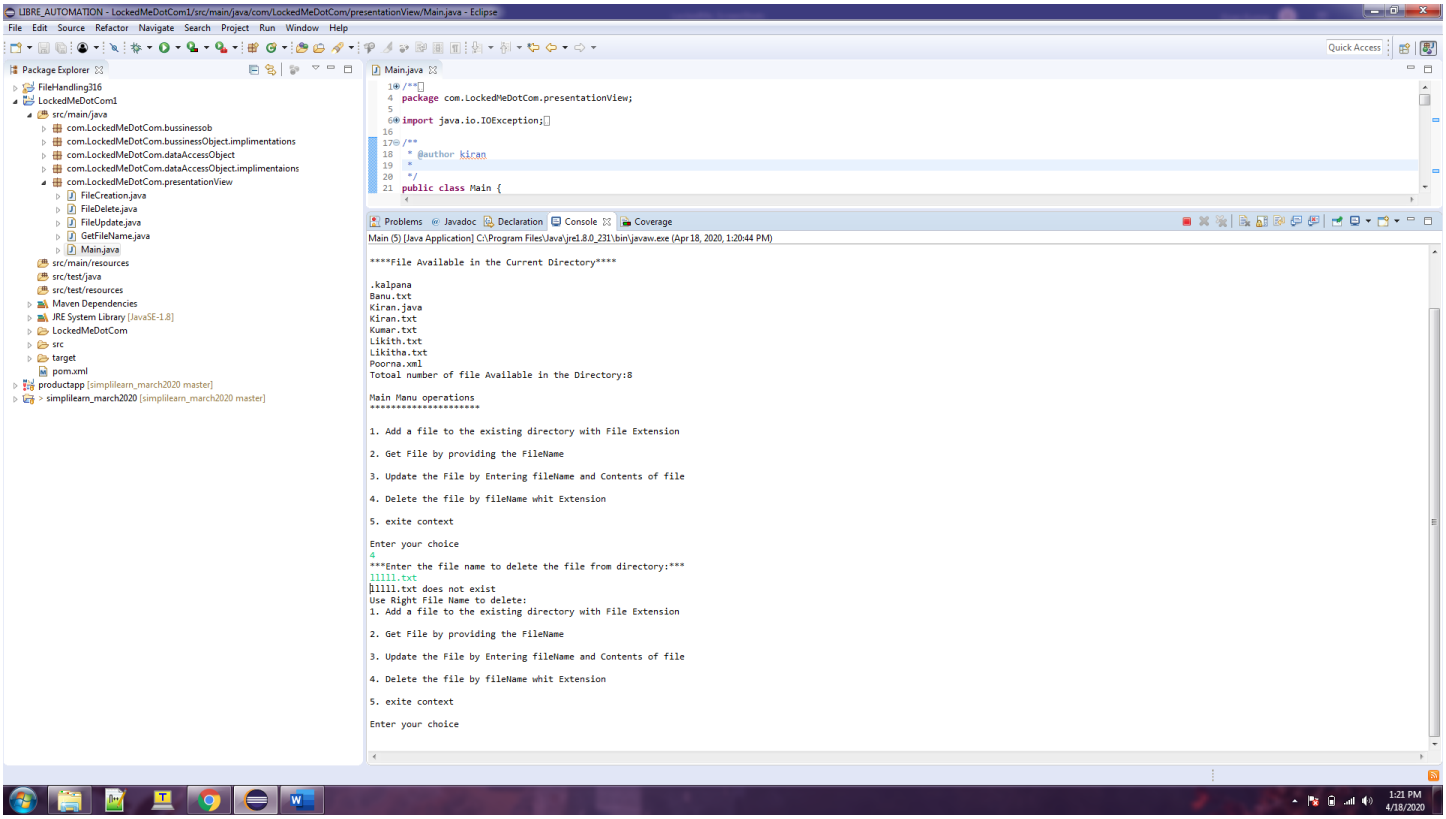
Screen shot 2.1: throws file does not exist when user enters a wrong file name.



Screen shot 2.2: when user enter wrong file application throw file does not exist, enter the right correct file to update.



Screen shot 2.3 throw file does not exist, use right file name to delete, when user enters a wrong file name.



Screen shot 2.4: application will be exit from the execution when user selects option 5.

Self-learning Practice Project assignment GitHub Like:

>> git@github.com:kiranc2210/SimplyLearnPracticeProjects.git

>> https://github.com/kiranc2210/GITHUB_Project.git

LockedmeDotCom Project Assignment GitHub Like

>> https://github.com/kiranc2210/virtualFileManagementApp.git