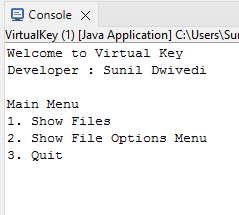
**Virtual Key:**

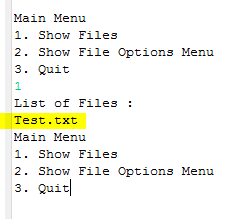
Virtual Key is an app which is used by user to show directory files and also user can create, delete and search a file in a specific directory.

The app is working on command like user can select option to perform the Operations.

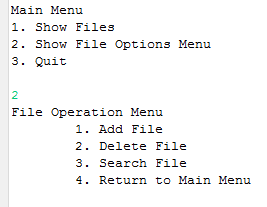
While we will run the app it will display:



When user will enter 1 to show directory files(Location is : D:\VirtualKey) where is a test.txt file is available in a directory.

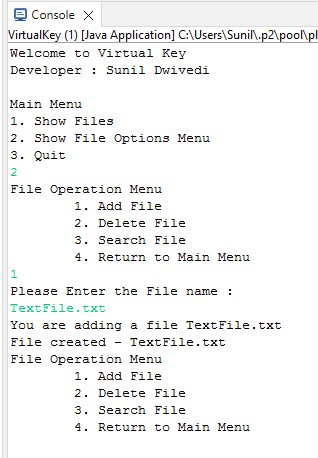


While user will select option 2. It will display four options-

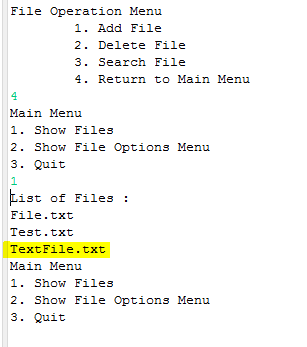


Using the file operation menu user can add, delete and search the file.

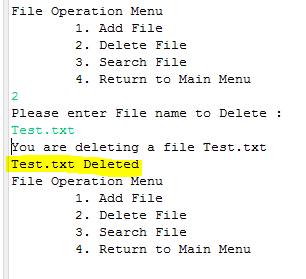
Exa: If user want to create a file, select option 1 it will ask for name of file



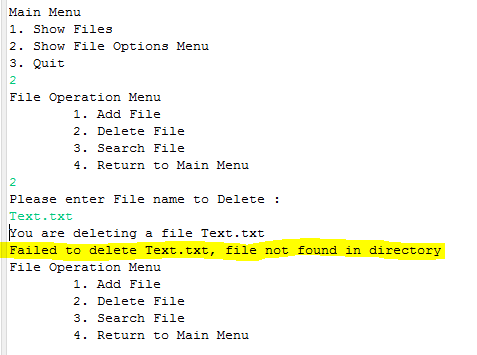
If I will check the directory location TexttFile.txt is available



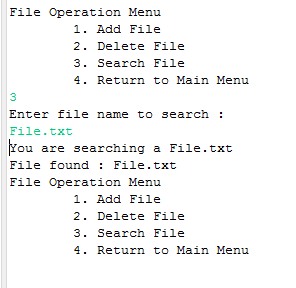
If user want to delete a file press 2 to select delete option and enter the file name:

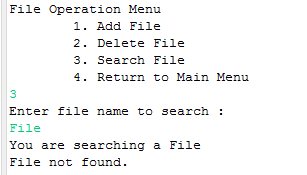


If user will enter wrong file name it will display the File not found.



Same for the searching option, select option 3, it will ask the name of file which you want to search, if file available in the directory it will display file found otherwise it will show file not found.





If, user want to exit from File Operation menu select option four by pressing 4, and if they want to close app select option 3 from main menu.

**Language and IDE Used:**

For developing Virtual Key app I have used Java Programming language to develop app.

Eclipse IDE used for coding.

For developing Virtual Key I have used core Java concept, Loops, Collections, Interface, Method Overloading Method Overriding, to display options used ArrayList also used the Java File Input/Output stream for the all file operation, String and Java Exception Handeling.

For sorting I have used the Collection.sort method to sort the files in ascending order.

**Code for VirtualKey**

**VirtualKey.java**

**package** virtualKeyApp;

**import** welcomePage.Welcome;

**public** **class** VirtualKey {

**public** **static** **void** main(String[] args) {

Welcome obj= **new** Welcome();

obj.intro();

obj.GetUserInput();

}

}

**Directory.java**

package welcomePage;

import java.io.File;

import java.nio.file.FileSystems;

import java.nio.file.Path;

import java.util.ArrayList;

import java.util.Collections;

public class Directory {

//Directory Location

public static final String name = "D:\\VirtualKey\\";

private ArrayList<File> files = new ArrayList<File>();

Path path = FileSystems.getDefault().getPath(name).toAbsolutePath();

File Dfiles = path.toFile();

public String getName() {

return name;

}

public void print() {

System.out.println("Existing Files : ");

files.forEach(f -> System.out.println(f));

}

public ArrayList<File> fillFiles(){

File[] directoryFiles = Dfiles.listFiles();

files.clear();

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

if(directoryFiles[i].isFile()) {

files.add(directoryFiles[i]);

}

}

//Collections.sort method is sorting the elements of ArrayList in ascending order.

Collections.sort(files);

return files;

}

public ArrayList<File> getFiles(){

fillFiles();

return files;

}

}

**DirectoryService.java**

**package** welcomePage;

**import** java.io.File;

**public** **class** DirectoryService {

**private** **static** Directory *fileDirectory* = **new** Directory();

**public** **static** **void** PrintFiles() {

*fileDirectory*.fillFiles();

**for**(File file : DirectoryService.*getFileDirectory*().getFiles()) {

System.***out***.println(file.getName());

}

}

**public** **static** Directory getFileDirectory() {

**return** *fileDirectory*;

}

}

**FileOption.java**

**package** welcomePage;

**import** java.io.File;

**import** java.io.IOException;

**import** java.nio.file.FileSystems;

**import** java.nio.file.Path;

**import** java.util.ArrayList;

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

**public** **class** FileOption **implements** Screen {

**private** Directory dir = **new** Directory();

**private** ArrayList<String> options = **new** ArrayList<>();

**public** FileOption() {

options.add("\t1. Add File");

options.add("\t2. Delete File");

options.add("\t3. Search File");

options.add("\t4. Return to Main Menu");

}

**public** **void** Show() {

System.***out***.println("File Operation Menu");

**for**(String s : options) {

System.***out***.println(s);

}

}

**public** **void** GetUserInput() {

**int** selectedOption;

**while**((selectedOption = **this**.getOption())!=4) {

**this**.NavigateOption(selectedOption);

}

}

@Override

**public** **void** NavigateOption(**int** option) {

**switch**(option) {

**case** 1:

**this**.AddFile();

**this**.Show();

**break**;

**case** 2:

**this**.DeleteFile();

**this**.Show();

**break**;

**case** 3:

**this**.SearchFile();

**this**.Show();

**break**;

**case** 4:

ScreenService.*setCurrentScreen*(ScreenService.*Welcome*);

ScreenService.*getCurrentScreen*().Show();

ScreenService.*getCurrentScreen*().GetUserInput();

**break**;

**default**:

System.***out***.println("Invalid Option");

**break**;

}

}

//While selected Add File Option

**public** **void** AddFile() {

System.***out***.println("Please Enter the File name : ");

String fileName = **this**.getInputString();

System.***out***.println("You are adding a file "+fileName);

**try** {

Path path = FileSystems.*getDefault*().getPath(Directory.***name***+fileName).toAbsolutePath();

File file = **new** File(dir.getName()+fileName);

**if**(file.createNewFile()) {

System.***out***.println("File created - "+file.getName());

dir.getFiles().add(file);

}

**else** {

System.***out***.println("File is Already Exist.");

}

}**catch**(IOException e) {

System.***out***.println(e);

}

}

//While selected DeleteFile Option

**public** **void** DeleteFile() {

System.***out***.println("Please enter File name to Delete : ");

String fileName = **this**.getInputString();

System.***out***.println("You are deleting a file "+fileName);

Path path =FileSystems.*getDefault*().getPath(Directory.***name***+fileName).toAbsolutePath();

File file = path.toFile();

**if**(file.delete()) {

System.***out***.println(file.getName()+" Deleted");

dir.getFiles().remove(file);

}

**else** {

System.***out***.println("Failed to delete "+fileName+", file not found in directory");

}

}

//While selected SearchFile Option

**public** **void** SearchFile() {

Boolean found = **false**;

System.***out***.println("Enter file name to search : ");

String fileName = **this**.getInputString();

System.***out***.println("You are searching a "+fileName);

ArrayList<File> files = dir.getFiles();

**for**(**int** i=0;i<files.size();i++) {

**if**(files.get(i).getName().equals(fileName)) {

System.***out***.println("File found : "+fileName);

found = **true**;

}

}

**if**(found == **false**) {

System.***out***.println("File not found.");

}

}

**private** String getInputString() {

Scanner sc = **new** Scanner(System.***in***);

**return**(sc.nextLine());

}

**private** **int** getOption() {

Scanner sc = **new** Scanner(System.***in***);

**int** returnOption = 0;

**try** {

returnOption = sc.nextInt();

}

**catch**(InputMismatchException ex) {

System.***out***.println("Invalid input");

}

**return** returnOption;

}

}

**Screen.java(Interface)**

**package** welcomePage;

**public** **interface** Screen {

**public** **void** Show();

**public** **void** NavigateOption(**int** option);

**public** **void** GetUserInput();

}

**ScreenService.java**

**package** welcomePage;

**public** **class** ScreenService {

**public** **static** Welcome *Welcome* = **new** Welcome();

**public** **static** FileOption *FileOptionScreen* = **new** FileOption();

**public** **static** Screen *CurrentScreen* = *Welcome*;

**public** **static** Screen getCurrentScreen() {

**return** *CurrentScreen*;

}

**public** **static** **void** setCurrentScreen(Screen currentScreen) {

*CurrentScreen* = currentScreen;

}

}

**Welcome.java**

package welcomePage;

import java.util.ArrayList;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Welcome implements Screen{

private String appName = "Welcome to Virtual Key";

private String devloperName = "Sunil Dwivedi";

private ArrayList<String> options =new ArrayList<>();

public Welcome() {

options.add("1. Show Files");

options.add("2. Show File Options Menu");

options.add("3. Quit");

}

public void intro() {

System.out.println(appName);

System.out.println("Developer : "+devloperName);

System.out.println();

Show();

}

//Override the Show method which is under Screen

@Override

public void Show() {

System.out.println("Main Menu");

for(String s : options) {

System.out.println(s);

}

}

public void GetUserInput() {

int selectedOption = 0;

while((selectedOption = this.getOption())!=3) {

this.NavigateOption(selectedOption);

}

}

@Override

public void NavigateOption(int option) {

switch(option) {

//list files directory

case 1:

this.ShowFiles();

this.Show();

break;

//Show File SubMenu

case 2:

ScreenService.setCurrentScreen(ScreenService.FileOptionScreen);

ScreenService.getCurrentScreen().Show();

ScreenService.getCurrentScreen().GetUserInput();

this.Show();

break;

default:

System.out.println("Invalid Option");

break;

}

}

//List Directory files

public void ShowFiles() {

System.out.println("List of Files : ");

DirectoryService.PrintFiles();

}

private int getOption() {

Scanner sc = new Scanner(System.in);

int returnOption = 0;

try {

returnOption = sc.nextInt();

}

catch(InputMismatchException ex) {

}

return returnOption;

}

}