package searchapp;

import java.io.BufferedInputStream;

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.IOException;

import java.io.OutputStream;

import java.io.PrintWriter;

import java.io.UnsupportedEncodingException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Set;

import javax.swing.table.DefaultTableModel;

public class IndexModel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

//static instance for singleton

//this will keep in store all the data per text file

public static Map<File , List<String> > indexFileMap = new HashMap<File, List<String>>();

public static Map<File, FileIndexStatus> indexFileStatus = new HashMap<File,FileIndexStatus>();

public static Map<File, Long> indexFileModified = new HashMap<File,Long>();

public static void INITMODEL(){

File f = new File(System.getProperty("user.home") + "/indexFileSearchApp.dat");

if(f.exists()) {

System.out.println("loading file...");

LoadFromFile();}

}

public static void AddFile(File f){

try {

//read the contents of the file

FileInputStream fis = new FileInputStream(f);

// Here BufferedInputStream is added for fast reading.

BufferedInputStream bis = new BufferedInputStream(fis);

DataInputStream dis = new DataInputStream(bis);

//now create a temporary set to put all the extracted data into

List<String> tempSet = new ArrayList<String>();

String tempString = "";

// dis.available() returns 0 if the file does not have more lines.

while (dis.available() != 0) {

// this statement reads the line from the file and print it to

// the console. we know it is presently deprecated and will be actively looking

//for a more efficient solution, in the meantime this works for plain text files

tempString += dis.readLine();

System.out.println("reading line: " + tempString);

}

// dispose all the resources after using them.

fis.close();

bis.close();

dis.close();

//use this to separate all the words, this will separate them by phrase, and from there the searching will be done word by word

String \_delimiters = "[.?!:\\-=+,&()<>@#$%^\\\*~` ]+";

String[] tokens = tempString.split(\_delimiters);

//now each word goes into the set

for(String str : tokens){

tempSet.add(str);

}

// When the temporary set is made, add the new map entry to the map index

indexFileMap.put(f, tempSet);

indexFileStatus.put(f, FileIndexStatus.INDEXED);

indexFileModified.put(f, f.lastModified());

} catch (FileNotFoundException e) {

System.out.println("no file selected");

}catch(IOException e){

System.out.println("no file selected");

}

SaveToFile();

}

public static void RemoveFile(String filename){

if(indexFileMap.keySet() != null)

for(File f : indexFileMap.keySet()){

System.out.println(f.getName());

System.out.println(filename);

if(f.getName()==filename){

Iterator<Map.Entry<File,List<String>>> iter = indexFileMap.entrySet().iterator();

while (iter.hasNext()) {

Entry<File, List<String>> entry = iter.next();

if(filename.equalsIgnoreCase(entry.getKey().getName())){

iter.remove();

}

}

Iterator<Map.Entry<File,Long>> iter1 = indexFileModified.entrySet().iterator();

while (iter1.hasNext()) {

Entry<File, Long> entry = iter1.next();

if(filename.equalsIgnoreCase(entry.getKey().getName())){

iter1.remove();

}

}

Iterator<Map.Entry<File,FileIndexStatus>> iter11 = indexFileStatus.entrySet().iterator();

while (iter11.hasNext()) {

Entry<File, FileIndexStatus> entry = iter11.next();

if(filename.equalsIgnoreCase(entry.getKey().getName())){

iter11.remove();

}

}

}

System.out.println(f.getPath()+", "+indexFileStatus.get(f)+ indexFileModified.get(f)+"\n");

}

SaveToFile();

}

public static void Rebuild(){

if(indexFileMap.keySet() != null)

for(File f : indexFileMap.keySet()){

AddFile(f);

}

SaveToFile();

}

public static void checkIfOutdated(){

//use this to check if files are outdated

if(indexFileStatus.keySet() != null)

for(File f : indexFileMap.keySet()){

//check if the file is deleted

if(!f.exists()){

indexFileMap.remove(f);

indexFileStatus.remove(f);

indexFileModified.remove(f);

}

//check if last modified of file is not equal to current index, if so set it to outdated

else if(f.lastModified() != indexFileModified.get(f)){

indexFileStatus.put(f, FileIndexStatus.OUTDATED);

}

}

SaveToFile();

}

public static String UpdateView(){

//we now fill the fileData with the current info to update the JTable

String tempstring = "";

if(indexFileStatus.keySet() != null)

for(File f : indexFileStatus.keySet()){

tempstring += f.getPath()+", "+indexFileStatus.get(f)+"\n";

System.out.println(f.getPath()+", "+indexFileStatus.get(f)+ indexFileModified.get(f)+"\n");

}

return tempstring;

}

private static void SaveToFile(){

//create file in home directory

String homedir = System.getProperty("user.home");

File save = new File(homedir+"/FileSearchApp.dat");

//store data as CSV of pathname,indexstatus, modified number, path data[each one a CSV value]

PrintWriter writer = null;

try {

writer = new PrintWriter(save, "UTF-8");

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (UnsupportedEncodingException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//writer.println("-----SEARCH APP-----SAVED AS OF-----"+System.currentTimeMillis());

for(File f: indexFileMap.keySet()){

String str = "";

String fileIndexData = "";

for(String s: indexFileMap.get(f)){

fileIndexData += s+",";

}

//writing the data, one line for one file

writer.println(f.getPath()+","+indexFileStatus.get(f)+","+indexFileModified.get(f)+","+fileIndexData);

}

writer.close();

}

private static void LoadFromFile(){

//Load data as CSV of pathname,indexstatus, modified number, path data[each one a CSV value]

String \_delimiters = "[,]+";

String dir = System.getProperty("user.home") + "/indexFileSearchApp.dat";

try {

BufferedReader reader = new BufferedReader(new FileReader(dir));

String line;

while ((line = reader.readLine()) != null)

{

//if(!line.contains("-----SEARCH APP-----SAVED AS OF-----"))

String[] parameters = line.split(\_delimiters);

//do file load work here

String path = parameters[0];

String IndexStatus = parameters[1];

Long Modified = Long.parseLong(parameters[2]);

File f = new File(path);

List<String> fileIndexData = new ArrayList<>();

for(int i =3; i < parameters.length-1;i++ ){

fileIndexData.add(parameters[i]);

}

System.out.println("loading file data: "+ fileIndexData);

indexFileMap.put(f, fileIndexData);

indexFileModified.put(f, Modified);

indexFileStatus.put(f, FileIndexStatus.valueOf(IndexStatus));

}

reader.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}