

<Medical File Tagging Program - 2016>

# Medical File Tagging Program — 2016

</Medical File Tagging Program — 2016>

A Console Application

Copyright (C) 2016. All Rights Reserved. Zakia Salod

INFT8F2H2 - PROGRAMMING MEDICAL INFORMATICS SYSTEMS

ASSIGNMENT 01 : SOFTWARE : MEDICAL FILE TAGGING PROGRAM - 2016

208501583 - Zakia Salod - zakia.salod@ukzn.ac.za

This document serves as documentation / an overview of the Software developed as part of Assignment 01 of this module. It includes an "About" of this program, the Technologies Used, Sample Run of Program, Source Code, & About the Programmer.

| Faculty of Health Sciences |

| School of Nursing & Public Health |

| University of Kwa-Zulu Natal, Durban, South Africa |



## INFT8F2H2 - PROGRAMMING MEDICAL INFORMATICS SYSTEMS

ASSIGNMENT 01: SOFTWARE: MEDICAL FILE TAGGING PROGRAM - 2016

#### ABOUT: MEDICAL FILE TAGGING PROGRAM - 2016 . . .

#### WHAT IS IT?

The Medical File Tagging Program - 2016 is a software used to tag keywords that are found in a text file. The program reads-in a text file, for example, a medical journal article, Doctor's notes and finds pre-defined keywords and tags them. The pre-defined keywords are read from another text file.

This program was created by the programmer, as part of a Medical Informatics Programming course at the University of KwaZulu-Natal Nelson R Mandela School of Medicine campus in South Africa in the year 2016.

#### **NOTE**

#### **INPUT:**

Sample Medical Text File can be found here:

o medicalinputfiles\sampleinputtxtfile.txt

Sample Medical Keywords Text File can be found here:

o medicalinputfiles\sampleinputtermsfile.txt

#### **OUTPUT:**

#### **DIRECTORY:**

- o Output file/s are stored in the "medicaloutputfiles" directory.
- o If the "medicaloutputfiles" directory does not exist, the program will create this directory for the first run of this program.
- o Subsequent runs of the program will store output files onto this directory.

#### FILE/S:

o The user is prompted for the output file name. If the file name already exists in the "medicaloutputfiles" directory, the program continues to prompt the user for a unique file name.

#### **ABSTRACT**

This document serves as documentation / an overview of the Software developed as part of Assignment 01 of this module. It includes an "About" of this program, the **Technologies** Used, Sample Run of Program, Source Code, & About the Programmer.

| NTACT |   |
|-------|---|
|       | o If you would like more info about the Medical File Tagging Program,   |
|       | or require free support for this program, please contact the programmer |
|       | at http://www.zakiasalod.weebly.com or <u>zakia.salod@gmail.com</u>     |
|       |   |
|       | Copyright (C) 2016. All Rights Reserved. Zakia Salod                    |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |
|       |   |

### TECHNOLOGIES USED . . .

- ✓ Programming Language : Java
- $\checkmark$  Integrated Development Environment (IDE) : Eclipse LUNA
- ✓ JRE: 1.7
- ✓ JDK : 1.7

#### SAMPLE RUN OF PROGRAM . . .

#### **INPUT**

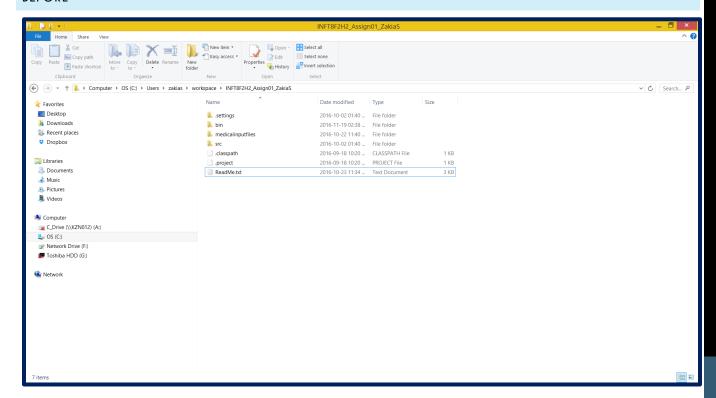
#### MEDICALINPUTFILES\SAMPLEINPUTTERMSFILE.TXT

Human immunodeficiency virus, tuberculosis, population, disease, virus

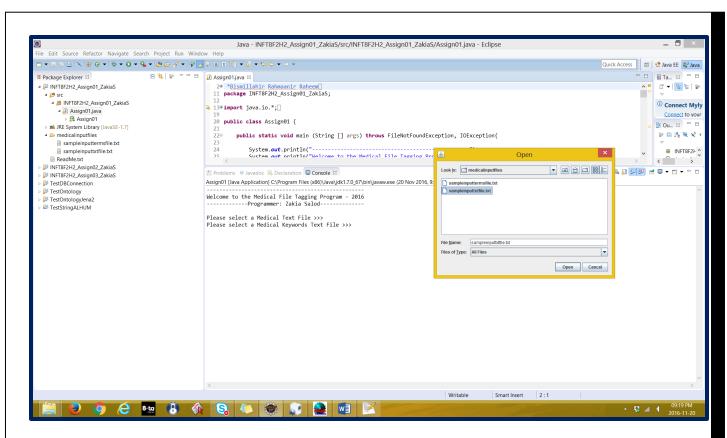
#### MEDICALINPUTFILES\SAMPLEINPUTTXTFILE.TXT

The Human immunodeficiency virus (HIV), tuberculosis and malaria together form a triple burden of disease for resource poor countries like those in Africa. There are currently almost 5.6 million people infected with HIV in South Africa, which is approximately 11% of the South African population.

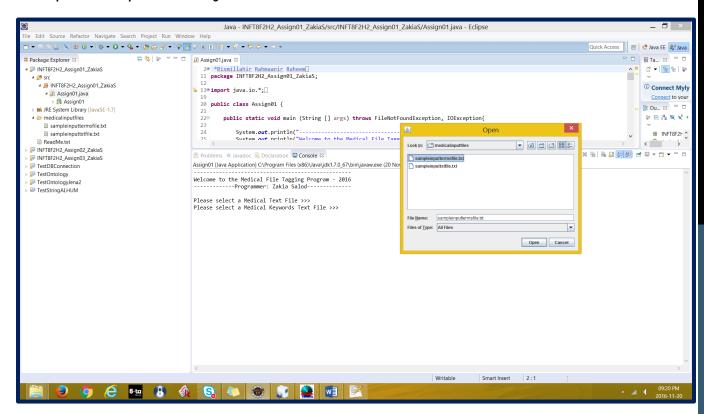
#### **BEFORE**



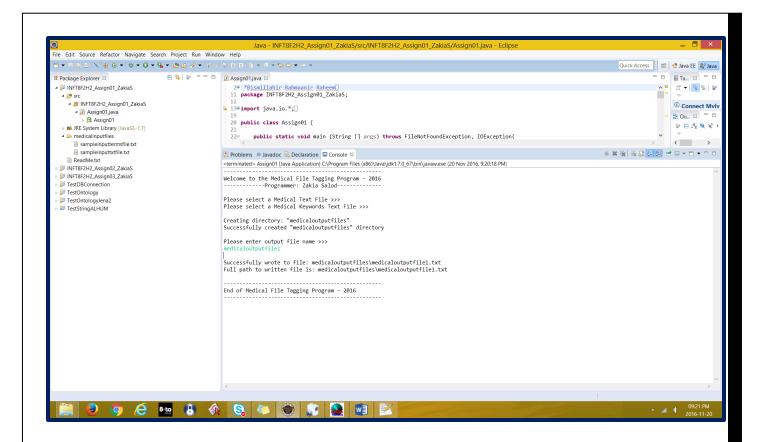
#### **JOURNAL**



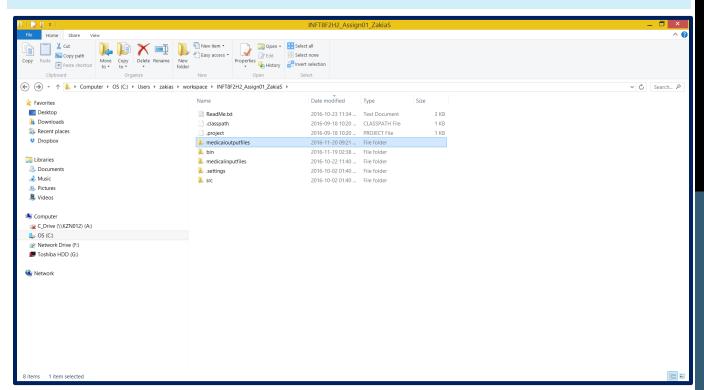
#### Click 'Open' on the 'Open' modal dialog box



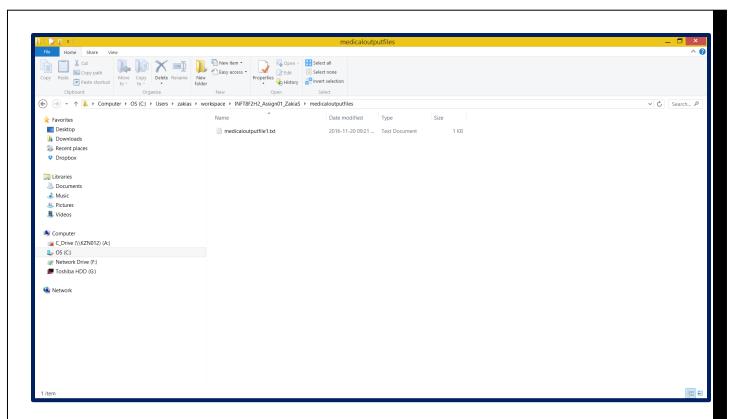
Click 'Open' on the 'Open' modal dialog box



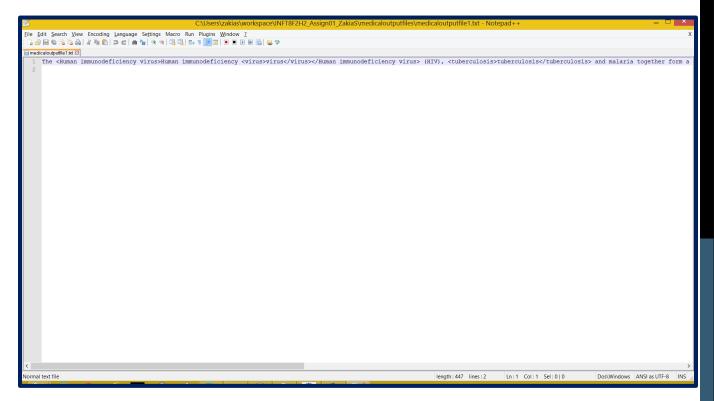
#### **AFTER**



#### Click 'medicaloutputfiles'



#### Open 'medicaloutputfile 1.txt'



#### Full-text copy-pasted from above file:

The <human immunodeficiency virus>Human immunodeficiency <virus>virus>/virus></human immunodeficiency virus> (HIV), <tuberculosis>tuberculosis</tuberculosis> and malaria together form a triple burden of <disease>disease</disease> for resource poor countries like those in Africa. There are currently almost 5.6 million people infected with HIV in South Africa, which is approximately 11% of the South African <population>population>,

#### ASSIGN01.JAVA

```
*Bismillahir Rahmaanir Raheem
*Almadadh Ya Gause Radi Allahu Ta'alah Anh - Ameen
*Student Number : 208501583
*Name : Zakia Salod
*Course : INFT8F2H2
*Assignment : 01
*Masters of Medical Science - Medical <u>Informatics</u>
*Year : 2016
package INFT8F2H2_Assign01_ZakiaS;
import java.io.*;
import java.nio.channels.FileChannel;
import java.nio.channels.FileLock;
import java.nio.channels.OverlappingFileLockException;
import javax.swing.JFileChooser;
import java.util.*;
public class Assign01 {
     public static void main (String [] args) throws FileNotFoundException, IOException{
          System.out.println("-----");
        System.out.println("Welcome to the Medical File Tagging Program - 2016");
        System.out.println("-----\n");
        System.out.println("Please select a Medical Text File >>>");
          System.out.println("Please select a Medical Keywords Text File >>>");
        JFileChooser fileChooser1 = new JFileChooser();
        JFileChooser fileChooser2 = new JFileChooser();
        Scanner input1 = null;
        Scanner input2 = null;
       //>>>>>>>> Prompt User To Choose Medical Text File & Medical
```

```
if(fileChooser1.showOpenDialog(null) == JFileChooser.APPROVE_OPTION &&
fileChooser2.showOpenDialog(null) == JFileChooser.APPROVE_OPTION){
                  File sourceInputTxtFile01 = fileChooser1.getSelectedFile();
                  File sourceInputTermsFile01 = fileChooser2.getSelectedFile();
                  try {
                         input1 = new Scanner(sourceInputTxtFile01);
                      input2 = new Scanner(sourceInputTermsFile01);
                      String strInput1 = "";
                      String strInput2 = "";
                      if (input1.hasNextLine()){
                             strInput1 = input1.nextLine();
                      }
                      if (input2.hasNextLine()){
                             strInput2 = input2.nextLine();
                      }
                         String [] stringsTerms = strInput2.split(", ");
                         String checkStringDouble1 = "";
                         String checkStringDouble2 = "";
                         boolean printToTargetFile = false;
                         if (!sourceInputTxtFile01.exists()){
                                     System.out.println("Source text file " +
sourceInputTxtFile01 + " does not exist");
                                     System.exit(0);
                         }//end if
                         else if (!sourceInputTermsFile01.exists()){
                                     System.out.println("Source terms file " +
sourceInputTermsFile01 + " does not exist");
                                     System.exit(0);
                         }//end else if
                         //>>>>> Read-In Medical Keywords Text File &
else {
                                     for (int i=0; i<stringsTerms.length; i++){</pre>
```

```
(strInput1.toLowerCase().contains(stringsTerms[i].toLowerCase())){
                                                  checkStringDouble1 = "
<"+stringsTerms[i]+">" + stringsTerms[i] + "</"+stringsTerms[i]+">>";
                                                  checkStringDouble2 = "
<<"+stringsTerms[i]+">" + stringsTerms[i] + "</"+stringsTerms[i]+">";
                                                  strInput1 =
strInput1.replace(stringsTerms[i], "<"+stringsTerms[i]+">" + stringsTerms[i] +
"</"+stringsTerms[i]+">");
(strInput1.toLowerCase().contains(checkStringDouble1.toLowerCase())){
                                                     strInput1 =
strInput1.replace(checkStringDouble1, " "+stringsTerms[i]+">");
(strInput1.toLowerCase().contains(checkStringDouble2.toLowerCase())){
strInput1.replace(checkStringDouble2, " "+stringsTerms[i]+"<");</pre>
                                                  }
                                           }//end if
                                     }//end for
                                     printToTargetFile = true;
                         }//end else
                         //>>>>>> Create Medical Output Directory If
File medicalOutputDir = new File("medicaloutputfiles");
                         if (!medicalOutputDir.exists()){
                               System.out.println("\nCreating directory: "+ '"' +
"medicaloutputfiles" + '"');
                               medicalOutputDir.mkdir();
                               System.out.println("Successfully created " + '"' +
"medicaloutputfiles" + '"' + " directory");
                         }
                         else{
                               System.out.println("Output file/s will be written to
directory: "+ '"' + "medicaloutputfiles" + '"');
                         }
                         //>>>>> Write to File >>>>>>>>>>>>>>
                         if(printToTargetFile == true && medicalOutputDir.exists() &&
medicalOutputDir.isDirectory()){
                               String outputFileName = "";
```

```
System.out.println("\nPlease enter output file name >>>
");
                              Scanner getOutputFileName = new Scanner(System.in);
                              outputFileName = getOutputFileName.nextLine();
                                 File targetOutputTxtFile;
                              boolean uniqueOutputFileName = true;
                                 targetOutputTxtFile = new
File(medicalOutputDir+"/"+outputFileName.trim()+".txt");
                               while (targetOutputTxtFile.exists()){
                                     uniqueOutputFileName = false;
                                     System.out.println("Target text file: " +
targetOutputTxtFile + " already exists");
                                              System.out.println("Please enter a different
output file name >>> ");
                                           outputFileName = getOutputFileName.nextLine();
                                              targetOutputTxtFile = new
File(medicalOutputDir+"/"+outputFileName.trim()+".txt");
                                            if(!targetOutputTxtFile.exists()){
                                              uniqueOutputFileName = true;
                                            }//end if
                                  }//end while
                               if (uniqueOutputFileName == true) {
                                  PrintWriter output = null;
                                  try{
                                        output = new PrintWriter(targetOutputTxtFile);
                                        output.println(strInput1);
                                        System.out.println("\nSuccessfully wrote to file: "
+ targetOutputTxtFile);
                                        System.out.println("Full path to written file is: "
+ targetOutputTxtFile);
                                        System.out.println();
                                  }//end try
                                        catch(FileNotFoundException e){
                                               e.printStackTrace();
                                        }//end catch
                                  finally{
                                           //close the file
```

```
if (output !=null)
                                                  output.close();
                            }//end finally
                          }//end if
                      }//end if
                }//end try
                catch(FileNotFoundException e){
                      e.printStackTrace();
                }//end catch
                finally{
                      //close the files
                      if (input1 !=null)
                         input1.close();
                      if (input2 !=null)
                         input2.close();
                       System.out.println("-----
----");
                       System.out.println("End of Medical File Tagging Program -
2016");
                       System.out.println("-----
----");
                }//end finally
       }//end if
   }//end main()
}//end class Assign01
```

| ABOUT THE PROGRAMMER  |  |  |
|---|--|--|
|   |  |  |
| Zakia Salod was born on December 24th 1989 in Durban. She is currently studying full-time towards her Masters in Medical Science Medical Informatics degree at the faculty of Health Sciences at UKZN. She is also working full-time as a Software Developer at a software company, 2Cana Solutions in La Lucia Ridge, Durban – on the Momentum medical aid system. |  |  |
| She graduated with a BSc in Computer Science and IS&T at UKZN in 2010. She had also graduated with a BCom IT Honours (Cum Laude) degree at UKZN in 2011, with first position in her degree from both Westville and Pietermaritzburg campuses.   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
| END OF DOCUMENT   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |