Locked me Source Code

```
package VirtualKeyRepo;

import java.io.File;

import java.util.Arrays;

import java.util.Scanner;

public class VirtualKey {

	public static void main(String[] args) {

		System.out.println("\n####Welcome to LockedMe.com developed by ALEKHYA####\n");

		Scanner sc = new Scanner(System.in);

	System.out.println("Enter the file path to use the application");

     boolean hh = true;

    while(hh) {

		 String path = sc.next();

		 File file = new File(path);

		 if(file.exists()) {

		 System.out.println("\nFolder path found successfully you can continue with the oprations");

		 boolean bb = true;

		 while(bb) {

		 System.out.println("\n1.To see the files or folder names in an ascending order\n2. To go inside and do the operatios \n3.To close the application");


		 System.out.println("\nEnter your choice");

		int key = sc.nextInt();


		switch (key) {


		case 1:

			File files[] = file.listFiles();

			Arrays.sort(files);


			for (int i = 0; i < files.length; i++)

				System.out.println(files[i]);
```

```
                    break;


                case 2:
                {
                    boolean flag = true;
                    while(flag) {
                    System.out.println("1.To add file\n2.To delete file\n3.To search the file\n4.To
return tos the main contex ");

                  System.out.println("\nEnter your choice");
                  int key1 = sc.nextInt();

                    switch(key1) {

                    case 1:
                        System.out.println("Enter the File Name to add");
                        String filename = sc.next();

                        File cf = new File(path, filename);

                        try {
                            boolean b =cf.createNewFile();
                            if (b) {
                                System.out.println("The new file is added to the given
path\n");
                            } else {
                                System.out.println("The file name is already exists in
the folder");
                            }
                        }catch (Exception e) {}

                            break;

                    case 2:
                        System.out.println("Enter the File Name to delete");
```

```
            String df = sc.next();

            File file1 = new File(file, df);

            try {
                boolean b = file1.delete();
                if (b) {
                    System.out.println("File is Deleted successfully");
                } else {
                    System.out.println("File Not Found");
                }
            } catch (Exception e) {}
                break;

        case 3:
            System.out.println("Enter the file name you want to search");
            String search = sc.next();

            boolean b = false;

            File file2[] = file.listFiles();

            for (int i = 0; i < file2.length; i++) {
                if (file2[i].getName().startsWith(search)) {
                    b = true;
                    System.out.println("File founded successfully\n");
                    System.out.println(file2[i]);
                }
            }
            if (b == false) {
                System.out.println("File not  found to search");
            }
                break;
        case 4:
```

```
                                System.out.println("You have returned to the main context");
                                flag =false;
                                        break;

                        default:{
                                System.out.println("You have made an invalid choice!\n");

                                System.out.println("Enter the correct choice");
                                        break;
                                }
                        }
                        }
                        break;
                }
        case 3:
        {
                System.out.println("\nClosing the Application\n");
                System.out.println("Thank you!! For using our application");
                bb= false;
                break;
        }
        default:
                System.out.println("You have made an invalid choice!\n");
                System.out.println("Enter the correct choice");
                break;
                }
                 }
}
 else {
                System.out.println("\nEnter the correct file path to use the application");

        }
                        hh= false;
}
```

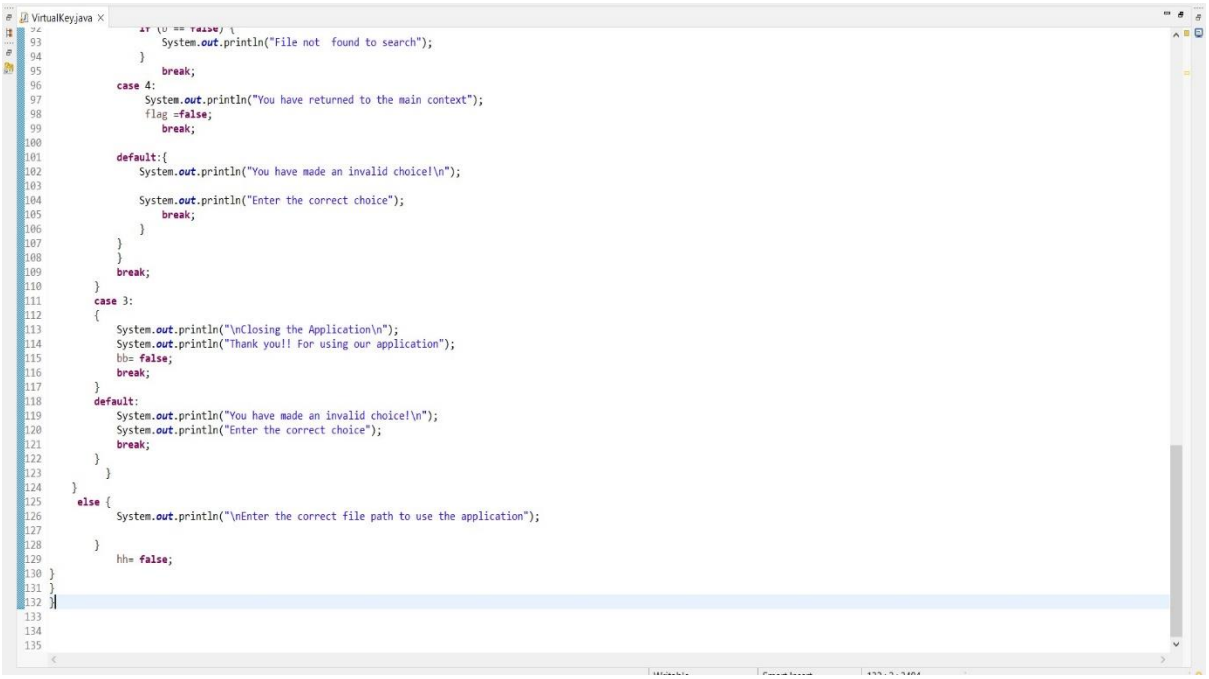
}

}

Output of Source code:

```
VirtualKey.java
        case 1:
            File files[] = file.listFiles();
            Arrays.sort(files);

            for (int i = 0; i < files.length; i++)
                System.out.println(files[i]);
            break;

        case 2:
        {
            boolean flag = true;
            while(flag) {
            System.out.println("1.To add file\n2.To delete file\n3.To search the file\n4.To return tos the main contex ");

            System.out.println("\nEnter your choice");
            int key1 = sc.nextInt();

            switch(key1) {

            case 1:
                System.out.println("Enter the File Name to add");
                String filename = sc.next();

                File cf = new File(path, filename);

                try {
                    boolean b =cf.createNewFile();
                    if (b) {
                        System.out.println("The new file is added to the given path\n");
                    } else {
                        System.out.println("The file name is already exists in the folder");
                    }
                }catch (Exception e) {}

                    break;

            case 2:
                System.out.println("Enter the File Name to delete");
                String df = sc.next();

                File file1 = new File(file, df);

                try {
```

```
VirtualKey.java
                try {
                    boolean b = file1.delete();
                    if (b) {
                        System.out.println("File is Deleted successfully");
                    } else {
                        System.out.println("File Not Found");
                    }
                } catch (Exception e) {}
                    break;

            case 3:
                System.out.println("Enter the file name you want to search");
                String search = sc.next();

                boolean b = false;

                File file2[] = file.listFiles();

                for (int i = 0; i < file2.length; i++) {
                    if (file2[i].getName().startsWith(search)) {
                        b = true;
                        System.out.println("File founded successfully\n");
                        System.out.println(file2[i]);
                    }
                }
                if (b == false) {
                    System.out.println("File not  found to search");
                }
                    break;
            case 4:
                 System.out.println("You have returned to the main context");
                 flag =false;
                    break;

            default:{
                System.out.println("You have made an invalid choice!\n");

                System.out.println("Enter the correct choice");
                    break;
                }
            }
            }
            break;
```


VirtualKeyjava
if (b == false) {
System.out.println("File not found to search");
}
break;
case 4:
System.out.println("You have returned to the main context");
flag =false;
break;
default:{
System.out.println("You have made an invalid choice!\n");
System.out.println("Enter the correct choice");
break;
}
}
break;
}
case 3:
{
System.out.println("\nClosing the Application\n");
System.out.println("Thank you!! For using our application");
bb= false;
break;
}
default:
System.out.println("You have made an invalid choice!\n");
System.out.println("Enter the correct choice");
break;
}
}
}
else {
System.out.println("\nEnter the correct file path to use the application");
}
hh= false;
}
}
}