```
package com.LockedMe;

import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;

public class LockedMe {

	public static String currentDir;
	public File folderName;

	// Welcome display
	void startDisplay() {
		System.out.println("************** LockedMe.com --
Sunil kala ******************");
		System.out.println("");
	}

	// Exit display
	void exitDisplay() {
		System.out.println("**************THANK YOU
******************");
		System.out.println("");
	}

	// Creating main menu with various lists
	public void mainMenuDisplay() {
		System.out.println();
		System.out.println("Select any one of the following:
");
		System.out.println("1.List ALL Files \n2.Business
level Operation \n3.Exit");

	}

	// creating submenus with adding ,deleting ,searching
features
	public void subMenuDisplay() {
		System.out.println("Select Operation:           ");
		System.out.println("a.Add a file \nb.Delete a file
\nc.Search a file\nd.Go Back To Main Menu\ne.Exit");

	}

	// main menu method
	void mainMenu() {
		do {
			mainMenuDisplay();

			try {
				Scanner sc = new Scanner(System.in);
				int option =
Integer.parseInt(sc.nextLine());
```

```
                        switch (option) {
                        case 1: {
                              // displaying all files.
                              ListAllFiles();
                              mainMenu();

                        }
                        case 2: {
                              subMenu();

                        }
                        case 3: {
                              System.out.println("Thank You");
                              System.exit(0);
                        }
                        default:
                              System.out.println("\n Invalid
input \n please enter from 1  2  3");
                              mainMenu();
                        }

                        sc.close();
                  } catch (Exception e) {
                        System.out.println("\n Invalid input \n
please enter from 1  2  3");
                        mainMenu();
                  }
            } while (true);
      }

      // creating submenus method
      public void subMenu() {
            subMenuDisplay();
            try {
                  Scanner sc = new Scanner(System.in);
                  char option =
sc.nextLine().toLowerCase().charAt(0);
                  switch (option) {
                  case 'a': {
                        System.out.print("Enter a file name to
add: ");
                        String fileName =
sc.next().trim().toLowerCase();
                        addFile(fileName);
                        break;
                  }
                  case 'b': {
                        System.out.print("Enter a file name to
delete: ");
                        String filename = sc.next().trim();
                        deleteFile(filename);
                        break;
                  }
                  case 'c': {
                        System.out.print("Enter file name to
```

```
search: ");
                        String filename = sc.next().trim();
                        searchFile(filename);
                        break;
                  }
                  case 'd': {
                        System.out.println("Back to Main Menu");
                        mainMenu();
                        break;
                  }
                  case 'e': {
                        System.out.println("Thank You");
                        System.exit(0);
                  }
                  default:
                        System.out.println("\n Invalid input \n
Please enter from  a, b, c, d");
                  }
                  subMenu();
            } catch (Exception e) {
                  System.out.println("\n Invalid input \n Please
enter a, b, c, d");
                  subMenu();
            }
      }

      //main menu listing files
      public void ListAllFiles() {
            if (folderName.exists()) { //
                  String[] files = folderName.list();
                  if (files != null && files.length > 0) {
                        Set<String> file = new TreeSet<>
(Arrays.asList(files));
                        for (String i : file) {
                              System.out.println(i);
                        }
                  } else {
                        System.out.println("Folder is empty");
                  }
            } else {
                  throw new NullPointerException("Directory
doesnt exist");
            }

      }


      //adding files to folder
      void addFile(String fileName) throws IOException {
            if (fileName == null || fileName.isEmpty()) {
                  throw new NullPointerException("Invalid file
name");
            }
            File filepath = new File(folderName + File.separator
+ fileName);
```

```
            if (filepath.createNewFile()) {
                  System.out.println(fileName + " is created at:
" + folderName);
            } else {
                  System.out.println(fileName + " already exists
at " + folderName);
            }

      }

      //deleting files
      void deleteFile(String fileName) throws IOException {
            if (fileName == null || fileName.isEmpty()) {
                  throw new NullPointerException("Invalid file
name");
            }
            File filepath = new File(folderName + File.separator
+ fileName);

            if (filepath.delete()) {
                  System.out.println(fileName + " is deleted
from:" + folderName);
            } else {
                  System.out.println(fileName + " files not
found" + "\n" + "operation unsucccessful.");
            }

      }

      //searching files
      void searchFile(String fileName) {
            String[] files = folderName.list();
            for (String i : files) {
                  if (i.equals(fileName)) {
                        System.out.println(fileName + " found at
" + folderName);
                        return;
                  }
            }
            System.out.println("File Not found");
      }

      static String dir;

      public LockedMe() {
            currentDir = System.getProperty("user.dir");
            folderName = new File(currentDir + File.separator +
"lockedMeFolder");
            if (folderName.exists()) {
                  dir = folderName.getAbsolutePath();
            } else {
                  folderName.mkdirs();
            }
      }

}
```