<Medical File & Database Tagging Program – 2016>

Medical File & Database Tagging Program – 2016

</Medical File & Database Tagging Program – 2016>

A Console Application

INFT8F2H2 - PROGRAMMING MEDICAL INFORMATICS SYSTEMS

ASSIGNMENT 02 : SOFTWARE : MEDICAL FILE & DATABASE 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 02 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 02 : SOFTWARE : MEDICAL FILE & DATABASE TAGGING PROGRAM - 2016

ABSTRACT

This document serves as documentation / an overview of the Software developed as part of Assignment 02 of this module. It includes an "About" of this program, the Technologies Used, Sample Run of Program, Source Code, & About the Programmer.

ABOUT : MEDICAL FILE & DATABASE TAGGING PROGRAM – 2016 . . .

WHAT IS IT?

The Medical File & Database 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 a database (medassign02) table (medkeyword).

After tagging, the result is stored onto another text file (name of this file input by user).

The result/s of the count of number of occurrences of the Medical terms from the input file, are stored onto a database (medassign02) table (medpaperkeycount).

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 :

- medicalinputfiles\sampleinputtxtfile.txt

Sample Medical Keywords From Database (medassign02) :

- table = medkeyword

OUTPUT :

DIRECTORY :

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

FILE/S :

- 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.

DATABASE TABLE/S:

o Number of occurrences of each term from the medkeyword database table, found in input file medicalinputfiles\sampleinputtxtfile.txt, is stored onto database table medpaperkeycount. Upon first execution of the program, this database table is created. Subsequent runs, stores / inserts data onto this table, by first deleting the current corresponding row for it, from this table. This ensures up-to-date data is stored for each input file, onto this table - i.e. cater for multiple runs of the same input file/s.

CONTACT

o If you would like more info about the Medical File & Database Tagging Program, or require free support for this program, please contact the programmer at http://www.zakiasalod.weebly.com or zakia.salod@gmail.com

---

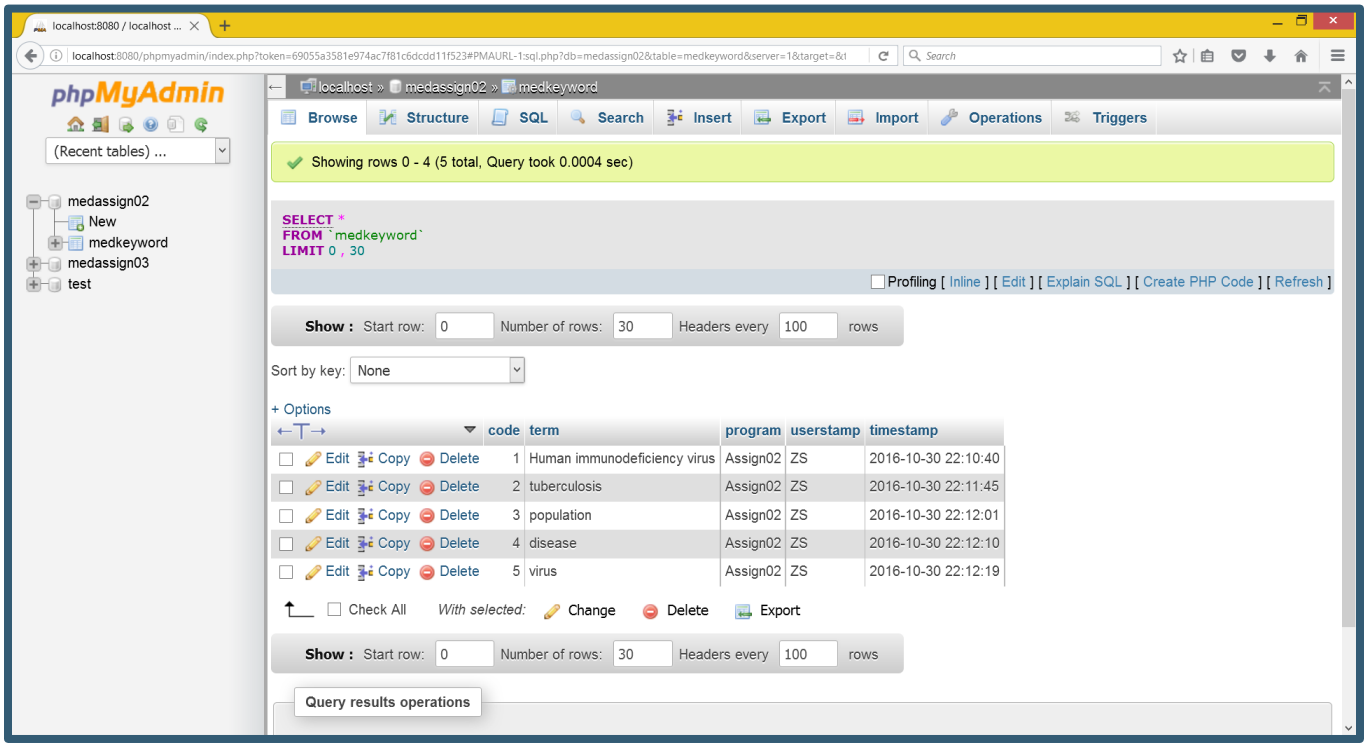
TECHNOLOGIES USED . . .

- ✓ Programming Language : Java
- ✓ Integrated Development Environment (IDE) : Eclipse LUNA
- ✓ JRE : 1.7
- ✓ JDK : 1.7
- ✓ Webservice : USBWebserver v8.6
- ✓ Database : MySQL
- ✓ Mysql-connector : mysql-connector-java-5.1.40-bin.jar

SAMPLE RUN OF PROGRAM . . .

INPUT

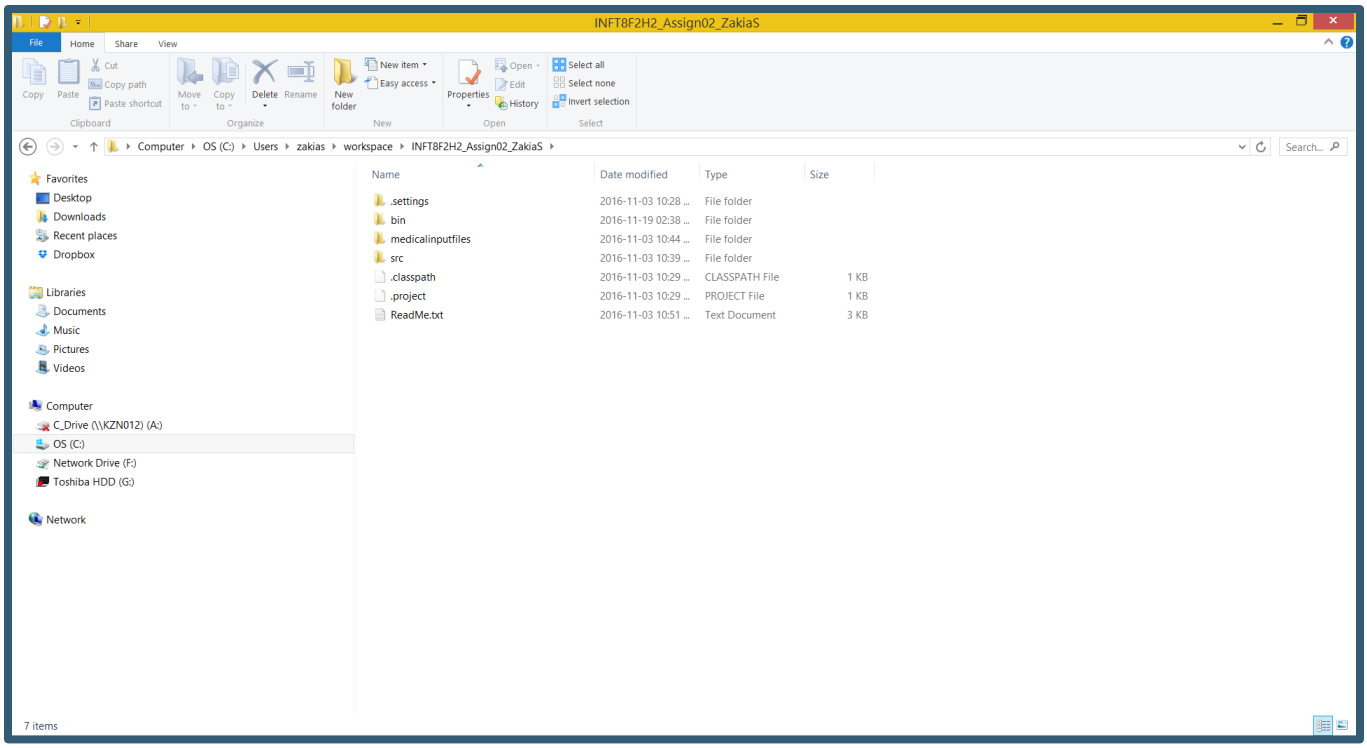
DATABASE TABLE : MEDKEYWORD



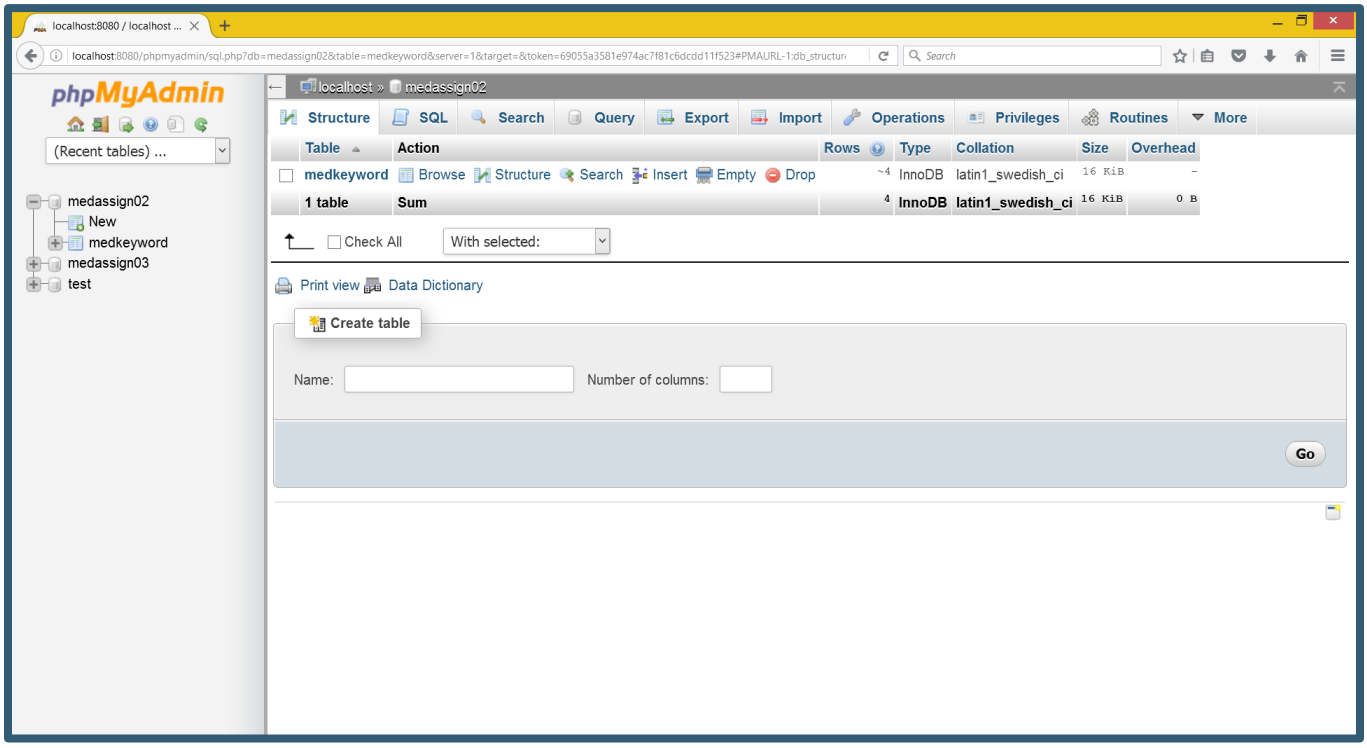
MEDICALINPUTFILES\SAMPLEINPUTTXTFILE.TXT

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.

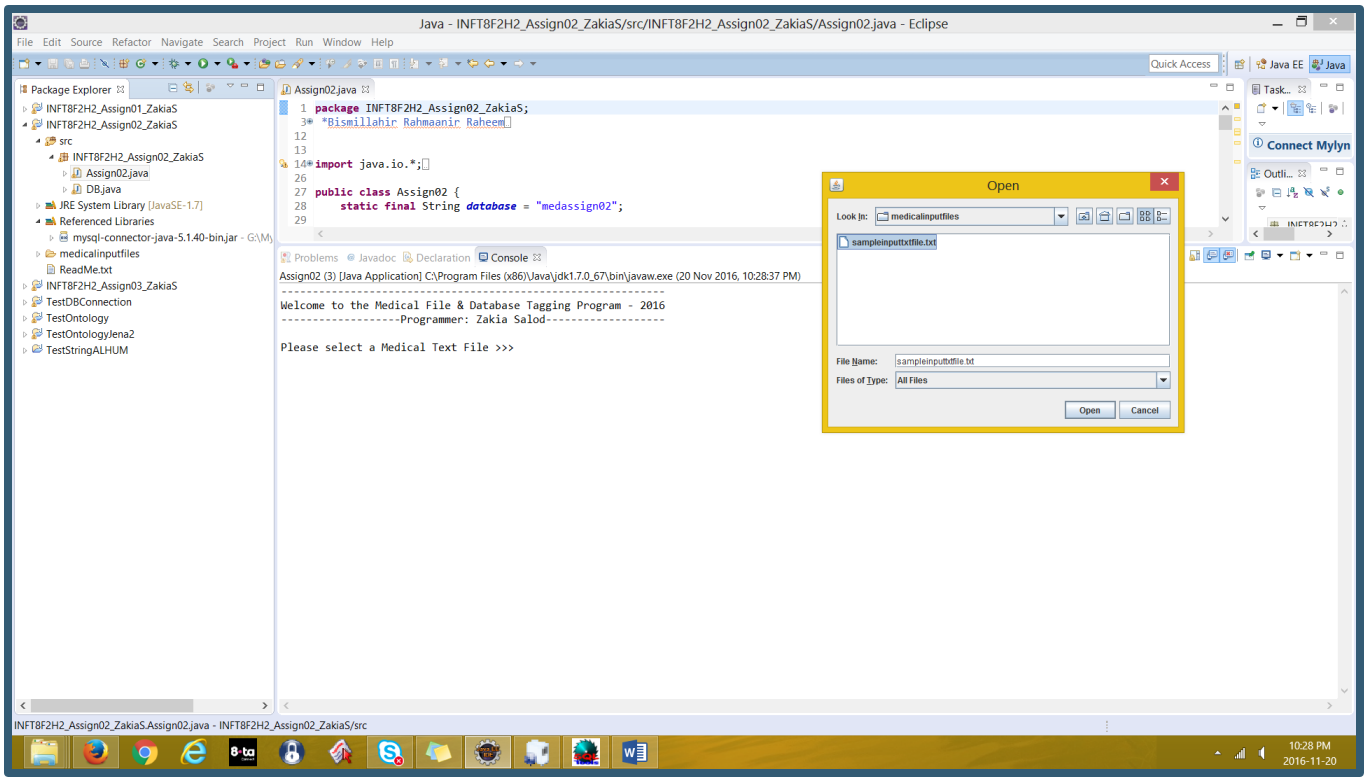
BEFORE



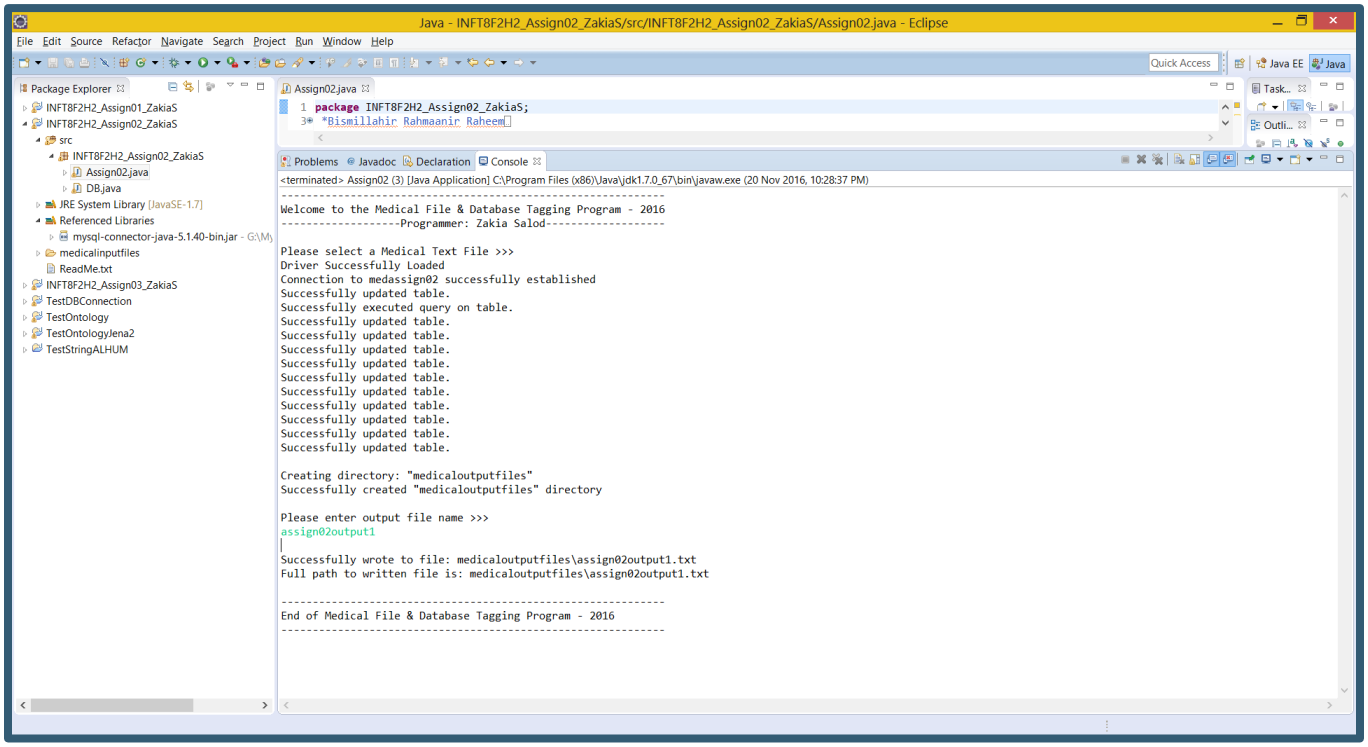
medpaperkeycount Database Table Does NOT Exist in medassign02 Database



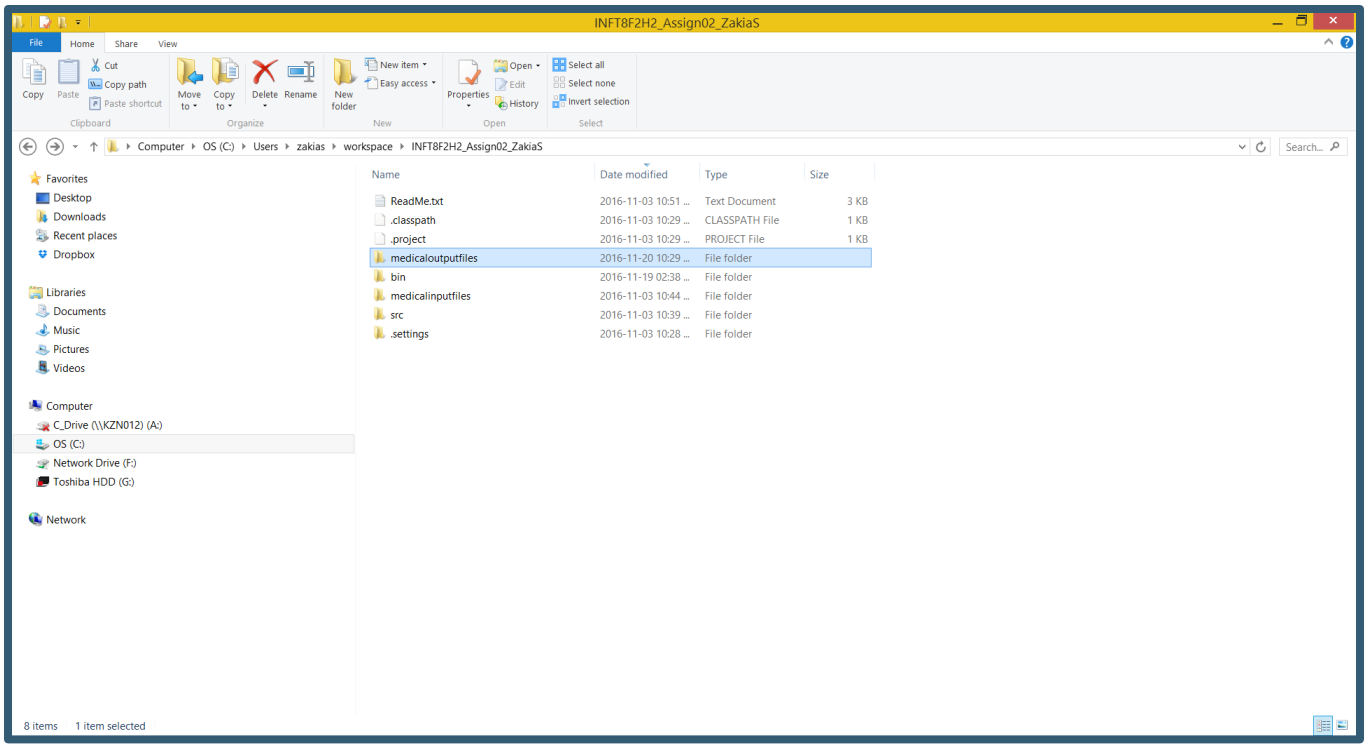
JOURNAL



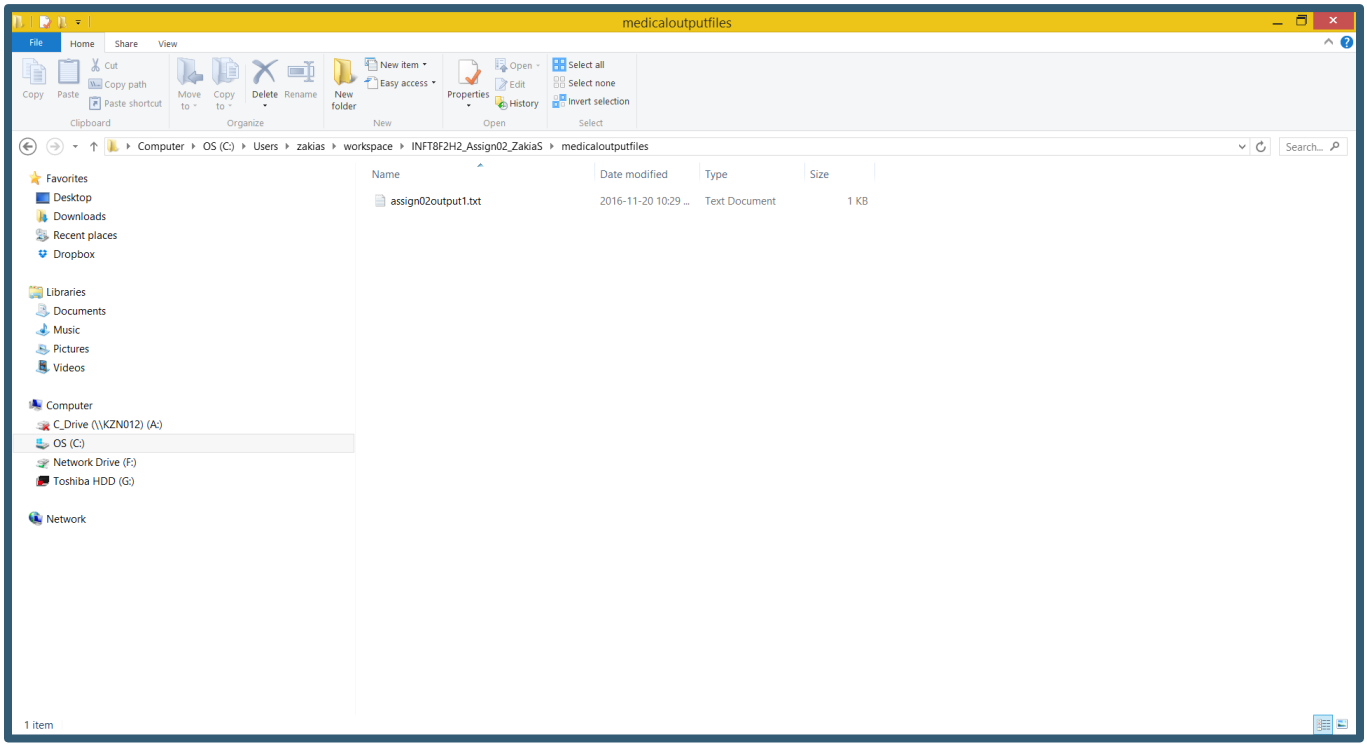
Click 'Open' on the 'Open' modal dialog box



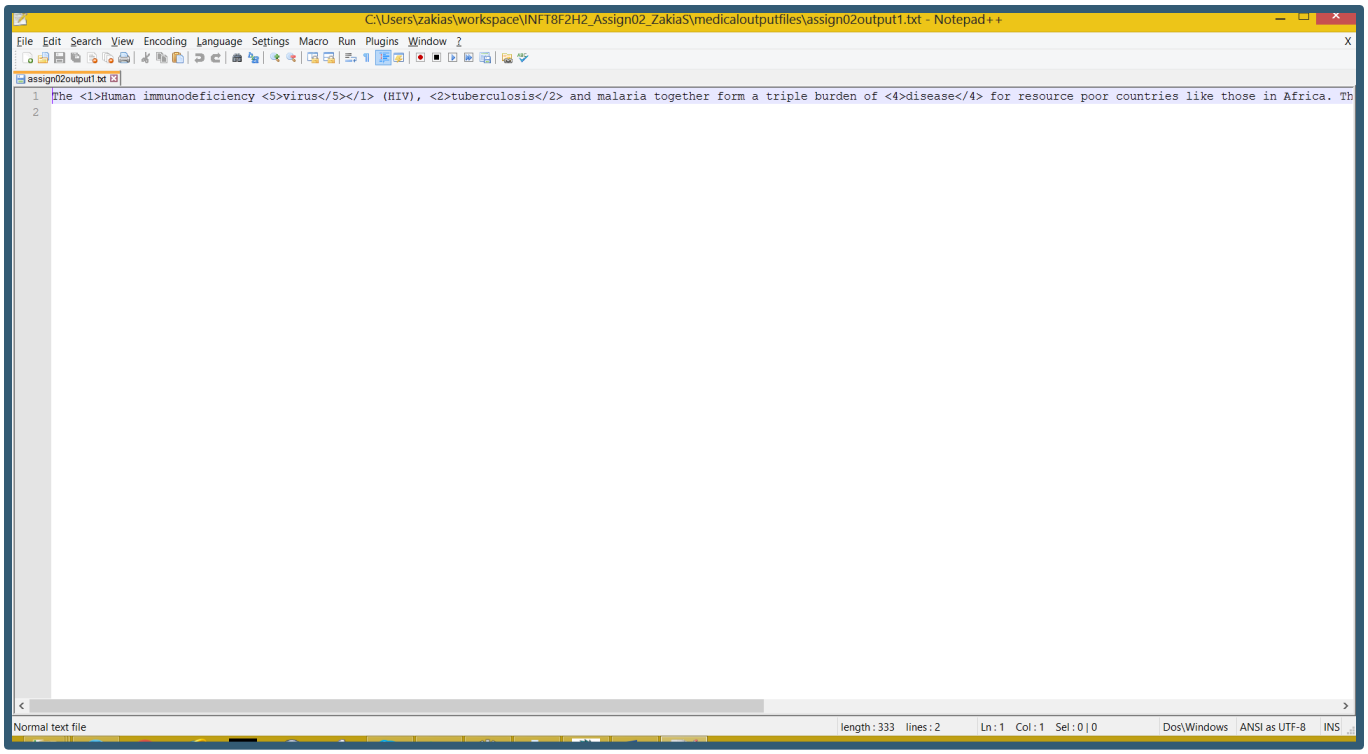
AFTER



Click 'medicaloutputfiles'



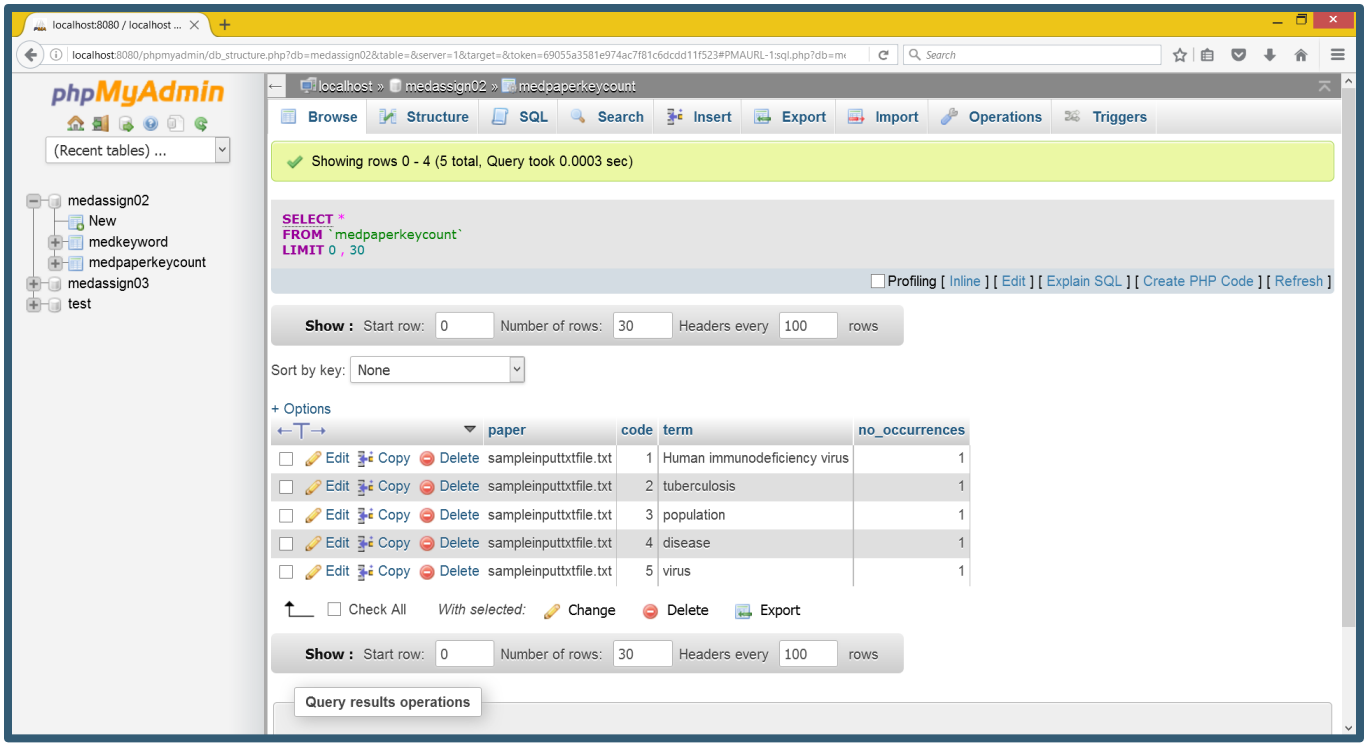
Open 'assign02output1.txt'



Full-text copy-pasted from above file :

The <1>Human immunodeficiency <5>virus</5></1> (HIV), <2>tuberculosis</2> and malaria together form a triple burden of <4>disease</4> 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 <3>population</3>.

medpaperkeycount Database Table DOES Exist in medassign02 Database



SOURCE CODE . . .

ASSIGN02.JAVA

```
package INFT8F2H2_Assign02_ZakiaS;
/*****************************************************************************
 *Bismillahir Rahmaanir Raheem
 *Almadadh Ya Gause Radi Allahu Ta'alah Anh - Ameen
 *Student Number : 208501583
 *Name : Zakia Salod
 *Course : INFT8F2H2
 *Assignment : 02
 *Masters of Medical Science - Medical Informatics
 *Year : 2016
 *****************************************************************************/


import java.io.*;
import java.nio.channels.FileChannel;
import java.nio.channels.FileLock;
import java.nio.channels.OverlappingFileLockException;

import javax.swing.JFileChooser;

import java.sql.*;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.io.*;

public class Assign02 {
    static final String database = "medassign02";

      public static void main (String [] args) throws FileNotFoundException, IOException,
SQLException{

            System.out.println("-------------------------------------------------------
---");
          System.out.println("Welcome to the Medical File & Database Tagging Program -
2016");
          System.out.println("-------------------Programmer: Zakia Salod-------------------
\n");

          System.out.println("Please select a Medical Text File >>>");
          JFileChooser fileChooser1 = new JFileChooser();
          Scanner input1 = null;

         //>>>>>>>>>>>>>>>>>>>>>>>  Prompt User To Choose Medical Text File
>>>>>>>>>>>>>>>>>>>>>>>>
          if(fileChooser1.showOpenDialog(null) == JFileChooser.APPROVE_OPTION){

                  File sourceInputTxtFile01 = fileChooser1.getSelectedFile();

                  try {
                        input1 = new Scanner(sourceInputTxtFile01);
```

```
                        String strInput1 = "";
                        String strInput2 = "";
                        if (input1.hasNextLine()){
                               strInput1 = input1.nextLine();
                        }


                           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


                           //>>>>>>>>>>>>>>>>>>>>>>>  Read-In Medical Keywords Text from
Database & Tag Medical Text File >>>>>>>>>>>>>>>>>>>>>>>
                           //>>>>>>>>>>>>>>>>>>>>>>>  Also insert into table as well too
>>>>>>>>>>>>>>>>>>>>>>>
                           else {
                                  try{
                                        //create output database table here
                                        //createDbTable();
                                        DB connectToAssign02Db = new DB(database);
                                        //create output database table here
                                        createDbTable(connectToAssign02Db);
                                        ResultSet resultSet =
connectToAssign02Db.queryTbl("SELECT * FROM medkeyword");
                                        ResultSetMetaData metaData =
resultSet.getMetaData();
                                        int numberOfColumns = metaData.getColumnCount();
                                        //loop through database table medkeyword, and tag
text, and insert into database table
                                        while(resultSet.next()){
                                                      if
(strInput1.toLowerCase().contains(resultSet.getString("term").toLowerCase())){
                                                                  checkStringDouble1 =  "
<"+resultSet.getString("code")+">" + resultSet.getString("code") +
"</"+resultSet.getString("code")+">>";
                                                                  checkStringDouble2  = "
<<"+resultSet.getString("code")+">" + resultSet.getString("code") +
"</"+resultSet.getString("code")+">";
                                                                  strInput1 =
strInput1.replace(resultSet.getString("term"), "<"+resultSet.getString("code")+">" +
resultSet.getString("term") + "</"+resultSet.getString("code")+">");

                                                                  //insert into database
output table
                                                                  int getNumKeyOccurrences =
0;
                                                                  getNumKeyOccurrences =
getNumStrOccurrences(strInput1, resultSet.getString("term"));


      deletemedpaperkeyentry(connectToAssign02Db, "medpaperkeycount",
```

```
sourceInputTxtFile01.getName(), Integer.parseInt(resultSet.getString("code")),
resultSet.getString("term"), getNumKeyOccurrences);

      insertmedpaperkeycount(connectToAssign02Db, "medpaperkeycount",
sourceInputTxtFile01.getName(), Integer.parseInt(resultSet.getString("code")),
resultSet.getString("term"), getNumKeyOccurrences);
                                                      }//end if
                                    }//end while

                                          printToTargetFile = true;
                                    }//end try
                                    catch(SQLException e){
                                          System.out.println("");
                                    }//end catch

                              }//end else

                              //>>>>>>>>>>>>>>>>>>>>>>>  Create Medical Output Directory If
Does Not Exist >>>>>>>>>>>>>>>>>>>>>>>>
                              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();

                                System.out.println("-----------------------------------------
-----------------");
                                System.out.println("End of Medical File & Database Tagging
Program - 2016");
                                     System.out.println("-------------------------------------
-----------------------");
                        }//end finally
                }//end if
            }//end main()


               private static void createDbTable(DB connectToAssign02Db){

                        try{
                            String createMedPaperKeyCount = "CREATE TABLE medpaperkeycount(paper
varchar(250) NOT NULL, "+
                                                             "code int(11) NOT NULL, term
varchar(255) NOT NULL, no_occurrences int(11) NULL NULL, PRIMARY KEY(paper, code, term))";

                            connectToAssign02Db.updateTbl(createMedPaperKeyCount);
                        }
                        catch(SQLException e){
```

```
                        System.out.println("Table already exists in the " + database + "
database");
                }
        }//end createDbTable()

        private static int getNumStrOccurrences(String fileStr, String findStr){
                int counter = 0;

                Pattern p = Pattern.compile(findStr);
                Matcher m = p.matcher(fileStr);

                while(m.find()){
                        counter++;
                }
                return counter;
        }//end getNumStrOccurrences()

        private static void deletemedpaperkeyentry(DB connectToAssign02Db, String tableName,
String paper, int code, String term, int no_occurrences){

                try{
                        connectToAssign02Db.updateTbl("DELETE FROM " + tableName +" WHERE
paper='" + paper +"' AND code=" + code + " AND term='" + term + "' AND no_occurrences =" +
no_occurrences);
                }
                catch(SQLException e){
                        e.printStackTrace();
                }
        }//end deletemedpaperkeyentry()

        private static void insertmedpaperkeycount(DB connectToAssign02Db, String tableName,
String paper, int code, String term, int no_occurrences){

                try{
                        connectToAssign02Db.updateTbl("INSERT INTO " + tableName +"(paper, code,
term, no_occurrences) VALUES('"+paper+"'," + code + ",'" + term + "'," +
no_occurrences+")");
                }
                catch(SQLException e){
                        e.printStackTrace();
                }
        }//end insertmedpaperkeycount()
}//end class Assign02
```

DB.JAVA

```
package INFT8F2H2_Assign02_ZakiaS;
/*****************************************************************************
 *Bismillahir Rahmaanir Raheem
 *Almadadh Ya Gause Radi Allahu Ta'alah Anh - Ameen
 *Student Number : 208501583
 *Name : Zakia Salod
 *Course : INFT8F2H2
 *Assignment : 02
 *Masters of Medical Science - Medical Informatics
 *Year : 2016
 *****************************************************************************/

import java.io.*;
import java.sql.*;

public class DB {
	Connection conn;

	DB(String database){
		final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
		final String DATABASE_URL = "jdbc:mysql://localhost:3307/"+database;

		//connect to mysql driver
		try{
			Class.forName(JDBC_DRIVER);
			System.out.println("Driver Successfully Loaded");
		}//end try
		catch(ClassNotFoundException e){
			System.out.println("Unable to connect");
			System.exit(1);
		}//end catch


		try{
			conn = DriverManager.getConnection(DATABASE_URL, "root", "usbw");
			System.out.println("Connection to " + database + " successfully
established" );
		}//end try
		catch(Exception e){
			System.out.println(e.getMessage());
		}//end catch
	}//end DB() Constructor



	//Method executes SQL queries, input as string argument
	ResultSet queryTbl(String sqlStmt) throws SQLException{
		Statement stmt = conn.createStatement();
		ResultSet rs = stmt.executeQuery(sqlStmt); // select * from a table
		System.out.println("Successfully executed query on table.");
		return rs;
	}



	void updateTbl(String update) throws SQLException{
```

```
            Statement stmt = conn.createStatement();
            stmt.executeUpdate(update);
            System.out.println("Successfully updated table.");
            stmt.close();
      }



      void closeDB() throws SQLException{
            conn.close();
            System.out.println("Successfully closed table.");
      }
}//end DB class
```

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