

## SOURCE CODE EXPLANATION

### **CEPMain**

```
package phase1Project; import java.util.*;
public class CEPMain {
    public static void main(String[] args) {
        CEPOperations cepo = new CEPOperations();
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Welcome to the LockedMe.com Application!");
        System.out.println("-----");
        System.out.println("Developed By Sowmiyaa.");
        System.out.println("-----");
        int choice=0; do {
            while (true) {
                System.out.println("\nEnter 1 to Retrieve\nEnter 2 to do Operations\nEnter
                3 to Exit");
                System.out.println("Pick ur choice:");
                try {
                    choice = sc.nextInt();
                    break;
                }
                catch (InputMismatchException e) {
                    System.out.println("Invalid input....! ");
                    sc.next();
                }
            }

            switch(choice)
            {
                case 1:
                    cepo.Retrieving();
                    break;
                case 2:
                    int t=1;
                    do {
                        System.out.println("\nEnter a to add\nEnter b to delete\nEnter c to
                        search\nEnter d to go back to home");
                        char ch = sc.next().charAt(0);
                        switch(ch) {
                            case 'a':
                                cepo.Add();
                                break;
                            case 'b':
                                cepo.delete();
```

```

break;
case 'c':
    cepo.search();
    break;
case 'd':
    t=0;
    break;

default:
    System.out.println("Invalid Input....!. ");
    break;
}
}
while(t!=0);
break;
case 3:
    System.out.println("-----");
    System.out.println("Thanks for using the app.");
    System.exit(0);
default:
    System.out.println("Invalid Input....!. ");
    break;
}
}
while(true);
}
}

```

### **CEPOperations**

```

package phase1Project; import java.io.*;
import java.util.*;
public class CEPOperations {

    Scanner sc = new Scanner(System.in);
    public void Retrieving() {
        File folder = new File("/Users/sowmiyaa/Downloads/phase1Project");
        if (!folder.exists())
        {
            folder.mkdirs();
        }
        File[] files = folder.listFiles();
        if(files.length==0) {
            System.out.println("No Files Found....");
        }
        else {

```

```

List<String> fileList = new ArrayList<>();
// Loop through each file in the folder and add its name to the list for (File
file : files) {
if (file.isFile())
{
fileList.add(file.getName());
}
}
int n = fileList.size();
for (int i = 0; i < n - 1; i++) {
for (int j = i + 1; j < n; j++) {
if (fileList.get(i).compareTo(fileList.get(j)) > 0) {
String temp = fileList.get(i);
fileList.set(i, fileList.get(j));
fileList.set(j, temp);
} }
}
// Print the list of file names for (String fileName : fileList) {
System.out.println(fileName); }
} }
public void Add() {
File folder = new File("/Users/sowmiyaa/Downloads/phase1Project");
if (!folder.exists())
{
folder.mkdirs();
}
File[] files = folder.listFiles();
boolean checker;
String nameoffile="";
do
{
System.out.println("Enter the name of the file to add: ");
nameoffile = sc.next();
checker = false;
for (File file : files) {
if (nameoffile.equalsIgnoreCase(file.getName()))
{
System.out.println("File name already exists....");
checker = true;
break;
}
}
}
while(checker);
System.out.println("Enter the content of the file: ");
sc.nextLine();
String contentoffile = sc.nextLine();

```

```

File file = new File(folder, nameoffile);
try {
    FileWriter writer = new FileWriter(file);
    writer.write(contentoffile);
    writer.close();
    System.out.println("File created as " + file.getName());
}
catch (IOException e) {
    System.out.println("An error occurred.");
    e.printStackTrace();
}
}

public void delete() {
    File folder = new File("/Users/sowmiyaa/Downloads/phase1Project");
    if (!folder.exists()) {
        folder.mkdirs();
    }
    File[] files = folder.listFiles();
    boolean checker;
    String nameoffile="";
    do {
        System.out.println("Enter the name of the file to delete: ");
        nameoffile = sc.next();
        checker = true;
        for (File file : files) {
            if (nameoffile.equalsIgnoreCase(file.getName())) {
                file.delete(); System.out.println("File deleted...");
                checker = false;
                break;
            }
        }
        if(checker==true)
        {
            System.out.println("File name doesn't exists...");
        }
    }
    while(checker);
}

public void search() {
    File folder = new File("/Users/sowmiyaa/Downloads/phase1Project"); if (!
    folder.exists()) {
        folder.mkdirs();
    }
    File[] files = folder.listFiles();
    boolean checker;
    String nameoffile="";

```

```

do {
System.out.println("Enter the name of the file to search: ");
nameoffile = sc.next();
checker = true;
for (File file : files) {
if (nameoffile.equalsIgnoreCase(file.getName()))
{
System.out.println("File content:");
try (BufferedReader reader = new BufferedReader(new FileReader(file)))
{
String line;
while ((line = reader.readLine()) != null)
{
System.out.println(line);
}
} catch (IOException e) {
System.err.println("Error reading file: " + e.getMessage());
}
checker = false;
break;
}
}
if(checker==true) {
System.out.println("File name doesn't exists...");
}
}
while(checker);
}
}

```

## **EXPLANATION**

Welcome to LockedMe.com Application! - By SOWMIYAA

Type 1 to Retrieve Type 2 to do Operations Type 3 to Exit

Enter your choice:

This is the interface, where user can interact with the application. Here the application gives user 3 choices which includes - to retrieve, to do operations , to exit.

Based on the choice of the user, the working will proceed.

Enter your choice: 2

Type a to add

Type b to delete

Type c to search

Type d to go back to home

If the user choice is 2.

The application will display certain option like - to add, delete, search, to go back home.

Based on the choice the application will proceed.

If it is a, then the application will add a file which the user will enter.

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
a
Enter the name of the file to add: |
sample.txt
Enter the content of the file:
hello everyone
File created as sample.txt

Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
a
Enter the name of the file to add:
sample1.txt
Enter the content of the file:
myself sowmiya
File created as sample1.txt
```

If it is b, then the application will delete the file which the user wants

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
b
Enter the name of the file to delete:
sample1.txt
File deleted...
```

If it is c, then the user will search for the file the user wants, if deleted it will ask to enter another file name to be searched

```
Enter a to add
Enter b to delete
Enter c to search
Enter d to go back to home
c
Enter the name of the file to search:
sample1.txt
File name doesn't exists...
Enter the name of the file to search:
sample.txt
File content:
hello everyone
```

If it is d, then the application will return the user to the main menu

```
Enter a to add  
Enter b to delete  
Enter c to search  
Enter d to go back to home
```

*d*

```
Enter 1 to Retrieve  
Enter 2 to do Operations  
Enter 3 to Exit  
Pick ur choice:
```

If the user types 1, then the application will retrieve all the files present in the folder

```
Enter 1 to Retrieve  
Enter 2 to do Operations  
Enter 3 to Exit  
Pick ur choice:
```

*1*

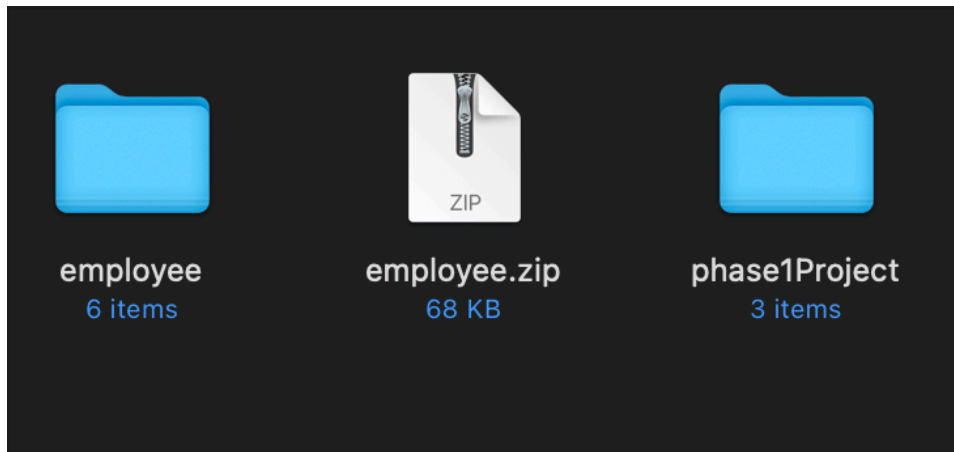
```
.DS_Store  
sample.txt  
sample2.txt
```

If the user types 3, then the application will exit with a thank you message

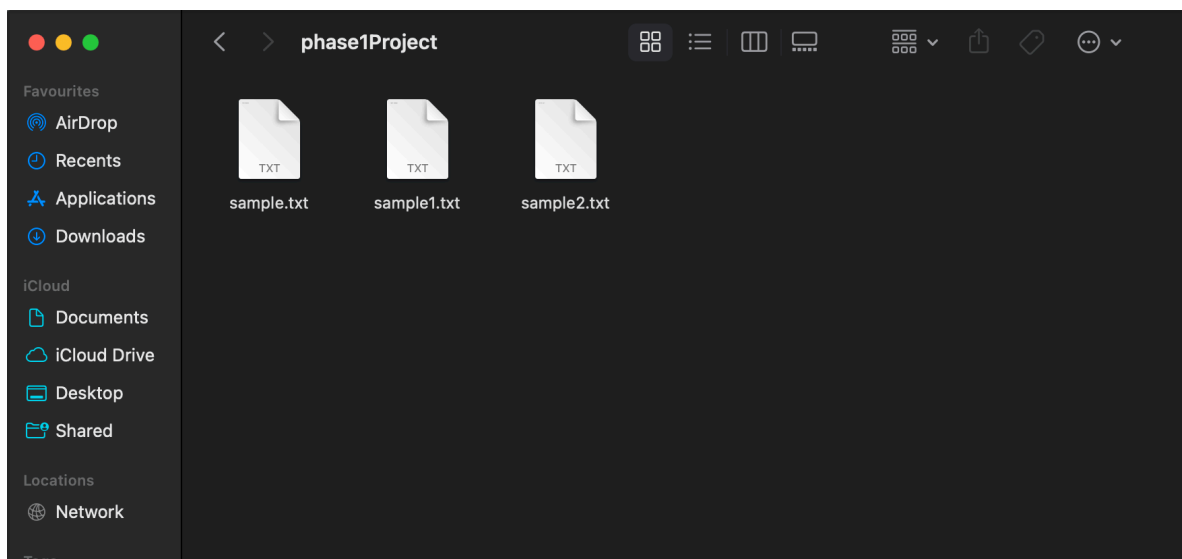
```
Enter 1 to Retrieve  
Enter 2 to do Operations  
Enter 3 to Exit  
Pick ur choice:  
3  
-----  
Thanks for using the app.
```

The folder created is





The files inside the folder is:



The content inside the file:

