

DEPRESSION DETECTION USING OCR AND NLP

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

SUNDEEP KUMAR (38110112)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL – 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of CH
SUNDEEP KUMAR (38110112) who carried out the project entitled
Depression Detection Using OCR and NLP under my supervision from
December 2019 to April 2020

Internal Guide

Dr.Cruz Antony M.E., Ph. D

Head of the Department

Dr. L.Lakshmanan M.E., Ph.D.,

Dr. S. Vigneshwari M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I SUNDEEP KUMAR hereby declare that the Project Report entitled Depression detection using OCR and NLP done by me under the guidance of **Dr.Cruz Antony, M.E., Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T. Sasikala M.E., Ph.D**, Dean, School of Computing , **Dr. S. Vigneshwari M.E., Ph.D. and Dr. L. Lakshmanan M.E., Ph.D. , Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express are sincere and deep sense of gratitude to my Project Guide **Dr.Cruz Antony, M.E., Ph.D** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Design is a multi- step that focuses on data structure software architecture, procedural details, procedure etc. and interface among modules. The design procedure also decodes the requirements into presentation of software that can be accessed for excellence before coding begins. Computer software design change continuously as novel methods; improved analysis and border understanding evolved. Software proposal is at relatively primary stage in its revolution Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are usually associated with more conventional engineering disciplines. However, methods for software designs do exist, criteria for design qualities are existing and design notation can be applied. System architecture is used to design and develop a web application, which provide an easy and convenient way to get information about depression levels of user by using machine learning algorithms and according to the location of user, the information about doctor is provided. The extraction class will perform the extraction of textual data from Facebook through Facebook graph API. Preprocessing class is used to preprocess the extracted data. Data must be clear, right and it is preprocessed for taking care of missing or repetitive attributes. The data ought to be complete and reliable data to deliver the best result from the data mining methodology. Preprocessing of data takes place by using techniques such as tokenization, lower case conversion, word stemming and words removal. Term frequency (tf) has been computed to measure term occurrence. In proposed system user is on Facebook, according to his Facebook post system can find out user is stressed or not as well as different quaternaries which is provided by the system. If users are not on Facebook, they can attempt only quaternaries which is provided by the system according to that we can find out user's is stressed or not.

LIST OF FIGURES

FIGURE NO.	NAME OF THE FIGURE	PAGE NO.
	ARCHITECTURE DIAGRAM	6
	BLOCK DIAGRAM OF OCR	8
	IMPLEMENTATION OF PROJECT	13
	USER REGISTRATION	14
	STORED USER LOGIN DATA IN DATABASE	15
	USER LOGIN	15
	APPLICATION DIALOG BOX	16
	IMAGE SELECTION	16
	DISPLAYING SENTIMENT RESULT	17

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	V
	LIST OF FIGURES	VI
1	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	1
2	LITERATURE SURVEY	3
3	METHODOLOGY	5
	3.1 SYSTEM REQUIREMENTS	5
	3.2 SYSTEM ARCHITECTURE	6
	3.3 SENTIEXTRACT	8
	3.4 TESSERACT OCR	8
	3.5 EXTRACTION OF TEXT FROM PICTURE FILES	8
	3.6 UML DIAGRAMS	9
4	RESULTS	14
	4.1 RESULTS	14
	4.2 DISCUSSIONS	14
5	CONCLUSION AND FUTURE WORK	18
	5.1 CONCLUSION	18
	5.2 FUTURE WORK	18
	REFERENCES	19
	APPENDIX	21
	A. SOURCE CODE	21
	B. PAPER WORK	

CHAPTER 1

INTRODUCTION

OUTLINE OF THE PROJECT

There is a huge quantity of data available in the Information Trade. This data is of no use until it is changed into valuable information. It is compulsory to analyze this huge quantity of data and extract valuable information from it.

Mining of information is not the only process we need to complete; data mining also involves other processes such as Data Cleaning, Data Integration, Data Transformation, Data Mining, Pattern Evaluation and Data Presentation. Once all these procedures are completed, we would be able to use this info in many applications such as Fraud Detection, Market Analysis, Construction Control, Science Exploration, etc.

WHAT IS DATA MINING?

Data Mining is defined as extracting info from vast sets of data. In other words, we can say that data mining is the way of mining knowledge from data. The information or knowledge extracted so can be used for any of the following applications –

- Market Examination
- Fake Detection
- Customer Retention
- Fabrication Control

DATA MINING APPLICATIONS

- Market Analysis and Management
- Corporate Analysis & Risk Management

- Fraud Detection

Apart from these, data mining can also be used in the zones of fabrication control, buyer retention, science exploration, sports, astrology, and Internet Web Surf-Aid.

DATA MINING – TASKS

Data mining deals with the kind of configurations that can be mined. On the basis of the kind of data to be mined, there are two groups of tasks involved in Data Mining –

- Descriptive
- Arrangement and Prediction

Descriptive Function

The descriptive function deals with the general properties of data in the database. Here is the list of descriptive functions –

- Class/Concept Description
- Mining of Repeated Patterns
- Mining of Links
- Mining of Correlations
- Mining of Clusters

Data Mining Task Primitives

- We can specify a data mining job in the form of a data mining query.
- This query is input to the system.

CHAPTER 2

LITERATURE SURVEY

A depression dataset to build automatic diagnoses in clinically depressed Saudi patient

Depression is a public health problem that has high effects on a person's functional and social relationships. Depression is a growing problem in the society. It causes pain and suffering not only to patients, but also to those who care about them. Depression disorder is hard to diagnose, because its symptoms could be confused with other disorders and has different cross-cultural symptoms. This paper proposes a framework that would best solve the problem of automatic depression detection in depressed Saudi patients. This paper particularly focuses on designing the collection of Saudi depression dataset using multiple modalities.

An improved model for depression detection in micro blog social network

Social networks contain a tremendous amount of node and linkage data, providing unprecedented opportunities for a wide variety of fields. As the world's fourth largest disease, depression has become one of the most significant research subjects. Previously, a depression classifier has been proposed to classify the users in online social networks to be depressed or not, however, the classifier takes only node features into account and neglects the influence of linkages. This paper proposes an improved model to calculate the probability of a user being depressed, which is based on both node and linkage features. The linkage features are measured in two aspects: tie strength and interaction content analysis. Moreover,

Toward the development of cost effective e-depression effective system

Diagnosis and prevention of depressive disorders at any scale have been attracting considerable attention of the public healthcare in Japan because depression is one of the most rapidly pervasive mental disorders in the country. A major issue that hinders the feasibility of depression screening for its prevention is the availability of some simple and cost-effective methods for depression detection

and monitoring. Here in this paper, we present the development of a computerized tool for depression detection. The tool utilizes the theory of chaos and systems complexity to extract robust dynamically statistical features of physiological signals provided by the low-cost technology of photoplethysmography.

Detection of Clinical Depression in Adolescents' Speech During Family Interactions

The properties of acoustic speech have previously been investigated as possible cues for depression in adults. However, these studies were restricted to small populations of patients and the speech recordings were made during patients' clinical interviews or fixed-text reading sessions. Symptoms of depression often first appear during adolescence at a time when the voice is changing, in both males and females, suggesting that specific studies of these phenomena in adolescent populations are warranted. This study investigated acoustic correlates of depression in a large sample of 139 adolescents (68 clinically depressed and 71 controls).

CHAPTER 3

METHODOLOGY

SYSTEM REQUIREMENTS

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description as functional representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

HARDWARE REQUIREMENTS

System : Pentium IV 2.4 GHz

Hard Disk : 40 GB

Floppy Drive : 1.44 Mb

Monitor : 15 VGA Colour

Mouse : Logitech

Ram : 512 Mb

SOFTWARE REQUIREMENTS

Operating system : Windows 10

IDE : Eclipse

Coding Language : Java

Backend : SQLyog

SYSTEM ARCHITECTURE

INTRODUCTION

Design is a multi- step that focuses on data structure software architecture, procedural details, procedure etc... and interface among modules. The design procedure also decode the requirements into presentation of software that can be accessed for excellence before coding begins. Computer software design change continuously as novel methods; improved analysis and border understanding evolved. Software proposal is at relatively primary stage in its revolution Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are usually associated with more conventional engineering disciplines. However methods for software designs do exit, criteria for design qualities are existing and design notation can be applied.

ARCHITECTURE DIAGRAM:

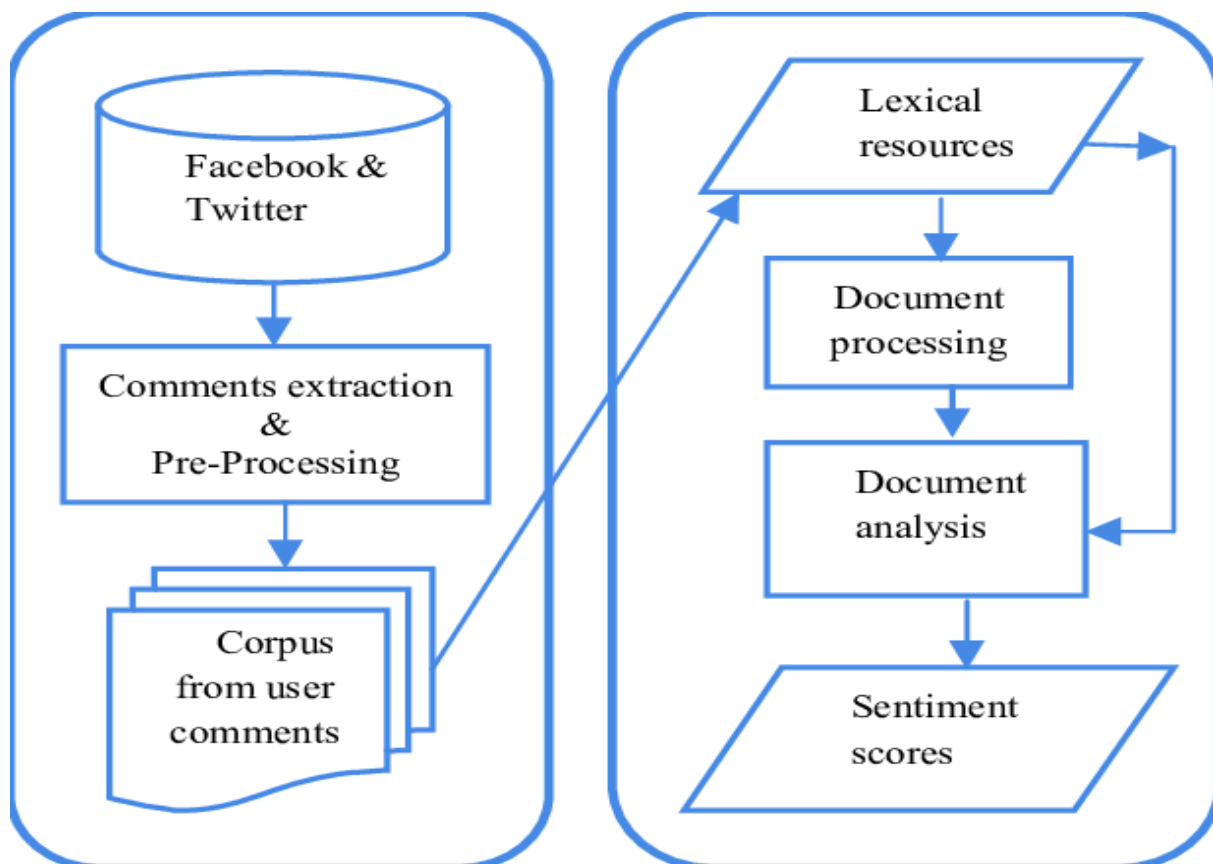


Fig 3.1 ARCHITECTURE DIAGRAM

System architecture is used to design and develop an web application, which provide an easy and convenient way to get information about depression levelsof user by using machine learning algorithms and according to the location of user, the information about doctor is provided. The extraction class will performed the extraction of textual data from facebook through facebook graph API. Preprocessing class is used to preprocess the extracted data. Data must be clear, right and it is preprocessed for taking care of missing or repetitive attributes. The data ought to be complete and reliable data to deliver the best result from the data mining methodology. Preprocessing of data takes place by using techniques such as tokenization, lower case conversion, word stemming and words removal. Term frequency (tf) has been computed to measure term occurrence. In proposed system user is on Facebook, according to his Facebook post system can find out user in stressed or not as well as different quaternaries which is provided by the system. If user's are not on Facebook they can attempt only quaternaries which is provided by the system according to that we can find out user's in stressed or not.

Sentiment analysis:

Sentiment analysis is one of the most common applications in natural language processing. With Sentiment analysis, we can decide what emotion a text is written.

Here we are using the standford- corenlp for finding the users sentimental. The following **outline** is provided as an overview of and topical guide to natural language processing:

Natural language processing

Computer activity in which computers are entailed to analyze, understand, alter, or generate natural language. This includes the automation of any or all linguistic forms, activities, or methods of communication, such as conversation, correspondence, reading, written composition, dictation, publishing, translation, lip reading, and so on. Natural language processing is also the name of the branch of computer science, artificial intelligence, and linguistics concerned with enabling computers to engage in communication using natural language(s) in all forms, including but not limited to

speech, print, writing, and signing.

SENTIEXTRACT

Our system boasts of two highly efficient algorithms namely, tesseract OCR algorithm for image to text conversion and naïve bayes classifier for sentiment extraction of the converted text. A single or multiple image file can be given as input to our system for calculation of their sentiment polarity.

TESSERACT OCR

Tesseract OCR is an optical character reading engine developed by HP laboratories in 1985 and open sourced in 2005. Since 2006 it is developed by Google. Tesseract has Unicode (UTF-8) support and can recognize more than 100 languages “out of the box” and thus can be used for building different language scanning software also. Latest Tesseract version is Tesseract 4. It adds a new neural net (LSTM) based OCR engine which is focused on line recognition but also still supports the legacy Tesseract OCR engine which works by recognizing character patterns.

EXTRACTION OF TEXT FROM PICTURE FILES

The image file is firstly converted to a text file using the tesseract ocr algorithm, in our system we are using the pyocr wrapper for this purpose. This wrapper invokes the tesseract algorithm and gives us the required text file, which is then used for further analysis. Image files of any size can be given as input to the system, but care should be taken as blurred or skewed images may not fetch you optimal results.

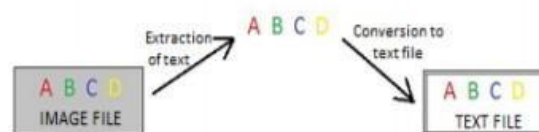


Fig 3.2 Block Diagram Of OCR

Number of words a file may contain has no constraints, but the words should be of English language as it is the only language currently supported by our system

UML DIAGRAMS

UML is a standard language for specifying, visualizing, and documenting of software systems and formed by Object Management Group (OMG) in 1997. There are three imperative type of UML modeling are Structural model, Behavioral model, and Architecture model. To model a system the most essential aspect is to capture the dynamic behavior which has some internal or external factors for creating the collaboration. These internal or external agents are known as actors. It consists of actors, use cases and their associations. In this fig we represent the Use Case diagram for our development.

USECASE DIAGRAM

Use case diagrams overview the usage requirements for a system. They describe “the meat” of the actual necessities.

CLASS DIAGRAM

A class is a collection of things that have alike attributes and common behaviors. Class diagrams are used to label the structure of the system. Objects are instances of classes that are shaped, altered, and destroyed during the execution of the system. An object has state that comprises the values of its attributes and it links with other objects

SEQUENCE DIAGRAM

A sequence diagram depicts the categorization of actions that occur in a system. The supplication of methods in each object, and the order in which the invocation occurs is captured in a sequence diagram. This makes the sequence diagram a very beneficial tool to easily represent the dynamic behavior of a system.

COLLABORATION DIAGRAM

A collaboration diagrams goes to a group of UML diagrams called interaction diagrams. Collaboration diagrams, like categorization diagrams, show how substances interact over the course of time. However, instead of viewing the sequence by totaling the messages of the diagram. This creates it easier to show how the substances are

linked together, but harder to see the categorization at a glance.

DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical demonstration of the "flow" of data through an information system, modelling its process characteristics. A DFD is often used as a initial step to create an overview of the system, which can later be explained.

SOFTWARE ENVIRONMENT JAVA TECHNOLOGY

Java technology is both a programming language and a platform. Java is a general-purpose computer programming language that is concurrent, class- based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is envisioned to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2015, one of the most widespread programming languages in use, predominantly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language originates much of its syntax from C and C++, but it has fewer low-level facilities than either of them. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licences. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed substitute enactments of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plugin for applets).

JAVA PLATFORM

One strategy goal of Java is portability, which means that programs written for the Java platform must run likewise on any amalgamation of hardware and operating system with adequate runtime provision. This is attained by compiling the Java language code to an in-between illustration called Java byte code, instead of directly to architecture-specific machine code. Java byte code instructions are analogous to machine code, but they are envisioned to be achieved by a virtual machine (VM) written precisely for the host hardware. End users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a web browser for Java applets. Standardized libraries offer a generic way to access host-specific structures such as graphics, threading, and networking.

A foremost benefit of using byte code is porting. However, the upstairs of understanding means that interpreted programs nearly always run more slowly than programs compiled to native executable would. Just-in-Time (JIT) compilers.

were presented from an primary stage that compiles byte codes to machine code during runtime. Java is platform independent. But as Java virtual machine must convert Java byte code into machine language which depends on the operating system being used, it is platform dependent.

JAVA VIRTUAL MACHINE

The heart of the Java platform is the idea of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what hardware or operating system the program is running under. There is a JIT (Just in Time) compiler within the Java Virtual Machine, or JVM. The JIT compiler interprets the Java byte code into native processor instructions at run-time and caches the native code in memory during execution. The use of byte code as an middle language permits Java programs to run on any platform that has a virtual machine obtainable.

The use of a JIT compiler means that Java applications, after a short postponement during piling and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has comprised a just-in-time compiler instead of an interpreter. Although Java programs are cross-platform or platform independent, the code of the Java Virtual Machines (JVM) that implements these programs is not. Every supported

operating platform has its own JVM.

JAVA CLASS LIBRARY

The Java Class Library (JCL) is a set of dynamically loadable libraries that Java applications can call at run time. Because the Java Platform is not dependent on a precise operating system, submissions cannot rely on any of the platform-native libraries. Instead, the Java Platform provides a comprehensive set of standard class libraries, comprising the functions common to modern operating systems.

JCL serves three purposes within the Java Platform:

- Like other standard code libraries, they offer the computer programmer a well-known set of useful amenities, such as container classes and regular expression processing.
- The library delivers an abstract interface to tasks that would customarily depend heavily on the hardware and operating system, such as network access and file access.
- Some underlying platforms may not support all of the features a Java submission expects. In these cases, the library execution can either emulate those topographies or provide a consistent way to check for the existence of a precise feature.

IMPLEMENTATIONS

Oracle Corporation is the current owner of the official execution of the Java SE platform, following their acquisition of Sun Microsystems on January 27, 2010. This execution is based on the new implementation of Java by Sun. The Oracle implementation is available for Microsoft Windows, Mac OS X, Linux and Solaris. Because Java lacks any formal normalization renowned by Ecma International, ISO/IEC, ANSI, or other third-party standards organization, the Oracle enactment is the de facto standard.

The Oracle implementation is packaged into two dissimilar distributions: The Java Runtime Environment (JRE) which comprises the parts of the Java SE platform required to run Java programs and is proposed for end users, and the Java Development Kit (JDK), which is intended for software developers and includes development tools such as the Java compiler, Java doc, Jar, and a debugger. Open JDK is another notable Java SE implementation that is licensed under the GNU GPL.

The implementation started when Sun began releasing the Java source code under the GPL. As of Java SE 7, Open JDK is the official Java reference implementation. The goal of Java is to make all implementations of Java compatible. Historically, Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with Microsoft after Sun claimed that the Microsoft implementation did not support RMI or JNI and had added platform-specific features of their own.

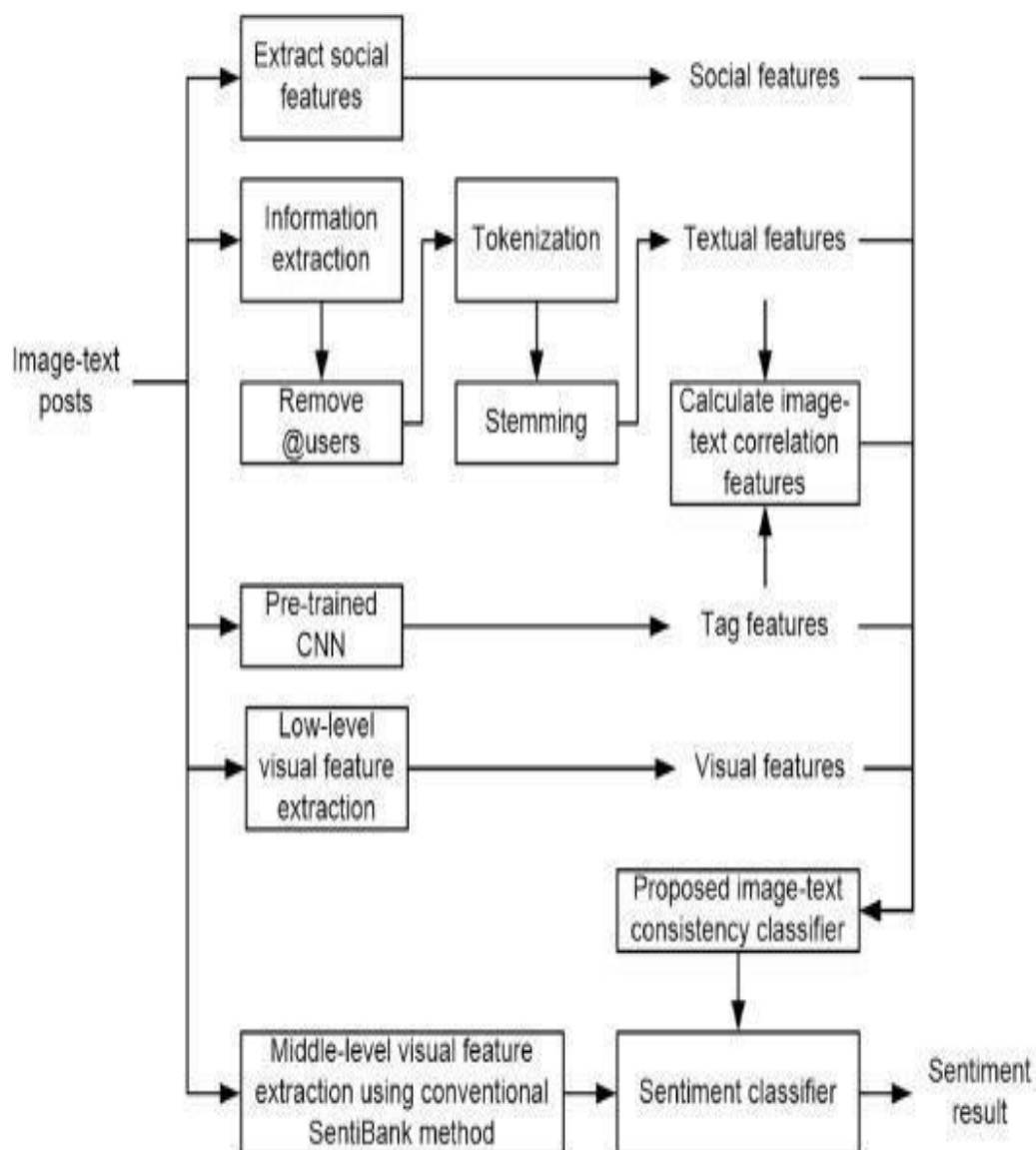


Fig 3.3 IMPLEMENTATION OF PROJECT

Chapter 4

RESULTS AND DISCUSSIONS

The purpose of testing is to discover faults. Testing is the process of trying to determine every possible fault or weakness in a work invention. It provides a way to check the functionality of workings, sub-assemblies, meetings and/or a ended product it is the process of training software with the intent of ensuring that the Software system meets its requirements and user opportunities and does not fail in an deplorable manner. There are various types of test. Each test type addresses a specific testing condition.

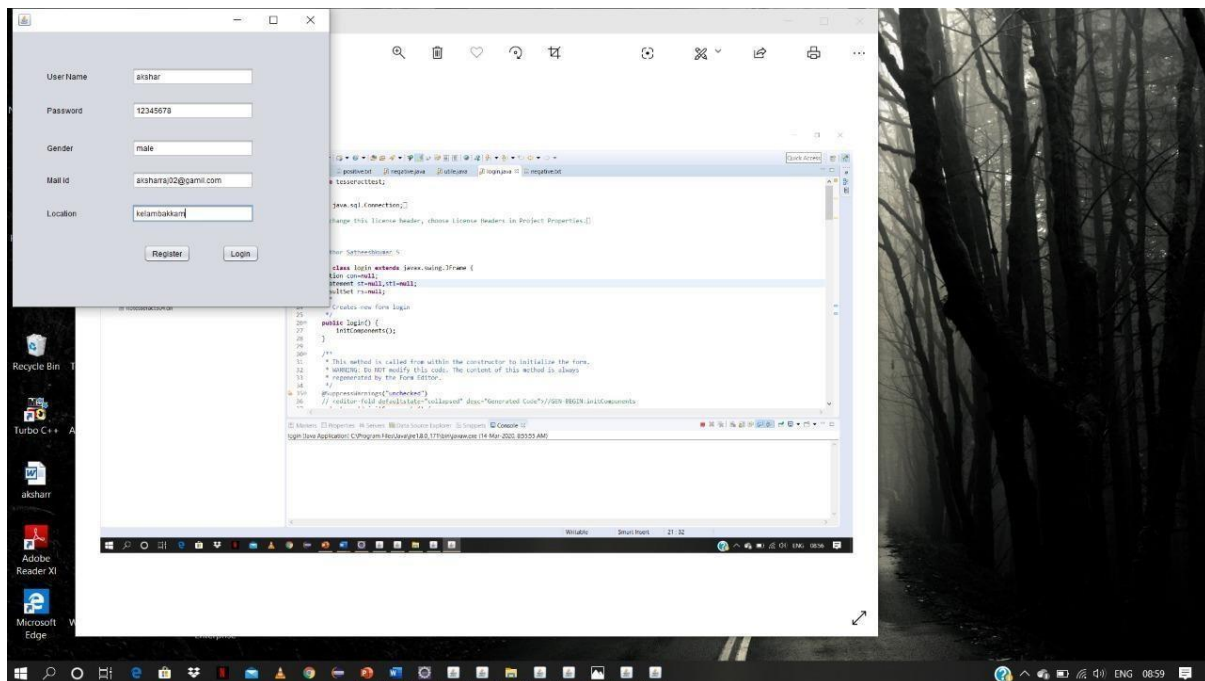


Fig 4.1 USER REGISTRATION

The user needs to register first to login with the same credentials for uploading the images.

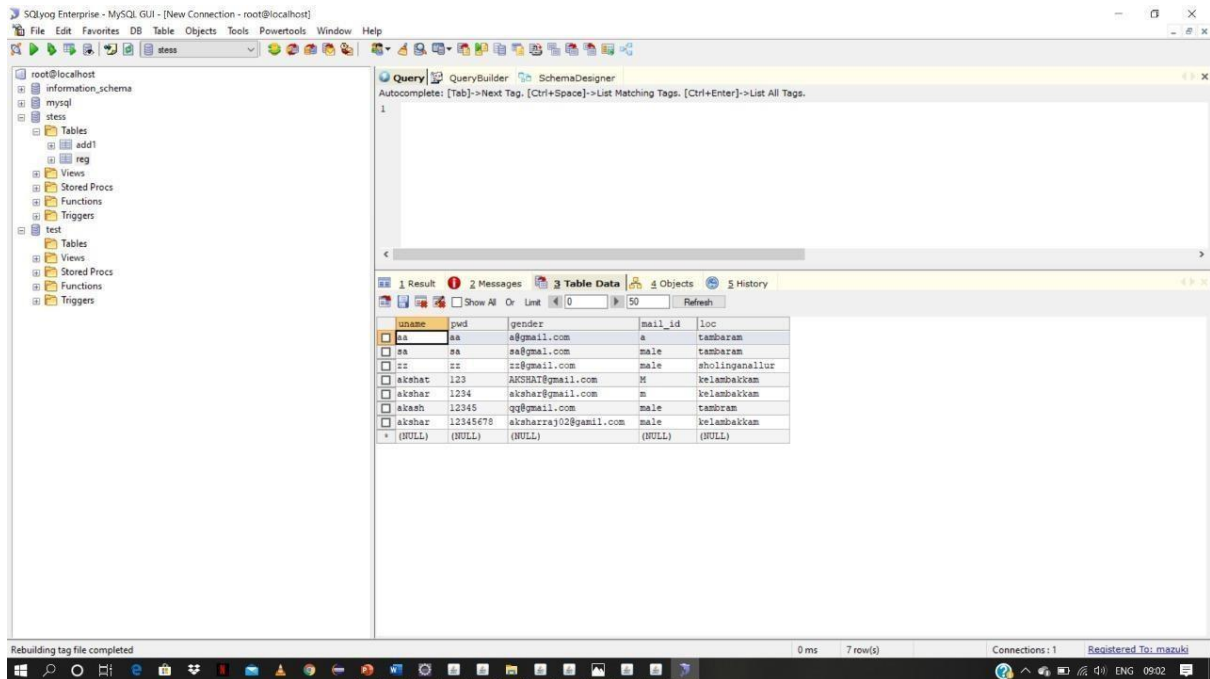


Fig 4.2 STORED USER LOGIN DATA IN DATABASE

The user login details are stored in a database. So that at the time of login only the legitimate users can access the application.

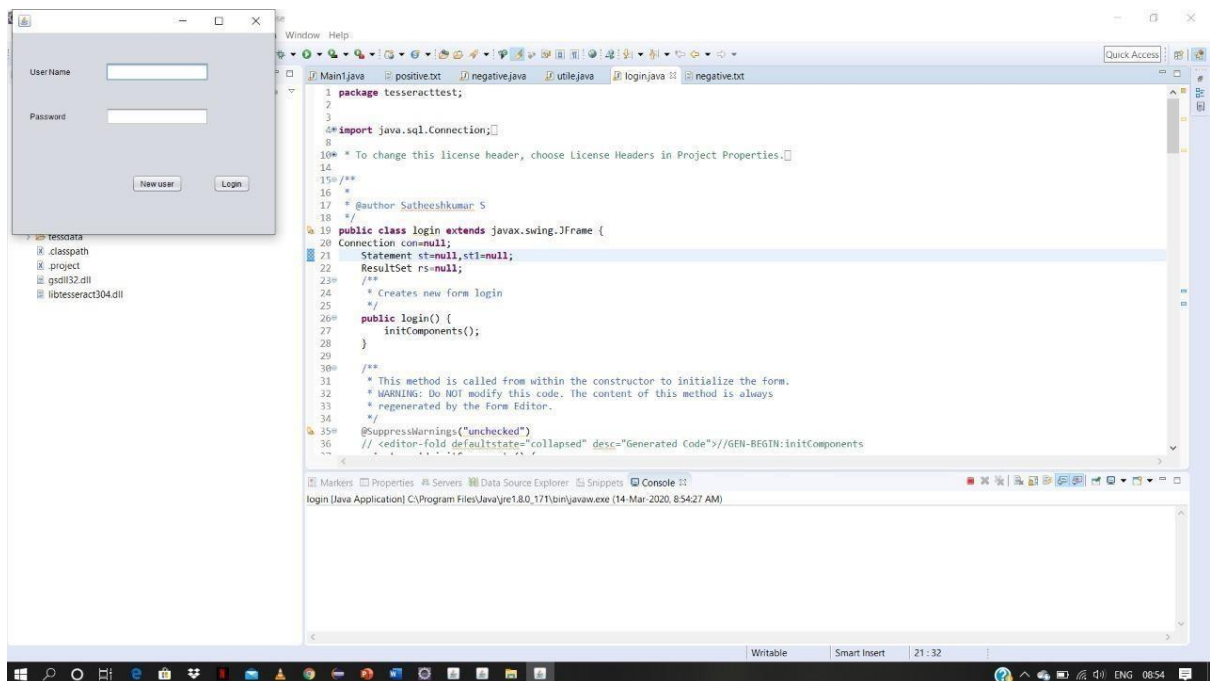


Fig 4.3 USER LOGIN

Respective user will use their login credentials to get logged in. After getting logged in one dialog box will be opened where user can post the image from where the images needs to be extracted.

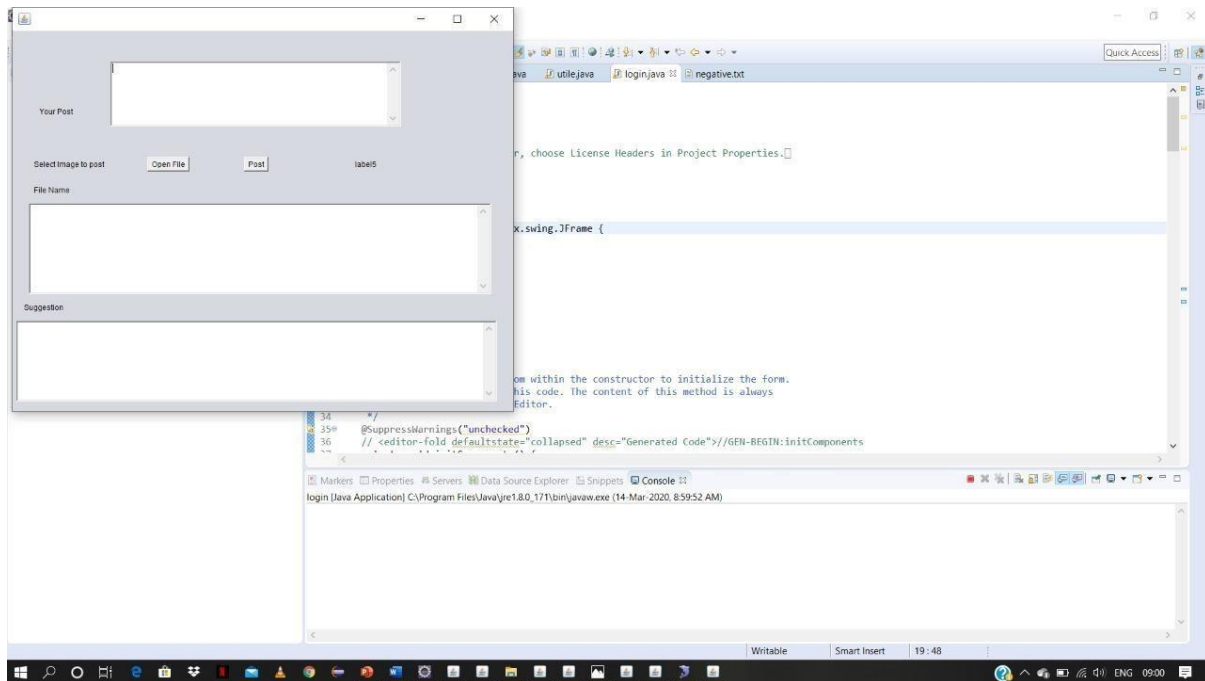


Fig 4.4 APPLICATION DIALOG BOX

Dialog box of the application from where the user can post the image for which the sentiment is needed to be analysed.

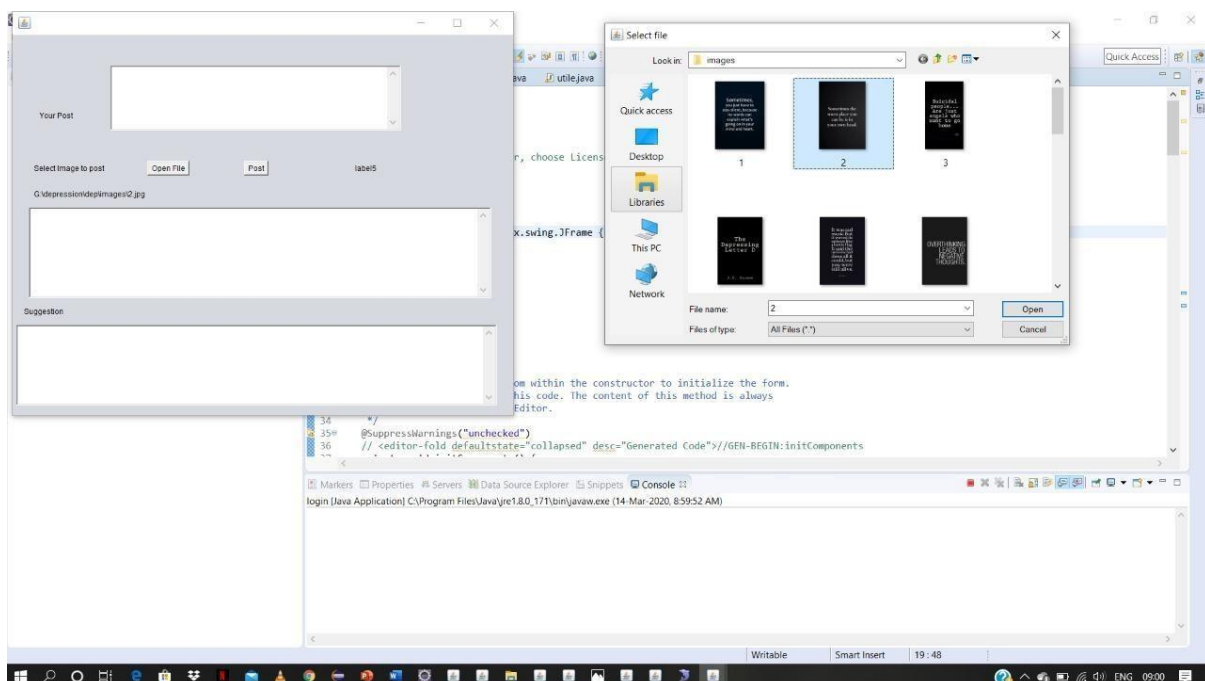


Fig 4.5 IMAGE SELECTION

From the opened dialog box user can post any image for the extraction of words from those images.

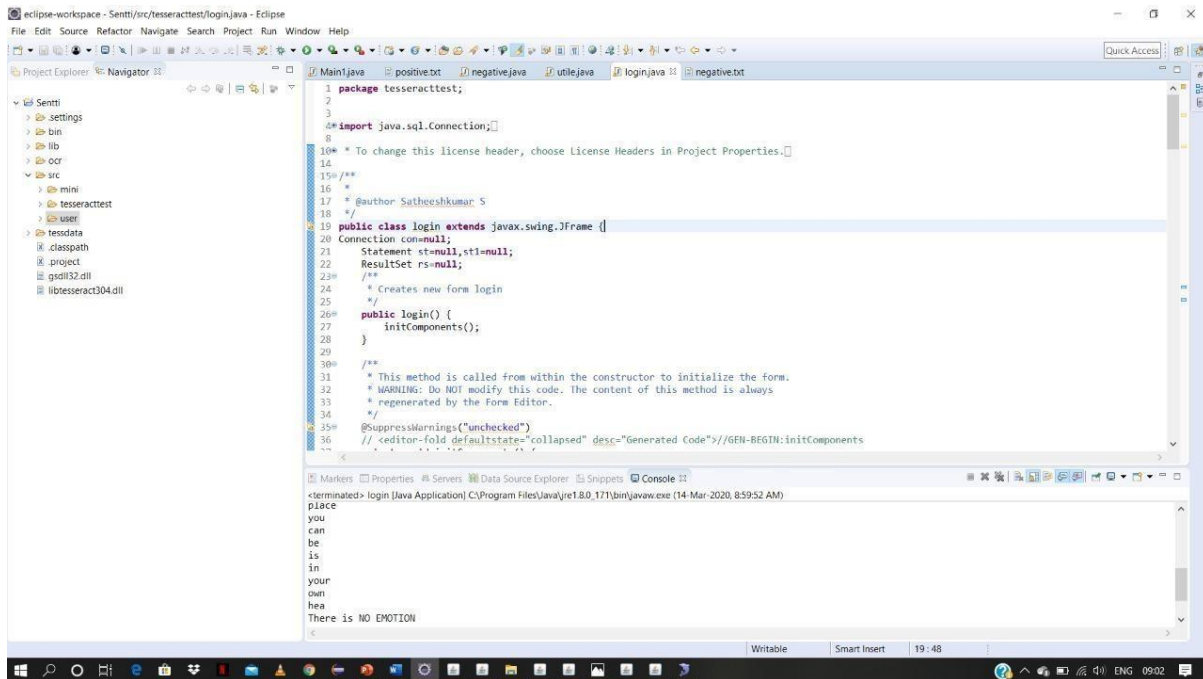


Fig 4.6 DISPLAYING SENTIMENT RESULT

The sentiment result will be displayed finally on the basis of the extracted words. - Whether the post is having negative, positive, mixed or with no emotion.

Chapter 5

CONCLUSION & FUTURE WORK

CONCLUSION

The experiments results show that the use of behavioral information on Facebook, both in forms of messages and activities, could predict depression. However, the sample of this research is relatively small because Facebook has limited their permission to collect personal information and the process of gaining approval has become more complicated. Thus, the results getting from this study might not cover all relevant factors. Moreover, as the language-related features had to be translated from Thai to English for analyzing the process, there might be some errors due to this process because some important sentiment polar words might have been eliminated during the translation process.

FUTURE ENHANCEMENT

For the future research, we intend to collect more data to get more relevant and valid features. Manual annotating all complex attributes using crowdsourcing and deeper dimensions should also be analyzed in order to be able to create a better depression detection algorithm.

REFERENCES

- [1] Wikipedia, "Twitter, " <http://en.wikipedia.org/wiki/Twitter>.2014.
- [2] Xinhuanet, "Sina Microblog Achieves over 500 Million Users," http://news.xinhuanet.com/tech/2012-02/29/c_122769084.htm. 2014.
- [3] D. Perito, C. Castelluccia, M.A. Kaafar, and P. Manils, "How unique and traceable are usernames?," Privacy Enhancing Technologies (PETS'11), pp. 1-17, 2011.
- [4] J. Liu, F. Zhang, X. Song, Y.I. Song, C.Y. Lin, and H.W. Hon, "What's in a name?: an unsupervised approach to link users across communities," Proc. of the 6th ACM international conference on Web search and data mining(WDM'13), pp. 495- 504, 2013.
- [5] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," Proc. of the 3rd International ICWSM Conference, pp. 354-357, 2009.
- [6] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach, " Proc. of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13), pp.41-49, 2013
- [7] A. Acquisti, R. Gross and F. Stutzman, "Privacy in the age of augmented reality," Proc. National Academy of Sciences, 2011.
- [8] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying users across social tagging systems," Proc. of the 5th International AAAI Conference on Weblogs and Social Media, pp. 522-525, 2011.
- [9] M. Motoyama and G. Varghese, "I seek you: searching and matching individuals in social networks," Proc. of the 11th international workshop on Web Information and Data Management (WIDM'09), pp. 67-75, 2009.

- [10] O. Goga, D. Perito, H. Lei, R. Teixeira, and R. Sommer, "Largescale Correlation of Accounts across Social Networks," Technical report, 2013.
- [11] K. Cortis, S. Scerri, I. Rivera, and S. Handschuh, "An ontology based technique for online profile resolution," Social Informatics, Berlin: Springer, pp. 284-298, 2013.
- [12] F. Abel, E. Herder, G.J. Houben, N. Henze, and D. Krause, "Cross-system user modeling and personalization on the social web," U.

APPENDIX

A. SOURCE CODE:

#register.java:

```
<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">

  <Properties>

    <Property name="defaultCloseOperation" type="int" value="3"/>

  </Properties>

  <SyntheticProperties>

    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>

    <SyntheticProperty name="generateCenter" type="boolean" value="false"/>

  </SyntheticProperties>

  <AuxValues>

    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>

    <AuxValue name="FormSettings_autoSetComponentName"
type="java.lang.Boolean" value="false"/>

    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>

    <AuxValue name="FormSettings_generateMnemonicsCode"
type="java.lang.Boolean" value="false"/>

    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
```

```
<AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
```

```
<AuxValue name="FormSettings_listenerGenerationStyle"
type="java.lang.Integer" value="0"/>
```

```
<AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
```

```
<AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
```

```
</AuxValues>
```

```
<Layout>
```

```
<DimensionLayout dim="0">
```

```
<Group type="103" groupAlignment="0" attributes="0">
```

```
<Group type="102" attributes="0">
```

```
<Group type="103" groupAlignment="0" attributes="0">
```

```
<Group type="102" alignment="0" attributes="0">
```

```
<EmptySpace min="-2" pref="55" max="-2" attributes="0"/>
```

```
<Group type="103" groupAlignment="0" attributes="0">
```

```
<Component id="jLabel1" alignment="0" min="-2" max="-2"
attributes="0"/>
```

```
<Component id="jLabel2" alignment="0" min="-2" max="-2"
attributes="0"/>
```

```
<Component id="jLabel3" alignment="0" min="-2" max="-2"
attributes="0"/>
```

```
<Component id="jLabel4" alignment="0" min="-2" max="-2"
attributes="0"/>
```

```

        <Component id="jLabel5" alignment="0" min="-2" max="-2"
attributes="0"/>

    </Group>

    <EmptySpace min="-2" pref="72" max="-2" attributes="0"/>

    <Group type="103" groupAlignment="0" max="-2" attributes="0">

        <Component id="jTextField1" alignment="0" max="32767"
attributes="0"/>

        <Component id="jTextField2" alignment="0" max="32767"
attributes="0"/>

        <Component id="jTextField4" alignment="0" max="32767"
attributes="0"/>

        <Component id="jTextField3" alignment="0" max="32767"
attributes="0"/>

        <Component id="jTextField5" alignment="1" pref="193"
max="32767" attributes="0"/>

    </Group>

</Group>

<Group type="102" alignment="0" attributes="0">

    <EmptySpace min="-2" pref="209" max="-2" attributes="0"/>

    <Component id="jButton1" min="-2" max="-2" attributes="0"/>

    <EmptySpace min="-2" pref="50" max="-2" attributes="0"/>

    <Component id="jButton2" min="-2" max="-2" attributes="0"/>

</Group>

</Group>

<EmptySpace pref="112" max="32767" attributes="0"/>

```

```

    </Group>

</Group>

</DimensionLayout>

<DimensionLayout dim="1">

    <Group type="103" groupAlignment="0" attributes="0">

        <Group type="102" alignment="0" attributes="0">

            <EmptySpace min="-2" pref="57" max="-2" attributes="0"/>

            <Group type="103" groupAlignment="3" attributes="0">

                <Component id="jLabel1" alignment="3" min="-2" max="-2"
attributes="0"/>

                <Component id="jTextField1" alignment="3" min="-2" max="-2"
attributes="0"/>

            </Group>

            <EmptySpace min="-2" pref="26" max="-2" attributes="0"/>

            <Group type="103" groupAlignment="3" attributes="0">

                <Component id="jLabel2" alignment="3" min="-2" max="-2"
attributes="0"/>

                <Component id="jTextField2" alignment="3" min="-2" max="-2"
attributes="0"/>

            </Group>

            <EmptySpace min="-2" pref="32" max="-2" attributes="0"/>

            <Group type="103" groupAlignment="3" attributes="0">

                <Component id="jLabel3" alignment="3" min="-2" max="-2"
attributes="0"/>

```

```
<Component id="jTextField4" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
</Group>
```

```
<EmptySpace min="-2" pref="23" max="-2" attributes="0"/>
```

```
<Group type="103" groupAlignment="3" attributes="0">
```

```
<Component id="jLabel4" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
<Component id="jTextField3" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
</Group>
```

```
<EmptySpace min="-2" pref="25" max="-2" attributes="0"/>
```

```
<Group type="103" groupAlignment="3" attributes="0">
```

```
<Component id="jLabel5" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
<Component id="jTextField5" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
</Group>
```

```
<EmptySpace pref="36" max="32767" attributes="0"/>
```

```
<Group type="103" groupAlignment="3" attributes="0">
```

```
<Component id="jButton1" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
<Component id="jButton2" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
</Group>
```

```
<EmptySpace min="-2" pref="71" max="-2" attributes="0"/>
```

```

        </Group>

    </Group>

</DimensionLayout>

</Layout>

<SubComponents>

    <Component class="javax.swing.JLabel" name="jLabel1">

        <Properties>

            <Property name="text" type="java.lang.String" value="User Name"/>

        </Properties>

    </Component>

    <Component class="javax.swing.JLabel" name="jLabel2">

        <Properties>

            <Property name="text" type="java.lang.String" value="Password"/>

        </Properties>

    </Component>

    <Component class="javax.swing.JLabel" name="jLabel3">

        <Properties>

            <Property name="text" type="java.lang.String" value="Gender"/>

        </Properties>

    </Component>

    <Component class="javax.swing.JLabel" name="jLabel4">

        <Properties>

            <Property name="text" type="java.lang.String" value="Mail id"/>

```

```

</Properties>

</Component>

<Component class="javax.swing.JTextField" name="jTextField1">

    <Events>

        <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jTextField1ActionPerformed"/>

    </Events>

</Component>

<Component class="javax.swing.JTextField" name="jTextField2">

</Component>

<Component class="javax.swing.JTextField" name="jTextField3">

</Component>

<Component class="javax.swing.JTextField" name="jTextField4">

</Component>

<Component class="javax.swing.JButton" name="jButton1">

    <Properties>

        <Property name="text" type="java.lang.String" value="Register"/>

    </Properties>

    <Events>

        <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton1ActionPerformed"/>

    </Events>

```

```

</Component>

<Component class="javax.swing.JButton" name="jButton2">

  <Properties>

    <Property name="text" type="java.lang.String" value="Login"/>

  </Properties>

  <Events>

    <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton2ActionPerformed"/>

  </Events>

</Component>

<Component class="javax.swing.JLabel" name="jLabel5">

  <Properties>

    <Property name="text" type="java.lang.String" value="Location"/>

    <Property name="toolTipText" type="java.lang.String" value=""/>

  </Properties>

</Component>

<Component class="javax.swing.JTextField" name="jTextField5">

</Component>

</SubComponents></Form>

```

#login.java:

```
package tesseracttest;
```

```

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Satheeshkumar S
 */

public class login extends javax.swing.JFrame {

    Connection con=null;

    Statement st=null,st1=null;

    ResultSet rs=null;

    /**
     * Creates new form login
     */

    public login() {

        initComponents();

    }

```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents

private void initComponents() {

    jButton1 = new javax.swing.JButton();

    jButton2 = new javax.swing.JButton();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jTextField1 = new javax.swing.JTextField();

    jTextField2 = new javax.swing.JTextField();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jButton1.setText("New user");

    jButton1.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            jButton1ActionPerformed(evt);

        }

    });

    jButton2.setText("Login");

```

```

jButton2.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton2ActionPerformed(evt);

    }

});

jLabel1.setText("User Name");

jLabel2.setText("Password");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

            .addGap(28, 28, 28)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(jLabel2)

                    .addComponent(jLabel1))

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 98,
Short.MAX_VALUE)

        .addComponent(jButton1)

        .addGap(49, 49, 49)

        .addComponent(jButton2)

        .addGap(39, 39, 39))

        .addGroup(layout.createSequentialGroup())

        .addGap(57, 57, 57)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addComponent(jTextField1)

        .addComponent(jTextField2,
javax.swing.GroupLayout.DEFAULT_SIZE, 163, Short.MAX_VALUE))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

            .addGap(48, 48, 48)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
E)

```

```

        .addComponent(jLabel1)

        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(43, 43, 43)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel2)

        .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 79,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jButton1)

        .addComponent(jButton2))

        .addGap(67, 67, 67))

);

pack();

} // </editor-fold> // GEN-END: initComponents

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton1ActionPerformed

    // TODO add your handling code here:

    Register r=new Register();

    r.setVisible(true);

    this.setVisible(false);

}//GEN-LAST:event_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed

    // TODO add your handling code here:

    try
    {

        Class.forName("com.mysql.jdbc.Driver");

        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stess","root","password");

        st1 = con.createStatement();

        rs = st1.executeQuery("select * from reg where
uname='"+jTextField1.getText().toString()+"' &&
pwd='"+jTextField2.getText().toString()+"'");

        if(rs.next()) {

            utile u=new utile();

            u.setloc(rs.getString(5));

            u.setuname(jTextField1.getText().toString());

            jTextField1.setText("");

```

```

        jTextField2.setText("");

        Main1 l=new Main1();

        l.setVisible(true);

        this.setVisible(false);

//sn.setAttribute("dpm",department);

    }

}

catch(Exception ex)

{

}

}

}

//GEN-LAST:event_jButton2ActionPerformed

/**

 * @param args the command line arguments

 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.

    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

    */

```

```

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);

        }

    }

}
//</editor-fold>

```

```

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new login().setVisible(true);

    }

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

// End of variables declaration//GEN-END:variables

}

```

#main.java :

```

package tesseracttest;

import java.awt.Color;

import java.awt.image.BufferedImage;

```

```

import java.io.*;

import java.util.StringTokenizer;

import javax.imageio.ImageIO;

import mini.readfile;

import net.sourceforge.tess4j.*;

public class Main {

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

int psentiment=0 ,nsentiment=0,f;

    String s = new String();

    String key= new String();

String imgFilePath = "C:\\Users\\admin\\Pictures\\Saved Pictures";

        File file= new File(imgFilePath);

        BufferedImage image = ImageIO.read(file);

        // Getting pixel color by position x=100 and y=40

        int clr= image.getRGB(100,40);

        int red = (clr & 0x00ff0000) >> 16;

        int green = (clr & 0x0000ff00) >> 8;

        int blue = clr & 0x000000ff;

        ColorUtils C=new ColorUtils();

        String hex = C.getColorNameFromRgb(red, green,

blue);

        System.out.println("image Color name");

```

```

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

System.out.println("Color name " + hex.toString()

File imgfile = new File(imgFilePath);

Tesseract instance = Tesseract.getInstance(); //

//System.out.println(imgfile.canRead());

try {

    String result = instance.doOCR(imgfile);

    String rs=result.substring(1);

    rs=rs.substring(0, rs.length() - 4);

    rs=rs.trim();

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

    System.out.println("String Extraction from Image");

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

    System.out.println(rs);

    //System.out.println();

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

    System.out.println("sentiment Extraction from
Image Text");

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

    s=rs;

readfile rf = new readfile();

```

```

//tokenizing text

StringTokenizer st = new StringTokenizer(s," =,!?");

while(st.hasMoreTokens())

{

    f=0;

    key=st.nextToken();

    key=key.toLowerCase();

    System.out.println(key);

    rf.openFileP();

    while(rf.z.hasNext())

        {

            if(key.equals(rf.readFileP()))

                {

                    psentiment++;

                    f=1;

                    break;

                }

        }

}

rf.closeFileP();

if(f!=1)

{

    rf.openFileN();

    while(rf.w.hasNext())

```

```

        {
            if(key.equals(rf.readFileN()))
            {
                nsentiment++;
                break;
            }
        }
        rf.closeFileN();
    }
}

if (psentiment>nsentiment)

    System.out.println("The overall sentiment of the text is
POSITIVE");

else if (psentiment==0 && nsentiment==0)

    System.out.println("There is NO EMOTION");

else if (psentiment==nsentiment)

    System.out.println("The overall sentiment of text is
MIXED, as same positive and negative notions");

else

    System.out.println("The overall sentiment of text is
NEGATIVE");

System.out.println(".....END ..... ");

} catch (TesseractException e) {

```

```

        System.err.println(e.getMessage());
    }

}

public static Color[][] loadPixelsFromImage(File file) throws IOException {
    BufferedImage image = ImageIO.read(file);

    Color[][] colors = new Color[image.getWidth()][image.getHeight()];

    for (int x = 0; x < image.getWidth(); x++) {
        for (int y = 0; y < image.getHeight(); y++) {
            colors[x][y] = new Color(image.getRGB(x, y));
        }
    }

    return colors;
}
}

```