```
package com.fullstack.learining ;
import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import java.util.Scanner;


public class Phase1 {
   static String DIRECTORY;
   File folder_name;

   public  Phase1() {
      DIRECTORY = System.getProperty("user.dir");
      folder_name = new File(DIRECTORY+"/files");
      if (!folder_name.exists())
         folder_name.mkdirs();
      System.out.println("DIRECTORY : "+ folder_name.getAbsolutePath());
   }

   private static final String WELCOME_PROMPT =
         "\n***   Phase1.com ***";
            private static final String MAIN_MENU_PROMPT =
         "\nMAIN MENU - Select any of the following: \n"+
               "1 -> List files in directory\n"+
               "2 -> Add, Delete or Search\n"+
               "3 -> Exit Program";

   private static final String SECONDARY_MENU_PROMPT =
         "   \nSelect any of the following: \n"+
```

```
            "   a -> Add a file\n"+
            "   b -> Delete a file\n"+
            "   c -> Search a file\n"+
            "   d -> GoBack";

void showPrimaryMenu() {
   System.out.println(MAIN_MENU_PROMPT);
   try(Scanner scanner = new Scanner(System.in)){

      int option = scanner.nextInt();
      switch (option){
         case 1 : {
            showFiles();
            showPrimaryMenu();
         }
         case 2 : {
            showSecondaryMenu();
         }
         case 3 : {
            System.out.println("Thank You");
            System.exit(0);
         }
         default: showPrimaryMenu();
      }
   }
   catch (Exception e){
      System.out.println("Please enter 1, 2 or 3");
      showPrimaryMenu();
   }
```

```
}

void showSecondaryMenu() {
   System.out.println(SECONDARY_MENU_PROMPT);
   try(Scanner scanner = new Scanner(System.in))
   {
      char[] input = scanner.nextLine().toLowerCase().trim().toCharArray();
      char option = input[0];

      switch (option){
         case 'a' : {
            System.out.print("↳ Adding a file...Please Enter a File Name : ");
            String filename = scanner.next().trim().toLowerCase();
            addFile(filename);
            break;
         }
         case 'b' : {
            System.out.print("↳ Deleting a file...Please Enter a File Name : ");
            String filename = scanner.next().trim();
            deleteFile(filename);
            break;
         }
         case 'c' : {
            System.out.print("↳ Searching a file...Please Enter a File Name : ");
            String filename = scanner.next().trim();
            searchFile(filename);
            break;
         }
         case 'd' : {
```

```
            System.out.println("Going Back to MAIN menu");
            showPrimaryMenu();
            break;
          }
          default : System.out.println("Please enter a, b, c or d");
       }
       showSecondaryMenu();
    }
    catch (Exception e){
       System.out.println("Please enter a, b, c or d");
       showSecondaryMenu();
    }
}

void showFiles() {
    if (folder_name.list().length==0)
       System.out.println("The folder is empty");
    else {
       String[] list = folder_name.list();
       System.out.println("The files in "+ folder_name +" are :");
       Arrays.sort(list);
       for (String str:list) {
          System.out.println(str);
       }
    }
}

void addFile(String filename) throws IOException {
    File filepath = new File(folder_name +"/"+filename);
```

```
    String[] list = folder_name.list();
    for (String file: list) {
        if (filename.equalsIgnoreCase(file)) {
            System.out.println("File " + filename + " already exists at " + folder_name);
            return;
        }
    }
    filepath.createNewFile();
    System.out.println("File "+filename+" added to "+ folder_name);
}

void deleteFile(String filename) {
    File filepath = new File(folder_name +"/"+filename);
    String[] list = folder_name.list();
    for (String file: list) {
        if (filename.equals(file) && filepath.delete()) {
            System.out.println("File " + filename + " deleted from " + folder_name);
            return;
        }
    }
    System.out.println("Delete Operation failed. FILE NOT FOUND");
}

void searchFile(String filename) {
    String[] list = folder_name.list();
    for (String file: list) {
        if (filename.equals(file)) {
            System.out.println("FOUND : File " + filename + " exists at " + folder_name);
            return;
```

```
            }
        }
        System.out.println("File NOT found (FNF)");
    }

    public static void main(String[] args) {
        System.out.println(WELCOME_PROMPT);
        Phase1 menu = new  Phase1();
        menu.showPrimaryMenu();
    }
}
```