SOFTWARE DESIGN DOCUMENT

Table of Contents

[**1.0 INTRODUCTION** 1](#_Toc10025942)

[1.1 Purpose 1](#_Toc10025943)

[1.2 Scope 1](#_Toc10025944)

[1.3 Overview 1](#_Toc10025945)

[1.4 Reference Material 2](#_Toc10025946)

[1.5 Definitions and Acronyms 2](#_Toc10025947)

[**2.0 SYSTEM OVERVIEW** 3](#_Toc10025948)

[**3.0 SYSTEM ARCHITECTURE** 4](#_Toc10025949)

[3.1 Architectural Design 4](#_Toc10025950)

[3.2 Decomposition Description 6](#_Toc10025951)

[3.3 Design Rationale 7](#_Toc10025952)

[**4.0 DATA DESIGN** 7](#_Toc10025953)

[4.1 Data Description 7](#_Toc10025954)

[4.2 Data Dictionary 10](#_Toc10025955)

[**5.0 COMPONENT DESIGN** 11](#_Toc10025956)

[**6.0 HUMAN INTERFACE DESIGN** 12](#_Toc10025957)

[6.1 Overview of User Interface 12](#_Toc10025958)

[6.2 Screen Images 12](#_Toc10025959)

[Home Screen 12](#_Toc10025960)

[6.3 Screen Objects and Actions 14](#_Toc10025961)

[**7.0 REQUIREMENTS MATRIX** 15](#_Toc10025962)

**LIST OF TABLES**

[Table 1 Table showing entities and their attributes 10](#_Toc10025931)

[Table 2 : screen objects and their corresponding actions 14](#_Toc10025932)

[Table 3: Requirements Matrix 15](#_Toc10025933)

**LIST OF FIGURES**

[Figure 2.1 Generalized overview of how the user interacts Social Media Posts Sentiment Classifier 3](#_Toc10025905)

[Figure 3.1Architectural design 4](#_Toc10025906)

[Figure 3.2 Context data flow diagram for SPSC 5](#_Toc10025907)

[Figure 3.3 Level 1 data flow diagram for SPSC 6](#_Toc10025908)

[Figure 3.4 Level 2 data flow diagram for SPSC 7](#_Toc10025909)

[Figure 4.1 Entity Relationship Diagram showing entities in the database 9](file:///F:\semester2%20year%204\SEMESTER%20ONE\BSE%204100%20Software%20Engineering%20Project%201\group19-5\docs\BSE19-5%20SDD.docx#_Toc10025910)

[Figure 5.1 Component diagram for SPSC 11](#_Toc10025912)

[Figure 6.1 Home screen 12](#_Toc10025913)

[Figure 6.2 summarized Results 13](#_Toc10025914)

[Figure 6.3 shows upload file page. 14](#_Toc10025915)

# **1.0 INTRODUCTION**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project and other supporting requirement information.

## 1.1 Purpose

The purpose of this document is to present a detailed description of the architecture and designs of the Social Media Posts Sentiment Classifier, created by group BSE19-5 Makerere University for final year project. Firstly, this document is intended for the programming group in BSSE, to use the designs as guidelines to implement the project. Equally, this document is also for the project instructor, as it fulfils one of the requirements of the project. Lastly, this document could be used for the Social Media Posts Sentiment Classifier users as well as the designers who might want to embed it in their existing software.

## 1.2 Scope

Social Media Posts Sentiment Classifier is a Natural Language Processing classifier that will classify and determine the sentiment of comments depending on the article title/post. This project is limited at analyzing reactions to social media posts on Facebook, classifying them as positive, negative or neutral. The study is going to be carried out on reactions to posted articles by Uganda’s newspaper publishing companies which include Daily Monitor, New Vision, The Observer and Red pepper on Facebook. The project will cover comments in English only. Comments in any other language will not be covered in this project. This is because English is the most commonly used and spoken language in most parts of Uganda.

The dataset used will be downloaded from public Facebook pages of newspaper publishing companies mentioned above.

## 1.3 Overview

This document is written according to the standards for Software Design Documentation explained in “IEEE Recommended Practice for Software Design Documentation”.

The Software Design Document is divided into 7 sections with various subsections. The

sections of the Software Design Document are:

1. INTRODUCTION: This section includes a description of the project, its purpose, the overview, reference material and definitions and acronyms
2. SYSTEM OVERVIEW: This section includes the general description of functionality, context and design of the software.
3. SYSTEM ARCHITECTURE: This contains architectural design, decomposition description and design rationale.
4. DATA DESIGN: Includes data description and data dictionary
5. COMPONENT DESIGN: Contains the pseudocode of the functionalities.
6. HUMAN INTERFACE DESIGN: This will detail all aspects of the User Interface and its design, screen images and screen objects and actions.
7. REQUIREMENTS MATRIX: Includes requirements Traceability Matrix

## 1.4 Reference Material

The user of this SDD may need the following document for reference:

IEEE Standard 1016-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

[1]. ​ Nlp.stanford.edu. (2016). Stemming and lemmatization. [online] Available at: http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html [Accessed 19 Feb. 2019].

## 1.5 Definitions and Acronyms

BSSE – Bachelor of Science in Software Engineering

SDD – Software Design Document

SPSC – Social Media Posts Sentiment Classifier

# **2.0 SYSTEM OVERVIEW**

During our research study, we found out that every English newspaper publishing company in Uganda manages a Facebook page account. These Companies use these accounts to post different articles. People who have liked their pages can react or comment on these posted articles. These companies have handlers who try to analyze the comments to come up with a general feeling of the population who react about the posted article. It has always been a tiresome job for a single person (handler) to analyze a huge number of comments one by one.

In order to address the above challenge, we are proposing a software Classifier called **Social Media Posts Sentiment** **classifier.**

Generally, the software will display a user interface which will allow users to upload data (posts/article and comments) in a csv file format or to directly input the post and the corresponding comment(s) in the specific text areas displayed. The directly input data will be analyzed and the classifier shall display the results. The classifier will also analyze uploaded data and the output will show the sentiment or the classification of comments as positive, negative or neutral.

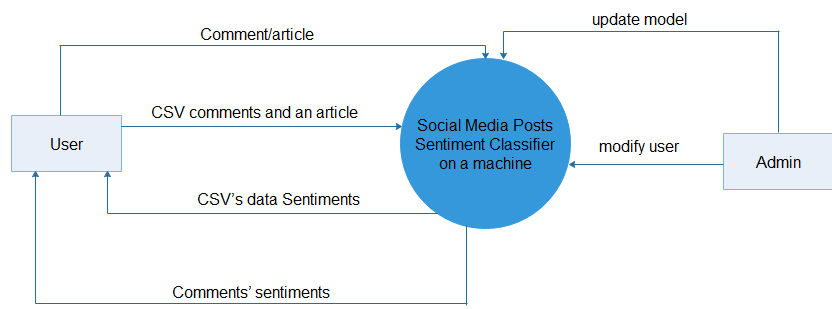


Figure 2.1 Generalized overview of how the user interacts Social Media Posts Sentiment Classifier

In the figure2.1, the user enters data to be classified into the classifier. The classifier classifies data. The data can be in form of a csv file uploaded or data directly entered into the classifier. Either way, the classifier displays the results back to the user. The classifier outputs comments’ sentiments. On the other side, the admin updates the model by uploading a csv file containing the post, comments and the sentiment of the comments. He can also modify the user by adding, deleting and promoting a user to an admin.

# **3.0 SYSTEM ARCHITECTURE**

## 3.1 Architectural Design

The proposed software is social media posts sentiment classifier. The architecture used for the system is a Client/Server Architecture.

The classifier is composed of two main components, Server Component and Interface Component. These components also are decomposed into smaller modules according to their structure and functionality.

The Server Component is responsible for holding all the user and system data and doing the classification logic. There will be one huge server for the system which will hold all the data about the Social Media Posits Sentiment Classifier. Also, with the help of modules it includes, server component will be responsible for doing all the classification logic including tokenization, stemming and lemmatization, classification, compiling and running project.

Second main component of the system is the Interface Component. Interface component mainly will be responsible for interaction of user with the Server Component. Its main functionality is the serving the Web front end of the classifier. Other functionalities of this component will be uploading data, allowing data input and displaying results output.

The figure 3.1 shows the architectural design of the classifier.

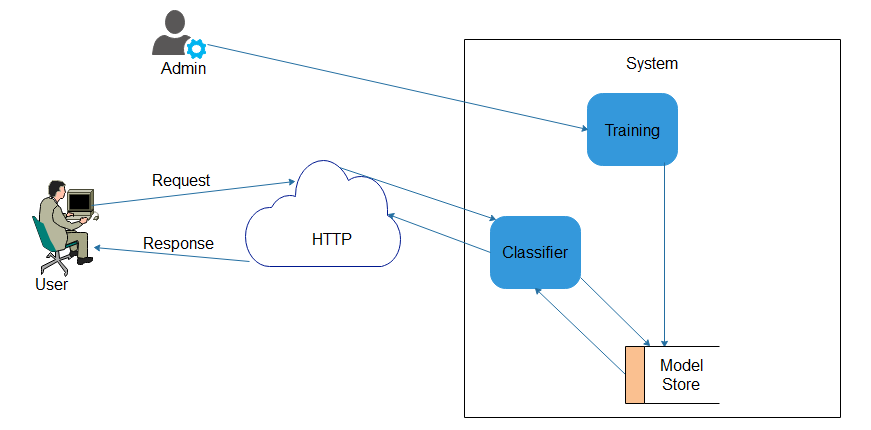


Figure 3.1Architectural design

**Structure**

The Admin trains the model and saves it to a data store called model store.

The user opens the classifier in any browser, sends an HTTP request by entering an article with its comments or uploading csv file containing an article plus its comments. The entered data/ uploaded data is processed and classified. The classifier uploads the stored model and its used to classify the new data. The classifier sends the comments’ sentiments as a response to the client through HTTP.

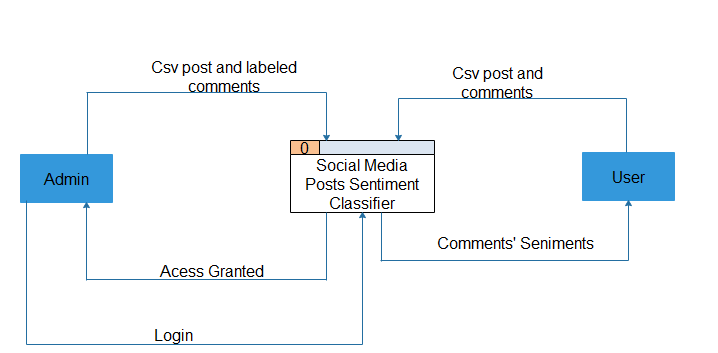


Figure 3.2 Context data flow diagram for SPSC

The figure 3.2 is a data flow diagram for SPSC. The Admin and the user are the only external entities. The Admin logs in and uploads the csv file containing posts and their labeled corresponding comments (training dat. The article and the post words are used interchangeably.

The user first registers with the system. The user then logs in to enter the new data which he/she needs to be classified. The new data includes article and its comment entered directly and an uploaded csv file which contains an article and its comments. The classifier outputs the comments’ sentiments as the result output.

## 3.2 Decomposition Description

**External Interface Module Description**

External interface will be entry point of Social Media Posts Sentiment Classifier for the users and Admin. Users will enter data, upload file/data and view graphs through the external interface and operations will be done on the server.The Admin will also use the external interface to logon to the system. External interface has a web front end.

Web front is a web page which will be the single-entry point for the users Admin. Users will not be required to login.

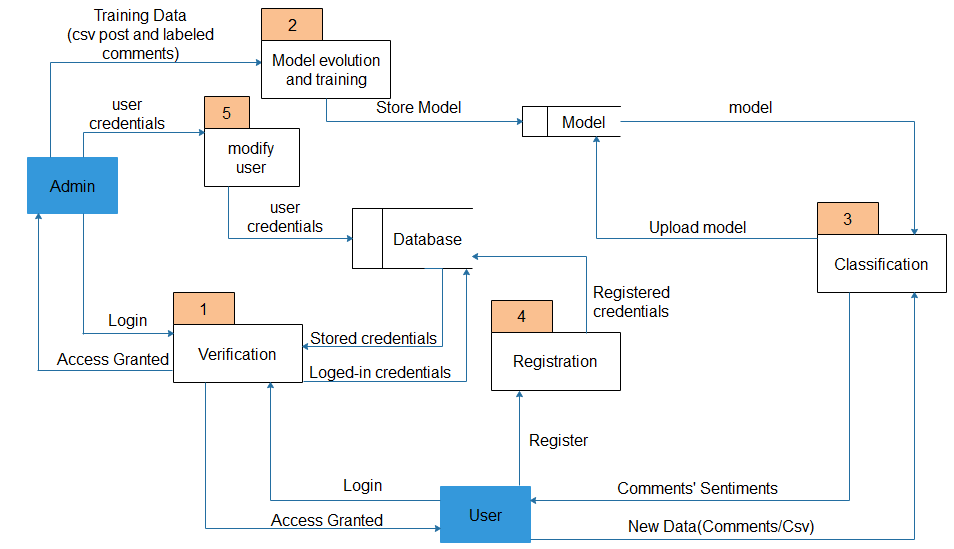


Figure 3.3 Level 1 data flow diagram for SPSC

In figure 3.3, the Admin logs in with username and password. The login credentials are verified. The Admin is granted access. The login credentials are stored in a database. The training data entered by the Admin will be preprocessed and trained (model evolution and training). The user will login to enter the new data to the classifier. The classifier will output the comments’ sentiments to the user. If the sentiment of the predicted comment is not right, the user updates it. The updated comments are stored into the database which later are used to update the model.

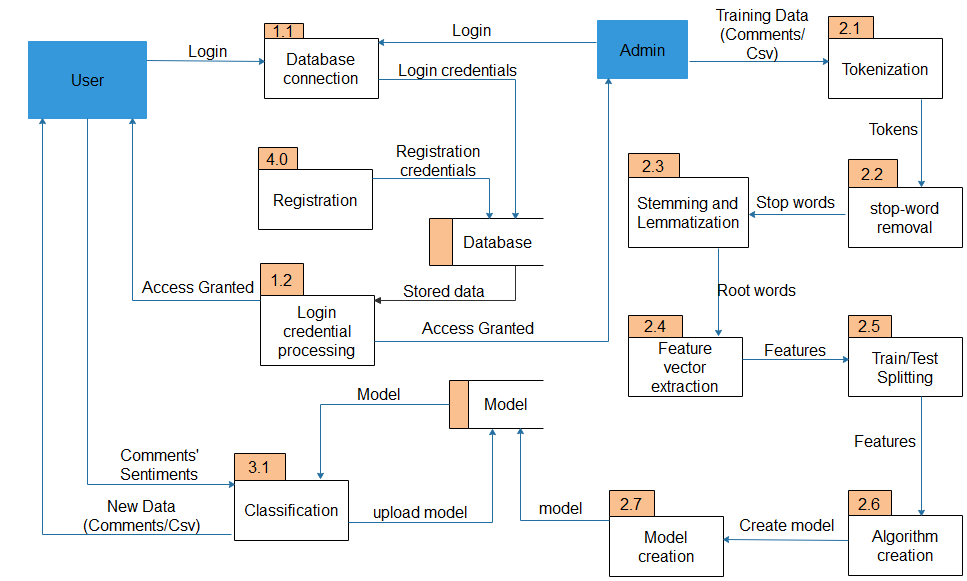


Figure 3.4 Level 2 data flow diagram for SPSC

## 3.3 Design Rationale

We selected client/server architecture because there is a request made by the browser (client) to upload the data. The data is uploaded as the way of servicing the request from the client.

# **4.0 DATA DESIGN**

## 4.1 Data Description

The training data entered by the Admin will flow under the following processes.

**Pre-processing**

Under pre-processing, the data will undergo through a number of sub-processes which will clean the data making it ready for classification. The output of pre-processing is a feature vector.

These sub-processes include, tokenization, stop-word removal, feature vector extraction, stemming and lemmatization.

**Tokenization**

The training data will be tokenized. Tokenization splits longer strings of text into smaller pieces, or tokens. Larger chunks of text will be tokenized into sentences, sentences can be tokenized into words. Further processing will generally be performed after a piece of text has been appropriately tokenized. The tokens are stored in a variable.

**Stop-word removal**

Under this process, “stop words”, the most common words in a language such as “the”, “a”, “on”, “is”, “all” will be removed. This process works on the tokenized data stored in a variable. The data with no stop words is also stored in a variable

**Stemming and Lemmatization**

The data with no stop-words is later transferred for stemming, the process will reduce words to their word stem, base or root form. for example, books — book, looked — look [1].

The aim of lemmatization, like stemming, will be to reduce inflectional forms to a common base form. Lemmatization will not simply chop off inflections. Instead it will use lexical knowledge bases to get the correct base forms of words.

**Feature vector extraction**

The output of this process is feature vector. This process extracts the features of both the comments and the post.

The training and new data features will later be split. The features will be used to build the model. The model is saved in a data store.

The user on the other hand enters new data for classification. The classifier uploads the trained and built model. The uploaded model classifies the new data. The classifier outputs the results. If the output data is not correct, the user updates it. The updated comments are stored into a database. These updated comments are later used to updated the model. This helps the model to learn.

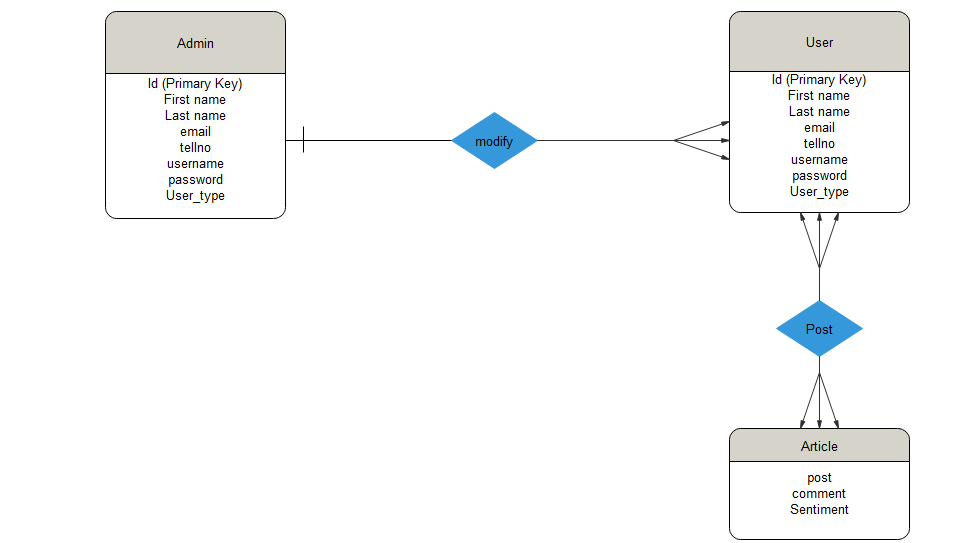


Figure 4.1 Entity Relationship Diagram showing entities in the database

Figure 4.1shows the relationship between database entities. One Admin can modify one or many users and one or many users can be modified by one Admin. One or many users can post one or many articles.

## 4.2 Data Dictionary

Table 1 Table showing entities and their attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Entity type** | **Null** | **UNIQUE** |
| User/admin | ID | INT (PK) | NOTNULL | Yes |
|  | First name | TEXT | NOT NULL | No |
|  | Last name | TEXT | NOT NULL | No |
|  | email | TEXT | NOT NULL | Yes |
|  | tellno | NUMBER | NOT NULL | Yes |
|  | username | TEXT | NOT NULL | Yes |
|  | password | PASSWOR | NOT NULL | No |
|  | User\_type | TEXT | NULL | No |
|  |  |  |  |  |
| Article | post | TEXT | NOT NULL | No |
|  | comment | TEXT | NOT NULL | No |
|  | Sentiment | TEXT | NOT NULL | No |

# **5.0 COMPONENT DESIGN**

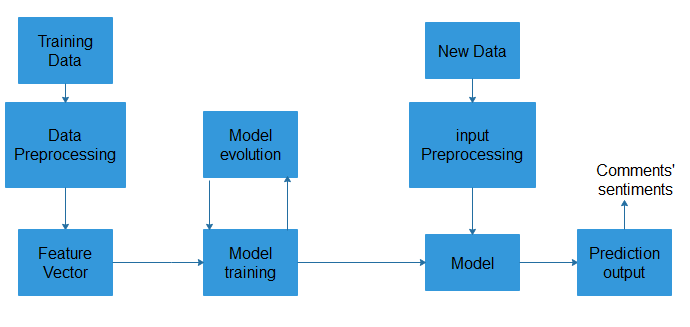


Figure 5.1 Component diagram for SPSC

The classifier will have three components which include Comments/csv, data (training data and testing data) and data processing/classification (train data, classifier, model, classifier algorithm).

**Pseudocode**

**ENTER POST**

If text area empty

Enter the post

Else Proceed to the comment text area

**ENTER COMMENT**

If text area empty

Enter the comment

Else proceed to classify button

**CLASSIFY**

If post and comment entered

Click classify button

Else enter post and comment

# **6.0 HUMAN INTERFACE DESIGN**

## 6.1 Overview of User Interface

The User Interface is a crucial aspect of the system in terms of both what the client wants and needs. For this reason, there is an overview of the User Interface. This section will detail all aspects of the User Interface and its design.

The user will be prompted to input post and its comments to their respective text areas. The user can either first upload the file which contains the posts and comments to be analyzed, or directly enter the post and its relevant comments which need to be classified. Once the user is successful on entering the post and the comments to their specific text areas, he can then click the classify button to display the results (classified comments).

## 6.2 Screen Images

## Home Screen

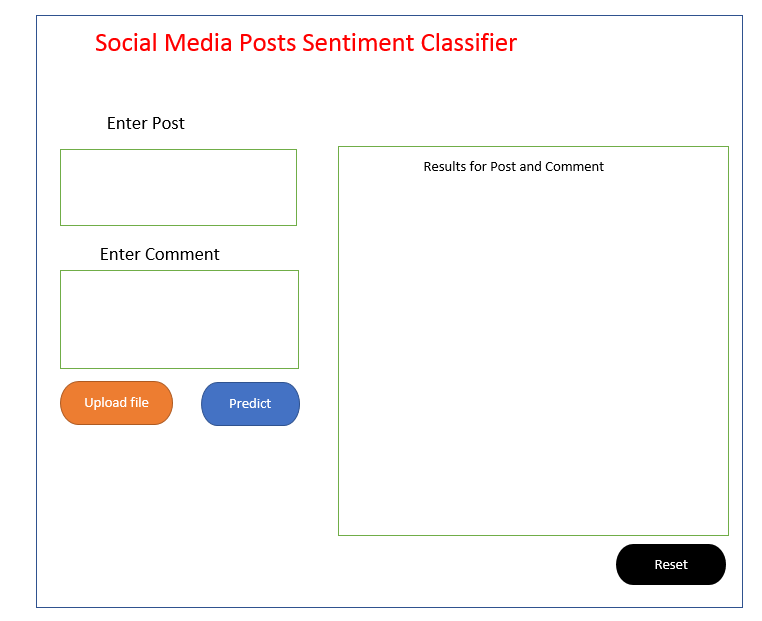


Figure 6.1 Home screen

The figure 6.1 shows what the user interfaces with after running the classifiers’ URL in the browser. It shows the different functionalities of the classifier

**Displaying Results**

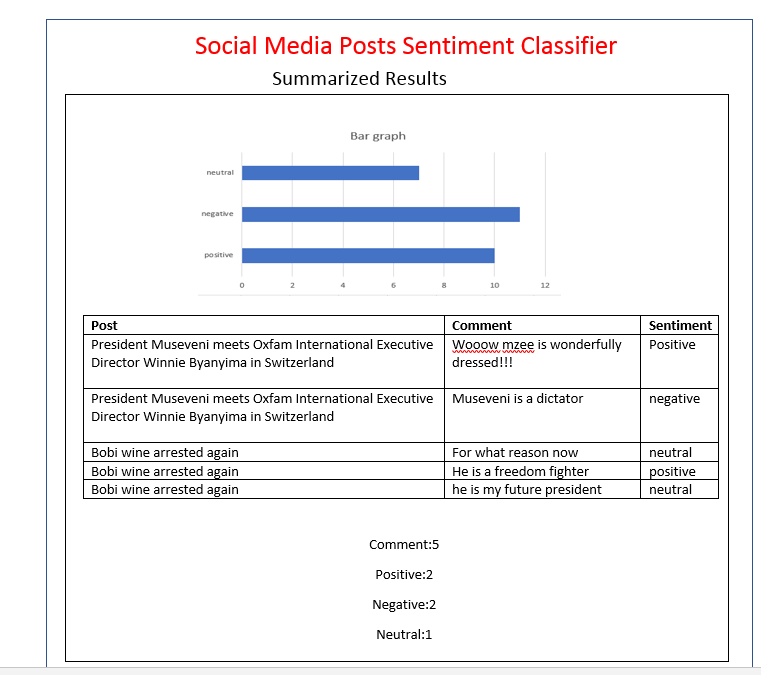


Figure 6.2 summarized Results

Figure 6.2 displays summarized results form the uploaded file data.

**Uploading the file**



Figure 6.3 shows upload file page.

## 6.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

Table 2 : screen objects and their corresponding actions

|  |  |
| --- | --- |
| **Object** | **Action** |
| **Upload file** | This object is a button that will take you to a page from where to upload a csv file that contains data to be analyzed and classified |
| **Predict** | this object initiates the inner operation to perform analysis and classification. It also displays out results. |
| **Reset** | This object will allow the user to clear all the displayed content so as to add a new post and comment for predicting |
| **Choose file** | This will open a window on the machine which prompts the user to choose a csv file that contains data to be classified. |
| **Upload** | This object button uploads the chosen file into the system. It also displays the summarized results. |

# **7.0 REQUIREMENTS MATRIX**

Table 3: Requirements Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Requirement Name** | **UC** | **Description** |
| 1 | Enter data |  | The use case begins when the user loads the browser. The user enters data (post and comments) into the specified text areas. |
| 2 | Classify |  | The use case begins after the user is done inputting data in the specified text areas and then clicks classify button. The use case ends when the user clicks classify button. |
| 3 | Display Classified data |  | The use case begins after the user clicks the classifies button. The use case ends when the results are displayed. |

**Implementation, testing and validation report for  
Social Media Post Classification System**

DOCUMENT NO: 1.0

PREPARED BY: BSE 19-5

DATE: 31-05-2019

VERSION: 1.0

Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Date** | **Signature** |
| BSE 19-5 | Author(s) |  |  |
| NTANDA MOSES | Validation |  |  |
|  | Client |  |  |