**Virtual Key for Your Repositories**

**Specification document**

Product’s capabilities, User interactions:

* This application is capable of listing files and folders in the directory folders.
* This application allows the user to add, delete and sort the files of current directory.

**Number and duration of sprints required :**

3 sprints, each of 5 days.

Sprint #1: Planning and Requirement analysis

This stage deals with planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Sprint #2:Design and develop

Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification

Sprint #3:Testing

This stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

**GitHub account**

<https://github.com/chandangowdatp/java-fsd.git>

**Data Structures :**

Sorting algorithm: Bubble sort

begin BubbleSort(list)

for all elements of list

if list[i] > list[i+1]

swap(list[i], list[i+1])

end if

end for

return list

end BubbleSort

Searching algorithm: Binary search

Binary\_Search(a, lower\_bound, upper\_bound, val) // 'a' is the given array, 'lower\_bound' is the index of the first array element, 'upper\_bound' is the index of the last array element, 'val' is the value to search

Step 1: set beg = lower\_bound, end = upper\_bound, pos = - 1

Step 2: repeat steps 3 and 4 while beg <=end

Step 3: set mid = (beg + end)/2

Step 4: if a[mid] = val

set pos = mid

print pos

go to step 6

else if a[mid] > val

set end = mid - 1

else

set beg = mid + 1

[end of if]

[end of loop]

Step 5: if pos = -1

print "value is not present in the array"

[end of if]

Step 6: exit

**Code:**

**import** java.io.IOException;

**import** java.io.File;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** virtualKeyRepo {

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

//Welocome screen

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.***out***.println("\t Welcome to Repository Management\n");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("\t It is a virtual key for your repository");

System.***out***.println("\n\nDeveloped by,\n\tChandangowda T P");

*optionsSelection*();

}

**private** **static** **void** optionsSelection() {

//Main directory

System.***out***.println("\n\n--------- Main Directory -----------\n");

File f = **new** File("C:\\Users\\chandan\\Desktop");

String[] arr = {"1. I wish to list current File",

"2. I wish to perform actions",

"3. I wish to close application"

};

**int**[] arr1 = {1,2,3};

**int** slen = arr1.length;

**for**(**int** i=0; i<slen;i++){

System.***out***.println(arr[i]);

}

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

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

**int** options = sc.nextInt();

**switch** (options){

**case** 1:

System.***out***.println("These are the list of files present in the current directory");

String[] pathnames;

pathnames = f.list();

String[] sortedPaths=*sortFiles*(pathnames);

**for** (String pathname : sortedPaths) {

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

}

*optionsSelection*();

**break**;

**case** 2:

*selectedAction*(f);

**break**;

**case** 3:

*closeApp*();

**break**;

**default**:

System.***out***.println("You have made an invalid choice!");

*optionsSelection*();

**break**;

}

sc.close();

}

**private** **static** **void** selectedAction(File f) {

String[] arr = {"\n\n1.I wish to Add File ",

"2.I wish to Delete File",

"3.I wish to Search File",

"4.I wish to Main context"

};

**int**[] arr1 = {1,2,3,4};

**int** slen = arr1.length;

**for**(**int** i=0; i<slen;i++){

System.***out***.println(arr[i]);

}

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

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

**int** options = in.nextInt();

**switch** (options){

**case** 1:

**try** {

System.***out***.println("\nEnter file name that you want to add:\t");

String name = in.next();

*addFiles*(name);

}**catch**(Exception e){

e.printStackTrace();

}

*selectedAction*(f);

**break**;

**case** 2:

System.***out***.println("\nEnter file name that you want to delete:\t");

String fname2=in.next();

*deleteFiles*(fname2);

*selectedAction*(f);

**break**;

**case** 3:

System.***out***.println("\nEnter file name that you want to search:\t");

String fname3=in.next();

**int** index=*searchFile*(f,fname3);

String res=index > -1 ? fname3+" is present in the sorted list at position : " + (index+1) : fname3+" is is not found in the the list.";

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

*selectedAction*(f);

**break**;

**case** 4:

*optionsSelection*();

**break**;

**default**:

System.***out***.println("You have made an invalid choice!");

*selectedAction*(f);

**break**;

}

in.close();

}

**private** **static** **void** addFiles(String fname) {

File fnew= **new** File("C:\\Users\\chandan\\Desktop\\"+fname);

**try** {

**boolean** result;

result = fnew.createNewFile();

**if**(result)

{

System.***out***.println("file created ");

}

**else**

{

System.***out***.println("File already exist at location: "+fnew.getCanonicalPath());

}

}

**catch**(IOException e){

e.printStackTrace();

}

}

**private** **static** **void** deleteFiles(String fname) {

File fnew= **new** File("C:\\Users\\chandan\\Desktop\\"+fname);

**try** {

**boolean** res=fnew.delete();

**if**(res)

{

System.***out***.println("file deleted ");

}

**else**

{

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

}

}**catch**(Exception e){

e.printStackTrace();

}

}

**private** **static** **int** searchFile(File f,String fname) {

String[] filenames;

filenames = f.list();

String[] sortedPaths=*sortFiles*(filenames);

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

**if** (sortedPaths[i].equals(fname)) {

**return** i;

}

}

**return** -1;

}

**private** **static** **void** closeApp() {

System.***out***.println("Closing your application... \nThank you!");

}

**private** **static** String[] sortFiles(String[] pathnames) {

**int** len = pathnames.length;

//Complete the method. The expenses should be sorted in ascending order.

**for** (**int** i = 0; i < len; i++) {

**for** (**int** j = len - 1; j > i; j--) {

**if** (pathnames[i].compareTo(pathnames[j])>0) {

String tmp = pathnames[i];

pathnames[i]= pathnames[j];

pathnames[j]=tmp;

}

}

}

**return** pathnames;

}

}